

Package ‘wBoot’

January 27, 2016

Type Package

Title Bootstrap Methods

Version 1.0.3

Date 2016-01-26

Author Neil A. Weiss

Maintainer Neil A. Weiss <neil.a.weiss@asu.edu>

Description Supplies bootstrap alternatives to traditional hypothesis-test and confidence-interval procedures such as one-sample and two-sample inferences for means, medians, standard deviations, and proportions; simple linear regression; and more. Suitable for general audiences, including individual and group users, introductory statistics courses, and more advanced statistics courses that desire an introduction to bootstrap methods.

License GPL (>= 2)

Depends R (>= 3.1.0), boot, simpleboot

NeedsCompilation no

Repository CRAN

Date/Publication 2016-01-27 18:18:43

R topics documented:

wBoot-package	2
boot.cond.mean.bca	2
boot.cond.mean.per	4
boot.cor.bca	6
boot.cor.per	8
boot.one.bca	10
boot.one.per	12
boot.paired.bca	14
boot.paired.per	16
boot.ratio.sd.bca	18
boot.ratio.sd.per	20
boot.slope.bca	23

boot.slope.per	25
boot.two.bca	27
boot.two.per	29
elmendorf	31
fonts	32
homes	33
loss	34
print.boot.one	35
print.boot.paired	35
print.boot.regcor	36
print.boot.two	37
tees	38

Index 39

wBoot-package *Bootstrap Methods*

Description

Supplies bootstrap alternatives to traditional hypothesis-test and confidence-interval procedures such as one-sample and two-sample inferences for means, medians, standard deviations, and proportions; simple linear regression; and more. Suitable for general audiences, including individual and group users, introductory statistics courses, and more advanced statistics courses that desire an introduction to bootstrap methods.

Details

Package: wBoot
 Type: Package
 Version: 1.0.3
 Date: 2016-01-26
 License: GPL (>= 2)

Author(s)

Neil A. Weiss
 Maintainer: Neil A. Weiss <neil.a.weiss@asu.edu>

boot.cond.mean.bca *BCa Bootstrap Conditional Mean CI in Simple Linear Regression*

Description

Determines a confidence interval for a conditional mean in simple linear regression, using the BCa bootstrap method.

Usage

```
boot.cond.mean.bca(x, y, xp, conf.level = 0.95, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of predictor-variable data values.
y	the corresponding numeric vector of response-variable data values.
xp	the value of the predictor variable for which to find the CI for the conditional mean of the response variable.
conf.level	the confidence level (between 0 and 1); default is 0.95.
R	the number of bootstrap replications; default is 9999.

Value

A list with class "boot.regcor" containing the following components:

Boot.values	the point estimates (fits) obtained from the bootstrap.
Confidence.limits	the upper and lower limits of the confidence interval.
Header	the main title for the output.
Variable.1	the predictor variable.
Variable.2	the response variable.
n	the sample size.
Statistic	the name of the statistic, here fit.
Observed	the observed point estimate (fit).
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * \text{Bias} / \text{Observed}$.
Null	always NULL for this function.
Alternative	always NULL for this function.
P.value	always NULL for this function.
p.value	always NULL for this function.
Level	the confidence level.
Type	always NULL for this function.
Confidence.interval	the confidence interval.
cor.ana	a logical; always FALSE for this function.

Author(s)

Neil A. Weiss

Examples

```
# Lot size, house size, and value for a sample of homes in a particular area.
data("homes")
str(homes)
attach(homes)

# 95% (default) CI for the conditional mean value of a 3000 sq.ft. home,
# with 999 bootstrap replications.
boot.cond.mean.bca(HOUSE.SIZE, VALUE, 3000, R = 999)

# 90% CI for the conditional mean value of a 3000 sq.ft. home, with
# 999 bootstrap replications.
boot.cond.mean.bca(HOUSE.SIZE, VALUE, 3000, conf.level = 0.90, R = 999)

detach(homes) # clean up
```

boot.cond.mean.per	<i>Percentile Bootstrap Conditional Mean CI in Simple Linear Regression</i>
--------------------	---

Description

Determines a confidence interval for a conditional mean in simple linear regression, using the percentile bootstrap method.

Usage

```
boot.cond.mean.per(x, y, xp, conf.level = 0.95, R = 9999)
```

Arguments

x	a (nonempty) numeric vector of predictor-variable data values.
y	the corresponding numeric vector of response-variable data values.
xp	the value of the predictor variable for which to find the CI for the conditional mean of the response variable.
conf.level	the confidence level (between 0 and 1); default is 0.95.
R	the number of bootstrap replications; default is 9999.

Value

A list with class "boot.regcor" containing the following components:

<code>Boot.values</code>	the point estimates (fits) obtained from the bootstrap.
<code>Confidence.limits</code>	the upper and lower limits of the confidence interval.
<code>Header</code>	the main title for the output.
<code>Variable.1</code>	the predictor variable.
<code>Variable.2</code>	the response variable.
<code>n</code>	the sample size.
<code>Statistic</code>	the name of the statistic, here fit.
<code>Observed</code>	the observed point estimate (fit).
<code>Replications</code>	the number of bootstrap replications.
<code>Mean</code>	the mean of the bootstrap values.
<code>SE</code>	the standard deviation of the bootstrap values.
<code>Bias</code>	the difference between the mean of the bootstrap values and the observed value.
<code>Percent.bias</code>	the percentage bias: $100 * Bias/Observed $.
<code>Null</code>	always NULL for this function.
<code>Alternative</code>	always NULL for this function.
<code>P.value</code>	always NULL for this function.
<code>p.value</code>	always NULL for this function.
<code>Level</code>	the confidence level.
<code>Type</code>	always NULL for this function.
<code>Confidence.interval</code>	the confidence interval.
<code>cor.ana</code>	a logical; always FALSE for this function.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# Lot size, house size, and value for a sample of homes in a particular area.
data("homes")
str(homes)
attach(homes)

# 95% (default) CI for the conditional mean value of a 3000 sq.ft. home,
# with 999 bootstrap replications.
boot.cond.mean.per(HOUSE.SIZE, VALUE, 3000, R = 999)

# 90% CI for the conditional mean value of a 3000 sq.ft. home, with
# 999 bootstrap replications.
boot.cond.mean.per(HOUSE.SIZE, VALUE, 3000, conf.level = 0.90, R = 999)

detach(homes) # clean up
```

boot.cor.bca

BCa Bootstrap Correlation Test and CI

Description

Obtains a confidence interval and (optionally) performs a hypothesis test for the Pearson correlation, using the BCa bootstrap method.

Usage

```
boot.cor.bca(x, y, null.hyp = NULL,
             alternative = c("two.sided", "less", "greater"),
             conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of data values.
y	a (non-empty) numeric vector of data values.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two-sided", "upper-bound", or "lower-bound"; defaults to "two-sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Value

A list with class "boot.regcor" containing the following components:

Boot.values	the point estimates (correlations) obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable.1	the first variable.
Variable.2	the second variable.
n	the sample size.
Statistic	the name of the statistic, here correlation.
Observed	the observed point estimate (correlation).
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.
cor.ana	a logical; always TRUE for this function.

Author(s)

Neil A. Weiss

Examples

```
# Lot size, house size, and value for a sample of homes in a particular area.
data("homes")
str(homes)
attach(homes)

# 95% (default) confidence interval for the correlation between lot size and value.
boot.cor.bca(LOT.SIZE, VALUE)

# 95% (default) lower confidence bound for the correlation between house size
# and value, and a right-tailed test with null hypothesis 0.5.
```

```
boot.cor.bca(HOUSE.SIZE, VALUE, null.hyp = 0.5, alternative = "greater")

# 90% two-sided confidence interval for the correlation between house size and value,
# a right-tailed test with null hypothesis 0.5, and 999 bootstrap replications.
boot.cor.bca(HOUSE.SIZE, VALUE, null.hyp = 0.5, alternative = "greater",
  conf.level = 0.90, type = "two-sided", R = 999)

detach(homes) # clean up
```

 boot.cor.per

Percentile Bootstrap Correlation Test and CI

Description

Obtains a confidence interval and (optionally) performs a hypothesis test for the Pearson correlation, using the percentile bootstrap method.

Usage

```
boot.cor.per(x, y, null.hyp = NULL,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of data values.
y	a (non-empty) numeric vector of data values.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Value

A list with class "boot.regcor" containing the following components:

Boot.values	the point estimates (correlations) obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.

Variable.1	the first variable.
Variable.2	the second variable.
n	the sample size.
Statistic	the name of the statistic, here correlation.
Observed	the observed point estimate (correlation).
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias / Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.
cor.ana	a logical; always TRUE for this function.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# NOTE: See the preceding warning!

# Lot size, house size, and value for a sample of homes in a particular area.
data("homes")
str(homes)
attach(homes)

# 95% (default) confidence interval for the correlation between lot size and value.
boot.cor.per(LOT.SIZE, VALUE)

# 95% (default) lower confidence bound for the correlation between house size
# and value, and a right-tailed test with null hypothesis 0.5.
boot.cor.per(HOUSE.SIZE, VALUE, null.hyp = 0.5, alternative = "greater")
```

```
# 90% two-sided confidence interval for the correlation between house size and value,
# a right-tailed test with null hypothesis 0.5, and 999 bootstrap replications.
boot.cor.per(HOUSE.SIZE, VALUE, null.hyp = 0.5, alternative = "greater",
conf.level = 0.90, type = "two-sided", R = 999)

detach(homes) # clean up
```

boot.one.bca

BCa Bootstrap One-Sample Test and CI

Description

Obtains a confidence interval and (optionally) performs a hypothesis test for one population mean, median, proportion, standard deviation, or user-defined function such as a trimmed mean, using the BCa bootstrap method.

Usage

```
boot.one.bca(x, parameter, null.hyp = NULL,
             alternative = c("two.sided", "less", "greater"),
             conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of data values.
parameter	the parameter under consideration.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

For a proportion, the data must consist of 1s and 0s, with 1 corresponding to a success.

Value

A list with class "boot.one" containing the following components:

Boot.values	the point estimates for the parameter obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable	the name of the variable under consideration.
n	the sample size.
Statistic	the name of the statistic.
Observed	the observed point estimate for the parameter.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Author(s)

Neil A. Weiss

Examples

```
# Losses ($) for a sample of 25 pickpocket offenses.
data("loss")
str(loss)

# 95% (default) confidence interval for the mean loss of all pickpocket offenses.
boot.one.bca(loss, mean)

# 95% (default) lower confidence bound for the mean loss of all pickpocket
# offenses, and a right-tailed test with null hypothesis 500.
boot.one.bca(loss, mean, null.hyp = 500, alternative = "greater")

# 90% two-sided confidence interval for the mean loss of all pickpocket
# offenses, and a right-tailed test with null hypothesis 500.
```

```
boot.one.bca(loss, mean, null.hyp = 500, alternative = "greater", conf.level = 0.90,
type = "two-sided")

# 95% (default) confidence interval for the standard deviation of losses of all
# pickpocket offenses.
boot.one.bca(loss, sd)

# 95% (default) confidence interval for the 20% trimmed mean.
twen.tm <- function(x) mean(x, trim = 0.20)
boot.one.bca(loss, twen.tm)
```

boot.one.per

Percentile Bootstrap One-Sample Test and CI

Description

Obtains a confidence interval and (optionally) performs a hypothesis test for one population mean, median, proportion, standard deviation, or user-defined function such as a trimmed mean, using the percentile bootstrap method.

Usage

```
boot.one.per(x, parameter, null.hyp = NULL,
             alternative = c("two.sided", "less", "greater"),
             conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of data values.
parameter	the parameter under consideration.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

For a proportion, the data must consist of 1s and 0s, with 1 corresponding to a success.

Value

A list with class "boot.one" containing the following components:

Boot.values	the point estimates for the parameter obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable	the name of the variable under consideration.
n	the sample size.
Statistic	the name of the statistic.
Observed	the observed point estimate for the parameter.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# Losses ($) for a sample of 25 pickpocket offenses.
data("loss")
str(loss)

# 95% (default) confidence interval for the mean loss of all pickpocket offenses.
boot.one.per(loss, mean)

# 95% (default) lower confidence bound for the mean loss of all pickpocket
```

```

# offenses, and a right-tailed test with null hypothesis 500.
boot.one.per(loss, mean, null.hyp = 500, alternative = "greater")

# 90% two-sided confidence interval for the mean loss of all pickpocket
# offenses, and a right-tailed test with null hypothesis 500.
boot.one.per(loss, mean, null.hyp = 500, alternative = "greater", conf.level = 0.90,
type = "two-sided")

# 95% (default) confidence interval for the standard deviation of losses of all
# pickpocket offenses.
boot.one.per(loss, sd)
# See the preceding warning!

# 95% (default) confidence interval for the 20% trimmed mean.
twen.tm <- function(x) mean(x, trim = 0.20)
boot.one.per(loss, twen.tm)

```

boot.paired.bca

BCa Bootstrap Paired-Samples Test and CI for Two Means

Description

Obtains a paired-samples confidence interval and (optionally) performs a paired-samples hypothesis test for the difference between two population means, using the BCa bootstrap method.

Usage

```

boot.paired.bca(x, y, variable = NULL, null.hyp = NULL,
                alternative = c("two.sided", "less", "greater"),
                conf.level = 0.95, type = NULL, R = 9999)

```

Arguments

x	a (non-empty) numeric vector of data values.
y	a (non-empty) numeric vector of data values.
variable	an optional character string that gives the name of the variable under consideration.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

Note that x and y must have the same length, as together they represent the paired data. Also note, for instance, that `alternative = "greater"` is the alternative that x variable has a larger mean than y variable.

Value

A list with class "boot.paired" containing the following components:

Boot.values	the point estimates for the differences between the means obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL
Pop.1	the first population.
Pop.2	the second population.
n	the sample size.
Statistic	the name of the statistic, here <code>diff.mean</code> .
Observed	the observed point estimate for the difference between the means.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Author(s)

Neil A. Weiss

Examples

```

# The number of inappropriate words out of 10 that were identified in the
# Times New Roman (TNR) and Gigi fonts by each of 25 participants.
data("fonts")
str(fonts)
attach(fonts)

# 90% confidence interval for the difference between the mean number of
# inappropriate words out of 10 identified for the TNR and Gigi fonts.
boot.paired.bca(TNR, GIGI, conf.level = 0.90)

# A right-tailed test with null hypothesis 2, and a 95% (default) lower
# confidence bound for the difference between the mean number of
# inappropriate words out of 10 identified for the TNR and Gigi fonts.
boot.paired.bca(TNR, GIGI, null.hyp = 2, alternative = "greater")
# Not significant at the 5% level.

# A right-tailed test with null hypothesis 1, and a 95% (default) lower
# confidence bound for the difference between the mean number of
# inappropriate words out of 10 identified for the TNR and Gigi fonts.
boot.paired.bca(TNR, GIGI, null.hyp = 1, alternative = "greater")
# Significant at the 5% level.

detach(fonts) # clean up

```

boot.paired.per

Percentile Bootstrap Paired-Samples Test and CI for Two Means

Description

Obtains a paired-samples confidence interval and (optionally) performs a paired-samples hypothesis test for the difference between two population means, using the percentile bootstrap method.

Usage

```

boot.paired.per(x, y, variable = NULL, null.hyp = NULL,
               alternative = c("two.sided", "less", "greater"),
               conf.level = 0.95, type = NULL, R = 9999)

```

Arguments

x	a (non-empty) numeric vector of data values.
y	a (non-empty) numeric vector of data values.
variable	an optional character string that gives the name of the variable under consideration.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".

conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two-sided", "upper-bound", or "lower-bound"; defaults to "two-sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

Note that x and y must have the same length, as together they represent the paired data. Also note, for instance, that `alternative = "greater"` is the alternative that x variable has a larger mean than y variable.

Value

A list with class "boot.paired" containing the following components:

Boot.values	the point estimates for the differences between the means obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL
Pop.1	the first population.
Pop.2	the second population.
n	the sample size.
Statistic	the name of the statistic, here <code>diff.mean</code> .
Observed	the observed point estimate for the difference between the means.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias / Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# The number of inappropriate words out of 10 that were identified in the
# Times New Roman (TNR) and Gigi fonts by each of 25 participants.
data("fonts")
str(fonts)
attach(fonts)

# 90% confidence interval for the difference between the mean number of
# inappropriate words out of 10 identified for the TNR and Gigi fonts.
boot.paired.per(TNR, GIGI, conf.level = 0.90)

# A right-tailed test with null hypothesis 2, and a 95% (default) lower
# confidence bound for the difference between the mean number of
# inappropriate words out of 10 identified for the TNR and Gigi fonts.
boot.paired.per(TNR, GIGI, null.hyp = 2, alternative = "greater")
# Not significant at the 5% level.

# A right-tailed test with null hypothesis 1, and a 95% (default) lower
# confidence bound for the difference between the mean number of
# inappropriate words out of 10 identified for the TNR and Gigi fonts.
boot.paired.per(TNR, GIGI, null.hyp = 1, alternative = "greater")
# Significant at the 5% level.

detach(fonts) # clean up
```

boot.ratio.sd.bca	<i>BCa Bootstrap Independent-Samples Test and CI for Two Standard Deviations</i>
-------------------	--

Description

Obtains an independent-samples confidence interval and (optionally) performs an independent-samples hypothesis test for the ratio of two population standard deviations, using the BCa bootstrap method.

Usage

```
boot.ratio.sd.bca(x, y, stacked = TRUE, variable = NULL, null.hyp = NULL,
                 alternative = c("two.sided", "less", "greater"),
                 conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a numeric vector of observations of the variable (stacked case) or a numeric vector of data values representing the first of the two samples (unstacked case).
y	a vector of corresponding population identifiers (stacked case) or a numeric vector of data values representing the second of the two samples (unstacked case).
stacked	a logical value (default TRUE) indicating whether the data are stacked.
variable	an optional string that gives the name of the variable under consideration; ignored if stacked is TRUE.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user supplied, must be one of "two-sided", "upper-bound", or "lower-bound"; defaults to "two-sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Value

A list with class "boot.two" containing the following components:

Stacked	a logical indicating whether the data are stacked (TRUE) or not (FALSE).
Boot.values	the point estimates for the ratio of the standard deviations obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Parameter	the parameter under consideration, here standard deviation.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL
Pop.1	the first population.
Pop.2	the second population.
n.1	the sample size for the first population.
n.2	the sample size for the second population.
Statistic	the name of the statistic, here ratio.sd.
Observed.1	the observed point estimate for the standard deviation of the first population.
Observed.2	the observed point estimate for the standard deviation of the second population.
Observed	the observed point estimate for the ratio of the two standard deviations.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.

Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Author(s)

Neil A. Weiss

Examples

```
# Elmendorf tear strengths, in grams, for independent samples of
# Brand A and Brand B vinyl floor coverings.
data("elmendorf")
str(elmendorf)
attach(elmendorf)
# Note that the data are stacked.

# 90% confidence interval for the ratio of the population standard
# deviations of tear strength for Brands A and B.
boot.ratio.sd.bca(STRENGTH, BRAND, conf.level = 0.90)

# 95% (default) confidence interval for the ratio of the population
# standard deviations of tear strength for Brands A and B, and a
# two-tailed hypothesis test with null hypothesis 1 (i.e., the
# population standard deviations are equal).
boot.ratio.sd.bca(STRENGTH, BRAND, null.hyp = 1)

detach(elmendorf) # clean up
```

boot.ratio.sd.per	<i>Percentile Bootstrap Independent-Samples Test and CI for Two Standard Deviations</i>
-------------------	---

Description

Obtains an independent-samples confidence interval and (optionally) performs an independent-samples hypothesis test for the ratio of two population standard deviations, using the percentile bootstrap method.

Usage

```
boot.ratio.sd.per(x, y, stacked = TRUE, variable = NULL, null.hyp = NULL,
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a numeric vector of observations of the variable (stacked case) or a numeric vector of data values representing the first of the two samples (unstacked case).
y	a vector of corresponding population identifiers (stacked case) or a numeric vector of data values representing the second of the two samples (unstacked case).
stacked	a logical value (default TRUE) indicating whether the data are stacked.
variable	an optional string that gives the name of the variable under consideration; ignored if stacked is TRUE.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Value

A list with class "boot.two" containing the following components:

Stacked	a logical indicating whether the data are stacked (TRUE) or not (FALSE).
Boot.values	the point estimates for the ratio of the standard deviations obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Parameter	the parameter under consideration, here standard deviation.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL
Pop.1	the first population.
Pop.2	the second population.
n.1	the sample size for the first population.
n.2	the sample size for the second population.
Statistic	the name of the statistic, here ratio.sd.
Observed.1	the observed point estimate for the standard deviation of the first population.

Observed.2	the observed point estimate for the standard deviation of the second population.
Observed	the observed point estimate for the ratio of the two standard deviations.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# Elmendorf tear strengths, in grams, for independent samples of
# Brand A and Brand B vinyl floor coverings.
data("elmendorf")
str(elmendorf)
attach(elmendorf)
# Note that the data are stacked.

# 90% confidence interval for the ratio of the population standard
# deviations of tear strength for Brands A and B.
boot.ratio.sd.per(STRENGTH, BRAND, conf.level = 0.90)
# See the preceding warning!

# 95% (default) confidence interval for the ratio of the population
# standard deviations of tear strength for Brands A and B, and a
# two-tailed hypothesis test with null hypothesis 1 (i.e., the
# population standard deviations are equal).
boot.ratio.sd.per(STRENGTH, BRAND, null.hyp = 1)
# See the preceding warning!

detach(elmendorf) # clean up
```

boot.slope.bca	<i>BCa Bootstrap Test and CI for the Slope of a Population Regression Line in Simple Linear Regression</i>
----------------	--

Description

Obtains a confidence interval and (optionally) performs a hypothesis test for the slope of a population regression line in simple linear regression, using the BCa bootstrap method.

Usage

```
boot.slope.bca(x, y, null.hyp = NULL,
               alternative = c("two.sided", "less", "greater"),
               conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of predictor-variable data values.
y	the corresponding numeric vector of response-variable data values.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	character string specifying the type of CI; if user-supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

If `null.hyp = 0` and `alternative = "two.sided"`, then the hypothesis test is equivalent to testing whether the predictor variable is useful for making predictions.

Value

A list with class "boot.regcor" containing the following components:

Boot.values	the point estimates for the slope obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable.1	the predictor variable.
Variable.2	the response variable.

n	the sample size.
Statistic	the name of the statistic, here slope.
Observed	the observed point estimate for the slope.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias / Observed$.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.
cor.ana	a logical; always FALSE for this function.

Author(s)

Neil A. Weiss

Examples

```
# Lot size, house size, and value for a sample of homes in a particular area.
data("homes")
str(homes)
attach(homes)

# 95% (default) lower confidence bound for the slope of the population regression
# line relating lot size and value, a right-tailed test with null hypothesis 0,
# and 999 bootstrap replications.
boot.slope.bca(LOT.SIZE, VALUE, null.hyp = 0, alternative = "greater", R = 999)

# 90% two-sided confidence interval for the slope of the population regression line
# relating house size and value, a right-tailed test with null hypothesis 0, and
# 999 bootstrap replications.
boot.slope.bca(HOUSE.SIZE, VALUE, null.hyp = 0, alternative = "greater",
conf.level = 0.90, type = "two-sided", R = 999)

detach(homes) # clean up
```

boot.slope.per	<i>Percentile Bootstrap Test and CI for the Slope of a Population Regression Line in Simple Linear Regression</i>
----------------	---

Description

Obtains a confidence interval and (optionally) performs a hypothesis test for the slope of a population regression line in simple linear regression, using the percentile bootstrap method.

Usage

```
boot.slope.per(x, y, null.hyp = NULL,
               alternative = c("two.sided", "less", "greater"),
               conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a (non-empty) numeric vector of predictor-variable data values.
y	the corresponding numeric vector of response-variable data values.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	character string specifying the type of CI; if user-supplied, must be one of "two.sided", "upper-bound", or "lower-bound"; defaults to "two.sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

If `null.hyp = 0` and `alternative = "two.sided"`, then the hypothesis test is equivalent to testing whether the predictor variable is useful for making predictions.

Value

A list with class "boot.regcor" containing the following components:

Boot.values	the point estimates for the slope obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Header	the main title for the output.
Variable.1	the predictor variable.
Variable.2	the response variable.

n	the sample size.
Statistic	the name of the statistic, here slope.
Observed	the observed point estimate for the slope.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias / Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.
cor.ana	a logical; always FALSE for this function.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# Lot size, house size, and value for a sample of homes in a particular area.
data("homes")
str(homes)
attach(homes)

# 95% (default) lower confidence bound for the slope of the population regression
# line relating lot size and value, a right-tailed test with null hypothesis 0,
# and 999 bootstrap replications.
boot.slope.per(LOT.SIZE, VALUE, null.hyp = 0, alternative = "greater", R = 999)
# See the preceding warning!

# 90% two-sided confidence interval for the slope of the population regression line
# relating house size and value, a right-tailed test with null hypothesis 0, and
# 999 bootstrap replications.
boot.slope.per(HOUSE.SIZE, VALUE, null.hyp = 0, alternative = "greater",
conf.level = 0.90, type = "two-sided", R = 999)

detach(homes) # clean up
```

boot.two.bca	<i>BCa Bootstrap Independent Two-Samples Test and CI</i>
--------------	--

Description

Obtains an independent-samples confidence interval and (optionally) performs an independent-samples hypothesis test for the difference between two population means, medians, proportions, or some user-defined function, using the BCa bootstrap method.

Usage

```
boot.two.bca(x, y, parameter, stacked = TRUE, variable = NULL,
             null.hyp = NULL, alternative = c("two.sided", "less", "greater"),
             conf.level = 0.95, type = NULL, R = 9999)
```

Arguments

x	a numeric vector of observations of the variable (stacked case) or a numeric vector of data values representing the first of the two samples (unstacked case).
y	a vector of corresponding population identifiers (stacked case) or a numeric vector of data values representing the second of the two samples (unstacked case).
parameter	the parameter under consideration.
stacked	a logical value (default TRUE) indicating whether the data are stacked.
variable	an optional string that gives the name of the variable under consideration; ignored if stacked is TRUE.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".
conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two-sided", "upper-bound", or "lower-bound"; defaults to "two-sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

For a proportion, the data must consist of 1s and 0s, with 1 corresponding to a success.

Value

A list with class "boot.two" containing the following components:

Stacked	a logical indicating whether the data are stacked (TRUE) or not (FALSE).
---------	--

Boot.values	the point estimates for the difference between the parameter values obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Parameter	the parameter under consideration.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL.
Pop.1	the first population.
Pop.2	the second population.
n.1	the sample size for the first population.
n.2	the sample size for the second population.
Statistic	the name of the statistic.
Observed.1	the observed point estimate for the parameter value of the first population.
Observed.2	the observed point estimate for the parameter value of the second population.
Observed	the observed point estimate for the difference between the parameter values.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias/Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Author(s)

Neil A. Weiss

Examples

```
# Driving distances, in yards, for independent samples of drives off a
# 2-3/4" wooden tee and off a 3" Stinger Competition golf tee.
data("tees")
str(tees)
attach(tees)
# Note that the data are unstacked.
```

```

# 99% confidence interval for the difference between the mean driving
# distances of the two types of tees. Name variable DISTANCE.
boot.two.bca(REGULAR, STINGER, mean, stacked = FALSE, variable = "DISTANCE",
conf.level = 0.99)

# 95% (default) upper confidence bound for the difference between the mean
# driving distances of the two types of tees, a left-tailed test with null
# hypothesis -10 (i.e., the difference between the mean driving distances
# is -10 yards), and 99999 bootstrap replications.
boot.two.bca(REGULAR, STINGER, mean, stacked = FALSE, null.hyp = -10,
alternative = "less", R = 99999)

# 95% (default) confidence interval for the difference between the standard
# deviations of the driving distances, and a two-tailed test with null
# hypothesis 0 (i.e., the standard deviations are equal). Name variable DISTANCE.
boot.two.bca(REGULAR, STINGER, sd, stacked = FALSE, variable = "DISTANCE", null.hyp = 0)

detach(tees) # clean up

```

boot.two.per

Percentile Bootstrap Independent Two-Samples Test and CI

Description

Obtains an independent-samples confidence interval and (optionally) performs an independent-samples hypothesis test for the difference between two population means, medians, proportions, or some user-defined function, using the percentile bootstrap method.

Usage

```

boot.two.per(x, y, parameter, stacked = TRUE, variable = NULL,
null.hyp = NULL, alternative = c("two.sided", "less", "greater"),
conf.level = 0.95, type = NULL, R = 9999)

```

Arguments

x	a numeric vector of observations of the variable (stacked case) or a numeric vector of data values representing the first of the two samples (unstacked case).
y	a vector of corresponding population identifiers (stacked case) or a numeric vector of data values representing the second of the two samples (unstacked case).
parameter	the parameter under consideration.
stacked	a logical value (default TRUE) indicating whether the data are stacked.
variable	an optional string that gives the name of the variable under consideration; ignored if stacked is TRUE.
null.hyp	the null-hypothesis value; if omitted, no hypothesis test is performed.
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "greater", or "less".

conf.level	the confidence level (between 0 and 1); default is 0.95.
type	a character string specifying the type of CI; if user-supplied, must be one of "two-sided", "upper-bound", or "lower-bound"; defaults to "two-sided" if alternative is "two.sided", "upper-bound" if alternative is "less", and "lower-bound" if alternative is "greater".
R	the number of bootstrap replications; default is 9999.

Details

For a proportion, the data must consist of 1s and 0s, with 1 corresponding to a success.

Value

A list with class "boot.two" containing the following components:

Stacked	a logical indicating whether the data are stacked (TRUE) or not (FALSE).
Boot.values	the point estimates for the difference between the parameter values obtained from the bootstrap.
Confidence.limits	the confidence limit(s) for the confidence interval.
Parameter	the parameter under consideration.
Header	the main title for the output.
Variable	the name of the variable under consideration or NULL.
Pop.1	the first population.
Pop.2	the second population.
n.1	the sample size for the first population.
n.2	the sample size for the second population.
Statistic	the name of the statistic.
Observed.1	the observed point estimate for the parameter value of the first population.
Observed.2	the observed point estimate for the parameter value of the second population.
Observed	the observed point estimate for the difference between the parameter values.
Replications	the number of bootstrap replications.
Mean	the mean of the bootstrap values.
SE	the standard deviation of the bootstrap values.
Bias	the difference between the mean of the bootstrap values and the observed value.
Percent.bias	the percentage bias: $100 * Bias / Observed $.
Null	the null-hypothesis value or NULL.
Alternative	the alternative hypothesis or NULL.
P.value	the P-value or a statement like $P < 0.001$ or NULL.
p.value	the P-value or NULL.
Level	the confidence level.
Type	the type of confidence interval.
Confidence.interval	the confidence interval.

Warning

This routine should be used only when bias is small and the sampling distribution is roughly symmetric, as indicated by the output of the bootstrap. Otherwise, use the BCa version.

Author(s)

Neil A. Weiss

Examples

```
# Driving distances, in yards, for independent samples of drives off a
# 2-3/4" wooden tee and off a 3" Stinger Competition golf tee.
data("tees")
str(tees)
attach(tees)
# Note that the data are unstacked.

# 99% confidence interval for the difference between the mean driving
# distances of the two types of tees. Name variable DISTANCE.
boot.two.per(REGULAR, STINGER, mean, stacked = FALSE, variable = "DISTANCE",
conf.level = 0.99)

# 95% (default) upper confidence bound for the difference between the mean
# driving distances of the two types of tees, a left-tailed test with null
# hypothesis  $-10$  (i.e., the difference between the mean driving distances
# is  $-10$  yards), and 99999 bootstrap replications.
boot.two.per(REGULAR, STINGER, mean, stacked = FALSE, null.hyp = -10,
alternative = "less", R = 99999)

# 95% (default) confidence interval for the difference between the standard
# deviations of the driving distances, and a two-tailed test with null
# hypothesis  $0$  (i.e., the standard deviations are equal). Name variable DISTANCE.
boot.two.per(REGULAR, STINGER, sd, stacked = FALSE, variable = "DISTANCE", null.hyp = 0)

detach(tees) # clean up
```

elmendorf

Elmendorf Tear-Strength Data

Description

Elmendorf tear strengths, in grams, for independent samples of Brand A and Brand B vinyl floor coverings.

Usage

```
data("elmendorf")
```

Format

A data frame with 20 observations on the following 2 variables.

BRAND a factor with levels BRAND.A BRAND.B

STRENGTH a numeric vector

Details

Variation within a method used for testing a product is an essential factor in deciding whether the method should be employed. Indeed, when the variation of such a test is high, ascertaining the true quality of a product is difficult. The Elmendorf tear test is used to evaluate material strength for various manufactured products. In the article cited below, researchers investigated the variation of that test. For one aspect of the study, they randomly and independently obtained the data in "elmendorf" on Elmendorf tear strength, in grams, of two different brands of vinyl floor coverings.

Source

Phillips, A., Jeffries, R., Schneider, J, and Frankoski, S. (1997) Using Repeatability and Reproducibility Studies to Evaluate a Destructive Test Method. *Quality Engineering*, **10**, pp. 283-290.

Examples

```
data(elmendorf)
str(elmendorf)
plot(elmendorf)
attach(elmendorf)
detach(elmendorf)
```

fonts

Times New Roman and Gigi Font Data

Description

The number of inappropriate words out of 10 that were identified in the TNR and Gigi fonts by each of 25 participants.

Usage

```
data("fonts")
```

Format

A data frame with 25 observations on the following 2 variables.

TNR a numeric vector

GIGI a numeric vector

Details

In the online paper cited below, researchers S. Morrison and J. Noyes studied whether the type of font used in a document affects reading speed or comprehension. The fonts used for the comparisons were the serif font Times New Roman (TNR) and a more ornate sans serif font called Gigi. There were 10 substitution words used for testing the comprehensibility of the two fonts. The substitution words were inappropriate to the context of the passage and varied grammatically from the original words in the paragraphs. The "fonts" dataset gives the number of inappropriate words out of the 10 that were identified in the TNR and Gigi fonts by each of the 25 participants.

Source

Morrison, S. and Noyes, J. (2003) A Comparison of Two Computer Fonts: Serif versus Ornate Sans Serif. *Usability News*, **5.3**.

Examples

```
data(fonts)
str(fonts)
boxplot(fonts)
attach(fonts)
detach(fonts)
```

homes

House Size, Lot Size, and Value Data

Description

Data on house size (in square feet), lot size (in acres), and assessed value (in thousands of dollars) for a sample of homes in a particular area.

Usage

```
data("homes")
```

Format

A data frame with 44 observations on the following 3 variables.

HOUSE.SIZE a numeric vector

LOT.SIZE a numeric vector

VALUE a numeric vector

Details

The document cited below describes how county assessors use computerized systems to value single-family residential properties for property tax purposes. The house size (in square feet), lot size (in acres), and assessed value (in thousands of dollars) for a sample of 44 homes in a particular area are presented in the "homes" dataset.

Source

Arizona Residential Property Valuation System, published by the Arizona Department of Revenue.

Examples

```
data(homes)
str(homes)
plot(homes)
attach(homes)
detach(homes)
```

loss

Pickpocket Offenses Data

Description

Losses (\$) for a sample of 25 pickpocket offenses

Usage

```
data("loss")
```

Format

The format is: num [1:25] 447 207 627 430 883 313 844 253 397 214 ...

Details

The Federal Bureau of Investigation (FBI) compiles data on robbery and property crimes and publishes the information in *Population-at-Risk Rates and Selected Crime Indicators*. A random sample of 25 pickpocket offenses yielded the losses, in dollars, presented in the "loss" dataset.

Source

Federal Bureau of Investigation, *Population-at-Risk Rates and Selected Crime Indicators*.

Examples

```
data(loss)
hist(loss)
boxplot(loss)
dotchart(loss)
```

print.boot.one *Print Output for Class "boot.one"*

Description

This is a method for the function `print()` to print objects of class "boot.one".

Usage

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

Arguments

x an object of class "boot.one".
... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a bootstrap function with output list of class "boot.one".

Value

The `boot.one` object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[boot.one.bca](#), [boot.one.per](#)

print.boot.paired *Print Output for Class "boot.paired"*

Description

This is a method for the function `print()` to print objects of class "boot.paired".

Usage

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

Arguments

x an object of class "boot.paired".
... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a bootstrap function with output list of class "boot.paired".

Value

The `boot.paired` object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[boot.paired.bca](#), [boot.paired.per](#)

`print.boot.regcor` *Print Output for Class "boot.regcor"*

Description

This is a method for the function `print()` to print objects of class "boot.regcor".

Usage

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

Arguments

x an object of class "boot.regcor".
... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a bootstrap function with output list of class "boot.regcor".

Value

The `boot.regcor` object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[boot.cond.mean.bca](#), [boot.cond.mean.per](#), [boot.cor.bca](#), [boot.cor.per](#), [boot.slope.bca](#), [boot.slope.per](#)

print.boot.two	<i>Print Output for Class "boot.two"</i>
----------------	--

Description

This is a method for the function `print()` to print objects of class "boot.two".

Usage

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

Arguments

x an object of class "boot.two".
... further arguments passed to or from other methods.

Details

This print method summarizes and formats for easy reading the results of a bootstrap function with output list of class "boot.two".

Value

The `boot.two` object is returned invisibly.

Author(s)

Neil A. Weiss

See Also

[boot.ratio.sd.bca](#), [boot.ratio.sd.per](#), [boot.two.bca](#), [boot.two.per](#)

tees

Driving Distances Data for Regular and Stinger Golf Tees

Description

Driving distances, in yards, for independent samples of drives off a 2-3/4" wooden tee and off a 3" Stinger Competition golf tee.

Usage

```
data("tees")
```

Format

A data frame with 30 observations on the following 2 variables.

REGULAR a numeric vector

STINGER a numeric vector

Details

An independent golf equipment testing facility compared the difference in the performance of golf balls hit off a regular 2-3/4" wooden tee to those hit off a 3" Stinger Competition golf tee. A Callaway Great Big Bertha driver with 10 degrees of loft was used for the test, and a robot swung the club head at approximately 95 miles per hour. Data on total distance traveled (in yards) with each type of tee, based on the test results, are provided in the "tees" dataset.

Source

Golf Laboratories, Inc.; San Diego, CA; August 18, 2003.

Examples

```
data(tees)
str(tees)
boxplot(tees)
attach(tees)
detach(tees)
```

Index

*Topic **datasets**

elmendorf, [31](#)

fonts, [32](#)

homes, [33](#)

loss, [34](#)

tees, [38](#)

*Topic **package, datasets, graphs**

wBoot-package, [2](#)

boot.cond.mean.bca, [2](#), [37](#)

boot.cond.mean.per, [4](#), [37](#)

boot.cor.bca, [6](#), [37](#)

boot.cor.per, [8](#), [37](#)

boot.one.bca, [10](#), [35](#)

boot.one.per, [12](#), [35](#)

boot.paired.bca, [14](#), [36](#)

boot.paired.per, [16](#), [36](#)

boot.ratio.sd.bca, [18](#), [37](#)

boot.ratio.sd.per, [20](#), [37](#)

boot.slope.bca, [23](#), [37](#)

boot.slope.per, [25](#), [37](#)

boot.two.bca, [27](#), [37](#)

boot.two.per, [29](#), [37](#)

elmendorf, [31](#)

fonts, [32](#)

homes, [33](#)

loss, [34](#)

print.boot.one, [35](#)

print.boot.paired, [35](#)

print.boot.regcor, [36](#)

print.boot.two, [37](#)

tees, [38](#)

wBoot (wBoot-package), [2](#)

wBoot-package, [2](#)