Package 'BootstrapTests'

December 22, 2025

Type Package

Title Bootstrap-Based Hypothesis Testing using Different Resampling Schemes

Version 0.1.0

Description Perform bootstrap-based hypothesis testing procedures on three statistical problems. In particular, it covers independence testing, testing the slope in a linear regression setting, and goodness-of-fit testing, following (Derumigny, Galanis, Schipper and Van der Vaart, 2025) <doi:10.48550/arXiv.2512.10546>.

License GPL-3

URL https://github.com/AlexisDerumigny/BootstrapTests

BugReports https://github.com/AlexisDerumigny/BootstrapTests/issues

Imports pbapply (>= 1.7.2), stats (>= 4.4.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Alexis Derumigny [aut] (ORCID: <https://orcid.org/0000-0002-6163-8097>), Miltiadis Galanis [aut], Wieger Schipper [aut, cre] (ORCID: <https://orcid.org/0009-0004-5661-4949>), Aad van der Vaart [aut] (ORCID: <https://orcid.org/0000-0002-8074-2375>)

Maintainer Wieger Schipper <W.R.Schipper@tudelft.nl>

Repository CRAN

Date/Publication 2025-12-22 20:00:11 UTC

Contents

`perform_GoF_test` *Perform a univariate goodness-of-fit (GoF) hypothesis test via bootstrap resampling*

Description

This function performs a bootstrap goodness-of-fit hypothesis test for a specific univariate parametric family. The null hypothesis corresponds to the sample coming from the specified parametric family, while the alternative hypothesis corresponds to the sample not coming from the specified parametric family. This function implements a parametric bootstrap and a non-parametric bootstrap. The test statistic is the Kolmogorov-Smirnov test statistic. To estimate the parameters of the parametric family, either a minimum distance estimator, or a MLE estimator (the sample mean and variance) is used. On the bootstrap sample, we have also implemented a centered MD estimator, as in the paper. For now, only a test of normality is implemented. This function gives the corresponding p-values, the true test statistic and the bootstrap-version test statistics. The default (and valid) method implemented in this function is the parametric bootstrap, together with the equivalent test statistic and the MLE parameter estimator. Via the `bootstrapOptions` argument, the user can specify other bootstrap resampling schemes, test statistics, and parameter estimators.

Usage

```
perform_GoF_test(
  X_data,
  parametric_fam = "normal",
  nBootstrap = 100,
  mygrid = NULL,
  show_progress = TRUE,
  bootstrapOptions = NULL,
  verbose = 0
)
```

Arguments

`X_data`	numerical input vector. Perform a GoF test whether or not this sample comes from `"parametric_fam"`, a specified parametric distribution.
`parametric_fam`	name of the parametric family. For the moment, only `"normal"` is supported.
`nBootstrap`	numeric value of the number of bootstrap resamples. Defaults to 100.
`mygrid`	description of the grid used to compute the CDFs on. This must be one of

- NULL: a regularly spaced grid from the minimum value to the maximum value with 100 points is used. This is the default.
- A numeric of size 1. This is used at the length of the grid, replacing 100 in the above explanation.
- A numeric vector of size larger than 1. This is directly used as the grid.

show_progress logical value indicating whether to show a progress bar

bootstrapOptions

This can be one of

- NULL. This uses the default options type_boot = "param", type_stat = "eq" and type_estimator_bootstrap = "MLE".
- a list with at most 3 elements named:
  - type_boot type of bootstrap resampling scheme. It must be one of
    - "param" for the parametric bootstrap (i.e. under the null). This is the default.
    - "NP" for the non-parametric bootstrap (i.e. n out of n bootstrap).
  - type_stat type of test statistic to be used. It must be one of
    - "eq" for the equivalent test statistic $T_n^* = \sqrt{n}||\hat{F}^* - F_{\hat{\theta}^*}||$
    - "cent" for the centered test statistic $T_n^* = \sqrt{n}||\hat{F}^* - \hat{F} + F_{\hat{\theta}} - F_{\hat{\theta}^*}||$

    For each type_boot there is only one valid choice of type_stat to be made. If type_stat is not specified, the valid choice is automatically used.
  - type_estimator_bootstrap: the bootstrap parameter estimator to be used. It must be one of:
    - "MLE" for the MLE estimator (for the normal distribution, this corresponds to the usual empirical mean and variance).
      This is always a valid choice in the case that the combination (type_boot, type_stat) is valid (as defined above). Therefore, this is the default option. It is also the fastest type of estimator.
    - "MD-eq" for the Minimum Distance estimator. This is a valid choice if and only if type_stat = "eq". It is necessary in this case to use an equivalent bootstrap estimator to match the equivalent bootstrap test statistic. This bootstrap parameter estimator is given as: $\theta_n^{*,MD} = \arg\min_\theta ||\hat{F}^* - F_\theta||$
    - "MD-cent" for the centered Minimum Distance estimator. This is a valid choice if and only if type_stat = "cent". It is necessary in this case to perform a centering on the bootstrap estimator to match the centered bootstrap test statistic. This bootstrap parameter estimator is given as: $\theta_n^{*,MD,cent} = \arg\min_\theta ||\hat{F}^* - F_\theta - \hat{F} + F_{\hat{\theta}}||$
- "all" this gives test results for all theoretically valid combinations of bootstrap resampling schemes.
- "all and also invalid" this gives test results for all possible combinations of bootstrap resampling schemes and test statistics, including invalid ones.

A warning is raised if the given combination of type_boot, type_stat, and type_estimator_bootstrap is theoretically invalid.

verbose — If `verbose = 0`, this function is silent and does not print anything. Increasing values of `verbose` print more details about the progress of the computations.

Value

A class object with components

- `pvals_df` a dataframe of p-values and bootstrapped test statistics:
  These are the p-values for the combinations of bootstrap resampling schemes, test statistics (centered and equivalent), and different parameter estimators.
  It also contains the vectors of bootstrap test statistics for each of these combinations.
- `true_stat` a named vector of size 2 containing the true test statistics. The first entry is the Kolmogorov-Smirnov test statistic for the Minimum Distance estimator, and the second entry is the Kolmogorov-Smirnov test statistic for the MLE parameter estimator.
- `nBootstrap` number of bootstrap repetitions.
- `nameMethod` string for the name of the method used.

References

Derumigny, A., Galanis, M., Schipper, W., & van der Vaart, A. (2025). Bootstrapping not under the null? ArXiv preprint, doi:10.48550/arXiv.2512.10546

See Also

`perform_regression_test`,`perform_independence_test`. The print and plot methods, such as `plot.bootstrapTest`.

Examples

```
n <- 100
# Under H1
X_data <- rgamma(n,2,3)
result <- perform_GoF_test(X_data,
                           nBootstrap = 100,
                           bootstrapOptions = list(type_boot = "param",
                                                   type_stat = "eq",
                                                   type_estimator_bootstrap = "MLE")
                          )
print(result)
plot(result)

# Under H0
X_data <- rnorm(n)
result <- perform_GoF_test(X_data, nBootstrap = 100)
print(result)
plot(result)
```

perform_independence_test *Perform a hypothesis test of independence*

Description

Perform a hypothesis test of statistical independence by means of bootstrapping. The null hypothesis is that of independence between the two random variables, versus the alternative of dependence between them. This procedure gives a total of 8 combinations of bootstrap resampling schemes (nonparametric and independent), test statistics (centered and equivalent), and Kolmogorov-Smirnov or L2-type of true test statistic. This function gives the corresponding p-values, the true test statistic and the bootstrap-version test statistics. The default (and valid) method implemented in this function is the null bootstrap, together with the equivalent test statistic and Kolmogorov-Smirnov test statistic. Via the `bootstrapOptions` argument, the user can specify other bootstrap resampling schemes and test statistics.

Usage

```
perform_independence_test(
  X1,
  X2,
  my_grid = NULL,
  nBootstrap = 100,
  show_progress = TRUE,
  bootstrapOptions = NULL
)
```

Arguments

`X1`, `X2`	numerical vectors of the same size. The independence test tests whether `X1` is independent from `X2`.
`my_grid`	the grid on which the CDFs are estimated. This must be one of • `NULL`: a regularly spaced grid from the minimum value to the maximum value of each variable with `20` points is used. This is the default. • A numeric of size 1. This is used at the length of both grids, replacing `20` in the above explanation. • A numeric vector of size larger than 1. This is directly used as the grid for both variables. • A list of two numeric vectors, which are used as the grids for both variables `X1` and `X2` respectively.
`nBootstrap`	number of bootstrap repetitions.
`show_progress`	logical value indicating whether to show a progress bar
`bootstrapOptions`	This can be one of

- NULL This uses the default options type_boot = "indep", type_stat = "eq" and type_norm = "KS".
- a list with at most 3 elements names
  - type_boot type of bootstrap resampling scheme. It must be one of
    - "indep" for the independence bootstrap (i.e. under the null). This is the default.
    - "NP" for the non-parametric bootstrap (i.e. n out of n bootstrap).
  - type_stat type of test statistic to be used. It must be one of
    - "eq" for the equivalent test statistic
    $$T_n^* = \sqrt{n}||\hat{F}^*_{(X,Y)} - \hat{F}^*_X \hat{F}^*_Y||$$
    - "cent" for the centered test statistic
    $$T_n^* = \sqrt{n}||\hat{F}^*_{(X,Y)} - \hat{F}^*_X \hat{F}^*_Y - (\hat{F}_{(X,Y)} - \hat{F}_X \hat{F}_Y)||$$

    For each type_boot there is only one valid choice of type_stat to be made. If type_stat is not specified, the valid choice is automatically used.
  - type_norm type of norm to be used for the test statistic. It must be one of
    - "KS" for the Kolmogorov-Smirnov type test statistic. This is the default. It is given as
    $$T_n = \sqrt{n} \sup_{(x,y)\in\mathbb{R}^{p+q}} \left|\hat{F}_{(X,Y),n}(x,y) - \hat{F}_{X,n}(x)\hat{F}_{Y,n}(y)\right|$$
    - "L2" for the squared L2-norm test statistic.
    $$T_n = \sqrt{n} \int_{(x,y)\in\mathbb{R}^{p+q}} \left(\hat{F}_{(X,Y),n}(x,y) - \hat{F}_{X,n}(x)\hat{F}_{Y,n}(y)\right)^2 \mathrm{d}x\mathrm{d}y$$
- "all" this gives test results for all theoretically valid combinations of bootstrap resampling schemes.
- "all and also invalid" this gives test results for all possible combinations of bootstrap resampling schemes and test statistics, including invalid ones.

A warning is raised if the given combination of type_boot_user and type_stat_user is theoretically invalid.

Value

A class object with components

- pvals_df: a dataframe of p-values and bootstrapped test statistics:
  These are the p-values for the 8 combinations of bootstrap resampling schemes (nonparametric and independent), test statistics (centered and equivalent), and Kolmogorov-Smirnov or L2-type of true test statistic.
  It also contains the vectors of bootstrap test statistics for each of the combinations.
- true_stats a named vector of size 2 containing the true test statistics for the L2 and KS distances.
- nBootstrap Number of bootstrap repetitions.
- nameMethod string for the name of the method used.

References

Derumigny, A., Galanis, M., Schipper, W., & van der Vaart, A. (2025). Bootstrapping not under the null? ArXiv preprint, doi:10.48550/arXiv.2512.10546

See Also

`perform_GoF_test`, `perform_regression_test`. The print and plot methods, such as `plot.bootstrapTest`.

Examples

```
n <- 100

# Under H1
X1 <- rnorm(n)
X2 <- X1 + rnorm(n)
result <- perform_independence_test(
   X1, X2, nBootstrap = 50,
   bootstrapOptions = list(type_boot = "indep",
                           type_stat = "eq",
                           type_norm = "KS") )
print(result)
plot(result)

# Under H0
X1 <- rnorm(n)
X2 <- rnorm(n)
result <- perform_independence_test(X1, X2, nBootstrap = 50)
print(result)
plot(result)
```

`perform_regression_test`

Perform a test on the slope coefficient of a univariate linear regression

---

Description

This function performs a bootstrap regression test for given data X,Y. The null hypothesis corresponds of a slope coefficient of zero, versus the alternative hypothesis of a non-zero slope coefficient. It uses an independence/null bootstrap `"indep"`, a non-parametric `"NP"`, a residual bootstrap `"res_bs"`, a fixed design bootstrap `"fixed_design_bs"`, a fixed design null bootstrap `"fixed_design_bs_Hnull"`, a hybrid null bootstrap `"hybrid_null_bs"` as bootstrap resampling schemes to perform the bootstrap. This function gives the corresponding p-values, the true test statistic and the bootstrap-version test statistics. Furthermore, it also gives the estimated slope.The default (and valid) method implemented in this function is the null bootstrap, together with the equivalent test statistic. Via the `bootstrapOptions` argument, the user can specify other bootstrap resampling schemes and test statistics.

Usage

```
perform_regression_test(
  X,
  Y,
  nBootstrap = 100,
  show_progress = TRUE,
  bootstrapOptions = NULL
)
```

Arguments

X	numeric univariate input vector resembling the independent variables
Y	numeric univariate input vector the dependent variables
nBootstrap	numeric value of the amount of bootstrap resamples
show_progress	logical value indicating whether to show a progress bar
bootstrapOptions	This can be one of • NULL This uses the default options type_boot = "indep", type_stat = "eq". • a list with at most 2 elements names – type_boot type of bootstrap resampling scheme. It must be one of ∗ "indep" for the independence bootstrap (i.e. under the null). This is the default. ∗ "NP" for the non-parametric bootstrap (i.e. n out of n bootstrap). ∗ "res_bs" for the residual bootstrap. ∗ "hybrid_null_bs" for the hybrid null bootstrap ∗ "fixed_design_bs" for the fixed design bootstrap ∗ "fixed_design_bs_Hnull" for the fixed design null bootstrap. – type_stat type of test statistic to be used. It must be one of ∗ "eq" for the equivalent test statistic $T_n^* = \sqrt{n}\lvert\hat{b}^*\rvert$. This is the default. ∗ "cent" for the centered test statistic $T_n^* = \sqrt{n}\lvert\hat{b}^* - \hat{b}\rvert$ For each type_boot there is only one valid choice of type_stat to be made. If type_stat is not specified, the valid choice is automatically used. • "all" this gives test results for all theoretically valid combinations of bootstrap resampling schemes. • "all and also invalid" this gives test results for all possible combinations of bootstrap resampling schemes and test statistics, including invalid ones. A warning is raised if the given combination of type_boot and type_stat is theoretically invalid.

Value

A class object with components

- `pvals_df` a dataframe of p-values and bootstrapped test statistics:
  These are the p-values for the combinations of bootstrap resampling schemes, test statistics (centered and equivalent).
  It also contains the vectors of bootstrap test statistics for each of the combinations.
- `true_stat` a named vector of size 1 containing the true test statistic.
- `nBootstrap` Number of bootstrap repetitions.
- `data` named list of the used input data, i.e. X and Y.
- `nameMethod` string for the name of the method used.
- `beta` numeric value of the estimated slope of the regression model.

References

Derumigny, A., Galanis, M., Schipper, W., & van der Vaart, A. (2025). Bootstrapping not under the null? ArXiv preprint, doi:10.48550/arXiv.2512.10546

See Also

`perform_GoF_test`, `perform_independence_test`. The print and plot methods, such as `plot.bootstrapTest`.

Examples

```
n <- 100

# Under H1
X_data <- rnorm(n)
Y_data <-  X_data + rnorm(n)   #Y = X + epsilon
result <- perform_regression_test(X_data, Y_data, nBootstrap = 100,
                        bootstrapOptions =  list(type_boot = "indep",
                                                 type_stat = "eq"))
print(result)
plot(result)

# Under H0
X_data <- rnorm(n)
Y_data <-  0 * X_data + rnorm(n)   # (as b = 0 under H0)
result <- perform_regression_test(X_data, Y_data, nBootstrap = 100)
print(result)
plot(result)
```

`plot.bootstrapTest` *Plot and print the bootstrap test statistics distribution*

Description

The `plot` and `print` methods work for objects of class `bootstrapTest`. The `print` method prints the summary of the bootstrap test results. The `plot` method plots the distribution of bootstrapped test statistics as a histogram, with the true test statistic and the 95 bootstrapped test statistics highlighted. In the regression test case, the estimated regression line is plotted as well.

Usage

```
## S3 method for class 'bootstrapTest'
plot(
  x,
  xlim = NULL,
  breaks = NULL,
  legend.x = NULL,
  legend.y = NULL,
  ask = interactive(),
  plot_estimated_line = NULL,
  ...
)

## S3 method for class 'bootstrapTest'
print(x, ...)
```

Arguments

`x`	an object of class `bootstrapTest_independence` or `bootstrapTest`
`xlim`	limits for the x-axis of the histogram
`breaks`	breaks for the histogram
`legend.x`	position of the legend on the x-axis
`legend.y`	position of the legend on the y-axis
`ask`	if `TRUE`, the user is asked to press Return to see the next plot. Used only if `x` is an object of class `bootstrapTest_regression`.
`plot_estimated_line`	Boolean describing whether to plot the estimated regression line in case `x` is of class `"bootstrapTest_regression"`, i.e. output from `perform_regression_test`. By default, `plot_estimated_line = NULL`, with the meaning that the plot is done only if one estimated way of bootstrapping is given.
`...`	additional arguments passed to the `hist` function (in the case of the `plot` method) or ignored (in the case of the `print` method).

Value

These functions have no return value and are called solely for their side effects.

References

Derumigny, A., Galanis, M., Schipper, W., & van der Vaart, A. (2025). Bootstrapping not under the null? ArXiv preprint, doi:10.48550/arXiv.2512.10546

See Also

`perform_independence_test`, `perform_GoF_test`,`perform_regression_test`, which are the functions that generate such object `x`.

Index