

# Package: npbootprmFBar (via r-universe)

October 11, 2024

**Title** Informative Nonparametric Bootstrap Test with Pooled Resampling

**Version** 0.2.0.9000

**Description** Sample sizes are often small due to hard to reach target populations, rare target events, time constraints, limited budgets, or ethical considerations. Two statistical methods with promising performance in small samples are the nonparametric bootstrap test with pooled resampling method, which is the focus of Dwivedi, Mallawaarachchi, and Alvarado (2017) <[doi:10.1002/sim.7263](https://doi.org/10.1002/sim.7263)>, and informative hypothesis testing, which is implemented in the 'restriktor' package. The 'npbootprmFBar' package uses the nonparametric bootstrap test with pooled resampling method to implement informative hypothesis testing. The bootFbar() function can be used to analyze data with this method and the persimon() function can be used to conduct performance simulations on type-one error and statistical power.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**URL** <https://github.com/mightymetrika/npbootprmFBar>

**BugReports** <https://github.com/mightymetrika/npbootprmFBar/issues>

**Imports** DT, fGarch, lmPerm, mmints, npbootprm, restriktor, shiny

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Repository** <https://mightymetrika.r-universe.dev>

**RemoteUrl** <https://github.com/mightymetrika/npbootprmfbar>

**RemoteRef** HEAD

**RemoteSha** b793076758228ed81ae4754a09860a5a61e73652

## Contents

bootFbar . . . . .	2
persimon . . . . .	3
persimon_app . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

bootFbar	<i>Nonparametric Bootstrap Test with Pooled Resampling for Informative Hypothesis Testing</i>
----------	---

---

## Description

Performs a nonparametric bootstrap test with pooled resampling based on the methods described by Dwivedi et al. (2017) and using the `restriktor::iht` function as proposed by Vanbrabant and Rosseel (2020).

## Usage

```
bootFbar(
  data,
  formula,
  grp,
  constraints,
  nboot = 1000,
  conf.level = 0.95,
  seed = NULL,
  na_rm = FALSE
)
```

## Arguments

<code>data</code>	A data frame containing the dataset for analysis.
<code>formula</code>	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
<code>grp</code>	A string specifying the grouping variable in the data.
<code>constraints</code>	A matrix or data frame of constraints for the hypothesis test.
<code>nboot</code>	An integer indicating the number of bootstrap resamples (default is 1000).
<code>conf.level</code>	A numeric value specifying the confidence level for the interval (default is 0.95).
<code>seed</code>	An optional integer setting the seed for random number generation (default is NULL).
<code>na_rm</code>	A logical value indicating whether NA values should be removed (default is FALSE).

**Value**

A list containing the following components:

- pvalueA, pvalueB, pvalueF: P-values for the different test types.
- ciA, ciB, ciF: Confidence intervals for the test statistics.
- TsA, TsB, TsF: Bootstrap test statistics.
- modelo: The linear model object.
- ihto: The initial iht computation result.

**References**

Dwivedi, A. K., Mallawaarachchi, I., & Alvarado, L. A. (2017). Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method. *Statistics in Medicine*, 36(14), 2187–2205. <https://doi.org/10.1002/sim.7263>

Vanbrabant, L., & Rosseel, Y. (2020). An Introduction to Restriktor: Evaluating informative hypotheses for linear models. In R. van de Schoot & M. Miocevic (Eds.), *Small Sample Size Solutions: A Guide for Applied Researchers and Practitioners* (1st ed., pp. 157 -172). Routledge. <https://doi.org/10.4324/9780429273872-14>

**Examples**

```
bootFbar(data = iris, formula = Sepal.Length ~ -1 + Species,
          grp = "Species",
          constraints = 'Speciessetosa < Speciesversicolor < Speciesvirginica',
          nboot = 10, conf.level = 0.95, seed = NULL, na_rm = FALSE)
```

---

persimon

*Performance Simulation On Type I Error or Statistical Power*

---

**Description**

Simulates performance (Type I Error or Statistical Power) for a range of statistical tests, including bootFbar, default setting restriktor::iht, and parametric bootstrap restriktor::iht. The function is set up to replicate the table structure in Dwivedi et al. (2017) Supplemental Tables 2 & 3.

**Usage**

```
persimon(
  M1 = 5,
  S1 = 1,
  M2 = 5,
  S2 = 1,
  M3 = 5,
  S3 = 1,
  Sk1 = NULL,
```

```

Sk2 = NULL,
Sk3 = NULL,
n1 = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 15),
n2 = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 15),
n3 = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 15),
n_simulations = 10000,
nboot = 1000,
conf.level = 0.95
)

```

### Arguments

M1	Mean value for group 1.
S1	Standard deviation for group 1.
M2	Mean value for group 2.
S2	Standard deviation for group 2.
M3	Mean value for group 3.
S3	Standard deviation for group 3.
Sk1	Skewness parameter for group 1; NULL for normal distribution.
Sk2	Skewness parameter for group 2; NULL for normal distribution.
Sk3	Skewness parameter for group 3; NULL for normal distribution.
n1	Vector of sample sizes for group 1.
n2	Vector of sample sizes for group 2.
n3	Vector of sample sizes for group 3.
n_simulations	Number of simulations to run (default is 10000).
nboot	Number of bootstrap samples (default is 1000).
conf.level	Confidence level for the tests (default is 0.95).

### Details

The `persimon` function generates data for three groups with specified mean, standard deviation, and skewness, and then applies a range of statistical tests to this data, simulating the process across a specified number of iterations to assess performance in terms of Type I Error or Statistical Power. The output mirrors the format of the supplemental tables in Dwivedi et al. (2017).

### Value

A list of data frames. One with the proportions of rejecting the null hypothesis for each test and sample size combination, and the other with the number of models which did not produce errors for each combination.

### References

Dwivedi, A. K., Mallawaarachchi, I., & Alvarado, L. A. (2017). Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method. *Statistics in Medicine*, 36(14), 2187–2205. <https://doi.org/10.1002/sim.7263>

**Examples**

```
set.seed(135)
persimon(M1 = 5, S1 = 1, M2 = 5, S2 = 1, M3 = 5, S3 = 1,
         n1 = 6, n2 = 6, n3 = 6, n_simulations = 2, nboot = 4,
         conf.level = 0.95)
```

---

persimon\_app

*Run Shiny Application for bootFbar Performance Simulation*

---

**Description**

This function launches a 'shiny' application that allows users to run performance simulations for the bootFbar method and other statistical tests. The app provides an interface to set simulation parameters, run simulations, view results, and store them in a 'PostgreSQL' database.

**Usage**

```
persimon_app(dbname, datatable, host, port, user, password)
```

**Arguments**

dbname	A string specifying the name of the 'PostgreSQL' database to connect to.
datatable	A string specifying the name of the table in the database where results will be stored.
host	A string specifying the host name or IP address of the 'PostgreSQL' server.
port	An integer specifying the port number on which the 'PostgreSQL' server is listening.
user	A string specifying the username for the 'PostgreSQL' database connection.
password	A string specifying the password for the 'PostgreSQL' database connection.

**Details**

The 'shiny' application provides a user interface for setting simulation parameters, running simulations using the `persimon` function, and visualizing the results. It allows users to:

- Set parameters for the simulation (means, standard deviations, sample sizes, etc.)
- Run simulations with the specified parameters
- View simulation results and success rates in interactive tables
- Store simulation results in a PostgreSQL database
- Download stored results from the database
- View relevant citations for the methods used

The application uses the `mmints` package for database interactions and citation handling.

**Value**

A 'shiny' application object.

**See Also**

[persimon](#) for the underlying simulation function.

**Examples**

```
if(interactive()){  
  persimon_app(dbname = "my_database", datatable = "simulation_results",  
               host = "localhost", port = 5432,  
               user = "username", password = "password")  
}
```

# Index

bootFbar, 2

persimon, 3, 6

persimon\_app, 5