Package 'distfromq'

September 13, 2024

Title Reconstruct a Distribution from a Collection of Quantiles

Version 1.0.4

Description Given a set of predictive quantiles from a distribution, estimate the distribution and create `d`, `p`, `q`, and `r` functions to evaluate its density function, distribution function, and quantile function, and generate random samples. On the interior of the provided quantiles, an interpolation method such as a monotonic cubic spline is used; the tails are approximated by a location-scale family.

License GPL (>= 3)

URL http://reichlab.io/distfromq/

Imports checkmate, purrr, splines, stats, utils, zeallot

Suggests dplyr, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Evan Ray [aut, cre],
Aaron Gerding [aut],
Li Shandross [ctb],
Nick Reich [ctb]

Maintainer Evan Ray <elray@umass.edu>

Repository CRAN

Date/Publication 2024-09-13 18:00:06 UTC

Contents

`duplicated_tol` *Identify duplicated values in a sorted numeric vector, where comparison is up to a specified numeric tolerance. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.*

Description

Identify duplicated values in a sorted numeric vector, where comparison is up to a specified numeric tolerance. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.

Usage

```
duplicated_tol(x, tol = 1e-06, incl_first = FALSE)
```

Arguments

`x`	a numeric vector in which to identify duplicates
`tol`	numeric tolerance for identifying duplicates
`incl_first`	boolean indicator of whether or not the first entry in a run of duplicates should be indicated as a duplicate. `FALSE` mirrors the behavior of the base R function `duplicated`.

Value

a boolean vector of the same length as `x`

`get_dup_run_inds` *Get indices of starts and ends of runs of duplicate values*

Description

Get indices of starts and ends of runs of duplicate values

Usage

```
get_dup_run_inds(dups)
```

Arguments

`dups`	a boolean vector that would result from calling `duplicated_tol(..., incl_first = FALSE)`

Value

named list with entries `starts` giving indices of the first element in each sequence of runs of duplicate values and `ends` giving indices of the last element in each sequence of runs of duplicate values.

`make_d_fn` *Creates a function that evaluates the probability density function of an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.*

Description

Creates a function that evaluates the probability density function of an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.

Usage

```
make_d_fn(
  ps,
  qs,
  interior_method = "spline_cdf",
  interior_args = list(),
  tail_dist = "norm",
  dup_tol = 1e-06,
  zero_tol = 1e-12
)
```

Arguments

ps	vector of probability levels
qs	vector of quantile values corresponding to ps
interior_method	method for interpolating the distribution on the interior of the provided qs. This package provides one method for this, "spline_cdf". The user may also provide a custom function; see the details for more.
interior_args	an optional named list of arguments that are passed on to the interior_method
tail_dist	name of parametric distribution for the tails
dup_tol	numeric tolerance for identifying duplicated values indicating a discrete component of the distribution. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.
zero_tol	numeric tolerance for identifying values in qs that are (approximately) zero.

Details

The default interior_method, "spline_cdf", represents the distribution as a sum of a discrete component at any points where there are duplicated qs for multiple different ps and a continuous component that is estimated by using a monotonic cubic spline that interpolates the provided (q, p) pairs as an estimate of the CDF. The density function is then obtained by differentiating this estimate of the CDF.

Optionally, the user may provide another function that accepts arguments ps, qs, tail_dist, and fn_type (which will be either "d", "p", or "q"), and optionally additional named arguments to be specified via interior_args. This function should return a function with arguments x, log that evaluates the pdf or its logarithm.

Value

a function with arguments x and log that can be used to evaluate the approximate density function (or its log) at the points x.

---

make_p_fn *Creates a function that evaluates the cumulative distribution function of an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.*

Description

Creates a function that evaluates the cumulative distribution function of an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.

Usage

```
make_p_fn(
  ps,
  qs,
  interior_method = "spline_cdf",
  interior_args = list(),
  tail_dist = "norm",
  dup_tol = 1e-06,
  zero_tol = 1e-12
)
```

Arguments

ps	vector of probability levels
qs	vector of quantile values corresponding to ps
interior_method	method for interpolating the distribution on the interior of the provided qs. This package provides one method for this, "spline_cdf". The user may also provide a custom function; see the details for more.
interior_args	an optional named list of arguments that are passed on to the interior_method
tail_dist	name of parametric distribution for the tails
dup_tol	numeric tolerance for identifying duplicated values indicating a discrete component of the distribution. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.
zero_tol	numeric tolerance for identifying values in qs that are (approximately) zero.

Details

The default interior_method, "spline_cdf", represents the distribution as a sum of a discrete component at any points where there are duplicated qs for multiple different ps and a continuous component that is estimated by using a monotonic cubic spline that interpolates the provided (q, p) pairs as an estimate of the CDF.

Optionally, the user may provide another function that accepts arguments ps, qs, tail_dist, and fn_type (which will be either "d", "p", or "q"), and optionally additional named arguments to be specified via interior_args. This function should return a function with arguments x, log that evaluates the pdf or its logarithm.

Value

a function with arguments q and log.p that can be used to evaluate the approximate cumulative distribution function (or its log) at the points q.

make_q_fn *Creates a function that evaluates the quantile function of an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.*

Description

Creates a function that evaluates the quantile function of an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.

Usage

```
make_q_fn(
  ps,
  qs,
  interior_method = "spline_cdf",
  interior_args = list(),
  tail_dist = "norm",
  dup_tol = 1e-06,
  zero_tol = 1e-12
)
```

Arguments

ps	vector of probability levels
qs	vector of quantile values corresponding to ps
interior_method	method for interpolating the distribution on the interior of the provided qs. This package provides one method for this, "spline_cdf". The user may also provide a custom function; see the details for more.
interior_args	an optional named list of arguments that are passed on to the interior_method
tail_dist	name of parametric distribution for the tails
dup_tol	numeric tolerance for identifying duplicated values indicating a discrete component of the distribution. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.
zero_tol	numeric tolerance for identifying values in qs that are (approximately) zero.

Details

The default interior_method, "spline_cdf", represents the distribution as a sum of a discrete component at any points where there are duplicated qs for multiple different ps and a continuous component that is estimated by using a monotonic cubic spline that interpolates the provided (q, p) pairs as an estimate of the CDF. The quantile function is then obtained by inverting this estimate of the CDF.

Optionally, the user may provide another function that accepts arguments ps, qs, tail_dist, and fn_type (which will be either "d", "p", or "q"), and optionally additional named arguments to be specified via interior_args. This function should return a function with argument p that evaluates the quantile function.

Value

a function with argument p that can be used to calculate quantiles of the approximated distribution at the probability levels p.

---

make_r_fn *Creates a function that generates random deviates from an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.*

Description

Creates a function that generates random deviates from an approximation to a distribution obtained by interpolating and extrapolating from a set of quantiles of the distribution.

Usage

```
make_r_fn(
  ps,
  qs,
  interior_method = "spline_cdf",
  interior_args = list(),
  tail_dist = "norm",
  dup_tol = 1e-06,
  zero_tol = 1e-12
)
```

Arguments

ps	vector of probability levels
qs	vector of quantile values corresponding to ps
interior_method	method for interpolating the distribution on the interior of the provided qs. This package provides one method for this, "spline_cdf". The user may also provide a custom function; see the details for more.
interior_args	an optional named list of arguments that are passed on to the interior_method
tail_dist	name of parametric distribution for the tails
dup_tol	numeric tolerance for identifying duplicated values indicating a discrete component of the distribution. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.
zero_tol	numeric tolerance for identifying values in qs that are (approximately) zero.

Details

The default `interior_method`, `"spline_cdf"`, represents the distribution as a sum of a discrete component at any points where there are duplicated `qs` for multiple different `ps` and a continuous component that is estimated by using a monotonic cubic spline that interpolates the provided `(q, p)` pairs as an estimate of the CDF. The quantile function is then obtained by inverting this estimate of the CDF.

Optionally, the user may provide another function that accepts arguments `ps`, `qs`, `tail_dist`, and `fn_type` (which will be either `"d"`, `"p"`, or `"q"`), and optionally additional named arguments to be specified via `interior_args`. This function should return a function with argument `p` that evaluates the quantile function.

Value

a function with argument `n` that can be used to generate a sample of size `n` from the approximated distribution.

`mono_Hermite_spline` *Create a polySpline object representing a monotonic Hermite spline interpolating a given set of points.*

Description

Create a polySpline object representing a monotonic Hermite spline interpolating a given set of points.

Usage

```
mono_Hermite_spline(x, y, m)
```

Arguments

`x`	vector giving the x coordinates of the points to be interpolated.
`y`	vector giving the y coordinates of the points to be interpolated. Must be increasing or decreasing for 'method = "hyman"'.
`m`	(for 'splinefunH()') vector of *slopes* m_i at the points (x_i, y_i); these together determine the *H*ermite "spline" which is piecewise cubic, (only) *once* differentiable continuously.

Details

This function essentially reproduces `stats::splinefunH`, but it returns a polynomial spline object as used in the `splines` package rather than a function that evaluates the spline, and potentially makes adjustments to the input slopes `m` to enforce monotonicity.

Value

An object of class `polySpline` with the spline object, suitable for use with other functionality from the `splines` package.

spline_cdf *Approximate density function, CDF, or quantile function on the interior of provided quantiles by representing the distribution as a sum of a discrete part at any duplicated* qs *and a continuous part for which the CDF is estimated using a monotonic Hermite spline. See details for more.*

Description

Approximate density function, CDF, or quantile function on the interior of provided quantiles by representing the distribution as a sum of a discrete part at any duplicated qs and a continuous part for which the CDF is estimated using a monotonic Hermite spline. See details for more.

Usage

```
spline_cdf(ps, qs, tail_dist, fn_type = c("d", "p", "q"), n_grid = 20)
```

Arguments

ps	vector of probability levels
qs	vector of quantile values corresponding to ps
tail_dist	name of parametric distribution for the tails
fn_type	the type of function that is requested: "d" for a PDF, "p" for a CDF, or "q" for a quantile function.
n_grid	grid size to use when augmenting the input qs to obtain a finer grid of points along which we form a piecewise linear approximation to the spline. n_grid evenly-spaced points are inserted between each pair of consecutive values in qs. The default value is 20. This can be set to NULL, in which case the piecewise linear approximation is not used. This is not recommended if the fn_type is "q".

Details

The CDF of the continuous part of the distribution is estimated using a monotonic degree 3 Hermite spline that interpolates the quantiles after subtracting the discrete distribution and renormalizing. In theory, an estimate of the quantile function could be obtained by directly inverting this spline. However, in practice, we have observed that this can suffer from numerical problems. Therefore, the default behavior of this function is to evaluate the "stage 1" CDF estimate corresponding to discrete point masses plus monotonic spline at a fine grid of points, and use the "stage 2" CDF estimate that linearly interpolates these points with steps at any duplicated q values. The quantile function estimate is obtained by inverting this "stage 2" CDF estimate. When the distribution is continuous, we can obtain an estimate of the PDF by differentiating the CDF estimate, resulting in a discontinuous "histogram density". The size of the grid can be specified with the n_grid argument. In settings where it is desirable to obtain a continuous density function, the "stage 1" CDF estimate can be used by setting n_grid = NULL.

Value

a function to evaluate the PDF, CDF, or quantile function.

`split_disc_cont_ps_qs` *Split ps and qs into those corresponding to discrete and continuous parts of a distribution.*

Description

Split ps and qs into those corresponding to discrete and continuous parts of a distribution.

Usage

```
split_disc_cont_ps_qs(
  ps,
  qs,
  dup_tol = 1e-06,
  zero_tol = 1e-12,
  is_hurdle = FALSE
)
```

Arguments

`ps`	vector of probability levels
`qs`	vector of quantile values corresponding to ps
`dup_tol`	numeric tolerance for identifying duplicated values indicating a discrete component of the distribution. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as duplicates even if not all values in the run are within the tolerance.
`zero_tol`	numeric tolerance for identifying values in `qs` that are (approximately) zero.
`is_hurdle`	boolean indicating whether or not this is a hurdle model. If so, qs of zero always indicate the presence of a point mass at 0. In this case, 0 is not included among the returned `cont_qs`. Setting this argument to `TRUE` is primarily appropriate when we are working with a distributional family that is bounded above 0 (and may have density 0 at 0) such as a lognormal.

Value

named list with the following entries:

- `disc_weight`: estimated numeric weight of the discrete part of the distribution.
- `disc_ps`: estimated probabilities of discrete components. May be `numeric(0)` if there are no estimated discrete components.
- `disc_qs`: locations of discrete components, corresponding to duplicated values in the input `qs`. May be `numeric(0)` if there are no discrete components.

- `cont_ps`: probability levels for the continuous part of the distribution
- `cont_qs`: quantile values for the continuous part of the distribution
- `disc_ps_range`: a list of length equal to the number of point masses in the discrete distribution. Each entry is a numeric vector of length two with the value of the CDF approaching the point mass from the left and from the right.

`step_interp_factory` *A factory that returns a function that performs linear interpolation, allowing for "steps" or discontinuities.*

Description

A factory that returns a function that performs linear interpolation, allowing for "steps" or discontinuities.

Usage

```
step_interp_factory(x, y, cont_dir = c("right", "left"), increasing = TRUE)
```

Arguments

`x`	numeric vector with the "horizontal axis" coordinates of the points to interpolate.
`y`	numeric vector with the "vertical axis" coordinates of the points to interpolate.
`cont_dir`	at steps or discontinuities, the direction from which the function is continuous. This will be "right" for a CDF or "left" for a QF.
`increasing`	boolean indicating whether the function is increasing or decreasing. Only used in the degenerate case where there is only one unique value of `x`.

Value

a function with argument `x` that performs linear approximation of the input data points.

`unique_tol` *Get unique values in a sorted numeric vector, where comparison is up to a specified numeric tolerance. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as corresponding to a single unique value even if not all values in the run are within the tolerance.*

Description

Get unique values in a sorted numeric vector, where comparison is up to a specified numeric tolerance. If there is a run of values where each consecutive pair is closer together than the tolerance, all are labeled as corresponding to a single unique value even if not all values in the run are within the tolerance.

Usage

```
unique_tol(x, tol = 1e-06, ties = mean)
```

Arguments

x	a numeric vector in which to identify duplicates
tol	numeric tolerance for identifying duplicates
ties	a function that is used to summarize groups of values that fall within the tolerance

Value

a numeric vector of the unique values in x

Index