# Package ‘lmboot’

July 22, 2025

**Type** Package

**Title** Bootstrap in Linear Models

**Version** 0.0.1

**Date** 2019-05-13

**Description** Various efficient and robust bootstrap methods are implemented for linear models with least squares estimation. Functions within this package allow users to create bootstrap sampling distributions for model parameters, test hypotheses about parameters, and visualize the bootstrap sampling or null distributions. Methods implemented for linear models include the wild bootstrap by Wu (1986) <doi:10.1214/aos/1176350142>, the residual and paired bootstraps by Efron (1979, ISBN:978-1-4612-4380-9), the delete-1 jackknife by Quenouille (1956) <doi:10.2307/2332914>, and the Bayesian bootstrap by Rubin (1981) <doi:10.1214/aos/1176345338>.

**Depends** R (>= 3.5.0)

**Imports** evd (>= 2.3.0), stats (>= 3.6.0)

**License** GPL-2

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Megan Heyman [aut, cre]

**Maintainer** Megan Heyman <heyman@rose-hulman.edu>

**Repository** CRAN

**Date/Publication** 2019-06-03 13:10:11 UTC

## Contents

lmboot-package . . . . . . . . . . 2
ANOVA.boot . . . . . . . . . . 4
bayesian.boot . . . . . . . . . . 6
jackknife . . . . . . . . . . 7
paired.boot . . . . . . . . . . 8
residual.boot . . . . . . . . . . 10
wild.boot . . . . . . . . . . 11

**Index** **13**

## lmboot-package *Bootstrap in Linear Models*

### Description

Various efficient and robust bootstrap methods are implemented for linear models with least squares estimation. Functions within this package allow users to create bootstrap sampling distributions for model parameters, test hypotheses about parameters, and visualize the bootstrap sampling or null distributions. Methods implemented for linear models include the wild bootstrap by Wu (1986) <doi:10.1214/aos/1176350142>, the residual and paired bootstraps by Efron (1979, ISBN:978-1-4612-4380-9), the delete-1 jackknife by Quenouille (1956) <doi:10.2307/2332914>, and the Bayesian bootstrap by Rubin (1981) <doi:10.1214/aos/1176345338>.

### Details

| | |
|---|---|
| Package: | lmboot |
| Type: | Package |
| Title: | Bootstrap in Linear Models |
| Version: | 0.0.1 |
| Date: | 2019-05-13 |
| Authors@R: | person("Megan", "Heyman", email="heyman@rose-hulman.edu", role=c("aut","cre")) |
| Description: | Various efficient and robust bootstrap methods are implemented for linear models with least squares estimati |
| Depends: | R (>= 3.5.0) |
| Imports: | evd (>= 2.3.0), stats (>= 3.6.0) |
| License: | GPL-2 |
| RoxygenNote: | 6.1.1 |
| Encoding: | UTF-8 |
| Author: | Megan Heyman [aut, cre] |
| Maintainer: | Megan Heyman <heyman@rose-hulman.edu> |

Index of help topics:

```
ANOVA.boot              Residual and wild bootstrap in 1-way and 2-way
                        ANOVA
bayesian.boot           Bayesian Bootstrap in Linear Models
jackknife               Delete-1 Jackknife in Linear Models
lmboot-package          Bootstrap in Linear Models
paired.boot             Paired Bootstrap in Linear Models
residual.boot           Residual bootstrap in linear models
wild.boot               Wild Bootstrap in Linear Models
```

This package is useful to users who wish to perform bootstrap in linear models. The package contains functions to create the sampling distributions for linear model parameters using either efficient or robust bootstrap methods.

As classified by Liu and Singh (1992), efficient bootstrap types include the residual bootstrap (`residual.boot()`). These types of bootstrap are useful when it is not reasonable to assume that errors come from a normal distribution, but you may make other classical assumptions: errors are independent, have mean 0, and have constant variance.

Robust bootstrap types include the paired bootstrap (`paired.boot`), wild bootstrap (`wild.boot`), and the jackknife (`jackknife`). These types of bootstrap are useful when it is not reasonable to assumet that errors have constant variance, but you may make other classical assumptions: errors are independent and have mean 0.

The package also contains a function for Bayesian bootstrap (`bayesian.boot` and a function to perform bootstrap in the ANOVA hypothesis test (`ANOVA.boot`). The ANOVA bootstrap function has options to use the wild or residual bootstrap techniques and has been tested to work in 2-way ANOVA. Its functionality allows K-way ANOVA, however those capabilities have not been fully tested.

Currently, the user must manipulate the output of the function to conduct hypothesis tests and create confidence intervals for the predictor coefficients. More convenient/streamlined output is expected in future package versions.

## Author(s)

NA

Maintainer: NA

## References

Efron, B. (1979). "Bootstrap methods: Another look at the jackknife." *Annals of Statistics*. Vol. 7, pp.1-26.

Liu, R. Y. and Singh, K. (1992). "Efficiency and Robustness in Resampling." *Annals of Statistics*. Vol. 20, No. 1, pp.370-384.

Rubin, D. B. (1981). "The Bayesian Bootstrap." *Annals of Statistics*. Vol. 9, No. 1, pp.130-134.

Wu, C.F.J. (1986). "Jackknife, Bootstrap, and Other Resampling Methods in Regression Analysis." *Annals of Statistics*. Vol. 14, No. 4, pp.1261 - 1295.

## Examples

```
Seed <- 14
set.seed(Seed)
y <- rnorm(20) #randomly generated response
x <- rnorm(20) #randomly generated predictor

ResidObj <- residual.boot(y~x, B=100, seed=Seed) #perform the residual bootstrap
WildObj <- wild.boot(y~x, B=100, seed=Seed) #perform the wild bootstrap

#residual bootstrap 95% CI for slope parameter (percentile method)
quantile(ResidObj$bootEstParam[,2], probs=c(.025, .975))

#bootstrap 95% CI for slope parameter (percentile method)
quantile(WildObj$bootEstParam[,2], probs=c(.025, .975))
```

## `ANOVA.boot` *Residual and wild bootstrap in 1-way and 2-way ANOVA*

### Description

This function performs the residual bootstrap as described by Efron (1979) and wild bootstrap as described by Wu (1986) for ANOVA hypothesis testing. Linear models incorporating categorical and/or quantitative predictor variables with a quantitative response are allowed. The function output creates the bootstrap null distribution for each term to be tested. Estimation is performed via least squares and only Type I sum of squares are calculated.

### Usage

```
ANOVA.boot(formula, B = 1000, type = "residual", wild.dist = "normal",
            seed = NULL, data = NULL, keep.boot.resp = FALSE)
```

### Arguments

`formula` input a linear model formula of the form `response~predictors` as you would in the `lm()` function. All variables must contain non-missing entries.

`B` number of bootstrap samples. This should be a large, positive integer value.

`type` type of bootstrap to perform. Select either "residual" for residual bootstrap or "wild" for wild bootstrap.

`wild.dist` distribution used to create the wild bootstrap weights for the residuals. Allowed distributions include `"normal"`, `"uniform"`, `"exponential"`, `"laplace"`, `"lognormal"`, `"gumbel"`, `"t5"`, `"t8"`, and `"t14"`. The numbers after the t-distributions indicate the degrees of freedom. Any selected distribution creates weights with mean 0 and variance 1 from the named distribution.

`seed` optionally, set a value for the seed for the bootstrap sample generation. The default `NULL` will pick a random value for the seed.

`data` optionally, input the name of the dataset where variables appearing in the model are stored.

`keep.boot.resp` a boolean indicating whether the list of returns includes raw bootstrap responses. Setting this to TRUE may not be possible for larger datasets or too many bootstrap samples due to memory usage.

### Details

Currently, the user must manipulate the output of the function manually to view the bootstrap ANOVA table components and visualize the null distribution. More convenient/streamlined output is expected in future package versions.

Thanks to Bochuan Lyu who helped to coding to this function.

## Value

`terms` names of the terms/rows of the ANOVA table. These correspond to each predictor variable input to the formula.

`df` degrees of freedom associated with each term/row in the ANOVA table. These correspond to the number of categories in each predictor variable (or are 1 for quantitative predictors)

`origFStats` original F-statistic value. Same value as obtained by `aov()` using type I sum of squares.

`origSSE` original sum of squares, error. Same value as obtained by `aov()` using type I sum of squares.

`origSSTr` original sum of squares, treatment. Vector containing the sum of squares for each term in the ANOVA model. These are the same values as obtained by `aov()` using type I sum of squares.

`bootFStats` matrix containing the bootstrap F statistics. Each column corresponds to a term in the ANOVA table. There are `B` rows.

`bootSSE` matrix containing the bootstrap sum of squares, error. Each column corresponds to a term in the ANOVA table. There are `B` rows. These are calculated using type I sum of squares.

`bootSSTr` matrix containing the bootstrap sum of squares, treatment. Each column corresponds to a term in the ANOVA table. There are `B` rows. These are calculated using type I sum of squares.

`‘p-values‘` vector containing the bootstrap p-values for each predictor term in the ANOVA model. These are calculated by counting the number of bootstrap test statistics which are greater than the original observed test statistic and dividing by `B`

## Author(s)

Megan Heyman, heyman@rose-hulman.edu

## References

Efron, B. (1979). "Bootstrap methods: Another look at the jackknife." *Annals of Statistics*. Vol. 7, pp.1-26.

Wu, C.F.J. (1986). "Jackknife, Bootstrap, and Other Resampling Methods in Regression Analysis." *Annals of Statistics*. Vol. 14, No. 4, pp.1261 - 1295.

## See Also

`wild.boot`, `residual.boot`

## Examples

```
data(mtcars)         #load an example dataset
myANOVA2 <- ANOVA.boot(mpg~as.factor(cyl)*as.factor(am), data=mtcars)
myANOVA2$`p-values`  #bootstrap p-values for 2-way interactions model
```

```
myANOVA1 <- ANOVA.boot(mpg~as.factor(cyl), data=mtcars)
myANOVA1$`p-values` #bootstrap p-values for 1-way model

myANOVA2a <- ANOVA.boot(mpg~as.factor(cyl)+as.factor(am), data=mtcars)
myANOVA2a$`p-values` #bootstrap p-values for 1-way additive model
```

## `bayesian.boot` *Bayesian Bootstrap in Linear Models*

### Description

This function performs the bayesian bootstrap in linear models as described by Rubin (1981) <doi:10.1214/aos/1176345338>. Linear models incorporating categorical and/or quantitative predictor variables with a quantitative response are allowed. The function output creates the bootstrap sampling distribution for each coefficient. Estimation is performed via least squares.

### Usage

```
bayesian.boot(formula, B = 1000, seed = NULL, data = NULL)
```

### Arguments

| | |
|---|---|
| `formula` | input a linear model formula of the form `response~predictors` as you would in the `lm()` function. All variables must contain non-missing entries. |
| `B` | number of bootstrap samples. This should be a large, positive integer value. |
| `seed` | optionally, set a value for the seed for the bootstrap sample generation. The default `NULL` will pick a random value for the seed. |
| `data` | optionally, input the name of the dataset where variables appearing in the model are stored. |

### Details

Currently, the user must manipulate the output of the function to conduct hypothesis tests and create confidence intervals for the predictor coefficients. More convenient/streamlined output is expected in future package versions.

### Value

| | |
|---|---|
| `bootEstParam` | matrix containing the bootstrap parameter estimates. Each column corresponds to a coefficient. There are `B` rows, each corresponding to a bootstrap sample. |
| `origEstParam` | vector containing the least squares parameter estimates. These are the same as estimates obtained from `lm`. |
| `seed` | numerical value set for the seed. This is associated with the set of bootstrap parameter estimates and helps the process to be reproducible. |

### Author(s)

Megan Heyman, heyman@rose-hulman.edu

### References

Rubin, D. B. (1981). "The Bayesian Bootstrap." *Annals of Statistics*. Vol. 9, No. 1, pp.130-134.

### Examples

```
Seed <- 14
set.seed(Seed)
y <- rnorm(20) #randomly generated response
x <- rnorm(20) #randomly generated predictor
BayesObj <- bayesian.boot(y~x, B=100, seed=Seed) #perform the Bayesian bootstrap

#plot the sampling distribution of the slope coefficient
hist(BayesObj$bootEstParam[,2], main="Bayesian Bootstrap Sampling Distn.",
     xlab="Slope Estimate")

#bootstrap 95% CI for slope parameter (percentile method)
quantile(BayesObj$bootEstParam[,2], probs=c(.025, .975))
```

## jackknife *Delete-1 Jackknife in Linear Models*

### Description

This function performs the delete-1 jackknife in linear models as described by Quenouille (1956) <doi:10.2307/2332914>. Linear models incorporating categorical and/or quantitative predictor variables with a quantitative response are allowed. The function output creates the jackknife sampling distribution for each coefficient. Estimation is performed via least squares.

### Usage

```
jackknife(formula, data = NULL)
```

### Arguments

| | |
|---|---|
| `formula` | input a linear model formula of the form `response~predictors` as you would in the `lm()` function. All variables must contain non-missing entries. |
| `data` | optionally, input the name of the dataset where variables appearing in the model are stored. |

### Details

Currently, the user must manipulate the output of the function to conduct hypothesis tests and create confidence intervals for the predictor coefficients. More convenient/streamlined output is expected in future package versions.

### Value

bootEstParam matrix containing the jackknife parameter estimates. Each column corresponds to a coefficient. There are n-1 rows, each corresponding to a jackknife sample.

origEstParam vector containing the least squares parameter estimates. These are the same as estimates obtained from lm.

### Author(s)

Megan Heyman, heyman@rose-hulman.edu

### References

Quenouille, M. (1956). "Notes on bias in estimation." *Biometrika*. Vol. 61, pp.1-15

### Examples

```
Seed <- 14
set.seed(Seed)
y <- rnorm(20) #randomly generated response
x <- rnorm(20) #randomly generated predictor
JackObj <- jackknife(y~x) #perform the jackknife

#plot the sampling distribution of the slope coefficient
hist(JackObj$bootEstParam[,2], main="Jackknife Sampling Distn.",
     xlab="Slope Estimate")

#jackknife 95% CI for slope parameter (percentile method)
quantile(JackObj$bootEstParam[,2], probs=c(.025, .975))
```

## paired.boot *Paired Bootstrap in Linear Models*

### Description

This function performs the paired bootstrap in linear models as described by Efron (1979, ISBN:978-1-4612-4380-9). Linear models incorporating categorical and/or quantitative predictor variables with a quantitative response are allowed. The function output creates the bootstrap sampling distribution for each coefficient. Estimation is performed via least squares.

### Usage

```
paired.boot(formula, B = 1000, seed = NULL, data = NULL)
```

## Arguments

`formula` input a linear model formula of the form `response~predictors` as you would in the `lm()` function. All variables must contain non-missing entries.

`B` number of bootstrap samples. This should be a large, positive integer value.

`seed` optionally, set a value for the seed for the bootstrap sample generation. The default `NULL` will pick a random value for the seed.

`data` optionally, input the name of the dataset where variables appearing in the model are stored.

## Details

Currently, the user must manipulate the output of the function to conduct hypothesis tests and create confidence intervals for the predictor coefficients. More convenient/streamlined output is expected in future package versions.

## Value

`bootEstParam` matrix containing the bootstrap parameter estimates. Each column corresponds to a coefficient. There are `B` rows, each corresponding to a bootstrap sample.

`origEstParam` vector containing the least squares parameter estimates. These are the same as estimates obtained from `lm`.

`seed` numerical value set for the seed. This is associated with the set of bootstrap parameter estimates and helps the process to be reproducible.

## Author(s)

Megan Heyman, heyman@rose-hulman.edu

## References

Efron, B. (1979). "Bootstrap methods: Another look at the jackknife." *Annals of Statistics*. Vol. 7, pp.1-26.

## Examples

```
Seed <- 14
set.seed(Seed)
y <- rnorm(20) #randomly generated response
x <- rnorm(20) #randomly generated predictor
PairObj <- paired.boot(y~x, B=100, seed=Seed) #perform the paired bootstrap

#plot the sampling distribution of the slope coefficient
hist(PairObj$bootEstParam[,2], main="Paired Bootstrap Sampling Distn.",
     xlab="Slope Estimate")

#bootstrap 95% CI for slope parameter (percentile method)
quantile(PairObj$bootEstParam[,2], probs=c(.025, .975))
```

## `residual.boot` *Residual bootstrap in linear models*

### Description

This function performs the residual bootstrap in linear models as described by Efron (1979, ISBN:978-1-4612-4380-9). Linear models incorporating categorical and/or quantitative predictor variables with a quantitative response are allowed. The function output creates the bootstrap sampling distribution for each coefficient. Estimation is performed via least squares.

### Usage

```
residual.boot(formula, B = 1000, data = NULL, seed = NULL)
```

### Arguments

`formula` input a linear model formula of the form `response~predictors` as you would in the `lm()` function. All variables must contain non-missing entries.

`B` number of bootstrap samples. This should be a large, positive integer value.

`data` optionally, input the name of the dataset where variables appearing in the model are stored.

`seed` optionally, set a value for the seed for the bootstrap sample generation. The default `NULL` will pick a random value for the seed.

### Details

Currently, the user must manipulate the output of the function to conduct hypothesis tests and create confidence intervals for the predictor coefficients. More convenient/streamlined output is expected in future package versions.

### Value

`bootEstParam` matrix containing the bootstrap parameter estimates. Each column corresponds to a coefficient. There are `B` rows, each corresponding to a bootstrap sample.

`origEstParam` vector containing the least squares parameter estimates. These are the same as estimates obtained from `lm`.

`seed` numerical value set for the seed. This is associated with the set of bootstrap parameter estimates and helps the process to be reproducible.

### Author(s)

Megan Heyman, heyman@rose-hulman.edu

### References

Efron, B. (1979). "Bootstrap methods: Another look at the jackknife." *Annals of Statistics*. Vol. 7, pp.1-26.

## Examples

```
Seed <- 14
set.seed(Seed)
y <- rnorm(20) #randomly generated response
x <- rnorm(20) #randomly generated predictor
ResidObj <- residual.boot(y~x, B=100, seed=Seed) #perform the residual bootstrap

#plot the sampling distribution of the slope coefficient
hist(ResidObj$bootEstParam[,2], main="Residual Bootstrap Sampling Distn.",
     xlab="Slope Estimate")

#bootstrap 95% CI for slope parameter (percentile method)
quantile(ResidObj$bootEstParam[,2], probs=c(.025, .975))
```

---

# wild.boot *Wild Bootstrap in Linear Models*

---

## Description

This function performs the wild/external bootstrap in linear models as described by Wu (1986) <doi:10.1214/aos/1176350142>. Linear models incorporating categorical and/or quantitative predictor variables with a quantitative response are allowed. The function output creates the bootstrap sampling distribution for each coefficient. Estimation is performed via least squares.

## Usage

```
wild.boot(formula, B = 1000, data = NULL, seed = NULL, bootDistn = "normal")
```

## Arguments

| | |
|---|---|
| `formula` | input a linear model formula of the form `response~predictors` as you would in the `lm()` function. All variables must contain non-missing entries. |
| `B` | number of bootstrap samples. This should be a large, positive integer value. |
| `data` | optionally, input the name of the dataset where variables appearing in the model are stored. |
| `seed` | optionally, set a value for the seed for the bootstrap sample generation. The default `NULL` will pick a random value for the seed. |
| `bootDistn` | distribution used to create the wild bootstrap weights for the residuals. Allowed distributions include `"normal"`, `"uniform"`, `"exponential"`, `"laplace"`, `"lognormal"`, `"gumbel"`, `"t5"`, `"t8"`, and `"t14"`. The numbers after the t-distributions indicate the degrees of freedom. Any selected distribution creates weights with mean 0 and variance 1 from the named distribution. |

## Details

Currently, the user must manipulate the output of the function to conduct hypothesis tests and create confidence intervals for the predictor coefficients. More convenient/streamlined output is expected in future package versions.

## Value

`bootEstParam` matrix containing the bootstrap parameter estimates. Each column corresponds to a coefficient. There are `B` rows, each corresponding to a bootstrap sample.

`origEstParam` vector containing the least squares parameter estimates. These are the same as estimates obtained from `lm`.

`seed` numerical value set for the seed. This is associated with the set of bootstrap parameter estimates and helps the process to be reproducible.

`bootDistn` type of distribution used to generate the wild bootstrap weights for the residuals

## Author(s)

Megan Heyman, heyman@rose-hulman.edu

## References

Wu, C.F.J. (1986). "Jackknife, Bootstrap, and Other Resampling Methods in Regression Analysis." *Annals of Statistics*. Vol. 14, No. 4, pp.1261 - 1295.

## Examples

```
Seed <- 14
set.seed(Seed)
y <- rnorm(20) #randomly generated response
x <- rnorm(20) #randomly generated predictor
WildObj <- wild.boot(y~x, B=100, seed=Seed) #perform the wild bootstrap

#plot the sampling distribution of the slope coefficient
hist(WildObj$bootEstParam[,2], main="Wild Bootstrap Sampling Distn.",
     xlab="Slope Estimate")

#bootstrap 95% CI for slope parameter (percentile method)
quantile(WildObj$bootEstParam[,2], probs=c(.025, .975))
```

# Index

∗ **distribution**
    bayesian.boot, 6
    wild.boot, 11
∗ **models regression**
    ANOVA.boot, 4
    bayesian.boot, 6
    jackknife, 7
    paired.boot, 8
    residual.boot, 10
    wild.boot, 11
∗ **nonparametric**
    ANOVA.boot, 4
    bayesian.boot, 6
    jackknife, 7
    paired.boot, 8
    residual.boot, 10
    wild.boot, 11
∗ **package**
    lmboot-package, 2

ANOVA.boot, 4

bayesian.boot, 6

jackknife, 7

lmboot *(*lmboot-package*)*, 2
lmboot-package, 2

paired.boot, 8

residual.boot, *5*, 10

wild.boot, *5*, 11