

# Package ‘simpleboot’

April 19, 2009

**Version** 1.1-3

**Date** 2008-04-30

**Depends** R (>= 2.7.0), boot

**Imports** stats, graphics

**Title** Simple Bootstrap Routines

**Author** Roger D. Peng <rpeng@jhsph.edu>

**Maintainer** Roger D. Peng <rpeng@jhsph.edu>

**Description** Simple bootstrap routines

**License** GPL (>= 2)

**Repository** CRAN

**Date/Publication** 2008-05-02 08:59:45

## R topics documented:

hist.simpleboot . . . . .	2
lm.boot . . . . .	3
lm.simpleboot.methods . . . . .	4
loess.boot . . . . .	5
loess.simpleboot.methods . . . . .	7
one.boot . . . . .	8
pairs.boot . . . . .	9
perc . . . . .	10
plot.lm.simpleboot . . . . .	11
plot.loess.simpleboot . . . . .	12
samples . . . . .	13
two.boot . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

hist.simpleboot      *Histograms for bootstrap sampling distributions.*

---

## Description

Construct a histogram of the bootstrap distribution of univariate statistic.

## Usage

```
## S3 method for class 'simpleboot':  
hist(x, do.rug = FALSE, xlab = "Bootstrap samples", main = "", ...)
```

## Arguments

x	An object of class "simpleboot" returned from either one.boot, two.boot, or pairs.boot.
do.rug	Should a rug of the bootstrap distribution be plotted under the histogram?
xlab	The label for the x-axis.
main	The title for the histogram.
...	Other arguments passed to hist.

## Details

hist constructs a histogram for the bootstrap distribution of a univariate statistic. It cannot be used with linear model or loess bootstraps. In the histogram a red dotted line is plotted denoting the observed value of the statistic.

## Value

Nothing is returned.

## Author(s)

Roger D. Peng

## Examples

```
x <- rnorm(100)  
  
## Bootstrap the 75th percentile  
b <- one.boot(x, quantile, R = 1000, probs = 0.75)  
hist(b)
```

---

lm.boot                      *Linear model bootstrap.*

---

### Description

Bootstrapping of linear model fits (using `lm`). Bootstrapping can be done by either resampling rows of the original data frame or resampling residuals from the original model fit.

### Usage

```
lm.boot(lm.object, R, rows = TRUE, new.xpts = NULL, ngrid = 100,
        weights = NULL)
```

### Arguments

<code>lm.object</code>	A linear model fit, produced by <code>lm</code> .
<code>R</code>	The number of bootstrap replicates to use.
<code>rows</code>	Should we resample rows? Setting <code>rows</code> to <code>FALSE</code> indicates resampling of residuals.
<code>new.xpts</code>	Values at which you wish to make new predictions. If specified, fitted values from each bootstrap sample will be stored.
<code>ngrid</code>	If <code>new.xpts</code> is <code>NULL</code> and the regression is 2 dimensional, then predictions are made on an evenly spaced grid (containing <code>ngrid</code> points) spanning the range of the predictor values.
<code>weights</code>	Resampling weights; a vector of length equal to the number of observations.

### Details

Currently, "`lm.simpleboot`" objects have a simple `print` method (which shows the original fit), a `summary` method and a `plot` method.

### Value

An object of class "`lm.simpleboot`" (which is a list) containing the elements:

<code>method</code>	Which method of bootstrapping was used (rows or residuals).
<code>boot.list</code>	A list containing values from each of the bootstrap samples. Currently, bootstrapped values are model coefficients, residual sum of squares, R-square, and fitted values for predictions.
<code>orig.lm</code>	The original model fit.
<code>new.xpts</code>	The locations where predictions were made.
<code>weights</code>	The resampling weights. If none were used, this component is <code>NULL</code>

### Author(s)

Roger D. Peng

**See Also**

The `plot.lm.simpleboot` method.

**Examples**

```
data(airquality)
attach(airquality)
set.seed(30)
lmodel <- lm(Ozone ~ Wind)
lboot <- lm.boot(lmodel, R = 1000)
summary(lboot)

## With weighting
w <- runif(nrow(model.frame(lmodel)))
lbootw <- lm.boot(lmodel, R = 1000, weights = w)
summary(lbootw)

## Resample residuals
lboot2 <- lm.boot(lmodel, R = 1000, rows = FALSE)
summary(lboot2)
```

---

lm.simpleboot.methods

*Methods for linear model bootstrap.*

---

**Description**

Methods for "lm.simpleboot" class objects.

**Usage**

```
## S3 method for class 'lm.simpleboot':
summary(object, ...)
## S3 method for class 'summary.lm.simpleboot':
print(x, ...)
## S3 method for class 'lm.simpleboot':
fitted(object, ...)
```

**Arguments**

<code>object</code>	An object of class "lm.simpleboot", as returned by <code>lm.boot</code> .
<code>x</code>	An object of class "summary.lm.simpleboot", a result of a call to <code>summary.lm.simpleboot</code> .
<code>...</code>	Other arguments passed to and from other methods.

## Details

`print` is essentially the same as the usual printing of a linear model fit, except the bootstrap standard errors are printed for each model coefficient.

`fitted` returns the fitted values from each bootstrap sample for the predictor values specified by the `new.xpts` in `lm.boot` (or from the grid if `new.xpts` was not specified). This is a  $p \times R$  matrix where  $p$  is the number of points where prediction was desired and  $R$  is the number of bootstrap samples specified. Using `fitted` is the equivalent of using `samples(object, name = "fitted")`.

## Value

`summary` returns a list containing the original estimated coefficients and their bootstrap standard errors.

## Author(s)

Roger D. Peng

## See Also

`lm.boot`.

## Examples

```
data(airquality)
attach(airquality)
lmodel <- lm(Ozone ~ Wind + Solar.R)
lboot <- lm.boot(lmodel, R = 300)
summary(lboot)
```

---

`loess.boot`

*2-D Loess bootstrap.*

---

## Description

Bootstrapping of loess fits produced by the `loess` function in the `modreg` package. Bootstrapping can be done by resampling rows from the original data frame or resampling residuals from the original model fit.

## Usage

```
loess.boot(lo.object, R, rows = TRUE, new.xpts = NULL, ngrid = 100,
           weights = NULL)
```

**Arguments**

<code>lo.object</code>	A loess fit, produced by <code>loess</code> .
<code>R</code>	The number of bootstrap replicates.
<code>rows</code>	Should we resample rows? Setting <code>rows</code> to <code>FALSE</code> indicates resampling of residuals.
<code>new.xpts</code>	Locations where new predictions are to be made. If <code>new.xpts</code> is <code>NULL</code> , then an evenly spaced grid spanning the range of <code>X</code> (containing <code>ngrid</code> points) is used. In either case
<code>ngrid</code>	Number of grid points to use if <code>new.xpts</code> is <code>NULL</code> .
<code>weights</code>	Resampling weights; a vector with length equal to the number of observations.

**Details**

The user can specify locations for new predictions through `new.xpts` or an evenly spaced grid will be used. In either case, fitted values at each new location will be stored from each bootstrap sample. These fitted values can be retrieved using either the `fitted` method or the `samples` function.

Note that the `loess` function has many parameters for the user to set that can be difficult to reproduce in the bootstrap setting. Right now, the user can only specify the `span` argument to `loess` in the original fit.

**Value**

An object of class "`loess.simpleboot`" (which is a list) containing the elements:

<code>method</code>	Which method of bootstrapping was used (rows or residuals).
<code>boot.list</code>	A list containing values from each of the bootstrap samples. Currently, only residual sum of squares and fitted values are stored.
<code>orig.loess</code>	The original loess fit.
<code>new.xpts</code>	The locations where predictions were made (specified in the original call to <code>loess.boot</code> ).

**Author(s)**

Roger D. Peng

**Examples**

```
set.seed(1234)

x <- runif(100)

## Simple sine function simulation
y <- sin(2*pi*x) + .2 * rnorm(100)
plot(x, y) ## Sine function with noise
lo <- loess(y ~ x, span = .4)
```

```
## Bootstrap with resampling of rows
lo.b <- loess.boot(lo, R = 500)

## Plot original fit with +/- 2 std. errors
plot(lo.b)

## Plot all loess bootstrap fits
plot(lo.b, all.lines = TRUE)

## Bootstrap with resampling residuals
lo.b2 <- loess.boot(lo, R = 500, rows = FALSE)
plot(lo.b2)
```

---

loess.simpleboot.methods

*Methods for loess bootstrap.*

---

## Description

Methods for "loess.simpleboot" class objects.

## Usage

```
## S3 method for class 'loess.simpleboot':
fitted(object, ...)
```

## Arguments

object	An object of class "loess.simpleboot" as returned by the function loess.boot.
...	Other arguments passed to and from other methods.

## Details

fitted returns a  $n \times R$  matrix of fitted values where  $n$  is the number of new locations at which predictions were made and  $R$  is the number of bootstrap replications used in the original loess bootstrap. This is the equivalent of calling `samples(object, "fitted")`.

## Value

Nothing is returned.

## Author(s)

Roger D. Peng

---

 one.boot

*One sample bootstrap of a univariate statistic.*


---

### Description

one.boot is used for bootstrapping a univariate statistic for one sample problems. Examples include the mean, median, etc.

### Usage

```
one.boot(data, FUN, R, student = FALSE, M, weights = NULL, ...)
```

### Arguments

data	The data. This should be a vector of numbers.
FUN	The statistic to be bootstrapped. This can be either a quoted string containing the name of a function or simply the function name.
R	The number of bootstrap replicates to use.
student	Should we do a studentized bootstrap? This requires a double bootstrap so it might take longer.
M	If student is set to TRUE, then M is the number of internal bootstrap replications to do.
weights	Resampling weights; a vector of length equal to the number of observations.
...	Other (named) arguments that should be passed to FUN.

### Value

An object of class "simpleboot", which is almost identical to the regular "boot" object. For example, the boot.ci function can be used on this object.

### Author(s)

Roger D. Peng

### Examples

```
set.seed(20)
x <- rgamma(100, 1)
b.mean <- one.boot(x, mean, 1000)
print(b.mean)
boot.ci(b.mean) ## No studentized interval here
hist(b.mean)

## This next line could take some time on a slow computer
b.median <- one.boot(x, median, R = 500, student = TRUE, M = 50)
boot.ci(b.median)
hist(b.median)
```

```
## Bootstrap with weights
set.seed(10)
w <- runif(100)
bw <- one.boot(x, median, 1000, weights = w)
print(bw)

## Studentized
bw.stud <- one.boot(x, median, R = 500, student = TRUE, M = 50,
                    weights = w)
boot.ci(bw.stud, type = "stud")
```

---

pairs.boot                      *Two sample bootstrap.*

---

### Description

`pairs.boot` is used to bootstrap a statistic which operates on two samples and returns a single value. An example of such a statistic is the correlation coefficient (i.e. `cor`). Resampling is done pairwise, so `x` and `y` must have the same length (and be ordered correctly). One can alternatively pass a two-column matrix to `x`.

### Usage

```
pairs.boot(x, y = NULL, FUN, R, student = FALSE, M, weights = NULL, ...)
```

### Arguments

<code>x</code>	Either a vector of numbers representing the first sample or a two column matrix containing both samples.
<code>y</code>	If <code>NULL</code> it is assumed that <code>x</code> is a two-column matrix. Otherwise, <code>y</code> is the second sample.
<code>FUN</code>	The statistic to bootstrap. If <code>x</code> and <code>y</code> are separate vectors then <code>FUN</code> should operate on separate vectors. Similarly, if <code>x</code> is a matrix, then <code>FUN</code> should operate on two-column matrices. <code>FUN</code> can be either a quoted string or a function name.
<code>R</code>	The number of bootstrap replicates.
<code>student</code>	Should we do a studentized bootstrap? This requires a double bootstrap so it might take longer.
<code>M</code>	If <code>student</code> is set to <code>TRUE</code> , then <code>M</code> is the number of internal bootstrap replications to do.
<code>weights</code>	Resampling weights.
<code>...</code>	Other (named) arguments that should be passed to <code>FUN</code> .

### Value

An object of class `"simpleboot"`, which is almost identical to the regular `"boot"` object. For example, the `boot.ci` function can be used on this object.

**Author(s)**

Roger D. Peng

**Examples**

```

set.seed(1)
x <- rnorm(100)
y <- 2 * x + rnorm(100)
boot.cor <- pairs.boot(x, y, FUN = cor, R = 1000)
boot.ci(boot.cor)

## With weighting
set.seed(20)
w <- (100:1)^2
bw <- pairs.boot(x, y, FUN = cor, R = 5000, weights = w)
boot.ci(bw, type = c("norm", "basic", "perc"))

```

perc

*Extract percentiles from a bootstrap sampling distribution.***Description**

perc can be used to extract percentiles from the sampling distribution of a statistic.

**Usage**

```

perc(boot.out, p = c(0.025, 0.975))
perc.lm(lm.boot.obj, p)

```

**Arguments**

boot.out      Output from either one.boot, two.boot, or pairs.boot.  
p              numeric vector with values in [0, 1].  
lm.boot.obj    An object of class "lm.simpleboot", returned from lm.boot.

**Details**

perc automatically calls perc.lm if boot.out is of the class "lm.simpleboot" so there is no need to use perc.lm separately.

**Value**

For bootstraps which are not linear model bootstraps, perc returns a vector of percentiles of length length(p). Linear interpolation of percentiles is done if necessary. perc.lm returns a matrix of percentiles of each of the model coefficients. For example, if there are k model coefficients, the perc.lm returns a length(p) by k matrix.

**Author(s)**

Roger D. Peng

**Examples**

```
x <- rnorm(100)
b <- one.boot(x, median, R = 1000)
perc(b, c(.90, .95, .99))
```

---

```
plot.lm.simpleboot Plot method for linear model bootstraps.
```

---

**Description**

Plot regression lines with bootstrap standard errors. This method only works for 2-D regression fits.

**Usage**

```
## S3 method for class 'lm.simpleboot':
plot(x, add = FALSE, ...)
```

**Arguments**

x	An object of class "lm.simpleboot" returned by <code>lm.boot</code> .
add	Switch indicating whether the regression line should be added to the current plot.
...	Additional arguments passed down to <code>plot</code> . Ignored if <code>add = TRUE</code> .

**Details**

This function plots the data and the original regression line fit along with  $\pm 2$  bootstrap standard errors at locations specified by the `new.xpts` argument to `lm.boot` (or on an evenly spaced grid).

**Value**

Nothing is returned.

**Author(s)**

Roger D. Peng

**Examples**

```
## None right now
```

---

```
plot.loess.simpleboot
```

*Plot method for loess bootstraps.*

---

### Description

Plot loess lines with bootstrap standard errors.

### Usage

```
## S3 method for class 'loess.simpleboot':  
plot(x, add = FALSE, all.lines = FALSE, ...)
```

### Arguments

<code>x</code>	An object of class "loess.simpleboot" as returned by the function <code>loess.boot</code> .
<code>add</code>	Should the loess line be plotted over the current plot?
<code>all.lines</code>	Should we plot each of the individual loess lines from the bootstrap samples?
<code>...</code>	Other arguments passed to <code>plot</code> .

### Details

`plot` constructs (and plots) the original loess fit and  $\pm 2$  bootstrap standard errors at locations specified in the `new.xpts` in `loess.boot` (or on an evenly spaced grid).

### Value

Nothing is returned.

### Author(s)

Roger D. Peng

### Examples

```
## See the help page for `loess.boot' for an example.
```

---

`samples`*Extract sampling distributions from bootstrapped linear/loess models.*

---

**Description**

Extract sampling distributions of various entities from either a linear model or a loess bootstrap. Entities for linear models are currently, model coefficients, residual sum of squares, R-square, and fitted values (given a set of X values in the original bootstrap). For loess, one can extract residual sum of squares and fitted values.

**Usage**

```
samples(object, name = c("fitted", "coef", "rsquare", "rss"))
```

**Arguments**

<code>object</code>	The output from either <code>lm.boot</code> or <code>loess.boot</code> .
<code>name</code>	The name of the entity to extract. The default is fitted values.

**Value**

Either a vector or matrix depending on the entity extracted. For example, when extracting the sampling distributions for linear model coefficients, the return value is  $p \times R$  matrix where  $p$  is the number of coefficients and  $R$  is the number of bootstrap replicates.

**Author(s)**

Roger D. Peng

**Examples**

```
data(airquality)
attach(airquality)
lmodel <- lm(Ozone ~ Solar.R + Wind)
lboot <- lm.boot(lmodel, R = 500)

## Get sampling distributions for coefficients
s <- samples(lboot, "coef")

## Histogram for the intercept
hist(s[1,])
```

---

`two.boot`*Two sample bootstrap of differences between univariate statistics.*

---

### Description

`two.boot` is used to bootstrap the difference between various univariate statistics. An example is the difference of means. Bootstrapping is done by independently resampling from `sample1` and `sample2`.

### Usage

```
two.boot(sample1, sample2, FUN, R, student = FALSE, M, weights = NULL, ...)
```

### Arguments

<code>sample1</code>	First sample; a vector of numbers.
<code>sample2</code>	Second sample; a vector of numbers.
<code>FUN</code>	The statistic which is applied to each sample. This can be a quoted string or a function name.
<code>R</code>	Number of bootstrap replicates.
<code>student</code>	Should we do a studentized bootstrap? This requires a double bootstrap so it might take longer.
<code>M</code>	If <code>student</code> is set to <code>TRUE</code> , then <code>M</code> is the number of internal bootstrap replications to do.
<code>weights</code>	Resampling weights; a list with two components. The first component of the list is a vector of weights for <code>sample1</code> and the second component of the list is a vector of weights for <code>sample2</code> .
<code>...</code>	Other (named) arguments that should be passed to <code>FUN</code> .

### Details

The differences are always taken as `FUN(sample1) - FUN(sample2)`. If you want the difference to be reversed you need to reverse the order of the arguments `sample1` and `sample2`.

### Value

An object of class `"simpleboot"`, which is almost identical to the regular `"boot"` object. For example, the `boot.ci` function can be used on this object.

### Author(s)

Roger D. Peng

**Examples**

```
set.seed(50)
x <- rnorm(100, 1) ## Mean 1 normals
y <- rnorm(100, 0) ## Mean 0 normals
b <- two.boot(x, y, median, R = 1000)
boot.ci(b) ## No studentized confidence intervals
hist(b) ## Histogram of the bootstrap replicates

b <- two.boot(x, y, quantile, R = 1000, probs = .75)

## With weighting

## Here all members of the first group has equal weighting
## but members of the the second have unequal weighting
w <- list(rep(1, 100), 100:1)
bw <- two.boot(x, y, median, R = 1000, weights = w)
boot.ci(b)

## Studentized
bstud <- two.boot(x, y, median, R = 500, student = TRUE, M = 50)
boot.ci(bstud, type = "stud")

## Studentized with weights
bwstud <- two.boot(x, y, median, R = 500, student = TRUE, M = 50,
                  weights = w)
boot.ci(bstud, type = "stud")
```

# Index

- \*Topic **hplot**
  - hist.simpleboot, 1
- \*Topic **loess**
  - loess.boot, 5
  - loess.simpleboot.methods, 7
  - plot.loess.simpleboot, 12
- \*Topic **regression**
  - lm.boot, 2
  - lm.simpleboot.methods, 4
  - plot.lm.simpleboot, 11
- \*Topic **univar**
  - one.boot, 7
  - pairs.boot, 9
  - perc, 10
  - two.boot, 14
- \*Topic **utilities**
  - samples, 13

fitted.lm.simpleboot  
(*lm.simpleboot.methods*), 4

fitted.loess.simpleboot  
(*loess.simpleboot.methods*),  
7

hist.simpleboot, 1

lm.boot, 2  
lm.simpleboot.methods, 4  
loess.boot, 5  
loess.simpleboot.methods, 7

one.boot, 7

pairs.boot, 9  
perc, 10  
plot.lm.simpleboot, 11  
plot.loess.simpleboot, 12  
print.lm.simpleboot (*lm.boot*), 2  
print.loess.simpleboot  
(*loess.boot*), 5

print.summary.lm.simpleboot  
(*lm.simpleboot.methods*), 4

samples, 13  
summary.lm.simpleboot  
(*lm.simpleboot.methods*), 4

two.boot, 14