

Package ‘phenesse’

October 14, 2022

Title Estimate Phenological Metrics using Presence-Only Data

Version 0.1.2

Description Generates Weibull-parameterized estimates of phenology for any percentile of a distribution using the framework established in Cooke (1979) <[doi:10.1093/biomet/66.2.367](https://doi.org/10.1093/biomet/66.2.367)>. Extensive testing against other estimators suggest the `weib_percentile()` function is especially useful in generating more accurate and less biased estimates of onset and offset (Belitz et al. 2020 <[doi.org:10.1111/2041-210X.13448](https://doi.org/10.1111/2041-210X.13448)>. Non-parametric bootstrapping can be used to generate confidence intervals around those estimates, although this is computationally expensive. Additionally, this package offers an easy way to perform non-parametric bootstrapping to generate confidence intervals for quantile estimates, mean estimates, or any statistical function of interest.

Maintainer Michael Belitz <mbelitz@ufl.edu>

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Depends R (>= 3.5)

Imports boot, fitdistrplus, stats

License CC0

URL <https://github.com/mbelitz/phenesse>

BugReports <https://github.com/mbelitz/phenesse/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Michael Belitz [aut, cre] (<<https://orcid.org/0000-0002-8162-5998>>),
Caitlin Campbell [ctb] (<<https://orcid.org/0000-0002-8199-7775>>),
Daijiang Li [ctb] (<<https://orcid.org/0000-0002-0925-3421>>)

Repository CRAN

Date/Publication 2020-07-28 23:00:06 UTC

R topics documented:

estimate_ci	2
inat_examples	3
mean_ci	4
quantile_ci	5
weib_percentile	6
weib_percentile_ci	7

Index	9
--------------	----------

estimate_ci	<i>Calculate confidence intervals using bootstrap of any statistical function of interest.</i>
-------------	--

Description

Calculate confidence intervals using bootstrap of any statistical function of interest.

Usage

```
estimate_ci(
  observations,
  .f,
  n_boots,
  parallelize = "no",
  ncpus = getOption("boot.ncpus", 1L),
  cl = NULL,
  type = "perc",
  conf = 0.95
)
```

Arguments

observations	is a vector of dates/time of observations given as numeric values
.f	function to use
n_boots	is the number of bootstraps you want to run to create the CIs
parallelize	The type of parallel operation to be used (if any). If missing, the default is that no parallelization will occur. Parallelization options are "multicore" and "snow"
ncpus	An integer that represents the number of processes to be used in parallel operation.
cl	An optional parallel or snow cluster for use if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the boot call.
type	A vector of character strings representing the type of intervals required to calculate the CI. Defaults to "perc". See ??boot.ci for more information.
conf	The confidence level wanted. Defaults to 95% CI.

Value

A data frame with estimate, and the lower and upper points of its confidence interval

inat_examples

Example iNaturalist-sourced data

Description

Example data downloaded from iNaturalist.org for the Washington, DC area using the bounding box bounds = c(38, -77, 39, -76). Data was downloaded on 10/30/2019 for four species: Speyeria cybele, Danaus plexippus, Rudbeckia hirta, and Asclepias syriaca. These data were not scored to mark phenology, so all life stages/reproductive stages are included in the download. The download only includes 2019 observations and the doy (day of year) column was added post data download by MW Belitz using the lubridate package.

Usage

```
data(inat_examples)
```

Format

A data frame with 252 rows and 6 variables:

scientific_name binomial of species

latitude latitude where observations occurred

longitude longitude where observations occurred

common_name common name related to species

observed_on original date listed of observation

doy day of year when the observation occurred, variable created by MW Belitz using the package lubridate

References

<https://inaturalist.org>

Examples

```
data(inat_examples)
## Not run:
View(inat_examples)

## End(Not run)
```

mean_ci	<i>Calculating the confidence intervals (CIs) of an arithmetic mean.</i>
---------	--

Description

mean_ci Function estimates CIs using nonparametric bootstrapping around a mean estimate.

Usage

```
mean_ci(observations, bootstraps = 1e+05, conf = 0.95, type = "perc")
```

Arguments

observations	A vector of observations given as numeric values
bootstraps	The number of bootstraps you want to run to create the CIs, defaults to 100000
conf	The confidence level wanted. Defaults to 95% CI.
type	A vector of character strings representing the type of intervals required to calculate the CI. Defaults to "bca". See ??boot.ci for more information.

Value

The estimated CIs around a mean estimate.

Functions

- mean_ci: Estimates CIs around a mean percentile estimate using non-parametric bootstrapping from the boot package

Examples

```
# Estimate when the mean observation of Rudbeckia hirta for the year 2019 up
# to October
data(inat_examples)
r_hirta <- subset(inat_examples, scientific_name == "Rudbeckia hirta")
mean_ci(observations = r_hirta$doy , bootstraps = 100)

# note low number of bootstraps for quick processing speed
```

quantile_ci	<i>Calculating the confidence intervals (CIs) of a quantile estimate of a vector of observations using non-parametric bootstrapping.</i>
-------------	--

Description

quantile_ci Estimates CIs around a quantile percentile estimate using non-parametric bootstrapping from the boot package

Usage

```
quantile_ci(
  observations,
  percentile,
  bootstraps = 1e+05,
  conf = 0.95,
  type = "perc"
)
```

Arguments

observations	A vector of observations given as numeric values
percentile	The percentile of interest
bootstraps	The number of bootstraps you want to run to create the CIs, defaults to 100000
conf	The confidence level wanted. Defaults to 95% CI.
type	A vector of character strings representing the type of intervals required to calculate the CI. Defaults to "bca". See ??boot.ci for more information.

Value

The quantile estimate and confidence intervals.

Examples

```
# Gather sightings of iNaturalist observations for four species:
# Danaus plexippus, Speyeria cybele, Rudbeckia hirta, and Asclepias syriaca

# Estimate when the first 10 percent of individuals of the butterfly species
# Speyeria cybele are in flight.

data(inat_examples)
s_cybele <- subset(inat_examples, scientific_name == "Speyeria cybele")
quantile_ci(observations = s_cybele$doy, percentile = 0.1, bootstraps = 100)

# note low number of bootstraps for quick processing speed
```

weib_percentile	<i>Calculating a percentile estimate of a seasonal abundance distribution from incidental observations.</i>
-----------------	---

Description

weib_percentile uses empirical bootstrapping to estimate a percentile of the Weibull distribution, given random variables.

Usage

```
weib_percentile(observations, percentile, iterations = 500)
```

Arguments

observations	is a vector of dates/time of observations given as integers
percentile	is the percentile of the cumulative distribution function of interest
iterations	is the number of iterations you want to use to bootstrap an estimate of bias of the original CDF. The bias is used to calculate a Weibull-corrected estimate of the percentile bound.

Value

The Weibull-corrected estimate of the percentile of interest.

Examples

```
# Gather sightings of iNaturalist observations for four species:
# Danaus plexippus, Speyeria cybele, Rudbeckia hirta, and Asclepias syriaca

# Estimate when the first 50 percent of individuals of the milkweed species
# Asclepias syriaca have been observed.

data(inat_examples)
a_syriaca <- subset(inat_examples, scientific_name == "Asclepias syriaca")
weib_percentile(a_syriaca$doy, percentile = 0.5, iterations = 500)

# Estimate when 90 percent of individuals of the milkweed species A. syriaca
# have been observed, using only 100 iterations for quicker processing. To
# get a more stable result, more iterations should be used.

weib_percentile(a_syriaca$doy, percentile = 0.9, iterations = 100)
```

weib_percentile_ci	<i>Calculating the CIs of a percentile estimate of a seasonal abundance distribution using the non-parametric bootstrapping.</i>
--------------------	--

Description

weib_percentile_ci uses non-parametric bootstrapping from the boot package to estimate 95

Usage

```
weib_percentile_ci(
  observations,
  iterations,
  percentile,
  bootstraps,
  type = "perc",
  conf = 0.95,
  parallelize = "no",
  ncpus = getOption("boot.ncpus", 1L),
  cl = NULL
)
```

Arguments

observations	is a vector of dates/time of observations given as numeric values
iterations	is the number of iterations you want to run to create empirical bootstrapping to estimate bias of original CDF. The bias is used to calculate a bias corrected estimate of the percentile bound.
percentile	is the percentile of the cumulative distribution function of interest
bootstraps	is the number of bootstraps you want to run to create the CIs
type	A vector of character strings representing the type of intervals required to calculate the CI. Defaults to "perc". See <code>??boot.ci</code> for more information.
conf	The confidence level wanted. Defaults to 95% CI.
parallelize	The type of parallel operation to be used (if any). If missing, the default is that no parallelization will occur. Parallelization options are "multicore" and "snow"
ncpus	An integer that represents the number of processes to be used in parallel operation.
cl	An optional parallel or snow cluster for use if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the boot call.

Value

The Weibull-corrected estimate of the percentile of interest and CIs.

Examples

```
# Gather sightings of iNaturalist observations for four species:
# Danaus plexippus, Speyeria cybele, Rudbeckia hirta, and Asclepias syriaca

# Estimate when the first 50 percent of individuals of the butterfly species
# Speyeria cybele are in flight. Note, only 10 iterations are being used
# in this example to keep computation time low, but more iterations should
# be used to get a more stable result. See vignette for parallelization
# options.

data(inat_examples)
s_cybele <- subset(inat_examples, scientific_name == "Speyeria cybele")
weib_percentile_ci(observations = s_cybele$doy, iterations = 10,
                  percentile = 0.5, bootstraps = 100)
```


Index

- * **datasets**
 - inat_examples, 3
- * **estimates**
 - mean_ci, 4
- * **mean**
 - mean_ci, 4
- * **percentile**
 - quantile_ci, 5
 - weib_percentile, 6
 - weib_percentile_ci, 7
- * **phenology**
 - mean_ci, 4
 - quantile_ci, 5
 - weib_percentile, 6
 - weib_percentile_ci, 7
- * **quantile**
 - quantile_ci, 5
- * **weibull**
 - weib_percentile, 6
 - weib_percentile_ci, 7

estimate_ci, 2

inat_examples, 3

mean_ci, 4

quantile_ci, 5

weib_percentile, 6

weib_percentile_ci, 7