

Package ‘twang’

April 19, 2009

Version 1.0-1

Date 2006-6-12

Title Toolkit for Weighting and Analysis of Nonequivalent Groups

Author Greg Ridgeway <gregr@rand.org>, Dan McCaffrey <danielm@rand.org>, Andrew Morral <morral@rand.org>

Maintainer Greg Ridgeway <gregr@rand.org>

Depends R (>= 2.2), gbm (>= 1.5-3), survey, xtable

Description This package offers functions for propensity score estimating and weighting, nonresponse weighting, and diagnosis of the weights

License GPL (>= 2)

Repository CRAN

Date/Publication 2006-06-29 08:37:28

R topics documented:

twang-package	2
bal.stat	3
bal.table	4
check.err	5
desc.wts	6
diag.plot	7
dx.wts	8
egsingle	10
get.weights	11
ks.stat	11
lalonge	13
metric.i	14
print.dxwts	14
ps	15

ps.summary	18
raceprofiling	20
sensitivity	21
stop.methods	23
summary.ps	23

Index	25
--------------	-----------

twang-package	<i>Toolkit for Weighting and Analysis of Nonequivalent Groups</i>
---------------	---

Description

This package offers functions for propensity score estimating and weighting, nonresponse weighting, and diagnosis of the weights

Details

Package: twang
 Version: 0.6-7
 Date: 2006-5-17
 Depends: R (>= 2.2), gbm (>= 1.5-3), survey
 License: GPL (version 2 or newer)
 Built: R 2.2.1; ; 2006-02-15 10:35:20; windows

Index:

bal.stat	Calculate weighted balance statistics
bal.table	Compute balance table
check.err	Reports on errors and warnings
desc.wts	Diagnosis of weights
diag.plot	Create diagnostic plots
dx.wts	Propensity score diagnostics
ks.stat	Functions for evaluating balance
lalonde	Lalonde's National Supported Work Demonstration data
metric.i	Losses for *.stat functions
print.dxwts	Print a diagnosis of the weights
ps	Propensity score estimation
ps.summary	Computes balance measures
sensitivity	Sensitivity analysis
stop.methods	Rules for selecting the propensity scores
summary.ps	Summarize a ps object

Further information is available in the following vignettes:

twang Toolkit for Weighting and Analysis of Nonequivalent Groups: A guide to the twang package (source, pdf)

Author(s)

Greg Ridgeway <gregr@rand.org>, Dan McCaffrey <danielm@rand.org>, Andrew Morral <morral@rand.org>
 Maintainer: Greg Ridgeway <gregr@rand.org>

References

Dan McCaffrey, G. Ridgeway, Andrew Morral (2004). "Propensity Score Estimation with Boosted Regression for Evaluating Adolescent Substance Abuse Treatment," *Psychological Methods* 9(4):403-425.

G. Ridgeway (2006). "Assessing the effect of race bias in post-traffic stop outcomes using propensity scores," *Journal of Quantitative Criminology* 22(1).

See Also

The [gbm](#) package

<code>bal.stat</code>	<i>Calculate weighted balance statistics</i>
-----------------------	--

Description

`bal.stat` compares the treatment and control subjects by means, standard deviations, effect size, and KS statistics

Usage

```
bal.stat(data,
         vars = NULL,
         treat.var,
         w.all,
         get.means = TRUE,
         get.ks = TRUE,
         na.action = "level")
```

Arguments

<code>data</code>	a data frame containing the data
<code>vars</code>	a vector of character strings with the names of the variables on which the function will assess the balance
<code>treat.var</code>	the name of the treatment variable
<code>w.all</code>	observation weights (e.g. propensity score weights, sampling weights, or both)
<code>get.means</code>	logical. If TRUE then <code>bal.stat</code> will compute means and variances
<code>get.ks</code>	logical. If TRUE then <code>bal.stat</code> will compute KS statistics
<code>na.action</code>	a character string indicating how <code>bal.stat</code> should handle missing values. Current options are "level", "exclude", or "lowest"

Details

bal.stat calls `ps.summary.f` and `ps.summary.n` for each variable and assembles the results in a table

Value

See `ps.summary` for details on the returned object. `get.means` and `get.ks` manipulate the inclusion of certain columns in the returned result.

See Also

The example for `ps` contains an example of the use of `bal.table`

<code>bal.table</code>	<i>Compute balance table</i>
------------------------	------------------------------

Description

Extract the balance table from `ps` and `dx.wts` objects

Usage

```
bal.table(x)
```

Arguments

`x` a `ps` or `dx.wts` object

Details

`bal.table` is a generic function for extracting balance tables from `ps` and `dx.wts` objects. These objects usually have several sets of candidate weights, one for an unweighted analysis and perhaps several `stop.methods`. `bal.table` will return a table for each set of weights combined into a list. Each list component will be named as given in the `x`, usually the name of the `stop.method`. The balance table labeled “unw” indicates the unweighted analysis.

Value

Returns a data frame containing the balance information.

<code>tx.mn</code>	The mean of the treatment group
<code>tx.sd</code>	The standard deviation of the treatment group
<code>ct.mn</code>	The mean of the control group
<code>ct.sd</code>	The standard deviation of the control group
<code>std.eff.sz</code>	The standardized effect size, $(tx.mn-ct.mn)/tx.sd$. If <code>tx.sd</code> is small or 0, the standardized effect size can be large or INF. Therefore standardized effect sizes greater than 500 are set to NA

<code>stat</code>	the t-statistic for numeric variables and the chi-square statistic for continuous variables
<code>p</code>	the p-value for the test associated with <code>stat</code>
<code>ks</code>	the KS statistic
<code>ks.pval</code>	the KS p-value computed using the analytic approximation, which does not necessarily work well with a lot of ties

See Also

The example for [ps](#) contains an example of the use of `bal.table`

<code>check.err</code>	<i>Reports on errors and warnings</i>
------------------------	---------------------------------------

Description

Reports on errors and warnings encountered while running [ps](#) and [desc.wts](#). This function is not intended for the user to call directly

Usage

```
check.err(cov.table, stage, alerts.stack)
```

Arguments

<code>cov.table</code>	a balance table, intended to be the <code>results</code> component of the list that bal.stat returns
<code>stage</code>	a title for the method “type” used to create the weights, used to label the results
<code>alerts.stack</code>	an object for collecting warnings issued during the analyses

Details

Checks for treatment standard deviations that are exceedingly small or zero and for effect sizes that are unusually large, both indicative of numerical problems or extreme sample imbalance

Value

`check.err` returns no objects but does alter the `alerts.stack` object

desc.wts

*Diagnosis of weights***Description**

desc.wts assesses the quality of a set of weights on balancing a treatment and control group.

Usage

```
desc.wts(data,
          w,
          vars = NULL,
          treat.var,
          tp,
          na.action = "level",
          perm.test.iters=0,
          verbose=TRUE,
          alerts.stack)
```

Arguments

data	a data frame containing the dataset
w	a vector of weights equal to <code>nrow(data)</code>
vars	a vector of variable names corresponding to data
treat.var	the name of the treatment variable
tp	a title for the method "type" used to create the weights, used to label the results
na.action	a string indicating the method for handling missing data
perm.test.iters	an non-negative integer giving the number of iterations of the permutation test for the KS statistic. If <code>perm.test.iters=0</code> then the function returns an analytic approximation to the p-value. This argument is ignored if <code>x</code> is a <code>ps</code> object. Setting <code>perm.test.iters=200</code> will yield precision to within 3% if the true p-value is 0.05. Use <code>perm.test.iters=500</code> to be within 2%
verbose	if TRUE, lots of information will be printed to monitor the the progress of the fitting
alerts.stack	an object for collecting warnings issued during the analyses

Details

desc.wts calls `bal.stat` to assess covariate balance. If `perm.test.iters>0` it will call `bal.stat` multiple times to compute Monte Carlo p-values for the KS statistics and the maximum KS statistic. It assembles the results into a list object, which usually becomes the `desc` component of `ps` objects that `ps` returns.

Value

See the description of the `desc` component of the `ps` object that `ps` returns

See Also

`ps`

diag.plot	<i>Create diagnostic plots</i>
-----------	--------------------------------

Description

Creates diagnostic plots of propensity scores including, a side-by-side boxplot of propensity scores, a histogram of propensity score weights, QQ plots of KS and t-statistic p-values, and a plot of absolute effect sizes

Usage

```
diag.plot(title=NULL,
          treat=NULL,
          p.s=NULL,
          w.ctrl=NULL,
          desc.unw=NULL,
          desc.w=NULL,
          plots="all")

## S3 method for class 'ps':
plot(x, label = "", ask=FALSE, plots="all", ...)

## S3 method for class 'dxwts':
plot(x, label = "", ask=FALSE, plots="all", ...)
```

Arguments

<code>title</code>	a title for the plots
<code>treat</code>	a vector of 0/1 treatment indicators
<code>p.s</code>	a vector of propensity scores (optional)
<code>w.ctrl</code>	weights for the control subjects
<code>desc.unw</code>	a list object containing the balance assessment without weights, usually the result of a call to <code>desc.wts</code> or the <code>desc</code> component of a <code>ps</code> object
<code>desc.w</code>	a list object containing the weighted balance assessment
<code>plots</code>	a character vector listing the plots to be created. The options are all (the default), optimize, ps boxplot, weight histogram, t pvalues, ks pvalues, es. Any other options (such as "none") will produce no plots
<code>x</code>	a <code>ps</code> object, usually one returned from <code>ps</code>

label	a character string for titling the plots
ask	logical. If TRUE then the graphics window waits for a response from the user before showing the next graph
...	other arguments passed to the plot function

Details

`plot.ps` and `plot.dxwts` are wrappers for `diag.plot`

The plots include

Boxplot of propensity scores for cases in the treatment and comparison conditions

Histogram of comparison condition case weights

P-value plots for unweighted and weighted t statistics, and KS statistics

Change in standardized effect size plot. For each model covariate, standardized effect sizes before and after comparison group case weighting are linked by blue lines if weighting reduces the effect size, and by red lines if weights increase the effect size. Significant effect sizes are indicated with a closed red circle. Standardized effect sizes are defined as the difference between the treatment and comparison group means, divided by the treatment group standard deviation. Very large effect sizes are omitted from these plots. When this occurs, a warning is placed at the top of the figure

Value

No returned objects

See Also

[ps](#)

<code>dx.wts</code>	<i>Propensity score diagnostics</i>
---------------------	-------------------------------------

Description

`dx.wts` takes a `ps` object or a set of propensity scores and computes diagnostics assessing covariates balance.

Usage

```
dx.wts(x,
      data,
      vars=NULL,
      treat.var,
      x.as.weights=TRUE,
      sampw=NULL,
      perm.test.iters=0,
      plots=TRUE,
      title)
```

Arguments

<code>x</code>	a data frame, matrix, or vector of propensity score weights or a <code>ps</code> object. <code>x</code> can also be a data frame, matrix, or vector of propensity scores if <code>x.as.weights=FALSE</code>
<code>data</code>	a data frame
<code>vars</code>	a vector of character strings naming variables in <code>data</code> on which to assess balance
<code>treat.var</code>	a character string indicating which variable in <code>data</code> contains the 0/1 treatment group indicator
<code>x.as.weights</code>	TRUE or FALSE indicating whether <code>x</code> specifies propensity score weights or propensity scores. Ignored if <code>x</code> is a <code>ps</code> object
<code>sampw</code>	optional sampling weights. If <code>x</code> is a <code>ps</code> object then the sampling weights should have been passed to <code>ps</code> and not specified here. <code>dx.wts</code> will issue a warning if <code>x</code> is a <code>ps</code> object and <code>sampw</code> is also specified
<code>perm.test.iters</code>	an non-negative integer giving the number of iterations of the permutation test for the KS statistic. If <code>perm.test.iters=0</code> then the function returns an analytic approximation to the p-value. This argument is ignored if <code>x</code> is a <code>ps</code> object. Setting <code>perm.test.iters=200</code> will yield precision to within 3% if the true p-value is 0.05. Use <code>perm.test.iters=500</code> to be within 2%
<code>plots</code>	if <code>plots=TRUE</code> then <code>dx.wts</code> will call <code>diag.plot</code> generating diagnostic plots
<code>title</code>	a short text title, it will be used in plots and saved files. By default this is set to the current date and time

Details

Creates a balance table that compares unweighted and weighted means and standard deviations, computes effect sizes, and KS statistics to assess the ability of the propensity scores to balance the treatment and control groups.

Value

Returns a list containing

<code>treat</code>	the vector of 0/1 treatment assignment indicators
<code>desc</code>	a nested list containing detailed diagnostic information on the weights. This includes the number of treatment and control subjects, the effective sample size, the largest KS statistic, the average absolute effect size, and the complete balance table
<code>summary.tab</code>	a data frame showing balance information
<code>ps</code>	the given propensity scores
<code>w</code>	the given weights
<code>datestamp</code>	the date and time of the call to <code>dx.wts</code>
<code>parameters</code>	the parameters used when calling <code>dx.wts</code>
<code>alerts</code>	text containing any warnings accumulated during the estimation

See Also

The example for `ps` contains an example of the use of `dx.wts`, `diag.plot`

egsingle

US Sustaining Effects study

Description

A subset of the mathematics scores from the U.S. Sustaining Effects Study. The subset consists of information on 1721 students from 60 schools. This dataset is available in the `mlmRev` package.

Usage

```
data(egsingle)
```

Format

A data frame with 7230 observations on the following 12 variables.

schoolid a factor of school identifiers

childid a factor of student identifiers

year a numeric vector indicating the year of the test

grade a numeric vector indicating the student's grade

math a numeric vector of test scores on the IRT scale score metric

retained a factor with levels 0 1 indicating if the student has been retained in a grade.

female a factor with levels Female Male

black a factor with levels 0 1 indicating if the student is Black

hispanic a factor with levels 0 1 indicating if the student is Hispanic

size a numeric vector indicating the number of students enrolled in the school

lowinc a numeric vector giving the percentage of low-income students in the school

mobility a numeric vector

Source

Reproduced from the `mlmRev` package for use in the section on nonresponse weighting in the `twang` package vignette. These data are distributed with the HLM software package (Bryk, Raudenbush, and Congdon, 1996). Conversion to the R format is described in Doran and Lockwood (2006).

References

Doran, H.C. and J.R. Lockwood (2006). "Fitting value-added models in R," *Journal of Educational and Behavioral Statistics*, 31(1)

get.weights *Extract propensity score weights*

Description

Extracts propensity score weights from a ps object.

Usage

```
get.weights(ps1,  
            type = c("ATT", "ATE")[1],  
            stop.method = NULL)
```

Arguments

ps1 a ps object

type indicates whether the weights are for the average treatment effect on the treated (ATT) or the average treatment effect on the population (ATE)

stop.method indicates which set of weights to retrieve from the ps object

Details

Weights for ATT are 1 for the treatment cases and $p/(1-p)$ for the control cases.

Weights for ATE are $1/p$ for the treatment cases and $1/(1-p)$ for the control cases.

Value

a vector of weights

See Also

[ps](#)

ks.stat *Functions for evaluating balance*

Description

These are a collection of functions that can be used as components of [stop.methods](#) for evaluating the balance of two groups

Usage

```

ks.stat(logw = NULL, w.ctrl = NULL,
        gbml = NULL, i = 1, data,
        sampw = rep(1, nrow(data)),
        rule.summary = mean, na.action = "level",
        vars, treat.var, collapse.by.var = FALSE,
        verbose = FALSE)

es.stat(logw = NULL, w.ctrl = NULL,
        gbml = NULL, i = 1, data,
        sampw = rep(1, nrow(data)),
        rule.summary = mean, na.action = "level",
        vars, treat.var, collapse.by.var = FALSE,
        verbose = FALSE)

strata.stat(logw = NULL, w.ctrl = NULL,
            gbml = NULL, i = 1, data,
            sampw = rep(1, nrow(data)),
            rule.summary = mean, na.action = "level",
            vars, treat.var, collapse.by.var = FALSE,
            verbose = FALSE)

```

Arguments

The weights be passed to these functions with any of the first three arguments

logw	the logarithm of the weights
w.ctrl	the weights for the control subjects
gbml	a gbm.object used for estimating the propensity scores, usually the gbm component of a ps object returned from ps
i	the iteration of gbm with which to compute the weights
data	a data frame with the data
sampw	optional sampling weights
rule.summary	a function for summarizing the total balance. Used to collapse statistics across all the covariates. Examples include <code>mean</code> and <code>max</code>
na.action	a string indicating the method for handling missing data
vars	a vector of variable names corresponding to data
treat.var	the name of the treatment variable
collapse.by.var	if TRUE, then statistics computed for factors are collapsed across the levels
verbose	if TRUE, lots of information will be printed to monitor the the progress of the fitting

Details

~~ If necessary, more details than the description above ~~

Value

~Describe the value returned If it is a LIST, use

`comp1` Description of 'comp1'

`comp2` Description of 'comp2'

...

See Also

[stop.methods](#)

lalonge

Lalonge's National Supported Work Demonstration data

Description

One of the datasets used by Dehejia and Wahba in their paper "Causal Effects in Non-Experimental Studies: Reevaluating the Evaluation of Training Programs." Also used as an example dataset in the `MatchIt` package.

Usage

```
data(lalonge)
```

Format

A data frame with 614 observations on the following 10 variables.

treat 1 if treated in the National Supported Work Demonstration, 0 if from the Current Population Survey

age age

educ years of education

black 1 if black, 0 otherwise

hispan 1 if Hispanic, 0 otherwise

married 1 if married, 0 otherwise

nodegree 1 if no degree, 0 otherwise

re74 earnings in 1974 (pretreatment)

re75 earnings in 1975 (pretreatment)

re78 earnings in 1978 (outcome)

Source

<http://www.columbia.edu/~rd247/nswdata.html> <http://cran.r-project.org/src/contrib/Descriptions/MatchIt.html>

References

Lalonde, R. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review* 76: 604-620.

Dehejia, R.H. and Wahba, S. (1999). Causal Effects in Nonexperimental Studies: Re-Evaluating the Evaluation of Training Programs. *Journal of the American Statistical Association* 94: 1053-1062.

metric.i *Losses for *.stat functions*

Description

Rearranges the arguments of the *.stat functions so that they may be passed to [optimize](#)

Usage

```
metric.i(i, fun = ks.stat, ...)
```

Arguments

i	the number of gbm iterations
fun	a valid *.stat function
...	other arguments to be passed to fun

Value

Evaluates fun at i

See Also

[ks.stat](#), [es.stat](#), [strata.stat](#)

print.dxwts *Print a diagnosis of the weights*

Description

Prints a diagnosis of the weights. Extracts summary.tab from the [dx.wts](#) object

Usage

```
print.dxwts(x, ...)
```


<code>pdf.plots</code>	if TRUE then all plots are dumped to a pdf file with the name specified in <code>title</code>
<code>n.trees</code>	number of <code>gbm</code> iterations passed on to <code>gbm</code>
<code>interaction.depth</code>	<code>interaction.depth</code> passed on to <code>gbm</code>
<code>shrinkage</code>	<code>shrinkage</code> passed on to <code>gbm</code>
<code>perm.test.iters</code>	a non-negative integer giving the number of iterations of the permutation test for the KS statistic. If <code>perm.test.iters=0</code> then the function returns an analytic approximation to the p-value. Setting <code>perm.test.iters=200</code> will yield precision to within 3% if the true p-value is 0.05. Use <code>perm.test.iters=500</code> to be within 2%
<code>print.level</code>	the amount of detail to print to the screen
<code>iterlim</code>	maximum number of iterations for the direct optimization
<code>verbose</code>	if TRUE, lots of information will be printed to monitor the the progress of the fitting

Details

`formula` should be something like "treatment ~ X1 + X2 + X3". The treatment variable should be a 0/1 indicator. There is no need to specify interaction terms in the formula. `interaction.depth` controls the level of interactions to allow in the propensity score model.

If `pdf.plots=TRUE` then `ps` causes plots to be saved as a single pdf file with the name "[title].pdf" in the working directory. See `diag.plot` for details of the plots.

Value

Returns an object of class `ps`, a list containing

<code>gbm.obj</code>	The returned <code>gbm</code> object
<code>ps</code>	a data frame containing the estimated propensity scores. Each column is associated with one of the methods selected in <code>stop.methods</code>
<code>w</code>	a data frame containing the propensity score weights. Each column is associated with one of the methods selected in <code>stop.methods</code> . If sampling weights were given then these are incorporated into these weights
<code>plot.info</code>	a list containing the raw data used to generate the plots
<code>desc</code>	a list containing balance tables for each method selected in <code>stop.methods</code> . Includes a component for the unweighted analysis names "unw". Each <code>desc</code> component includes a list with the following components <ul style="list-style-type: none"> ess The effective sample size of the control group n.treat The number of subjects in the treatment group n.ctrl The number of subjects in the control group max.es The largest effect size across the covariates mean.es The mean absolute effect size max.ks The largest KS statistic across the covariates mean.ks The average KS statistic across the covariates

	bal.tab a (potentially large) table summarizing the quality of the weights for equalizing the distribution of features across the two groups. This table is best extracted using the <code>bal.table</code> method. See the help for <code>bal.table</code> for details on the table's contents
	n.trees The estimated optimal number of <code>gbm</code> iterations to optimize the loss function for the associated <code>stop.methods</code>
datestamp	Records the date of the analysis
parameters	Saves the <code>ps</code> call
alerts	Text containing any warnings accumulated during the estimation

Author(s)

Greg Ridgeway <greg@rand.org>, Dan McCaffrey <danielm@rand.org>, Andrew Morral <morral@rand.org>

References

Dan McCaffrey, G. Ridgeway, Andrew Morral (2004). "Propensity Score Estimation with Boosted Regression for Evaluating Adolescent Substance Abuse Treatment," *Psychological Methods* 9(4):403-425.

See Also

[gbm](#)

Examples

```
data(lalonde)
print(nrow(lalonde))

ps.lalonde <- ps(treat ~ age + educ + black + hispan + nodegree +
  married + re74 + re75,
  data = lalonde,
  title="Lalonde example",
  stop.method=stop.methods[c("ks.stat.mean", "ks.stat.max")],
  # generate plots?
  plots="all",
  pdf.plots=FALSE,
  # gbm options
  n.trees=2000,
  interaction.depth=3,
  shrinkage=0.005,
  perm.test.iters=0,
  verbose=TRUE)

# get the balance tables
bal.table(ps.lalonde)

# diagnose the weights using a ps object
a <- dx.wts(ps.lalonde, data=lalonde, treat.var="treat")
print(a)
```

```

bal.table(a)

# diagnose the weights as propensity score weights
#   will be the same as before, except for MC variation in the KS p-values
#   when perm.test.iters is greater than 0
w <- with(ps.lalonde, ps/(1-ps))
w[lalonde$treat==1,] <- 1
dx.wts(w,data=lalonde,treat.var="treat",
       perm.test.iters=0)

# diagnose the weights as propensity scores
p <- ps.lalonde$ps
dx.wts(p,data=lalonde,treat.var="treat",x.as.weights=FALSE)

# look at propensity scores
names(ps.lalonde$ps)
hist(ps.lalonde$ps$ks.stat.max)
boxplot(split(ps.lalonde$ps$ks.stat.max,ps.lalonde$treat),
        ylab="estimated propensity scores",
        names=c("control","treatment"))

# check out the balance
names(ps.lalonde$desc)
# unweighted
ps.lalonde$desc$unw
# optimized for ks.stat.max
ps.lalonde$desc$ks.stat.max

# check out the gbm object, indicates which variables are most influential in
#   estimating the propensity score
summary(ps.lalonde$gbm.obj, n.trees=ps.lalonde$desc$ks.stat.max$n.trees)

# bal.stat() can use an arbitrary set of weights
bal.stat(data=lalonde,
        w.all=w[,1],
        vars=names(lalonde),
        treat.var="treat",
        get.means=TRUE,
        get.ks=TRUE,
        na.action="level")

# sensitivity analysis
sensitivity(ps.lalonde,lalonde,"re78")

```

ps.summary

Computes balance measures

Description

Computes balance measures (mean differences and KS statistics) for a particular covariate and a set of propensity score weights. This function is not intended to be called directly by the user but is

used by other functions in the package.

Usage

```
ps.summary(x, t, w, get.means = TRUE, get.ks = TRUE,
           na.action = c("level", "exclude", "lowest")[1],
           collapse.by.var = FALSE)
ps.summary.f(x, t, w, get.means = TRUE, get.ks = TRUE,
             na.action = c("level", "exclude", "lowest")[1],
             collapse.by.var = TRUE)
ps.summary.n(x, t, w, get.means = TRUE, get.ks = TRUE,
             na.action = c("level", "exclude", "lowest")[1],
             collapse.by.var = FALSE)
```

Arguments

<code>x</code>	a vector containing the data for a single covariate
<code>t</code>	a vector of the same length as <code>x</code> with the 0/1 treatment assignments
<code>w</code>	a vector of the same length as <code>x</code> with the weights
<code>get.means</code>	if TRUE, mean comparisons are computed
<code>get.ks</code>	if TRUE, the KS statistics are computed
<code>na.action</code>	a string indicating the method for handling missing data
<code>collapse.by.var</code>	if TRUE, then statistics computed for factors are collapsed across the levels

Details

`ps.summary` dispatches `ps.summary.n` or `ps.summary.f` depending on whether `x` is a numeric vector or a factor.

Value

Returns a data frame containing the balance information.

<code>tx.mn</code>	The mean of the treatment group
<code>tx.sd</code>	The standard deviation of the treatment group
<code>ct.mn</code>	The mean of the control group
<code>ct.sd</code>	The standard deviation of the control group
<code>std. eff.sz</code>	The standardized effect size, $(tx.mn - ct.mn)/tx.sd$
<code>stat</code>	the t-statistic for numeric variables and the chi-square statistic for continuous variables
<code>p</code>	the p-value for the test associated with <code>stat</code>
<code>ks</code>	the KS statistic
<code>ks.pval</code>	the KS p-value computed using the analytic approximation, which does not necessarily work well with a lot of ties

`get.means` and `get.ks` manipulate the inclusion of certain columns in the returned result.

See Also

[bal.stat](#), [ks.stat](#), [es.stat](#)

Examples

```
treat <- rbinom(100,1,0.5)
w      <- rexp(100)

# categorical data
x.cat <- factor(sample(letters[1:3],size=100,replace=TRUE))
ps.summary.f(x.cat,treat,w)

# numeric data
x.num <- rnorm(100)
ps.summary.n(x.num,treat,w)

# or let ps.summary figure out which to call
ps.summary(x.num,treat,w)
```

raceprofiling

Traffic stop data

Description

Simulated example data for assessing race bias in traffic stop outcomes

Usage

```
data(raceprofiling)
```

Format

A data frame with 5000 observations on the following 10 variables.

id an ID for each traffic stop

nhood a factor indicating the neighborhood in which the stop occurred.

reason The reason for the stop, mechanical/registration violations, dangerous moving violation, non-dangerous moving violation

resident an indicator whether the driver is a resident of the city

age driver's age

male an indicator whether the driver was male

race the race of the driver, with levels A, B, H, W

hour the hour of the stop (24-hour clock)

month and ordered factor indicating in which month the stop took place

citation an indicator of whether the driver received a citation

Source

This is simulated data to demonstrate how to use `twang` to adjust estimates of racial bias for important factors. This dataset does not represent real data from any real law enforcement agency.

References

G. Ridgeway (2006). "Assessing the effect of race bias in post-traffic stop outcomes using propensity scores," *Journal of Quantitative Criminology* 22(1).

<http://www.i-pensieri.com/gregr/rp.shtml>

Examples

```
data(raceprofiling)

# the first five lines of the dataset
raceprofiling[1:5,]
```

sensitivity	<i>Sensitivity analysis</i>
-------------	-----------------------------

Description

Produces a table to help the user assess the extent to which a hidden bias might remove any differences observed in the propensity score analysis.

Usage

```
sensitivity(ps1,
            data,
            outcome,
            order.by.importance = TRUE,
            verbose = TRUE)
```

Arguments

<code>ps1</code>	a <code>ps</code> object as returned from <code>ps</code>
<code>data</code>	the data frame used to fit <code>ps1</code>
<code>outcome</code>	a character string indicating the name of the variable in <code>data</code> to use as the outcome
<code>order.by.importance</code>	if TRUE then the variables are sorted by their relative influence in the <code>gbm.object</code> used to create <code>ps1</code>
<code>verbose</code>	if TRUE, lots of information will be printed to monitor the the progress of the fitting

Details

This function implements the sensitivity analysis described in Ridgeway (2006), Section 5.5. This analysis helps the user assess the extent to which a hidden bias might remove any differences observed in the propensity score analysis.

If there is an important unobserved factor the odds than the correct propensity score weight is not $w(x_i)$, as the propensity score model predicts, but actually $w(x_i, z_i)$ where z represents the unobserved factor. Let $a_i = w(x_i, z_i)/w(x_i)$. These a_i 's give an estimate of $g(a)$, the distribution of the multiplicative errors that we observe in the weights when excluding z_i . Changing the values of the a_i 's will affect the treatment effect estimate if a is correlated with y , the outcome. The stronger the correlation the more sensitive the results will be to the hidden bias. `sensitivity` computes over control group subjects a modified estimate of $E(Y_0|t = 1)$.

$$\frac{\sum_C a_i w_i y_i}{\sum_C a_i w_i}$$

subject to the constraint that $a_i \sim g(a)$ and $cor(a_i, y_i) = \rho$.

Several $g(a)$'s are considered by removing each variable from the propensity score model in turn and computing the ratio of the original weights to the weights with the variable removed. Several choices for ρ are also considered, making ρ as large as possible, as small as possible, and solving for the “break even” ρ , the ρ that eliminates any treatment effect.

Value

Returns a list where each component contains the sensitivity analysis for each `stop.method` used in fitting `ps1`. Each component contains a data frame with a row for each variable in the original propensity score model. The columns are

<code>var</code>	the name of the variable excluded from the model
<code>E0</code>	the estimated $E(Y_0 t = 1)$ with <code>var</code> excluded from the propensity score model
<code>a.min, a.max</code>	the smallest and largest values of a observed
<code>a.cor</code>	the observed correlation between a and y
<code>a.mincor, a.maxcor</code>	the smallest and largest values of ρ possible
<code>minE0, maxE0</code>	the smallest and largest values of estimated $E(Y_0 t = 1)$ possible
<code>breakeven.cor</code>	the break even correlation (see Details section)

Author(s)

Greg Ridgeway <gregr@rand.org>

References

G. Ridgeway (2006). “Assessing the effect of race bias in post-traffic stop outcomes using propensity scores,” *Journal of Quantitative Criminology* 22(1):1-29.

See Also

See [ps](#) for an example

 stop.methods

Rules for selecting the propensity scores

Description

A list of `stop.method` objects built into the `twang` package that encode rules for selecting propensity score weights

Details

The `ps` function uses a `stop.method` object for instructions on how to select the propensity score weights. `twang` has some `stop.method` objects built in but the user may implement their own if they wish and pass them to `ps` for it to optimize.

A valid `stop.method` object is a list that defines the following

metric a function that evaluates the similarity of the distribution of a variable across the treatment and control groups. Currently, the `twang` package has functions `es.stat`, `ks.stat`, and `strata.stat`. The user may implement their own.

rule.summary a function that takes the vector of results from the `metric` function and summarizes them into a single number. `twang` currently utilizes `mean` and `max` for `rule.summary`

direct logical. If `TRUE` then `ps` will try to optimize the weights directly rather than utilizing `gbm`

na.action a character string indicating how `bal.stat` should handle missing values. Current options are "level", "exclude", or "lowest"

name a character string, preferably unique from other `stop.methods` for labeling the resulting weights

In addition, the object must have `class(mystopmethod) == "stop.method"`

 summary.ps

Summarize a ps object

Description

Computes summary information about a stored `ps` object

Usage

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

Arguments

object a [ps](#) object
... additional arguments affecting the summary produced

Details

Compresses the information in the `desc` component of the `ps` object into a short summary table describing the size of the dataset and the quality of the propensity score weights.

Value

See [ps](#) for details on the returned table

See Also

[ps](#)

Index

- *Topic **datasets**
 - egsingle, 10
 - lalonge, 13
 - raceprofiling, 20
- *Topic **error**
 - check.err, 5
- *Topic **hplot**
 - diag.plot, 6
- *Topic **models**
 - bal.table, 4
 - desc.wts, 5
 - dx.wts, 8
 - ks.stat, 11
 - metric.i, 14
 - ps, 15
 - ps.summary, 18
 - sensitivity, 21
 - summary.ps, 23
- *Topic **multivariate**
 - bal.stat, 3
 - ps, 15
 - stop.methods, 23
- *Topic **package**
 - twang-package, 1
- *Topic **print**
 - print.dxwts, 14
- *Topic **utilities**
 - get.weights, 11
- bal.stat, 3, 5, 6, 20
- bal.table, 4, 17
- check.err, 5
- desc.wts, 5, 5, 7
- diag.plot, 6, 7, 9, 15, 16
- dx.wts, 4, 8, 14, 15
- egsingle, 10
- es.stat, 14, 20, 23
- es.stat (ks.stat), 11
- gbm, 2, 12, 14–17, 23
- gbm.object, 12, 21
- get.weights, 11
- ks.stat, 11, 14, 20, 23
- lalonge, 13
- max, 23
- mean, 23
- metric.i, 14
- optimize, 14
- plot.dxwts (diag.plot), 6
- plot.ps (diag.plot), 6
- print.dxwts, 14
- ps, 3–9, 11, 12, 15, 15, 21, 23, 24
- ps.summary, 3, 18
- ps.summary.f, 3
- ps.summary.n, 3
- raceprofiling, 20
- sensitivity, 21
- stop.methods, 4, 11, 13, 15, 17, 23
- strata.stat, 14, 23
- strata.stat (ks.stat), 11
- summary.ps, 23
- twang (twang-package), 1
- twang-package, 1