Package ‘tidyboot’

October 14, 2022

Type Package
Title Tidyverse-Compatible Bootstrapping
Version 0.1.1
Description Compute arbitrary non-parametric bootstrap statistics on data in tidy data frames.
Depends R (>= 3.4.0)
License GPL-3
URL https://github.com/langcog/tidyboot
BugReports http://github.com/langcog/tidyboot/issues
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
Imports dplyr (>= 0.7.4), modelr (>= 0.1.1), purrr (>= 0.2.4), rlang (>= 0.1.6), tidyr (>= 0.7.2)
NeedsCompilation no
Author Mika Braginsky [aut, cre],
Daniel Yurovsky [aut]
Maintainer Mika Braginsky <mika.br@gmail.com>
Repository CRAN
Date/Publication 2018-03-14 04:13:49 UTC

R topics documented:

  ci_lower .................................................. 2
  ci_upper .................................................. 2
  tidyboot .................................................. 3
  tidyboot.data.frame ..................................... 3
  tidyboot.logical .......................................... 4
  tidyboot.numeric ......................................... 5
  tidyboot_mean ............................................ 6

Index 7
## `ci_lower`

**Description**
Confidence interval (lower 2.5%)

**Usage**
```r
ci_lower(x, na.rm = FALSE)
```

**Arguments**
- `x` A numeric vector
- `na.rm` A logical value indicating whether NA values should be stripped before the computation proceeds.

**Value**
2.5

**Examples**
```r
x <- rnorm(1000, mean = 0, sd = 1)
ci_lower(x)
```

## `ci_upper`

**Description**
Confidence interval (upper 97.5%)

**Usage**
```r
ci_upper(x, na.rm = FALSE)
```

**Arguments**
- `x` A numeric vector
- `na.rm` A logical value indicating whether NA values should be stripped before the computation proceeds.

**Value**
97.5
Examples

```r
x <- rnorm(1000, mean = 0, sd = 1)
ci_upper(x)
```

```

## List of available methods
methods(tidyboot)
```

```

## S3 method for class 'data.frame'
tidyboot(data, column = NULL, summary_function = mean,
          statistics_functions, nboot = 1000, ...)
```

```r
## S3 method for class 'data.frame'
tidyboot(data, column = NULL, summary_function = mean,
          statistics_functions, nboot = 1000, ...)
```
Arguments

data A data frame.
column A column of data to bootstrap over (if not supplied, summary_function and statistic_function must operate over the appropriate data frame).
summary_function A function to be computed over each set of samples as a data frame, or a function to be computed over each set of samples as a single column of a data frame indicated by column (defaults to mean).
statistic_function A function to be computed over each set of samples as a data frame, or a function to be computed over each set of samples as a single column of a data frame indicated by column.
nboot The number of bootstrap samples to take (defaults to 1000).
... Other arguments passed from generic.

Examples

```r
## Mean and 95% confidence interval for 500 samples from two different normal distributions
require(dplyr)
gauss1 <- data_frame(value = rnorm(500, mean = 0, sd = 1), condition = 1)
gauss2 <- data_frame(value = rnorm(500, mean = 2, sd = 3), condition = 2)
df <- bind_rows(gauss1, gauss2)
df %>% group_by(condition) %>%
tidyboot(summary_function = function(x) x %>% summarise(mean = mean(value)),
       statistic_function = function(x) x %>%
       summarise_at(vars(mean), funs(ci_upper, mean, ci_lower)))
```

tidyboot.logical Non-parametric bootstrap for logical vector data

Description

Computes arbitrary bootstrap statistics on univariate data.

Usage

```r
## S3 method for class 'logical'
tidyboot(data, summary_function = mean,
         statistic_function = mean, nboot = 1000, size = 1, replace = TRUE, ...)
```

Arguments

data A logical vector of data to bootstrap over.
summary_function A function to be computed over each set of samples. This function needs to take a vector and return a single number (defaults to mean).
tidyboot.numeric

statistics_functions
   A named list of functions to be computed over the set of summary values from all samples.

nboot
   The number of bootstrap samples to take (defaults to 1000).

size
   The fraction of items to sample (defaults to 1).

replace
   Logical indicating whether to sample with replacement (defaults to TRUE).

... Other arguments passed from generic.

Examples

## Mean and 95% confidence interval for 500 samples from a binomial distribution
x <- as.logical(rbinom(500, 1, 0.5))
tidyboot(x, statistics_functions = c(ci_lower, mean, ci_upper))

Description
Computes arbitrary bootstrap statistics on univariate data.

Usage

## S3 method for class 'numeric'
tidyboot(data, summary_function = mean,
   statistics_functions, nboot = 1000, size = 1, replace = TRUE, ...)

Arguments

data A numeric vector of data to bootstrap over.

summary_function
   A function to be computed over each set of samples. This function needs to take a vector and return a single number (defaults to mean).

statistics_functions
   A named list of functions to be computed over the set of summary values from all samples.

nboot
   The number of bootstrap samples to take (defaults to 1000).

size
   The fraction of items to sample (defaults to 1).

replace
   Logical indicating whether to sample with replacement (defaults to TRUE).

... Other arguments passed from generic.

Examples

## Mean and 95% confidence interval for 500 samples from a normal distribution
x <- rnorm(500, mean = 0, sd = 1)
tidyboot(x, statistics_functions = list("ci_lower" = ci_lower,
   "mean" = mean,
   "ci_upper" = ci_upper))
Non-parametric bootstrap and empirical central tendency for data frames Designed to make standard use of tidyboot.data.frame easier

Description

Computes arbitrary bootstrap statistics on univariate data. NOTE: Both empirical functions and bootstrapping functions will be computed over the grouping variables currently specified for the data frame.

Usage

tidyboot_mean(data, column, nboot = 1000, na.rm = FALSE)

Arguments

data A data frame.
column A column of data to bootstrap over.
nboot The number of bootstrap samples to take (defaults to 1000).
na.rm A logical value indicating whether NA values should be stripped before the computation proceeds.

Examples

## Mean and 95% confidence interval for 500 samples from two different normal distributions
require(dplyr)
gauss1 <- data_frame(value = rnorm(500, mean = 0, sd = 1), condition = 1)
gauss2 <- data_frame(value = rnorm(500, mean = 2, sd = 3), condition = 2)
df <- bind_rows(gauss1, gauss2)
df %>%
  group_by(condition) %>%
tidyboot_mean(column = value)
Index

ci_lower, 2
  ci_upper, 2

tidyboot, 3
tidyboot.data.frame, 3
tidyboot.logical, 4
tidyboot.numeric, 5
tidyboot_mean, 6