Package ‘ThermalSampleR’

June 13, 2023

Type  Package

Title  Calculate Sample Sizes Required for Critical Thermal Limits Experiments

Version  0.1.1

Maintainer  Clarke JM van Steenderen <vsteenderen@gmail.com>

Description  We present a range of simulations to aid researchers in determining appropriate sample sizes when performing critical thermal limits studies (e.g. CTmin/CTmin experiments). A number of wrapper functions are provided for plotting and summarising outputs from these simulations. These simulations are presented in van Steenderen, C.J.M., Sutton, G.F., Owen, C.A., Martin, G.D., and Coetzee, J.A. Sample size assessments for thermal physiology studies: An R package and R Shiny GUI. 2023. Physiological Entomology. Under review. The GUI version of this package is available on the R Shiny online server at: <https://clarkevansteenderen.shinyapps.io/ThermalSampleR_Shiny/> , or it is accessible via GitHub at <https://github.com/clarkevansteenderen/ThermalSampleR_Shiny/>. We would like to thank Grant Duffy (University of Otago, Dunedin, New Zealand) for granting us permission to use the source code for the Test of Total Equivalency function.

License  GPL-3

Encoding  UTF-8

LazyData  true

Depends  R (>= 2.10)

Imports  tidyr, dplyr, purrr, rlang, MASS (>= 7.3), stats (>= 3.4.0), graphics (>= 3.4.0), base (>= 3.4.0), magrittr, utils, ggplot2, cowplot, EnvStats, sn, janitor, testthat

RoxygenNote  7.2.3

Suggests  rmarkdown, knitr

VignetteBuilder  knitr

NeedsCompilation  no
**boot_one**

**Author**  Guy Frederick Sutton [aut] (Wrote original R functions, and adapted the Duffy et. al. 2021 R function into this package.,
[https://orcid.org/0000-0003-2405-0945>),
Clarke JM van Steenderen [aut, cre] (Compiled the R package, and the associated R Shiny application.,
[https://orcid.org/0000-0002-4219-446X>),
Grant Duffy [cph] (Granted permission to use the source code for the Test of Total Equivalency function.,
[https://orcid.org/0000-0002-9031-8164>)

**Repository** CRAN

**Date/Publication** 2023-06-13 08:20:18 UTC

**R topics documented:**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_one</td>
<td>Bootstrap sampling to calculate summary statistics of CTL values</td>
<td>2</td>
</tr>
<tr>
<td>boot_two</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>coreid_data</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>equiv_tost</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>plot_one_group</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>plot_two_groups</td>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>

**Index** 8

---

**Description**

Calculate mean and CI’s of CTL for a single population

**Arguments**

- **data**  Data frame contains raw data. Must contain a column with a population identifier (e.g. population ID), and a column containing critical thermal limit data (e.g. temperatures at which critical limits are reached).
- **groups_col**  Factor. Column containing name(s) of population(s) of interest
- **groups_which**  Character. Which population should be analysed?
- **n_max**  Numeric. Maximum sample size to extrapolate simulations.
- **n_min**  Numeric. Minimum sample size to extrapolate simulations. Defaults to 3.
- **iter**  Numeric. Number of bootstrap samples. Defaults to 29.
- **response**  Numeric. Column containing thermal limit data for individual samples

**Value**

A data frame of CTL summary statistics from bootstrap resamples
Examples

```r
head(coreid_data)
sims <- boot_one(data = coreid_data,
                 groups_col = col,
                 groups_which = "Catorhintha schaffneri_APM",
                 response = response,
                 n_max = 49,
                 iter = 99)
```

**boot_two**  
Bootstrap sampling for difference in means between two groups

**Description**

Calculate difference in mean CT limits between two groups.

**Arguments**

- `data`: Data frame contain raw data.
- `groups_col`: Factor. Column containing names of two populations to compare.
- `n_max`: Numeric. Maximum sample size to extrapolate simulations.
- `n_min`: Numeric. Minimum sample size to extrapolate simulations. Defaults to 3.
- `response`: Numeric. Column containing thermal limit data for individual samples.
- `group1`: String. Name of first population to compare.
- `group2`: String. Name of second population to compare.
- `colour_exp`: Colour of the experimental data. Defaults to "blue".
- `colour_extrap`: Colour of the extrapolated data. Defaults to "red".
- `legend.position`: Position of the legend. Defaults to "top". Can be "bottom", "left", "right", or "none".
- `ggtheme`: The theme for the ggplot created. See ggplot2 themes for options. Default set to theme_classic().

**Value**

A data frame of bootstrap resamples
Examples

head(coreid_data)
sims <- boot_two(data = coreid_data,
                 groups.col = col,
                 response = response,
                 group1 = "Catorhintha schaffneri_APM",
                 group2 = "Catorhintha schaffneri_NPM",
                 n_max = 49,
                 iter = 99)

coreid_data  Example critical thermal limit data for ThermalSampleN package

Description

A dataset containing critical thermal limit (CTmin) data for the sap-sucking bug, Catorhintha schaffneri.

Usage

coreid_data

Format

A data frame with 60 rows and 2 variables:

col  insect population tested, string
response  CTmin value for individual insect, in degrees celsius ...

Source

"Unpublished data provided by Philippa Muskett (Centre for Biological Control, Rhodes University, South Africa)

equiv_tost  equiv_tost

Description

Perform a Test of Total Equivalence as adapted from Duffy et al. (2021) (<https://doi.org/10.1111/1365-2435.13928>)
Arguments

data Data frame contains raw data. Must contain a column with a population identifier (e.g. population ID), and a column containing critical thermal limit data (e.g. temperatures at which critical limits are reached).
groups_col Factor. Column containing the name of the population of interest (group ID)
groups_which Character. Which population should be analysed?
response Numeric. Column containing thermal limit data for individual samples
skews Numeric. Vector containing skewness parameter(s). Defaults to 0, 1, 2, 10, 50.
equiv_margin Numeric. Equivalence of subsets to full population CT estimate (unit = degree Celcius). Defaults to 1.
pop_n Numeric. Size of population to sample (will test subsamples of size pop_n - x against pop_n for equivalence) Defaults to population size = 30
colrs Character. Vector of colours for each skewness parameter value. E.g. if two skewness parameter values are set, choose two colours: colrs = c("blue", "red"). Defaults to "blue", "red", "orange", "forestgreen", "lightgrey".

Value

Two plots; (a) equivalence of means, and (b) equivalence of variances

Examples

```r
head(coreid_data)
res <- equiv_tost(data = coreid_data,
groups_col = col,
groups_which = "Catorhintha schaffneri_APM",
response = response,
skews = c(1,10),
colrs = c("lightblue", "lightpink"),
equiv_margin = 1,
pop_n = 5)
```

plot_one_group  Plot output from boot_sample

Description

Plot output from boot_one.
Arguments

- **x**: Output from `boot_one` function.
- **n_max**: Numeric. Maximum sample size to extrapolate simulations.
- **n_min**: Numeric. Minimum sample size to extrapolate simulations. Defaults to 3.
- **colour_exp**: Colour of the experimental data. Defaults to "blue".
- **colour_extrap**: Colour of the extrapolated data. Defaults to "red".
- **legend.position**: Position of the legend. Defaults to "top". Can be "bottom", "left", "right", or "none".
- **alpha_val**: Change the degree of shading of the graphs. Default is 0.2.
- **ggtheme**: The theme for the ggplot created. See ggplot2 themes for options. Default set to `theme_classic()`.

Value

Two plots; (a) precision of the CTmin estimate across experimental and extrapolated sample sizes; (b) the sampling distribution (range of plausible CTmin values) across experimental and extrapolated sample sizes.

Examples

```r
sims <- boot_one(coreid_data,
                 groups_col = col,
                 groups_which = "Catorhintha schaffneri_APM",
                 n_max = 30,
                 response = response)
plot_one_group(x = sims,
               n_min = 3,
               n_max = 15,
               colour_exp = "darkblue",
               colour_extrap = "green",
               legend.position = "right")
```

Description

- **plot_two_groups**: Plot output from `boot_two_groups`.
- **Description**: Plot output from `boot_two`.
**plot_two_groups**

**Arguments**

- `x`: Output from `boot_two_groups` function. Defaults to 'sims'.
- `n_max`: Numeric. Maximum sample size to extrapolate simulations.
- `n_min`: Numeric. Minimum sample size to extrapolate simulations. Defaults to 3.
- `colour_exp`: Colour of the experimental data. Defaults to "blue".
- `colour_extrap`: Colour of the extrapolated data. Defaults to "red".
- `legend.position`: Position of the legend. Defaults to "top". Can be "bottom", "left", "right", or "none".
- `alpha_val`: Change the degree of shading of the graphs. Default is 0.2.
- `ggtheme`: The theme for the ggplot created. See ggplot2 themes for options. Default set to `theme_classic()`.

**Value**

Two plots: (a) the precision of the estimates for the difference in CTmin between the two selected groups across sample sizes; (b) the 95%

**Examples**

```r
sims <- boot_two(data = coreid_data,
                 groups_col = col,
                 response = response,
                 group1 = "Catorhintha schaffneri_APM",
                 group2 = "Catorhintha schaffneri_NPM",
                 n_max = 30,
                 iter = 99)

plots <- plot_two_groups(x = sims,
                          n_min = 3,
                          n_max = 30,
                          colour_exp = "gold",
                          colour_extrap = "darkgreen",
                          legend.position = "right")
```

```
Index

* datasets
  coreid_data, 4

boot_one, 2
boot_two, 3

coreid_data, 4

equiv_tost, 4

plot_one_group, 5
plot_two_groups, 6