Package 'broom'

September 26, 2024

Type Package

Title Convert Statistical Objects into Tidy Tibbles

Version 1.0.7

Description Summarizes key information about statistical objects in tidy tibbles. This makes it easy to report results, create plots and consistently work with large numbers of models at once. Broom provides three verbs that each provide different types of information about a model. tidy() summarizes information about model components such as coefficients of a regression. glance() reports information about an entire model, such as goodness of fit measures like AIC and BIC. augment() adds information about individual observations to a dataset, such as fitted values or influence measures.

License MIT + file LICENSE

URL https://broom.tidymodels.org/, https://github.com/tidymodels/broom

BugReports https://github.com/tidymodels/broom/issues

Depends R (>= 3.5)

Imports backports, dplyr (>= 1.0.0), generics (>= 0.0.2), glue, lifecycle, purrr, rlang, stringr, tibble (>= 3.0.0), tidyr (>= 1.0.0)

Suggests AER, AUC, bbmle, betareg (>= 3.2-1), biglm, binGroup, boot, btergm (>= 1.10.6), car (>= 3.1-2), carData, caret, cluster, cmprsk, coda, covr, drc, e1071, emmeans, epiR, ergm (>= 3.10.4), fixest (>= 0.9.0), gam (>= 1.15), gee, geepack, ggplot2, glmnet, glmnetUtils, gmm, Hmisc, irlba, interp, joineRML, Kendall, knitr, ks, Lahman, lavaan (>= 0.6.18), leaps, lfe, lm.beta, lme4, lmodel2, lmtest (>= 0.9.38), lsmeans, maps, margins, MASS, mclust, mediation, metafor, mfx, mgcv, mlogit, modeldata, modeltests (>= 0.1.6), muhaz, multcomp, network, nnet, orcutt (>= 2.2), ordinal, plm, poLCA, psych, quantreg, rmarkdown, robust, robustbase, rsample, sandwich, spdep (>= 1.1), spatialreg, speedglm, spelling, survey, survival (>= 3.6-4), systemfit, testthat (>= 2.1.0), tseries, vars, zoo

VignetteBuilder knitr

Config/Needs/website tidyverse/tidytemplate

Encoding UTF-8

RoxygenNote 7.3.2

Language en-US

Collate 'aaa-documentation-helper.R' 'null-and-default-tidiers.R' 'aer-tidiers.R' 'auc-tidiers.R' 'base-tidiers.R' 'bbmle-tidiers.R' 'betareg-tidiers.R' 'biglm-tidiers.R' 'bingroup-tidiers.R' 'boot-tidiers.R' 'broom-package.R' 'broom.R' 'btergm-tidiers.R' 'car-tidiers.R' 'caret-tidiers.R' 'cluster-tidiers.R' 'cmprsk-tidiers.R' 'data-frame-tidiers.R' 'deprecated-0-7-0.R' 'drc-tidiers.R' 'emmeans-tidiers.R' 'epiR-tidiers.R' 'ergm-tidiers.R' 'fixest-tidiers.R' 'gam-tidiers.R' 'geepack-tidiers.R' 'glmnet-cv-glmnet-tidiers.R' 'glmnet-glmnet-tidiers.R' 'gmm-tidiers.R' 'hmisc-tidiers.R' 'joinerml-tidiers.R' 'kendall-tidiers.R' 'ks-tidiers.R' 'lavaan-tidiers.R' 'leaps-tidiers.R' 'lfe-tidiers.R' 'list-irlba.R' 'list-optim-tidiers.R' 'list-svd-tidiers.R' 'list-tidiers.R' 'list-xyz-tidiers.R' 'lm-beta-tidiers.R' 'lmodel2-tidiers.R' 'lmtest-tidiers.R' 'maps-tidiers.R' 'margins-tidiers.R' 'mass-fitdistr-tidiers.R' 'mass-negbin-tidiers.R' 'mass-polr-tidiers.R' 'mass-ridgelm-tidiers.R' 'stats-lm-tidiers.R' 'mass-rlm-tidiers.R' 'mclust-tidiers.R' 'mediation-tidiers.R' 'metafor-tidiers.R' 'mfx-tidiers.R' 'mgcv-tidiers.R' 'mlogit-tidiers.R' 'muhaz-tidiers.R' 'multcomp-tidiers.R' 'nnet-tidiers.R' 'nobs.R' 'orcutt-tidiers.R' 'ordinal-clm-tidiers.R' 'ordinal-clmm-tidiers.R' 'plm-tidiers.R' 'polca-tidiers.R' 'psych-tidiers.R' 'stats-nls-tidiers.R' 'quantreg-nlrq-tidiers.R' 'quantreg-rq-tidiers.R' 'quantreg-rqs-tidiers.R' 'robust-glmrob-tidiers.R' 'robust-lmrob-tidiers.R' 'robustbase-glmrob-tidiers.R' 'robustbase-lmrob-tidiers.R' 'sp-tidiers.R' 'spdep-tidiers.R' 'speedglm-speedglm-tidiers.R' 'speedglm-speedlm-tidiers.R' 'stats-anova-tidiers.R' 'stats-arima-tidiers.R' 'stats-decompose-tidiers.R' 'stats-factanal-tidiers.R' 'stats-glm-tidiers.R' 'stats-htest-tidiers.R' 'stats-kmeans-tidiers.R' 'stats-loess-tidiers.R' 'stats-mlm-tidiers.R' 'stats-prcomp-tidiers.R' 'stats-smooth.spline-tidiers.R' 'stats-summary-lm-tidiers.R' 'stats-time-series-tidiers.R' 'survey-tidiers.R' 'survival-aareg-tidiers.R' 'survival-cch-tidiers.R' 'survival-coxph-tidiers.R' 'survival-pyears-tidiers.R' 'survival-survdiff-tidiers.R' 'survival-survexp-tidiers.R' 'survival-survfit-tidiers.R' 'survival-survreg-tidiers.R' 'systemfit-tidiers.R' 'tseries-tidiers.R' 'utilities.R'

NeedsCompilation no

```
Author David Robinson [aut],
       Alex Hayes [aut] (<a href="https://orcid.org/0000-0002-4985-5160">https://orcid.org/0000-0002-4985-5160</a>),
       Simon Couch [aut, cre] (<a href="https://orcid.org/0000-0001-5676-5107">https://orcid.org/0000-0001-5676-5107</a>),
       Posit Software, PBC [cph, fnd],
       Indrajeet Patil [ctb] (<a href="https://orcid.org/0000-0003-1995-6531">https://orcid.org/0000-0003-1995-6531</a>),
       Derek Chiu [ctb],
       Matthieu Gomez [ctb],
       Boris Demeshev [ctb],
       Dieter Menne [ctb],
       Benjamin Nutter [ctb],
       Luke Johnston [ctb],
       Ben Bolker [ctb],
       François Briatte [ctb],
       Jeffrey Arnold [ctb],
       Jonah Gabry [ctb],
       Luciano Selzer [ctb],
       Gavin Simpson [ctb],
       Jens Preussner [ctb],
       Jay Hesselberth [ctb],
       Hadley Wickham [ctb],
       Matthew Lincoln [ctb],
       Alessandro Gasparini [ctb],
       Lukasz Komsta [ctb],
       Frederick Novometsky [ctb],
       Wilson Freitas [ctb],
       Michelle Evans [ctb],
       Jason Cory Brunson [ctb],
       Simon Jackson [ctb],
       Ben Whalley [ctb],
       Karissa Whiting [ctb],
       Yves Rosseel [ctb],
       Michael Kuehn [ctb],
       Jorge Cimentada [ctb],
       Erle Holgersen [ctb],
       Karl Dunkle Werner [ctb] (<a href="https://orcid.org/0000-0003-0523-7309">https://orcid.org/0000-0003-0523-7309</a>),
       Ethan Christensen [ctb],
       Steven Pav [ctb],
       Paul PJ [ctb],
       Ben Schneider [ctb],
       Patrick Kennedy [ctb],
       Lily Medina [ctb],
       Brian Fannin [ctb],
       Jason Muhlenkamp [ctb],
       Matt Lehman [ctb],
       Bill Denney [ctb] (<a href="https://orcid.org/0000-0002-5759-428X">https://orcid.org/0000-0002-5759-428X</a>),
       Nic Crane [ctb],
```

```
Andrew Bates [ctb],
Vincent Arel-Bundock [ctb] (<a href="https://orcid.org/0000-0003-2042-7063">https://orcid.org/0000-0003-2042-7063</a>),
Hideaki Hayashi [ctb],
Luis Tobalina [ctb],
Annie Wang [ctb],
Wei Yang Tham [ctb],
Clara Wang [ctb],
Abby Smith [ctb] (<https://orcid.org/0000-0002-3207-0375>),
Jasper Cooper [ctb] (<a href="https://orcid.org/0000-0002-8639-3188">https://orcid.org/0000-0002-8639-3188</a>),
E Auden Krauska [ctb] (<a href="https://orcid.org/0000-0002-1466-5850">https://orcid.org/0000-0002-1466-5850</a>),
Alex Wang [ctb],
Malcolm Barrett [ctb] (<https://orcid.org/0000-0003-0299-5825>),
Charles Gray [ctb] (<a href="https://orcid.org/0000-0002-9978-011X">https://orcid.org/0000-0002-9978-011X</a>),
Jared Wilber [ctb],
Vilmantas Gegzna [ctb] (<a href="https://orcid.org/0000-0002-9500-5167">https://orcid.org/0000-0002-9500-5167</a>),
Eduard Szoecs [ctb],
Frederik Aust [ctb] (<a href="https://orcid.org/0000-0003-4900-788X">https://orcid.org/0000-0003-4900-788X</a>),
Angus Moore [ctb],
Nick Williams [ctb],
Marius Barth [ctb] (<a href="https://orcid.org/0000-0002-3421-6665">https://orcid.org/0000-0002-3421-6665>),
Bruna Wundervald [ctb] (<a href="https://orcid.org/0000-0001-8163-220X">https://orcid.org/0000-0001-8163-220X</a>),
Joyce Cahoon [ctb] (<a href="https://orcid.org/0000-0001-7217-4702">https://orcid.org/0000-0001-7217-4702</a>),
Grant McDermott [ctb] (<a href="https://orcid.org/0000-0001-7883-8573">https://orcid.org/0000-0001-7883-8573</a>),
Kevin Zarca [ctb],
Shiro Kuriwaki [ctb] (<a href="https://orcid.org/0000-0002-5687-2647">https://orcid.org/0000-0002-5687-2647</a>),
Lukas Wallrich [ctb] (<a href="https://orcid.org/0000-0003-2121-5177">https://orcid.org/0000-0003-2121-5177</a>),
James Martherus [ctb] (<a href="https://orcid.org/0000-0002-8285-3300">https://orcid.org/0000-0002-8285-3300</a>),
Chuliang Xiao [ctb] (<a href="https://orcid.org/0000-0002-8466-9398">https://orcid.org/0000-0002-8466-9398</a>),
Joseph Larmarange [ctb],
Max Kuhn [ctb],
Michal Bojanowski [ctb],
Hakon Malmedal [ctb],
Clara Wang [ctb],
Sergio Oller [ctb],
Luke Sonnet [ctb],
Jim Hester [ctb],
Ben Schneider [ctb],
Bernie Gray [ctb] (<https://orcid.org/0000-0001-9190-6032>),
Mara Averick [ctb],
Aaron Jacobs [ctb],
Andreas Bender [ctb],
Sven Templer [ctb],
Paul-Christian Buerkner [ctb],
Matthew Kay [ctb],
Erwan Le Pennec [ctb],
Johan Junkka [ctb],
Hao Zhu [ctb],
Benjamin Soltoff [ctb],
```

Zoe Wilkinson Saldana [ctb],
Tyler Littlefield [ctb],
Charles T. Gray [ctb],
Shabbh E. Banks [ctb],
Serina Robinson [ctb],
Roger Bivand [ctb],
Riinu Ots [ctb],
Nicholas Williams [ctb],
Nina Jakobsen [ctb],
Michael Weylandt [ctb],
Lisa Lendway [ctb],
Karl Hailperin [ctb],
Josue Rodriguez [ctb],
Jenny Bryan [ctb],
Chris Jarvis [ctb],
Greg Macfarlane [ctb],
Brian Mannakee [ctb],
Drew Tyre [ctb],
Shreyas Singh [ctb],
Laurens Geffert [ctb],
Hong Ooi [ctb],
Henrik Bengtsson [ctb],
Eduard Szocs [ctb],
David Hugh-Jones [ctb],
Matthieu Stigler [ctb],
Hugo Tavares [ctb] (https://orcid.org/0000-0001-9373-2726),
R. Willem Vervoort [ctb],
Brenton M. Wiernik [ctb],
Josh Yamamoto [ctb], Jasme Lee [ctb],
Taren Sanders [ctb] (<https: 0000-0002-4504-6008="" orcid.org="">),</https:>
Ilaria Prosdocimi [ctb] (https://orcid.org/0000-0001-8565-094X),
Daniel D. Sjoberg [ctb] (https://orcid.org/0000-0003-0862-2018),
Alex Reinhart [ctb] (<https: 0000-0002-6658-514x="" orcid.org="">)</https:>
Maintainer Simon Couch <simon.couch@posit.co></simon.couch@posit.co>
Repository CRAN
Date/Publication 2024-09-26 21:00:13 UTC
Contents
augment.betamfx
augment.betareg
augment.clm
augment.coxph
augment.decomposed.ts
augment.drc
augment.factanal

ugment.felm		26
ugment.fixest		28
ugment.gam		31
ugment.glm		33
ugment.glmRob		35
ugment.glmrob		36
ugment.htest		38
ugment.ivreg		40
ugment.kmeans		42
ugment.lm		44
ugment.lmRob		48
ugment.lmrob		50
ugment.loess		52
ugment.Mclust		54
ugment.mfx		56
ugment.mjoint		59
ugment.mlogit		62
ugment.nlrq		64
ugment.nls		65
ugment.pam		67
ugment.plm		69
ugment.poLCA		71
		73
		75
ugment.prcomp		77
ugment.rlm		79
ugment.rma		
ugment.rq		8183
ugment.rqs		
ugment.sarlm		86
ugment.smooth.spline		88
ugment.speedlm		89
ugment.stl		91
ugment.survreg		92
ugment_columns		95
ootstrap		96
onfint_tidy		97
ata.frame_tidiers		98
urbinWatsonTest_tidiers		
inish_glance		
ix_data_frame		
lance.aareg		
lance.anova		
lance.aov	1	05
lance.Arima		
lance.betamfx		
lance.betareg		
lance.biglm	1	11
lance.binDesign	1	13

glance.cch	
glance.clm	116
glance.clmm	118
glance.coeftest	120
glance.coxph	122
glance.crr	124
glance.cv.glmnet	126
glance.drc	
glance.ergm	
glance factanal	
glance.felm	
glance.fitdistr	
glance.fixest	
glance.Gam	
glance.gam	
glance.garch	
glance.geeglm	
glance.glm	
glance.glmnet	
glance.glmRob	
glance.gmm	
glance.ivreg	
glance.kmeans	
glance.lavaan	
glance.lm	
glance.lmodel2	
glance.lmRob	
glance.lmrob	
glance.margins	
glance.Mclust	
glance.mfx	
glance.mjoint	
glance.mlogit	
glance.muhaz	
glance.multinom	174
glance.negbin	176
glance.nlrq	177
glance.nls	179
glance.orcutt	180
glance.pam	182
glance.plm	183
glance.poLCA	
glance.polr	
glance.pyears	
glance.ridgelm	
glance.rlm	
glance.rma	
glance.rq	

glance.sarlm	 198
glance.smooth.spline	 200
glance.speedglm	 201
glance.speedlm	 203
glance.summary.lm	 204
glance.survdiff	 207
glance.survexp	 209
glance.survfit	 210
glance.survreg	 212
glance.svyglm	 214
glance.svyolr	 216
glance.varest	 217
glance_optim	 219
leveneTest_tidiers	 220
list_tidiers	 221
null_tidiers	 222
sp_tidiers	 222
summary_tidiers	 223
tidy.aareg	 225
tidy.acf	 226
tidy.anova	 227
tidy.aov	 229
tidy.aovlist	 230
tidy.Arima	
tidy.betamfx	 232
tidy.betareg	
tidy.biglm	
tidy.binDesign	
tidy.binWidth	
tidy.boot	
tidy.btergm	
tidy.cch	
tidy.cld	
tidy.clm	
tidy.clmm	
tidy.coeftest	
tidy.confint.glht	
tidy.confusionMatrix	
tidy.coxph	
tidy.crr	
tidy.cv.glmnet	
tidy.density	
tidy.dist	
tidy.drc	
tidy.emmGrid	
tidy.epi.2by2	
tidy.ergm	
tidy.factanal	
	 · · -/1

tidy.felm	272
tidy.fitdistr	274
tidy.fixest	276
tidy.ftable	278
tidy.Gam	278
tidy.gam	
tidy.garch	
tidy.geeglm	
tidy.glht	
tidy.glm	
tidy.glmnet	
tidy.glmRob	
tidy.glmrob	
tidy.gmm	
tidy.htest	
tidy.ivreg	
•	
tidy.kappa	
tidy.kde	
tidy.Kendall	
tidy.kmeans	
tidy.lavaan	
tidy.lm	
tidy.lm.beta	
tidy.lmodel2	
tidy.lmRob	
tidy.lmrob	
tidy.lsmobj	
tidy.manova	316
tidy.map	318
tidy.margins	319
tidy.Mclust	321
tidy.mediate	323
tidy.mfx	325
tidy.mjoint	327
tidy.mle2	
tidy.mlm	
tidy.mlogit	333
tidy.muhaz	
tidy.multinom	
tidy.negbin	
tidy.nlrq	
tidy.nls	
tidy.numeric	
tidy.orcutt	
·	
tidy.pairwise.htest	
tidy.pam	
tidy.plm	
tidy.poLCA	348

10 augment.betamfx

tio	y.polr	50
tio	y.power.htest	52
tio	y.prcomp	53
tio	y.pyears	55
tio	y.rcorr	57
tio	y.ref.grid	59
tio	y.regsubsets	61
tio	y.ridgelm	52
tio	y.rlm	53
tio	y.rma	54
tio	y.roc	6
tio	y.rq	8
tio	y.rqs	9
tio	y.sarlm	1
tio	y.spec	13
tio	y.speedglm	14
	y.speedlm	
	y.summary.glht	
	y.summary.lm	
	y.summary_emm	
	y.survdiff	
	y.survexp	
	y.survfit	
	y.survreg	
	y.svyglm	
	y.svyolr	
	y.systemfit	
	y.table	
	y.ts	
	y.TukeyHSD	
	y.varest	
	y.zoo	
	y_irlba	
	y_optim	
	y_svd	
tio	y_xyz)5
Index	40)7

augment.betamfx 11

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

Arguments

x A betamfx object.

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

12 augment.betamfx

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Character indicating type of prediction to use. Passed to the type argument of

betareg::predict.betareg(). Defaults to "response".

type.residuals Character indicating type of residuals to use. Passed to the type argument of

betareg::residuals.betareg(). Defaults to "sweighted2.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This augment method wraps augment.betareg() for mfx::betamfx() objects.

Value

```
A tibble::tibble() with columns:
```

. cooksd Cooks distance.

.fitted Fitted or predicted value.

resid The difference between observed and fitted values.

See Also

```
augment.betareg(), mfx::betamfx()
Other mfx tidiers: augment.mfx(), glance.betamfx(), glance.mfx(), tidy.betamfx(), tidy.mfx()
```

Examples

```
library(mfx)

# Simulate some data
set.seed(12345)
n <- 1000
x <- rnorm(n)

# Beta outcome
y <- rbeta(n, shape1 = plogis(1 + 0.5 * x), shape2 = (abs(0.2 * x)))
# Use Smithson and Verkuilen correction
y <- (y * (n - 1) + 0.5) / n</pre>
```

augment.betareg 13

```
d <- data.frame(y, x)
mod_betamfx <- betamfx(y ~ x | x, data = d)

tidy(mod_betamfx, conf.int = TRUE)

# Compare with the naive model coefficients of the equivalent betareg call (not run)
# tidy(betamfx(y ~ x | x, data = d), conf.int = TRUE)

augment(mod_betamfx)
glance(mod_betamfx)</pre>
```

augment.betareg

Augment data with information from a(n) betareg object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'betareg'
augment(
    x,
```

14 augment.betareg

```
data = model.frame(x),
  newdata = NULL,
  type.predict,
  type.residuals,
   ...
)
```

Arguments

x A betareg object produced by a call to betareg::betareg().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Character indicating type of prediction to use. Passed to the type argument of

the stats::predict() generic. Allowed arguments vary with model class, so

be sure to read the predict.my_class documentation.

type.residuals Character indicating type of residuals to use. Passed to the type argument of

stats::residuals() generic. Allowed arguments vary with model class, so

be sure to read the residuals.my_class documentation.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For additional details on Cook's distance, see stats::cooks.distance().

Value

A tibble::tibble() with columns:

. cooksd Cooks distance.

.fitted Fitted or predicted value.

resid The difference between observed and fitted values.

augment.clm 15

See Also

```
augment(), betareg::betareg()
```

Examples

```
# load libraries for models and data
library(betareg)

# load dats
data("GasolineYield", package = "betareg")

# fit model
mod <- betareg(yield ~ batch + temp, data = GasolineYield)

mod

# summarize model fit with tidiers
tidy(mod)
tidy(mod, conf.int = TRUE)
tidy(mod, conf.int = TRUE, conf.level = .99)
augment(mod)
glance(mod)</pre>
```

augment.clm

Augment data with information from a(n) clm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

16 augment.clm

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'clm'
augment(
    x,
    data = model.frame(x),
    newdata = NULL,
    type.predict = c("prob", "class"),
    ...
)
```

Arguments

x A clm object returned from ordinal::clm().

data A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that

augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Which type of prediction to compute, either "prob" or "class", passed to

ordinal::predict.clm(). Defaults to "prob".

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

See Also

```
tidy, ordinal::clm(), ordinal::predict.clm()
Other ordinal tidiers: augment.polr(), glance.clm(), glance.clmm(), glance.polr(), glance.svyolr(),
tidy.clm(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

augment.coxph 17

Examples

```
# load libraries for models and data
library(ordinal)

# fit model
fit <- clm(rating ~ temp * contact, data = wine)

# summarize model fit with tidiers
tidy(fit)
tidy(fit, conf.int = TRUE, conf.level = 0.9)
tidy(fit, conf.int = TRUE, conf.type = "Wald", exponentiate = TRUE)
glance(fit)
augment(fit, type.predict = "prob")
augment(fit, type.predict = "class")

# ...and again with another model specification
fit2 <- clm(rating ~ temp, nominal = ~contact, data = wine)
tidy(fit2)
glance(fit2)</pre>
```

augment.coxph

Augment data with information from a(n) coxph object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters

18 augment.coxph

the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## $3 method for class 'coxph'
augment(
    x,
    data = model.frame(x),
    newdata = NULL,
    type.predict = "lp",
    type.residuals = "martingale",
    ...
)
```

Arguments

A coxph object returned from survival::coxph().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Character indicating type of prediction to use. Passed to the type argument of

the stats::predict() generic. Allowed arguments vary with model class, so

be sure to read the predict.my_class documentation.

type.residuals Character indicating type of residuals to use. Passed to the type argument of stats::residuals() generic. Allowed arguments vary with model class, so

be sure to read the residuals.my_class documentation.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

augment.coxph 19

Details

When the modeling was performed with na.action = "na.omit" (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with na.action = "na.exclude", one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to augment() and na.action = "na.exclude", a warning is raised and the incomplete rows are dropped.

Value

```
A tibble::tibble() with columns:

.fitted Fitted or predicted value.

.resid The difference between observed and fitted values.
.se.fit Standard errors of fitted values.
```

See Also

```
stats::na.action
augment(), survival::coxph()
Other coxph tidiers: glance.coxph(), tidy.coxph()
Other survival tidiers: augment.survreg(), glance.aareg(), glance.cch(), glance.coxph(),
glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

Examples

```
# load libraries for models and data
library(survival)

# fit model
cfit <- coxph(Surv(time, status) ~ age + sex, lung)

# summarize model fit with tidiers
tidy(cfit)
tidy(cfit, exponentiate = TRUE)

lp <- augment(cfit, lung)
risks <- augment(cfit, lung, type.predict = "risk")
expected <- augment(cfit, lung, type.predict = "expected")

glance(cfit)

# also works on clogit models
resp <- levels(logan$occupation)
n <- nrow(logan)
indx <- rep(1:n, length(resp))</pre>
```

```
logan2 <- data.frame(</pre>
  logan[indx, ],
  id = indx,
  tocc = factor(rep(resp, each = n))
)
logan2$case <- (logan2$occupation == logan2$tocc)</pre>
cl <- clogit(case ~ tocc + tocc:education + strata(id), logan2)</pre>
tidy(cl)
glance(cl)
library(ggplot2)
ggplot(lp, aes(age, .fitted, color = sex)) +
  geom_point()
ggplot(risks, aes(age, .fitted, color = sex)) +
  geom_point()
ggplot(expected, aes(time, .fitted, color = sex)) +
  geom_point()
```

augment.decomposed.ts Augment data with information from a(n) decomposed.ts object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters

the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'decomposed.ts'
augment(x, ...)
```

Arguments

- x A decomposed.ts object returned from stats::decompose().
- Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:
 - tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
 - augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble with one row for each observation in the original times series:

. seasonal The seasonal component of the decomposition.
. trend The trend component of the decomposition.

. remainder The remainder, or "random" component of the decomposition.

.weight The final robust weights (stl only).

. seasadj The seasonally adjusted (or "deseasonalised") series.

See Also

```
augment(), stats::decompose()
Other decompose tidiers: augment.stl()
```

Examples

```
# time series of temperatures in Nottingham, 1920-1939:
nottem

# perform seasonal decomposition on the data with both decompose
# and stl:
d1 <- decompose(nottem)
d2 <- stl(nottem, s.window = "periodic", robust = TRUE)</pre>
```

22 augment.drc

```
# compare the original series to its decompositions.
cbind(
 tidy(nottem), augment(d1),
 augment(d2)
)
# visually compare seasonal decompositions in tidy data frames.
library(tibble)
library(dplyr)
library(tidyr)
library(ggplot2)
decomps <- tibble(</pre>
 # turn the ts objects into data frames.
 series = list(as.data.frame(nottem), as.data.frame(nottem)),
 # add the models in, one for each row.
 decomp = c("decompose", "stl"),
 model = list(d1, d2)
) %>%
 rowwise() %>%
 # pull out the fitted data using broom::augment.
 mutate(augment = list(broom::augment(model))) %>%
 ungroup() %>%
 # unnest the data frames into a tidy arrangement of
 # the series next to its seasonal decomposition, grouped
 # by the method (stl or decompose).
 group_by(decomp) %>%
 unnest(c(series, augment)) %>%
 mutate(index = 1:n()) %>%
 ungroup() %>%
 select(decomp, index, x, adjusted = .seasadj)
ggplot(decomps) +
 geom\_line(aes(x = index, y = x), colour = "black") +
 geom_line(aes(
   x = index, y = adjusted, colour = decomp,
   group = decomp
 ))
```

augment.drc

Augment data with information from a(n) drc object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in

augment.drc 23

the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'drc'
augment(
    x,
    data = NULL,
    newdata = NULL,
    se_fit = FALSE,
    conf.int = FALSE,
    conf.level = 0.95,
    ...
)
```

Arguments

data

x A drc object produced by a call to drc::drm().

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

newdata

24 augment.drc

se_fit Logical indicating whether or not a .se.fit column should be added to the augmented output. For some models, this calculation can be somewhat time-consuming. Defaults to FALSE.

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

Conf.level The confidence level to use for the confidence interval if conf. int = TRUE. Must

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

.cooksd
.fitted
.fitted or predicted value.
.lower
.lower bound on interval for fitted values.
.resid
.se.fit
Standard errors of fitted values.
.upper
Upper bound on interval for fitted values.

See Also

```
augment(), drc::drm()
Other drc tidiers: glance.drc(), tidy.drc()
```

Examples

```
# load libraries for models and data
library(drc)

# fit model
mod <- drm(dead / total ~ conc, type,
   weights = total, data = selenium, fct = LL.2(), type = "binomial"
)

# summarize model fit with tidiers
tidy(mod)</pre>
```

augment.factanal 25

```
tidy(mod, conf.int = TRUE)
glance(mod)
augment(mod, selenium)
```

augment.factanal

Augment data with information from a(n) factanal object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'factanal'
augment(x, data, ...)
```

Arguments

x A factanal object created by stats::factanal().

26 augment.felm

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

When data is not supplied augment. factanal returns one row for each observation, with a factor score column added for each factor X, (.fsX). This is because stats::factanal(), unlike other stats methods like stats::lm(), does not retain the original data.

When data is supplied, augment.factanal returns one row for each observation, with a factor score column added for each factor X, (.fsX).

See Also

```
augment(), stats::factanal()
Other factanal tidiers: glance.factanal(), tidy.factanal()
```

augment.felm

Augment data with information from a(n) felm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

augment.felm 27

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'felm'
augment(x, data = model.frame(x), ...)
```

Arguments

Х

A felm object returned from lfe::felm().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

.fitted or predicted value.

resid The difference between observed and fitted values.

See Also

```
augment(), 1fe::felm()
Other felm tidiers: tidy.felm()
```

28 augment.fixest

Examples

```
# load libraries for models and data
library(lfe)
# use built-in `airquality` dataset
head(airquality)
# no FEs; same as lm()
est0 <- felm(Ozone ~ Temp + Wind + Solar.R, airquality)</pre>
# summarize model fit with tidiers
tidy(est0)
augment(est0)
# add month fixed effects
est1 <- felm(Ozone ~ Temp + Wind + Solar.R | Month, airquality)</pre>
# summarize model fit with tidiers
tidy(est1)
tidy(est1, fe = TRUE)
augment(est1)
glance(est1)
# the "se.type" argument can be used to switch out different standard errors
# types on the fly. In turn, this can be useful exploring the effect of
# different error structures on model inference.
tidy(est1, se.type = "iid")
tidy(est1, se.type = "robust")
# add clustered SEs (also by month)
est2 <- felm(Ozone ~ Temp + Wind + Solar.R | Month | 0 | Month, airquality)</pre>
# summarize model fit with tidiers
tidy(est2, conf.int = TRUE)
tidy(est2, conf.int = TRUE, se.type = "cluster")
tidy(est2, conf.int = TRUE, se.type = "robust")
tidy(est2, conf.int = TRUE, se.type = "iid")
```

augment.fixest

Augment data with information from a(n) fixest object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

augment.fixest 29

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'fixest'
augment(
    x,
    data = NULL,
    newdata = NULL,
    type.predict = c("link", "response"),
    type.residuals = c("response", "deviance", "pearson", "working"),
    ...
)
```

Arguments

x	A fixest object returned from any of the fixest estimators
data	A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. Do not pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.
newdata	A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.
type.predict	Passed to predict.fixest type argument. Defaults to "link" (like predict.glm).
type.residuals	Passed to predict.fixest type argument. Defaults to "response" (like residuals.lm, but unlike residuals.glm).

30 augment.fixest

... Additional arguments passed to summary and confint. Important arguments are se and cluster. Other arguments are dof, exact_dof, forceCovariance, and keepBounded. See summary.fixest.

Value

```
A tibble::tibble() with columns:
```

.fitted or predicted value.

resid The difference between observed and fitted values.

Note

Important note: fixest models do not include a copy of the input data, so you must provide it manually.

```
augment.fixest only works for fixest::feols(), fixest::feglm(), and fixest::femlm() models. It does not work with results from fixest::fenegbin(), fixest::feNmlm(), or fixest::fepois().
```

See Also

```
augment(), fixest::feglm(), fixest::femlm(), fixest::feols()
Other fixest tidiers: tidy.fixest()
```

Examples

```
# load libraries for models and data
library(fixest)

gravity <-
    feols(
        log(Euros) ~ log(dist_km) | Origin + Destination + Product + Year, trade
    )

tidy(gravity)
glance(gravity)
augment(gravity, trade)

# to get robust or clustered SEs, users can either:

# 1) specify the arguments directly in the `tidy()` call
tidy(gravity, conf.int = TRUE, cluster = c("Product", "Year"))

tidy(gravity, conf.int = TRUE, se = "threeway")

# 2) or, feed tidy() a summary.fixest object that has already accepted
# these arguments
gravity_summ <- summary(gravity, cluster = c("Product", "Year"))</pre>
```

augment.gam 31

```
tidy(gravity_summ, conf.int = TRUE)
# approach (1) is preferred.
```

augment.gam

Augment data with information from a(n) gam object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## $3 method for class 'gam'
augment(
    x,
    data = model.frame(x),
    newdata = NULL,
    type.predict,
    type.residuals,
    ...
)
```

32 augment.gam

Arguments

x A gam object returned from a call to mgcv::gam().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Character indicating type of prediction to use. Passed to the type argument of

the stats::predict() generic. Allowed arguments vary with model class, so be sure to read the predict.my_class documentation.

type.residuals Character indicating type of residuals to use. Passed to the type argument of

stats::residuals() generic. Allowed arguments vary with model class, so

be sure to read the residuals.my_class documentation.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For additional details on Cook's distance, see stats::cooks.distance().

Value

A tibble::tibble() with columns:

. cooksd Cooks distance.

.fitted Fitted or predicted value..hat Diagonal of the hat matrix.

resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

. sigma Estimated residual standard deviation when corresponding observation is dropped

from model.

See Also

```
augment(), mgcv::gam()
```

augment.glm 33

Examples

```
# load libraries for models and data
library(mgcv)

# fit model
g <- gam(mpg ~ s(hp) + am + qsec, data = mtcars)

# summarize model fit with tidiers
tidy(g)
tidy(g, parametric = TRUE)
glance(g)
augment(g)</pre>
```

augment.glm

Augment data with information from a(n) glm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

34 augment.glm

Usage

```
## S3 method for class 'glm'
augment(
 х,
 data = model.frame(x),
  newdata = NULL,
  type.predict = c("link", "response", "terms"),
  type.residuals = c("deviance", "pearson"),
  se_fit = FALSE,
)
```

Arguments

A glm object returned from stats::glm(). Х

data A base::data.frame or tibble::tibble() containing the original data that was

> used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

> dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

Passed to stats::predict.glm() type argument. Defaults to "link". type.predict

type.residuals Passed to stats::residuals.glm() and to stats::rstandard.glm() type

arguments. Defaults to "deviance".

Logical indicating whether or not a .se.fit column should be added to the augmented output. For some models, this calculation can be somewhat time-

consuming. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

If the weights for any of the observations in the model are 0, then columns ".infl" and ".hat" in the result will be 0 for those observations.

A . resid column is not calculated when data is specified via the newdata argument.

se_fit

augment.glmRob 35

Value

A tibble::tibble() with columns:

. cooksd Cooks distance.

.fitted Fitted or predicted value..hat Diagonal of the hat matrix.

resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

. sigma Estimated residual standard deviation when corresponding observation is dropped

from model.

.std.resid Standardised residuals.

See Also

```
stats::glm()
```

```
Other lm tidiers: augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(), glance.svyglm(), tidy.glm(), tidy.lm(), tidy.lm(), tidy.lm(), tidy.summary.lm()
```

augment.glmRob

Augment data with information from a(n) glmRob object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

36 augment.glmrob

Usage

```
## S3 method for class 'glmRob' augment(x, \ldots)
```

Arguments

x Unused.... Unused.

augment.glmrob

Augment data with information from a(n) glmrob object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'glmrob'
augment(
    x,
    data = model.frame(x),
    newdata = NULL,
    type.predict = c("link", "response"),
```

augment.glmrob 37

```
type.residuals = c("deviance", "pearson"),
  se_fit = FALSE,
    ...
)
```

Arguments

x A glmrob object returned from robustbase::glmrob().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Character indicating type of prediction to use. Passed to the type argument of

the stats::predict() generic. Allowed arguments vary with model class, so

be sure to read the $predict.my_class$ documentation.

type.residuals Character indicating type of residuals to use. Passed to the type argument of

stats::residuals() generic. Allowed arguments vary with model class, so

be sure to read the $residuals.my_class$ documentation.

se_fit Logical indicating whether or not a .se.fit column should be added to the

augmented output. For some models, this calculation can be somewhat time-

consuming. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will

be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

Value

A tibble::tibble() with columns:

.fitted or predicted value.

resid The difference between observed and fitted values.

38 augment.htest

See Also

```
robustbase::glmrob()
Other robustbase tidiers: augment.lmrob(), glance.lmrob(), tidy.glmrob(), tidy.lmrob()
```

Examples

```
if (requireNamespace("robustbase", quietly = TRUE)) {
 # load libraries for models and data
 library(robustbase)
 data(coleman)
 set.seed(0)
 m <- lmrob(Y ~ ., data = coleman)</pre>
 tidy(m)
 augment(m)
 glance(m)
 data(carrots)
 Rfit <- glmrob(cbind(success, total - success) ~ logdose + block,
   family = binomial, data = carrots, method = "Mqle",
   control = glmrobMqle.control(tcc = 1.2)
 )
 tidy(Rfit)
 augment(Rfit)
}
```

augment.htest

Augment data with information from a(n) htest object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

augment.htest 39

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'htest'
augment(x, ...)
```

Arguments

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

See stats::chisq.test() for more details on how residuals are computed.

Value

A tibble::tibble() with exactly one row and columns:

. observed Observed count.
.prop Proportion of the total.
.row.prop Row proportion (2 dimensions table only).
.col.prop Column proportion (2 dimensions table only).
.expected Expected count under the null hypothesis.
.resid Pearson residuals.
.std.resid Standardized residual.

40 augment.ivreg

See Also

```
augment(), stats::chisq.test()
Other htest tidiers: tidy.htest(), tidy.pairwise.htest(), tidy.power.htest()
```

Examples

```
tt <- t.test(rnorm(10))
tidy(tt)

# the glance output will be the same for each of the below tests
glance(tt)

tt <- t.test(mpg ~ am, data = mtcars)

tidy(tt)

wt <- wilcox.test(mpg ~ am, data = mtcars, conf.int = TRUE, exact = FALSE)

tidy(wt)

ct <- cor.test(mtcars$wt, mtcars$mpg)

tidy(ct)

chit <- chisq.test(xtabs(Freq ~ Sex + Class, data = as.data.frame(Titanic)))

tidy(chit)
augment(chit)</pre>
```

augment.ivreg

Augment data with information from a(n) ivreg object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

augment.ivreg 41

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'ivreg'
augment(x, data = model.frame(x), newdata = NULL, ...)
```

Arguments

Х

An ivreg object created by a call to AER::ivreg().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

newdata

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This tidier currently only supports ivreg-classed objects outputted by the AER package. The ivreg package also outputs objects of class ivreg, and will be supported in a later release.

42 augment.kmeans

Value

```
A tibble::tibble() with columns:

.fitted Fitted or predicted value.

.resid The difference between observed and fitted values.
```

See Also

```
augment(), AER::ivreg()
Other ivreg tidiers: glance.ivreg(), tidy.ivreg()
```

Examples

```
# load libraries for models and data
library(AER)
# load data
data("CigarettesSW", package = "AER")
# fit model
ivr <- ivreg(</pre>
 log(packs) ~ income | population,
 data = CigarettesSW,
 subset = year == "1995"
)
# summarize model fit with tidiers
tidy(ivr)
tidy(ivr, conf.int = TRUE)
tidy(ivr, conf.int = TRUE, instruments = TRUE)
augment(ivr)
augment(ivr, data = CigarettesSW)
augment(ivr, newdata = CigarettesSW)
glance(ivr)
```

 $\verb"augment.kmeans"$

Augment data with information from a(n) kmeans object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

augment.kmeans 43

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'kmeans'
augment(x, data, ...)
```

Arguments

Х

A kmeans object created by stats::kmeans().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:
.cluster Cluster assignment.

44 augment.lm

See Also

```
augment(), stats::kmeans()
Other kmeans tidiers: glance.kmeans(), tidy.kmeans()
```

Examples

```
library(cluster)
library(modeldata)
library(dplyr)

data(hpc_data)

x <- hpc_data[, 2:5]

fit <- pam(x, k = 4)

tidy(fit)
glance(fit)
augment(fit, x)</pre>
```

augment.lm

Augment data with information from a(n) lm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters

augment.lm 45

the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'lm'
augment(
  х,
  data = model.frame(x),
  newdata = NULL,
  se_fit = FALSE,
  interval = c("none", "confidence", "prediction"),
  conf.level = 0.95,
)
```

Arguments

An 1m object created by stats::lm(). Х

data A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. Do not pass new data to the data argument. Augment will report information such as influence and

> cooks distance for data passed to the data argument. These measures are only defined for the original training data.

A base::data.frame() or tibble::tibble() containing all the original prenewdata

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

se_fit Logical indicating whether or not a .se.fit column should be added to the

augmented output. For some models, this calculation can be somewhat time-

consuming. Defaults to FALSE.

Character indicating the type of confidence interval columns to be added to the

augmented output. Passed on to predict() and defaults to "none".

The confidence level to use for the interval created if interval is "confidence" or "prediction". Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence/prediction interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

interval

conf.level

46 augment.lm

Details

When the modeling was performed with na.action = "na.omit" (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with na.action = "na.exclude", one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to augment() and na.action = "na.exclude", a warning is raised and the incomplete rows are dropped.

Some unusual lm objects, such as rlm from MASS, may omit .cooksd and .std.resid. gam from mgcv omits .sigma.

When newdata is supplied, only returns .fitted, .resid and .se.fit columns.

Value

A tibble::tibble() with columns:

. cooksd Cooks distance.

.fitted Fitted or predicted value..hat Diagonal of the hat matrix.

.lower Lower bound on interval for fitted values.

resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

. sigma Estimated residual standard deviation when corresponding observation is dropped

from model.

.std.resid Standardised residuals.

. upper Upper bound on interval for fitted values.

See Also

```
stats::na.action
augment(), stats::predict.lm()
Other lm tidiers: augment.glm(), glance.glm(), glance.lm(), glance.summary.lm(), glance.svyglm(),
tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

```
library(ggplot2)
library(dplyr)

mod <- lm(mpg ~ wt + qsec, data = mtcars)

tidy(mod)
glance(mod)

# coefficient plot
d <- tidy(mod, conf.int = TRUE)</pre>
```

augment.Im 47

```
ggplot(d, aes(estimate, term, xmin = conf.low, xmax = conf.high, height = 0)) +
 geom_point() +
 geom_vline(xintercept = 0, lty = 4) +
 geom_errorbarh()
# aside: There are tidy() and glance() methods for lm.summary objects too.
# this can be useful when you want to conserve memory by converting large lm
# objects into their leaner summary.lm equivalents.
s <- summary(mod)</pre>
tidy(s, conf.int = TRUE)
glance(s)
augment(mod)
augment(mod, mtcars, interval = "confidence")
# predict on new data
newdata <- mtcars %>%
 head(6) %>%
 mutate(wt = wt + 1)
augment(mod, newdata = newdata)
# ggplot2 example where we also construct 95% prediction interval
# simpler bivariate model since we're plotting in 2D
mod2 <- lm(mpg ~ wt, data = mtcars)</pre>
au <- augment(mod2, newdata = newdata, interval = "prediction")</pre>
ggplot(au, aes(wt, mpg)) +
 geom_point() +
 geom\_line(aes(y = .fitted)) +
 geom_ribbon(aes(ymin = .lower, ymax = .upper), col = NA, alpha = 0.3)
# predict on new data without outcome variable. Output does not include .resid
newdata <- newdata %>%
 select(-mpg)
augment(mod, newdata = newdata)
au <- augment(mod, data = mtcars)</pre>
ggplot(au, aes(.hat, .std.resid)) +
 geom_vline(size = 2, colour = "white", xintercept = 0) +
 geom_hline(size = 2, colour = "white", yintercept = 0) +
 geom_point() +
 geom_smooth(se = FALSE)
plot(mod, which = 6)
ggplot(au, aes(.hat, .cooksd)) +
 geom_vline(xintercept = 0, colour = NA) +
 geom_abline(slope = seq(0, 3, by = 0.5), colour = "white") +
```

48 augment.lmRob

```
geom_smooth(se = FALSE) +
geom_point()

# column-wise models
a <- matrix(rnorm(20), nrow = 10)
b <- a + rnorm(length(a))
result <- lm(b ~ a)

tidy(result)</pre>
```

augment.1mRob

Augment data with information from a(n) lmRob object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'lmRob'
augment(x, data = model.frame(x), newdata = NULL, ...)
```

augment.lmRob 49

Arguments

x A lmRob object returned from robust::lmRob().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

See Also

```
robust::lmRob()
Other robust tidiers: glance.glmRob(), glance.lmRob(), tidy.glmRob(), tidy.lmRob()
```

```
# load modeling library
library(robust)

# fit model
m <- lmRob(mpg ~ wt, data = mtcars)

# summarize model fit with tidiers
tidy(m)
augment(m)
glance(m)</pre>
```

50 augment.lmrob

augment.lmrob

Augment data with information from a(n) lmrob object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it must be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the same number of rows as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'lmrob'
augment(x, data = model.frame(x), newdata = NULL, se_fit = FALSE, ...)
```

Arguments

A 1mrob object returned from robustbase::1mrob().

data A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that

augment(my_fit) returns the augmented original data. Do not pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been

passed to newdata. If newdata is specified, the data argument will be ignored.

augment.lmrob 51

se_fit

Logical indicating whether or not a .se.fit column should be added to the augmented output. For some models, this calculation can be somewhat time-consuming. Defaults to FALSE.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

Value

```
A tibble::tibble() with columns:
```

.fitted Fitted or predicted value.

.resid The difference between observed and fitted values.

See Also

```
robustbase::lmrob()
Other robustbase tidiers: augment.glmrob(), glance.lmrob(), tidy.glmrob(), tidy.lmrob()
```

```
if (requireNamespace("robustbase", quietly = TRUE)) {
  # load libraries for models and data
  library(robustbase)

  data(coleman)
  set.seed(0)

  m <- lmrob(Y ~ ., data = coleman)
  tidy(m)
  augment(m)
  glance(m)

  data(carrots)

Rfit <- glmrob(cbind(success, total - success) ~ logdose + block,
  family = binomial, data = carrots, method = "Mqle",
  control = glmrobMqle.control(tcc = 1.2)
)</pre>
```

52 augment.loess

```
tidy(Rfit)
augment(Rfit)
}
```

augment.loess

 $Tidy \ a(n) \ loess \ object$

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'loess'
augment(x, data = model.frame(x), newdata = NULL, se_fit = FALSE, ...)
```

Arguments

Χ

A loess objects returned by stats::loess().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

newdata

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

se_fit

Logical indicating whether or not a .se.fit column should be added to the augmented output. For some models, this calculation can be somewhat time-consuming. Defaults to FALSE.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

augment.loess 53

Details

When the modeling was performed with na.action = "na.omit" (as is the typical default), rows with NA in the initial data are omitted entirely from the augmented data frame. When the modeling was performed with na.action = "na.exclude", one should provide the original data as a second argument, at which point the augmented data will contain those rows (typically with NAs in place of the new columns). If the original data is not provided to augment() and na.action = "na.exclude", a warning is raised and the incomplete rows are dropped.

Note that loess objects by default will not predict on data outside of a bounding hypercube defined by the training data unless the original loess object was fit with control = loess.control(surface = \"direct\")). See stats::predict.loess() for details.

Value

A tibble::tibble() with columns:

.fitted or predicted value.

resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

See Also

```
stats::na.action
augment(), stats::loess(), stats::predict.loess()
```

```
lo <- loess(
  mpg ~ hp + wt,
  mtcars,
  control = loess.control(surface = "direct")
)
augment(lo)
# with all columns of original data
augment(lo, mtcars)
# with a new dataset
augment(lo, newdata = head(mtcars))</pre>
```

54 augment.Mclust

augment.Mclust

Augment data with information from a(n) Mclust object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'Mclust'
augment(x, data = NULL, ...)
```

Arguments

data

x An Mclust object return from mclust::Mclust().

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be

augment.Mclust 55

used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

. class Predicted class.

. uncertainty The uncertainty associated with the classification. Equal to one minus the model

class probability.

See Also

```
augment(), mclust::Mclust()
Other mclust tidiers: tidy.Mclust()
```

```
# load library for models and data
library(mclust)
# load data manipulation libraries
library(dplyr)
library(tibble)
library(purrr)
library(tidyr)
set.seed(27)
centers <- tibble(</pre>
  cluster = factor(1:3),
  # number points in each cluster
  num_points = c(100, 150, 50),
  # x1 coordinate of cluster center
  x1 = c(5, 0, -3),
  # x2 coordinate of cluster center
  x2 = c(-1, 1, -2)
)
points <- centers %>%
  mutate(
   x1 = map2(num_points, x1, rnorm),
   x2 = map2(num_points, x2, rnorm)
  ) %>%
  select(-num_points, -cluster) %>%
  unnest(c(x1, x2))
```

56 augment.mfx

```
# fit model
m <- Mclust(points)

# summarize model fit with tidiers
tidy(m)
augment(m, points)
glance(m)</pre>
```

augment.mfx

Augment data with information from a(n) mfx object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'mfx'
augment(
    x,
    data = model.frame(x$fit),
    newdata = NULL,
    type.predict = c("link", "response", "terms"),
```

augment.mfx 57

```
type.residuals = c("deviance", "pearson"),
  se_fit = FALSE,
)
## S3 method for class 'logitmfx'
augment(
 data = model.frame(x$fit),
 newdata = NULL,
  type.predict = c("link", "response", "terms"),
  type.residuals = c("deviance", "pearson"),
  se_fit = FALSE,
)
## S3 method for class 'negbinmfx'
augment(
 Х,
 data = model.frame(x$fit),
 newdata = NULL,
 type.predict = c("link", "response", "terms"),
  type.residuals = c("deviance", "pearson"),
 se_fit = FALSE,
)
## S3 method for class 'poissonmfx'
augment(
 Х,
 data = model.frame(x$fit),
 newdata = NULL,
  type.predict = c("link", "response", "terms"),
  type.residuals = c("deviance", "pearson"),
  se_fit = FALSE,
)
## S3 method for class 'probitmfx'
augment(
 Х,
 data = model.frame(x$fit),
 newdata = NULL,
  type.predict = c("link", "response", "terms"),
  type.residuals = c("deviance", "pearson"),
  se_fit = FALSE,
  . . .
)
```

58 augment.mfx

Arguments

x A logitmfx, negbinmfx, poissonmfx, or probitmfx object. (Note that betamfx objects receive their own set of tidiers.)

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

type.predict Passed to stats::predict.glm() type argument. Defaults to "link".

type.residuals Passed to stats::residuals.glm() and to stats::rstandard.glm() type

arguments. Defaults to "deviance".

se_fit Logical indicating whether or not a .se.fit column should be added to the

augmented output. For some models, this calculation can be somewhat time-

consuming. Defaults to FALSE.

.. Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it

will be ignored.

• augment() methods will warn when supplied a newdata argument if it will

be ignored.

Details

This generic augment method wraps augment.glm() for applicable objects from the mfx package.

Value

A tibble::tibble() with columns:

. cooksd Cooks distance.

.fitted Fitted or predicted value..hat Diagonal of the hat matrix.

.resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

. sigma Estimated residual standard deviation when corresponding observation is dropped

from model.

.std.resid Standardised residuals.

augment.mjoint 59

See Also

```
augment.glm(), mfx::logitmfx(), mfx::negbinmfx(), mfx::poissonmfx(), mfx::probitmfx()
Other mfx tidiers: augment.betamfx(), glance.betamfx(), glance.mfx(), tidy.betamfx(),
tidy.mfx()
```

Examples

```
# load libraries for models and data
library(mfx)
# get the marginal effects from a logit regression
mod_logmfx <- logitmfx(am ~ cyl + hp + wt, atmean = TRUE, data = mtcars)</pre>
tidy(mod_logmfx, conf.int = TRUE)
# compare with the naive model coefficients of the same logit call
 glm(am ~ cyl + hp + wt, family = binomial, data = mtcars),
 conf.int = TRUE
)
augment(mod_logmfx)
glance(mod_logmfx)
# another example, this time using probit regression
mod_probmfx <- probitmfx(am ~ cyl + hp + wt, atmean = TRUE, data = mtcars)</pre>
tidy(mod_probmfx, conf.int = TRUE)
augment(mod_probmfx)
glance(mod_probmfx)
```

augment.mjoint

Augment data with information from a(n) mjoint object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

60 augment.mjoint

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'mjoint'
augment(x, data = x$data, ...)
```

Arguments

x An mjoint object returned from joineRML::mjoint().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

See joineRML::fitted.mjoint() and joineRML::residuals.mjoint() for more information on the difference between population-level and individual-level fitted values and residuals.

If fitting a joint model with a single longitudinal process, make sure you are using a named list to define the formula for the fixed and random effects of the longitudinal submodel.

augment.mjoint 61

Value

A tibble::tibble() with one row for each original observation with addition columns:

.fitted_j_0 population-level fitted values for the j-th longitudinal process

.fitted_j_1 individuals-level fitted values for the j-th longitudinal process

.resid_j_0 population-level residuals for the j-th longitudinal process

.resid_j_1 individual-level residuals for the j-th longitudinal process

```
# broom only skips running these examples because the example models take a
# while to generate—they should run just fine, though!
## Not run:
# load libraries for models and data
library(joineRML)
# fit a joint model with bivariate longitudinal outcomes
data(heart.valve)
hvd <- heart.valve[!is.na(heart.valve$log.grad) &</pre>
 !is.na(heart.valve$log.lvmi) &
 heart.valve$num <= 50, ]
fit <- mjoint(</pre>
 formLongFixed = list(
    "grad" = log.grad \sim time + sex + hs,
   "lvmi" = log.lvmi ~ time + sex
 formLongRandom = list(
    "grad" = ~1 \mid num,
    "lvmi" = ~ time | num
 ),
 formSurv = Surv(fuyrs, status) ~ age,
 data = hvd,
 inits = list("gamma" = c(0.11, 1.51, 0.80)),
 timeVar = "time"
# extract the survival fixed effects
tidy(fit)
# extract the longitudinal fixed effects
tidy(fit, component = "longitudinal")
# extract the survival fixed effects with confidence intervals
tidy(fit, ci = TRUE)
# extract the survival fixed effects with confidence intervals based
```

62 augment.mlogit

```
# on bootstrapped standard errors
bSE <- bootsE(fit, nboot = 5, safe.boot = TRUE)
tidy(fit, boot_se = bSE, ci = TRUE)

# augment original data with fitted longitudinal values and residuals
hvd2 <- augment(fit)

# extract model statistics
glance(fit)

## End(Not run)</pre>
```

augment.mlogit

Augment data with information from a(n) mlogit object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'mlogit'
augment(x, data = x$model, ...)
```

augment.mlogit 63

Arguments

. . .

x an object returned from mlogit::mlogit().

data Not currently used

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

At the moment this only works on the estimation dataset. Need to set it up to predict on another dataset.

Value

```
A tibble::tibble() with columns:
```

. fitted Fitted or predicted value.

.probability Class probability of modal class.

resid The difference between observed and fitted values.

See Also

```
augment()
Other mlogit tidiers: glance.mlogit(), tidy.mlogit()
```

```
# load libraries for models and data
library(mlogit)

data("Fishing", package = "mlogit")
Fish <- dfidx(Fishing, varying = 2:9, shape = "wide", choice = "mode")

# fit model
m <- mlogit(mode ~ price + catch | income, data = Fish)

# summarize model fit with tidiers
tidy(m)
augment(m)
glance(m)</pre>
```

64 augment.nlrq

augment.nlrq

Tidy a(n) nlrq object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'nlrq'
augment(x, data = NULL, newdata = NULL, ...)
```

Arguments

Х

A nlrq object returned from quantreg::nlrq().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

newdata

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

See Also

```
augment(), quantreg::nlrq()
Other quantreg tidiers: augment.rq(), augment.rqs(), glance.nlrq(), glance.rq(), tidy.nlrq(),
tidy.rq(), tidy.rqs()
```

augment.nls 65

Examples

```
# fit model
n <- nls(mpg ~ k * e^wt, data = mtcars, start = list(k = 1, e = 2))
# summarize model fit with tidiers + visualization
tidy(n)
augment(n)
glance(n)
library(ggplot2)
ggplot(augment(n), aes(wt, mpg)) +
geom_point() +
geom_line(aes(y = .fitted))
newdata <- head(mtcars)
newdata$wt <- newdata$wt + 1
augment(n, newdata = newdata)</pre>
```

augment.nls

Augment data with information from a(n) nls object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

66 augment.nls

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'nls'
augment(x, data = NULL, newdata = NULL, ...)
```

Arguments

x An nls object returned from stats::nls().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

... Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

augment.nls does not currently support confidence intervals due to a lack of support in stats::predict.nls().

Value

```
A tibble::tibble() with columns:
```

.fitted Fitted or predicted value.

resid The difference between observed and fitted values.

See Also

```
tidy, stats::nls(), stats::predict.nls()
Other nls tidiers: glance.nls(), tidy.nls()
```

augment.pam 67

Examples

```
# fit model
n <- nls(mpg ~ k * e^wt, data = mtcars, start = list(k = 1, e = 2))
# summarize model fit with tidiers + visualization
tidy(n)
augment(n)
glance(n)
library(ggplot2)
ggplot(augment(n), aes(wt, mpg)) +
geom_point() +
geom_line(aes(y = .fitted))
newdata <- head(mtcars)
newdata$wt <- newdata$wt + 1
augment(n, newdata = newdata)</pre>
```

augment.pam

Augment data with information from a(n) pam object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

68 augment.pam

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'pam'
augment(x, data = NULL, ...)
```

Arguments

x An pam object returned from cluster::pam()

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

. cluster Cluster assignment.
. fitted Fitted or predicted value.

resid The difference between observed and fitted values.

See Also

```
augment(), cluster::pam()
Other pam tidiers: glance.pam(), tidy.pam()
```

```
# load libraries for models and data
library(dplyr)
library(ggplot2)
library(cluster)
library(modeldata)
```

augment.plm 69

```
data(hpc_data)

x <- hpc_data[, 2:5]
p <- pam(x, k = 4)

# summarize model fit with tidiers + visualization
tidy(p)
glance(p)
augment(p, x)

augment(p, x)

ggplot(aes(compounds, input_fields)) +
geom_point(aes(color = .cluster)) +
geom_text(aes(label = cluster), data = tidy(p), size = 10)</pre>
```

augment.plm

Augment data with information from a(n) plm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'plm'
augment(x, data = model.frame(x), ...)
```

70 augment.plm

Arguments

A plm objected returned by plm::plm().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it
 will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:

.fitted Fitted or predicted value.
```

The difference between observed and fitted values.

See Also

.resid

```
augment(), plm::plm()
Other plm tidiers: glance.plm(), tidy.plm()
```

```
# load libraries for models and data
library(plm)

# load data
data("Produc", package = "plm")

# fit model
zz <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp,
data = Produc, index = c("state", "year")
)

# summarize model fit with tidiers
summary(zz)

tidy(zz)
tidy(zz, conf.int = TRUE)</pre>
```

augment.poLCA 71

```
tidy(zz, conf.int = TRUE, conf.level = 0.9)
augment(zz)
glance(zz)
```

augment.poLCA

Augment data with information from a(n) poLCA object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'poLCA'
augment(x, data = NULL, ...)
```

Arguments ×

data

A poLCA object returned from poLCA::poLCA().

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and

72 augment.poLCA

cooks distance for data passed to the data argument. These measures are only defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

If the data argument is given, those columns are included in the output (only rows for which predictions could be made). Otherwise, the y element of the poLCA object, which contains the manifest variables used to fit the model, are used, along with any covariates, if present, in x.

Note that while the probability of all the classes (not just the predicted modal class) can be found in the posterior element, these are not included in the augmented output.

Value

```
A tibble::tibble() with columns:

.class Predicted class.

.probability Class probability of modal class.
```

See Also

```
augment(), poLCA::poLCA()
Other poLCA tidiers: glance.poLCA(), tidy.poLCA()
```

```
# load libraries for models and data
library(poLCA)
library(dplyr)

# generate data
data(values)

f <- cbind(A, B, C, D) ~ 1

# fit model
M1 <- poLCA(f, values, nclass = 2, verbose = FALSE)</pre>
M1
```

augment.polr 73

```
# summarize model fit with tidiers + visualization
augment(M1)
glance(M1)
library(ggplot2)
ggplot(tidy(M1), aes(factor(class), estimate, fill = factor(outcome))) +
 geom_bar(stat = "identity", width = 1) +
 facet_wrap(~variable)
# three-class model with a single covariate.
data(election)
f2a <- cbind(
 MORALG, CARESG, KNOWG, LEADG, DISHONG, INTELG,
 MORALB, CARESB, KNOWB, LEADB, DISHONB, INTELB
) ~ PARTY
nes2a <- poLCA(f2a, election, nclass = 3, nrep = 5, verbose = FALSE)</pre>
td <- tidy(nes2a)
td
ggplot(td, aes(outcome, estimate, color = factor(class), group = class)) +
 geom_line() +
 facet_wrap(~variable, nrow = 2) +
 theme(axis.text.x = element_text(angle = 90, hjust = 1))
au <- augment(nes2a)</pre>
au
count(au, .class)
# if the original data is provided, it leads to NAs in new columns
# for rows that weren't predicted
au2 <- augment(nes2a, data = election)</pre>
au2
dim(au2)
```

augment.polr

Augment data with information from a(n) polr object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in

74 augment.polr

the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na. action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'polr'
augment(
    x,
    data = model.frame(x),
    newdata = NULL,
    type.predict = c("class"),
    ...
)
```

Arguments

X	Apolr	object retui	ned from	MASS::	polr().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that $augment(my_fit)$ returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

 $type.predict \qquad Which type \ of \ prediction \ to \ compute, \ passed \ to \ MASS:::predict.polr(). \ Only$

supports "class" at the moment.

augment.prcomp 75

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

See Also

```
tidy(), MASS::polr()
Other ordinal tidiers: augment.clm(), glance.clm(), glance.clmm(), glance.polr(), glance.svyolr(),
tidy.clm(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

Examples

```
# load libraries for models and data
library(MASS)

# fit model
fit <- polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)

# summarize model fit with tidiers
tidy(fit, exponentiate = TRUE, conf.int = TRUE)

glance(fit)
augment(fit, type.predict = "class")

fit2 <- polr(factor(gear) ~ am + mpg + qsec, data = mtcars)

tidy(fit, p.values = TRUE)</pre>
```

augment.prcomp

Augment data with information from a(n) prcomp object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object.

76 augment.prcomp

Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'prcomp'
augment(x, data = NULL, newdata, ...)
```

Arguments

A prcomp object returned by stats::prcomp().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

newdata

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

augment.rlm 77

Value

A tibble::tibble containing the original data along with additional columns containing each observation's projection into PCA space.

See Also

```
stats::prcomp(), svd_tidiers
Other svd tidiers: tidy.prcomp(), tidy_irlba(), tidy_svd()
```

augment.rlm

Augment data with information from a(n) rlm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'rlm'
augment(x, data = model.frame(x), newdata = NULL, se_fit = FALSE, ...)
```

78 augment.rlm

Arguments

x An rlm object returned by MASS::rlm().

data A base::data.frame or tibble::tibble() containing the original data that was

used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

newdata A base::data.frame() or tibble::tibble() containing all the original pre-

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

se_fit Logical indicating whether or not a .se.fit column should be added to the

augmented output. For some models, this calculation can be somewhat time-

consuming. Defaults to FALSE.

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

 augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

.fitted Fitted or predicted value..hat Diagonal of the hat matrix.

. resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

. sigma Estimated residual standard deviation when corresponding observation is dropped

from model.

See Also

```
MASS::rlm()
```

Other rlm tidiers: glance.rlm(), tidy.rlm()

```
# load libraries for models and data
library(MASS)
```

augment.rma 79

```
# fit model
r <- rlm(stack.loss ~ ., stackloss)
# summarize model fit with tidiers
tidy(r)
augment(r)
glance(r)</pre>
```

augment.rma

Augment data with information from a(n) rma object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'rma'
augment(x, interval = c("prediction", "confidence"), ...)
```

80 augment.rma

Arguments

x An rma object such as those created by metafor::rma(), metafor::rma.uni(), metafor::rma.glmm(), metafor::rma.mh(), metafor::rma.mv(), or metafor::rma.peto().

interval For rma.mv models, should prediction intervals ("prediction", default) or con-

fidence intervals ("confidence") intervals be returned? For rma.uni models, prediction intervals are always returned. For rma.mh and rma.peto models,

confidence intervals are always returned.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it
 will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

.fitted Fitted or predicted value.

. lower Lower bound on interval for fitted values.

. moderator In meta-analysis, the moderators used to calculate the predicted values.

.moderator.level

In meta-analysis, the level of the moderators used to calculate the predicted

values

resid The difference between observed and fitted values.

.se.fit Standard errors of fitted values.

. upper Upper bound on interval for fitted values.. observed The observed values for the individual studies

```
# load modeling library
library(metafor)

# generate data and fit
df <-
    escalc(
    measure = "RR",
    ai = tpos,
    bi = tneg,
    ci = cpos,
    di = cneg,
    data = dat.bcg</pre>
```

augment.rq 81

```
meta_analysis <- rma(yi, vi, data = df, method = "EB")
# summarize model fit with tidiers
augment(meta_analysis)</pre>
```

augment.rq

Augment data with information from a(n) rq object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'rq'
augment(x, data = model.frame(x), newdata = NULL, ...)
```

Arguments

Х

An rq object returned from quantreg::rq().

82 augment.rq

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

newdata

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

. . .

Arguments passed on to quantreg::predict.rq

object object of class rq or rqs or rq.process produced by rq

interval type of interval desired: default is 'none', when set to 'confidence' the function returns a matrix predictions with point predictions for each of the 'newdata' points as well as lower and upper confidence limits.

level converage probability for the 'confidence' intervals.

type For predict.rq, the method for 'confidence' intervals, if desired. If 'percentile' then one of the bootstrap methods is used to generate percentile intervals for each prediction, if 'direct' then a version of the Portnoy and Zhou (1998) method is used, and otherwise an estimated covariance matrix for the parameter estimates is used. Further arguments to determine the choice of bootstrap method or covariance matrix estimate can be passed via the ... argument. For predict.rqs and predict.rq.process when stepfun = TRUE, type is "Qhat", "Fhat" or "fhat" depending on whether the user would like to have estimates of the conditional quantile, distribution or density functions respectively. As noted below the two former estimates can be monotonized with the function rearrange. When the "fhat" option is invoked, a list of conditional density functions is returned based on Silverman's adaptive kernel method as implemented in akj and approxfun.

na.action function determining what should be done with missing values in 'newdata'. The default is to predict 'NA'.

Details

Depending on the arguments passed on to predict.rq via ..., a confidence interval is also calculated on the fitted values resulting in columns .lower and .upper. Does not provide confidence intervals when data is specified via the newdata argument.

Value

A tibble::tibble() with columns:

.fitted Fitted or predicted value.

. resid The difference between observed and fitted values.

. tau Quantile.

augment.rqs 83

See Also

```
augment, quantreg::rq(), quantreg::predict.rq()
Other quantreg tidiers: augment.nlrq(), augment.rqs(), glance.nlrq(), glance.rq(), tidy.nlrq(),
tidy.rq(), tidy.rqs()
```

Examples

```
# load modeling library and data
library(quantreg)
data(stackloss)
# median (11) regression fit for the stackloss data.
mod1 <- rq(stack.loss ~ stack.x, .5)</pre>
# weighted sample median
mod2 <- rq(rnorm(50) \sim 1, weights = runif(50))
# summarize model fit with tidiers
tidy(mod1)
glance(mod1)
augment(mod1)
tidy(mod2)
glance(mod2)
augment(mod2)
# varying tau to generate an rqs object
mod3 \leftarrow rq(stack.loss \sim stack.x, tau = c(.25, .5))
tidy(mod3)
augment(mod3)
# glance cannot handle rqs objects like `mod3`--use a purrr
# `map`-based workflow instead
```

augment.rqs

Augment data with information from a(n) rqs object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object.

84 augment.rqs

Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the same number of rows as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'rgs'
augment(x, data = model.frame(x), newdata, ...)
```

Arguments

Х An rqs object returned from quantreg::rq().

data A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that

augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

A base::data.frame() or tibble::tibble() containing all the original prenewdata dictors used to create x. Defaults to NULL, indicating that nothing has been

passed to newdata. If newdata is specified, the data argument will be ignored.

Arguments passed on to quantreg::predict.rq

object object of class rq or rqs or rq.process produced by rq

interval type of interval desired: default is 'none', when set to 'confidence' the function returns a matrix predictions with point predictions for each of the 'newdata' points as well as lower and upper confidence limits.

level converage probability for the 'confidence' intervals.

type For predict.rg, the method for 'confidence' intervals, if desired. If 'percentile' then one of the bootstrap methods is used to generate percentile intervals for each prediction, if 'direct' then a version of the Portnoy and Zhou (1998) method is used, and otherwise an estimated covariance matrix for the parameter estimates is used. Further arguments to determine the choice of bootstrap method or covariance matrix estimate can be passed

augment.rqs 85

via the ...argument. For predict.rqs and predict.rq.process when stepfun = TRUE, type is "Qhat", "Fhat" or "fhat" depending on whether the user would like to have estimates of the conditional quantile, distribution or density functions respectively. As noted below the two former estimates can be monotonized with the function rearrange. When the "fhat" option is invoked, a list of conditional density functions is returned based on Silverman's adaptive kernel method as implemented in akj and approxfun.

na.action function determining what should be done with missing values in 'newdata'. The default is to predict 'NA'.

Details

Depending on the arguments passed on to predict.rq via ..., a confidence interval is also calculated on the fitted values resulting in columns .lower and .upper. Does not provide confidence intervals when data is specified via the newdata argument.

See Also

```
augment, quantreg::rq(), quantreg::predict.rqs()
Other quantreg tidiers: augment.nlrq(), augment.rq(), glance.nlrq(), glance.rq(), tidy.nlrq(),
tidy.rq(), tidy.rqs()
```

```
# load modeling library and data
library(quantreg)
data(stackloss)
# median (11) regression fit for the stackloss data.
mod1 <- rq(stack.loss ~ stack.x, .5)</pre>
# weighted sample median
mod2 \leftarrow rq(rnorm(50) \sim 1, weights = runif(50))
# summarize model fit with tidiers
tidy(mod1)
glance(mod1)
augment(mod1)
tidy(mod2)
glance(mod2)
augment(mod2)
# varying tau to generate an rqs object
mod3 \leftarrow rq(stack.loss \sim stack.x, tau = c(.25, .5))
tidy(mod3)
augment(mod3)
```

86 augment.sarlm

```
# glance cannot handle rqs objects like `mod3`--use a purrr
# `map`-based workflow instead
```

augment.sarlm

Augment data with information from a(n) spatialreg object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'sarlm'
augment(x, data = x$X, ...)
```

Arguments

X	An object returned from spatialreg::lagsarlm() or spatialreg::errorsarlm().
data	Ignored, but included for internal consistency. See the details below.
•••	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

augment.sarlm 87

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The predict method for sarlm objects assumes that the response is known. See ?predict.sarlm for more discussion. As a result, since the original data can be recovered from the fit object, this method currently does not take in data or newdata arguments.

Value

```
A tibble::tibble() with columns:

.fitted Fitted or predicted value.

.resid The difference between observed and fitted values.
```

See Also

```
augment()
Other spatialreg tidiers: glance.sarlm(), tidy.sarlm()
```

```
# load libraries for models and data
library(spatialreg)
library(spdep)
# load data
data(oldcol, package = "spdep")
listw <- nb2listw(COL.nb, style = "W")</pre>
# fit model
crime_sar <-</pre>
  lagsarlm(CRIME ~ INC + HOVAL,
    data = COL.OLD,
    listw = listw,
    method = "eigen"
  )
# summarize model fit with tidiers
tidy(crime_sar)
tidy(crime_sar, conf.int = TRUE)
glance(crime_sar)
augment(crime_sar)
# fit another model
```

```
crime_sem <- errorsarlm(CRIME ~ INC + HOVAL, data = COL.OLD, listw)</pre>
# summarize model fit with tidiers
tidy(crime_sem)
tidy(crime_sem, conf.int = TRUE)
glance(crime_sem)
augment(crime_sem)
# fit another model
crime_sac <- sacsarlm(CRIME ~ INC + HOVAL, data = COL.OLD, listw)</pre>
# summarize model fit with tidiers
tidy(crime_sac)
tidy(crime_sac, conf.int = TRUE)
glance(crime_sac)
augment(crime_sac)
```

augment.smooth.spline $Tidy \ a(n) \ smooth.spline \ object$

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'smooth.spline'
augment(x, data = x data, ...)
```

Arguments

data

A smooth.spline object returned from stats::smooth.spline(). Χ

> A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only

defined for the original training data.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

augment.speedlm 89

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:

.fitted Fitted or predicted value.

.resid The difference between observed and fitted values.
```

See Also

```
augment(), stats::smooth.spline(), stats::predict.smooth.spline()
Other smoothing spline tidiers: glance.smooth.spline()
```

Examples

```
# fit model
spl <- smooth.spline(mtcars$wt, mtcars$mpg, df = 4)
# summarize model fit with tidiers
augment(spl, mtcars)
# calls original columns x and y
augment(spl)
library(ggplot2)
ggplot(augment(spl, mtcars), aes(wt, mpg)) +
    geom_point() +
    geom_line(aes(y = .fitted))</pre>
```

augment.speedlm

Augment data with information from a(n) speedlm object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome

90 augment.speedlm

variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'speedlm'
augment(x, data = model.frame(x), newdata = NULL, ...)
```

Arguments

.

A speedlm object returned from speedglm::speedlm().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

newdata

A base::data.frame() or tibble::tibble() containing all the original predictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

. fitted or predicted value.

. resid The difference between observed and fitted values.

augment.stl 91

See Also

```
speedglm::speedlm()
Other speedlm tidiers: glance.speedglm(), glance.speedlm(), tidy.speedglm(), tidy.speedlm()
```

Examples

```
# load modeling library
library(speedglm)

# fit model
mod <- speedlm(mpg ~ wt + qsec, data = mtcars, fitted = TRUE)

# summarize model fit with tidiers
tidy(mod)
glance(mod)
augment(mod)</pre>
```

augment.stl

Augment data with information from a(n) stl object

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

92 augment.survreg

Usage

```
## S3 method for class 'stl'
augment(x, data = NULL, weights = TRUE, ...)
```

Arguments

x An stl object returned from stats::stl().

data Ignored, included for consistency with the augment generic signature only.

weights Logical indicating whether or not to include the robust weights in the output.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble with one row for each observation in the original times series:

. seasonal The seasonal component of the decomposition.. trend The trend component of the decomposition.

. remainder The remainder, or "random" component of the decomposition.

.weight The final robust weights, if requested.

. seasadj The seasonally adjusted (or "deseasonalised") series.

See Also

```
augment(), stats::stl()
```

Other decompose tidiers: augment.decomposed.ts()

augment.survreg 93

Description

Augment accepts a model object and a dataset and adds information about each observation in the dataset. Most commonly, this includes predicted values in the .fitted column, residuals in the .resid column, and standard errors for the fitted values in a .se.fit column. New columns always begin with a . prefix to avoid overwriting columns in the original dataset.

Users may pass data to augment via either the data argument or the newdata argument. If the user passes data to the data argument, it **must** be exactly the data that was used to fit the model object. Pass datasets to newdata to augment data that was not used during model fitting. This still requires that at least all predictor variable columns used to fit the model are present. If the original outcome variable used to fit the model is not included in newdata, then no .resid column will be included in the output.

Augment will often behave differently depending on whether data or newdata is given. This is because there is often information associated with training observations (such as influences or related) measures that is not meaningfully defined for new observations.

For convenience, many augment methods provide default data arguments, so that augment(fit) will return the augmented training data. In these cases, augment tries to reconstruct the original data based on the model object with varying degrees of success.

The augmented dataset is always returned as a tibble::tibble with the **same number of rows** as the passed dataset. This means that the passed data must be coercible to a tibble. If a predictor enters the model as part of a matrix of covariates, such as when the model formula uses splines::ns(), stats::poly(), or survival::Surv(), it is represented as a matrix column.

We are in the process of defining behaviors for models fit with various na.action arguments, but make no guarantees about behavior when data is missing at this time.

Usage

```
## S3 method for class 'survreg'
augment(
    x,
    data = model.frame(x),
    newdata = NULL,
    type.predict = "response",
    type.residuals = "response",
    ...
)
```

Arguments

x An survreg object returned from survival::survreg().

data

A base::data.frame or tibble::tibble() containing the original data that was used to produce the object x. Defaults to stats::model.frame(x) so that augment(my_fit) returns the augmented original data. **Do not** pass new data to the data argument. Augment will report information such as influence and cooks distance for data passed to the data argument. These measures are only defined for the original training data.

94 augment.survreg

A base::data.frame() or tibble::tibble() containing all the original prenewdata

dictors used to create x. Defaults to NULL, indicating that nothing has been passed to newdata. If newdata is specified, the data argument will be ignored.

Character indicating type of prediction to use. Passed to the type argument of type.predict

the stats::predict() generic. Allowed arguments vary with model class, so

be sure to read the predict.my_class documentation.

type.residuals Character indicating type of residuals to use. Passed to the type argument of stats::residuals() generic. Allowed arguments vary with model class, so

be sure to read the residuals.my_class documentation.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf. level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

Fitted or predicted value. .fitted

The difference between observed and fitted values. .resid

.se.fit Standard errors of fitted values.

See Also

```
augment(), survival::survreg()
Other survreg tidiers: glance.survreg(), tidy.survreg()
Other survival tidiers: augment.coxph(), glance.aareg(), glance.cch(), glance.coxph(),
glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)
# fit model
sr <- survreg(</pre>
 Surv(futime, fustat) ~ ecog.ps + rx,
 ovarian,
 dist = "exponential"
)
```

augment_columns 95

```
# summarize model fit with tidiers + visualization
tidy(sr)
augment(sr, ovarian)
glance(sr)

# coefficient plot
td <- tidy(sr, conf.int = TRUE)

library(ggplot2)

ggplot(td, aes(estimate, term)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
  geom_vline(xintercept = 0)</pre>
```

augment_columns

Add fitted values, residuals, and other common outputs to an augment call

Description

augment_columns is intended for use in the internals of augment methods only and is exported for developers extending the broom package. Please instead use augment() to appropriately make use of the functionality in augment_columns().

Usage

```
augment_columns(
    x,
    data,
    newdata = NULL,
    type,
    type.predict = type,
    type.residuals = type,
    se.fit = TRUE,
    ...
)
```

Arguments

x a model

data original data onto which columns should be added

newdata new data to predict on, optional

type Type of prediction and residuals to compute

type.predict Type of prediction to compute; by default same as type

96 bootstrap

type.residuals Type of residuals to compute; by default same as type

se.fit Value to pass to predict's se.fit, or NULL for no value. Ignored for model

types that do not accept an se.fit argument

... extra arguments (not used)

Details

Note that, in the case that a residuals() or influence() generic is not implemented for the supplied model x, the function will fail quietly.

bootstrap

Set up bootstrap replicates of a dplyr operation

Description

The bootstrap() function is deprecated and will be removed from an upcoming release of broom. For tidy resampling, please use the rsample package instead. Functionality is no longer supported for this method.

Usage

```
bootstrap(df, m, by_group = FALSE)
```

Arguments

df a data frame

m number of bootstrap replicates to perform

by_group If TRUE, then bootstrap within each group if df is a grouped tibble.

Details

This code originates from Hadley Wickham (with a few small corrections) here: https://github.com/tidyverse/dplyr/issues/269

See Also

```
Other deprecated: confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()
```

confint_tidy 97

C:	
confint_tidy	(Deprecated) Calculate confidence interval as a tidy data frame
_ •	\ 1

Description

This function is now deprecated and will be removed from a future release of broom.

Usage

```
confint_tidy(x, conf.level = 0.95, func = stats::confint, ...)
```

Arguments

x a model object for which confint() can be calculated

conf.level confidence level

func A function to compute a confidence interval for x. Calling func(x, level =

conf.level, ...) must return an object coercible to a tibble. This dataframe like object should have to columns corresponding the lower and upper bounds

on the confidence interval.

... extra arguments passed on to confint

Details

Return a confidence interval as a tidy data frame. This directly wraps the confint() function, but ensures it follows broom conventions: column names of conf.low and conf.high, and no row names.

```
confint_tidy
```

Value

A tibble with two columns: conf.low and conf.high.

See Also

```
Other deprecated: bootstrap(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()
```

98 data.frame_tidiers

data.frame_tidiers Tidiers for data.frame objects

Description

Data frame tidiers are deprecated and will be removed from an upcoming release of broom.

Usage

```
## S3 method for class 'data.frame'
tidy(x, ..., na.rm = TRUE, trim = 0.1)
## S3 method for class 'data.frame'
augment(x, data, ...)
## S3 method for class 'data.frame'
glance(x, ...)
```

Arguments

x	A data.frame
	Additional arguments for other methods.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Passed to the trim argument of mean
data	data, not used

Details

These perform tidy summaries of data.frame objects. tidy produces summary statistics about each column, while glance simply reports the number of rows and columns. Note that augment.data.frame will throw an error.

Value

n

tidy.data.frame produces a data frame with one row per original column, containing summary

statistics of each: name of original column column Number of valid (non-NA) values

mean mean

standard deviation sd

median median

trimmed trimmed mean, with trim defaulting to .1 data.frame_tidiers 99

mad median absolute deviation (from the median)
min minimum value
max maximum value
range range
skew skew
kurtosis kurtosis
se standard error

glance returns a one-row data.frame with

nrow number of rows ncol number of columns

complete.obs number of rows that have no missing values

na.fraction fraction of values across all rows and columns that are missing

Author(s)

David Robinson, Benjamin Nutter

Source

Skew and Kurtosis functions are adapted from implementations in the moments package: Lukasz Komsta and Frederick Novomestky (2015). moments: Moments, cumulants, skewness, kurtosis and related tests. R package version 0.14. https://CRAN.R-project.org/package=moments

See Also

```
Other deprecated: bootstrap(), confint_tidy(), finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()

Other deprecated: bootstrap(), confint_tidy(), finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()

Other deprecated: bootstrap(), confint_tidy(), finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()
```

```
durbinWatsonTest_tidiers
```

Tidy/glance a(n) durbinWatsonTest object

Description

For models that have only a single component, the tidy() and glance() methods are identical. Please see the documentation for both of those methods.

Usage

```
## $3 method for class 'durbinWatsonTest'
tidy(x, ...)
## $3 method for class 'durbinWatsonTest'
glance(x, ...)
```

Arguments

. . .

x An object of class durbinWatsonTest created by a call to car::durbinWatsonTest().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

alternative Alternative hypothesis (character).

autocorrelation

Autocorrelation.

p.value The two-sided p-value associated with the observed statistic.

statistic Test statistic for Durbin-Watson test.
method Always 'Durbin-Watson Test'.

See Also

```
tidy(), glance(), car::durbinWatsonTest()
Other car tidiers: leveneTest_tidiers
```

finish_glance 101

Examples

```
# load modeling library
library(car)

# fit model
dw <- durbinWatsonTest(lm(mpg ~ wt, data = mtcars))

# summarize model fit with tidiers
tidy(dw)

# same output for all durbinWatsonTests
glance(dw)</pre>
```

finish_glance

(Deprecated) Add logLik, AIC, BIC, and other common measurements to a glance of a prediction

Description

This function is now deprecated in favor of using custom logic and the appropriate nobs() method.

Usage

```
finish_glance(ret, x)
```

Arguments

ret a one-row data frame (a partially complete glance)

x the prediction model

Value

a one-row data frame with additional columns added, such as

logLik log likelihoods

AIC Akaike Information Criterion
BIC Bayesian Information Criterion

deviance deviance

df.residual residual degrees of freedom

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()
```

102 glance.aareg

fix_data_frame

Ensure an object is a data frame, with rownames moved into a column

Description

This function is deprecated as of broom 0.7.0 and will be removed from a future release. Please see tibble::as_tibble.

Usage

```
fix_data_frame(x, newnames = NULL, newcol = "term")
```

Arguments

x a data.frame or matrix

newnames new column names, not including the rownames

newcol the name of the new rownames column

Value

a data.frame, with rownames moved into a column and new column names assigned

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()
```

glance.aareg

Glance at a(n) aareg object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

glance.aareg 103

Usage

```
## S3 method for class 'aareg'
glance(x, ...)
```

Arguments

x An aareg object returned from survival::aareg().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

df Degrees of freedom used by the model.

nobs Number of observations used.

p.value P-value corresponding to the test statistic.

statistic Test statistic.

See Also

```
glance(), survival::aareg()
Other aareg tidiers: tidy.aareg()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.cch(), glance.coxph(),
glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# fit model
afit <- aareg(
   Surv(time, status) ~ age + sex + ph.ecog,
   data = lung,
   dfbeta = TRUE
)</pre>
```

104 glance.anova

```
# summarize model fit with tidiers
tidy(afit)
```

glance.anova

Glance at a(n) anova object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'anova'
glance(x, ...)
```

Arguments

An anova object, such as those created by stats::anova(), car::Anova(), car::leveneTest(), or car::linearHypothesis().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

deviance Deviance of the model.

df.residual Residual degrees of freedom.

glance.aov 105

Note

Note that the output of glance.anova() will vary depending on the initializing anova call. In some cases, it will just return an empty data frame. In other cases, glance.anova() may return columns that are also common to tidy.anova(). This is partly to preserve backwards compatibility with early versions of broom, but also because the underlying anova model yields components that could reasonably be interpreted as goodness-of-fit summaries too.

See Also

```
glance()
Other anova tidiers: glance.aov(), tidy.TukeyHSD(), tidy.anova(), tidy.aov(), tidy.aovlist(),
tidy.manova()
```

Examples

```
# fit models
a <- lm(mpg ~ wt + qsec + disp, mtcars)
b <- lm(mpg ~ wt + qsec, mtcars)

mod <- anova(a, b)

# summarize model fit with tidiers
tidy(mod)
glance(mod)

# car::linearHypothesis() example
library(car)
mod_lht <- linearHypothesis(a, "wt - disp")
tidy(mod_lht)
glance(mod_lht)</pre>
```

glance.aov

Glance at a(n) lm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

106 glance.aov

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'aov'
glance(x, ...)
```

Arguments

x An aov object, such as those created by stats::aov().

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

Note

Note that tidy.aov() now contains the numerator and denominator degrees of freedom, which were included in the output of glance.aov() in some previous versions of the package.

See Also

```
glance()
Other anova tidiers: glance.anova(), tidy.TukeyHSD(), tidy.anova(), tidy.aov(), tidy.aov(), tidy.aov(), tidy.aov(), tidy.aov()
```

```
a <- aov(mpg ~ wt + qsec + disp, mtcars)
tidy(a)</pre>
```

glance.Arima 107

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'Arima'
glance(x, ...)
```

Arguments

x An object of class Arima created by stats::arima().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

Number of observations used.

sigma Estimated standard error of the residuals.

108 glance.betamfx

See Also

```
stats::arima()
Other Arima tidiers: tidy.Arima()
```

Examples

```
# fit model
fit <- arima(lh, order = c(1, 0, 0))
# summarize model fit with tidiers
tidy(fit)
glance(fit)</pre>
```

glance.betamfx

Glance at a(n) betamfx object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'betamfx'
glance(x, ...)
```

Arguments

x A betamfx object.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

glance.betamfx 109

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This glance method wraps glance.betareg() for mfx::betamfx() objects.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df.null Degrees of freedom used by the null model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

pseudo.r.squared

Like the R squared statistic, but for situations when the R squared statistic isn't

defined.

See Also

```
glance.betareg(), mfx::betamfx()
Other mfx tidiers: augment.betamfx(), augment.mfx(), glance.mfx(), tidy.betamfx(), tidy.mfx()
```

```
library(mfx)

# Simulate some data
set.seed(12345)
n <- 1000
x <- rnorm(n)

# Beta outcome
y <- rbeta(n, shape1 = plogis(1 + 0.5 * x), shape2 = (abs(0.2 * x)))
# Use Smithson and Verkuilen correction
y <- (y * (n - 1) + 0.5) / n

d <- data.frame(y, x)
mod_betamfx <- betamfx(y ~ x | x, data = d)

tidy(mod_betamfx, conf.int = TRUE)

# Compare with the naive model coefficients of the equivalent betareg call (not run)
# tidy(betamfx(y ~ x | x, data = d), conf.int = TRUE)

augment(mod_betamfx)</pre>
```

110 glance.betareg

```
glance(mod_betamfx)
```

glance.betareg

Glance at a(n) betareg object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'betareg'
glance(x, ...)
```

Arguments

. . .

x A betareg object produced by a call to betareg::betareg().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df.null Degrees of freedom used by the null model.

glance.bigIm 111

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

pseudo.r.squared

Like the R squared statistic, but for situations when the R squared statistic isn't defined.

See Also

```
glance(), betareg::betareg()
```

Examples

```
# load libraries for models and data
library(betareg)

# load dats
data("GasolineYield", package = "betareg")

# fit model
mod <- betareg(yield ~ batch + temp, data = GasolineYield)

mod

# summarize model fit with tidiers
tidy(mod)
tidy(mod, conf.int = TRUE)
tidy(mod, conf.int = TRUE, conf.level = .99)
augment(mod)
glance(mod)</pre>
```

glance.biglm

Glance at a(n) biglm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

112 glance.biglm

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'biglm'
glance(x, ...)
```

Arguments

. . .

x A biglm object created by a call to biglm::biglm() or biglm::bigglm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

deviance Deviance of the model.

df.residual Residual degrees of freedom.

nobs Number of observations used.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

See Also

```
glance(), biglm::biglm(), biglm::bigglm()
Other biglm tidiers: tidy.biglm()
```

```
# load modeling library
library(biglm)

# fit model -- linear regression
bfit <- biglm(mpg ~ wt + disp, mtcars)
# summarize model fit with tidiers</pre>
```

glance.binDesign 113

```
tidy(bfit)
tidy(bfit, conf.int = TRUE)
tidy(bfit, conf.int = TRUE, conf.level = .9)
glance(bfit)

# fit model -- logistic regression
bgfit <- bigglm(am ~ mpg, mtcars, family = binomial())

# summarize model fit with tidiers
tidy(bgfit)
tidy(bgfit, exponentiate = TRUE)
tidy(bgfit, conf.int = TRUE)
tidy(bgfit, conf.int = TRUE, conf.level = .9)
tidy(bgfit, conf.int = TRUE, conf.level = .9, exponentiate = TRUE)
glance(bgfit)</pre>
```

glance.binDesign

Glance at a(n) binDesign object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'binDesign'
glance(x, ...)
```

Arguments

X

A binGroup::binDesign object.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

114 glance.cch

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

power Power achieved by the analysis.

n Sample size used to achieve this power.

power.reached Whether the desired power was reached.

maxit Number of iterations performed.

See Also

```
glance(), binGroup::binDesign()
Other bingroup tidiers: tidy.binDesign(), tidy.binWidth()
```

```
# load libraries for models and data
library(binGroup)

des <- binDesign(
   nmax = 300, delta = 0.06,
   p.hyp = 0.1, power = .8
)

glance(des)
tidy(des)
library(ggplot2)

ggplot(tidy(des), aes(n, power)) +
   geom_line()</pre>
```

glance.cch 115

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'cch'
glance(x, ...)
```

Arguments

x An cch object returned from survival::cch().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

iter Iterations of algorithm/fitting procedure completed.

p. value P-value corresponding to the test statistic.

rscore Robust log-rank statistic

score Score.

n number of predictions

nevent number of events

116 glance.clm

See Also

```
glance(), survival::cch()
Other cch tidiers: glance.survfit(), tidy.cch()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.coxph(),
glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)
# examples come from cch documentation
subcoh <- nwtco$in.subcohort</pre>
selccoh <- with(nwtco, rel == 1 | subcoh == 1)</pre>
ccoh.data <- nwtco[selccoh, ]</pre>
ccoh.data$subcohort <- subcoh[selccoh]</pre>
# central-lab histology
ccoh.data$histol <- factor(ccoh.data$histol, labels = c("FH", "UH"))</pre>
# tumour stage
ccoh.data$stage <- factor(ccoh.data$stage, labels = c("I", "II", "III", "IV"))</pre>
ccoh.data$age <- ccoh.data$age / 12 # age in years</pre>
# fit model
fit.ccP <- cch(Surv(edrel, rel) ~ stage + histol + age,</pre>
 data = ccoh.data,
 subcoh = ~subcohort, id = ~seqno, cohort.size = 4028
)
# summarize model fit with tidiers + visualization
tidy(fit.ccP)
# coefficient plot
library(ggplot2)
ggplot(tidy(fit.ccP), aes(x = estimate, y = term)) +
 geom_point() +
 geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
 geom_vline(xintercept = 0)
```

glance.clm 117

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'clm'
glance(x, ...)
```

Arguments

. . .

x A clm object returned from ordinal::clm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df.residual Residual degrees of freedom.
edf The effective degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

See Also

```
tidy, ordinal::clm()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clmm(), glance.polr(), glance.svyolr(),
tidy.clm(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

118 glance.clmm

Examples

```
# load libraries for models and data
library(ordinal)

# fit model
fit <- clm(rating ~ temp * contact, data = wine)

# summarize model fit with tidiers
tidy(fit)
tidy(fit, conf.int = TRUE, conf.level = 0.9)
tidy(fit, conf.int = TRUE, conf.type = "Wald", exponentiate = TRUE)
glance(fit)
augment(fit, type.predict = "prob")
augment(fit, type.predict = "class")

# ...and again with another model specification
fit2 <- clm(rating ~ temp, nominal = ~contact, data = wine)
tidy(fit2)
glance(fit2)</pre>
```

glance.clmm

Glance at a(n) clmm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'clmm'
glance(x, ...)
```

glance.clmm 119

Arguments

. . .

x A clmm object returned from ordinal::clmm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

edf The effective degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

See Also

```
tidy, ordinal::clmm()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.polr(), glance.svyolr(),
tidy.clm(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

```
# load libraries for models and data
library(ordinal)

# fit model
fit <- clmm(rating ~ temp + contact + (1 | judge), data = wine)

# summarize model fit with tidiers
tidy(fit)
tidy(fit, conf.int = TRUE, conf.level = 0.9)
tidy(fit, conf.int = TRUE, exponentiate = TRUE)

glance(fit)

# ...and again with another model specification
fit2 <- clmm(rating ~ temp + (1 | judge), nominal = ~contact, data = wine)
tidy(fit2)</pre>
```

120 glance.coeftest

```
glance(fit2)
```

glance.coeftest

Glance at a(n) coeftest object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'coeftest'
glance(x, ...)
```

Arguments

. . .

x A coeftest object returned from lmtest::coeftest().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking degrees of freedom into account.

AIC Akaike's Information Criterion for the model.

glance.coeftest 121

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

p.value P-value corresponding to the test statistic.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

statistic Test statistic.

Note

Because of the way that lmtest::coeftest() retains information about the underlying model object, the returned columns for glance.coeftest() will vary depending on the arguments. Specifically, four columns are returned regardless: "Loglik", "AIC", "BIC", and "nobs". Users can obtain additional columns (e.g. "r.squared", "df") by invoking the "save = TRUE" argument as part of lmtest::coeftest(). See examples.

As an aside, goodness-of-fit measures such as R-squared are unaffected by the presence of heteroskedasticity. For further discussion see, e.g. chapter 8.1 of Wooldridge (2016).

References

Wooldridge, Jeffrey M. (2016) *Introductory econometrics: A modern approach*. (6th edition). Nelson Education.

See Also

```
glance(), lmtest::coeftest()
```

```
# load libraries for models and data
library(lmtest)

m <- lm(dist ~ speed, data = cars)

coeftest(m)
tidy(coeftest(m))
tidy(coeftest(m, conf.int = TRUE))

# a very common workflow is to combine lmtest::coeftest with alternate
# variance-covariance matrices via the sandwich package. The lmtest
# tidiers support this workflow too, enabling you to adjust the standard
# errors of your tidied models on the fly.</pre>
```

122 glance.coxph

```
library(sandwich)

# "HC3" (default) robust SEs
tidy(coeftest(m, vcov = vcovHC))

# "HC2" robust SEs
tidy(coeftest(m, vcov = vcovHC, type = "HC2"))

# N-W HAC robust SEs
tidy(coeftest(m, vcov = NeweyWest))

# the columns of the returned tibble for glance.coeftest() will vary
# depending on whether the coeftest object retains the underlying model.
# Users can control this with the "save = TRUE" argument of coeftest().
glance(coeftest(m))
glance(coeftest(m, save = TRUE))
```

glance.coxph

Glance at a(n) coxph object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'coxph'
glance(x, ...)
```

Arguments

```
x A coxph object returned from survival::coxph().
```

For tidy(), additional arguments passed to summary(x, ...). Otherwise ignored.

glance.coxph 123

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

n The total number of observations.

nevent Number of events.

nobs Number of observations used.

See survival::coxph.object for additional column descriptions.

See Also

```
glance(), survival::coxph()
Other coxph tidiers: augment.coxph(), tidy.coxph()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)
# fit model
cfit <- coxph(Surv(time, status) ~ age + sex, lung)</pre>
# summarize model fit with tidiers
tidy(cfit)
tidy(cfit, exponentiate = TRUE)
lp <- augment(cfit, lung)</pre>
risks <- augment(cfit, lung, type.predict = "risk")
expected <- augment(cfit, lung, type.predict = "expected")</pre>
glance(cfit)
# also works on clogit models
resp <- levels(logan$occupation)</pre>
n <- nrow(logan)</pre>
indx <- rep(1:n, length(resp))</pre>
logan2 <- data.frame(</pre>
  logan[indx, ],
  id = indx,
  tocc = factor(rep(resp, each = n))
)
```

124 glance.crr

```
logan2$case <- (logan2$occupation == logan2$tocc)

cl <- clogit(case ~ tocc + tocc:education + strata(id), logan2)

tidy(cl)
glance(cl)

library(ggplot2)

ggplot(lp, aes(age, .fitted, color = sex)) +
    geom_point()

ggplot(risks, aes(age, .fitted, color = sex)) +
    geom_point()

ggplot(expected, aes(time, .fitted, color = sex)) +
    geom_point()</pre>
```

glance.crr

Glance at a(n) crr object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'crr'
glance(x, ...)
```

Arguments

A crr object returned from cmprsk::crr().

glance.crr 125

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

converged Logical indicating if the model fitting procedure was successful and converged.

df Degrees of freedom used by the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

statistic Test statistic.

See Also

```
glance(), cmprsk::crr()
Other cmprsk tidiers: tidy.crr()
```

```
library(cmprsk)

# time to loco-regional failure (lrf)
lrf_time <- rexp(100)
lrf_event <- sample(0:2, 100, replace = TRUE)
trt <- sample(0:1, 100, replace = TRUE)
strt <- sample(1:2, 100, replace = TRUE)

# fit model
x <- crr(lrf_time, lrf_event, cbind(trt, strt))

# summarize model fit with tidiers
tidy(x, conf.int = TRUE)
glance(x)</pre>
```

126 glance.cv.glmnet

glance.cv.glmnet

Glance at a(n) cv.glmnet object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'cv.glmnet'
glance(x, ...)
```

Arguments

x A cv.glmnet object returned from glmnet::cv.glmnet().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

lambda.1se The value of the penalization parameter lambda that results in the sparsest model while remaining within one standard error of the minimum loss.lambda.min The value of the penalization parameter lambda that achieved minimum loss as

estimated by cross validation.

nobs Number of observations used.

glance.cv.glmnet 127

See Also

```
glance(), glmnet::cv.glmnet()
Other glmnet tidiers: glance.glmnet(), tidy.cv.glmnet(), tidy.glmnet()
```

```
# load libraries for models and data
library(glmnet)
set.seed(27)
nobs <- 100
nvar <- 50
real <- 5
x <- matrix(rnorm(nobs * nvar), nobs, nvar)</pre>
beta <- c(rnorm(real, 0, 1), rep(0, nvar - real))
y \leftarrow c(t(beta) %*% t(x)) + rnorm(nvar, sd = 3)
cvfit1 <- cv.glmnet(x, y)</pre>
tidy(cvfit1)
glance(cvfit1)
library(ggplot2)
tidied_cv <- tidy(cvfit1)</pre>
glance_cv <- glance(cvfit1)</pre>
# plot of MSE as a function of lambda
g <- ggplot(tidied_cv, aes(lambda, estimate)) +</pre>
  geom_line() +
  scale_x_log10()
g
# plot of MSE as a function of lambda with confidence ribbon
g <- g + geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)</pre>
# plot of MSE as a function of lambda with confidence ribbon and choices
# of minimum lambda marked
g <- g +
  geom_vline(xintercept = glance_cv$lambda.min) +
  geom_vline(xintercept = glance_cv$lambda.1se, lty = 2)
g
# plot of number of zeros for each choice of lambda
ggplot(tidied_cv, aes(lambda, nzero)) +
  geom_line() +
  scale_x_log10()
```

128 glance.drc

```
# coefficient plot with min lambda shown
tidied <- tidy(cvfit1$glmnet.fit)

ggplot(tidied, aes(lambda, estimate, group = term)) +
    scale_x_log10() +
    geom_line() +
    geom_vline(xintercept = glance_cv$lambda.min) +
    geom_vline(xintercept = glance_cv$lambda.1se, lty = 2)</pre>
```

glance.drc

Glance at a(n) drc object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'drc'
glance(x, ...)
```

Arguments

A drc object produced by a call to drc::drm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

glance.ergm 129

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

AICc AIC corrected for small samples

See Also

```
glance(), drc::drm()
Other drc tidiers: augment.drc(), tidy.drc()
```

Examples

```
# load libraries for models and data
library(drc)

# fit model
mod <- drm(dead / total ~ conc, type,
    weights = total, data = selenium, fct = LL.2(), type = "binomial"
)

# summarize model fit with tidiers
tidy(mod)
tidy(mod, conf.int = TRUE)
glance(mod)
augment(mod, selenium)</pre>
```

glance.ergm

Glance at a(n) ergm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

130 glance.ergm

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'ergm'
glance(x, deviance = FALSE, mcmc = FALSE, ...)
```

Arguments

x An ergm object returned from a call to ergm::ergm().

deviance Logical indicating whether or not to report null and residual deviance for the

model, as well as degrees of freedom. Defaults to FALSE.

mcmc Logical indicating whether or not to report MCMC interval, burn-in and sample

size used to estimate the model. Defaults to FALSE.

... Additional arguments to pass to ergm::summary(). Cautionary note: Mis-

specified arguments may be silently ignored.

Value

glance.ergm returns a one-row tibble with the columns

independence Whether the model assumed dyadic independence

iterations The number of MCMLE iterations performed before convergence

logLik If applicable, the log-likelihood associated with the model

AIC The Akaike Information Criterion

BIC The Bayesian Information Criterion

If deviance = TRUE, and if the model supports it, the tibble will also contain the columns

null.deviance The null deviance of the model

df.null The degrees of freedom of the null deviance

residual.deviance

The residual deviance of the model

df.residual The degrees of freedom of the residual deviance

See Also

```
glance(), ergm::ergm(), ergm::summary.ergm()
Other ergm tidiers: tidy.ergm()
```

glance.factanal 131

glance.factanal Glance at a(n) factanal object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'factanal' glance(x, ...)
```

Arguments

x A factanal object created by stats::factanal().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

converged Logical indicating if the model fitting procedure was successful and converged.

df Degrees of freedom used by the model.

method Which method was used.

n The total number of observations.

n. factors The number of fitted factors.

132 glance.factanal

```
nobs Number of observations used.

p.value P-value corresponding to the test statistic.

statistic Test statistic.

total.variance Total cumulative proportion of variance accounted for by all factors.
```

See Also

```
glance(), stats::factanal()
Other factanal tidiers: augment.factanal(), tidy.factanal()
```

```
set.seed(123)
# generate data
library(dplyr)
library(purrr)
m1 <- tibble(
  v1 = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 4, 5, 6),
  v2 = c(1, 2, 1, 1, 1, 1, 2, 1, 2, 1, 3, 4, 3, 3, 3, 4, 6, 5),
  v3 = c(3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 4, 6),
  v4 = c(3, 3, 4, 3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 5, 6, 4),
  v5 = c(1, 1, 1, 1, 1, 3, 3, 3, 3, 1, 1, 1, 1, 1, 6, 4, 5),
  v6 = c(1, 1, 1, 2, 1, 3, 3, 3, 4, 3, 1, 1, 1, 2, 1, 6, 5, 4)
)
# new data
m2 <- map_dfr(m1, rev)</pre>
# factor analysis objects
fit1 <- factanal(m1, factors = 3, scores = "Bartlett")</pre>
fit2 <- factanal(m1, factors = 3, scores = "regression")</pre>
# tidying the object
tidy(fit1)
tidy(fit2)
# augmented dataframe
augment(fit1)
augment(fit2)
# augmented dataframe (with new data)
augment(fit1, data = m2)
augment(fit2, data = m2)
```

glance.felm 133

glance.felm Glance at a(n) felm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'felm'
glance(x, ...)
```

Arguments

. . .

x A felm object returned from lfe::felm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

Number of observations used.

p. value P-value corresponding to the test statistic.

134 glance.fitdistr

r.squared R squared statistic, or the percent of variation explained by the model. Also known as the coefficient of determination.

sigma Estimated standard error of the residuals.

statistic Test statistic.

```
# load libraries for models and data
library(lfe)
# use built-in `airquality` dataset
head(airquality)
# no FEs; same as lm()
est0 <- felm(Ozone ~ Temp + Wind + Solar.R, airquality)</pre>
# summarize model fit with tidiers
tidy(est0)
augment(est0)
# add month fixed effects
est1 <- felm(Ozone ~ Temp + Wind + Solar.R | Month, airquality)</pre>
# summarize model fit with tidiers
tidy(est1)
tidy(est1, fe = TRUE)
augment(est1)
glance(est1)
# the "se.type" argument can be used to switch out different standard errors
# types on the fly. In turn, this can be useful exploring the effect of
# different error structures on model inference.
tidy(est1, se.type = "iid")
tidy(est1, se.type = "robust")
# add clustered SEs (also by month)
est2 <- felm(Ozone ~ Temp + Wind + Solar.R | Month | 0 | Month, airquality)</pre>
# summarize model fit with tidiers
tidy(est2, conf.int = TRUE)
tidy(est2, conf.int = TRUE, se.type = "cluster")
tidy(est2, conf.int = TRUE, se.type = "robust")
tidy(est2, conf.int = TRUE, se.type = "iid")
```

glance.fitdistr 135

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'fitdistr'
glance(x, ...)
```

Arguments

x A fitdistr object returned by MASS::fitdistr().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

See Also

```
tidy(), MASS::fitdistr()
Other fitdistr tidiers: tidy.fitdistr()
```

136 glance.fixest

Examples

```
# load libraries for models and data
library(MASS)

# generate data
set.seed(2015)
x <- rnorm(100, 5, 2)

# fit models
fit <- fitdistr(x, dnorm, list(mean = 3, sd = 1))

# summarize model fit with tidiers
tidy(fit)
glance(fit)</pre>
```

glance.fixest

Glance at a(n) fixest object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'fixest'
glance(x, ...)
```

Arguments

x A fixest object returned from any of the fixest estimators

Additional arguments passed to summary and confint. Important arguments are se and cluster. Other arguments are dof, exact_dof, forceCovariance, and keepBounded. See summary.fixest.

glance.fixest 137

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

pseudo.r.squared

Like the R squared statistic, but for situations when the R squared statistic isn't

defined.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

within.r.squared

R squared within fixed-effect groups.

Note

All columns listed below will be returned, but some will be NA, depending on the type of model estimated. sigma, r.squared, adj.r.squared, and within.r.squared will be NA for any model other than feols. pseudo.r.squared will be NA for feols.

```
# load libraries for models and data
library(fixest)

gravity <-
    feols(
    log(Euros) ~ log(dist_km) | Origin + Destination + Product + Year, trade
)

tidy(gravity)
glance(gravity)
augment(gravity, trade)

# to get robust or clustered SEs, users can either:

# 1) specify the arguments directly in the `tidy()` call

tidy(gravity, conf.int = TRUE, cluster = c("Product", "Year"))

tidy(gravity, conf.int = TRUE, se = "threeway")

# 2) or, feed tidy() a summary.fixest object that has already accepted
# these arguments</pre>
```

138 glance.Gam

```
gravity_summ <- summary(gravity, cluster = c("Product", "Year"))
tidy(gravity_summ, conf.int = TRUE)
# approach (1) is preferred.</pre>
```

glance.Gam

Glance at a(n) Gam object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'Gam'
glance(x, ...)
```

Arguments

Х

A Gam object returned from a call to gam::gam().

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

Glance at gam objects created by calls to mgcv::gam() with glance.gam().

glance.gam 139

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

See Also

```
glance(), gam::gam()
Other gam tidiers: tidy.Gam()
```

glance.gam

Glance at a(n) gam object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'gam'
glance(x, ...)
```

140 glance.gam

Arguments

. . .

x A gam object returned from a call to mgcv::gam().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

npar Number of parameters in the model.

See Also

```
glance(), mgcv::gam()
Other mgcv tidiers: tidy.gam()
```

```
# load libraries for models and data
library(mgcv)

# fit model
g <- gam(mpg ~ s(hp) + am + qsec, data = mtcars)

# summarize model fit with tidiers
tidy(g)
tidy(g, parametric = TRUE)
glance(g)
augment(g)</pre>
```

glance.garch 141

glance.garch	Tidy a(n) garch object	

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'garch'
glance(x, test = c("box-ljung-test", "jarque-bera-test"), ...)
```

Arguments

x A garch object returned by tseries::garch().

test Character specification of which hypothesis test to use. The garch function

reports 2 hypothesis tests: Jarque-Bera to residuals and Box-Ljung to squared

residuals.

... Additional arguments. Not used. Needed to match generic signature only. Cau-

tionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

 tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

method Which method was used.

nobs Number of observations used.

p.value P-value corresponding to the test statistic.

statistic Test statistic.

parameter Parameter field in the htest, typically degrees of freedom.

142 glance.geeglm

See Also

```
glance(), tseries::garch(),[]
Other garch tidiers: tidy.garch()
```

glance.geeglm

Glance at a(n) geeglm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'geeglm'
glance(x, ...)
```

Arguments

x A geeglm object returned from a call to geepack::geeglm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

alpha Estimated correlation parameter for geepack::geeglm.

df.residual Residual degrees of freedom.

glance.glm 143

```
gamma Estimated scale parameter for geepack::geeglm.
max.cluster.size
Max number of elements in clusters.
n.clusters Number of clusters.
```

See Also

```
glance(), geepack::geeglm()
```

Examples

```
# load modeling library
library(geepack)

# load data
data(state)

ds <- data.frame(state.region, state.x77)

# fit model
geefit <- geeglm(Income ~ Frost + Murder,
    id = state.region,
    data = ds,
    corstr = "exchangeable"
)

# summarize model fit with tidiers
tidy(geefit)
tidy(geefit, conf.int = TRUE)</pre>
```

glance.glm

Glance at a(n) glm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

144 glance.glm

Usage

```
## S3 method for class 'glm'
glance(x, ...)
```

Arguments

x A glm object returned from stats::glm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df.null Degrees of freedom used by the null model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.
null.deviance Deviance of the null model.

See Also

```
stats::glm()
Other Im tidiers: augment.glm(), augment.lm(), glance.lm(), glance.summary.lm(), glance.svyglm(),
tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

```
g <- glm(am ~ mpg, mtcars, family = "binomial")
glance(g)</pre>
```

glance.glmnet 145

glance.glmnet

Glance at a(n) glmnet object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'glmnet'
glance(x, ...)
```

Arguments

x A glmnet object returned from glmnet::glmnet().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

nobs Number of observations used.

npasses Total passes over the data across all lambda values.

nulldev Null deviance.

146 glance.glmRob

See Also

```
glance(), glmnet::glmnet()
Other glmnet tidiers: glance.cv.glmnet(), tidy.cv.glmnet(), tidy.glmnet()
```

Examples

```
# load libraries for models and data
library(glmnet)
set.seed(2014)
x \leftarrow matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
fit1 <- glmnet(x, y)</pre>
# summarize model fit with tidiers + visualization
tidy(fit1)
glance(fit1)
library(dplyr)
library(ggplot2)
tidied <- tidy(fit1) %>% filter(term != "(Intercept)")
ggplot(tidied, aes(step, estimate, group = term)) +
  geom_line()
ggplot(tidied, aes(lambda, estimate, group = term)) +
  geom_line() +
  scale_x_log10()
ggplot(tidied, aes(lambda, dev.ratio)) +
  geom_line()
# works for other types of regressions as well, such as logistic
g2 <- sample(1:2, 100, replace = TRUE)</pre>
fit2 <- glmnet(x, g2, family = "binomial")</pre>
tidy(fit2)
```

glance.glmRob

Glance at a(n) glmRob object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

glance.glmRob 147

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'glmRob'
glance(x, ...)
```

Arguments

x A glmRob object returned from robust::glmRob().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

deviance Deviance of the model.

df.residual Residual degrees of freedom.

nobs Number of observations used.

null.deviance Deviance of the null model.

sigma Estimated standard error of the residuals.

See Also

```
robust::glmRob()
Other robust tidiers: augment.lmRob(), glance.lmRob(), tidy.glmRob(), tidy.lmRob()
```

148 glance.gmm

Examples

```
# load libraries for models and data
library(robust)

# fit model
gm <- glmRob(am ~ wt, data = mtcars, family = "binomial")

# summarize model fit with tidiers
tidy(gm)
glance(gm)</pre>
```

glance.gmm

Glance at a(n) gmm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'gmm'
glance(x, ...)
```

Arguments

Х

A gmm object returned from gmm::gmm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

glance.gmm 149

Value

A tibble::tibble() with exactly one row and columns:

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

Number of observations used.

p. value P-value corresponding to the test statistic.

statistic Test statistic.

See Also

```
glance(), gmm::gmm()
Other gmm tidiers: tidy.gmm()
```

```
# load libraries for models and data
library(gmm)
# examples come from the "gmm" package
# CAPM test with GMM
data(Finance)
r <- Finance[1:300, 1:10]
rm <- Finance[1:300, "rm"]
rf <- Finance[1:300, "rf"]
z <- as.matrix(r - rf)</pre>
t <- nrow(z)
zm <- rm - rf
h \leftarrow matrix(zm, t, 1)
res \leftarrow gmm(z \sim zm, x = h)
# tidy result
tidy(res)
tidy(res, conf.int = TRUE)
tidy(res, conf.int = TRUE, conf.level = .99)
# coefficient plot
library(ggplot2)
library(dplyr)
tidy(res, conf.int = TRUE) %>%
  mutate(variable = reorder(term, estimate)) %>%
  ggplot(aes(estimate, variable)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, color = "red", lty = 2)
# from a function instead of a matrix
```

150 glance.ivreg

```
g <- function(theta, x) {</pre>
  e \leftarrow x[, 2:11] - theta[1] - (x[, 1] - theta[1]) %*% matrix(theta[2:11], 1, 10)
  gmat <- cbind(e, e * c(x[, 1]))
  return(gmat)
}
x <- as.matrix(cbind(rm, r))</pre>
res_black <- gmm(g, x = x, t0 = rep(0, 11))
tidy(res_black)
tidy(res_black, conf.int = TRUE)
# APT test with Fama-French factors and GMM
f1 <- zm
f2 <- Finance[1:300, "hml"] - rf</pre>
f3 <- Finance[1:300, "smb"] - rf
h <- cbind(f1, f2, f3)
res2 <- gmm(z \sim f1 + f2 + f3, x = h)
td2 <- tidy(res2, conf.int = TRUE)
td2
# coefficient plot
td2 %>%
  mutate(variable = reorder(term, estimate)) %>%
  ggplot(aes(estimate, variable)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, color = "red", lty = 2)
```

glance.ivreg

Glance at a(n) ivreg object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

glance.ivreg 151

Usage

```
## S3 method for class 'ivreg'
glance(x, diagnostics = FALSE, ...)
```

Arguments

x An ivreg object created by a call to AER::ivreg().

diagnostics Logical indicating whether or not to return the Wu-Hausman and Sargan diag-

nostic information.

... Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

 augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This tidier currently only supports ivreg-classed objects outputted by the AER package. The ivreg package also outputs objects of class ivreg, and will be supported in a later release.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

Number of observations used.

r. squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

statistic Wald test statistic.

p.value P-value for the Wald test.

Note

Beginning 0.7.0, glance.ivreg returns statistics for the Wu-Hausman test for endogeneity and the Sargan test of overidentifying restrictions. Sargan test values are returned as NA if the number of instruments is not greater than the number of endogenous regressors.

152 glance.kmeans

See Also

```
glance(), AER::ivreg()
Other ivreg tidiers: augment.ivreg(), tidy.ivreg()
```

Examples

```
# load libraries for models and data
library(AER)
# load data
data("CigarettesSW", package = "AER")
# fit model
ivr <- ivreg(</pre>
 log(packs) ~ income | population,
 data = CigarettesSW,
 subset = year == "1995"
)
# summarize model fit with tidiers
tidy(ivr)
tidy(ivr, conf.int = TRUE)
tidy(ivr, conf.int = TRUE, instruments = TRUE)
augment(ivr)
augment(ivr, data = CigarettesSW)
augment(ivr, newdata = CigarettesSW)
glance(ivr)
```

glance.kmeans

Glance at a(n) kmeans object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

glance.kmeans 153

Usage

```
## S3 method for class 'kmeans' glance(x, ...)
```

Arguments

x A kmeans object created by stats::kmeans().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

betweenss The total between-cluster sum of squares.

iter Iterations of algorithm/fitting procedure completed.

tot.withinss The total within-cluster sum of squares.

totss The total sum of squares.

See Also

```
glance(), stats::kmeans()
Other kmeans tidiers: augment.kmeans(), tidy.kmeans()
```

```
library(cluster)
library(modeldata)
library(dplyr)

data(hpc_data)

x <- hpc_data[, 2:5]

fit <- pam(x, k = 4)

tidy(fit)
glance(fit)
augment(fit, x)</pre>
```

154 glance.lavaan

glance.lavaan

Glance at a(n) lavaan object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'lavaan'
glance(x, ...)
```

Arguments

x A lavaan object, such as those returned from lavaan::cfa(), and lavaan::sem().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A one-row tibble::tibble with columns:

chisq Model chi squared

npar Number of parameters in the model
rmsea Root mean square error of approximation

rmsea.conf.high

95 percent upper bound on RMSEA

glance.lavaan 155

srmr	Standardised root mean residual
agfi	Adjusted goodness of fit
cfi	Comparative fit index
tli	Tucker Lewis index
AIC	Akaike information criterion
BIC	Bayesian information criterion
ngroups	Number of groups in model
nobs	Number of observations included
norig	Number of observation in the original dataset
nexcluded	Number of excluded observations
converged	Logical - Did the model converge
estimator	Estimator used
missing_method	Method for eliminating missing data

For further recommendations on reporting SEM and CFA models see Schreiber, J. B. (2017). Update to core reporting practices in structural equation modeling. Research in Social and Administrative Pharmacy, 13(3), 634-643. https://doi.org/10.1016/j.sapharm.2016.06.006

See Also

```
glance(), lavaan::cfa(), lavaan::sem(), lavaan::fitmeasures()
Other lavaan tidiers: tidy.lavaan()
```

```
library(lavaan)
# fit model
cfa.fit <- cfa(
   "F =~ x1 + x2 + x3 + x4 + x5",
   data = HolzingerSwineford1939, group = "school"
)
# summarize model fit with tidiers
glance(cfa.fit)</pre>
```

156 glance.lm

glance.lm

Glance at a(n) lm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'lm'
glance(x, ...)
```

Arguments

. . .

x An lm object created by stats::lm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df.residual Residual degrees of freedom.

glance.lm 157

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

p. value P-value corresponding to the test statistic.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

statistic Test statistic.

df The degrees for freedom from the numerator of the overall F-statistic. This

is new in broom 0.7.0. Previously, this reported the rank of the design matrix, which is one more than the numerator degrees of freedom of the overall

F-statistic.

See Also

```
glance(), glance.summary.lm()
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.summary.lm(), glance.svyglm(),
tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

```
library(ggplot2)
library(dplyr)
mod <- lm(mpg ~ wt + qsec, data = mtcars)</pre>
tidy(mod)
glance(mod)
# coefficient plot
d <- tidy(mod, conf.int = TRUE)</pre>
ggplot(d, aes(estimate, term, xmin = conf.low, xmax = conf.high, height = 0)) +
  geom_point() +
  geom_vline(xintercept = 0, lty = 4) +
  geom_errorbarh()
# aside: There are tidy() and glance() methods for lm.summary objects too.
# this can be useful when you want to conserve memory by converting large lm
# objects into their leaner summary.lm equivalents.
s <- summary(mod)</pre>
tidy(s, conf.int = TRUE)
glance(s)
augment (mod)
augment(mod, mtcars, interval = "confidence")
# predict on new data
newdata <- mtcars %>%
```

158 glance.lmodel2

```
head(6) %>%
 mutate(wt = wt + 1)
augment(mod, newdata = newdata)
# ggplot2 example where we also construct 95% prediction interval
# simpler bivariate model since we're plotting in 2D
mod2 <- lm(mpg ~ wt, data = mtcars)</pre>
au <- augment(mod2, newdata = newdata, interval = "prediction")</pre>
ggplot(au, aes(wt, mpg)) +
 geom_point() +
 geom\_line(aes(y = .fitted)) +
 geom_ribbon(aes(ymin = .lower, ymax = .upper), col = NA, alpha = 0.3)
# predict on new data without outcome variable. Output does not include .resid
newdata <- newdata %>%
 select(-mpg)
augment(mod, newdata = newdata)
au <- augment(mod, data = mtcars)</pre>
ggplot(au, aes(.hat, .std.resid)) +
 geom_vline(size = 2, colour = "white", xintercept = 0) +
 geom_hline(size = 2, colour = "white", yintercept = 0) +
 geom_point() +
 geom_smooth(se = FALSE)
plot(mod, which = 6)
ggplot(au, aes(.hat, .cooksd)) +
 geom_vline(xintercept = 0, colour = NA) +
 geom_abline(slope = seq(0, 3, by = 0.5), colour = "white") +
 geom_smooth(se = FALSE) +
 geom_point()
# column-wise models
a <- matrix(rnorm(20), nrow = 10)
b <- a + rnorm(length(a))</pre>
result <- lm(b \sim a)
tidy(result)
```

glance.lmodel2 159

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'lmodel2'
glance(x, ...)
```

Arguments

. . .

x A lmodel2 object returned by lmodel2::lmodel2().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

nobs Number of observations used.

p. value P-value corresponding to the test statistic.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

theta Angle between OLS lines ' $lm(y \sim x)$ ' and ' $lm(x \sim y)$ '

H Statistic for computing confidence interval of major axis slope

See Also

```
glance(), lmodel2::lmodel2()
Other lmodel2 tidiers: tidy.lmodel2()
```

160 glance.lmRob

Examples

```
# load libraries for models and data
library(lmodel2)

data(mod2ex2)
Ex2.res <- lmodel2(Prey ~ Predators, data = mod2ex2, "relative", "relative", 99)
Ex2.res

# summarize model fit with tidiers + visualization
tidy(Ex2.res)
glance(Ex2.res)

# this allows coefficient plots with ggplot2
library(ggplot2)

ggplot(tidy(Ex2.res), aes(estimate, term, color = method)) +
    geom_point() +
    geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
    geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)))</pre>
```

glance.lmRob

Glance at a(n) lmRob object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'lmRob'
glance(x, ...)
```

glance.lmRob 161

Arguments

x A lmRob object returned from robust::lmRob().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

deviance Deviance of the model.

df.residual Residual degrees of freedom.

nobs Number of observations used.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

See Also

```
robust::lmRob()
Other robust tidiers: augment.lmRob(), glance.glmRob(), tidy.glmRob(), tidy.lmRob()
```

```
# load modeling library
library(robust)

# fit model
m <- lmRob(mpg ~ wt, data = mtcars)

# summarize model fit with tidiers
tidy(m)
augment(m)
glance(m)</pre>
```

162 glance.lmrob

glance.lmrob

Glance at a(n) lmrob object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'lmrob'
glance(x, ...)
```

Arguments Х

A 1mrob object returned from robustbase::1mrob().

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

Value

A tibble::tibble() with exactly one row and columns:

df.residual Residual degrees of freedom.

R squared statistic, or the percent of variation explained by the model. Also r.squared

known as the coefficient of determination.

Estimated standard error of the residuals. sigma

glance.margins 163

See Also

```
robustbase::lmrob()
```

Other robustbase tidiers: augment.glmrob(), augment.lmrob(), tidy.glmrob(), tidy.lmrob()

Examples

```
if (requireNamespace("robustbase", quietly = TRUE)) {
 # load libraries for models and data
 library(robustbase)
 data(coleman)
 set.seed(0)
 m <- lmrob(Y ~ ., data = coleman)</pre>
 tidy(m)
 augment(m)
 glance(m)
 data(carrots)
 Rfit <- glmrob(cbind(success, total - success) ~ logdose + block,
    family = binomial, data = carrots, method = "Mqle",
   control = glmrobMqle.control(tcc = 1.2)
 tidy(Rfit)
 augment(Rfit)
}
```

glance.margins

Glance at a(n) margins object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

164 glance.margins

Usage

```
## S3 method for class 'margins'
glance(x, ...)
```

Arguments

x A margins object returned from margins::margins().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

Number of observations used.

p.value P-value corresponding to the test statistic.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

statistic Test statistic.

```
# load libraries for models and data
library(margins)

# example 1: logit model
mod_log <- glm(am ~ cyl + hp + wt, data = mtcars, family = binomial)

# get tidied "naive" model coefficients
tidy(mod_log)

# convert to marginal effects with margins()
marg_log <- margins(mod_log)</pre>
```

glance.Mclust 165

```
# get tidied marginal effects
tidy(marg_log)
tidy(marg_log, conf.int = TRUE)
# requires running the underlying model again. quick for this example
glance(marg_log)
# augmenting `margins` outputs isn't supported, but
# you can get the same info by running on the underlying model
augment(mod_log)
# example 2: threeway interaction terms
mod_ie <- lm(mpg ~ wt * cyl * disp, data = mtcars)</pre>
# get tidied "naive" model coefficients
tidy(mod_ie)
# convert to marginal effects with margins()
marg_ie0 <- margins(mod_ie)</pre>
# get tidied marginal effects
tidy(marg_ie0)
glance(marg_ie0)
# marginal effects evaluated at specific values of a variable (here: cyl)
marg_ie1 \leftarrow margins(mod_ie, at = list(cyl = c(4,6,8)))
# summarize model fit with tidiers
tidy(marg_ie1)
# marginal effects of one interaction variable (here: wt), modulated at
# specific values of the two other interaction variables (here: cyl and drat)
marg_ie2 <- margins(mod_ie,</pre>
                    variables = "wt",
                    at = list(cyl = c(4,6,8), drat = c(3, 3.5, 4)))
# summarize model fit with tidiers
tidy(marg_ie2)
```

glance.Mclust

Glance at a(n) Mclust object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

166 glance.Mclust

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'Mclust'
glance(x, ...)
```

Arguments

x An Mclust object return from mclust::Mclust().

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

BIC Bayesian Information Criterion for the model.

df Degrees of freedom used by the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

model A string denoting the model type with optimal BIC

Number mixture components in optimal model

hypvol If the other model contains a noise component, the value of the hypervolume

parameter. Otherwise 'NA'.

```
# load library for models and data
library(mclust)

# load data manipulation libraries
library(dplyr)
library(tibble)
```

glance.mfx 167

```
library(purrr)
library(tidyr)
set.seed(27)
centers <- tibble(</pre>
  cluster = factor(1:3),
  # number points in each cluster
  num_points = c(100, 150, 50),
  # x1 coordinate of cluster center
  x1 = c(5, 0, -3),
  # x2 coordinate of cluster center
  x2 = c(-1, 1, -2)
points <- centers %>%
  mutate(
    x1 = map2(num\_points, x1, rnorm),
    x2 = map2(num_points, x2, rnorm)
  select(-num_points, -cluster) %>%
  unnest(c(x1, x2))
# fit model
m <- Mclust(points)</pre>
# summarize model fit with tidiers
tidy(m)
augment(m, points)
glance(m)
```

glance.mfx

Glance at a(n) mfx object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

168 glance.mfx

Usage

```
## S3 method for class 'mfx'
glance(x, ...)

## S3 method for class 'logitmfx'
glance(x, ...)

## S3 method for class 'negbinmfx'
glance(x, ...)

## S3 method for class 'poissonmfx'
glance(x, ...)

## S3 method for class 'probitmfx'
glance(x, ...)
```

Arguments

 $x \hspace{1cm} A \hspace{0.1cm} \texttt{logitmfx}, \hspace{0.1cm} \texttt{negbinmfx}, \hspace{0.1cm} \texttt{poissonmfx}, \hspace{0.1cm} \texttt{or} \hspace{0.1cm} \texttt{probitmfx} \hspace{0.1cm} \texttt{object.} \hspace{0.1cm} (Note \hspace{0.1cm} \texttt{that} \hspace{0.1cm} \texttt{betamfx} \hspace{0.1cm} \texttt{object.} \hspace{0.1cm} \texttt{obje$

objects receive their own set of tidiers.)

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This generic glance method wraps glance.glm() for applicable objects from the mfx package.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df.null Degrees of freedom used by the null model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used. null.deviance Deviance of the null model. glance.mjoint 169

See Also

```
glance.glm(), mfx::logitmfx(), mfx::negbinmfx(), mfx::poissonmfx(), mfx::probitmfx()
Other mfx tidiers: augment.betamfx(), augment.mfx(), glance.betamfx(), tidy.betamfx(),
tidy.mfx()
```

Examples

```
# load libraries for models and data
library(mfx)

# get the marginal effects from a logit regression
mod_logmfx <- logitmfx(am ~ cyl + hp + wt, atmean = TRUE, data = mtcars)

tidy(mod_logmfx, conf.int = TRUE)

# compare with the naive model coefficients of the same logit call
tidy(
    glm(am ~ cyl + hp + wt, family = binomial, data = mtcars),
    conf.int = TRUE
)

augment(mod_logmfx)
glance(mod_logmfx)

# another example, this time using probit regression
mod_probmfx <- probitmfx(am ~ cyl + hp + wt, atmean = TRUE, data = mtcars)

tidy(mod_probmfx, conf.int = TRUE)
augment(mod_probmfx)
glance(mod_probmfx)</pre>
```

glance.mjoint

Glance at a(n) mjoint object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

170 glance.mjoint

Usage

```
## S3 method for class 'mjoint'
glance(x, ...)
```

Arguments

x An mjoint object returned from joineRML::mjoint().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

sigma2_j The square root of the estimated residual variance for the j-th longitudinal pro-

cess

See Also

```
glance(), joineRML::mjoint()
Other mjoint tidiers: tidy.mjoint()
```

```
# broom only skips running these examples because the example models take a
# while to generate—they should run just fine, though!
## Not run:

# load libraries for models and data
library(joineRML)

# fit a joint model with bivariate longitudinal outcomes
data(heart.valve)

hvd <- heart.valve[!is.na(heart.valve$log.grad) &
    !is.na(heart.valve$log.lvmi) &</pre>
```

glance.mlogit 171

```
heart.valve$num <= 50, ]
fit <- mjoint(</pre>
  formLongFixed = list(
    "grad" = log.grad ~ time + sex + hs,
    "lvmi" = log.lvmi \sim time + sex
  formLongRandom = list(
    "grad" = ~ 1 | num,
    "lvmi" = ~ time | num
  ),
  formSurv = Surv(fuyrs, status) ~ age,
  data = hvd,
  inits = list("gamma" = c(0.11, 1.51, 0.80)),
  timeVar = "time"
)
# extract the survival fixed effects
tidy(fit)
# extract the longitudinal fixed effects
tidy(fit, component = "longitudinal")
# extract the survival fixed effects with confidence intervals
tidy(fit, ci = TRUE)
# extract the survival fixed effects with confidence intervals based
# on bootstrapped standard errors
bSE <- bootSE(fit, nboot = 5, safe.boot = TRUE)</pre>
tidy(fit, boot_se = bSE, ci = TRUE)
# augment original data with fitted longitudinal values and residuals
hvd2 <- augment(fit)</pre>
# extract model statistics
glance(fit)
## End(Not run)
```

glance.mlogit

Glance at a(n) mlogit object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

172 glance.mlogit

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'mlogit'
glance(x, ...)
```

Arguments

x an object returned from mlogit::mlogit().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

rho2 McFadden's rho squared with respect to a market shares (constants-only) model.
rho20 McFadden's rho squared with respect to an equal shares (no information) model.

See Also

```
glance(), mlogit::mlogit()
Other mlogit tidiers: augment.mlogit(), tidy.mlogit()
```

```
# load libraries for models and data
library(mlogit)
```

glance.muhaz 173

```
data("Fishing", package = "mlogit")
Fish <- dfidx(Fishing, varying = 2:9, shape = "wide", choice = "mode")
# fit model
m <- mlogit(mode ~ price + catch | income, data = Fish)
# summarize model fit with tidiers
tidy(m)
augment(m)
glance(m)</pre>
```

glance.muhaz

Glance at a(n) muhaz object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'muhaz'
glance(x, ...)
```

Arguments

x A muhaz object returned by muhaz::muhaz().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

174 glance.multinom

Value

A tibble::tibble() with exactly one row and columns:

max.hazard Maximal estimated hazard.

max.time The maximum observed event or censoring time.

min.hazard Minimal estimated hazard.

min. time The minimum observed event or censoring time.

nobs Number of observations used.

See Also

```
glance(), muhaz::muhaz()
Other muhaz tidiers: tidy.muhaz()
```

Examples

```
# load libraries for models and data
library(muhaz)
library(survival)

# fit model
x <- muhaz(ovarian$futime, ovarian$fustat)

# summarize model fit with tidiers
tidy(x)
glance(x)</pre>
```

 ${\tt glance.multinom}$

Glance at a(n) multinom object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

glance.multinom 175

Usage

```
## S3 method for class 'multinom'
glance(x, ...)
```

Arguments

x A multinom object returned from nnet::multinom().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

deviance Deviance of the model.

edf The effective degrees of freedom.

nobs Number of observations used.

See Also

```
glance(), nnet::multinom()
Other multinom tidiers: tidy.multinom()
```

```
# load libraries for models and data
library(nnet)
library(MASS)

example(birthwt)

bwt.mu <- multinom(low ~ ., bwt)

tidy(bwt.mu)
glance(bwt.mu)

# or, for output from a multinomial logistic regression
fit.gear <- multinom(gear ~ mpg + factor(am), data = mtcars)
tidy(fit.gear)</pre>
```

176 glance.negbin

```
glance(fit.gear)
```

glance.negbin

Glance at a(n) negbin object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'negbin'
glance(x, ...)
```

Arguments

. . .

x A negbin object returned by MASS::glm.nb().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

glance.nlrq 177

df.null Degrees of freedom used by the null model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used. null.deviance Deviance of the null model.

See Also

```
glance(), MASS::glm.nb()
Other glm.nb tidiers: tidy.negbin()
```

Examples

```
# load libraries for models and data
library(MASS)

# fit model
r <- glm.nb(Days ~ Sex / (Age + Eth * Lrn), data = quine)

# summarize model fit with tidiers
tidy(r)
glance(r)</pre>
```

glance.nlrq

Glance at a(n) nlrq object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'nlrq'
glance(x, ...)
```

178 glance.nlrq

Arguments

. . .

x A nlrq object returned from quantreg::nlrq().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

tau Quantile.

See Also

```
glance(), quantreg::nlrq()
Other quantreg tidiers: augment.nlrq(), augment.rq(), augment.rqs(), glance.rq(), tidy.nlrq(),
tidy.rq(), tidy.rqs()
```

```
# load modeling library
library(quantreg)

# build artificial data with multiplicative error
set.seed(1)
dat <- NULL
dat$x <- rep(1:25, 20)
dat$y <- SSlogis(dat$x, 10, 12, 2) * rnorm(500, 1, 0.1)

# fit the median using nlrq
mod <- nlrq(y ~ SSlogis(x, Asym, mid, scal),
    data = dat, tau = 0.5, trace = TRUE
)

# summarize model fit with tidiers
tidy(mod)
glance(mod)</pre>
```

glance.nls 179

augment(mod)

glance.nls

Glance at a(n) nls object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'nls'
glance(x, ...)
```

Arguments

. . .

x An nls object returned from stats::nls().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

180 glance.orcutt

df.residual Residual degrees of freedom.

finTol The achieved convergence tolerance.

isConv Whether the fit successfully converged.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

sigma Estimated standard error of the residuals.

See Also

```
tidy, stats::nls()
Other nls tidiers: augment.nls(), tidy.nls()
```

Examples

```
# fit model
n <- nls(mpg ~ k * e^wt, data = mtcars, start = list(k = 1, e = 2))
# summarize model fit with tidiers + visualization
tidy(n)
augment(n)
glance(n)
library(ggplot2)
ggplot(augment(n), aes(wt, mpg)) +
    geom_point() +
    geom_line(aes(y = .fitted))
newdata <- head(mtcars)
newdata$wt <- newdata$wt + 1
augment(n, newdata = newdata)</pre>
```

glance.orcutt

Glance at a(n) orcutt object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

glance.orcutt 181

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'orcutt'
glance(x, ...)
```

Arguments

x An orcutt object returned from orcutt::cochrane.orcutt().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

dw.original Durbin-Watson statistic of original fit.

dw. transformed Durbin-Watson statistic of transformed fit.

nobs Number of observations used.

number.interaction

Number of interactions.

p.value.original

P-value of original Durbin-Watson statistic.

p.value.transformed

P-value of autocorrelation after transformation.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

rho Spearman's rho autocorrelation

See Also

```
glance(), orcutt::cochrane.orcutt()
Other orcutt tidiers: tidy.orcutt()
```

182 glance.pam

Examples

```
# load libraries for models and data
library(orcutt)

# fit model and summarize results
reg <- lm(mpg ~ wt + qsec + disp, mtcars)
tidy(reg)

co <- cochrane.orcutt(reg)
tidy(co)
glance(co)</pre>
```

glance.pam

Glance at a(n) pam object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'pam'
glance(x, ...)
```

Arguments

x An pam object returned from cluster::pam()

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

glance.plm 183

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with exactly one row and columns:

avg.silhouette.width

The average silhouette width for the dataset.
```

See Also

```
glance(), cluster::pam()
Other pam tidiers: augment.pam(), tidy.pam()
```

```
# load libraries for models and data
library(dplyr)
library(ggplot2)
library(cluster)
library(modeldata)
data(hpc_data)
x <- hpc_data[, 2:5]</pre>
p \leftarrow pam(x, k = 4)
# summarize model fit with tidiers + visualization
tidy(p)
glance(p)
augment(p, x)
augment(p, x) %>%
  ggplot(aes(compounds, input_fields)) +
  geom_point(aes(color = .cluster)) +
  geom_text(aes(label = cluster), data = tidy(p), size = 10)
```

184 glance.plm

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'plm'
glance(x, ...)
```

Arguments

x A plm objected returned by plm::plm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

deviance Deviance of the model.

df.residual Residual degrees of freedom.

Number of observations used.

p.value P-value corresponding to the test statistic.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

statistic F-statistic

glance.poLCA 185

See Also

```
glance(), plm::plm()
Other plm tidiers: augment.plm(), tidy.plm()
```

Examples

```
# load libraries for models and data
library(plm)

# load data
data("Produc", package = "plm")

# fit model
zz <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp,
data = Produc, index = c("state", "year")
)

# summarize model fit with tidiers
summary(zz)

tidy(zz)
tidy(zz, conf.int = TRUE)
tidy(zz, conf.int = TRUE, conf.level = 0.9)
augment(zz)
glance(zz)</pre>
```

glance.poLCA

Glance at a(n) poLCA object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

186 glance.poLCA

Usage

```
## S3 method for class 'poLCA'
glance(x, ...)
```

Arguments

x A poLCA object returned from poLCA::poLCA().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

chi.squared The Pearson Chi-Square goodness of fit statistic for multiway tables.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

g. squared The likelihood ratio/deviance statistic

See Also

```
glance(), poLCA::poLCA()
Other poLCA tidiers: augment.poLCA(), tidy.poLCA()
```

```
# load libraries for models and data
library(poLCA)
library(dplyr)

# generate data
data(values)

f <- cbind(A, B, C, D) ~ 1</pre>
```

glance.polr 187

```
M1 <- poLCA(f, values, nclass = 2, verbose = FALSE)
M1
# summarize model fit with tidiers + visualization
tidy(M1)
augment(M1)
glance(M1)
library(ggplot2)
ggplot(tidy(M1), aes(factor(class), estimate, fill = factor(outcome))) +
  geom_bar(stat = "identity", width = 1) +
  facet_wrap(~variable)
# three-class model with a single covariate.
data(election)
f2a <- cbind(
  MORALG, CARESG, KNOWG, LEADG, DISHONG, INTELG,
  MORALB, CARESB, KNOWB, LEADB, DISHONB, INTELB
) ~ PARTY
nes2a <- poLCA(f2a, election, nclass = 3, nrep = 5, verbose = FALSE)</pre>
td <- tidy(nes2a)
ggplot(td, aes(outcome, estimate, color = factor(class), group = class)) +
  geom_line() +
  facet_wrap(\sim variable, nrow = 2) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
au <- augment(nes2a)</pre>
au
count(au, .class)
# if the original data is provided, it leads to NAs in new columns
# for rows that weren't predicted
au2 <- augment(nes2a, data = election)</pre>
au2
dim(au2)
```

188 glance.polr

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'polr'
glance(x, ...)
```

Arguments

x A polr object returned from MASS::polr().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df.residual Residual degrees of freedom.

edf The effective degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

glance.pyears 189

See Also

```
tidy, MASS::polr()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.clmm(), glance.svyolr(),
tidy.clm(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

Examples

```
# load libraries for models and data
library(MASS)

# fit model
fit <- polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)

# summarize model fit with tidiers
tidy(fit, exponentiate = TRUE, conf.int = TRUE)

glance(fit)
augment(fit, type.predict = "class")

fit2 <- polr(factor(gear) ~ am + mpg + qsec, data = mtcars)

tidy(fit, p.values = TRUE)</pre>
```

glance.pyears

Glance at a(n) pyears object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'pyears'
glance(x, ...)
```

190 glance.pyears

Arguments

x A pyears object returned from survival::pyears().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

nobs Number of observations used.

total number of person-years tabulated offtable total number of person-years off table

See Also

```
glance(), survival::pyears()
Other pyears tidiers: tidy.pyears()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.survdiff(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# generate and format data
temp.yr <- tcut(mgus$dxyr, 55:92, labels = as.character(55:91))
temp.age <- tcut(mgus$age, 34:101, labels = as.character(34:100))
ptime <- ifelse(is.na(mgus$pctime), mgus$futime, mgus$pctime)
pstat <- ifelse(is.na(mgus$pctime), 0, 1)
pfit <- pyears(Surv(ptime / 365.25, pstat) ~ temp.yr + temp.age + sex, mgus, data.frame = TRUE
)

# summarize model fit with tidiers
tidy(pfit)
glance(pfit)</pre>
```

glance.ridgelm 191

```
# if data.frame argument is not given, different information is present in
# output
pfit2 <- pyears(Surv(ptime / 365.25, pstat) ~ temp.yr + temp.age + sex, mgus)
tidy(pfit2)
glance(pfit2)</pre>
```

glance.ridgelm

Glance at a(n) ridgelm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'ridgelm'
glance(x, ...)
```

Arguments

Х

A ridgelm object returned from MASS::lm.ridge().

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This is similar to the output of select.ridgelm, but it is returned rather than printed.

192 glance.rlm

Value

A tibble::tibble() with exactly one row and columns:

kHKB modified HKB estimate of the ridge constant
kLW modified L-W estimate of the ridge constant
lambdaGCV choice of lambda that minimizes GCV

See Also

```
glance(), MASS::select.ridgelm(), MASS::lm.ridge()
Other ridgelm tidiers: tidy.ridgelm()
```

```
# load libraries for models and data
library(MASS)
names(longley)[1] <- "y"</pre>
# fit model and summarizd results
fit1 <- lm.ridge(y ~ ., longley)</pre>
tidy(fit1)
fit2 <- lm.ridge(y ~ ., longley, lambda = seq(0.001, .05, .001))
td2 <- tidy(fit2)
g2 <- glance(fit2)
# coefficient plot
library(ggplot2)
ggplot(td2, aes(lambda, estimate, color = term)) +
  geom_line()
# GCV plot
ggplot(td2, aes(lambda, GCV)) +
  geom_line()
# add line for the GCV minimizing estimate
ggplot(td2, aes(lambda, GCV)) +
  geom_line() +
  geom_vline(xintercept = g2$lambdaGCV, col = "red", lty = 2)
```

glance.rlm 193

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'rlm'
glance(x, ...)
```

Arguments

x An rlm object returned by MASS::rlm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

converged Logical indicating if the model fitting procedure was successful and converged.

deviance Deviance of the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

sigma Estimated standard error of the residuals.

See Also

```
glance(), MASS::rlm()
Other rlm tidiers: augment.rlm(), tidy.rlm()
```

194 glance.rma

Examples

```
# load libraries for models and data
library(MASS)

# fit model
r <- rlm(stack.loss ~ ., stackloss)

# summarize model fit with tidiers
tidy(r)
augment(r)
glance(r)</pre>
```

glance.rma

Glance at a(n) rma object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'rma'
glance(x, ...)
```

Arguments

An rma object such as those created by metafor::rma(), metafor::rma.uni(),
metafor::rma.glmm(), metafor::rma.mh(), metafor::rma.mv(), or metafor::rma.peto().

.. Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

glance.rma 195

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

In meta-analysis, test statistic for the Cochran's Q_e test of residual heterogenecochran.qe cochran.qm In meta-analysis, test statistic for the Cochran's Q_m omnibus test of coefficients. df.residual Residual degrees of freedom. h.squared Value of the H-Squared statistic. Value of the I-Squared statistic. i.squared measure The measure used in the meta-analysis. Which method was used. method nobs Number of observations used. p.value.cochran.qe In meta-analysis, p-value for the Cochran's Q_e test of residual heterogeneity. p.value.cochran.qm In meta-analysis, p-value for the Cochran's Q_m omnibus test of coefficients. In meta-analysis, estimated amount of residual heterogeneity. tau.squared tau.squared.se In meta-analysis, standard error of residual heterogeneity.

```
library(metafor)

df <-
    escalc(
    measure = "RR",
    ai = tpos,
    bi = tneg,
    ci = cpos,
    di = cneg,
    data = dat.bcg
)

meta_analysis <- rma(yi, vi, data = df, method = "EB")
glance(meta_analysis)</pre>
```

196 glance.rq

glance.rq

Glance at a(n) rq object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'rq'
glance(x, ...)
```

Arguments

Х

An rq object returned from quantreg::rq().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

Only models with a single tau value may be passed. For multiple values, please use a purrr::map() workflow instead, e.g.

```
taus %>%
  map(function(tau_val) rq(y ~ x, tau = tau_val)) %>%
  map_dfr(glance)
```

glance.rq 197

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

tau Quantile.

See Also

```
glance(), quantreg::rq()
Other quantreg tidiers: augment.nlrq(), augment.rq(), augment.rqs(), glance.nlrq(), tidy.nlrq(),
tidy.rq(), tidy.rqs()
```

```
# load modeling library and data
library(quantreg)
data(stackloss)
# median (11) regression fit for the stackloss data.
mod1 <- rq(stack.loss ~ stack.x, .5)</pre>
# weighted sample median
mod2 \leftarrow rq(rnorm(50) \sim 1, weights = runif(50))
# summarize model fit with tidiers
tidy(mod1)
glance(mod1)
augment(mod1)
tidy(mod2)
glance(mod2)
augment(mod2)
# varying tau to generate an rqs object
mod3 \leftarrow rq(stack.loss \sim stack.x, tau = c(.25, .5))
tidy(mod3)
augment(mod3)
# glance cannot handle rqs objects like `mod3`--use a purrr
# `map`-based workflow instead
```

198 glance.sarlm

alamaa aamlm	Clause at a(u) anatialnes chiest
glance.sarlm	Glance at a(n) spatialreg object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'sarlm'
glance(x, ...)
```

Arguments

x An object returned from spatialreg::lagsarlm() or spatialreg::errorsarlm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

glance.sarlm 199

See Also

```
glance(), spatialreg::lagsarlm(), spatialreg::errorsarlm(), spatialreg::sacsarlm()
Other spatialreg tidiers: augment.sarlm(), tidy.sarlm()
```

```
# load libraries for models and data
library(spatialreg)
library(spdep)
# load data
data(oldcol, package = "spdep")
listw <- nb2listw(COL.nb, style = "W")</pre>
# fit model
crime_sar <-</pre>
  lagsarlm(CRIME ~ INC + HOVAL,
    data = COL.OLD,
   listw = listw,
    method = "eigen"
# summarize model fit with tidiers
tidy(crime_sar)
tidy(crime_sar, conf.int = TRUE)
glance(crime_sar)
augment(crime_sar)
# fit another model
crime_sem <- errorsarlm(CRIME ~ INC + HOVAL, data = COL.OLD, listw)</pre>
# summarize model fit with tidiers
tidy(crime_sem)
tidy(crime_sem, conf.int = TRUE)
glance(crime_sem)
augment(crime_sem)
# fit another model
crime_sac <- sacsarlm(CRIME ~ INC + HOVAL, data = COL.OLD, listw)</pre>
# summarize model fit with tidiers
tidy(crime_sac)
tidy(crime_sac, conf.int = TRUE)
glance(crime_sac)
augment(crime_sac)
```

200 glance.smooth.spline

```
glance.smooth.spline Tidy \ a(n) \ smooth.spine \ object
```

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'smooth.spline'
glance(x, ...)
```

Arguments

x A smooth.spline object returned from stats::smooth.spline().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

crit Minimized criterion
cv.crit Cross-validation score

df Degrees of freedom used by the model.

lambda Choice of lambda corresponding to 'spar'.

nobs Number of observations used.

pen.crit Penalized criterion. spar Smoothing parameter.

See Also

```
augment(), stats::smooth.spline()
Other smoothing spline tidiers: augment.smooth.spline()
```

glance.speedglm 201

Examples

```
# fit model
spl <- smooth.spline(mtcars$wt, mtcars$mpg, df = 4)
# summarize model fit with tidiers
augment(spl, mtcars)
# calls original columns x and y
augment(spl)
library(ggplot2)
ggplot(augment(spl, mtcars), aes(wt, mpg)) +
    geom_point() +
    geom_line(aes(y = .fitted))</pre>
```

glance.speedglm

Glance at a(n) speedglm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'speedglm'
glance(x, ...)
```

Arguments

x A speedglm object returned from speedglm::speedglm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

202 glance.speedglm

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df.null Degrees of freedom used by the null model.

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.
null.deviance Deviance of the null model.

See Also

```
speedglm::speedlm()
Other speedlm tidiers: augment.speedlm(), glance.speedlm(), tidy.speedglm(), tidy.speedlm()
```

```
# load libraries for models and data
library(speedglm)

# generate data
clotting <- data.frame(
    u = c(5, 10, 15, 20, 30, 40, 60, 80, 100),
    lot1 = c(118, 58, 42, 35, 27, 25, 21, 19, 18)
)

# fit model
fit <- speedglm(lot1 ~ log(u), data = clotting, family = Gamma(log))

# summarize model fit with tidiers
tidy(fit)
glance(fit)</pre>
```

glance.speedIm 203

glance.speedlm

Glance at a(n) speedlm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'speedlm'
glance(x, ...)
```

Arguments

. . .

x A speedlm object returned from speedglm::speedlm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df Degrees of freedom used by the model.

204 glance.summary.lm

df.residual Residual degrees of freedom.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

p. value P-value corresponding to the test statistic.

r.squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

statistic F-statistic.

See Also

```
speedglm::speedlm()
Other speedlm tidiers: augment.speedlm(), glance.speedglm(), tidy.speedglm(), tidy.speedlm()
```

Examples

```
# load modeling library
library(speedglm)

# fit model
mod <- speedlm(mpg ~ wt + qsec, data = mtcars, fitted = TRUE)

# summarize model fit with tidiers
tidy(mod)
glance(mod)
augment(mod)</pre>
```

glance.summary.lm

Glance at a(n) summary.lm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

glance.summary.lm 205

Usage

```
## S3 method for class 'summary.lm'
glance(x, ...)
```

Arguments

. . .

x An lm object created by stats::lm().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The glance.summary.lm() method is a potentially useful alternative to glance.lm(). For instance, if users have already converted large lm objects into their leaner summary.lm equivalents to conserve memory. Note, however, that this method does not return all of the columns of the non-summary method (e.g. AIC and BIC will be missing.)

Value

A tibble::tibble() with exactly one row and columns:

adj.r.squared Adjusted R squared statistic, which is like the R squared statistic except taking

degrees of freedom into account.

df.residual Residual degrees of freedom.

nobs Number of observations used.

p. value P-value corresponding to the test statistic.

r. squared R squared statistic, or the percent of variation explained by the model. Also

known as the coefficient of determination.

sigma Estimated standard error of the residuals.

statistic Test statistic.

df The degrees for freedom from the numerator of the overall F-statistic. This

is new in broom 0.7.0. Previously, this reported the rank of the design matrix, which is one more than the numerator degrees of freedom of the overall

F-statistic.

See Also

```
glance(), glance.summary.lm()
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.svyglm(),
tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

206 glance.summary.lm

```
library(ggplot2)
library(dplyr)
mod <- lm(mpg ~ wt + qsec, data = mtcars)</pre>
tidy(mod)
glance(mod)
# coefficient plot
d <- tidy(mod, conf.int = TRUE)</pre>
ggplot(d, aes(estimate, term, xmin = conf.low, xmax = conf.high, height = 0)) +
  geom_point() +
  geom_vline(xintercept = 0, lty = 4) +
  geom_errorbarh()
# aside: There are tidy() and glance() methods for lm.summary objects too.
# this can be useful when you want to conserve memory by converting large lm
# objects into their leaner summary.lm equivalents.
s <- summary(mod)</pre>
tidy(s, conf.int = TRUE)
glance(s)
augment(mod)
augment(mod, mtcars, interval = "confidence")
# predict on new data
newdata <- mtcars %>%
  head(6) %>%
  mutate(wt = wt + 1)
augment(mod, newdata = newdata)
# ggplot2 example where we also construct 95% prediction interval
# simpler bivariate model since we're plotting in 2D
mod2 <- lm(mpg \sim wt, data = mtcars)
au <- augment(mod2, newdata = newdata, interval = "prediction")</pre>
ggplot(au, aes(wt, mpg)) +
  geom_point() +
  geom\_line(aes(y = .fitted)) +
  geom_ribbon(aes(ymin = .lower, ymax = .upper), col = NA, alpha = 0.3)
# predict on new data without outcome variable. Output does not include .resid
newdata <- newdata %>%
  select(-mpg)
augment(mod, newdata = newdata)
```

glance.survdiff 207

```
au <- augment(mod, data = mtcars)</pre>
ggplot(au, aes(.hat, .std.resid)) +
 geom_vline(size = 2, colour = "white", xintercept = 0) +
 geom_hline(size = 2, colour = "white", yintercept = 0) +
 geom_point() +
 geom_smooth(se = FALSE)
plot(mod, which = 6)
ggplot(au, aes(.hat, .cooksd)) +
 geom_vline(xintercept = 0, colour = NA) +
 geom_abline(slope = seq(0, 3, by = 0.5), colour = "white") +
 geom_smooth(se = FALSE) +
 geom_point()
# column-wise models
a <- matrix(rnorm(20), nrow = 10)
b <- a + rnorm(length(a))</pre>
result <- lm(b \sim a)
tidy(result)
```

glance.survdiff

Glance at a(n) survdiff object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'survdiff'
glance(x, ...)
```

208 glance.survdiff

Arguments

. . .

x An survdiff object returned from survival::survdiff().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

df Degrees of freedom used by the model.
p.value P-value corresponding to the test statistic.

statistic Test statistic.

See Also

```
glance(), survival::survdiff()
Other survdiff tidiers: tidy.survdiff()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survexp(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# fit model
s <- survdiff(
   Surv(time, status) ~ pat.karno + strata(inst),
   data = lung
)

# summarize model fit with tidiers
tidy(s)
glance(s)</pre>
```

glance.survexp 209

glance.survexp

Glance at a(n) survexp object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'survexp'
glance(x, ...)
```

Arguments

x An survexp object returned from survival::survexp().

.. Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

n.max Maximum number of subjects at risk.
n.start Initial number of subjects at risk.

timepoints Number of timepoints.

210 glance.survfit

See Also

```
glance(), survival::survexp()
Other survexp tidiers: tidy.survexp()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survfit(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

Examples

```
# load libraries for models and data
library(survival)

# fit model
sexpfit <- survexp(
  futime ~ 1,
  rmap = list(
    sex = "male",
    year = accept.dt,
    age = (accept.dt - birth.dt)
  ),
  method = "conditional",
  data = jasa
)

# summarize model fit with tidiers
tidy(sexpfit)
glance(sexpfit)</pre>
```

glance.survfit

Glance at a(n) survfit object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

glance.survfit 211

Usage

```
## S3 method for class 'survfit'
glance(x, ...)
```

Arguments

x An survfit object returned from survival::survfit().

Additional arguments passed to survival::summary.survfit(). Important

arguments include rmean.

Value

A tibble::tibble() with exactly one row and columns:

events Number of events.

n.max Maximum number of subjects at risk.
n.start Initial number of subjects at risk.

Number of observations used.

records Number of observations

rmean Restricted mean (see [survival::print.survfit()]).

rmean.std.error

Restricted mean standard error.

conf.low lower end of confidence interval on median conf.high upper end of confidence interval on median

median median survival

See Also

```
glance(), survival::survfit()
Other cch tidiers: glance.cch(), tidy.cch()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survreg(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# fit model
cfit <- coxph(Surv(time, status) ~ age + sex, lung)
sfit <- survfit(cfit)

# summarize model fit with tidiers + visualization</pre>
```

212 glance.survreg

```
tidy(sfit)
glance(sfit)

library(ggplot2)

ggplot(tidy(sfit), aes(time, estimate)) +
    geom_line() +
    geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)

# multi-state
fitCI <- survfit(Surv(stop, status * as.numeric(event), type = "mstate") ~ 1,
    data = mgus1, subset = (start == 0)
)

td_multi <- tidy(fitCI)

td_multi
ggplot(td_multi, aes(time, estimate, group = state)) +
    geom_line(aes(color = state)) +
    geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)</pre>
```

glance.survreg

Glance at a(n) survreg object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'survreg'
glance(x, ...)
```

glance.survreg 213

Arguments

x An survreg object returned from survival::survreg().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

df Degrees of freedom used by the model.

df.residual Residual degrees of freedom.

iter Iterations of algorithm/fitting procedure completed.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

nobs Number of observations used.

p.value P-value corresponding to the test statistic.

statistic Chi-squared statistic.

See Also

```
glance(), survival::survreg()
Other survreg tidiers: augment.survreg(), tidy.survreg()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# fit model
sr <- survreg(
   Surv(futime, fustat) ~ ecog.ps + rx,
   ovarian,
   dist = "exponential"</pre>
```

214 glance.svyglm

```
# summarize model fit with tidiers + visualization
tidy(sr)
augment(sr, ovarian)
glance(sr)

# coefficient plot
td <- tidy(sr, conf.int = TRUE)

library(ggplot2)

ggplot(td, aes(estimate, term)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
  geom_vline(xintercept = 0)</pre>
```

glance.svyglm

Glance at a(n) svyglm object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'svyglm'
glance(x, maximal = x, ...)
```

Arguments

x A svyglm object returned from survey::svyglm().

maximal

A svyglm object corresponding to the maximal model against which to compute the BIC. See Lumley and Scott (2015) for details. Defaults to x, which is equivalent to not using a maximal model.

glance.svyglm 215

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

AIC Akaike's Information Criterion for the model.

BIC Bayesian Information Criterion for the model.

deviance Deviance of the model.

df.null Degrees of freedom used by the null model.

df.residual Residual degrees of freedom. null.deviance Deviance of the null model.

References

Lumley T, Scott A (2015). AIC and BIC for modelling with complex survey data. *Journal of Survey Statistics and Methodology*, 3(1).

See Also

```
survey::svyglm(), stats::glm(), survey::anova.svyglm
Other lm tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(),
tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

```
# load libraries for models and data
library(survey)

set.seed(123)
data(api)

# survey design
dstrat <-
    svydesign(
    id = ~1,
        strata = ~stype,
        weights = ~pw,
        data = apistrat,
        fpc = ~fpc</pre>
```

216 glance.svyolr

```
# model
m <- svyglm(
  formula = sch.wide ~ ell + meals + mobility,
  design = dstrat,
  family = quasibinomial()
)
glance(m)</pre>
```

glance.svyolr

Glance at a(n) svyolr object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

Usage

```
## S3 method for class 'svyolr'
glance(x, ...)
```

Arguments

x A svyolr object r

A svyolr object returned from survey::svyolr().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

glance.varest 217

Value

```
A tibble::tibble() with exactly one row and columns:
```

```
df.residual Residual degrees of freedom.

edf The effective degrees of freedom.

Number of observations used.
```

See Also

```
tidy, survey::svyolr()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.clmm(), glance.polr(),
tidy.clm(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

Examples

```
library(broom)
library(survey)

data(api)
dclus1 <- svydesign(id = ~dnum, weights = ~pw, data = apiclus1, fpc = ~fpc)
dclus1 <- update(dclus1, mealcat = cut(meals, c(0, 25, 50, 75, 100)))

m <- svyolr(mealcat ~ avg.ed + mobility + stype, design = dclus1)

m

tidy(m, conf.int = TRUE)</pre>
```

glance.varest

Glance at a(n) varest object

Description

Glance accepts a model object and returns a tibble::tibble() with exactly one row of model summaries. The summaries are typically goodness of fit measures, p-values for hypothesis tests on residuals, or model convergence information.

Glance never returns information from the original call to the modeling function. This includes the name of the modeling function or any arguments passed to the modeling function.

Glance does not calculate summary measures. Rather, it farms out these computations to appropriate methods and gathers the results together. Sometimes a goodness of fit measure will be undefined. In these cases the measure will be reported as NA.

Glance returns the same number of columns regardless of whether the model matrix is rank-deficient or not. If so, entries in columns that no longer have a well-defined value are filled in with an NA of the appropriate type.

218 glance.varest

Usage

```
## S3 method for class 'varest'
glance(x, ...)
```

Arguments

A varest object produced by a call to vars::VAR().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

lag.order Lag order.

logLik The log-likelihood of the model. [stats::logLik()] may be a useful reference.

n The total number of observations.

Number of observations used.

See Also

```
glance(), vars::VAR()
```

```
# load libraries for models and data
library(vars)

# load data
data("Canada", package = "vars")

# fit models
mod <- VAR(Canada, p = 1, type = "both")

# summarize model fit with tidiers
tidy(mod)
glance(mod)</pre>
```

glance_optim 219

glance_optim

Tidy a(n) optim object masquerading as list

Description

Broom tidies a number of lists that are effectively S3 objects without a class attribute. For example, stats::optim(), svd() and interp::interp() produce consistent output, but because they do not have a class attribute, they cannot be handled by S3 dispatch.

These functions look at the elements of a list and determine if there is an appropriate tidying method to apply to the list. Those tidiers are implemented as functions of the form tidy_<function> or glance_<function> and are not exported (but they are documented!).

If no appropriate tidying method is found, they throw an error.

Usage

```
glance_optim(x, ...)
```

Arguments

x A list returned from stats::optim().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with exactly one row and columns:

```
convergence Convergence code.

function.count Number of calls to 'fn'.

gradient.count Number of calls to 'gr'.

value Minimized or maximized output value.
```

```
glance(), stats::optim()
Other list tidiers: list_tidiers, tidy_irlba(), tidy_optim(), tidy_svd(), tidy_xyz()
```

220 leveneTest_tidiers

Examples

```
f \leftarrow function(x) (x[1] - 2)^2 + (x[2] - 3)^2 + (x[3] - 8)^2
o \leftarrow optim(c(1, 1, 1), f)
```

leveneTest_tidiers

Tidy/glance a(n) leveneTest object

Description

For models that have only a single component, the tidy() and glance() methods are identical. Please see the documentation for both of those methods.

Usage

```
## S3 method for class 'leveneTest'
tidy(x, ...)
```

Arguments

x An object of class anova created by a call to car::leveneTest().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

df Degrees of freedom used by this term in the model.

p. value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

df.residual Residual degrees of freedom.

```
tidy(), glance(), car::leveneTest()
Other car tidiers: durbinWatsonTest_tidiers
```

list_tidiers 221

Examples

```
# load libraries for models and data
library(car)

data(Moore)

lt <- with(Moore, leveneTest(conformity, fcategory))

tidy(lt)
glance(lt)</pre>
```

list_tidiers

Tidying methods for lists / returned values that are not S3 objects

Description

Broom tidies a number of lists that are effectively S3 objects without a class attribute. For example, stats::optim(), base::svd() and interp::interp() produce consistent output, but because they do not have a class attribute, they cannot be handled by S3 dispatch.

Usage

```
## S3 method for class 'list'
tidy(x, ...)
## S3 method for class 'list'
glance(x, ...)
```

Arguments

- x A list, potentially representing an object that can be tidied.
- . . . Additionally, arguments passed to the tidying function.

Details

These functions look at the elements of a list and determine if there is an appropriate tidying method to apply to the list. Those tidiers are themselves are implemented as functions of the form tidy_<function> or glance_<function> and are not exported (but they are documented!).

If no appropriate tidying method is found, throws an error.

```
Other list tidiers: glance_optim(), tidy_irlba(), tidy_optim(), tidy_svd(), tidy_xyz()
```

222 sp_tidiers

null_tidiers

Tidiers for NULL inputs

Description

tidy(NULL), glance(NULL) and augment(NULL) all return an empty tibble::tibble. This empty tibble can be treated a tibble with zero rows, making it convenient to combine with other tibbles using functions like purrr::map_df() on lists of potentially NULL objects.

Usage

```
## $3 method for class '`NULL`'
tidy(x, ...)
## $3 method for class '`NULL`'
glance(x, ...)
## $3 method for class '`NULL`'
augment(x, ...)
```

Arguments

x The value NULL.

. . . Additional arguments (not used).

Value

An empty tibble::tibble.

See Also

tibble::tibble

sp_tidiers

Tidy a(n) SpatialPolygonsDataFrame object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Note that the sf package now defines tidy spatial objects and is the recommended approach to spatial data. sp tidiers are now deprecated in favor of sf::st_as_sf() and coercion methods found in other packages. See https://r-spatial.org/r/2023/05/15/evolution4.html for more on migration from retiring spatial packages.

summary_tidiers 223

Usage

```
## S3 method for class 'SpatialPolygonsDataFrame'
tidy(x, region = NULL, ...)

## S3 method for class 'SpatialPolygons'
tidy(x, ...)

## S3 method for class 'Polygons'
tidy(x, ...)

## S3 method for class 'Polygon'
tidy(x, ...)

## S3 method for class 'SpatialLinesDataFrame'
tidy(x, ...)

## S3 method for class 'Lines'
tidy(x, ...)

## S3 method for class 'Lines'
tidy(x, ...)
```

Arguments

x A SpatialPolygonsDataFrame, SpatialPolygons, Polygon, SpatialLinesDataFrame, Lines or Line object.

region name of variable used to split up regions
... not used by this method

```
summary_tidiers (Deprecated) Tidy summaryDefault objects
```

Description

Tidiers for summary Default objects have been deprecated as of broom 0.7.0 in favor of skimr::skim().

Usage

```
## S3 method for class 'summaryDefault'
tidy(x, ...)
## S3 method for class 'summaryDefault'
glance(x, ...)
```

224 summary_tidiers

Arguments

x A summaryDefault object, created by calling summary() on a vector.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A one-row tibble::tibble with columns:

minimum Minimum value in original vector.

q1 First quartile of original vector.

median Median of original vector.

mean Mean of original vector.

q3 Third quartile of original vector.

maximum Maximum value in original vector.

na Number of NA values in original vector. Column present only when original

vector had at least one NA entry.

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()

Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), tidy.density(), tidy.dist(), tidy.ftable(), tidy.numeric()
```

```
v <- rnorm(1000)
s <- summary(v)
s

tidy(s)
glance(s)

v2 <- c(v,NA)
tidy(summary(v2))</pre>
```

225 tidy.aareg

tidy an eg hay a(n) dureg object	tidy.aareg	Tidy a(n) aareg object	
	tiuy.aareg	riay a(n) dareg object	

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'aareg'
tidy(x, ...)
```

Arguments

. . .

An aareg object returned from survival::aareg().

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

robust. se is only present when x was created with dfbeta = TRUE.

Value

z

A tibble::tibble() with columns:

estimate The estimated value of the regression term. p.value The two-sided p-value associated with the observed statistic. robust.se robust version of standard error estimate. statistic The value of a T-statistic to use in a hypothesis that the regression term is nonstd.error The standard error of the regression term. The name of the regression term. term z score.

226 tidy.acf

See Also

```
tidy(), survival::aareg()
Other aareg tidiers: glance.aareg()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

Examples

```
# load libraries for models and data
library(survival)

# fit model
afit <- aareg(
   Surv(time, status) ~ age + sex + ph.ecog,
   data = lung,
   dfbeta = TRUE
)

# summarize model fit with tidiers
tidy(afit)</pre>
```

tidy.acf

Tidy a(n) acf object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'acf'
tidy(x, ...)
```

Arguments

x An acf object created by stats::acf(), stats::pacf() or stats::ccf().

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.anova 227

 tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:

acf Autocorrelation.

lag Lag values.
```

See Also

```
tidy(), stats::acf(), stats::pacf(), stats::ccf()
Other time series tidiers: tidy.spec(), tidy.ts(), tidy.zoo()
```

Examples

```
tidy(acf(lh, plot = FALSE))
tidy(ccf(mdeaths, fdeaths, plot = FALSE))
tidy(pacf(lh, plot = FALSE))
```

tidy.anova

Tidy a(n) anova object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'anova' tidy(x, ...)
```

Arguments

An anova object, such as those created by stats::anova(), car::Anova(), car::leveneTest(), or car::linearHypothesis().

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

228 tidy.anova

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The term column of an ANOVA table can come with leading or trailing whitespace, which this tidying method trims.

For documentation on the tidier for car::leveneTest() output, see tidy.leveneTest()

Value

A tibble::tibble() with columns:

df Degrees of freedom used by this term in the model.

meansq Mean sum of squares. Equal to total sum of squares divided by degrees of

freedom.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero

sumsq Sum of squares explained by this term.
term The name of the regression term.

See Also

```
tidy(), stats::anova(), car::Anova(), car::leveneTest()
Other anova tidiers: glance.anova(), glance.aov(), tidy.TukeyHSD(), tidy.aov(), tidy.aovlist(), tidy.manova()
```

```
# fit models
a <- lm(mpg ~ wt + qsec + disp, mtcars)
b <- lm(mpg ~ wt + qsec, mtcars)

mod <- anova(a, b)

# summarize model fit with tidiers
tidy(mod)
glance(mod)

# car::linearHypothesis() example
library(car)
mod_lht <- linearHypothesis(a, "wt - disp")
tidy(mod_lht)
glance(mod_lht)</pre>
```

tidy.aov 229

tidy.aov

 $Tidy \ a(n) \ aov \ object$

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'aov'
tidy(x, intercept = FALSE, ...)
```

Arguments

x An aov object, such as those created by stats::aov().

intercept A logical indicating whether information on the intercept ought to be included.

Passed to stats::summary.aov().

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The term column of an ANOVA table can come with leading or trailing whitespace, which this tidying method trims.

For documentation on the tidier for car::leveneTest() output, see tidy.leveneTest()

See Also

```
tidy(), stats::aov()
Other anova tidiers: glance.anova(), glance.aov(), tidy.TukeyHSD(), tidy.anova(), tidy.aovlist(),
tidy.manova()
```

```
a <- aov(mpg ~ wt + qsec + disp, mtcars)
tidy(a)</pre>
```

230 tidy.aovlist

tidy.aovlist	Tidy a(n) aovlist object	

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'aovlist'
tidy(x, ...)
```

Arguments

. . .

x An aovlist objects, such as those created by stats::aov().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The term column of an ANOVA table can come with leading or trailing whitespace, which this tidying method trims.

For documentation on the tidier for car::leveneTest() output, see tidy.leveneTest()

Value

A tibble::tibble() with columns:

df Degrees of freedom used by this term in the model.

meansq Mean sum of squares. Equal to total sum of squares divided by degrees of freedom.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

stratum The error stratum.

sumsq Sum of squares explained by this term.
term The name of the regression term.

231 tidy.Arima

See Also

```
tidy(), stats::aov()
Other anova tidiers: glance.anova(), glance.aov(), tidy.TukeyHSD(), tidy.anova(), tidy.aov(),
tidy.manova()
```

Examples

```
a <- aov(mpg ~ wt + qsec + Error(disp / am), mtcars)</pre>
tidy(a)
```

tidy.Arima

Tidy a(n) Arima object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'Arima'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

An object of class Arima created by stats::arima(). Х

conf.int Logical indicating whether or not to include a confidence interval in the tidied

output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cau**tionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

232 tidy.betamfx

Value

```
A tibble::tibble() with columns:
```

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

The standard error of the regression term.

term The name of the regression term.

See Also

```
stats::arima()
Other Arima tidiers: glance.Arima()
```

Examples

```
# fit model
fit <- arima(lh, order = c(1, 0, 0))
# summarize model fit with tidiers
tidy(fit)
glance(fit)</pre>
```

tidy.betamfx

Tidy a(n) betamfx object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'betamfx'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x A betamfx object.

conf.int Logical indicating whether or not to include a confidence interval in the tidied

output. Defaults to FALSE.

tidy.betamfx 233

conf.level

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The mfx package provides methods for calculating marginal effects for various generalized linear models (GLMs). Unlike standard linear models, estimated model coefficients in a GLM cannot be directly interpreted as marginal effects (i.e., the change in the response variable predicted after a one unit change in one of the regressors). This is because the estimated coefficients are multiplicative, dependent on both the link function that was used for the estimation and any other variables that were included in the model. When calculating marginal effects, users must typically choose whether they want to use i) the average observation in the data, or ii) the average of the sample marginal effects. See vignette("mfxarticle") from the mfx package for more details.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.
atmean	TRUE if the marginal effects were originally calculated as the partial effects for the average observation. If FALSE, then these were instead calculated as average partial effects.

```
tidy.betareg(), mfx::betamfx()
Other mfx tidiers: augment.betamfx(), augment.mfx(), glance.betamfx(), glance.mfx(),
tidy.mfx()
```

234 tidy.betareg

Examples

```
library(mfx)
# Simulate some data
set.seed(12345)
n <- 1000
x <- rnorm(n)
# Beta outcome
y < - rbeta(n, shape1 = plogis(1 + 0.5 * x), shape2 = (abs(0.2 * x)))
# Use Smithson and Verkuilen correction
y \leftarrow (y * (n - 1) + 0.5) / n
d <- data.frame(y, x)</pre>
mod_betamfx \leftarrow betamfx(y \sim x \mid x, data = d)
tidy(mod_betamfx, conf.int = TRUE)
# Compare with the naive model coefficients of the equivalent betareg call (not run)
# tidy(betamfx(y \sim x | x, data = d), conf.int = TRUE)
augment(mod_betamfx)
glance(mod_betamfx)
```

tidy.betareg

Tidy a(n) betareg object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'betareg'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

Х	A betareg object produced by a call to betareg::betareg().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

tidy.betareg 235

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

. . .

The tibble has one row for each term in the regression. The component column indicates whether a particular term was used to model either the "mean" or "precision". Here the precision is the inverse of the variance, often referred to as phi. At least one term will have been used to model the precision phi.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.
component	Whether a particular term was used to model the mean or the precision in the regression. See details.

See Also

```
tidy(), betareg::betareg()
```

```
# load libraries for models and data
library(betareg)

# load dats
data("GasolineYield", package = "betareg")

# fit model
mod <- betareg(yield ~ batch + temp, data = GasolineYield)</pre>
```

236 tidy.biglm

```
mod

# summarize model fit with tidiers
tidy(mod)
tidy(mod, conf.int = TRUE)
tidy(mod, conf.int = TRUE, conf.level = .99)
augment(mod)
glance(mod)
```

tidy.biglm

Tidy a(n) biglm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'biglm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

x A biglm object created by a call to biglm::biglm() or biglm::bigglm().

conf. int Logical indicating whether or not to include a confidence interval in the tidied

output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must

be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to $\,$

a 95 percent confidence interval.

exponentiate Logical indicating whether or not to exponentiate the the coefficient estimates.

This is typical for logistic and multinomial regressions, but a bad idea if there is

no log or logit link. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

tidy.biglm 237

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

std.error The standard error of the regression term.

term The name of the regression term.

See Also

```
tidy(), biglm::biglm(), biglm::bigglm()
Other biglm tidiers: glance.biglm()
```

```
# load modeling library
library(biglm)
# fit model -- linear regression
bfit <- biglm(mpg ~ wt + disp, mtcars)</pre>
# summarize model fit with tidiers
tidy(bfit)
tidy(bfit, conf.int = TRUE)
tidy(bfit, conf.int = TRUE, conf.level = .9)
glance(bfit)
# fit model -- logistic regression
bgfit <- bigglm(am ~ mpg, mtcars, family = binomial())</pre>
# summarize model fit with tidiers
tidy(bgfit)
tidy(bgfit, exponentiate = TRUE)
tidy(bgfit, conf.int = TRUE)
tidy(bgfit, conf.int = TRUE, conf.level = .9)
tidy(bgfit, conf.int = TRUE, conf.level = .9, exponentiate = TRUE)
glance(bgfit)
```

238 tidy.binDesign

tidy.binDesign

Tidy a(n) binDesign object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'binDesign'
tidy(x, ...)
```

Arguments

x A binGroup::binDesign() object.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

Number of trials in given iteration.powerPower achieved for given value of n.

See Also

```
tidy(), binGroup::binDesign()
Other bingroup tidiers: glance.binDesign(), tidy.binWidth()
```

```
library(binGroup)
des <- binDesign(
  nmax = 300, delta = 0.06,
  p.hyp = 0.1, power = .8</pre>
```

239 tidy.binWidth

```
)
glance(des)
tidy(des)
# the ggplot2 equivalent of plot(des)
library(ggplot2)
ggplot(tidy(des), aes(n, power)) +
 geom_line()
```

tidy.binWidth

Tidy a(n) binWidth object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'binWidth'
tidy(x, ...)
```

Arguments Х

A binGroup::binWidth() object.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

n

A tibble::tibble() with columns:

alternative Alternative hypothesis (character). ci.width Expected width of confidence interval. True proportion. р

Total sample size

240 tidy.boot

See Also

```
tidy(), binGroup::binWidth()
Other bingroup tidiers: glance.binDesign(), tidy.binDesign()
```

Examples

```
# load libraries
library(binGroup)
# fit model
bw <- binWidth(100, .1)
bw
# summarize model fit with tidiers
tidy(bw)</pre>
```

tidy.boot

Tidy a(n) boot object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'boot'
tidy(
    x,
    conf.int = FALSE,
    conf.level = 0.95,
    conf.method = c("perc", "bca", "basic", "norm"),
    exponentiate = FALSE,
    ...
)
```

Arguments

```
x A boot::boot() object.
```

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

tidy.boot 241

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

conf.method Passed to the type argument of boot::boot.ci(). Defaults to "perc". The al-

lowed types are "perc", "basic", "bca", and "norm". Does not support "stud"

or "all".

exponentiate Logical indicating whether or not to exponentiate the the coefficient estimates.

This is typical for logistic and multinomial regressions, but a bad idea if there is

no log or logit link. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

If weights were provided to the boot function, an estimate column is included showing the weighted bootstrap estimate, and the standard error is of that estimate.

If there are no original statistics in the "boot" object, such as with a call to tsboot with orig.t = FALSE, the original and statistic columns are omitted, and only estimate and std.error columns shown.

Value

A tibble::tibble() with columns:

bias Bias of the statistic.

std.error The standard error of the regression term.

term The name of the regression term. statistic Original value of the statistic.

See Also

```
tidy(), boot::boot(), boot::tsboot(), boot::boot.ci(), rsample::bootstraps()
```

```
# load modeling library
library(boot)

clotting <- data.frame(
    u = c(5, 10, 15, 20, 30, 40, 60, 80, 100),</pre>
```

242 tidy.btergm

```
lot1 = c(118, 58, 42, 35, 27, 25, 21, 19, 18),
lot2 = c(69, 35, 26, 21, 18, 16, 13, 12, 12)
)

# fit models
g1 <- glm(lot2 ~ log(u), data = clotting, family = Gamma)
bootfun <- function(d, i) {
   coef(update(g1, data = d[i, ]))
}
bootres <- boot(clotting, bootfun, R = 999)

# summarize model fits with tidiers
tidy(g1, conf.int = TRUE)
tidy(bootres, conf.int = TRUE)</pre>
```

tidy.btergm

Tidy a(n) btergm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

This method tidies the coefficients of a bootstrapped temporal exponential random graph model estimated with the **xergm**. It simply returns the coefficients and their confidence intervals.

Usage

```
## S3 method for class 'btergm'
tidy(x, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

x A btergm::btergm() object.

conf.level Confidence level for confidence intervals. Defaults to 0.95.

exponentiate Logical indicating whether or not to exponentiate the the coefficient estimates.

This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.btergm 243

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

term The name of the regression term.

See Also

```
tidy(), btergm::btergm()
```

```
library(btergm)
library(network)
set.seed(5)
# create 10 random networks with 10 actors
networks <- list()</pre>
for (i in 1:10) {
  mat <- matrix(rbinom(100, 1, .25), nrow = 10, ncol = 10)
  diag(mat) <- 0
  nw <- network(mat)</pre>
  networks[[i]] <- nw</pre>
}
# create 10 matrices as covariates
covariates <- list()</pre>
for (i in 1:10) {
  mat <- matrix(rnorm(100), nrow = 10, ncol = 10)
  covariates[[i]] <- mat</pre>
# fit the model
mod <- btergm(networks ~ edges + istar(2) + edgecov(covariates), R = 100)</pre>
# summarize model fit with tidiers
tidy(mod)
```

244 tidy.cch

|--|

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'cch'
tidy(x, conf.level = 0.95, ...)
```

Arguments

An cch object returned from survival::cch().

conf.level confidence level for CI

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

tidy.cld 245

See Also

```
tidy(), survival::cch()
Other cch tidiers: glance.cch(), glance.survfit()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.coxph(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)
# examples come from cch documentation
subcoh <- nwtco$in.subcohort</pre>
selccoh <- with(nwtco, rel == 1 | subcoh == 1)</pre>
ccoh.data <- nwtco[selccoh, ]</pre>
ccoh.data$subcohort <- subcoh[selccoh]</pre>
# central-lab histology
ccoh.data$histol <- factor(ccoh.data$histol, labels = c("FH", "UH"))</pre>
# tumour stage
ccoh.data$stage <- factor(ccoh.data$stage, labels = c("I", "II", "III", "IV"))</pre>
ccoh.data$age <- ccoh.data$age / 12 # age in years</pre>
# fit model
fit.ccP <- cch(Surv(edrel, rel) ~ stage + histol + age,</pre>
 data = ccoh.data,
 subcoh = ~subcohort, id = ~seqno, cohort.size = 4028
)
# summarize model fit with tidiers + visualization
tidy(fit.ccP)
# coefficient plot
library(ggplot2)
ggplot(tidy(fit.ccP), aes(x = estimate, y = term)) +
 geom_point() +
 geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
 geom_vline(xintercept = 0)
```

246 tidy.cld

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'cld'
tidy(x, ...)
```

Arguments

A cld object created by calling multcomp::cld() on a glht, confint.glht() or summary.glht() object.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:

contrast Levels being compared.

letters Compact letter display denoting all pair-wise comparisons.
```

See Also

```
tidy(), multcomp::cld(), multcomp::summary.glht(), multcomp::confint.glht(), multcomp::glht()
Other multcomp tidiers: tidy.confint.glht(), tidy.glht(), tidy.summary.glht()
```

```
# load libraries for models and data
library(multcomp)
library(ggplot2)

amod <- aov(breaks ~ wool + tension, data = warpbreaks)
wht <- glht(amod, linfct = mcp(tension = "Tukey"))
tidy(wht)</pre>
```

tidy.clm 247

```
ggplot(wht, aes(lhs, estimate)) +
   geom_point()

CI <- confint(wht)

tidy(CI)

ggplot(CI, aes(lhs, estimate, ymin = lwr, ymax = upr)) +
   geom_pointrange()

tidy(summary(wht))

ggplot(mapping = aes(lhs, estimate)) +
   geom_linerange(aes(ymin = lwr, ymax = upr), data = CI) +
   geom_point(aes(size = p), data = summary(wht)) +
   scale_size(trans = "reverse")

cld <- cld(wht)
tidy(cld)</pre>
```

tidy.clm

Tidy a(n) clm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'clm'
tidy(
    x,
    conf.int = FALSE,
    conf.level = 0.95,
    conf.type = c("profile", "Wald"),
    exponentiate = FALSE,
    ...
)
```

Arguments

x A clm object returned from ordinal::clm().

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

248 tidy.clm

conf.level The confidence level to use for the confidence interval if conf. int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval. conf.type Whether to use "profile" or "Wald" confidence intervals, passed to the type argument of ordinal::confint.clm(). Defaults to "profile".

> Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is

no log or logit link. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

exponentiate

In broom 0.7.0 the coefficient_type column was renamed to coef.type, and the contents were changed as well.

Note that intercept type coefficients correspond to alpha parameters, location type coefficients correspond to beta parameters, and scale type coefficients correspond to zeta parameters.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

```
tidy, ordinal::clm(), ordinal::confint.clm()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.clm(), glance.polr(),
glance.svyolr(), tidy.clmm(), tidy.polr(), tidy.svyolr()
```

tidy.clmm 249

Examples

```
# load libraries for models and data
library(ordinal)

# fit model
fit <- clm(rating ~ temp * contact, data = wine)

# summarize model fit with tidiers
tidy(fit)
tidy(fit, conf.int = TRUE, conf.level = 0.9)
tidy(fit, conf.int = TRUE, conf.type = "Wald", exponentiate = TRUE)

glance(fit)
augment(fit, type.predict = "prob")
augment(fit, type.predict = "class")

# ...and again with another model specification
fit2 <- clm(rating ~ temp, nominal = ~contact, data = wine)

tidy(fit2)
glance(fit2)</pre>
```

tidy.clmm

Tidy a(n) clmm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'clmm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

x	A clmm object returned from ordinal::clmm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

250 tidy.clmm

exponentiate

Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

Note

In broom 0.7.0 the coefficient_type column was renamed to coef.type, and the contents were changed as well.

Note that intercept type coefficients correspond to alpha parameters, location type coefficients correspond to beta parameters, and scale type coefficients correspond to zeta parameters.

See Also

```
tidy, ordinal::clmm(), ordinal::confint.clm()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.clmm(), glance.polr(),
glance.svyolr(), tidy.clm(), tidy.polr(), tidy.svyolr()
```

```
# load libraries for models and data
library(ordinal)

# fit model
fit <- clmm(rating ~ temp + contact + (1 | judge), data = wine)</pre>
```

tidy.coeftest 251

```
# summarize model fit with tidiers
tidy(fit)
tidy(fit, conf.int = TRUE, conf.level = 0.9)
tidy(fit, conf.int = TRUE, exponentiate = TRUE)
glance(fit)
# ...and again with another model specification
fit2 <- clmm(rating ~ temp + (1 | judge), nominal = ~contact, data = wine)
tidy(fit2)
glance(fit2)</pre>
```

tidy.coeftest

Tidy a(n) coeftest object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'coeftest'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x A coeftest object returned from lmtest::coeftest().

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

252 tidy.coeftest

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate. estimate The estimated value of the regression term. p.value The two-sided p-value associated with the observed statistic. The value of a T-statistic to use in a hypothesis that the regression term is nonstatistic zero. std.error

The standard error of the regression term.

The name of the regression term. term

See Also

```
tidy(), lmtest::coeftest()
```

```
# load libraries for models and data
library(lmtest)
m <- lm(dist ~ speed, data = cars)</pre>
coeftest(m)
tidy(coeftest(m))
tidy(coeftest(m, conf.int = TRUE))
# a very common workflow is to combine lmtest::coeftest with alternate
# variance-covariance matrices via the sandwich package. The lmtest
# tidiers support this workflow too, enabling you to adjust the standard
# errors of your tidied models on the fly.
library(sandwich)
# "HC3" (default) robust SEs
tidy(coeftest(m, vcov = vcovHC))
# "HC2" robust SEs
tidy(coeftest(m, vcov = vcovHC, type = "HC2"))
# N-W HAC robust SEs
tidy(coeftest(m, vcov = NeweyWest))
# the columns of the returned tibble for glance.coeftest() will vary
# depending on whether the coeftest object retains the underlying model.
# Users can control this with the "save = TRUE" argument of coeftest().
glance(coeftest(m))
glance(coeftest(m, save = TRUE))
```

tidy.confint.glht 253

tidy.confint.glht Tidy a(n) confint.glht object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'confint.glht'
tidy(x, ...)
```

Arguments

x A confint.glht object created by calling multcomp::confint.glht() on a glht object created with multcomp::glht().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

contrast Levels being compared.

estimate The estimated value of the regression term.

See Also

```
tidy(), multcomp::confint.glht(), multcomp::glht()
Other multcomp tidiers: tidy.cld(), tidy.glht(), tidy.summary.glht()
```

254 tidy.confusionMatrix

Examples

```
# load libraries for models and data
library(multcomp)
library(ggplot2)
amod <- aov(breaks ~ wool + tension, data = warpbreaks)</pre>
wht <- glht(amod, linfct = mcp(tension = "Tukey"))</pre>
tidy(wht)
ggplot(wht, aes(lhs, estimate)) +
  geom_point()
CI <- confint(wht)</pre>
tidy(CI)
ggplot(CI, aes(lhs, estimate, ymin = lwr, ymax = upr)) +
  geom_pointrange()
tidy(summary(wht))
ggplot(mapping = aes(lhs, estimate)) +
  geom_linerange(aes(ymin = lwr, ymax = upr), data = CI) +
  geom_point(aes(size = p), data = summary(wht)) +
  scale_size(trans = "reverse")
cld <- cld(wht)</pre>
tidy(cld)
```

tidy.confusionMatrix Tidy a(n) confusionMatrix object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'confusionMatrix'
tidy(x, by_class = TRUE, ...)
```

255 tidy.confusionMatrix

Arguments

An object of class confusionMatrix created by a call to caret::confusionMatrix().

by_class Logical indicating whether or not to show performance measures broken down

by class. Defaults to TRUE. When by_class = FALSE only returns a tibble with

accuracy, kappa, and McNemar statistics.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

The class under consideration. class

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

term The name of the regression term.

p.value P-value for accuracy and kappa statistics.

See Also

```
tidy(), caret::confusionMatrix()
```

```
# load libraries for models and data
library(caret)
set.seed(27)
# generate data
two_class_sample1 <- as.factor(sample(letters[1:2], 100, TRUE))</pre>
two_class_sample2 <- as.factor(sample(letters[1:2], 100, TRUE))</pre>
two_class_cm <- confusionMatrix(</pre>
 two_class_sample1,
 two_class_sample2
)
# summarize model fit with tidiers
```

256 tidy.coxph

```
tidy(two_class_cm)
tidy(two_class_cm, by_class = FALSE)

# multiclass example
six_class_sample1 <- as.factor(sample(letters[1:6], 100, TRUE))
six_class_sample2 <- as.factor(sample(letters[1:6], 100, TRUE))

six_class_cm <- confusionMatrix(
    six_class_sample1,
    six_class_sample2
)

# summarize model fit with tidiers
tidy(six_class_cm)
tidy(six_class_cm, by_class = FALSE)</pre>
```

tidy.coxph

Tidy a(n) coxph object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'coxph'
tidy(x, exponentiate = FALSE, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

Х	A coxph object returned from survival::coxph().
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if $conf.int = TRUE$. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	For tidy(), additional arguments passed to summary(x ,). Otherwise ignored.

tidy.coxph 257

Value

```
A tibble::tibble() with columns:
```

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

std.error The standard error of the regression term.

See Also

```
tidy(), survival::coxph()
Other coxph tidiers: augment.coxph(), glance.coxph()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.cch(), tidy.pyears(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)
# fit model
cfit <- coxph(Surv(time, status) ~ age + sex, lung)</pre>
# summarize model fit with tidiers
tidy(cfit)
tidy(cfit, exponentiate = TRUE)
lp <- augment(cfit, lung)</pre>
risks <- augment(cfit, lung, type.predict = "risk")
expected <- augment(cfit, lung, type.predict = "expected")</pre>
glance(cfit)
# also works on clogit models
resp <- levels(logan$occupation)</pre>
n <- nrow(logan)</pre>
indx <- rep(1:n, length(resp))</pre>
logan2 <- data.frame(</pre>
  logan[indx, ],
  id = indx,
  tocc = factor(rep(resp, each = n))
)
logan2$case <- (logan2$occupation == logan2$tocc)</pre>
cl <- clogit(case ~ tocc + tocc:education + strata(id), logan2)</pre>
```

258 tidy.crr

```
tidy(cl)
glance(cl)

library(ggplot2)

ggplot(lp, aes(age, .fitted, color = sex)) +
    geom_point()

ggplot(risks, aes(age, .fitted, color = sex)) +
    geom_point()

ggplot(expected, aes(time, .fitted, color = sex)) +
    geom_point()
```

tidy.crr

Tidy a(n) cmprsk object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'crr'
tidy(x, exponentiate = FALSE, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	A crr object returned from cmprsk::crr().
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if $conf.int = TRUE$. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.crr 259

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.

See Also

```
tidy(), cmprsk::crr()
Other cmprsk tidiers: glance.crr()
```

```
library(cmprsk)

# time to loco-regional failure (lrf)
lrf_time <- rexp(100)
lrf_event <- sample(0:2, 100, replace = TRUE)
trt <- sample(0:1, 100, replace = TRUE)
strt <- sample(1:2, 100, replace = TRUE)

# fit model
x <- crr(lrf_time, lrf_event, cbind(trt, strt))

# summarize model fit with tidiers
tidy(x, conf.int = TRUE)
glance(x)</pre>
```

260 tidy.cv.glmnet

tidy.cv.glmnet	Tidy a(n) cv.glmnet object	

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'cv.glmnet' tidy(x, ...)
```

Arguments

x A cv.glmnet object returned from glmnet::cv.glmnet().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

lambda

A tibble::tibble() with columns:

	1 7 1
nzero	Number of non-zero coefficients for the given lambda.
std.error	The standard error of the regression term.
conf.low	lower bound on confidence interval for cross-validation estimated loss.
conf.high	upper bound on confidence interval for cross-validation estimated loss.
estimate	Median loss across all cross-validation folds for a given lamdba

Value of penalty parameter lambda.

See Also

```
tidy(), glmnet::cv.glmnet()
Other glmnet tidiers: glance.cv.glmnet(), glance.glmnet(), tidy.glmnet()
```

tidy.cv.glmnet 261

```
# load libraries for models and data
library(glmnet)
set.seed(27)
nobs <- 100
nvar <- 50
real <- 5
x <- matrix(rnorm(nobs * nvar), nobs, nvar)</pre>
beta <- c(rnorm(real, 0, 1), rep(0, nvar - real))
y \leftarrow c(t(beta) %*% t(x)) + rnorm(nvar, sd = 3)
cvfit1 <- cv.glmnet(x, y)</pre>
tidy(cvfit1)
glance(cvfit1)
library(ggplot2)
tidied_cv <- tidy(cvfit1)</pre>
glance_cv <- glance(cvfit1)</pre>
# plot of MSE as a function of lambda
g <- ggplot(tidied_cv, aes(lambda, estimate)) +</pre>
  geom_line() +
  scale_x_log10()
# plot of MSE as a function of lambda with confidence ribbon
g <- g + geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)
# plot of MSE as a function of lambda with confidence ribbon and choices
# of minimum lambda marked
  geom_vline(xintercept = glance_cv$lambda.min) +
  geom_vline(xintercept = glance_cv$lambda.1se, lty = 2)
# plot of number of zeros for each choice of lambda
ggplot(tidied_cv, aes(lambda, nzero)) +
  geom_line() +
  scale_x_log10()
# coefficient plot with min lambda shown
tidied <- tidy(cvfit1$glmnet.fit)</pre>
ggplot(tidied, aes(lambda, estimate, group = term)) +
  scale_x_log10() +
```

262 tidy.density

```
geom_line() +
geom_vline(xintercept = glance_cv$lambda.min) +
geom_vline(xintercept = glance_cv$lambda.1se, lty = 2)
```

tidy.density

(Deprecated) Tidy density objects

Description

(Deprecated) Tidy density objects

Usage

```
## S3 method for class 'density' tidy(x, ...)
```

Arguments

. . .

x A density object returned from stats::density().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble with two columns: points x where the density is estimated, and estimated density y.

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.dist(), tidy.ftable(), tidy.numeric()
```

tidy.dist 263

tidy.dist

(Deprecated) Tidy dist objects

Description

(Deprecated) Tidy dist objects

Usage

```
## S3 method for class 'dist'
tidy(x, diagonal = attr(x, "Diag"), upper = attr(x, "Upper"), ...)
```

Arguments

x A dist object returned from stats::dist().

diagonal Logical indicating whether or not to tidy the diagonal elements of the distance

matrix. Defaults to whatever was based to the diag argument of stats::dist().

upper Logical indicating whether or not to tidy the upper half of the distance matrix.

Defaults to whatever was based to the upper argument of stats::dist().

. . . Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be

ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

tidy() methods will warn when supplied an exponentiate argument if it
will be ignored.

• augment() methods will warn when supplied a newdata argument if it will

be ignored.

Details

If the distance matrix does not include an upper triangle and/or diagonal, the tidied version will not either.

Value

A tibble::tibble with one row for each pair of items in the distance matrix, with columns:

item1 First item item2 Second item

distance Distance between items

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.ftable(), tidy.numeric()
```

264 tidy.drc

Examples

```
cars_dist <- dist(t(mtcars[, 1:4]))
cars_dist

tidy(cars_dist)
tidy(cars_dist, upper = TRUE)
tidy(cars_dist, diagonal = TRUE)</pre>
```

tidy.drc

Tidy a(n) drc object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'drc'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

Χ

A drc object produced by a call to drc::drm().

conf.int

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

conf.level

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The tibble has one row for each curve and term in the regression. The curveid column indicates the curve.

tidy.emmGrid 265

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

std.error The standard error of the regression term.

The name of the regression term.

Index identifying the course

curve Index identifying the curve.

See Also

```
tidy(), drc::drm()
Other drc tidiers: augment.drc(), glance.drc()
```

Examples

```
# load libraries for models and data
library(drc)

# fit model
mod <- drm(dead / total ~ conc, type,
   weights = total, data = selenium, fct = LL.2(), type = "binomial"
)

# summarize model fit with tidiers
tidy(mod)
tidy(mod, conf.int = TRUE)
glance(mod)
augment(mod, selenium)</pre>
```

tidy.emmGrid

Tidy a(n) emmGrid object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

266 tidy.emmGrid

Usage

```
## S3 method for class 'emmGrid'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	An emmGrid object.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid(). Cautionary note: misspecified arguments may be silently ignored!

Details

Returns a data frame with one observation for each estimated marginal mean, and one column for each combination of factors. When the input is a contrast, each row will contain one estimated contrast.

There are a large number of arguments that can be passed on to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid().

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
df	Degrees of freedom used by this term in the model.
p.value	The two-sided p-value associated with the observed statistic.
std.error	The standard error of the regression term.
estimate	Expected marginal mean
statistic	T-ratio statistic

See Also

```
tidy(), emmeans::ref_grid(), emmeans::emmeans(), emmeans::contrast()
Other emmeans tidiers: tidy.lsmobj(), tidy.ref.grid(), tidy.summary_emm()
```

tidy.epi.2by2

```
# load libraries for models and data
library(emmeans)
# linear model for sales of oranges per day
oranges_lm1 <- lm(sales1 ~ price1 + price2 + day + store, data = oranges)</pre>
# reference grid; see vignette("basics", package = "emmeans")
oranges_rg1 <- ref_grid(oranges_lm1)</pre>
td <- tidy(oranges_rg1)</pre>
td
# marginal averages
marginal <- emmeans(oranges_rg1, "day")</pre>
tidy(marginal)
# contrasts
tidy(contrast(marginal))
tidy(contrast(marginal, method = "pairwise"))
# plot confidence intervals
library(ggplot2)
ggplot(tidy(marginal, conf.int = TRUE), aes(day, estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# by multiple prices
by_price <- emmeans(oranges_lm1, "day",</pre>
 by = "price2",
  at = list(
    price1 = 50, price2 = c(40, 60, 80),
    day = c("2", "3", "4")
  )
)
by_price
tidy(by_price)
ggplot(tidy(by_price, conf.int = TRUE), aes(price2, estimate, color = day)) +
  geom_line() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# joint_tests
tidy(joint_tests(oranges_lm1))
```

268 tidy.epi.2by2

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'epi.2by2'
tidy(x, parameters = c("moa", "stat"), ...)
```

Arguments

A epi.2by2 object produced by a call to epiR::epi.2by2()

Return measures of association (moa) or test statistics (stat), default is moa (measures of association)

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The tibble has a column for each of the measures of association or tests contained in massoc or massoc.detail when epiR::epi.2by2() is called.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

Lower bound on the confidence interval for the estimate.

Degrees of freedom used by this term in the model.

p.value The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

The name of the regression term.

Estimated measure of association

See Also

estimate

```
tidy(), epiR::epi.2by2()
```

tidy.ergm 269

Examples

```
# load libraries for models and data
library(epiR)

# generate data
dat <- matrix(c(13, 2163, 5, 3349), nrow = 2, byrow = TRUE)

rownames(dat) <- c("DF+", "DF-")
colnames(dat) <- c("FUS+", "FUS-")

# fit model
fit <- epi.2by2(
   dat = as.table(dat), method = "cross.sectional",
   conf.level = 0.95, units = 100, outcome = "as.columns"
)

# summarize model fit with tidiers
tidy(fit, parameters = "moa")
tidy(fit, parameters = "stat")</pre>
```

tidy.ergm

Tidy a(n) ergm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

The methods should work with any model that conforms to the **ergm** class, such as those produced from weighted networks by the **ergm.count** package.

Usage

```
## S3 method for class 'ergm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

X	An ergm object returned from a call to ergm::ergm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

270 tidy.ergm

Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
 Additional arguments to pass to ergm::summary(). Cautionary note: Misspecified arguments may be silently ignored.

Value

A tibble::tibble with one row for each coefficient in the exponential random graph model, with columns:

term The term in the model being estimated and tested

estimate The estimated coefficient
std.error The standard error
mcmc.error The MCMC error
p.value The two-sided p-value

References

Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris M (2008b). **ergm**: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks. *Journal of Statistical Software*, 24(3). https://www.jstatsoft.org/v24/i03/.

See Also

```
tidy(), ergm::ergm(), ergm::control.ergm(), ergm::summary()
Other ergm tidiers: glance.ergm()
```

```
# load libraries for models and data
library(ergm)

# load the Florentine marriage network data
data(florentine)

# fit a model where the propensity to form ties between
# families depends on the absolute difference in wealth
gest <- ergm(flomarriage ~ edges + absdiff("wealth"))

# show terms, coefficient estimates and errors
tidy(gest)

# show coefficients as odds ratios with a 99% CI
tidy(gest, exponentiate = TRUE, conf.int = TRUE, conf.level = 0.99)

# take a look at likelihood measures and other
# control parameters used during MCMC estimation</pre>
```

tidy.factanal 271

```
glance(gest)
glance(gest, deviance = TRUE)
glance(gest, mcmc = TRUE)
```

tidy.factanal

 $Tidy \ a(n) \ factanal \ object$

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'factanal' tidy(x, ...)
```

Arguments

x A factanal object created by stats::factanal().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

variable Variable under consideration.

uniqueness Proportion of residual, or unexplained variance

f1X Factor loading for level X.

See Also

```
tidy(), stats::factanal()
Other factanal tidiers: augment.factanal(), glance.factanal()
```

272 tidy.felm

Examples

```
set.seed(123)
# generate data
library(dplyr)
library(purrr)
m1 <- tibble(
  v1 = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 4, 5, 6),
  v2 = c(1, 2, 1, 1, 1, 1, 2, 1, 2, 1, 3, 4, 3, 3, 3, 4, 6, 5),
  v3 = c(3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 4, 6),
  v4 = c(3, 3, 4, 3, 3, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 5, 6, 4),
  v5 = c(1, 1, 1, 1, 1, 3, 3, 3, 3, 1, 1, 1, 1, 1, 6, 4, 5),
  v6 = c(1, 1, 1, 2, 1, 3, 3, 3, 4, 3, 1, 1, 1, 2, 1, 6, 5, 4)
)
# new data
m2 <- map_dfr(m1, rev)</pre>
# factor analysis objects
fit1 <- factanal(m1, factors = 3, scores = "Bartlett")</pre>
fit2 <- factanal(m1, factors = 3, scores = "regression")</pre>
# tidying the object
tidy(fit1)
tidy(fit2)
# augmented dataframe
augment(fit1)
augment(fit2)
# augmented dataframe (with new data)
augment(fit1, data = m2)
augment(fit2, data = m2)
```

tidy.felm

Tidy a(n) felm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'felm'
```

tidy.felm 273

```
tidy(
   x,
   conf.int = FALSE,
   conf.level = 0.95,
   fe = FALSE,
   se.type = c("default", "iid", "robust", "cluster"),
   ...
)
```

Arguments

x A felm object returned from lfe::felm().

conf.int Logical indicating whether or not to include a confidence interval in the tidied

output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must

be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

fe Logical indicating whether or not to include estimates of fixed effects. Defaults

to FALSE.

se. type Character indicating the type of standard errors. Defaults to using those of the

underlying felm() model object, e.g. clustered errors for models that were provided a cluster specification. Users can override these defaults by specifying an appropriate alternative: "iid" (for homoskedastic errors), "robust" (for Eicker-Huber-White robust errors), or "cluster" (for clustered standard errors; if the

model object supports it).

.. Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be

ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it

will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf. low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p. value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

std.error The standard error of the regression term.

term The name of the regression term.

274 tidy.fitdistr

See Also

```
tidy(), lfe::felm()
Other felm tidiers: augment.felm()
```

```
# load libraries for models and data
library(lfe)
# use built-in `airquality` dataset
head(airquality)
# no FEs; same as lm()
est0 <- felm(Ozone ~ Temp + Wind + Solar.R, airquality)</pre>
# summarize model fit with tidiers
tidy(est0)
augment(est0)
# add month fixed effects
est1 <- felm(Ozone ~ Temp + Wind + Solar.R | Month, airquality)</pre>
# summarize model fit with tidiers
tidy(est1)
tidy(est1, fe = TRUE)
augment(est1)
glance(est1)
# the "se.type" argument can be used to switch out different standard errors
# types on the fly. In turn, this can be useful exploring the effect of
# different error structures on model inference.
tidy(est1, se.type = "iid")
tidy(est1, se.type = "robust")
# add clustered SEs (also by month)
est2 <- felm(Ozone ~ Temp + Wind + Solar.R | Month | 0 | Month, airquality)</pre>
# summarize model fit with tidiers
tidy(est2, conf.int = TRUE)
tidy(est2, conf.int = TRUE, se.type = "cluster")
tidy(est2, conf.int = TRUE, se.type = "robust")
tidy(est2, conf.int = TRUE, se.type = "iid")
```

tidy.fitdistr 275

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'fitdistr'
tidy(x, ...)
```

Arguments

x A fitdistr object returned by MASS::fitdistr().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

estimate The estimated value of the regression term.

std.error The standard error of the regression term.

term The name of the regression term.

See Also

```
tidy(), MASS::fitdistr()
Other fitdistr tidiers: glance.fitdistr()
```

```
# load libraries for models and data
library(MASS)

# generate data
set.seed(2015)
x <- rnorm(100, 5, 2)

# fit models</pre>
```

276 tidy.fixest

```
fit <- fitdistr(x, dnorm, list(mean = 3, sd = 1))
# summarize model fit with tidiers
tidy(fit)
glance(fit)</pre>
```

tidy.fixest

Tidy a(n) fixest object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'fixest'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A fixest object returned from any of the fixest estimators
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to summary and confint. Important arguments are se and cluster. Other arguments are dof, exact_dof, forceCovariance, and keepBounded. See summary.fixest.

Details

The fixest package provides a family of functions for estimating models with arbitrary numbers of fixed-effects, in both an OLS and a GLM context. The package also supports robust (i.e. White) and clustered standard error reporting via the generic summary.fixest() command. In a similar vein, the tidy() method for these models allows users to specify a desired standard error correction either 1) implicitly via the supplied fixest object, or 2) explicitly as part of the tidy call. See examples below.

Note that fixest confidence intervals are calculated assuming a normal distribution – this assumes infinite degrees of freedom for the CI. (This assumption is distinct from the degrees of freedom used to calculate the standard errors. For more on degrees of freedom with clusters and fixed effects, see https://github.com/lrberge/fixest/issues/6 and https://github.com/sgaure/lfe/issues/1#issuecomment-530646990)

tidy.fixest 277

Value

```
A tibble::tibble() with columns:
```

```
conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

std.error The standard error of the regression term.

The name of the regression term.
```

See Also

```
tidy(), fixest::feglm(), fixest::fenegbin(), fixest::feNmlm(), fixest::femlm(), fixest::feols(),
fixest::fepois()
Other fixest tidiers: augment.fixest()
```

```
# load libraries for models and data
library(fixest)
gravity <-
 feols(
   log(Euros) ~ log(dist_km) | Origin + Destination + Product + Year, trade
tidy(gravity)
glance(gravity)
augment(gravity, trade)
# to get robust or clustered SEs, users can either:
# 1) specify the arguments directly in the `tidy()` call
tidy(gravity, conf.int = TRUE, cluster = c("Product", "Year"))
tidy(gravity, conf.int = TRUE, se = "threeway")
# 2) or, feed tidy() a summary.fixest object that has already accepted
# these arguments
gravity_summ <- summary(gravity, cluster = c("Product", "Year"))</pre>
tidy(gravity_summ, conf.int = TRUE)
# approach (1) is preferred.
```

278 tidy.Gam

tidy.ftable

(Deprecated) Tidy ftable objects

Description

This function is deprecated. Please use tibble::as_tibble() instead.

Usage

```
## S3 method for class 'ftable' tidy(x, ...)
```

Arguments

Х

An ftable object returned from stats::ftable().

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

An ftable contains a "flat" contingency table. This melts it into a tibble::tibble with one column for each variable, then a Freq column.

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.numeric()
```

tidy.Gam

Tidy a(n) Gam object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

tidy.Gam 279

Usage

```
## S3 method for class 'Gam'
tidy(x, ...)
```

Arguments

x A Gam object returned from a call to gam::gam().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

Tidy gam objects created by calls to mgcv::gam() with tidy.gam().

Value

```
A tibble::tibble() with columns:
```

df Degrees of freedom used by this term in the model.

meansq Mean sum of squares. Equal to total sum of squares divided by degrees of

freedom.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

sumsq Sum of squares explained by this term.

term The name of the regression term.

See Also

```
tidy(), gam::gam(), tidy.anova(), tidy.gam()
Other gam tidiers: glance.Gam()
```

```
# load libraries for models and data
library(gam)

# fit model
g <- gam(mpg ~ s(hp, 4) + am + qsec, data = mtcars)</pre>
```

280 tidy.gam

```
# summarize model fit with tidiers
tidy(g)
glance(g)
```

 ${\tt tidy.gam}$

Tidy a(n) gam object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'gam'
tidy(
    x,
    parametric = FALSE,
    conf.int = FALSE,
    conf.level = 0.95,
    exponentiate = FALSE,
    ...
)
```

Arguments

X	A gam object returned from a call to mgcv::gam().
parametric	Logical indicating if parametric or smooth terms should be tidied. Defaults to FALSE, meaning that smooth terms are tidied by default.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

tidy.gam 281

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

When parametric = FALSE return columns edf and ref.df rather than estimate and std.error.

Value

A tibble::tibble() with columns:

The estimated value of the regression term.

p.value
The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

std.error
The standard error of the regression term.

term
The name of the regression term.

edf
The effective degrees of freedom. Only reported when 'parametric = FALSE'

ref.df
The reference degrees of freedom. Only reported when 'parametric = FALSE'

See Also

```
tidy(), mgcv::gam()
Other mgcv tidiers: glance.gam()
```

```
# load libraries for models and data
library(mgcv)

# fit model
g <- gam(mpg ~ s(hp) + am + qsec, data = mtcars)

# summarize model fit with tidiers
tidy(g)
tidy(g, parametric = TRUE)
glance(g)
augment(g)</pre>
```

282 tidy.garch

tidy.garch $Tidy \ a(n) \ garch \ object$	idy.garch	Tidy a(n) garch object	
-------------------------------------------	-----------	------------------------	--

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'garch'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A garch object returned by tseries::garch().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-
	zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

tidy.geeglm 283

See Also

```
tidy(), tseries::garch()
Other garch tidiers: glance.garch()
```

Examples

```
# load libraries for models and data
library(tseries)

# load data
data(EuStockMarkets)

# fit model
dax <- diff(log(EuStockMarkets))[, "DAX"]
dax.garch <- garch(dax)
dax.garch

# summarize model fit with tidiers
tidy(dax.garch)
glance(dax.garch)</pre>
```

tidy.geeglm

Tidy a(n) geeglm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'geeglm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

x	A geeglm object returned from a call to geepack::geeglm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

284 tidy.geeglm

exponentiate

Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

If conf.int = TRUE, the confidence interval is computed with the an internal confint.geeglm() function.

If you have missing values in your model data, you may need to refit the model with na.action = na.exclude or deal with the missingness in the data beforehand.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), geepack::geeglm()
```

```
# load modeling library
library(geepack)
# load data
data(state)
```

tidy.glht 285

```
ds <- data.frame(state.region, state.x77)
# fit model
geefit <- geeglm(Income ~ Frost + Murder,
   id = state.region,
   data = ds,
   corstr = "exchangeable"
)
# summarize model fit with tidiers
tidy(geefit)
tidy(geefit, conf.int = TRUE)</pre>
```

tidy.glht

Tidy a(n) glht object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'glht'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

Χ	A glht object returned by multcomp::glht().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to summary() and tidy.summary.glht().

Value

A tibble::tibble() with columns:

contrast Levels being compared.

estimate The estimated value of the regression term.

null.value Value to which the estimate is compared.

286 tidy.glm

See Also

```
tidy(), multcomp::glht()
Other multcomp tidiers: tidy.cld(), tidy.confint.glht(), tidy.summary.glht()
```

Examples

```
# load libraries for models and data
library(multcomp)
library(ggplot2)
amod <- aov(breaks ~ wool + tension, data = warpbreaks)</pre>
wht <- glht(amod, linfct = mcp(tension = "Tukey"))</pre>
tidy(wht)
ggplot(wht, aes(lhs, estimate)) +
 geom_point()
CI <- confint(wht)
tidy(CI)
ggplot(CI, aes(lhs, estimate, ymin = lwr, ymax = upr)) +
 geom_pointrange()
tidy(summary(wht))
ggplot(mapping = aes(lhs, estimate)) +
 geom_linerange(aes(ymin = lwr, ymax = upr), data = CI) +
 geom_point(aes(size = p), data = summary(wht)) +
 scale_size(trans = "reverse")
cld <- cld(wht)</pre>
tidy(cld)
```

tidy.glm

Tidy a(n) glm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'glm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

tidy.glmnet 287

Arguments

X	A glm object returned from stats::glm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:
	• tidy() methods will warn when supplied an avnoyantiate argument if it

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

See Also

```
stats::glm()
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(),
glance.svyglm(), tidy.lm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

tidy.glmnet

Tidy a(n) glmnet object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'glmnet'
tidy(x, return_zeros = FALSE, ...)
```

288 tidy.glmnet

Arguments

x A glmnet object returned from glmnet::glmnet().

in the results. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

Note that while this representation of GLMs is much easier to plot and combine than the default structure, it is also much more memory-intensive. Do not use for large, sparse matrices.

No augment method is yet provided even though the model produces predictions, because the input data is not tidy (it is a matrix that may be very wide) and therefore combining predictions with it is not logical. Furthermore, predictions make sense only with a specific choice of lambda.

Value

```
A tibble::tibble() with columns:
```

dev.ratio Fraction of null deviance explained at each value of lambda.

estimate The estimated value of the regression term.

lambda Value of penalty parameter lambda. step Which step of lambda choices was used.

term The name of the regression term.

See Also

```
tidy(), glmnet::glmnet()
Other glmnet tidiers: glance.cv.glmnet(), glance.glmnet(), tidy.cv.glmnet()
```

```
# load libraries for models and data
library(glmnet)

set.seed(2014)
x <- matrix(rnorm(100 * 20), 100, 20)
y <- rnorm(100)
fit1 <- glmnet(x, y)</pre>
```

tidy.glmRob 289

```
# summarize model fit with tidiers + visualization
tidy(fit1)
glance(fit1)
library(dplyr)
library(ggplot2)
tidied <- tidy(fit1) %>% filter(term != "(Intercept)")
ggplot(tidied, aes(step, estimate, group = term)) +
 geom_line()
ggplot(tidied, aes(lambda, estimate, group = term)) +
 geom_line() +
 scale_x_log10()
ggplot(tidied, aes(lambda, dev.ratio)) +
 geom_line()
# works for other types of regressions as well, such as logistic
g2 <- sample(1:2, 100, replace = TRUE)
fit2 <- glmnet(x, g2, family = "binomial")</pre>
tidy(fit2)
```

tidy.glmRob

Tidy a(n) glmRob object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'glmRob'
tidy(x, ...)
```

Arguments

. . .

x A glmRob object returned from robust::glmRob().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

290 tidy.glmrob

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

See Also

```
robust::glmRob()
Other robust tidiers: augment.lmRob(), glance.glmRob(), glance.lmRob(), tidy.lmRob()
```

Examples

```
# load libraries for models and data
library(robust)

# fit model
gm <- glmRob(am ~ wt, data = mtcars, family = "binomial")

# summarize model fit with tidiers
tidy(gm)
glance(gm)</pre>
```

tidy.glmrob

Tidy a(n) glmrob object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'glmrob'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

tidy.glmrob 291

Arguments

A glmrob object returned from robustbase::glmrob().

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

Value

```
A tibble::tibble() with columns:
```

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
robustbase::glmrob()
Other robustbase tidiers: augment.glmrob(), augment.lmrob(), glance.lmrob(), tidy.lmrob()
```

```
if (requireNamespace("robustbase", quietly = TRUE)) {
  # load libraries for models and data
  library(robustbase)

data(coleman)
```

292 tidy.gmm

```
set.seed(0)

m <- lmrob(Y ~ ., data = coleman)
tidy(m)
augment(m)
glance(m)

data(carrots)

Rfit <- glmrob(cbind(success, total - success) ~ logdose + block,
  family = binomial, data = carrots, method = "Mqle",
  control = glmrobMqle.control(tcc = 1.2)
)

tidy(Rfit)
augment(Rfit)
}</pre>
```

tidy.gmm

Tidy a(n) gmm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'gmm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

х	A gmm object returned from gmm::gmm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to exponentiate the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.gmm 293

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

```
conf.high Upper bound on the confidence interval for the estimate.

Lower bound on the confidence interval for the estimate.

The estimated value of the regression term.

The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

The standard error of the regression term.

The name of the regression term.
```

See Also

```
tidy(), gmm::gmm()
Other gmm tidiers: glance.gmm()
```

```
# load libraries for models and data
library(gmm)
# examples come from the "gmm" package
# CAPM test with GMM
data(Finance)
r <- Finance[1:300, 1:10]
rm <- Finance[1:300, "rm"]
rf <- Finance[1:300, "rf"]
z <- as.matrix(r - rf)</pre>
t <- nrow(z)
zm <- rm - rf
h <- matrix(zm, t, 1)</pre>
res <- gmm(z \sim zm, x = h)
# tidy result
tidy(res)
tidy(res, conf.int = TRUE)
tidy(res, conf.int = TRUE, conf.level = .99)
# coefficient plot
library(ggplot2)
```

294 tidy.htest

```
library(dplyr)
tidy(res, conf.int = TRUE) %>%
  mutate(variable = reorder(term, estimate)) %>%
  ggplot(aes(estimate, variable)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, color = "red", lty = 2)
# from a function instead of a matrix
g <- function(theta, x) {</pre>
  e \leftarrow x[, 2:11] - theta[1] - (x[, 1] - theta[1]) %*% matrix(theta[2:11], 1, 10)
  gmat \leftarrow cbind(e, e * c(x[, 1]))
  return(gmat)
}
x <- as.matrix(cbind(rm, r))</pre>
res_black <- gmm(g, x = x, t0 = rep(0, 11))
tidy(res_black)
tidy(res_black, conf.int = TRUE)
# APT test with Fama-French factors and GMM
f1 <- zm
f2 <- Finance[1:300, "hml"] - rf</pre>
f3 <- Finance[1:300, "smb"] - rf
h <- cbind(f1, f2, f3)
res2 <- gmm(z \sim f1 + f2 + f3, x = h)
td2 <- tidy(res2, conf.int = TRUE)
td2
# coefficient plot
td2 %>%
  mutate(variable = reorder(term, estimate)) %>%
  ggplot(aes(estimate, variable)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  geom_vline(xintercept = 0, color = "red", lty = 2)
```

tidy.htest

Tidy/glance a(n) htest object

Description

For models that have only a single component, the tidy() and glance() methods are identical. Please see the documentation for both of those methods.

tidy.htest 295

Usage

```
## S3 method for class 'htest'
tidy(x, ...)
## S3 method for class 'htest'
glance(x, ...)
```

Arguments

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

alternative	Alternative hypothesis (character).
conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
estimate1	Sometimes two estimates are computed, such as in a two-sample t-test.
estimate2	Sometimes two estimates are computed, such as in a two-sample t-test.
method	Method used.
p.value	The two-sided p-value associated with the observed statistic.
parameter	The parameter being modeled.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

See Also

```
tidy(), stats::cor.test(), stats::t.test(), stats::wilcox.test(), stats::chisq.test()
Other htest tidiers: augment.htest(), tidy.pairwise.htest(), tidy.power.htest()
```

296 tidy.ivreg

Examples

```
tt <- t.test(rnorm(10))
tidy(tt)

# the glance output will be the same for each of the below tests
glance(tt)

tt <- t.test(mpg ~ am, data = mtcars)
tidy(tt)

wt <- wilcox.test(mpg ~ am, data = mtcars, conf.int = TRUE, exact = FALSE)
tidy(wt)

ct <- cor.test(mtcars$wt, mtcars$mpg)
tidy(ct)
chit <- chisq.test(xtabs(Freq ~ Sex + Class, data = as.data.frame(Titanic)))
tidy(chit)
augment(chit)</pre>
```

tidy.ivreg

Tidy a(n) ivreg object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'ivreg'
tidy(x, conf.int = FALSE, conf.level = 0.95, instruments = FALSE, ...)
```

Arguments

X	An ivreg object created by a call to AER::ivreg().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

tidy.ivreg 297

instruments Logical indicating whether to return coefficients from the second-stage or diag-

nostics tests for each endogenous regressor (F-statistics). Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This tidier currently only supports ivreg-classed objects outputted by the AER package. The ivreg package also outputs objects of class ivreg, and will be supported in a later release.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p. value The two-sided p-value associated with the observed statistic.

p.value.Sargan p-value for Sargan test of overidentifying restrictions.

p.value.weakinst

p-value for weak instruments test.

p.value.Wu.Hausman

p-value for Wu-Hausman weak instruments test for endogeneity.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

statistic.Sargan

Statistic for Sargan test of overidentifying restrictions.

statistic.weakinst

Statistic for Wu-Hausman test.

statistic.Wu.Hausman

Statistic for Wu-Hausman weak instruments test for endogeneity.

std.error The standard error of the regression term.

term The name of the regression term.

See Also

```
tidy(), AER::ivreg()
```

Other ivreg tidiers: augment.ivreg(), glance.ivreg()

298 tidy.kappa

Examples

```
# load libraries for models and data
library(AER)
# load data
data("CigarettesSW", package = "AER")
# fit model
ivr <- ivreg(</pre>
 log(packs) ~ income | population,
 data = CigarettesSW,
 subset = year == "1995"
)
# summarize model fit with tidiers
tidy(ivr, conf.int = TRUE)
tidy(ivr, conf.int = TRUE, instruments = TRUE)
augment(ivr)
augment(ivr, data = CigarettesSW)
augment(ivr, newdata = CigarettesSW)
glance(ivr)
```

tidy.kappa

Tidy a(n) kappa object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'kappa' tidy(x, ...)
```

Arguments

x A kappa object returned from psych::cohen.kappa().

... Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be

tidy.kappa 299

used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

Note that confidence level (alpha) for the confidence interval cannot be set in tidy. Instead you must set the alpha argument to psych::cohen.kappa() when creating the kappa object.

Value

```
A tibble::tibble() with columns:
```

```
conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

type Either 'weighted' or 'unweighted'.
```

See Also

```
tidy(), psych::cohen.kappa()
```

```
# load libraries for models and data
library(psych)

# generate example data
rater1 <- 1:9
rater2 <- c(1, 3, 1, 6, 1, 5, 5, 6, 7)

# fit model
ck <- cohen.kappa(cbind(rater1, rater2))

# summarize model fit with tidiers + visualization
tidy(ck)

# graph the confidence intervals
library(ggplot2)

ggplot(tidy(ck), aes(estimate, type)) +
    geom_point() +
    geom_errorbarh(aes(xmin = conf.low, xmax = conf.high))</pre>
```

300 tidy.kde

tidy.kde	Tidy a(n) kde object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'kde' tidy(x, ...)
```

Arguments

x A kde object returned from ks::kde().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

Returns a data frame in long format with four columns. Use tidyr::pivot_wider(..., names_from = variable, values_from = value) on the output to return to a wide format.

Value

A tibble::tibble() with columns:

estimate The estimated value of the regression term.

obs weighted observed number of events in each group.

value The value/estimate of the component. Results from data reshaping.

variable Variable under consideration.

See Also

```
tidy(), ks::kde()
```

tidy.Kendall 301

Examples

```
# load libraries for models and data
library(ks)
# generate data
dat <- replicate(2, rnorm(100))</pre>
k <- kde(dat)
# summarize model fit with tidiers + visualization
td <- tidy(k)
td
library(ggplot2)
library(dplyr)
library(tidyr)
td %>%
  pivot_wider(c(obs, estimate),
    names_from = variable,
    values_from = value
  ggplot(aes(x1, x2, fill = estimate)) +
  geom_tile() +
  theme_void()
# also works with 3 dimensions
dat3 <- replicate(3, rnorm(100))</pre>
k3 <- kde(dat3)
td3 \leftarrow tidy(k3)
td3
```

tidy.Kendall

Tidy a(n) Kendall object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'Kendall' tidy(x, ...)
```

302 tidy.Kendall

Arguments

. . .

x A Kendall object returned from a call to Kendall::Kendall(), Kendall::MannKendall(), or Kendall::SeasonalMannKendall().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

kendall_score Kendall score.

p.value The two-sided p-value associated with the observed statistic.

var_kendall_score

Variance of the kendall_score.

statistic Kendall's tau statistic

denominator The denominator, which is tau=kendall_score/denominator.

See Also

```
tidy(), Kendall::Kendall(), Kendall::MannKendall(), Kendall::SeasonalMannKendall()
```

```
# load libraries for models and data
library(Kendall)

A <- c(2.5, 2.5, 2.5, 2.5, 5, 6.5, 6.5, 10, 10, 10, 10, 10, 14, 14, 14, 16, 17)
B <- c(1, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2)

# fit models and summarize results
f_res <- Kendall(A, B)
tidy(f_res)

s_res <- MannKendall(B)
tidy(s_res)

t_res <- SeasonalMannKendall(ts(A))
tidy(t_res)</pre>
```

tidy.kmeans 303

tidy.kmeans

Tidy a(n) kmeans object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'kmeans'
tidy(x, col.names = colnames(x$centers), ...)
```

Arguments

x A kmeans object created by stats::kmeans().

col.names Dimension names. Defaults to the names of the variables in x. Set to NULL to

get names x1, x2,

... Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

cluster A factor describing the cluster from 1:k.
size Number of points assigned to cluster.
withinss The within-cluster sum of squares.

See Also

```
tidy(), stats::kmeans()
Other kmeans tidiers: augment.kmeans(), glance.kmeans()
```

304 tidy.lavaan

Examples

```
library(cluster)
library(modeldata)
library(dplyr)

data(hpc_data)

x <- hpc_data[, 2:5]

fit <- pam(x, k = 4)

tidy(fit)
glance(fit)
augment(fit, x)</pre>
```

tidy.lavaan

Tidy a(n) lavaan object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'lavaan'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A lavaan object, such as those returned from lavaan::cfa(), and lavaan::sem().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to lavaan::parameterEstimates(). Cautionary note: Misspecified arguments may be silently ignored.

tidy.lavaan 305

Value

A tibble::tibble() with one row for each estimated parameter and columns:

term	The result of paste(lhs, op, rhs)
ор	The operator in the model syntax (e.g. \sim for covariances, or \sim for regression parameters)
group	The group (if specified) in the lavaan model
estimate	The parameter estimate (may be standardized)
std.error	
statistic	The z value returned by lavaan::parameterEstimates()
p.value	
conf.low	
conf.high	
std.lv	Standardized estimates based on the variances of the (continuous) latent variables only
std.all	Standardized estimates based on both the variances of both (continuous) observed and latent variables.
std.nox	Standardized estimates based on both the variances of both (continuous) observed and latent variables, but not the variances of exogenous covariates.

See Also

```
tidy(), lavaan::cfa(), lavaan::sem(), lavaan::parameterEstimates()
Other lavaan tidiers: glance.lavaan()
```

```
# load libraries for models and data
library(lavaan)

cfa.fit <- cfa("F =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9",
    data = HolzingerSwineford1939, group = "school"
)

tidy(cfa.fit)</pre>
```

306 tidy.lm

tidy.lm

Tidy a(n) lm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'lm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

An lm object created by stats::lm().

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

exponentiate Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.

... Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

If the linear model is an mlm object (multiple linear model), there is an additional column response. See tidy.mlm().

Value

```
A tibble::tibble() with columns:
```

conf.high Upper bound on the confidence interval for the estimate.

tidy.lm 307

conf.low Lower bound on the confidence interval for the estimate.

The estimated value of the regression term.

The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

Std.error The standard error of the regression term.

The name of the regression term.

See Also

```
tidy(), stats::summary.lm()
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(),
glance.svyglm(), tidy.glm(), tidy.lm.beta(), tidy.mlm(), tidy.summary.lm()
```

```
library(ggplot2)
library(dplyr)
mod <- lm(mpg ~ wt + qsec, data = mtcars)</pre>
tidy(mod)
glance(mod)
# coefficient plot
d <- tidy(mod, conf.int = TRUE)</pre>
ggplot(d, aes(estimate, term, xmin = conf.low, xmax = conf.high, height = 0)) +
  geom_point() +
  geom_vline(xintercept = 0, lty = 4) +
  geom_errorbarh()
# aside: There are tidy() and glance() methods for lm.summary objects too.
# this can be useful when you want to conserve memory by converting large lm
# objects into their leaner summary.lm equivalents.
s <- summary(mod)</pre>
tidy(s, conf.int = TRUE)
glance(s)
augment(mod)
augment(mod, mtcars, interval = "confidence")
# predict on new data
newdata <- mtcars %>%
  head(6) %>%
  mutate(wt = wt + 1)
augment(mod, newdata = newdata)
# ggplot2 example where we also construct 95% prediction interval
```

308 tidy.lm.beta

```
# simpler bivariate model since we're plotting in 2D
mod2 <- lm(mpg ~ wt, data = mtcars)</pre>
au <- augment(mod2, newdata = newdata, interval = "prediction")</pre>
ggplot(au, aes(wt, mpg)) +
  geom_point() +
  geom\_line(aes(y = .fitted)) +
  geom_ribbon(aes(ymin = .lower, ymax = .upper), col = NA, alpha = 0.3)
# predict on new data without outcome variable. Output does not include .resid
newdata <- newdata %>%
  select(-mpg)
augment(mod, newdata = newdata)
au <- augment(mod, data = mtcars)</pre>
ggplot(au, aes(.hat, .std.resid)) +
  geom_vline(size = 2, colour = "white", xintercept = 0) +
  geom_hline(size = 2, colour = "white", yintercept = 0) +
  geom_point() +
  geom_smooth(se = FALSE)
plot(mod, which = 6)
ggplot(au, aes(.hat, .cooksd)) +
  geom_vline(xintercept = 0, colour = NA) +
  geom_abline(slope = seq(0, 3, by = 0.5), colour = "white") +
  geom_smooth(se = FALSE) +
  geom_point()
# column-wise models
a <- matrix(rnorm(20), nrow = 10)
b <- a + rnorm(length(a))</pre>
result <- lm(b \sim a)
tidy(result)
```

tidy.lm.beta

Tidy a(n) lm.beta object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

tidy.lm.beta 309

Usage

```
## S3 method for class 'lm.beta'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

An lm.beta object created by lm.beta::lm.beta.

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

If the linear model is an mlm object (multiple linear model), there is an additional column response. If you have missing values in your model data, you may need to refit the model with na.action = na.exclude.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(), glance.svyglm(), tidy.glm(), tidy.lm(), tidy.mlm(), tidy.summary.lm()
```

310 tidy.lmodel2

Examples

```
# load libraries for models and data
library(lm.beta)
# fit models
mod <- stats::lm(speed ~ ., data = cars)</pre>
std <- lm.beta(mod)</pre>
# summarize model fit with tidiers
tidy(std, conf.int = TRUE)
# generate data
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt < -c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl", "Trt"))</pre>
weight <- c(ctl, trt)</pre>
# fit models
mod2 <- lm(weight ~ group)</pre>
std2 <- lm.beta(mod2)</pre>
# summarize model fit with tidiers
tidy(std2, conf.int = TRUE)
```

tidy.lmodel2

Tidy a(n) lmodel2 object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'lmodel2' tidy(x, ...)
```

Arguments

x A lmodel2 object returned by lmodel2::lmodel2().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.lmodel2 311

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

There are always only two terms in an lmodel2: "Intercept" and "Slope". These are computed by four methods: OLS (ordinary least squares), MA (major axis), SMA (standard major axis), and RMA (ranged major axis).

The returned p-value is one-tailed and calculated via a permutation test. A permutational test is used because distributional assumptions may not be valid. More information can be found in vignette("mod2user", package = "lmodel2").

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

term The name of the regression term.
method Either OLS/MA/SMA/RMA

See Also

```
tidy(), lmodel2::lmodel2()
Other lmodel2 tidiers: glance.lmodel2()
```

```
# load libraries for models and data
library(lmodel2)

data(mod2ex2)
Ex2.res <- lmodel2(Prey ~ Predators, data = mod2ex2, "relative", "relative", 99)
Ex2.res

# summarize model fit with tidiers + visualization
tidy(Ex2.res)
glance(Ex2.res)

# this allows coefficient plots with ggplot2
library(ggplot2)

ggplot(tidy(Ex2.res), aes(estimate, term, color = method)) +
    geom_point() +</pre>
```

312 tidy.lmRob

```
geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
geom_errorbarh(aes(xmin = conf.low, xmax = conf.high))
```

tidy.lmRob

Tidy a(n) lmRob object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'lmRob' tidy(x, ...)
```

Arguments

x A lmRob object returned from robust::lmRob().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For tidiers for robust models from the MASS package see tidy.rlm().

See Also

```
robust::lmRob()
Other robust tidiers: augment.lmRob(), glance.glmRob(), glance.lmRob(), tidy.glmRob()
```

tidy.lmrob 313

Examples

```
# load modeling library
library(robust)

# fit model
m <- lmRob(mpg ~ wt, data = mtcars)

# summarize model fit with tidiers
tidy(m)
augment(m)
glance(m)</pre>
```

tidy.lmrob

Tidy a(n) lmrob object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'lmrob'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x A lmrob object returned from robustbase::lmrob().

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

314 tidy.lsmobj

Details

For tidiers for robust models from the MASS package see tidy.rlm().

See Also

```
robustbase::lmrob()
Other robustbase tidiers: augment.glmrob(), augment.lmrob(), glance.lmrob(), tidy.glmrob()
```

Examples

```
if (requireNamespace("robustbase", quietly = TRUE)) {
 # load libraries for models and data
 library(robustbase)
 data(coleman)
 set.seed(0)
 m <- lmrob(Y ~ ., data = coleman)</pre>
 tidy(m)
 augment(m)
 glance(m)
 data(carrots)
 Rfit <- glmrob(cbind(success, total - success) ~ logdose + block,
   family = binomial, data = carrots, method = "Mqle",
   control = glmrobMqle.control(tcc = 1.2)
 tidy(Rfit)
 augment(Rfit)
}
```

tidy.lsmobj

Tidy a(n) lsmobj object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'lsmobj'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

tidy.lsmobj 315

Arguments

X	An 1smobj object.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid(). Cautionary note: misspecified arguments may be silently ignored!

Details

Returns a data frame with one observation for each estimated marginal mean, and one column for each combination of factors. When the input is a contrast, each row will contain one estimated contrast.

There are a large number of arguments that can be passed on to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid().

Value

```
A tibble::tibble() with columns:
                  Upper bound on the confidence interval for the estimate.
conf.high
conf.low
                  Lower bound on the confidence interval for the estimate.
                  Levels being compared.
contrast
df
                  Degrees of freedom used by this term in the model.
null.value
                  Value to which the estimate is compared.
p.value
                  The two-sided p-value associated with the observed statistic.
std.error
                  The standard error of the regression term.
                  Expected marginal mean
estimate
statistic
                  T-ratio statistic
```

See Also

```
tidy(), emmeans::ref_grid(), emmeans::emmeans(), emmeans::contrast()
Other emmeans tidiers: tidy.emmGrid(), tidy.ref.grid(), tidy.summary_emm()
```

```
# load libraries for models and data
library(emmeans)

# linear model for sales of oranges per day
oranges_lm1 <- lm(sales1 ~ price1 + price2 + day + store, data = oranges)</pre>
```

316 tidy.manova

```
# reference grid; see vignette("basics", package = "emmeans")
oranges_rg1 <- ref_grid(oranges_lm1)</pre>
td <- tidy(oranges_rg1)</pre>
td
# marginal averages
marginal <- emmeans(oranges_rg1, "day")</pre>
tidy(marginal)
# contrasts
tidy(contrast(marginal))
tidy(contrast(marginal, method = "pairwise"))
# plot confidence intervals
library(ggplot2)
ggplot(tidy(marginal, conf.int = TRUE), aes(day, estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# by multiple prices
by_price <- emmeans(oranges_lm1, "day",</pre>
  by = "price2",
  at = list(
    price1 = 50, price2 = c(40, 60, 80),
    day = c("2", "3", "4")
)
by_price
tidy(by_price)
ggplot(tidy(by_price, conf.int = TRUE), aes(price2, estimate, color = day)) +
  geom_line() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# joint_tests
tidy(joint_tests(oranges_lm1))
```

tidy.manova

Tidy a(n) manova object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

tidy.manova 317

Usage

```
## S3 method for class 'manova'
tidy(x, test = "Pillai", ...)
```

Arguments

X	A manova object return from stats::manova().
test	One of "Pillai" (Pillai's trace), "Wilks" (Wilk's lambda), "Hotelling-Lawley" (Hotelling-Lawley trace) or "Roy" (Roy's greatest root) indicating which test statistic should be used. Defaults to "Pillai".
	Arguments passed on to stats::summary.manova
	object An object of class "manova" or an aov object with multiple responses.
	intercept logical. If TRUE, the intercept term is included in the table.
	tol tolerance to be used in deciding if the residuals are rank-deficient: see qr.

Details

Depending on which test statistic is specified only one of pillai, wilks, hl or roy is included.

Value

```
A tibble::tibble() with columns:
```

den.df Degrees of freedom of the denominator.

num. df Degrees of freedom.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

term The name of the regression term.

pillai Pillai's trace. wilks Wilk's lambda.

hl Hotelling-Lawley trace.
roy Roy's greatest root.

See Also

```
tidy(), stats::summary.manova()
Other anova tidiers: glance.anova(), glance.aov(), tidy.TukeyHSD(), tidy.anova(), tidy.aov(),
tidy.aovlist()
```

```
npk2 <- within(npk, foo <- rnorm(24))
m <- manova(cbind(yield, foo) \sim block + N * P * K, npk2)
tidy(m)
```

318 tidy.map

tidy.map

Tidy a(n) map object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'map' tidy(x, ...)
```

Arguments

x A map object returned from maps::map().

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

term The name of the regression term.

long Longitude.

Latitude.

Remaining columns give information on geographic attributes and depend on the inputted map object. See ?maps::map for more information.

See Also

```
tidy(), maps::map()
```

tidy.margins 319

Examples

```
# load libraries for models and data
library(maps)
library(ggplot2)

ca <- map("county", "ca", plot = FALSE, fill = TRUE)

tidy(ca)

qplot(long, lat, data = ca, geom = "polygon", group = group)

tx <- map("county", "texas", plot = FALSE, fill = TRUE)
tidy(tx)
qplot(long, lat,
    data = tx, geom = "polygon", group = group,
    colour = I("white")
)</pre>
```

tidy.margins

Tidy a(n) margins object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'margins'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A margins object returned from margins::margins().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

320 tidy.margins

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The margins package provides a way to obtain coefficient marginal effects for a variety of (non-linear) models, such as logit or models with multiway interaction terms. Note that the glance.margins() method requires rerunning the underlying model again, which can take some time. Similarly, an augment.margins() method is not currently supported, but users can simply run the underlying model to obtain the same information.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), margins::margins()
```

```
# load libraries for models and data
library(margins)

# example 1: logit model
mod_log <- glm(am ~ cyl + hp + wt, data = mtcars, family = binomial)

# get tidied "naive" model coefficients
tidy(mod_log)

# convert to marginal effects with margins()
marg_log <- margins(mod_log)

# get tidied marginal effects
tidy(marg_log)
tidy(marg_log, conf.int = TRUE)</pre>
```

tidy.Mclust 321

```
# requires running the underlying model again. quick for this example
glance(marg_log)
# augmenting `margins` outputs isn't supported, but
# you can get the same info by running on the underlying model
augment(mod_log)
# example 2: threeway interaction terms
mod_ie <- lm(mpg ~ wt * cyl * disp, data = mtcars)</pre>
# get tidied "naive" model coefficients
tidy(mod_ie)
# convert to marginal effects with margins()
marg_ie0 <- margins(mod_ie)</pre>
# get tidied marginal effects
tidy(marg_ie0)
glance(marg_ie0)
# marginal effects evaluated at specific values of a variable (here: cyl)
marg_ie1 \leftarrow margins(mod_ie, at = list(cyl = c(4,6,8)))
# summarize model fit with tidiers
tidy(marg_ie1)
# marginal effects of one interaction variable (here: wt), modulated at
# specific values of the two other interaction variables (here: cyl and drat)
marg_ie2 <- margins(mod_ie,</pre>
                    variables = "wt",
                    at = list(cyl = c(4,6,8), drat = c(3, 3.5, 4)))
# summarize model fit with tidiers
tidy(marg_ie2)
```

tidy.Mclust

Tidy a(n) Mclust object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'Mclust' tidy(x, ...)
```

322 tidy.Mclust

Arguments

. . .

x An Mclust object return from mclust::Mclust().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

proportion The mixing proportion of each component size Number of points assigned to cluster.

mean The mean for each component. In case of 2+ dimensional models, a column

with the mean is added for each dimension. NA for noise component

variance In case of one-dimensional and spherical models, the variance for each compo-

nent, omitted otherwise. NA for noise component

component Cluster id as a factor.

See Also

```
tidy(), mclust::Mclust()
Other mclust tidiers: augment.Mclust()
```

```
# load library for models and data
library(mclust)

# load data manipulation libraries
library(dplyr)
library(tibble)
library(purrr)
library(tidyr)

set.seed(27)

centers <- tibble(
   cluster = factor(1:3),
    # number points in each cluster
   num_points = c(100, 150, 50),
    # x1 coordinate of cluster center</pre>
```

tidy.mediate 323

```
x1 = c(5, 0, -3),
# x2 coordinate of cluster center
x2 = c(-1, 1, -2)
)

points <- centers %>%
  mutate(
    x1 = map2(num_points, x1, rnorm),
    x2 = map2(num_points, x2, rnorm)
) %>%
  select(-num_points, -cluster) %>%
  unnest(c(x1, x2))

# fit model
m <- Mclust(points)

# summarize model fit with tidiers
tidy(m)
augment(m, points)
glance(m)</pre>
```

tidy.mediate

Tidy a(n) mediate object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'mediate'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A mediate object produced by a call to mediation::mediate().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
•••	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be

324 tidy.mediate

used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The tibble has four rows. The first two indicate the mediated effect in the control and treatment groups, respectively. And the last two the direct effect in each group.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), mediation::mediate()
```

```
# load libraries for models and data
library(mediation)

data(jobs)

# fit models
b <- lm(job_seek ~ treat + econ_hard + sex + age, data = jobs)
c <- lm(depress2 ~ treat + job_seek + econ_hard + sex + age, data = jobs)
mod <- mediate(b, c, sims = 50, treat = "treat", mediator = "job_seek")

# summarize model fit with tidiers
tidy(mod)
tidy(mod, conf.int = TRUE)
tidy(mod, conf.int = TRUE, conf.level = .99)</pre>
```

tidy.mfx 325

tidy.mfx	Tidy a(n) mfx object
----------	----------------------

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

The particular functions below provide generic tidy methods for objects returned by the mfx package, preserving the calculated marginal effects instead of the naive model coefficients. The returned tidy tibble will also include an additional "atmean" column indicating how the marginal effects were originally calculated (see Details below).

Usage

```
## S3 method for class 'mfx'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
## S3 method for class 'logitmfx'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
## S3 method for class 'negbinmfx'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
## S3 method for class 'poissonmfx'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
## S3 method for class 'probitmfx'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A logitmfx, negbinmfx, poissonmfx, or probitmfx object. (Note that betamfx objects receive their own set of tidiers.)
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

326 tidy.mfx

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The mfx package provides methods for calculating marginal effects for various generalized linear models (GLMs). Unlike standard linear models, estimated model coefficients in a GLM cannot be directly interpreted as marginal effects (i.e., the change in the response variable predicted after a one unit change in one of the regressors). This is because the estimated coefficients are multiplicative, dependent on both the link function that was used for the estimation and any other variables that were included in the model. When calculating marginal effects, users must typically choose whether they want to use i) the average observation in the data, or ii) the average of the sample marginal effects. See vignette("mfxarticle") from the mfx package for more details.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-
	zero.
std.error	The standard error of the regression term.
term	The name of the regression term.
atmean	TRUE if the marginal effects were originally calculated as the partial effects for the average observation. If FALSE, then these were instead calculated as average partial effects.

See Also

```
tidy(), mfx::logitmfx(), mfx::negbinmfx(), mfx::poissonmfx(), mfx::probitmfx()
Other mfx tidiers: augment.betamfx(), augment.mfx(), glance.betamfx(), glance.mfx(),
tidy.betamfx()
```

```
# load libraries for models and data
library(mfx)

# get the marginal effects from a logit regression
mod_logmfx <- logitmfx(am ~ cyl + hp + wt, atmean = TRUE, data = mtcars)
tidy(mod_logmfx, conf.int = TRUE)</pre>
```

tidy.mjoint 327

```
# compare with the naive model coefficients of the same logit call
tidy(
   glm(am ~ cyl + hp + wt, family = binomial, data = mtcars),
   conf.int = TRUE
)

augment(mod_logmfx)
glance(mod_logmfx)

# another example, this time using probit regression
mod_probmfx <- probitmfx(am ~ cyl + hp + wt, atmean = TRUE, data = mtcars)

tidy(mod_probmfx, conf.int = TRUE)
augment(mod_probmfx)
glance(mod_probmfx)</pre>
```

tidy.mjoint

 $Tidy \ a(n) \ mjoint \ object$

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## $3 method for class 'mjoint'
tidy(
    x,
    component = "survival",
    conf.int = FALSE,
    conf.level = 0.95,
    boot_se = NULL,
    ...
)
```

Arguments

x An mjoint object returned from joineRML::mjoint().

component Character specifying whether to tidy the survival or the lor

Character specifying whether to tidy the survival or the longitudinal component of the model. Must be either "survival" or "longitudinal". Defaults to "survival".

conf.int Logical indicat

t Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

328 tidy.mjoint

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

boot_se Optionally a bootSE object from joineRML::bootSE(). If specified, calculates

confidence intervals via the bootstrap. Defaults to NULL, in which case standard $\,$

errors are calculated from the empirical information matrix.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

std.error The standard error of the regression term.

term The name of the regression term.

See Also

```
tidy(), joineRML::mjoint(), joineRML::bootSE()
Other mjoint tidiers: glance.mjoint()
```

```
# broom only skips running these examples because the example models take a
# while to generate—they should run just fine, though!
## Not run:

# load libraries for models and data
library(joineRML)

# fit a joint model with bivariate longitudinal outcomes
data(heart.valve)
```

tidy.mle2

```
hvd <- heart.valve[!is.na(heart.valve$log.grad) &</pre>
  !is.na(heart.valve$log.lvmi) &
  heart.valve$num <= 50, ]
fit <- mjoint(</pre>
  formLongFixed = list(
    "grad" = log.grad ~ time + sex + hs,
    "lvmi" = log.lvmi ~ time + sex
  ),
  formLongRandom = list(
    "grad" = ~ 1 | num,
    "lvmi" = ~ time | num
  ),
  formSurv = Surv(fuyrs, status) ~ age,
  data = hvd,
  inits = list("gamma" = c(0.11, 1.51, 0.80)),
  timeVar = "time"
)
# extract the survival fixed effects
tidy(fit)
# extract the longitudinal fixed effects
tidy(fit, component = "longitudinal")
# extract the survival fixed effects with confidence intervals
tidy(fit, ci = TRUE)
# extract the survival fixed effects with confidence intervals based
# on bootstrapped standard errors
bSE <- bootSE(fit, nboot = 5, safe.boot = TRUE)</pre>
tidy(fit, boot_se = bSE, ci = TRUE)
# augment original data with fitted longitudinal values and residuals
hvd2 <- augment(fit)</pre>
# extract model statistics
glance(fit)
## End(Not run)
```

tidy.mle2

Tidy a(n) mle2 object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers

330 tidy.mle2

to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'mle2'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

An mle2 object created by a call to bbmle::mle2().

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), bbmle::mle2(), tidy_optim()
```

tidy.mlm 331

Examples

```
# load libraries for models and data
library(bbmle)

# generate data
x <- 0:10
y <- c(26, 17, 13, 12, 20, 5, 9, 8, 5, 4, 8)
d <- data.frame(x, y)

# fit model
fit <- mle2(y ~ dpois(lambda = ymean),
    start = list(ymean = mean(y)), data = d
)

# summarize model fit with tidiers
tidy(fit)</pre>
```

tidy.mlm

Tidy a(n) mlm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'mlm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	An mlm object created by stats::lm() with a matrix as the response.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if $conf.int = TRUE$. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

332 tidy.mlm

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

In contrast to 1m object (simple linear model), tidy output for m1m (multiple linear model) objects contain an additional column response.

If you have missing values in your model data, you may need to refit the model with na.action = na.exclude.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy()
```

```
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(), glance.svyglm(), tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.summary.lm()
```

```
# fit model
mod <- lm(cbind(mpg, disp) ~ wt, mtcars)
# summarize model fit with tidiers
tidy(mod, conf.int = TRUE)</pre>
```

tidy.mlogit 333

tidy.mlogit	Tidying methods for logit models

Description

These methods tidy the coefficients of mnl and nl models generated by the functions of the mlogit package.

Usage

```
## S3 method for class 'mlogit'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	an object returned from mlogit::mlogit().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

334 tidy.muhaz

See Also

```
tidy(), mlogit::mlogit()
Other mlogit tidiers: augment.mlogit(), glance.mlogit()
```

Examples

```
# load libraries for models and data
library(mlogit)

data("Fishing", package = "mlogit")
Fish <- dfidx(Fishing, varying = 2:9, shape = "wide", choice = "mode")

# fit model
m <- mlogit(mode ~ price + catch | income, data = Fish)

# summarize model fit with tidiers
tidy(m)
augment(m)
glance(m)</pre>
```

tidy.muhaz

Tidy a(n) muhaz object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'muhaz' tidy(x, ...)
```

Arguments

x A muhaz object returned by muhaz::muhaz().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

tidy.multinom 335

Value

```
A tibble::tibble() with columns:

time Point in time.

estimate Estimated hazard rate.
```

See Also

```
tidy(), muhaz::muhaz()
Other muhaz tidiers: glance.muhaz()
```

Examples

```
# load libraries for models and data
library(muhaz)
library(survival)

# fit model
x <- muhaz(ovarian$futime, ovarian$fustat)

# summarize model fit with tidiers
tidy(x)
glance(x)</pre>
```

tidy.multinom

Tidying methods for multinomial logistic regression models

Description

These methods tidy the coefficients of multinomial logistic regression models generated by multinom of the nnet package.

Usage

```
## S3 method for class 'multinom'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

conf.level

x	A multinom object returned from nnet::multinom().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

336 tidy.multinom

exponentiate

Logical indicating whether or not to exponentiate the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate. estimate The estimated value of the regression term. p.value The two-sided p-value associated with the observed statistic. statistic The value of a T-statistic to use in a hypothesis that the regression term is nonstd.error The standard error of the regression term. The name of the regression term. term y.value The response level.

See Also

```
tidy(), nnet::multinom()
Other multinom tidiers: glance.multinom()
```

```
# load libraries for models and data
library(nnet)
library(MASS)

example(birthwt)

bwt.mu <- multinom(low ~ ., bwt)

tidy(bwt.mu)
glance(bwt.mu)

# or, for output from a multinomial logistic regression</pre>
```

tidy.negbin 337

```
fit.gear <- multinom(gear ~ mpg + factor(am), data = mtcars)
tidy(fit.gear)
glance(fit.gear)</pre>
```

tidy.negbin

Tidy a(n) negbin object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'negbin'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

×	A glm.nb object returned by MASS::glm.nb().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
	For tidy(), additional arguments passed to summary(). Otherwise ignored.

See Also

```
MASS::glm.nb()
Other glm.nb tidiers: glance.negbin()
```

```
# load libraries for models and data
library(MASS)
# fit model
r <- glm.nb(Days ~ Sex / (Age + Eth * Lrn), data = quine)</pre>
```

338 tidy.nlrq

```
# summarize model fit with tidiers
tidy(r)
glance(r)
```

tidy.nlrq

Tidy a(n) nlrq object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'nlrq'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

A nlrq object returned from quantreg::nlrq().

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.

tidy.nls 339

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-

zero.

std.error The standard error of the regression term.

term The name of the regression term.

See Also

```
tidy(), quantreg::nlrq()
Other quantreg tidiers: augment.nlrq(), augment.rq(), augment.rqs(), glance.nlrq(), glance.rq(),
tidy.rq(), tidy.rqs()
```

Examples

```
# load modeling library
library(quantreg)

# build artificial data with multiplicative error
set.seed(1)
dat <- NULL
dat$x <- rep(1:25, 20)
dat$y <- SSlogis(dat$x, 10, 12, 2) * rnorm(500, 1, 0.1)

# fit the median using nlrq
mod <- nlrq(y ~ SSlogis(x, Asym, mid, scal),
    data = dat, tau = 0.5, trace = TRUE
)

# summarize model fit with tidiers
tidy(mod)
glance(mod)
augment(mod)</pre>
```

tidy.nls

Tidy a(n) nls object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'nls'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

340 tidy.nls

Arguments

An nls object returned from stats::nls().

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

Conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

```
conf.high Upper bound on the confidence interval for the estimate.

Lower bound on the confidence interval for the estimate.

The estimated value of the regression term.

P.value The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

Std.error The standard error of the regression term.

The name of the regression term.
```

See Also

```
tidy, stats::nls(), stats::summary.nls()
Other nls tidiers: augment.nls(), glance.nls()
```

```
# fit model
n <- nls(mpg ~ k * e^wt, data = mtcars, start = list(k = 1, e = 2))
# summarize model fit with tidiers + visualization
tidy(n)
augment(n)
glance(n)</pre>
```

tidy.numeric 341

```
library(ggplot2)
ggplot(augment(n), aes(wt, mpg)) +
  geom_point() +
  geom_line(aes(y = .fitted))

newdata <- head(mtcars)
newdata$wt <- newdata$wt + 1

augment(n, newdata = newdata)</pre>
```

tidy.numeric

Tidy atomic vectors

Description

Vector tidiers are deprecated and will be removed from an upcoming release of broom.

Usage

```
## S3 method for class 'numeric'
tidy(x, ...)
## S3 method for class 'character'
tidy(x, ...)
## S3 method for class 'logical'
tidy(x, ...)
```

Arguments

x An object of class "numeric", "integer", "character", or "logical". Most likely a named vector

... Extra arguments (not used)

Details

Turn atomic vectors into data frames, where the names of the vector (if they exist) are a column and the values of the vector are a column.

See Also

```
Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable()

Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable()

Other deprecated: bootstrap(), confint_tidy(), data.frame_tidiers, finish_glance(), fix_data_frame(), summary_tidiers, tidy.density(), tidy.dist(), tidy.ftable()
```

342 tidy.orcutt

Examples

```
## Not run:
x <- 1:5
names(x) <- letters[1:5]
tidy(x)
## End(Not run)</pre>
```

tidy.orcutt

Tidy a(n) orcutt object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'orcutt' tidy(x, ...)
```

Arguments

x An orcutt object returned from orcutt::cochrane.orcutt().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

rhe estimated value of the regression term.

p.value
The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

Std.error
The standard error of the regression term.

The name of the regression term.

tidy.pairwise.htest 343

See Also

```
orcutt::cochrane.orcutt()
Other orcutt tidiers: glance.orcutt()
```

Examples

```
# load libraries for models and data
library(orcutt)

# fit model and summarize results
reg <- lm(mpg ~ wt + qsec + disp, mtcars)
tidy(reg)

co <- cochrane.orcutt(reg)
tidy(co)
glance(co)</pre>
```

tidy.pairwise.htest

Tidy a(n) pairwise.htest object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'pairwise.htest' tidy(x, ...)
```

Arguments

A pairwise.htest object such as those returned from stats::pairwise.t.test()
or stats::pairwise.wilcox.test().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

344 tidy.pam

Details

Note that in one-sided tests, the alternative hypothesis of each test can be stated as "group1 is greater/less than group2".

Note also that the columns of group1 and group2 will always be a factor, even if the original input is (e.g.) numeric.

Value

```
A tibble::tibble() with columns:
```

```
group1 First group being compared.
group2 Second group being compared.
```

p.value The two-sided p-value associated with the observed statistic.

See Also

```
stats::pairwise.t.test(), stats::pairwise.wilcox.test(), tidy()
Other htest tidiers: augment.htest(), tidy.htest(), tidy.power.htest()
```

Examples

```
attach(airquality)
Month <- factor(Month, labels = month.abb[5:9])
ptt <- pairwise.t.test(Ozone, Month)
tidy(ptt)

library(modeldata)
data(hpc_data)
attach(hpc_data)
ptt2 <- pairwise.t.test(compounds, class)
tidy(ptt2)

tidy(pairwise.t.test(compounds, class, alternative = "greater"))
tidy(pairwise.t.test(compounds, class, alternative = "less"))

tidy(pairwise.wilcox.test(compounds, class))</pre>
```

tidy.pam

Tidy a(n) pam object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

tidy.pam 345

Usage

```
## S3 method for class 'pam'
tidy(x, col.names = paste0("x", 1:ncol(x$medoids)), ...)
```

Arguments

An pam object returned from cluster::pam()

Col.names

Column names in the input data frame. Defaults to the names of the variables in x.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

For examples, see the pam vignette.

Value

A tibble::tibble() with columns:

size	Size of each cluster.
max.diss	Maximal dissimilarity between the observations in the cluster and that cluster's medoid.
avg.diss	Average dissimilarity between the observations in the cluster and that cluster's medoid.
diameter	Diameter of the cluster.
separation	Separation of the cluster.
avg.width	Average silhouette width of the cluster.
cluster	A factor describing the cluster from 1:k.

See Also

```
tidy(), cluster::pam()
Other pam tidiers: augment.pam(), glance.pam()
```

346 tidy.plm

Examples

```
# load libraries for models and data
library(dplyr)
library(ggplot2)
library(cluster)
library(modeldata)
data(hpc_data)
x <- hpc_data[, 2:5]</pre>
p \leftarrow pam(x, k = 4)
# summarize model fit with tidiers + visualization
tidy(p)
glance(p)
augment(p, x)
augment(p, x) %>%
  ggplot(aes(compounds, input_fields)) +
  geom_point(aes(color = .cluster)) +
  geom_text(aes(label = cluster), data = tidy(p), size = 10)
```

tidy.plm

Tidy a(n) plm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'plm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A plm objected returned by plm::plm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

tidy.plm 347

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

. . .

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), plm::plm(), tidy.lm()
Other plm tidiers: augment.plm(), glance.plm()
```

```
# load libraries for models and data
library(plm)

# load data
data("Produc", package = "plm")

# fit model
zz <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp,
    data = Produc, index = c("state", "year")
)

# summarize model fit with tidiers
summary(zz)

tidy(zz)
tidy(zz, conf.int = TRUE)
tidy(zz, conf.int = TRUE, conf.level = 0.9)</pre>
```

348 tidy.poLCA

```
augment(zz)
glance(zz)
```

tidy.poLCA

Tidy a(n) poLCA object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'poLCA' tidy(x, ...)
```

Arguments

. . .

x A poLCA object returned from poLCA::poLCA().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

class The class under consideration.
outcome Outcome of manifest variable.

std.error The standard error of the regression term.

variable Manifest variable

estimate Estimated class-conditional response probability

See Also

```
tidy(), poLCA::poLCA()
Other poLCA tidiers: augment.poLCA(), glance.poLCA()
```

tidy.poLCA 349

```
# load libraries for models and data
library(poLCA)
library(dplyr)
# generate data
data(values)
f \leftarrow cbind(A, B, C, D) \sim 1
# fit model
M1 <- poLCA(f, values, nclass = 2, verbose = FALSE)
# summarize model fit with tidiers + visualization
tidy(M1)
augment(M1)
glance(M1)
library(ggplot2)
ggplot(tidy(M1), aes(factor(class), estimate, fill = factor(outcome))) +
  geom_bar(stat = "identity", width = 1) +
  facet_wrap(~variable)
# three-class model with a single covariate.
data(election)
f2a <- cbind(
  MORALG, CARESG, KNOWG, LEADG, DISHONG, INTELG,
  MORALB, CARESB, KNOWB, LEADB, DISHONB, INTELB
) ~ PARTY
nes2a <- poLCA(f2a, election, nclass = 3, nrep = 5, verbose = FALSE)</pre>
td <- tidy(nes2a)
td
ggplot(td, aes(outcome, estimate, color = factor(class), group = class)) +
  geom_line() +
  facet_wrap(\sim variable, nrow = 2) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
au <- augment(nes2a)</pre>
au
count(au, .class)
# if the original data is provided, it leads to NAs in new columns
```

350 tidy.polr

```
# for rows that weren't predicted
au2 <- augment(nes2a, data = election)
au2
dim(au2)</pre>
```

tidy.polr

Tidy a(n) polr object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## $3 method for class 'polr'
tidy(
    x,
    conf.int = FALSE,
    conf.level = 0.95,
    exponentiate = FALSE,
    p.values = FALSE,
    ...
)
```

Arguments

Χ	A polr object returned from MASS::polr().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf. int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to exponentiate the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.
p.values	Logical. Should p-values be returned, based on chi-squared tests from MASS::dropterm(). Defaults to FALSE.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.polr 351

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

In broom 0.7.0 the coefficient_type column was renamed to coef.type, and the contents were changed as well. Now the contents are coefficient and scale, rather than coefficient and zeta.

Calculating p.values with the dropterm() function is the approach suggested by the MASS package author. This approach is computationally intensive so that p.values are only returned if requested explicitly. Additionally, it only works for models containing no variables with more than two categories. If this condition is not met, a message is shown and NA is returned instead of p-values.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy, MASS::polr()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.clmm(), glance.polr(),
glance.svyolr(), tidy.clm(), tidy.clmm(), tidy.svyolr()
```

```
# load libraries for models and data
library(MASS)

# fit model
fit <- polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)

# summarize model fit with tidiers
tidy(fit, exponentiate = TRUE, conf.int = TRUE)

glance(fit)
augment(fit, type.predict = "class")</pre>
```

352 tidy.power.htest

```
fit2 <- polr(factor(gear) ~ am + mpg + qsec, data = mtcars)
tidy(fit, p.values = TRUE)</pre>
```

tidy.power.htest

 $Tidy \ a(n) \ power.htest \ object$

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'power.htest'
tidy(x, ...)
```

Arguments

. . .

x A power.htest object such as those returned from stats::power.t.test().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

delta True difference in means.

Number of observations by component.powerPower achieved for given value of n.

sd Standard deviation.

sig.level Significance level (Type I error probability).

tidy.prcomp 353

See Also

```
stats::power.t.test()
Other htest tidiers: augment.htest(), tidy.htest(), tidy.pairwise.htest()
```

Examples

```
ptt <- power.t.test(n = 2:30, delta = 1)
tidy(ptt)
library(ggplot2)
ggplot(tidy(ptt), aes(n, power)) +
  geom_line()</pre>
```

tidy.prcomp

Tidy a(n) prcomp object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'prcomp'
tidy(x, matrix = "u", ...)
```

Arguments

Х

A prcomp object returned by stats::prcomp().

matrix

Character specifying which component of the PCA should be tidied.

- "u", "samples", "scores", or "x": returns information about the map from the original space into principle components space.
- "v", "rotation", "loadings" or "variables": returns information about the map from principle components space back into the original space.
- "d", "eigenvalues" or "pcs": returns information about the eigenvalues.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

354 tidy.prcomp

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

See https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca for information on how to interpret the various tidied matrices. Note that SVD is only equivalent to PCA on centered data.

Value

A tibble::tibble with columns depending on the component of PCA being tidied.

If matrix is "u", "samples", "scores", or "x" each row in the tidied output corresponds to the original data in PCA space. The columns are:

row ID of the original observation (i.e. rowname from original data).

PC Integer indicating a principal component.

value The score of the observation for that particular principal component. That is, the

location of the observation in PCA space.

If matrix is "v", "rotation", "loadings" or "variables", each row in the tidied output corresponds to information about the principle components in the original space. The columns are:

row The variable labels (colnames) of the data set on which PCA was performed.

PC An integer vector indicating the principal component.

value The value of the eigenvector (axis score) on the indicated principal component.

If matrix is "d", "eigenvalues" or "pcs", the columns are:

PC An integer vector indicating the principal component.

std.dev Standard deviation explained by this PC.

percent Fraction of variation explained by this component (a numeric value between 0

and 1).

cumulative Cumulative fraction of variation explained by principle components up to this

component (a numeric value between 0 and 1).

See Also

```
stats::prcomp(), svd_tidiers
```

Other svd tidiers: augment.prcomp(), tidy_irlba(), tidy_svd()

tidy.pyears 355

Examples

```
pc <- prcomp(USArrests, scale = TRUE)</pre>
# information about rotation
tidy(pc)
# information about samples (states)
tidy(pc, "samples")
# information about PCs
tidy(pc, "pcs")
# state map
library(dplyr)
library(ggplot2)
library(maps)
pc %>%
  tidy(matrix = "samples") %>%
  mutate(region = tolower(row)) %>%
  inner_join(map_data("state"), by = "region") %>%
  ggplot(aes(long, lat, group = group, fill = value)) +
  geom_polygon() +
  facet_wrap(~PC) +
  theme_void() +
  ggtitle("Principal components of arrest data")
au <- augment(pc, data = USArrests)</pre>
au
ggplot(au, aes(.fittedPC1, .fittedPC2)) +
  geom_point() +
  geom_text(aes(label = .rownames), vjust = 1, hjust = 1)
```

tidy.pyears

Tidy a(n) pyears object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'pyears' tidy(x, ...)
```

356 tidy.pyears

Arguments

. . .

x A pyears object returned from survival::pyears().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

expected is only present in the output when if a ratetable term is present.

If the data. frame = TRUE argument is supplied to pyears, this is simply the contents of x\$data.

Value

```
A tibble::tibble() with columns:
```

expected Expected number of events.

pyears Person-years of exposure.

n number of subjects contributing time

event observed number of events

See Also

```
tidy(), survival::pyears()
Other pyears tidiers: glance.pyears()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.survdiff(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# generate and format data
temp.yr <- tcut(mgus$dxyr, 55:92, labels = as.character(55:91))
temp.age <- tcut(mgus$age, 34:101, labels = as.character(34:100))
ptime <- ifelse(is.na(mgus$pctime), mgus$futime, mgus$pctime)
pstat <- ifelse(is.na(mgus$pctime), 0, 1)
pfit <- pyears(Surv(ptime / 365.25, pstat) ~ temp.yr + temp.age + sex, mgus,</pre>
```

tidy.rcorr 357

```
data.frame = TRUE
)

# summarize model fit with tidiers
tidy(pfit)
glance(pfit)

# if data.frame argument is not given, different information is present in
# output
pfit2 <- pyears(Surv(ptime / 365.25, pstat) ~ temp.yr + temp.age + sex, mgus)
tidy(pfit2)
glance(pfit2)</pre>
```

tidy.rcorr

Tidy a(n) rcorr object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'rcorr'
tidy(x, diagonal = FALSE, ...)
```

Arguments

Y

An rcorr object returned from Hmisc::rcorr().

diagonal

Logical indicating whether or not to include diagonal elements of the correlation matrix, or the correlation of a column with itself. For the elements, estimate is always 1 and p.value is always NA. Defaults to FALSE.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

358 tidy.rcorr

Details

Suppose the original data has columns A and B. In the correlation matrix from rcorr there may be entries for both the cor(A, B) and cor(B, A). Only one of these pairs will ever be present in the tidy output.

Value

A tibble::tibble() with columns:

column1 Name or index of the first column being described.

column2 Name or index of the second column being described.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

Number of observations used to compute the correlation

See Also

```
tidy(), Hmisc::rcorr()
```

```
# load libraries for models and data
library(Hmisc)
mat <- replicate(52, rnorm(100))</pre>
# add some NAs
mat[sample(length(mat), 2000)] <- NA</pre>
# also, column names
colnames(mat) <- c(LETTERS, letters)</pre>
# fit model
rc <- rcorr(mat)</pre>
# summarize model fit with tidiers + visualization
td <- tidy(rc)
td
library(ggplot2)
ggplot(td, aes(p.value)) +
  geom\_histogram(binwidth = .1)
ggplot(td, aes(estimate, p.value)) +
  geom_point() +
  scale_y_log10()
```

tidy.ref.grid 359

tray it (ii) reports object	tidy.ref.grid	Tidy a(n) ref.grid object	
-----------------------------	---------------	---------------------------	--

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'ref.grid'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A ref.grid object created by emmeans::ref_grid().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
•••	Additional arguments passed to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid(). Cautionary note: misspecified arguments may be silently ignored!

Details

Returns a data frame with one observation for each estimated marginal mean, and one column for each combination of factors. When the input is a contrast, each row will contain one estimated contrast.

There are a large number of arguments that can be passed on to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid().

Value

A tibble::tibble() with columns:

```
conf.high Upper bound on the confidence interval for the estimate.

Lower bound on the confidence interval for the estimate.

Degrees of freedom used by this term in the model.

P.value The two-sided p-value associated with the observed statistic.

The standard error of the regression term.

Expected marginal mean

Statistic T-ratio statistic
```

360 tidy.ref.grid

See Also

```
tidy(), emmeans::ref_grid(), emmeans::emmeans(), emmeans::contrast()
Other emmeans tidiers: tidy.emmGrid(), tidy.lsmobj(), tidy.summary_emm()
```

```
# load libraries for models and data
library(emmeans)
# linear model for sales of oranges per day
oranges_lm1 <- lm(sales1 ~ price1 + price2 + day + store, data = oranges)</pre>
# reference grid; see vignette("basics", package = "emmeans")
oranges_rg1 <- ref_grid(oranges_lm1)</pre>
td <- tidy(oranges_rg1)</pre>
td
# marginal averages
marginal <- emmeans(oranges_rg1, "day")</pre>
tidy(marginal)
# contrasts
tidy(contrast(marginal))
tidy(contrast(marginal, method = "pairwise"))
# plot confidence intervals
library(ggplot2)
ggplot(tidy(marginal, conf.int = TRUE), aes(day, estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# by multiple prices
by_price <- emmeans(oranges_lm1, "day",</pre>
  by = "price2",
 at = list(
   price1 = 50, price2 = c(40, 60, 80),
    day = c("2", "3", "4")
  )
)
by_price
tidy(by_price)
ggplot(tidy(by_price, conf.int = TRUE), aes(price2, estimate, color = day)) +
  geom_line() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# joint_tests
tidy(joint_tests(oranges_lm1))
```

tidy.regsubsets 361

tidy.regsubsets

Tidy a(n) regsubsets object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'regsubsets' tidy(x, ...)
```

Arguments

Χ

9

A regsubsets object created by leaps::regsubsets().

... Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

r.squared R squared statistic, or the percent of variation explained by the model.

adj.r.squared Adjusted R squared statistic

BIC Bayesian information criterion for the component.

mallows_cp Mallow's Cp statistic.

See Also

```
tidy(), leaps::regsubsets()
```

362 tidy.ridgelm

Examples

```
# load libraries for models and data
library(leaps)

# fit model
all_fits <- regsubsets(hp ~ ., mtcars)

# summarize model fit with tidiers
tidy(all_fits)</pre>
```

tidy.ridgelm

Tidy a(n) ridgelm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'ridgelm'
tidy(x, ...)
```

Arguments

. . .

x A ridgelm object returned from MASS::lm.ridge().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

GCV Generalized cross validation error estimate.

lambda Value of penalty parameter lambda. term The name of the regression term. tidy.rlm 363

estimate estimate of scaled coefficient using this lambda scale Scaling factor of estimated coefficient

See Also

```
tidy(), MASS::lm.ridge()
Other ridgelm tidiers: glance.ridgelm()
```

Examples

```
# load libraries for models and data
library(MASS)
names(longley)[1] <- "y"</pre>
# fit model and summarizd results
fit1 <- lm.ridge(y ~ ., longley)</pre>
tidy(fit1)
fit2 <- lm.ridge(y \sim ., longley, lambda = seq(0.001, .05, .001))
td2 <- tidy(fit2)
g2 <- glance(fit2)
# coefficient plot
library(ggplot2)
ggplot(td2, aes(lambda, estimate, color = term)) +
 geom_line()
# GCV plot
ggplot(td2, aes(lambda, GCV)) +
 geom_line()
# add line for the GCV minimizing estimate
ggplot(td2, aes(lambda, GCV)) +
 geom_line() +
 geom_vline(xintercept = g2$lambdaGCV, col = "red", lty = 2)
```

tidy.rlm

Tidy a(n) rlm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

364 tidy.rma

Usage

```
## S3 method for class 'rlm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x An rlm object returned by MASS::rlm().

conf.int Logical indicating whether or not to include a confidence interval in the tidied

output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must

be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

See Also

```
MASS::rlm()
Other rlm tidiers: augment.rlm(), glance.rlm()
```

tidy.rma

Tidy a(n) rma object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'rma'
tidy(
    x,
    conf.int = FALSE,
    conf.level = 0.95,
    exponentiate = FALSE,
```

tidy.rma 365

```
include_studies = FALSE,
measure = "GEN",
...
)
```

Arguments

x An rma object such as those created by metafor::rma(), metafor::rma.uni(), metafor::rma.glmm(), metafor::rma.mh(), metafor::rma.mv(), or metafor::rma.peto().

conf.int Logical indicating whether or not to include a confidence interval in the tidied

output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must

be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to

a 95 percent confidence interval.

exponentiate Logical indicating whether or not to exponentiate the the coefficient estimates.

This is typical for logistic and multinomial regressions, but a bad idea if there is

no log or logit link. Defaults to FALSE.

include_studies

Logical. Should individual studies be included in the output? Defaults to FALSE.

measure Measure type. See metafor::escalc()

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate. estimate The estimated value of the regression term. p.value The two-sided p-value associated with the observed statistic. statistic The value of a T-statistic to use in a hypothesis that the regression term is non-The standard error of the regression term. std.error The name of the individual study term The estimate type (summary vs individual study) type

366 tidy.roc

Examples

```
# load libraries for models and data
library(metafor)

df <-
    escalc(
        measure = "RR",
        ai = tpos,
        bi = tneg,
        ci = cpos,
        di = cneg,
        data = dat.bcg
)

meta_analysis <- rma(yi, vi, data = df, method = "EB")

tidy(meta_analysis)</pre>
```

tidy.roc

Tidy a(n) roc object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'roc'
tidy(x, ...)
```

Arguments

x

An roc object returned from a call to AUC::roc().

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

tidy.roc 367

Value

A tibble::tibble() with columns:

cutoff The cutoff used for classification. Observations with predicted probabilities

above this value were assigned class 1, and observations with predicted prob-

abilities below this value were assigned class 0.

fpr False positive rate.

tpr The true positive rate at the given cutoff.

See Also

```
tidy(), AUC::roc()
```

```
# load libraries for models and data
library(AUC)
# load data
data(churn)
# fit model
r <- roc(churn$predictions, churn$labels)</pre>
# summarize with tidiers + visualization
td <- tidy(r)
td
library(ggplot2)
ggplot(td, aes(fpr, tpr)) +
  geom_line()
# compare the ROC curves for two prediction algorithms
library(dplyr)
library(tidyr)
rocs <- churn %>%
  pivot_longer(contains("predictions"),
   names_to = "algorithm",
    values_to = "value"
  nest(data = -algorithm) %>%
  mutate(tidy_roc = purrr::map(data, ~ tidy(roc(.x$value, .x$labels)))) %>%
  unnest(tidy_roc)
ggplot(rocs, aes(fpr, tpr, color = algorithm)) +
  geom_line()
```

368 tidy.rq

tidy.rq	Tidy $a(n)$ rq object	

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'rq'
tidy(x, se.type = NULL, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	An rq object returned from quantreg::rq().
se.type	Character specifying the method to use to calculate standard errors. Passed to quantreg::summary.rq() se argument. Defaults to "rank" if the sample size is less than 1000, otherwise defaults to "nid".
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to quantreg::summary.rq().

Details

If se.type = "rank" confidence intervals are calculated by summary.rq and statistic and p.value values are not returned. When only a single predictor is included in the model, no confidence intervals are calculated and the confidence limits are set to NA.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

tidy.rqs 369

See Also

```
tidy(), quantreg::rq()
Other quantreg tidiers: augment.nlrq(), augment.rq(), augment.rqs(), glance.nlrq(), glance.rq(),
tidy.nlrq(), tidy.rqs()
```

Examples

```
# load modeling library and data
library(quantreg)
data(stackloss)
# median (11) regression fit for the stackloss data.
mod1 <- rq(stack.loss ~ stack.x, .5)</pre>
# weighted sample median
mod2 \leftarrow rq(rnorm(50) \sim 1, weights = runif(50))
# summarize model fit with tidiers
tidy(mod1)
glance(mod1)
augment(mod1)
tidy(mod2)
glance(mod2)
augment(mod2)
# varying tau to generate an rqs object
mod3 \leftarrow rq(stack.loss \sim stack.x, tau = c(.25, .5))
tidy(mod3)
augment(mod3)
# glance cannot handle rqs objects like `mod3`--use a purrr
# `map`-based workflow instead
```

tidy.rqs

Tidy a(n) rqs object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

370 tidy.rqs

Usage

```
## S3 method for class 'rqs'
tidy(x, se.type = "rank", conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

Х	An rqs object returned from quantreg::rq().
se.type	Character specifying the method to use to calculate standard errors. Passed to quantreg::summary.rq() se argument. Defaults to "rank".
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments passed to quantreg::summary.rqs()

Details

If se.type = "rank" confidence intervals are calculated by summary.rq. When only a single predictor is included in the model, no confidence intervals are calculated and the confidence limits are set to NA.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.
quantile	Linear conditional quantile.

See Also

```
tidy(), quantreg::rq()
Other quantreg tidiers: augment.nlrq(), augment.rq(), augment.rqs(), glance.nlrq(), glance.rq(),
tidy.nlrq(), tidy.rq()
```

tidy.sarlm 371

Examples

```
# load modeling library and data
library(quantreg)
data(stackloss)
# median (11) regression fit for the stackloss data.
mod1 <- rq(stack.loss ~ stack.x, .5)</pre>
# weighted sample median
mod2 <- rq(rnorm(50) \sim 1, weights = runif(50))
# summarize model fit with tidiers
tidy(mod1)
glance(mod1)
augment(mod1)
tidy(mod2)
glance(mod2)
augment(mod2)
# varying tau to generate an rqs object
mod3 \leftarrow rq(stack.loss \sim stack.x, tau = c(.25, .5))
tidy(mod3)
augment(mod3)
# glance cannot handle rqs objects like `mod3`--use a purrr
# `map`-based workflow instead
```

tidy.sarlm

Tidying methods for spatially autoregressive models

Description

These methods tidy the coefficients of spatial autoregression models generated by functions in the spatialreg package.

Usage

```
## S3 method for class 'sarlm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x An object returned from spatialreg::lagsarlm() or spatialreg::errorsarlm().

372 tidy.sarlm

conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

conf.level The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

... Additional arguments. Not used. Needed to match generic signature only. Cau-

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

Lower bound on the confidence interval for the estimate.

The estimated value of the regression term.

The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

The standard error of the regression term.

The name of the regression term.

See Also

```
tidy(), spatialreg::lagsarlm(), spatialreg::errorsarlm(), spatialreg::sacsarlm()
Other spatialreg tidiers: augment.sarlm(), glance.sarlm()
```

```
# load libraries for models and data
library(spatialreg)
library(spdep)

# load data
data(oldcol, package = "spdep")
listw <- nb2listw(COL.nb, style = "W")
# fit model</pre>
```

tidy.spec 373

```
crime_sar <-</pre>
  lagsarlm(CRIME ~ INC + HOVAL,
    data = COL.OLD,
    listw = listw,
    method = "eigen"
  )
# summarize model fit with tidiers
tidy(crime_sar)
tidy(crime_sar, conf.int = TRUE)
glance(crime_sar)
augment(crime_sar)
# fit another model
crime_sem <- errorsarlm(CRIME ~ INC + HOVAL, data = COL.OLD, listw)</pre>
# summarize model fit with tidiers
tidy(crime_sem)
tidy(crime_sem, conf.int = TRUE)
glance(crime_sem)
augment(crime_sem)
# fit another model
crime_sac <- sacsarlm(CRIME ~ INC + HOVAL, data = COL.OLD, listw)</pre>
# summarize model fit with tidiers
tidy(crime_sac)
tidy(crime_sac, conf.int = TRUE)
glance(crime_sac)
augment(crime_sac)
```

tidy.spec

Tidy a(n) spec object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'spec'
tidy(x, ...)
```

374 tidy.speedglm

Arguments

x A spec object created by stats::spectrum().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

freq Vector of frequencies at which the spectral density is estimated.

spec Vector (for univariate series) or matrix (for multivariate series) of estimates of

the spectral density at frequencies corresponding to freq.

See Also

```
tidy(), stats::spectrum()
Other time series tidiers: tidy.acf(), tidy.ts(), tidy.zoo()
```

Examples

```
spc <- spectrum(lh)
tidy(spc)

library(ggplot2)
ggplot(tidy(spc), aes(freq, spec)) +
  geom_line()</pre>
```

tidy.speedglm

Tidy a(n) speedglm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

tidy.speedglm 375

Usage

```
## S3 method for class 'speedglm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

A speedglm object returned from speedglm::speedglm(). Χ conf.int Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE. conf.level The confidence level to use for the confidence interval if conf. int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval. Logical indicating whether or not to exponentiate the the coefficient estimates. exponentiate This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE. Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
speedglm::speedglm()
Other speedlm tidiers: augment.speedlm(), glance.speedglm(), glance.speedlm(), tidy.speedlm()
```

376 tidy.speedlm

Examples

```
# load libraries for models and data
library(speedglm)

# generate data
clotting <- data.frame(
    u = c(5, 10, 15, 20, 30, 40, 60, 80, 100),
    lot1 = c(118, 58, 42, 35, 27, 25, 21, 19, 18)
)

# fit model
fit <- speedglm(lot1 ~ log(u), data = clotting, family = Gamma(log))
# summarize model fit with tidiers
tidy(fit)
glance(fit)</pre>
```

tidy.speedlm

Tidy a(n) speedlm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'speedlm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	A speedlm object returned from speedglm::speedlm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed

using conf.level = 0.95. Two exceptions here are:

tidy.summary.glht 377

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

estimate The estimated value of the regression term.

p.value The two-sided p-value associated with the observed statistic.

The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

std.error The standard error of the regression term.

The name of the regression term.

See Also

```
speedglm::speedlm(), tidy.lm()
Other speedlm tidiers: augment.speedlm(), glance.speedglm(), glance.speedlm(), tidy.speedglm()
```

Examples

```
# load modeling library
library(speedglm)

# fit model
mod <- speedlm(mpg ~ wt + qsec, data = mtcars, fitted = TRUE)

# summarize model fit with tidiers
tidy(mod)
glance(mod)
augment(mod)</pre>
```

tidy.summary.glht

Tidy a(n) summary.glht object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

378 tidy.summary.glht

Usage

```
## S3 method for class 'summary.glht'
tidy(x, ...)
```

Arguments

A summary.glht object created by calling multcomp::summary.glht() on a glht object created with multcomp::glht().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

contrast Levels being compared.

estimate The estimated value of the regression term.

null.value Value to which the estimate is compared.

p.value The two-sided p-value associated with the observed statistic.

statistic The value of a T-statistic to use in a hypothesis that the regression term is non-zero.

std.error The standard error of the regression term.

See Also

```
tidy(), multcomp::summary.glht(), multcomp::glht()
Other multcomp tidiers: tidy.cld(), tidy.confint.glht(), tidy.glht()
```

```
# load libraries for models and data
library(multcomp)
library(ggplot2)

amod <- aov(breaks ~ wool + tension, data = warpbreaks)
wht <- glht(amod, linfct = mcp(tension = "Tukey"))
tidy(wht)</pre>
```

tidy.summary.lm 379

```
ggplot(wht, aes(lhs, estimate)) +
   geom_point()

CI <- confint(wht)

tidy(CI)

ggplot(CI, aes(lhs, estimate, ymin = lwr, ymax = upr)) +
   geom_pointrange()

tidy(summary(wht))
ggplot(mapping = aes(lhs, estimate)) +
   geom_linerange(aes(ymin = lwr, ymax = upr), data = CI) +
   geom_point(aes(size = p), data = summary(wht)) +
   scale_size(trans = "reverse")

cld <- cld(wht)
tidy(cld)</pre>
```

tidy.summary.lm

Tidy a(n) summary.lm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'summary.lm'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

X	A summary.lm object created by stats::summary.lm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

380 tidy.summary.lm

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The tidy.summary.lm() method is a potentially useful alternative to tidy.lm(). For instance, if users have already converted large lm objects into their leaner summary.lm equivalents to conserve memory.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), stats::summary.lm()
Other Im tidiers: augment.glm(), augment.lm(), glance.glm(), glance.lm(), glance.summary.lm(),
glance.svyglm(), tidy.glm(), tidy.lm(), tidy.lm.beta(), tidy.mlm()
```

```
# fit model
mod <- lm(mpg ~ wt + qsec, data = mtcars)
modsumm <- summary(mod)

# summarize model fit with tidiers
tidy(mod, conf.int = TRUE)

# equivalent to the above
tidy(modsumm, conf.int = TRUE)
glance(mod)

# mostly the same, except for a few missing columns
glance(modsumm)</pre>
```

tidy.summary_emm 381

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'summary_emm'
tidy(x, null.value = NULL, ...)
```

Arguments

```
x A summary_emm object.

null.value Value to which estimate is compared.

... Additional arguments passed to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid().

Cautionary note: misspecified arguments may be silently ignored!
```

Details

Returns a data frame with one observation for each estimated marginal mean, and one column for each combination of factors. When the input is a contrast, each row will contain one estimated contrast.

There are a large number of arguments that can be passed on to emmeans::summary.emmGrid() or lsmeans::summary.ref.grid().

Value

level1

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate. contrast Levels being compared. den.df Degrees of freedom of the denominator. df Degrees of freedom used by this term in the model. null.value Value to which the estimate is compared. num.df Degrees of freedom. p.value The two-sided p-value associated with the observed statistic. std.error The standard error of the regression term.

One level of the factor being contrasted

382 tidy.summary_emm

```
level2The other level of the factor being contrastedtermModel term in joint testsestimateExpected marginal meanstatisticT-ratio statistic or F-ratio statistic
```

See Also

```
tidy(), emmeans::ref_grid(), emmeans::emmeans(), emmeans::contrast()
Other emmeans tidiers: tidy.emmGrid(), tidy.lsmobj(), tidy.ref.grid()
```

```
# load libraries for models and data
library(emmeans)
# linear model for sales of oranges per day
oranges_lm1 <- lm(sales1 ~ price1 + price2 + day + store, data = oranges)</pre>
# reference grid; see vignette("basics", package = "emmeans")
oranges_rg1 <- ref_grid(oranges_lm1)</pre>
td <- tidy(oranges_rg1)</pre>
td
# marginal averages
marginal <- emmeans(oranges_rg1, "day")</pre>
tidy(marginal)
# contrasts
tidy(contrast(marginal))
tidy(contrast(marginal, method = "pairwise"))
# plot confidence intervals
library(ggplot2)
ggplot(tidy(marginal, conf.int = TRUE), aes(day, estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# by multiple prices
by_price <- emmeans(oranges_lm1, "day",</pre>
  by = "price2",
  at = list(
    price1 = 50, price2 = c(40, 60, 80),
    day = c("2", "3", "4")
by_price
tidy(by_price)
```

tidy.survdiff 383

```
ggplot(tidy(by_price, conf.int = TRUE), aes(price2, estimate, color = day)) +
  geom_line() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high))
# joint_tests
tidy(joint_tests(oranges_lm1))
```

tidy.survdiff

Tidy a(n) survdiff object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'survdiff' tidy(x, ...)
```

Arguments

x An survdiff object returned from survival::survdiff().

.. Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

exp Weighted expected number of events in each group.

Number of subjects in each group.

obs weighted observed number of events in each group.

384 tidy.survexp

See Also

```
tidy(), survival::survdiff()
Other survdiff tidiers: glance.survdiff()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survexp(),
tidy.survfit(), tidy.survreg()
```

Examples

```
# load libraries for models and data
library(survival)

# fit model
s <- survdiff(
   Surv(time, status) ~ pat.karno + strata(inst),
   data = lung
)

# summarize model fit with tidiers
tidy(s)
glance(s)</pre>
```

tidy.survexp

Tidy a(n) survexp object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'survexp'
tidy(x, ...)
```

Arguments

. . .

x An survexp object returned from survival::survexp().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

tidy.survexp 385

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

n.risk Number of individuals at risk at time zero.

time Point in time.
estimate Estimate survival

See Also

```
tidy(), survival::survexp()
Other survexp tidiers: glance.survexp()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(),
tidy.survfit(), tidy.survreg()
```

```
# load libraries for models and data
library(survival)

# fit model
sexpfit <- survexp(
  futime ~ 1,
  rmap = list(
    sex = "male",
    year = accept.dt,
    age = (accept.dt - birth.dt)
  ),
  method = "conditional",
  data = jasa
)

# summarize model fit with tidiers
tidy(sexpfit)
glance(sexpfit)</pre>
```

tidy.survfit 386

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'survfit'
tidy(x, ...)
```

Arguments

An survfit object returned from survival::survfit(). Χ

For glance.survfit(), additional arguments passed to summary(). Otherwise ignored.

Value

n.event

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate. Number of censored events. n.censor Number of events at time t.

n.risk Number of individuals at risk at time zero. std.error The standard error of the regression term.

Point in time. time

estimate of survival or cumulative incidence rate when multistate estimate

state state if multistate survfit object input strata if stratified survfit object input strata

See Also

```
tidy(), survival::survfit()
```

```
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(),
tidy.survexp(), tidy.survreg()
```

tidy.survreg 387

Examples

```
# load libraries for models and data
library(survival)
# fit model
cfit <- coxph(Surv(time, status) ~ age + sex, lung)</pre>
sfit <- survfit(cfit)</pre>
# summarize model fit with tidiers + visualization
tidy(sfit)
glance(sfit)
library(ggplot2)
ggplot(tidy(sfit), aes(time, estimate)) +
  geom_line() +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)
# multi-state
fitCI <- survfit(Surv(stop, status * as.numeric(event), type = "mstate") ~ 1,</pre>
  data = mgus1, subset = (start == 0)
td_multi <- tidy(fitCI)</pre>
td_multi
ggplot(td_multi, aes(time, estimate, group = state)) +
  geom_line(aes(color = state)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .25)
```

tidy.survreg

Tidy a(n) survreg object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'survreg'
tidy(x, conf.level = 0.95, conf.int = FALSE, ...)
```

388 tidy.survreg

Arguments

An survreg object returned from survival::survreg().

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.
p.value	The two-sided p-value associated with the observed statistic.
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.
std.error	The standard error of the regression term.
term	The name of the regression term.

See Also

```
tidy(), survival::survreg()
Other survreg tidiers: augment.survreg(), glance.survreg()
Other survival tidiers: augment.coxph(), augment.survreg(), glance.aareg(), glance.cch(),
glance.coxph(), glance.pyears(), glance.survdiff(), glance.survexp(), glance.survfit(),
glance.survreg(), tidy.aareg(), tidy.cch(), tidy.coxph(), tidy.pyears(), tidy.survdiff(),
tidy.survexp(), tidy.survfit()
```

```
# load libraries for models and data
library(survival)
```

tidy.svyglm 389

```
# fit model
sr <- survreg(</pre>
  Surv(futime, fustat) ~ ecog.ps + rx,
  ovarian,
  dist = "exponential"
)
# summarize model fit with tidiers + visualization
tidy(sr)
augment(sr, ovarian)
glance(sr)
# coefficient plot
td <- tidy(sr, conf.int = TRUE)</pre>
library(ggplot2)
ggplot(td, aes(estimate, term)) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0) +
  geom_vline(xintercept = 0)
```

tidy.svyglm

Tidy a(n) svyglm object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'svyglm'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

X	A svyglm object returned from survey::svyglm().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
exponentiate	Logical indicating whether or not to exponentiate the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.

390 tidy.svyolr

• • •

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

See Also

```
survey::svyglm(), stats::glm()
```

tidy.svyolr

Tidy a(n) svyolr object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'svyolr'
tidy(x, conf.int = FALSE, conf.level = 0.95, exponentiate = FALSE, ...)
```

Arguments

X	A svyolr object returned from survey::svyolr().	
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.	
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.	
exponentiate	Logical indicating whether or not to exponentiate the the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE.	
	Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed	

using conf.level = 0.95. Two exceptions here are:

tidy.svyolr 391

• tidy() methods will warn when supplied an exponentiate argument if it will be ignored.

• augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

The tidy.svyolr() tidier is a light wrapper around tidy.polr(). However, the implementation for p-value calculation in tidy.polr() is both computationally intensive and specific to that model, so the p.values argument to tidy.svyolr() is currently ignored, and will raise a warning when passed.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.	
conf.low	Lower bound on the confidence interval for the estimate.	
estimate	The estimated value of the regression term.	
p.value	The two-sided p-value associated with the observed statistic.	
statistic	The value of a T-statistic to use in a hypothesis that the regression term is non-zero.	
std.error	The standard error of the regression term.	
term	The name of the regression term.	

See Also

```
tidy, survey::svyolr()
Other ordinal tidiers: augment.clm(), augment.polr(), glance.clm(), glance.clmm(), glance.polr(),
glance.svyolr(), tidy.clm(), tidy.clmm(), tidy.polr()
```

```
library(broom)
library(survey)

data(api)
dclus1 <- svydesign(id = ~dnum, weights = ~pw, data = apiclus1, fpc = ~fpc)
dclus1 <- update(dclus1, mealcat = cut(meals, c(0, 25, 50, 75, 100)))

m <- svyolr(mealcat ~ avg.ed + mobility + stype, design = dclus1)

m

tidy(m, conf.int = TRUE)</pre>
```

392 tidy.systemfit

tidy.systemfit	Tidy a(n) systemfit object
----------------	----------------------------

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'systemfit'
tidy(x, conf.int = TRUE, conf.level = 0.95, ...)
```

Arguments

A systemfit object produced by a call to systemfit::systemfit().

Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.

The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

This tidy method works with any model objects of class systemfit. Default returns a tibble of six columns.

Value

A tibble::tibble() with columns:

conf.high	Upper bound on the confidence interval for the estimate.
conf.low	Lower bound on the confidence interval for the estimate.
estimate	The estimated value of the regression term.

tidy.table 393

 $p.\, value \qquad \qquad The \,\, two\text{-}sided \,\, p\text{-}value \,\, associated \,\, with \,\, the \,\, observed \,\, statistic.$

std.error The standard error of the regression term.

term The name of the regression term.

See Also

```
tidy(), systemfit::systemfit()
```

Examples

```
set.seed(27)
# load libraries for models and data
library(systemfit)

# generate data
df <- data.frame(
    X = rnorm(100),
    Y = rnorm(100),
    Z = rnorm(100),
    W = rnorm(100)
)

# fit model
fit <- systemfit(formula = list(Y ~ Z, W ~ X), data = df, method = "SUR")

# summarize model fit with tidiers
tidy(fit)
tidy(fit, conf.int = TRUE)</pre>
```

tidy.table

Tidy a(n) table object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Deprecated. Please use tibble::as_tibble() instead.

Usage

```
## S3 method for class 'table' tidy(x, ...)
```

394 tidy.ts

Arguments

x A base::table object.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

```
Directly calls tibble::as_tibble() on a base::table object.
```

Value

A tibble::tibble in long-form containing frequency information for the table in a Freq column. The result is much like what you get from tidyr::pivot_longer().

See Also

```
tibble::as_tibble.table()
```

tidy.ts

Tidy a(n) ts object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'ts' tidy(x, ...)
```

tidy.ts 395

Arguments

x A univariate or multivariate ts times series object.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

series column is only present for multivariate ts objects.

Value

```
A tibble::tibble() with columns:
```

index Index (i.e. date or time) for a 'ts' or 'zoo' object.

series Name of the series (present only for multivariate time series).

value The value/estimate of the component. Results from data reshaping.

See Also

```
tidy(), stats::ts()
Other time series tidiers: tidy.acf(), tidy.spec(), tidy.zoo()
```

```
set.seed(678)

tidy(ts(1:10, frequency = 4, start = c(1959, 2)))

z <- ts(matrix(rnorm(300), 100, 3), start = c(1961, 1), frequency = 12)
colnames(z) <- c("Aa", "Bb", "Cc")

tidy(z)</pre>
```

396 tidy.TukeyHSD

	tidy.TukeyHSD	Tidy a(n) TukeyHSD object
--	---------------	---------------------------

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'TukeyHSD'
tidy(x, ...)
```

Arguments

x A TukeyHSD object return from stats::TukeyHSD().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble() with columns:

```
adj.p.value P-value adjusted for multiple comparisons.

conf.high Upper bound on the confidence interval for the estimate.

conf.low Lower bound on the confidence interval for the estimate.

contrast Levels being compared.

estimate The estimated value of the regression term.

value to which the estimate is compared.

term The name of the regression term.
```

See Also

```
tidy(), stats::TukeyHSD()
Other anova tidiers: glance.anova(), glance.aov(), tidy.anova(), tidy.aov(), tidy.aov(), tidy.aov(), tidy.aov()
```

tidy.varest 397

Examples

```
fm1 <- aov(breaks ~ wool + tension, data = warpbreaks)
thsd <- TukeyHSD(fm1, "tension", ordered = TRUE)
tidy(thsd)

# may include comparisons on multiple terms
fm2 <- aov(mpg ~ as.factor(gear) * as.factor(cyl), data = mtcars)
tidy(TukeyHSD(fm2))</pre>
```

tidy.varest

Tidy a(n) varest object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

Usage

```
## S3 method for class 'varest'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

x	A varest object produced by a call to vars::VAR().
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	The confidence level to use for the confidence interval if conf.int = TRUE. Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
	For glance(), additional arguments passed to summary(). Otherwise ignored.

Details

The tibble has one row for each term in the regression. The component column indicates whether a particular term was used to model either the "mean" or "precision". Here the precision is the inverse of the variance, often referred to as phi. At least one term will have been used to model the precision phi.

The vars package does not include a confint method and does not report confidence intervals for varest objects. Setting the tidy argument conf.int = TRUE will return a warning.

398 tidy.zoo

Value

A tibble::tibble() with columns:

conf.high Upper bound on the confidence interval for the estimate. conf.low Lower bound on the confidence interval for the estimate. estimate The estimated value of the regression term. The two-sided p-value associated with the observed statistic. p.value statistic The value of a T-statistic to use in a hypothesis that the regression term is nonzero. The standard error of the regression term. std.error The name of the regression term. term component Whether a particular term was used to model the mean or the precision in the regression. See details.

See Also

```
tidy(), vars::VAR()
```

Examples

```
# load libraries for models and data
library(vars)

# load data
data("Canada", package = "vars")

# fit models
mod <- VAR(Canada, p = 1, type = "both")

# summarize model fit with tidiers
tidy(mod)
glance(mod)</pre>
```

tidy.zoo

Tidy a(n) zoo object

Description

Tidy summarizes information about the components of a model. A model component might be a single term in a regression, a single hypothesis, a cluster, or a class. Exactly what tidy considers to be a model component varies across models but is usually self-evident. If a model has several distinct types of components, you will need to specify which components to return.

tidy.zoo 399

Usage

```
## S3 method for class 'zoo'
tidy(x, ...)
```

Arguments

A zoo object such as those created by zoo::zoo().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

```
A tibble::tibble() with columns:
```

index Index (i.e. date or time) for a 'ts' or 'zoo' object.

series Name of the series (present only for multivariate time series).

value The value/estimate of the component. Results from data reshaping.

See Also

```
tidy(), zoo::zoo()
Other time series tidiers: tidy.acf(), tidy.spec(), tidy.ts()
```

Examples

```
# load libraries for models and data
library(zoo)
library(ggplot2)

set.seed(1071)

# generate data
Z.index <- as.Date(sample(12450:12500, 10))
Z.data <- matrix(rnorm(30), ncol = 3)
colnames(Z.data) <- c("Aa", "Bb", "Cc")
Z <- zoo(Z.data, Z.index)

# summarize model fit with tidiers + visualization
tidy(Z)
ggplot(tidy(Z), aes(index, value, color = series)) +</pre>
```

400 tidy_irlba

```
geom_line()
ggplot(tidy(Z), aes(index, value)) +
  geom_line() +
  facet_wrap(~series, ncol = 1)

Zrolled <- rollmean(Z, 5)
ggplot(tidy(Zrolled), aes(index, value, color = series)) +
  geom_line()</pre>
```

tidy_irlba

Tidy a(n) irlba object masquerading as list

Description

Broom tidies a number of lists that are effectively S3 objects without a class attribute. For example, stats::optim(), svd() and interp::interp() produce consistent output, but because they do not have a class attribute, they cannot be handled by S3 dispatch.

These functions look at the elements of a list and determine if there is an appropriate tidying method to apply to the list. Those tidiers are implemented as functions of the form tidy_<function> or glance_<function> and are not exported (but they are documented!).

If no appropriate tidying method is found, they throw an error.

Usage

```
tidy_irlba(x, ...)
```

Arguments

Х

A list returned from irlba::irlba().

... Additional arguments. Not used. Needed to match generic signature only. Cautionary note: Misspelled arguments will be absorbed in ..., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

A very thin wrapper around tidy_svd().

tidy_irlba 401

Value

A tibble::tibble with columns depending on the component of PCA being tidied.

If matrix is "u", "samples", "scores", or "x" each row in the tidied output corresponds to the original data in PCA space. The columns are:

row ID of the original observation (i.e. rowname from original data).

PC Integer indicating a principal component.

value The score of the observation for that particular principal component. That is, the

location of the observation in PCA space.

If matrix is "v", "rotation", "loadings" or "variables", each row in the tidied output corresponds to information about the principle components in the original space. The columns are:

row The variable labels (colnames) of the data set on which PCA was performed.

PC An integer vector indicating the principal component.

value The value of the eigenvector (axis score) on the indicated principal component.

If matrix is "d", "eigenvalues" or "pcs", the columns are:

PC An integer vector indicating the principal component.

std.dev Standard deviation explained by this PC.

percent Fraction of variation explained by this component (a numeric value between 0

and 1).

cumulative Cumulative fraction of variation explained by principle components up to this

component (a numeric value between 0 and 1).

See Also

```
tidy(), irlba::irlba()
Other list tidiers: glance_optim(), list_tidiers, tidy_optim(), tidy_svd(), tidy_xyz()
Other svd tidiers: augment.prcomp(), tidy.prcomp(), tidy_svd()
```

Examples

```
library(modeldata)
data(hpc_data)

mat <- scale(as.matrix(hpc_data[, 2:5]))
s <- svd(mat)

tidy_u <- tidy(s, matrix = "u")
tidy_u

tidy_d <- tidy(s, matrix = "d")
tidy_d</pre>
```

402 tidy_optim

```
tidy_v <- tidy(s, matrix = "v")
tidy_v

library(ggplot2)
library(dplyr)

ggplot(tidy_d, aes(PC, percent)) +
  geom_point() +
  ylab("% of variance explained")

tidy_u %>%
  mutate(class = hpc_data$class[row]) %>%
  ggplot(aes(class, value)) +
  geom_boxplot() +
  facet_wrap(~PC, scale = "free_y")
```

tidy_optim

Tidy a(n) optim object masquerading as list

Description

Broom tidies a number of lists that are effectively S3 objects without a class attribute. For example, stats::optim(), svd() and interp::interp() produce consistent output, but because they do not have a class attribute, they cannot be handled by S3 dispatch.

These functions look at the elements of a list and determine if there is an appropriate tidying method to apply to the list. Those tidiers are implemented as functions of the form tidy_<function> or glance_<function> and are not exported (but they are documented!).

If no appropriate tidying method is found, they throw an error.

Usage

```
tidy_optim(x, ...)
```

Arguments

X

A list returned from stats::optim().

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

tidy_svd 403

Value

A tibble::tibble() with columns:

parameter The parameter being modeled.

std.error The standard error of the regression term.

value The value/estimate of the component. Results from data reshaping.

std.error is only provided as a column if the Hessian is calculated.

Note

This function assumes that the provided objective function is a negative log-likelihood function. Results will be invalid if an incorrect function is supplied.

```
tidy(o) glance(o)
```

See Also

```
tidy(), stats::optim()
Other list tidiers: glance_optim(), list_tidiers, tidy_irlba(), tidy_svd(), tidy_xyz()
```

Examples

```
f <- function(x) (x[1] - 2)^2 + (x[2] - 3)^2 + (x[3] - 8)^2
o <- optim(c(1, 1, 1), f)
```

tidy_svd

Tidy a(n) svd object masquerading as list

Description

Broom tidies a number of lists that are effectively S3 objects without a class attribute. For example, stats::optim(), svd() and interp::interp() produce consistent output, but because they do not have a class attribute, they cannot be handled by S3 dispatch.

These functions look at the elements of a list and determine if there is an appropriate tidying method to apply to the list. Those tidiers are implemented as functions of the form tidy_<function> or glance_<function> and are not exported (but they are documented!).

If no appropriate tidying method is found, they throw an error.

Usage

```
tidy_svd(x, matrix = "u", ...)
```

404 tidy_svd

Arguments

x A list with components u, d, v returned by base::svd().

matrix Character specifying which component of the PCA should be tidied.

• "u", "samples", "scores", or "x": returns information about the map from the original space into principle components space.

- "v", "rotation", "loadings" or "variables": returns information about the map from principle components space back into the original space.
- "d", "eigenvalues" or "pcs": returns information about the eigenvalues.

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . ., where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Details

See https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca for information on how to interpret the various tidied matrices. Note that SVD is only equivalent to PCA on centered data.

Value

A tibble::tibble with columns depending on the component of PCA being tidied.

If matrix is "u", "samples", "scores", or "x" each row in the tidied output corresponds to the original data in PCA space. The columns are:

row ID of the original observation (i.e. rowname from original data).

PC Integer indicating a principal component.

value The score of the observation for that particular principal component. That is, the

location of the observation in PCA space.

If matrix is "v", "rotation", "loadings" or "variables", each row in the tidied output corresponds to information about the principle components in the original space. The columns are:

row The variable labels (colnames) of the data set on which PCA was performed.

PC An integer vector indicating the principal component.

value The value of the eigenvector (axis score) on the indicated principal component.

If matrix is "d", "eigenvalues" or "pcs", the columns are:

PC An integer vector indicating the principal component.

std.dev Standard deviation explained by this PC.

tidy_xyz 405

percent Fraction of variation explained by this component (a numeric value between 0

and 1).

cumulative Cumulative fraction of variation explained by principle components up to this

component (a numeric value between 0 and 1).

See Also

```
base::svd()
Other svd tidiers: augment.prcomp(), tidy.prcomp(), tidy_irlba()
Other list tidiers: glance_optim(), list_tidiers, tidy_irlba(), tidy_optim(), tidy_xyz()
```

Examples

```
library(modeldata)
data(hpc_data)
mat <- scale(as.matrix(hpc_data[, 2:5]))</pre>
s <- svd(mat)
tidy_u <- tidy(s, matrix = "u")</pre>
tidy_u
tidy_d <- tidy(s, matrix = "d")</pre>
tidy_d
tidy_v <- tidy(s, matrix = "v")</pre>
tidy_v
library(ggplot2)
library(dplyr)
ggplot(tidy_d, aes(PC, percent)) +
  geom_point() +
  ylab("% of variance explained")
tidy_u %>%
  mutate(class = hpc_data$class[row]) %>%
  ggplot(aes(class, value)) +
  geom_boxplot() +
  facet_wrap(~PC, scale = "free_y")
```

406 tidy_xyz

Description

Broom tidies a number of lists that are effectively S3 objects without a class attribute. For example, stats::optim(), svd() and interp::interp() produce consistent output, but because they do not have a class attribute, they cannot be handled by S3 dispatch.

These functions look at the elements of a list and determine if there is an appropriate tidying method to apply to the list. Those tidiers are implemented as functions of the form tidy_<function> or glance_<function> and are not exported (but they are documented!).

If no appropriate tidying method is found, they throw an error.

xyz lists (lists where x and y are vectors of coordinates and z is a matrix of values) are typically used by functions such as graphics::persp() or graphics::image() and returned by interpolation functions such as interp::interp().

Usage

```
tidy_xyz(x, ...)
```

Arguments

Х

A list with component x, y and z, where x and y are vectors and z is a matrix. The length of x must equal the number of rows in z and the length of y must equal the number of columns in z.

. . .

Additional arguments. Not used. Needed to match generic signature only. **Cautionary note:** Misspelled arguments will be absorbed in . . . , where they will be ignored. If the misspelled argument has a default value, the default value will be used. For example, if you pass conf.lvel = 0.9, all computation will proceed using conf.level = 0.95. Two exceptions here are:

- tidy() methods will warn when supplied an exponentiate argument if it will be ignored.
- augment() methods will warn when supplied a newdata argument if it will be ignored.

Value

A tibble::tibble with vector columns x, y and z.

See Also

```
tidy(), graphics::persp(), graphics::image(), interp::interp()
Other list tidiers: glance_optim(), list_tidiers, tidy_irlba(), tidy_optim(), tidy_svd()
```

Examples

```
A <- list(x = 1:5, y = 1:3, z = matrix(runif(5 * 3), nrow = 5)) image(A) tidy(A)
```

Index

* Arima tidiers	glance.coxph, 122
glance.Arima, 107	tidy.coxph, 256
tidy.Arima, 231	* crr tidiers
* aareg tidiers	glance.crr, 124
glance.aareg, 102	* decompose tidiers
tidy.aareg, 225	augment.decomposed.ts, 20
* anova tidiers	augment.stl, 91
glance.anova, 104	* deprecated
glance.aov, 105	bootstrap, 96
tidy.anova, 227	<pre>confint_tidy, 97</pre>
tidy.aov, 229	data.frame_tidiers, 98
tidy.aovlist, 230	finish_glance, 101
tidy.manova, 316	<pre>fix_data_frame, 102</pre>
tidy.TukeyHSD,396	summary_tidiers, 223
* betareg tidiers	tidy.density, 262
tidy.betareg, 234	tidy.dist, 263
* biglm tidiers	tidy.ftable, 278
glance.biglm, 111	tidy.numeric, 341
tidy.biglm, 236	* drc tidiers
* bingroup tidiers	augment.drc, 22
glance.binDesign, 113	glance.drc,128
tidy.binDesign,238	tidy.drc, 264
tidy.binWidth,239	* emmeans tidiers
* car tidiers	tidy.emmGrid, 265
durbinWatsonTest_tidiers, 100	tidy.lsmobj, 314
leveneTest_tidiers, 220	tidy.ref.grid,359
* cch tidiers	tidy.summary_emm,381
glance.cch, 114	* epiR tidiers
glance.survfit, 210	tidy.epi.2by2,267
tidy.cch, 244	* ergm tidiers
* cmprsk tidiers	glance.ergm, 129
glance.crr, 124	tidy.ergm, 269
tidy.crr, 258	* factanal tidiers
* coeftest tidiers	augment.factanal, 25
tidy.coeftest,251	glance.factanal, 131
* coeftest_tidiers	tidy.factanal,271
glance.coeftest, 120	* felm tidiers
* coxph tidiers	augment.felm, 26
augment.coxph, 17	tidy.felm, 272

* fitdistr tidiers	* lm tidiers
glance.fitdistr,134	augment.glm, 33
tidy.fitdistr,274	augment.lm,44
* fixest tidiers	glance.glm, 143
augment.fixest,28	glance.lm, 156
tidy.fixest, 276	glance.summary.lm, 204
* gam tidiers	glance.svyglm, 214
glance.Gam, 138	tidy.glm, 286
tidy.Gam, 278	tidy.lm, 306
* garch tidiers	tidy.lm.beta,308
glance.garch, 141	tidy.mlm, 331
tidy.garch,282	tidy.summary.lm,379
* geepack tidiers	* lmodel2 tidiers
glance.geeglm, 142	glance.lmodel2,158
* glm.nb tidiers	tidy.lmodel2,310
glance.negbin, 176	* margins tidiers
tidy.negbin,337	tidy.margins, 319
* glmnet tidiers	* mclust tidiers
glance.cv.glmnet, 126	augment.Mclust, 54
glance.glmnet, 145	tidy.Mclust, 321
tidy.cv.glmnet,260	* mediate tidiers
tidy.glmnet,287	tidy.mediate, 323
* gmm tidiers	* mfx tidiers
glance.gmm, 148	augment.betamfx, 10
tidy.gmm, 292	augment.mfx, 56
* htest tidiers	glance.betamfx, 108
augment.htest, 38	glance.mfx, 167
tidy.htest,294	tidy.betamfx, 232
tidy.pairwise.htest, 343	tidy.mfx, 325
tidy.power.htest, 352	* mgcv tidiers
* ivreg tidiers	glance.gam, 139
augment.ivreg, 40	tidy.gam, 280
glance.ivreg, 150	* mjoint tidiers
tidy.ivreg, 296	glance.mjoint, 169
* kmeans tidiers	tidy.mjoint, 327
augment.kmeans,42	* mlogit tidiers
glance.kmeans, 152	augment. $mlogit, 62$
tidy.kmeans, 303	glance.mlogit, 171
* lavaan tidiers	tidy.mlogit, 333
glance.lavaan, 154	* muhaz tidiers
tidy.lavaan, 304	glance.muhaz, 173
* list tidiers	tidy.muhaz,334
<pre>glance_optim, 219</pre>	* multcomp tidiers
list_tidiers, 221	tidy.cld, 245
tidy_irlba,400	tidy.confint.glht, 253
tidy_optim, 402	tidy.glht, 285
tidy_svd, 403	tidy.summary.glht,377
tidy xyz, 405	* multinom tidiers

glance.multinom, 174	augment.rlm,77
tidy.multinom, 335	glance.rlm, 192
* nls tidiers	tidy.rlm, 363
augment.nls,65	* robust tidiers
glance.nls, 179	augment.lmRob,48
tidy.nls, 339	glance.glmRob, 146
* orcutt tidiers	glance.lmRob, 160
glance.orcutt, 180	tidy.glmRob, 289
tidy.orcutt, 342	tidy.1mRob, 312
* ordinal tidiers	* robustbase tidiers
augment.clm, 15	augment.glmrob,36
augment.polr, 73	augment.lmrob, 50
glance.clm, 116	glance.lmrob, 162
glance.clmm, 118	tidy.glmrob, 290
glance.polr, 187	tidy.lmrob, 313
glance.svyolr, 216	* smoothing spline tidiers
tidy.clm, 247	augment.smooth.spline, 88
tidy.clmm, 249	glance.smooth.spline, 200
tidy.polr, 350	* spatialreg tidiers
tidy.svyolr, 390	augment.sarlm, 86
* pam tidiers	glance.sarlm, 198
augment.pam, 67	tidy.sarlm, 371
glance.pam, 182	* speedlm tidiers
tidy.pam, 344	augment.speedlm, 89
* plm tidiers	glance.speedglm, 201
augment.plm, 69	glance.speedlm, 203
glance.plm, 183	tidy.speedglm, 374
tidy.plm, 346	tidy.speedlm, 376
* poLCA tidiers	* survdiff tidiers
augment.poLCA, 71	glance.survdiff, 207
glance.poLCA, 185	tidy.survdiff, 383
tidy.poLCA, 348	* survexp tidiers
* pyears tidiers	glance.survexp, 209
glance.pyears, 189	tidy.survexp, 384
tidy.pyears, 355	* survey tidiers
* quantreg tidiers	tidy.svyglm, 389
augment.nlrq, 64	* survfit tidiers
augment.rq, 81	tidy.survfit, 386
augment.rqs, 83	* survival tidiers
glance.nlrq, 177	augment.coxph, 17
glance.rq, 196	augment.survreg, 92
tidy.nlrq, 338	glance.aareg, 102
tidy.rq, 368	glance.cch, 114
tidy.rqs, 369	glance.coxph, 122
* ridgelm tidiers	glance.pyears, 189
glance.ridgelm, 191	glance.survdiff, 207
tidy.ridgelm, 362	glance.survexp, 209
* rlm tidiers	glance.survfit, 210
· · · · · · · · · · · · · · · · · · ·	5 - ance . 5 an vi 1 c, 210

glance.survreg, 212	augment.drc, 22, 129, 263
tidy.aareg, 225	augment.factanal, 25, <i>132</i> , <i>271</i>
tidy.cch, 244	augment.felm, 26, 274
tidy.coxph, 256	augment.fixest, 28, 277
tidy.pyears, 355	augment.gam, 31
tidy.survdiff,383	augment.glm, 33, 46, 144, 157, 205, 215, 287
tidy.survexp,384	307, 309, 332, 380
tidy.survfit,386	augment.glm(), 58, 59
tidy.survreg,387	augment.glmRob,35
* survreg tidiers	augment.glmrob, 36, 51, 163, 291, 314
augment.survreg, 92	augment.htest, 38, 295, 344, 353
glance.survreg, 212	augment.ivreg, 40, 152, 297
tidy.survreg, 387	augment.kmeans, 42, 153, 303
* svd tidiers	augment.1m, 35, 44, 144, 157, 205, 215, 287,
augment.prcomp, 75	307, 309, 332, 380
tidy.prcomp, 353	augment.lmRob, 48, <i>147</i> , <i>161</i> , <i>290</i> , <i>312</i>
tidy_irlba,400	augment. 1mrob, 38, 50, 163, 291, 314
tidy_svd, 403	augment.loess, 52
* systemfit tidiers	augment.logitmfx (augment.mfx), 56
tidy.systemfit,392	augment.Mclust, 54, 322
* time series tidiers	augment.mfx, 12, 56, 109, 169, 233, 326
tidy.acf, 226	augment.mioint, 59
tidy.spec, 373	
tidy.ts, 394	augment.mlogit, 62, 172, 334 augment.negbinmfx (augment.mfx), 56
tidy.zoo, 398	
* vars tidiers	augment.nlrq, 64, 83, 85, 178, 197, 339, 369
tidy.varest, 397	370
	augment.nls, 65, 180, 340
<pre>aareg_tidiers (tidy.aareg), 225</pre>	augment.NULL (null_tidiers), 222
AER::ivreg(), 41, 42, 151, 152, 296, 297	augment.pam, 67, 183, 345
aer_tidiers(tidy.ivreg),296	augment.plm, 69, 185, 347
Arima_tidiers(tidy.Arima), 231	augment.poissonmfx (augment.mfx), 56
AUC::roc(), 366, 367	augment.poLCA, 71, 186, 348
auc_tidiers(tidy.roc), 366	augment.polr, 16, 73, 117, 119, 189, 217,
augment, <i>83</i> , <i>85</i>	248, 250, 351, 391
augment(), 15, 19, 21, 24, 26, 27, 30, 32, 40,	augment.prcomp, 75, 354, 401, 405
42, 44, 46, 53, 55, 63, 64, 68, 70, 72,	augment.probitmfx(augment.mfx), 56
87, 89, 92, 94, 95, 200	augment.rlm, 77, 193, 364
augment.betamfx, 10, 59, 109, 169, 233, 326	augment.rma,79
augment.betareg, 13	augment.rq, 64, 81, 85, 178, 197, 339, 369,
<pre>augment.betareg(), 12</pre>	370
augment.clm, 15, 75, 117, 119, 189, 217, 248, 250, 351, 391	augment.rqs, 64, 83, 83, 178, 197, 339, 369, 370
augment.coxph, 17, 94, 103, 116, 123, 190,	augment.sarlm, 86, 199, 372
208, 210, 211, 213, 226, 245, 257,	augment.smooth.spline, 88, 200
356, 384–386, 388	augment.speedlm, 89, 202, 204, 375, 377
augment.data.frame	augment.stl, 21, 91
(data.frame_tidiers), 98	augment.survreg, 19, 92, 103, 116, 123, 190
augment.decomposed.ts, 20, 92	208, 210, 211, 213, 226, 245, 257,
	,,,,,,,,,,

356, 384–386, 388	confint_tidy, 96, 97, 99, 101, 102, 224, 262
augment_columns, 95	263, 278, 341
	confusionMatrix_tidiers
base::data.frame, 11, 14, 16, 18, 23, 26, 27,	(tidy.confusionMatrix), 254
29, 32, 34, 37, 41, 43, 45, 49, 50, 52,	<pre>coxph_tidiers (tidy.coxph), 256</pre>
54, 58, 60, 64, 66, 68, 70, 71, 74, 76,	
<i>78</i> , <i>82</i> , <i>84</i> , <i>88</i> , <i>90</i> , <i>93</i>	data.frame_tidiers, 96, 97, 98, 101, 102,
base::data.frame(), 12, 14, 16, 18, 23, 29,	224, 262, 263, 278, 341
32, 34, 37, 41, 45, 49, 50, 52, 58, 64,	decompose_tidiers
66, 74, 76, 78, 82, 84, 90, 94	(augment.decomposed.ts), 20
base::svd(), 221, 404, 405	drc::drm(), 23, 24, 128, 129, 264, 265
base::table, <i>394</i>	drc_tidiers (tidy.drc), 264
bbmle::mle2(), 330	durbinWatsonTest_tidiers, 100, 220
bbmle_tidiers(tidy.mle2), 329	ua: 52:a550555_524265, 1555, 22 5
betareg::betareg(), 14, 15, 110, 111, 234,	emmeans::contrast(), 266, 315, 360, 382
235	emmeans::emmeans(), 266, 315, 360, 382
betareg::predict.betareg(), 12	emmeans::ref_grid(), 266, 315, 359, 360,
betareg::residuals.betareg(), 12	382
betareg_tidiers (tidy.betareg), 234	emmeans::summary.emmGrid(), 266, 315,
biglm::bigglm(), 112, 236, 237	359, 381
biglm::biglm(), 112, 236, 237 biglm::biglm(), 112, 236, 237	emmeans_tidiers(tidy.lsmobj), 314
bindesign_tidiers (tidy.binDesign), 238	epiR::epi.2by2(), 268
binGroup::binDesign, 113	epiR_tidiers (tidy.epi.2by2), 267
binGroup::binDesign(), 114, 238	ergm::control.ergm(), 270
binGroup::binWidth(), 239, 240	ergm::ergm(), 130, 269, 270
binwidth_tidiers (tidy.binWidth), 239	ergm::summary(), 130, 270
boot::boot(), 240, 241	ergm::summary.ergm(), 130
boot::boot.ci(), 241	ergm_tidiers(tidy.ergm), 269
boot::tsboot(), 241	0
boot_tidiers(tidy.boot), 240	factanal_tidiers(tidy.factanal), 271
bootstrap, 96, 97, 99, 101, 102, 224, 262,	felm_tidiers(tidy.felm), 272
263, 278, 341	finish_glance, 96, 97, 99, 101, 102, 224,
btergm::btergm(), 242, 243	262, 263, 278, 341
btergm_tidiers (tidy.btergm), 242	fitdistr_tidiers(tidy.fitdistr),274
	fix_data_frame, 96, 97, 99, 101, 102, 224,
car::Anova(), 104, 227, 228	262, 263, 278, 341
<pre>car::durbinWatsonTest(), 100</pre>	fixest::feglm(), <i>30</i> , <i>277</i>
car::leveneTest(), 104, 220, 227-230	fixest::femlm(), 30, 277
car::linearHypothesis(), 104, 227	fixest::fenegbin(), 30, 277
<pre>caret::confusionMatrix(), 255</pre>	fixest::feNmlm(), 30, 277
<pre>caret_tidiers(tidy.confusionMatrix),</pre>	fixest::feols(), 30, 277
254	fixest::fepois(), 30, 277
cch_tidiers(tidy.cch), 244	
cfa_tidiers(tidy.lavaan), 304	gam::gam(), 138, 139, 279
cluster::pam(), 68, 182, 183, 345	Gam_tidiers(tidy.Gam), 278
cmprsk::crr(), 124, 125, 258, 259	gam_tidiers(tidy.gam), 280
cmprsk_tidiers(tidy.crr), 258	garch_tidiers(tidy.garch), 282
coeftest_tidiers (tidy.coeftest), 251	<pre>geeglm_tidiers (tidy.geeglm), 283</pre>
confint(), 97	geepack::geeglm(), 142, 143, 283, 284

<pre>geepack_tidiers(tidy.geeglm), 283</pre>	glance.glm, 35, 46, 143, 157, 205, 215, 287,
glance(), 100, 103, 105, 106, 111, 112, 114,	307, 309, 332, 380
116, 121, 123, 125, 127, 129, 130,	glance.glm(), 168, 169
132, 139, 140, 142, 143, 146, 149,	glance.glmnet, 127, 145, 260, 288
152, 153, 155, 157, 159, 170, 172,	glance.glmRob, 49, 146, 161, 290, 312
174, 175, 177, 178, 181, 183, 185,	glance.gmm, 148, 293
186, 190, 192, 193, 197, 199, 205,	glance.htest (tidy.htest), 294
208, 210, 211, 213, 218–220, 294	
glance.aareg, 19, 94, 102, 116, 123, 190,	glance.ivreg, 42, 150, 297
	glance.kmeans, 44, 152, 303
208, 210, 211, 213, 226, 245, 257,	glance.lavaan, 154, 305
356, 384–386, 388	glance.list(list_tidiers), 221
glance.anova, 104, 106, 228, 229, 231, 317,	glance.lm, 35, 46, 144, 156, 205, 215, 287,
396	307, 309, 332, 380
glance.aov, 105, 105, 228, 229, 231, 317, 396	glance.lm(), 205
glance.Arima, 107, 232	glance.lmodel2, 158, 311
glance.betamfx, 12, 59, 108, 169, 233, 326	glance.lmRob, 49, 147, 160, 290, 312
glance.betareg, 110	glance.lmrob, 38, 51, 162, 291, 314
glance.betareg(), 109	glance.logitmfx(glance.mfx), 167
glance.biglm, 111, 237	glance.margins, 163
glance.binDesign, 113, 238, 240	glance.Mclust, 165
glance.cch, 19, 94, 103, 114, 123, 190, 208,	glance.mfx, 12, 59, 109, 167, 233, 326
210, 211, 213, 226, 245, 257, 356,	glance.mjoint, 169, 328
384–386, 388	glance.mlogit, 63, 171, 334
glance.clm, 16, 75, 116, 119, 189, 217, 248,	glance.miogic, 65, 171, 554 glance.muhaz, 173, 335
250, 351, 391	_
	glance.multinom, 174, 336
glance.clmm, 16, 75, 117, 118, 189, 217, 248,	glance.negbin, 176, 337
250, 351, 391	glance.negbinmfx(glance.mfx), 167
glance.coeftest, 120	glance.nlrq, 64, 83, 85, 177, 197, 339, 369,
glance.coxph, 19, 94, 103, 116, 122, 190,	370
208, 210, 211, 213, 226, 245, 257,	glance.nls, 66, 179, 340
356, 384–386, 388	glance.NULL (null_tidiers), 222
glance.crr, 124, 259	<pre>glance.optim (glance_optim), 219</pre>
glance.cv.glmnet, 126, 146, 260, 288	glance.orcutt, 180, <i>343</i>
<pre>glance.data.frame (data.frame_tidiers),</pre>	glance.pam, 68, 182, 345
98	glance.plm, 70, 183, 347
glance.drc, 24, 128, 265	glance.poissonmfx(glance.mfx), 167
glance.durbinWatsonTest	glance.poLCA, 72, 185, 348
(durbinWatsonTest_tidiers), 100	glance.polr, 16, 75, 117, 119, 187, 217, 248,
glance.ergm, 129, 270	250, 351, 391
glance.factanal, 26, 131, 271	glance.probitmfx (glance.mfx), 167
glance.felm, 133	glance.pyears, 19, 94, 103, 116, 123, 189,
glance.fitdistr, 134, 275	208, 210, 211, 213, 226, 245, 257,
glance.fixest, 136	356, 384–386, 388
glance.Gam, 138, 279	
	glance.ridgelm, 191, 363
glance.gam, 139, 281	glance.rlm, 78, 192, 364
glance.gam(), 138	glance.rma, 194
glance.garch, 141, 283	glance.rq, 64, 83, 85, 178, 196, 339, 369, 370
glance.geeglm, 142	glance.sarlm, 87, 198, 372

glance.smooth.spline, 89, 200	<pre>joinerml_tidiers(tidy.mjoint), 327</pre>
glance.speedglm, 91, 201, 204, 375, 377	
glance.speedlm, 91, 202, 203, 375, 377	kappa_tidiers(tidy.kappa),298
glance.summary.lm, 35, 46, 144, 157, 204,	kde_tidiers(tidy.kde),300
215, 287, 307, 309, 332, 380	Kendall::Kendall(), 302
glance.summary.lm(), 157, 205	Kendall::MannKendall(), 302
glance.summaryDefault	Kendall::SeasonalMannKendall(), 302
(summary_tidiers), 223	Kendall_tidiers(tidy.Kendall), 301
glance.survdiff, 19, 94, 103, 116, 123, 190,	kendall_tidiers(tidy.Kendall), 301
207, 210, 211, 213, 226, 245, 257,	kmeans_tidiers (tidy.kmeans), 303
356, 384–386, 388	ks::kde(), 300
glance.survexp, 19, 94, 103, 116, 123, 190,	ks_tidiers (tidy.kde), 300
-	RS_traters (tray. Rae), 500
208, 209, 211, 213, 226, 245, 257,	lavaan::cfa(), <i>154</i> , <i>155</i> , <i>304</i> , <i>305</i>
356, 384–386, 388	lavaan::fitmeasures(), <i>155</i>
glance.survfit, 19, 94, 103, 116, 123, 190,	lavaan::parameterEstimates(), 304, 305
208, 210, 210, 213, 226, 245, 257,	
356, 384–386, 388	lavaan::sem(), 154, 155, 304, 305
glance.survreg, 19, 94, 103, 116, 123, 190,	lavaan_tidiers (tidy.lavaan), 304
208, 210, 211, 212, 226, 245, 257,	leaps::regsubsets(), 361
356, 384–386, 388	leaps_tidiers (tidy.regsubsets), 361
glance.svyglm, 35, 46, 144, 157, 205, 214,	leveneTest_tidiers, 100, 220
287, 307, 309, 332, 380	lfe::felm(), 27, 133, 273, 274
glance.svyolr, 16, 75, 117, 119, 189, 216,	lfe_tidiers (tidy.felm), 272
248, 250, 351, 391	list_tidiers, 219, 221, 401, 403, 405, 406
glance.varest, 217	lm.beta::lm.beta, 309
glance_optim, 219, 221, 401, 403, 405, 406	lm_tidiers (tidy.lm), 306
glm.nb_tidiers(glance.negbin), 176	lmodel2::lmodel2(), 159, 310, 311
glmnet::cv.glmnet(), <i>126</i> , <i>127</i> , <i>260</i>	<pre>lmodel2_tidiers(tidy.lmodel2), 310</pre>
glmnet::glmnet(), 145, 146, 288	<pre>lmtest::coeftest(), 120, 121, 251, 252</pre>
<pre>glmnet_tidiers(tidy.glmnet), 287</pre>	<pre>lmtest_tidiers(tidy.coeftest), 251</pre>
gmm::gmm(), 148, 149, 292, 293	loess_tidiers(augment.loess), 52
<pre>gmm_tidiers(tidy.gmm), 292</pre>	lsmeans::summary.ref.grid(), 266, 315,
graphics::image(), 406	359, 381
graphics::persp(), 406	
	maps::map(), 318
Hmisc::rcorr(), 357, 358	<pre>maps_tidiers(tidy.map), 318</pre>
<pre>Hmisc_tidiers (tidy.rcorr), 357</pre>	margins::margins(), <i>164</i> , <i>319</i> , <i>320</i>
htest_tidiers (tidy.htest), 294	margins_tidiers(tidy.margins), 319
	MASS::dropterm(), 350
interp::interp(), 219, 221, 400, 402, 403,	MASS::fitdistr(), <i>135</i> , <i>275</i>
406	MASS::glm.nb(), <i>176</i> , <i>177</i> , <i>337</i>
irlba::irlba(), 400, 401	MASS::lm.ridge(), <i>191</i> , <i>192</i> , <i>362</i> , <i>363</i>
irlba_tidiers (tidy_irlba), 400	MASS::polr(), 74, 75, 188, 189, 350, 351
<pre>ivreg_tidiers (tidy.ivreg), 296</pre>	MASS::rlm(), 78, 193, 364
	MASS::select.ridgelm(), 192
<pre>joineRML::bootSE(), 328</pre>	mclust::Mclust(), 54, 55, 166, 322
<pre>joineRML::fitted.mjoint(), 60</pre>	mclust_tidiers(tidy.Mclust), 321
joineRML::mjoint(), 60, 170, 327, 328	mean, 98
<pre>joineRML::residuals.mjoint(),60</pre>	<pre>mediate_tidiers (tidy.mediate), 323</pre>

mediation::mediate(), <i>323</i> , <i>324</i>	<pre>prcomp_tidiers(tidy.prcomp), 353</pre>
metafor::escalc(), 365	predict.fixest, 29
metafor::rma(), 80, 194, 365	psych::cohen.kappa(), 298, 299
metafor::rma.glmm(), 80, 194, 365	psych_tidiers(tidy.kappa), 298
metafor::rma.mh(), 80, 194, 365	purrr::map(), 196
metafor::rma.mv(), 80, 194, 365	purrr::map_df(), 222
metafor::rma.peto(), 80, 194, 365	<pre>pyears_tidiers (tidy.pyears), 355</pre>
metafor::rma.uni(), 80, 194, 365	, , , , , , , , , , , , , , , , , , , ,
mfx::betamfx(), 12, 109, 233	qr, <i>317</i>
mfx::logitmfx(), 59, 169, 326	quantreg::nlrq(), 64, 178, 338, 339
mfx::negbinmfx(), 59, 169, 326	quantreg::predict.rq, 82, 84
mfx::poissonmfx(), 59, 169, 326	quantreg::predict.rq(),83
mfx::probitmfx(), 59, 169, 326	quantreg::predict.rqs(), 85
mgcv::gam(), 32, 138, 140, 279–281	quantreg::rq(), 81, 83-85, 196, 197,
mgcv_tidiers (tidy.gam), 280	368–370
mjoint_tidiers (tidy.mjoint), 327	quantreg::summary.rq(), 368, 370
mle2_tidiers (tidy.mle2), 329	quantreg::summary.rqs(), 370
mlogit::mlogit(), 63, 172, 333, 334	quantreg_tidiers (tidy.rq), 368
mlogit_tidiers (tidy.mlogit), 333	, , , , , , , , , , , , , , , , , , , ,
muhaz::muhaz(), 173, 174, 334, 335	<pre>rcorr_tidiers (tidy.rcorr), 357</pre>
muhaz_tidiers (tidy.muhaz), 334	ridgelm_tidiers(tidy.ridgelm), 362
multcomp::cld(), 246	rlm_tidiers(glance.rlm), 192
multcomp::confint.glht(), 246, 253	robust::glmRob(), <i>147</i> , <i>289</i> , <i>290</i>
multcomp::glht(), 246, 253, 285, 286, 378	robust::1mRob(), 49, 161, 312
multcomp::summary.glht(), 246, 378	<pre>robust_tidiers(tidy.lmRob), 312</pre>
multcomp_tidiers (tidy.glht), 285	robustbase::glmrob(), 37, 38, 291
multinom_tidiers (tidy.multinom), 335	robustbase::lmrob(), 50, 51, 162, 163, 313
nlrq_tidiers(tidy.nlrq), 338	<pre>robustbase_tidiers(tidy.lmrob), 313</pre>
nls_tidiers (tidy.nls), 339	roc_tidiers (tidy.roc), 366
nnet::multinom(), 175, 335, 336	rq_tidiers (tidy.rq), 368
nnet_tidiers (tidy.multinom), 335	rqs_tidiers (tidy.rqs), 369
null_tidiers, 222	rsample::bootstraps(), 241
Tid11_tid101 0, 222	,
optim_tidiers(tidy_optim), 402	sem_tidiers(tidy.lavaan),304
orcutt::cochrane.orcutt(), 181, 342, 343	<pre>sexpfit_tidiers(tidy.survexp), 384</pre>
orcutt_tidiers(tidy.orcutt),342	<pre>smooth.spline_tidiers</pre>
ordinal::clm(), 16, 117, 247, 248	(augment.smooth.spline), 88
ordinal::clmm(), 119, 249, 250	sp_tidiers, 222
ordinal::confint.clm(), 248, 250	spatialreg::errorsarlm(), 86, 198, 199,
ordinal::predict.clm(), 16	371, 372
ordinal_tidiers(tidy.clm), 247	spatialreg::lagsarlm(), 86, 198, 199, 371
<pre>pam_tidiers(tidy.pam), 344</pre>	spatialreg::sacsarlm(), 199, 372
plm::plm(), 70, 184, 185, 346, 347	spatialreg_tidiers(tidy.sarlm), 371
plm_tidiers (tidy.plm), 346	speedglm::speedglm(), 201, 375
poLCA::poLCA(), 71, 72, 186, 348	speedglm::speedlm(), 90, 91, 202-204, 376
poLCA_tidiers (tidy.poLCA), 348	377
polr_tidiers (tidy.polr), 350	<pre>speedglm_tidiers(tidy.speedglm), 374</pre>

<pre>speedlm_tidiers (tidy.speedlm), 376</pre>	stats::summary.lm(), 307, 379, 380
splines::ns(), 11, 13, 16, 18, 21, 23, 25, 27,	stats::summary.manova, 317
29, 31, 33, 35, 36, 39, 41, 43, 45, 48,	stats::summary.manova(), 317
50, 54, 56, 60, 62, 65, 67, 69, 71, 74,	stats::summary.nls(), 340
76, 77, 79, 81, 84, 86, 90, 91, 93	stats::t.test(), 39, 295
stats::acf(), 226, 227	stats::ts(), <i>395</i>
stats::anova(), 104, 227, 228	stats::TukeyHSD(), 396
stats::aov(), 106, 229-231	stats::wilcox.test(), 39, 295
stats::arima(), 107, 108, 231, 232	summary(), 224, 285, 337, 386, 397
stats::ccf(), 226, 227	summary.fixest, 30, 136, 276
stats::chisq.test(), 39, 40, 295	summary_tidiers, 96, 97, 99, 101, 102, 223,
stats::cooks.distance(), 14, 32	262, 263, 278, 341
stats::cor.test(), 39, 295	<pre>survdiff_tidiers(tidy.survdiff), 383</pre>
stats::decompose(), 21	<pre>survexp_tidiers(tidy.survexp), 384</pre>
stats::density(), 262	survey::anova.svyglm,215
stats::dist(), 263	survey::svyglm(), 214, 215, 389, 390
stats::factanal(), 25, 26, 131, 132, 271	survey::svyolr(), 216, 217, 390, 391
stats::ftable(), 278	<pre>survfit_tidiers(tidy.survfit), 386</pre>
stats::glm(), 34, 35, 144, 215, 287, 390	survival::aareg(), 103, 225, 226
stats::kmeans(), 43, 44, 153, 303	survival::cch(), 115, 116, 244, 245
stats::lm(), 26, 45, 156, 205, 306, 331	survival::coxph(), 18, 19, 122, 123, 256,
stats::loess(), <i>52</i> , <i>53</i>	257
stats::manova(), <i>317</i>	survival::pyears(), <i>190</i> , <i>356</i>
stats::na.action, <i>19</i> , <i>46</i> , <i>53</i>	<pre>survival::summary.survfit(), 211</pre>
stats::nls(), 66, 179, 180, 340	survival::Surv(), 11, 13, 16, 18, 21, 23, 25
stats::optim(), 219, 221, 400, 402, 403, 406	27, 29, 31, 33, 35, 36, 39, 41, 43, 45
stats::pacf(), 226, 227	48, 50, 54, 56, 60, 62, 65, 67, 69, 71
stats::pairwise.t.test(), <i>343</i> , <i>344</i>	74, 76, 77, 79, 81, 84, 86, 90, 91, 93
stats::pairwise.wilcox.test(), 343, 344	survival::survdiff(), 208, 383, 384
stats::poly(), 11, 13, 16, 18, 21, 23, 25, 27,	survival::survexp(), 209, 210, 384, 385
29, 31, 33, 35, 36, 39, 41, 43, 45, 48,	survival::survfit(), 211, 386
50, 54, 56, 60, 62, 65, 67, 69, 71, 74,	survival::survreg(), 93, 94, 213, 388
76, 77, 79, 81, 84, 86, 90, 91, 93	<pre>survreg_tidiers(tidy.survreg), 387</pre>
stats::power.t.test(), <i>352</i> , <i>353</i>	svd(), 219, 400, 402, 403, 406
stats::prcomp(), 76, 77, 353, 354	svd_tidiers, <i>77</i> , <i>354</i>
stats::predict(), 14, 18, 32, 37, 94	<pre>svd_tidiers(tidy_svd), 403</pre>
stats::predict.glm(), 34,58	<pre>svyolr_tidiers (tidy.svyolr), 390</pre>
stats::predict.lm(),46	<pre>systemfit::systemfit(), 392, 393</pre>
stats::predict.loess(),53	<pre>systemfit_tidiers(tidy.systemfit), 392</pre>
stats::predict.nls(), 66	
stats::predict.smooth.spline(), 89	tibble::as_tibble(), 278, 393, 394
stats::residuals(), <i>14</i> , <i>18</i> , <i>32</i> , <i>37</i> , <i>94</i>	tibble::as_tibble.table(), 394
stats::residuals.glm(), 34 , 58	tibble::tibble, 11, 13, 16, 17, 20, 21, 23,
stats::rstandard.glm(), 34 , 58	25, 27, 29, 31, 33, 35, 36, 39, 41, 43
stats::smooth.spline(), 88, 89, 200	44, 48, 50, 54, 56, 60, 62, 65, 67, 69
stats::spectrum(), 374	71, 74, 76, 77, 79, 81, 84, 86, 90–93
stats::stl(), 92	154, 222, 224, 262, 263, 270, 278,
stats::summary.aov(), 229	354, 394, 401, 404, 406

	224
tibble::tibble(), 11, 12, 14, 16, 18, 19, 23,	tidy.betareg, 234
24, 26, 27, 29, 30, 32, 34, 35, 37, 39,	tidy.betareg(), 233
41–43, 45, 46, 49–55, 58, 60, 61, 63,	tidy.biglm, <i>112</i> , 236
64, 66, 68, 70–72, 74, 76, 78, 80, 82,	tidy.binDesign, <i>114</i> ,238,240
84, 87–90, 93, 94, 100, 102–115,	tidy.binWidth, <i>114</i> , <i>238</i> , 239
117–120, 122–126, 128, 129, 131,	tidy.boot, 240
133, 135–154, 156, 159–186,	tidy.btergm, 242
188–198, 200–205, 207–220, 225,	tidy.cch, 19, 94, 103, 116, 123, 190, 208,
227, 228, 230, 232, 233, 235,	210, 211, 213, 226, 244, 257, 356,
237–239, 241, 243, 244, 246, 248,	<i>384–386, 388</i>
250, 252, 253, 255, 257, 259, 260,	tidy.character(tidy.numeric), 341
265, 266, 268, 271, 273, 275, 277,	tidy.cld, 245, 253, 286, 378
279, 281, 282, 284, 285, 288, 291,	tidy.clm, 16, 75, 117, 119, 189, 217, 247,
293, 295, 297, 299, 300, 302, 303,	250, 351, 391
305, 306, 309, 311, 315, 317, 318,	tidy.clmm, 16, 75, 117, 119, 189, 217, 248,
320, 322, 324, 326, 328, 330, 332,	249, 351, 391
333, 335, 336, 338, 340, 342, 344,	tidy.coeftest, 251
345, 347, 348, 351, 352, 356, 358,	tidy.confint.glht, 246, 253, 286, 378
359, 361, 362, 365, 367, 368, 370,	tidy.confusionMatrix, 254
372, 374, 375, 377, 378, 380, 381,	•
383, 385, 386, 388, 391, 392, 395,	tidy.coxph, 19, 94, 103, 116, 123, 190, 208,
396, 398, 399, 403	210, 211, 213, 226, 245, 256, 356,
tidy, 16, 66, 117, 119, 180, 189, 217, 248,	384–386, 388
250, 340, 351, 391	tidy.crr, 125, 258
tidy(), 75, 100, 135, 220, 226–229, 231, 235,	tidy.cv.glmnet, 127, 146, 260, 288
237, 238, 240, 241, 243, 245, 246,	tidy.data.frame(data.frame_tidiers), 98
252, 253, 255, 257, 259, 260, 265,	tidy.density, 96, 97, 99, 101, 102, 224, 262,
266, 268, 270, 271, 274, 275, 277,	263, 278, 341
279, 281, 283, 284, 286, 288,	tidy.dist, 96, 97, 99, 101, 102, 224, 262,
293–295, 297, 299, 300, 302, 303,	263, 278, 341
305, 307, 311, 315, 317, 318, 320,	tidy.drc, 24, 129, 264
322, 324, 326, 328, 330, 332,	tidy.durbinWatsonTest
334–336, 339, 344, 345, 347, 348,	(durbinWatsonTest_tidiers), 100
	tidy.emmGrid, 265, 315, 360, 382
356, 358, 360, 361, 363, 367, 369,	tidy.epi.2by2, 267
<i>370, 372, 374, 378, 380, 382,</i>	tidy.ergm, <i>130</i> , 269
<i>384–386</i> , <i>388</i> , <i>393</i> , <i>395</i> , <i>396</i> , <i>398</i> ,	tidy.factanal, 26, 132, 271
399, 401, 403, 406	tidy.felm, 27, 272
tidy.aareg, 19, 94, 103, 116, 123, 190, 208,	tidy.fitdistr, <i>135</i> , 274
210, 211, 213, 225, 245, 257, 356,	tidy.fixest, 30, 276
384–386, 388	tidy.ftable, 96, 97, 99, 101, 102, 224, 262,
tidy.acf, 226, <i>374</i> , <i>395</i> , <i>399</i>	263, 278, 341
tidy.anova, 105, 106, 227, 229, 231, 317, 396	tidy.Gam, <i>139</i> , 278
tidy.anova(), 279	tidy.gam, <i>140</i> , 280
tidy.aov, 105, 106, 228, 229, 231, 317, 396	tidy.gam(), 279
tidy.aovlist, 105, 106, 228, 229, 230, 317,	tidy.garch, 142, 282
396	tidy.geeglm, 283
tidy.Arima, <i>108</i> , 231	tidy.glht, 246, 253, 285, 378
tidy.betamfx, 12, 59, 109, 169, 232, 326	tidy.glm, 250, 253, 285, 576 tidy.glm, 35, 46, 144, 157, 205, 215, 286,
cray. De cami A, 12, 57, 107, 107, 252, 520	cray. griii, 33, 70, 177, 137, 203, 213, 200,

307, 309, 332, 380	tidy.nls, 66, 180, 339
tidy.glmnet, 127, 146, 260, 287	tidy.NULL(null_tidiers),222
tidy.glmRob, 49, 147, 161, 289, 312	tidy.numeric, 96, 97, 99, 101, 102, 224, 262,
tidy.glmrob, 38, 51, 163, 290, 314	263, 278, 341
tidy.gmm, <i>149</i> , 292	tidy.optim(tidy_optim), 402
tidy.htest, 40, 294, 344, 353	tidy.orcutt, <i>181</i> , 342
tidy.irlba(tidy_irlba), 400	tidy.pairwise.htest, 40, 295, 343, 353
tidy.ivreg, 42, 152, 296	tidy.pam, 68, 183, 344
tidy.kappa, 298	tidy.plm, 70, 185, 346
tidy.kde, 300	tidy.poissonmfx(tidy.mfx), 325
tidy.Kendall, 301	tidy.poLCA, 72, 186, 348
tidy.kmeans, 44, 153, 303	tidy.polr, 16, 75, 117, 119, 189, 217, 248,
tidy.lavaan, 155, 304	250, 350, 391
tidy.leveneTest (leveneTest_tidiers),	tidy.polr(), <i>391</i>
220	tidy.Polygon(sp_tidiers),222
tidy.leveneTest(), $228-230$	tidy.Polygons(sp_tidiers),222
tidy.Line(sp_tidiers), 222	tidy.power.htest, 40, 295, 344, 352
tidy.Lines(sp_tidiers), 222	tidy.prcomp, 77, 353, 401, 405
tidy.list(list_tidiers), 221	tidy.probitmfx(tidy.mfx), 325
tidy.lm, 35, 46, 144, 157, 205, 215, 287, 306,	tidy.pyears, 19, 94, 103, 116, 123, 190, 208,
309, 332, 380	210, 211, 213, 226, 245, 257, 355,
tidy.lm(), 347, 377, 380	<i>384–386</i> , <i>388</i>
tidy.lm.beta, 35, 46, 144, 157, 205, 215,	tidy.rcorr, 357
287, 307, 308, 332, 380	tidy.ref.grid, 266, 315, 359, 382
tidy.lmodel2, <i>159</i> , 310	tidy.regsubsets,361
tidy.lmRob, 49, 147, 161, 290, 312	tidy.ridgelm, <i>192</i> , 362
tidy.lmrob, 38, 51, 163, 291, 313	tidy.rlm, 78, 193, 363
tidy.logical(tidy.numeric),341	tidy.rlm(), 37, 49, 51, 162, 290, 291, 312,
tidy.logitmfx(tidy.mfx), 325	314
tidy.lsmobj, 266, 314, 360, 382	tidy.rma,364
tidy.manova, 105, 106, 228, 229, 231, 316,	tidy.roc, 366
396	tidy.rq, 64, 83, 85, 178, 197, 339, 368, 370
tidy.map, 318	tidy.rqs, 64, 83, 85, 178, 197, 339, 369, 369
tidy.margins, 319	tidy.sarlm, <i>87</i> , <i>199</i> , 371
tidy.Mclust, <i>55</i> , 321	tidy.SpatialLinesDataFrame
tidy.mediate, 323	(sp_tidiers), 222
tidy.mfx, 12, 59, 109, 169, 233, 325	tidy.SpatialPolygons(sp_tidiers), 222
tidy.mjoint, <i>170</i> , 327	tidy.SpatialPolygonsDataFrame
tidy.mle2, 329	(sp_tidiers), 222
tidy.mlm, 35, 46, 144, 157, 205, 215, 287,	tidy.spec, 227, 373, 395, 399
307, 309, 331, 380	tidy.speedglm, 91, 202, 204, 374, 377
tidy.mlm(), 306	tidy.speedlm, 91, 202, 204, 375, 376
tidy.mlogit, 63, 172, 333	tidy.summary.glht, 246, 253, 286, 377
tidy.muhaz, <i>174</i> , 334	tidy.summary.glht(), 285
tidy.multinom, <i>175</i> , 335	tidy.summary.lm, 35, 46, 144, 157, 205, 215,
tidy.negbin, <i>177</i> , 337	287, 307, 309, 332, 379
tidy.negbinmfx(tidy.mfx), 325	tidy.summary_emm, 266, 315, 360, 381
tidy.nlrq, 64, 83, 85, 178, 197, 338, 369, 370	<pre>tidy.summaryDefault(summary_tidiers),</pre>

```
tidy.survdiff, 19, 94, 103, 116, 123, 190,
         208, 210, 211, 213, 226, 245, 257,
         356, 383, 385, 386, 388
tidy.survexp, 19, 94, 103, 116, 123, 190,
         208, 210, 211, 213, 226, 245, 257,
         356, 384, 384, 386, 388
tidy.survfit, 19, 94, 103, 116, 123, 190,
         208, 210, 211, 213, 226, 245, 257,
         356, 384, 385, 386, 388
tidy.survreg, 19, 94, 103, 116, 123, 190,
         208, 210, 211, 213, 226, 245, 257,
         356, 384–386, 387
tidy.svyglm, 389
tidy.svyolr, 16, 75, 117, 119, 189, 217, 248,
         250, 351, 390
tidy.systemfit, 392
tidy.table, 393
tidy.ts, 227, 374, 394, 399
tidy. TukeyHSD, 105, 106, 228, 229, 231, 317,
         396
tidy.varest, 397
tidy.zoo, 227, 374, 395, 398
tidy_irlba, 77, 219, 221, 354, 400, 403, 405,
         406
tidy_optim, 219, 221, 401, 402, 405, 406
tidy_optim(), 330
tidy_svd, 77, 219, 221, 354, 401, 403, 403,
         406
tidy_svd(), 400
tidy_xyz, 219, 221, 401, 403, 405, 405
tidyr::pivot_longer(), 394
tseries::garch(), 141, 142, 282, 283
vars::VAR(), 218, 397, 398
vars_tidiers(tidy.varest), 397
xyz_tidiers (tidy_xyz), 405
zoo::zoo(), 399
zoo_tidiers (tidy.zoo), 398
```