

# Package ‘PSAboot’

March 22, 2023

**Type** Package

**Title** Bootstrapping for Propensity Score Analysis

**Version** 1.3.6

**Date** 2023-03-21

**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 (>= 3.0)

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

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyLoad** yes

**RoxygenNote** 7.2.3

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2023-03-22 19:50:05 UTC

**R topics documented:**

PSAboot-package . . . . .	2
as.data.frame.PSAbootSummary . . . . .	3
balance . . . . .	3
balance.matching . . . . .	4
boot.ctree . . . . .	5
boot.matching . . . . .	5
boot.matchit . . . . .	6
boot.rpart . . . . .	7
boot.strata . . . . .	8
boxplot.PSAboot . . . . .	8
boxplot.PSAboot.balance . . . . .	9
getPSAbootMethods . . . . .	10
hist.PSAboot . . . . .	11
matrixplot . . . . .	12
pisa.psa.cols . . . . .	12
pisalux . . . . .	12
pisausa . . . . .	14
plot.PSAboot . . . . .	16
plot.PSAboot.balance . . . . .	17
print.PSAboot . . . . .	18
print.PSAboot.balance . . . . .	18
print.PSAbootSummary . . . . .	19
psa.strata . . . . .	19
PSAboot . . . . .	20
q25 . . . . .	22
q75 . . . . .	22
summary.PSAboot . . . . .	23
<b>Index</b>	<b>24</b>

---

 PSAboot-package

*Bootstrapping for Propensity Score Analysis*


---

**Description**

Bootstrapping procedures for Propensity Score Analysis.

---

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 <a href="#">summary.PSABoot</a>
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

psaboot	results from <a href="#">PSABoot</a> .
na.rm	should NAs be removed. NAs generally occur when there is insufficient sample for a particular covariate or an unused level.
pool.fun	a function specifying how the effect sizes across all covariates should be combined. Possible values include mean (default), <a href="#">q25</a> , <a href="#">q75</a> , 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)
```

---

<code>balance.matching</code>	<i>Returns balance for each covariate from propensity score matching.</i>
-------------------------------	---

---

**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.

---

boot.ctree                      *Stratification using classification trees for bootstrapping.*

---

### 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 <a href="#">cv.trans.psa</a>
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 <a href="#">PSAboot</a>

### 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                      *Matching package implementation for bootstrapping.*

---

### 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 <a href="#">cv.trans.psa</a>
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 <a href="#">Match</a> for more details.
...	other parameters passed to <a href="#">Match</a> .

**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

---

*MatchIt package implementation for bootstrapping.*


---

**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 <a href="#">cv.trans.psa</a>
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 <a href="#">PSAboot</a>

**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 [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	<i>Stratification implementation for bootstrapping.</i>
-------------	---

---

### 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 <a href="#">cv.trans.psa</a>
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 <a href="#">PSAboot</a>

### 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**

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

**Description**

The current default methods are:

**Stratification** [boot.strata](#)

**ctree** [boot.ctree](#)

**rpart** [boot.rpart](#)

**Matching** [boot.matching](#)

**MatchIt** [boot.matchit](#)

**Usage**

```
getPSAbootMethods()
```

**Details**

The default methods can be changed by setting the PSAboot .methods option using `options('PSAboot.methods'=c(...))` where ... is a named list of functions.

**Value**

a vector of methods for use by [PSAboot](#)

---

hist.PSAboot	<i>Histogram of PSA bootstrap results</i>
--------------	---

---

**Description**

Histogram of PSA bootstrap results

**Usage**

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

**Arguments**

x                    result of [PSAboot](#).  
 ...                  other parameters passed to [geom\\_histogram](#)

**Value**

a ggplot2 expression.

---

`matrixplot`*Matrix Plot of Bootstrapped Propensity Score Analysis*

---

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

---

**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`*Programme of International Student Assessment (PISA) results from the Luxembourg in 2009.*

---

**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). <https://www.oecd.org/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 <https://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). <https://www.oecd.org/pisa/>

---

plot.PSAboot

*Plot the results of PSAboot*

---

### 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

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



plot.overall	whether to plot vertical lines for the overall (non-bootstrapped) estimate and confidence interval.
plot.bootstrap	whether to plot vertical lines for the bootstrap pooled estimate and confidence interval.
...	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**

x	results from <a href="#">balance</a>
unadjusted.color	color of the vertical line representing the mean unadjusted effect size for all covariates.
complete.color	color of the vertical line representing the mean adjusted effect size for all covariates using the complete dataset.
pooled.color	color of the vertical line representing the mean adjusted effect size for all covariates across all bootstrapped samples.
...	currently unused.

**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)
plot(bm.usa.bal)
```

---

```
print.PSAboot          Print results of PSAboot
```

---

**Description**

Print results of PSAboot

**Usage**

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

**Arguments**

x	result of <a href="#">PSAboot</a> .
...	currently unused.

**Value**

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

---

```
print.PSAboot.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 'PSAboot.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.

---

 PSAboot

---

*Bootstrapping for propensity score analysis*


---

**Description**

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**

```
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**

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

**Value**

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)
```

---

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 <a href="#">quantile</a> .

**Value**

the 25th percentile.

---

q75 *Returns the 75th percentile.*

---

**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 [quantile](#).

**Value**

the 75th percentile.

---

summary.PSAboot	<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](#), [12](#)
- [as.data.frame.PSAbbootSummary](#), [3](#)
- [balance](#), [3](#), [10](#), [17](#), [19](#)
- [balance.matching](#), [4](#)
- [boot.ctree](#), [5](#), [11](#)
- [boot.matching](#), [5](#), [11](#)
- [boot.matchit](#), [6](#), [11](#)
- [boot.rpart](#), [7](#), [11](#)
- [boot.strata](#), [8](#), [11](#)
- [boxplot.PSAbboot](#), [8](#)
- [boxplot.PSAbboot.balance](#), [9](#)
- [cv.trans.psa](#), [4–8](#)
- [facet\\_wrap](#), [10](#)
- [geom\\_histogram](#), [11](#)
- [getPSAbbootMethods](#), [10](#)
- [hist.PSAbboot](#), [11](#)
- [Match](#), [6](#), [21](#)
- [matrixplot](#), [12](#)
- [pisa.psa.cols](#), [12](#)
- [pisalux](#), [12](#)
- [pisausa](#), [14](#)
- [plot.PSAbboot](#), [16](#)
- [plot.PSAbboot.balance](#), [17](#)
- [print.PSAbboot](#), [18](#)
- [print.PSAbboot.balance](#), [18](#)
- [print.PSAbbootSummary](#), [19](#)
- [psa.strata](#), [19](#), [21](#)
- [PSAbboot](#), [3](#), [5–12](#), [16](#), [18](#), [20](#), [23](#)
- [PSAbboot-package](#), [2](#)
- [q25](#), [3](#), [22](#)
- [q75](#), [3](#), [22](#)
- [quantile](#), [22](#), [23](#)
- [summary.PSAbboot](#), [3](#), [19](#), [23](#)