

Package ‘PSAboot’

May 7, 2026

Type Package

Title Bootstrapping for Propensity Score Analysis

Version 1.3.9

Date 2025-10-22

Maintainer Jason Bryer <jason@bryer.org>

Description It is often advantageous to test a hypothesis more than once in the context of propensity score analysis (Rosenbaum, 2012) <doi:10.1093/biomet/ass032>. The functions in this package facilitate bootstrapping for propensity score analysis (PSA). By default, bootstrapping using two classification tree methods (using 'rpart' and 'ctree' functions), two matching methods (using 'Matching' and 'MatchIt' packages), and stratification with logistic regression. A framework is described for users to implement additional propensity score methods. Visualizations are emphasized for diagnosing balance; exploring the correlation relationships between bootstrap samples and methods; and to summarize results.

License GPL

URL <https://github.com/jbryer/PSAboot>

BugReports <https://github.com/jbryer/PSAboot/issues>

Depends ggplot2, graphics, PSAgraphics, R (>= 4.1.0)

Imports ggthemes, Matching, MatchIt, modeltools, parallel, party, psych, reshape2, rpart, stats, TriMatch, utils

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyLoad yes

RoxygenNote 7.3.2

NeedsCompilation no

Author Jason Bryer [aut, cre] (ORCID: <<https://orcid.org/0000-0002-2454-0402>>)

Repository CRAN

Date/Publication 2025-10-22 23:30:16 UTC

Contents

PSAboot-package	2
as.data.frame.PSAbootSummary	3
balance	3
balance.matching	4
boot.ctree	5
boot.matching	6
boot.matchit	6
boot.rpart	7
boot.strata	8
boot.weighting	9
boxplot.PSAboot	9
boxplot.PSAboot.balance	10
calculate_ps_weights	11
getPSAbootMethods	12
hist.PSAboot	14
matrixplot	15
pisa.psa.cols	15
pisalux	15
pisausa	17
plot.PSAboot	19
plot.PSAboot.balance	20
print.PSAboot	21
print.PSAboot.balance	21
print.PSAbootSummary	22
psa.strata	22
q25	23
q75	24
summary.PSAboot	24
Index	26

 PSAboot-package

Bootstrapping for Propensity Score Analysis

Description

Bootstrapping procedures for Propensity Score Analysis.

Author(s)

Maintainer: Jason Bryer <jason@bryer.org> ([ORCID](#))

See Also

Useful links:

- <https://github.com/jbryer/PSABoot>
- Report bugs at <https://github.com/jbryer/PSABoot/issues>

as.data.frame.PSABootSummary

Convert the results of PSABoot summary to a data frame.

Description

Convert the results of PSABoot summary to a data frame.

Usage

```
## S3 method for class 'PSABootSummary'  
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	results of '[summary.PSABoot()]'
row.names	row names.
optional	unused.
...	unused.

Value

a data.frame.

balance

Returns a summary of the balance for all bootstrap samples.

Description

This method provides some crude overall measures of balance.

Usage

```
balance(psaboot, na.rm = TRUE, pool.fun = mean)
```

Arguments

<code>psaboot</code>	results from '[PSAboot()]'.
<code>na.rm</code>	should NAs be removed. NAs generally occur when there is insufficient sample for a particular covariate or an unused level.
<code>pool.fun</code>	a function specifying how the effect sizes across all covariates should be combined. Possible values include 'mean()' (default), 'q25()', 'q75()', 'median()', 'max()', or any function that takes a vector of numeric values.

Value

a list with three elements:

unadjusted named numeric vector with unadjusted effect size before adjustment for each covariate

complete a matrix with adjusted effect size for each covariate (columns) for each method (rows).

pooled a matrix with mean adjusted effect size for all covariates for each method (columns) and each bootstrap sample (rows).

balances a list with an $M \times n$ covariates matrix for each method.

Examples

```
library(PSAboot)
data(pisa.psa.cols)
data(pisausa)
bm.usa <- PSAboot(Tr = as.integer(pisausa$PUBPRIV) - 1,
  Y = pisausa$Math,
  X = pisausa[,pisa.psa.cols],
  control.ratio = 5, M = 100, seed = 2112)
bm.usa.bal <- balance(bm.usa)
```

`balance.matching` *Returns balance for each covariate from propensity score matching.*

Description

This function is primarily used by [PSAboot::balance()] and probably does not need to be called directly.

Usage

```
balance.matching(index.treated, index.control, covs)
```

Arguments

`index.treated` a vector with the index of treated rows in `covs`.
`index.control` a vector with the index of control rows in `covs`.
`covs` data frame or matrix of covariates. Factors should already be recoded. See `'[cv.trans.psa()'`

Value

a named vector with one element per covariate.

<code>boot.ctree</code>	<i>Stratification using classification trees for bootstrapping.</i>
-------------------------	---

Description

Stratification using classification trees for bootstrapping.

Usage

```
boot.ctree(Tr, Y, X, X.trans, formu, minStrata = 5, ...)
```

Arguments

`Tr` vector indicating treatment assignment.
`Y` vector of outcome.
`X` matrix or data frame of covariates.
`X.trans` a data frame of `'X'` with factors recoded. See `'[PSAgraphics::cv.trans.psa()'`
`formu` the formula to use to estimate propensity scores. Note that the dependent variable (i.e. treatment variable) name will be updated using the `Tr` vector.
`minStrata` minimum number of treatment or control units within a strata to include that strata.
`...` other parameters passed from `'[PSAboot()'`

Value

a list with three elements:

`summary` a named numeric vector (with at minimum `estimate`, `ci.min`, and `ci.max` but other values allowed)

`balance` a named numeric vector with one element per covariate listed in `X.trans` representing a balance statistic (usually standardized effect size after adjustment)

`details` an arbitrary object that contains the full results of the analysis

boot.matching	<i>Matching package implementation for bootstrapping.</i>
---------------	---

Description

Matching package implementation for bootstrapping.

Usage

```
boot.matching(Tr, Y, X, X.trans, formu, estimand = "ATE", ...)
```

Arguments

Tr	vector indicating treatment assignment.
Y	vector of outcome.
X	matrix or data frame of covariates.
X.trans	a data frame of 'X' with factors recoded. See '[PSAgraphics::cv.trans.psa()]'
formu	the formula to use to estimate propensity scores. Note that the dependent variable (i.e. treatment variable) name will be updated using the Tr vector.
estimand	character string for estimand, either ATE, ATT, or ATC. See '[Matching::Match()]' for more details.
...	other parameters passed to '[Matching::Match()]'.

Value

a list with three elements:

summary	a named numeric vector (with at minimum estimate, ci.min, and ci.max but other values allowed)
balance	a named numeric vector with one element per covariate listed in X.trans representing a balance statistic (usually standardized effect size after adjustment)
details	an arbitrary object that contains the full results of the analysis

boot.matchit	<i>MatchIt package implementation for bootstrapping.</i>
--------------	--

Description

MatchIt package implementation for bootstrapping.

Usage

```
boot.matchit(Tr, Y, X, X.trans, formu, ...)
```

Arguments

Tr	vector indicating treatment assignment.
Y	vector of outcome.
X	matrix or data frame of covariates.
X.trans	a data frame of 'X' with factors recoded. See '[PSAgraphics::cv.trans.psa()]'
formu	the formula to use to estimate propensity scores. Note that the dependent variable (i.e. treatment variable) name will be updated using the Tr vector.
...	other parameters passed from '[PSAboot()]'

Value

a list with three elements:

summary a named numeric vector (with at minimum estimate, ci.min, and ci.max but other values allowed)

balance a named numeric vector with one element per covariate listed in X.trans representing a balance statistic (usually standardized effect size after adjustment)

details an arbitrary object that contains the full results of the analysis

 boot.rpart

Stratification using classification trees for bootstrapping.

Description

Stratification using classification trees for bootstrapping.

Usage

```
boot.rpart(Tr, Y, X, X.trans, formu, minStrata = 5, ...)
```

Arguments

Tr	vector indicating treatment assignment.
Y	vector of outcome.
X	matrix or data frame of covariates.
X.trans	a data frame of 'X' with factors recoded. See '[PSAgraphics::cv.trans.psa()]'
formu	the formula to use to estimate propensity scores. Note that the dependent variable (i.e. treatment variable) name will be updated using the Tr vector.
minStrata	minimum number of treatment or control units within a strata to include that strata.
...	other parameters passed from '[PSAboot()]'

Value

a list with three elements:

summary a named numeric vector (with at minimum estimate, ci.min, and ci.max but other values allowed)

balance a named numeric vector with one element per covariate listed in X.trans representing a balance statistic (usually standardized effect size after adjustment)

details an arbitrary object that contains the full results of the analysis

 boot.strata

Stratification implementation for bootstrapping.

Description

Stratification implementation for bootstrapping.

Usage

```
boot.strata(Tr, Y, X, X.trans, formu, nstrata = 5, ...)
```

Arguments

Tr vector indicating treatment assignment.

Y vector of outcome.

X matrix or data frame of covariates.

X.trans a data frame of 'X' with factors recoded. See '[PSAgraphics::cv.trans.psa()]'

formu the formula to use to estimate propensity scores. Note that the dependent variable (i.e. treatment variable) name will be updated using the Tr vector.

nstrata number of strata to divide the propensity scores.

... other parameters passed from '[PSAboot()]'

Value

a list with three elements:

summary a named numeric vector (with at minimum estimate, ci.min, and ci.max but other values allowed)

balance a named numeric vector with one element per covariate listed in X.trans representing a balance statistic (usually standardized effect size after adjustment)

details an arbitrary object that contains the full results of the analysis

boot.weighting	<i>Propensity score weighting implementation for bootstrapping.</i>
----------------	---

Description

Propensity score weighting implementation for bootstrapping.

Usage

```
boot.weighting(Tr, Y, X, X.trans, formu, estimand = "ATE", ...)
```

Arguments

Tr	vector indicating treatment assignment.
Y	vector of outcome.
X	matrix or data frame of covariates.
X.trans	a data frame of 'X' with factors recoded. See '[PSAgraphics::cv.trans.psa()]'
formu	the formula to use to estimate propensity scores. Note that the dependent variable (i.e. treatment variable) name will be updated using the Tr vector.
estimand	which treatment effect to estimate. Values can be ATE, ATT, ATC, or ATM.
...	other parameters passed from '[PSAboot()]'

Value

a list with three elements:

summary a named numeric vector (with at minimum estimate, ci.min, and ci.max but other values allowed)

balance a named numeric vector with one element per covariate listed in X.trans representing a balance statistic (usually standardized effect size after adjustment)

details an arbitrary object that contains the full results of the analysis

boxplot.PSAboot	<i>Boxplot of PSA bootstrap results.</i>
-----------------	--

Description

Boxplot of PSA bootstrap results.

Usage

```
## S3 method for class 'PSAbboot'
boxplot(
  x,
  bootstrap.mean.color = "blue",
  bootstrap.ci.color = "green",
  bootstrap.ci.width = 0.5,
  bootstrap.ci.size = 3,
  overall.mean.color = "red",
  tufte = FALSE,
  coord.flip = TRUE,
  ...
)
```

Arguments

<code>x</code>	result of <code>[PSAbboot()]</code> .
<code>bootstrap.mean.color</code>	the color of the point for the bootstrap mean, or NA to omit.
<code>bootstrap.ci.color</code>	the color of the confidence intervals of the bootstrap samples, or NA to omit.
<code>bootstrap.ci.width</code>	the width of the confidence interval lines at the end.
<code>bootstrap.ci.size</code>	the size of the confidence interval lines.
<code>overall.mean.color</code>	the color of the point for the overall (before bootstrapping) mean, or NA to omit.
<code>tufte</code>	use Tufte's boxplot style. Requires the <code>ggthemes</code> package.
<code>coord.flip</code>	Whether to flip the coordinates.
<code>...</code>	unused

Value

a `ggplot2` expression.

`boxplot.PSAbboot.balance`

Boxplot of the balance statistics for bootstrapped samples.

Description

Boxplot of the balance statistics for bootstrapped samples.

Usage

```
## S3 method for class 'PSAboot.balance'  
boxplot(  
  x,  
  unadjusted.color = "red",  
  pooled.color = "blue",  
  point.size = 3,  
  point.alpha = 0.5,  
  ...  
)
```

Arguments

x	results of '[balance()]'
unadjusted.color	the color used for the unadjusted effect size.
pooled.color	the color used for the mean bootstrap effect size.
point.size	the size of the points.
point.alpha	the transparency level for the points.
...	other parameters passed to '[facet_wrap()]'

Value

a ggplot2 expression.

Examples

```
library(PSAboot)  
data(pisa.psa.cols)  
data(pisausa)  
bm.usa <- PSAboot(Tr = as.integer(pisausa$PUBPRIV) - 1,  
  Y = pisausa$Math,  
  X = pisausa[,pisa.psa.cols],  
  control.ratio = 5, M = 100, seed = 2112)  
bm.usa.bal <- balance(bm.usa)  
boxplot(bm.usa.bal, nrow = 1)
```

calculate_ps_weights *Calculates propensity score weights.*

Description

Calculates propensity score weights.

Usage

```
calculate_ps_weights(treatment, ps, estimand = "ATE")
```

Arguments

treatment	a logical vector for treatment status.
ps	numeric vector of propensity scores
estimand	character string indicating which estimand to be used. Possible values are ATE (average treatment effect), ATT (average treatment effect for the treated), ATC (average treatment effect for the controls), ATM (Average Treatment Effect Among the Evenly Matched), ATO (Average Treatment Effect Among the Overlap Population)

getPSAbootMethods	<i>Returns a vector with the default methods used by 'PSAboot()'.</i>
-------------------	---

Description

The current default methods are:

Bootstrapping has become a popular resampling method for estimating sampling distributions. And propensity score analysis (PSA) has become popular for estimating causal effects in observational studies. This function implements bootstrapping specifically for PSA. Like typical bootstrapping methods, this function estimates treatment effects for M random samples. However, unlike typical bootstrap methods, this function allows for separate sample sizes for treatment and control units. That is, under certain circumstances (e.g. when the ratio of treatment-to-control units is large) bootstrapping only the control units may be desirable. Additionally, this function provides a framework to use multiple PSA methods for each bootstrap sample.

Usage

```
getPSAbootMethods()
```

```
PSAboot(
  Tr,
  Y,
  X,
  M = 100,
  formu = as.formula(paste0("treat ~ ", paste0(names(X), collapse = " + "))),
  control.ratio = 5,
  control.sample.size = min(control.ratio * min(table(Tr)), max(table(Tr))),
  control.replace = TRUE,
  treated.sample.size = min(table(Tr)),
  treated.replace = TRUE,
  methods = getPSAbootMethods(),
  parallel = TRUE,
  seed = NULL,
```

```
    ...
  )
```

Arguments

Tr	numeric (0 or 1) or logical vector of treatment indicators.
Y	vector of outcome variable
X	matrix or data frame of covariates used to estimate the propensity scores.
M	number of bootstrap samples to generate.
formu	formula used for estimating propensity scores. The default is to use all covariates in X.
control.ratio	the ratio of control units to sample relative to the treatment units.
control.sample.size	the size of each bootstrap sample of control units.
control.replace	whether to use replacement when sampling from control units.
treated.sample.size	the size of each bootstrap sample of treatment units. The default uses all treatment units for each bootstrap sample.
treated.replace	whether to use replacement when sampling from treated units.
methods	a named vector of functions for each PSA method to use.
parallel	whether to run the bootstrap samples in parallel.
seed	random seed. Each iteration, i, will use a seed of seed + i.
...	other parameters passed to <code>[Match()]</code> and <code>[psa.strata()]</code>

Details

Be sure to check the ‘PSAboot’ vignette for more details. The **Applied Propensity Score Analysis with R** online book also has some worked examples: <https://psa.bryer.org>

Value

a vector of methods for use by `PSAboot()`

a list with following elements:

overall.summary Data frame with the results using the complete dataset (i.e. unbootstrapped results).

overall.details Objects returned from each method for complete dataset.

pooled.summary Data frame with results of each bootstrap sample.

pooled.details List of objects returned from each method for each bootstrap sample.

control.sample.size sample size used for control units.

treated.sample.size sample size used for treated units.

control.replace whether control units were sampled with replacement.

treated.replace whether treated units were sampled with replacement.

Tr vector of treatment assignment.

Y vector out outcome.

X matrix or data frame of covariates.

M number of bootstrap samples.

See Also

getPSAbootMethods

Examples

```
library(PSAboot)
data(pisa.psa.cols)
data(pisausa)
bm.usa <- PSAboot(Tr = as.integer(pisausa$PUBPRIV) - 1,
  Y = pisausa$Math,
  X = pisausa[,pisa.psa.cols],
  control.ratio = 5, M = 100, seed = 2112)
```

hist.PSAboot

Histogram of PSA bootstrap results

Description

Histogram of PSA bootstrap results

Usage

```
## S3 method for class 'PSAboot'
hist(x, ...)
```

Arguments

x result of '[PSAboot()]'.

... other parameters passed to '[ggplot2::geom_histogram()]

Value

a ggplot2 expression.

matrixplot	<i>Matrix Plot of Bootstrapped Propensity Score Analysis</i>
------------	--

Description

Matrix Plot of Bootstrapped Propensity Score Analysis

Usage

```
matrixplot(bm)
```

Arguments

bm result from '[PSAboot()]'.

Value

Nothing returned. Creates a plot using the [graphics::pairs()] function.

pisa.psa.cols	<i>Character vector representing the list of covariates used for estimating propensity scores.</i>
---------------	--

Description

Character vector representing the list of covariates used for estimating propensity scores.

Format

a character vector with covariate names for estimating propensity scores.

pisalux	<i>Programme of International Student Assessment (PISA) results from the Luxembourg in 2009.</i>
---------	--

Description

Student results from the 2009 Programme of International Student Assessment (PISA) as provided by the Organization for Economic Co-operation and Development (OECD). See <https://www.oecd.org/pisa/> for more information including the code book.

Format

a data frame with 4,622 rows and 65 columns.

CNT Country
SCHOOLID SchoolID
ST01Q01 Grade
ST04Q01 Sex
ST05Q01 Attend
ST06Q01 Age
ST07Q01 Repeat
ST08Q01 At home mother
ST08Q02 At home father
ST08Q03 At home brothers
ST08Q04 At home sisters
ST08Q05 At home grandparents
ST08Q06 At home others
ST10Q01 Mother highest schooling
ST12Q01 Mother current job status
ST14Q01 Father highest schooling
ST16Q01 Father current job status
ST19Q01 Language at home
ST20Q01 Desk
ST20Q02 Own room
ST20Q03 Study place
ST20Q04 Computer
ST20Q05 Software
ST20Q06 Internet
ST20Q07 Literature
ST20Q08 Poetry
ST20Q09 Art
ST20Q10 Textbooks
ST20Q12 Dictionary
ST20Q13 Dishwasher
ST20Q14 DVD
ST21Q01 How many cellphones
ST21Q02 How many TVs
ST21Q03 How many computers
ST21Q04 How many cars

ST21Q05 How many rooms bath or shower
 ST22Q01 How many books
 ST23Q01 Reading enjoyment time
 ST31Q01 Enrich in test language
 ST31Q02 Enrich in mathematics
 ST31Q03 Enrich in science
 ST31Q05 Remedial in test language
 ST31Q06 Remedial in mathematics
 ST31Q07 Remedial in science
 ST32Q01 Out of school lessons in test language
 ST32Q02 Out of school lessons maths
 ST32Q03 Out of school lessons in science
 PUBPRIV Public or private school
 STRATIO Student to teacher ratio in school

Details

Note that missing values have been imputed using the mice package. Details on the specific procedure are in the `pisa.impute` function in the [pisa package](#).

References

Organisation for Economic Co-operation and Development (2009). Programme for International Student Assessment (PISA).

pisausa	<i>Programme of International Student Assessment (PISA) results from the United States in 2009.</i>
---------	---

Description

Student results from the 2009 Programme of International Student Assessment (PISA) as provided by the Organization for Economic Co-operation and Development (OECD). See www.oecd.org/pisa/ for more information including the code book.

Format

a data frame with 5,233 rows and 65 columns.

CNT Country
 SCHOOLID SchoolID
 ST01Q01 Grade
 ST04Q01 Sex

ST05Q01 Attend
ST06Q01 Age
ST07Q01 Repeat
ST08Q01 At home mother
ST08Q02 At home father
ST08Q03 At home brothers
ST08Q04 At home sisters
ST08Q05 At home grandparents
ST08Q06 At home others
ST10Q01 Mother highest schooling
ST12Q01 Mother current job status
ST14Q01 Father highest schooling
ST16Q01 Father current job status
ST19Q01 Language at home
ST20Q01 Desk
ST20Q02 Own room
ST20Q03 Study place
ST20Q04 Computer
ST20Q05 Software
ST20Q06 Internet
ST20Q07 Literature
ST20Q08 Poetry
ST20Q09 Art
ST20Q10 Textbooks
ST20Q12 Dictionary
ST20Q13 Dishwasher
ST20Q14 DVD
ST21Q01 How many cellphones
ST21Q02 How many TVs
ST21Q03 How many computers
ST21Q04 How many cars
ST21Q05 How many rooms bath or shower
ST22Q01 How many books
ST23Q01 Reading enjoyment time
ST31Q01 Enrich in test language
ST31Q02 Enrich in mathematics
ST31Q03 Enrich in science

ST31Q05 Remedial in test language
 ST31Q06 Remedial in mathematics
 ST31Q07 Remedial in science
 ST32Q01 Out of school lessons in test language
 ST32Q02 Out of school lessons maths
 ST32Q03 Out of school lessons in science
 PUBPRIV Public or private school
 STRATIO Student to teacher ratio in school

Details

Note that missing values have been imputed using the mice package. Details on the specific procedure are in the `pisa.impute` function in the [pisa package](#).

References

Organisation for Economic Co-operation and Development (2009). Programme for International Student Assessment (PISA).

plot.PSABoot	<i>Plot the results of PSABoot</i>
--------------	------------------------------------

Description

Plot the results of PSABoot

Usage

```
## S3 method for class 'PSABoot'
plot(
  x,
  sort = "all",
  ci.sig.color = "red",
  plot.overall = FALSE,
  plot.bootstrap = TRUE,
  ...
)
```

Arguments

<code>x</code>	result of <code>[PSABoot()]</code> .
<code>sort</code>	how the sort the rows by mean difference. Options are to sort using the mean difference from matching, stratification, both individually, or no sorting.
<code>ci.sig.color</code>	the color used for confidence intervals that do not span zero.

<code>plot.overall</code>	whether to plot vertical lines for the overall (non-bootstrapped) estimate and confidence interval.
<code>plot.bootstrap</code>	whether to plot vertical lines for the bootstrap pooled estimate and confidence interval.
<code>...</code>	currently unused.

Value

a ggplot2 expression.

`plot.PSAbboot.balance` *Plot method for balance.*

Description

Plot method for balance.

Usage

```
## S3 method for class 'PSAbboot.balance'
plot(
  x,
  unadjusted.color = "red",
  complete.color = "blue",
  pooled.color = "black",
  ...
)
```

Arguments

<code>x</code>	results from '[balance()]'
<code>unadjusted.color</code>	color of the vertical line representing the mean unadjusted effect size for all covariates.
<code>complete.color</code>	color of the vertical line representing the mean adjusted effect size for all covariates using the complete dataset.
<code>pooled.color</code>	color of the vertical line representing the mean adjusted effect size for all covariates across all bootstrapped samples.
<code>...</code>	currently unused.

Value

a ggplot2 expression.

Examples

```
library(PSAbboot)
data(pisa.psa.cols)
data(pisausa)
bm.usa <- PSAbboot(Tr = as.integer(pisausa$PUBPRIV) - 1,
  Y = pisausa$Math,
  X = pisausa[,pisa.psa.cols],
  control.ratio = 5, M = 100, seed = 2112)
bm.usa.bal <- balance(bm.usa)
plot(bm.usa.bal)
```

```
print.PSAbboot      Print results of PSAbboot
```

Description

Print results of PSAbboot

Usage

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

Arguments

```
x          result of '[PSAbboot()]'.
...        currently unused.
```

Value

Nothing returned. S3 generic function that calls the [PSAbboot::summary()] function.

```
print.PSAbboot.balance Print method for balance.
```

Description

This is a crude measure of overall balance. Absolute value of the standardized effect sizes are calculated for each covariate. Overall balance statistics are the mean of those effect sizes after adjustment for each method across all bootstrap samples.

Usage

```
## S3 method for class 'PSAbboot.balance'
print(x, na.rm = TRUE, ...)
```

Arguments

x	results from '[balance()]'
na.rm	whether NA balance statistics should be removed before averaging them.
...	currently unused.

Value

No valued returned.

`print.PSABootSummary` *Print method for PSABoot Summary.*

Description

Print method for PSABoot Summary.

Usage

```
## S3 method for class 'PSABootSummary'
print(x, digits = 3, ...)
```

Arguments

x	result of '[summary.PSABoot()]'
digits	desired number of digits after the decimal point.
...	unused.

Value

Nothing returned.

`psa.strata` *Propensity Score Analysis using Stratification*

Description

Propensity Score Analysis using Stratification

Usage

```
psa.strata(Y, Tr, strata, trim = 0, minStrata = 5)
```

Arguments

Y	response variable.
Tr	treatment variable.
strata	strata identifier.
trim	allows for a trimmed mean as outcome measure, where trim is from 0 to .5 (.5 implying median).
minStrata	minimum number of treatment or control units within a strata to include that strata.

Value

a character vector containing summary.strata, ATE, se.wtd, approx.t, df, and CI.95.

q25 *Return the 25th percentile.*

Description

Return the 25th percentile.

Usage

```
q25(x, na.rm = FALSE, ...)
```

Arguments

x	numeric vector.
na.rm	logical; if true, any NA and NaN's are removed from x before the quantiles are computed
...	other parameters passed to '[PSAboot()]'.

Value

the 25th percentile.

q75	<i>Returns the 75th percentile.</i>
-----	-------------------------------------

Description

Returns the 75th percentile.

Usage

```
q75(x, na.rm = FALSE, ...)
```

Arguments

x	numeric vector.
na.rm	logical; if true, any NA and NaN's are removed from x before the quantiles are computed
...	other parameters passed to '[PSAboot()]'.

Value

the 75th percentile.

summary.PSAbboot	<i>Summary of pooled results from PSAboot</i>
------------------	---

Description

Summary of pooled results from PSAboot

Usage

```
## S3 method for class 'PSAboot'
summary(object, ...)
```

Arguments

object	result of '[PSAboot()]'.
...	currently unused.

Value

a list with pooled summary statistics.

a list with the results from each PSA method. For each method a list contains the following elements:

sig.tot.per Percentage of bootstrap samples where the confidence interval does not span zero.

bootstrap.mean Weighted mean difference across all bootstrap samples.

bootstrap.ci Overall confidence interval across all bootstrap samples.

bootstrap.weighted.mean Overall weighted bootstrap mean.

percent.sig Contingency table of the number of bootstrap samples that don't span zero.

complete Results of the summary of the PSA method.

Index

* datasets

- [pisa.psa.cols](#), 15
- [as.data.frame.PSABootSummary](#), 3
- [balance](#), 3
- [balance.matching](#), 4
- [boot.ctree](#), 5
- [boot.matching](#), 6
- [boot.matchit](#), 6
- [boot.rpart](#), 7
- [boot.strata](#), 8
- [boot.weighting](#), 9
- [boxplot.PSABoot](#), 9
- [boxplot.PSABoot.balance](#), 10
- [calculate_ps_weights](#), 11
- [getPSABootMethods](#), 12
- [hist.PSABoot](#), 14
- [matrixplot](#), 15
- [pisa.psa.cols](#), 15
- [pisalux](#), 15
- [pisausa](#), 17
- [plot.PSABoot](#), 19
- [plot.PSABoot.balance](#), 20
- [print.PSABoot](#), 21
- [print.PSABoot.balance](#), 21
- [print.PSABootSummary](#), 22
- [psa.strata](#), 22
- [PSABoot \(getPSABootMethods\)](#), 12
- [PSABoot-package](#), 2
- [q25](#), 23
- [q75](#), 24
- [summary.PSABoot](#), 24