

Package ‘multilevelPSA’

July 2, 2014

License GPL (>= 2)

Title Multilevel Propensity Score Analysis

Type Package

Author Jason Bryer <jason@bryer.org>

Maintainer Jason Bryer <jason@bryer.org>

Description Functions to estimate and visualize propensity score analysis for multilevel, or clustered, data.

Version 1.2

URL <http://jason.bryer.org/multilevelPSA>,<http://github.com/jbryer/multilevelPSA>

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

Depends ggplot2, plyr, psych, reshape, proto, grid, xtable, R (>= 3.0)

Imports xtable

Suggests testthat, party, MASS

Date 2013-01-25

Collate 'geom_rug_alt.R' 'multilevelPSA-package.R' 'align.R' 'mlpsa.R' 'mlpsa.ctree.R' 'mlpsa.print.R' 'plot.missing.R' 'plot.mlpsa.R' 'plot.mlpsa.difference.R' 'plot.mlpsa.distribution.R' 'plot.tree.R' 'lsos.R' 'plot.difftable.R' 'plot.mlpsa.circ.R' 'mlpsa.logistic.R' 'plot.loess.R' 'psrange.r' 'xtable.mlpsa.r'

NeedsCompilation no

Repository CRAN

Date/Publication 2013-05-10 19:27:33

R topics documented:

multilevelPSA-package	2
align.plots	3
diffable.plot	4
geom_rug_alt	4
getPropensityScores	5
getStrata	6
is.mlpsa	6
loess.plot	7
lsos	8
missing.plot	8
mlpsa	9
mlpsa.circ.plot	10
mlpsa.ctree	12
mlpsa.difference.plot	13
mlpsa.distribution.plot	14
mlpsa.logistic	15
pisa.colnames	15
pisa.countries	16
pisa.psa.cols	16
pisana	16
plot.mlpsa	17
plot.psrangle	18
print.mlpsa	19
print.psrangle	19
print.xmlpsa	20
psrangle	20
summary.mlpsa	21
summary.psrangle	21
tree.plot	22
xtable.mlpsa	23
Index	24

multilevelPSA-package *Multilevel Propensity Score Analysis*

Description

This packages provides functions to estimate and visualize multilvel propensity score analysis.

Details

This package extends the principles put forth by the PSAgraphics (Helmreich, Pruzek, & Xiong, 2010) for multilevel, or clustered, data.

Propensity score analyses are typically done in two phases. In phase I, a statistical model predicting treatment using the available individual covaraites is estimated. This package currently provides functions to perform propensity score estimates using logistic regression (see `mlpsa.logistic`) and conditional inference trees (see `mlpsa.ctree`). The latter method provides explicit stratifications as defined by each leaf node. The former however, results in a numerical value ranging from zero to one (i.e. the fitted values). A common approach is to then create stratifications using quintiles. However, other approaches such as Loess regression are also provided.

Phase II of typical pronsity score analyses concerns with the comparison of an outcome between the treatment and comparison groups. The `mlpsa` method will perform this analysis in a multilevel, or clustered, fashion. That is, the results of the `mlpsa` procedure produce summary results at level one (i.e. each strata within each cluster), level two (i.e. overall results for each cluster), and overall (i.e. overall results across all clusters).

This package also provides a number of visualizaions that provide a critical part in presenting, understanding, and interpreting the results. See `plot.mlpsa` for details.

Author(s)

Jason Bryer <jason@bryer.org>

References

<http://cran.r-project.org/web/packages/PSAgraphics/PSAgraphics.pdf> <http://www.jstatsoft.org/v29/i06/>

See Also

PSAgraphics

align.plots

Adapted from ggExtra package which is no longer available. This is related to an experimental mlpsa plot that will combine the circular plot along with the two individual distributions.

Description

Adapted from ggExtra package which is no longer available. This is related to an experimental mlpsa plot that will combine the circular plot along with the two individual distributions.

Usage

```
align.plots(gl, ...)
```

Arguments

gl	grid.layout
...	graphic elements to combine.

difftable.plot	<i>This function produces a ggplot2 figure containing the mean differences for each level two, or cluster.</i>
----------------	--

Description

This function produces a ggplot2 figure containing the mean differences for each level two, or cluster.

Usage

```
difftable.plot(x, fill.colours = NULL, legendlab = NULL,
              ...)
```

Arguments

x	the results of <code>mlopsa</code> .
fill.colours	the colours to use for each level two.
legendlab	the label to use for the legend, or NULL to exclude.
...	currently unused.

Value

a ggplot2 figure

geom_rug_alt	<i>Alternate Marginal Rug Plot Geom for ggplot2</i>
--------------	---

Description

This alternate rug plot geom simply plots the rug tufts on the top and right (instead of the bottom and left)

Usage

```
geom_rug_alt(mapping = NULL, data = NULL,
             stat = "identity", position = "identity", ...)
```

Arguments

mapping	mapping between variables and aesthetics generated by aes
data	dataset used in this layer, if not specified uses plot dataset
stat	statistic used by this layer
position	position adjustment used by this layer
...	ignored

Author(s)

Hadley Wickham <http://had.co.nz>

With contributions by:

William E. J. Doane <wil@drdoane.com>

Brian A. Danielak <brian@briandk.com>

References

Wickham, H. (2009). Ggplot2: Elegant Graphics for Data Analysis. New York: Springer.

Wilkinson, L. (1999). The Grammar of Graphics. Statistics and computing. New York: Springer.

See Also

[geom_rug](#)

getPropensityScores	<i>Returns a data frame with two columns corresponding to the level 2 variable and the fitted value from the logistic regression.</i>
---------------------	---

Description

Returns a data frame with two columns corresponding to the level 2 variable and the fitted value from the logistic regression.

Usage

```
getPropensityScores(lr.results)
```

Arguments

lr.results the results of [mlpsa.logistic](#)

Value

a data frame

See Also

[mlpsa.logistic](#)

getStrata	<i>Returns a data frame with two columns corresponding to the level 2 variable and the leaves from the conditional inference trees.</i>
-----------	---

Description

Returns a data frame with two columns corresponding to the level 2 variable and the leaves from the conditional inference trees.

Usage

```
getStrata(party.results, data, level2)
```

Arguments

party.results	the results of <code>mlpsa.ctree</code>
data	the data frame to merge results to
level2	the name of the level 2 variable.

Value

a data frame

See Also

`mlpsa.ctree`

is.mlpsa	<i>Returns true if the object is of type mlpsa</i>
----------	--

Description

Returns true if the object is of type mlpsa

Usage

```
is.mlpsa(x)
```

Arguments

x	the object to test
---	--------------------

loess.plot	<i>Loess plot with density distributions for propensity scores and outcomes on top and right, respectively.</i>
------------	---

Description

Loess plot with density distributions for propensity scores and outcomes on top and right, respectively.

Usage

```
loess.plot(x, response, treatment, responseTitle = "",
           treatmentTitle = "Treatment",
           percentPoints.treat = 0.1,
           percentPoints.control = 0.01, ...)
```

Arguments

x	vector of propensity scores.
response	the response variable.
treatment	the treatment variable as a logical type.
percentPoints.treat	the percentage of treatment points to randomly plot.
percentPoints.control	the percentage of control points to randomly plot.
responseTitle	the label to use for the y-axis (i.e. the name of the response variable)
treatmentTitle	the label to use for the treatment legend.
...	other parameters passed to geom_smooth and stat_smooth .

Value

a ggplot2 figure

See Also

[plot.mlpsa](#)

Examples

```
## Not run:
require(multilevelPSA)
require(party)
data(pisana)
data(pisa.psa.cols)
cnt = 'USA' #Can change this to USA, MEX, or CAN
pisana2 = pisana[pisana$CNT == cnt,]
```

```

pisana2$treat <- as.integer(pisana2$PUBPRIV) %% 2
lr.results <- glm(treat ~ ., data=pisana2[,c('treat', 'psa.cols')], family='binomial')
st = data.frame(ps=fitted(lr.results),
math=apply(pisana2[,paste('PV', 1:5, 'MATH', sep='')], 1, mean),
pubpriv=pisana2$treat)
st$treat = as.logical(st$pubpriv)
loess.plot(st$ps, response=st$math, treatment=st$treat, percentPoints.control = 0.4,
percentPoints.treat=0.4)

## End(Not run)

```

lsos	<i>Nicer list of objects in memory. Particularly useful for analysis of large data. #http://stackoverflow.com/questions/1358003/tricks-to-manage-the-available-memory-in-an-r-session</i>
------	--

Description

Nicer list of objects in memory. Particularly useful for analysis of large data. #<http://stackoverflow.com/questions/1358003/tricks-to-manage-the-available-memory-in-an-r-session>

Usage

```
lsos(..., n = 10)
```

Arguments

n	the number of objects to return.
...	not used.

Value

a list of objects loaded sorted by size.

missing.plot	<i>Returns a heat map graphic representing missinging of variables grouped by the given grouping vector.</i>
--------------	--

Description

NOTE: This is an experimental function and the results may vary depending on the nature of the dataset.

Usage

```
missing.plot(x, grouping, grid = FALSE,
             widths = c(unit(3, "null"), unit(1, "inches")),
             heights = c(unit(1, "inches"), unit(3, "null")),
             color = "red", ...)
```

Arguments

x	a data frame containing the variables to visualize missingness
grouping	a vector of length nrow(vars) corresponding to how missing will be grouped by
grid	whether to draw a grid between tiles
widths	the ratio of the widths of the heatmap and histogram.
heights	the ratio of the heights of the heatmap and histogram.
color	the color used for indicating missingness.
...	currently unused.

Value

a ggplot2 expression

See Also

plot.mlpsa

mlpsa	<i>This function will perform phase II of the multilevel propensity score analysis.</i>
-------	---

Description

TODO: Need more details

Usage

```
mlpsa(response, treatment = NULL, strata = NULL,
       level2 = NULL, minN = 5)
```

Arguments

response	vector containing the response values
treatment	vector containing the treatment conditions
strata	vector containing the strata for each response
level2	vector containing the level 2 specifications
minN	the minimum number of subjects per strata for that strata to be included in the analysis.

Value

a mlpsa class

See Also

[mlpsa.ctree](#) [mlpsa.logistic](#)

Examples

```
## Not run:
require(multilevelPSA)
require(party)
data(pisana)
data(pisa.colnames)
data(pisa.psa.cols)
mlctree = mlpsa.ctree(pisana[,c('CNT', 'PUBPRIV', pisa.psa.cols)], formula=PUBPRIV ~ ., level2='CNT')
student.party = getStrata(mlctree, pisana, level2='CNT')
student.party$mathscore = apply(student.party[,paste0('PV', 1:5, 'MATH')], 1, sum) / 5
results.psa.math = mlpsa(response=student.party$mathscore,
  treatment=student.party$PUBPRIV,
  strata=student.party$strata,
  level2=student.party$CNT, minN=5)
results.psa.math
summary(results.psa.math)

## End(Not run)
```

mlpsa.circ.plot

Plots the results of a multilevel propensity score model.

Description

The plot created uses the ggplot2 framework. As such, additional modifications can be made. This plot is an extension of the circ.psa function in the PSAgraphics package for multilevel models.

Usage

```
mlpsa.circ.plot(x,
  xlab = names(multilevelPSA$level2.summary)[4],
  ylab = names(multilevelPSA$level2.summary)[5],
  legendlab = "Level 2", title = NULL,
  overall.col = "blue", overall.ci.col = "green",
  level1.plot = FALSE, level1.point.size = NULL,
  level1.rug.plot = NULL,
  level1.projection.lines = FALSE, level2.plot = TRUE,
  level2.point.size = NULL,
  level2.rug.plot = geom_rug_alt,
  level2.projection.lines = TRUE, level2.label = FALSE,
```

```
unweighted.means = FALSE, weighted.means = FALSE,
fill.colours = NULL, ...)
```

Arguments

<code>x</code>	the results of <code>mlpsa</code> .
<code>xlab</code>	label for the x-axis.
<code>ylab</code>	label for the y-axis.
<code>legendlab</code>	the label for the legend, or <code>NULL</code> to exclude.
<code>title</code>	title for the figure.
<code>overall.col</code>	the color used for the overall results.
<code>overall.ci.col</code>	the color used for the confidence intervals.
<code>level1.plot</code>	logical value indicating whether level 1 points should be plotted.
<code>level1.point.size</code>	the size of level 1 points
<code>level1.rug.plot</code>	the geom to use for plotting a level 1 rug. Possible values are <code>geom_rug</code> (for left and bottom), <code>geom_rug_alt</code> (for top and right), or <code>NULL</code> (to exclude).
<code>level1.projection.lines</code>	logical value indicating whether level 1 project lines (parallel to the unit line) are drawn.
<code>level2.plot</code>	logical value indicating whether level 2 points should be plotted.
<code>level2.point.size</code>	the size of level 2 points
<code>level2.rug.plot</code>	the geom to use for plotting a level 2 rug. Possible values are <code>geom_rug</code> (for left and bottom), <code>geom_rug_alt</code> (for top and right), or <code>NULL</code> (to exclude).
<code>level2.projection.lines</code>	logical value indicating whether level 2 project lines (parallel to the unit line) are drawn.
<code>level2.label</code>	logical value indicating whether level 2 points should be labeled.
<code>unweighted.means</code>	logical value indicating whether horizontal and vertical lines are drawn representing the unweighted (i.e. unadjusted from phase I of PSA) means for each level 2, or cluster.
<code>weighted.means</code>	logical value indicating whether horizontal and vertical lines are drawn representing the weighted means for each level 2, or cluster.
<code>fill.colours</code>	if specified, the colors to use for level 2 points.
<code>...</code>	currently unused.

See Also

`plot.mlpsa`

Examples

```
## Not run:
data(pisana)
data(pisa.colnames)
data(pisa.psa.cols)
mlctree = mlpsa.ctree(pisana[,c('CNT', 'PUBPRIV', pisa.psa.cols)], formula=PUBPRIV ~ ., level2='CNT')
student.party = getStrata(mlctree, pisana, level2='CNT')
student.party$mathscore = apply(student.party[,paste0('PV', 1:5, 'MATH')], 1, sum) / 5
results.psa.math = mlpsa(response=student.party$mathscore,
  treatment=student.party$PUBPRIV,
  strata=student.party$strata,
  level2=student.party$CNT, minN=5)
mlpsa.circ.plot(results.psa.math, legendlab=FALSE)

## End(Not run)
```

mlpsa.ctree	<i>Estimates propensity scores using the recursive partitioning in a conditional inference framework.</i>
-------------	---

Description

This function will estimate propensity scores using the conditional inference framework as outlined in the party package. Specifically, a separate tree will be estimated for each level 2 (or cluster). A key advantage of this framework over other methods for estimating propensity scores is that this method will work on data sets containing missing values.

Usage

```
mlpsa.ctree(vars, formula, level2, ...)
```

Arguments

vars	a data frame containing the covariates to use for estimating the propensity scores.
formula	the model for estimating the propensity scores. For example, treat ~ .
level2	the name of the column in vars specifying the level 2 (or cluster).
...	currently unused.

Value

a list of BinaryTree-class classes for each level 2

References

Torsten Hothorn, Kurt Hornik and Achim Zeileis (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15(3), 651–674.

See Also

[getStrata](#)
[tree.plot](#)

`mlpsa.difference.plot` *Creates a graphic summarizing the differences between treatment and comparison groups within and across level two clusters.*

Description

Creates a graphic summarizing the differences between treatment and comparison groups within and across level two clusters.

Usage

```
mlpsa.difference.plot(x, xlab = "Difference Score",
  ylab = NULL, title = NULL, overall.col = "blue",
  overall.ci.col = "green", level2.point.size = NULL,
  level1.points = TRUE, errorbars = TRUE,
  level2.rug.plot = TRUE, jitter = TRUE, reorder = TRUE,
  labelLevel2 = TRUE, sd = NULL, ...)
```

Arguments

<code>x</code>	the results of mlpsa .
<code>xlab</code>	label for the x-axis, or NULL to exclude.
<code>ylab</code>	label for the y-axis, or NULL to exclude.
<code>title</code>	title of the figure, or NULL to exclude.
<code>overall.col</code>	the color of the overall results line.
<code>overall.ci.col</code>	the color of the overall confidence interval.
<code>level2.point.size</code>	the point size of level 2 points.
<code>level1.points</code>	logical value indicating whether level 1 strata should be plotted.
<code>errorbars</code>	logical value indicating whether error bars should be plotted for for each level 1.
<code>level2.rug.plot</code>	logical value indicating whether a rug plot should be plotted for level 2.
<code>jitter</code>	logical value indicating whether level 1 points should be jittered.
<code>reorder</code>	logical value indicating whether the level two clusters should be reordered from largest difference to smallest.
<code>labelLevel2</code>	logical value indicating whether the difference for each level 2 should be labeled.
<code>sd</code>	If specified, effect sizes will be plotted instead of difference in the native unit.
<code>...</code>	currently unused.

See Also

plot.mlpsa

Examples

```
## Not run:
data(pisana)
data(pisa.colnames)
data(pisa.psa.cols)
mlctree = mlpsa.ctree(pisana[,c('CNT', 'PUBPRIV', pisa.psa.cols)], formula=PUBPRIV ~ ., level2='CNT')
student.party = getStrata(mlctree, pisana, level2='CNT')
student.party$mathscore = apply(student.party[,paste0('PV', 1:5, 'MATH')], 1, sum) / 5
results.psa.math = mlpsa(response=student.party$mathscore,
  treatment=student.party$PUBPRIV,
  strata=student.party$strata,
  level2=student.party$CNT, minN=5)
mlpsa.difference.plot(results.psa.math, sd=mean(student.party$mathscore, na.rm=TRUE))

## End(Not run)
```

mlpsa.distribution.plot

Plots distribution for either the treatment or comparison group.

Description

Plots distribution for either the treatment or comparison group.

Usage

```
mlpsa.distribution.plot(x, treat, fill.colours = NULL,
  flip = TRUE, label = treat, level2.label = NULL,
  legendlab = NULL, axis.text.size = 8, ...)
```

Arguments

x	the results of mlpsa .
treat	the group to plot. This must be one of the two levels of the treatment variable.
fill.colours	if specified, the colors to use for level 2 points.
flip	if TRUE, the level 2 clusters will be on the y-axis and the outcome variable on the x-axis. Otherwise reversed.
label	the label to use for the axis.
level2.label	the axis label for the level 2 indicators.
legendlab	the label for the legend, or NULL to exclude a legend.
axis.text.size	the size of the axis text
...	currently unused.

See Also

plot.mlpsa

mlpsa.logistic	<i>Estimates propensity scores using logistic regression.</i>
----------------	---

Description

This method will estimate a separate logistic regression model for each level 2 (or cluster).

Usage

```
mlpsa.logistic(vars, formula, level2, stepAIC = FALSE,
  ...)
```

Arguments

vars	data frame containing the variables to estimate the logistic regression
formula	the logistic regression formula to use
level2	the name of the column containing the level 2 specification
stepAIC	if true, the stepAIC from the MASS package will be used within each level.
...	currently unused.

Value

a list of glm classes for each level 2 or stepwise-selected model if stepAIC is true.

See Also

getPropensityScores

pisa.colnames	<i>Mapping of variables in pisa with full descriptions.</i>
---------------	---

Description

This data frame provides three variables, `Variable` corresponding to the column names in [pisa](#), `ShortDesc` providing a short description of the variable as a valid R object name, and `Desc` providing a longer description of the variable.

Format

a data frame with 50 rows of 3 variables.

`pisa.countries` *Data frame mapping PISA countries to their three letter abbreviation.*

Description

This data frame has two columns, CNT3 for the three letter abbreviation of each country and Country that provides the full country name in English.

Format

data frame with 65 rows of 2 variables.

`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.

`pisana` *North American (i.e. Canada, Mexico, and United States) student results of the 2009 Programm of International Student Assessment.*

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 <http://www.pisa.oecd.org/> for more information including the code book.

Format

a data frame with 66,548 observations of 65 variables.

Details

Note that missing values have been imputed using the <http://cran.r-project.org/web/packages/mice/index.html> package. Details on the specific procedure are in the `pisa.impute` function in the `pisa` package.

Source

Organization for Economic Co-operation and Development

References

Organisation for Economic Co-operation and Development (2009). Programme for International Student Assessment (PISA). <http://www.pisa.oecd.org/>

plot.mlpsa	<i>Plots the results of a multilevel propensity score model.</i>
------------	--

Description

The plot created uses the ggplot2 framework. As such, additional modifications can be made. This plot is an extension of the circ.psa function in the PSAgraphics package for multilevel models.

Usage

```
## S3 method for class 'mlpsa'
plot(x, ratio = c(1, 2),
     plotExtra = NULL, ...)
```

Arguments

x	the results of <code>mlpsa</code> .
ratio	the ratio of the size of the distribution plots (left and bottom) to the circular plot.
plotExtra	a plot to place in the lower left corner.
...	parameters passed to <code>mlpsa.circ.plot</code> and <code>mlpsa.distribution.plot</code>

Examples

```
## Not run:
require(multilevelPSA)
require(party)
data(pisana)
data(pisa.colnames)
data(pisa.psa.cols)
mlctree = mlpsa.ctree(pisana[,c('CNT', 'PUBPRIV', pisa.psa.cols)], formula=PUBPRIV ~ ., level2='CNT')
student.party = getStrata(mlctree, pisana, level2='CNT')
student.party$mathscore = apply(student.party[,paste0('PV', 1:5, 'MATH')], 1, sum) / 5
results.psa.math = mlpsa(response=student.party$mathscore,
                        treatment=student.party$PUBPRIV,
                        strata=student.party$strata,
                        level2=student.party$CNT, minN=5)
plot(results.psa.math)

## End(Not run)
```

plot.psrangle

Plots densities and ranges for the propensity scores.

Description

Plots densities and ranges for the propensity scores.

Usage

```
## S3 method for class 'psrange'
plot(x, xlab = NULL, ylab = NULL,
     labels = c("Comparison", "Treatment"),
     text.ratio.size = 4, text.ncontrol.size = 3,
     point.size = 1, point.alpha = 0.6, line.width = 6,
     density.alpha = 0.2, rect.color = "green",
     rect.alpha = 0.2, ...)
```

Arguments

x	the result of psrange.
xlab	label for x-axis.
ylab	label for y-axis.
labels	labels for the comparison and treatment legend.
text.ratio.size	size of the text for the ratio.
text.ncontrol.size	size of the text for the number of control units.
point.size	size of the points for the minimum and maximum ranges for each model.
point.alpha	the alpha (transparency) level for the points.
line.width	the width of the line between the median of the minimum and maximum ranges.
density.alpha	the alpha (transparency) level of the density curves.
rect.color	the color of the rectangle surrounding the range of minimum and maximum ranges.
rect.alpha	the alpha (transparency) level of the rectangle.
...	currently unused.

Value

a ggplot2 object

print.mlpsa	<i>Prints basic information about a mlpsa class.</i>
-------------	--

Description

Prints basic information about a mlpsa class.

Usage

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

Arguments

x	the mlpsa class.
...	unused.

print.psrangle	<i>Prints information about a psrange result.</i>
----------------	---

Description

Prints information about a psrange result.

Usage

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

Arguments

x	psrange to print info about.
...	currently unused

`print.xmlpsa` *Prints the results of `mlpsa` and `xtable.mlpsa`.*

Description

Print method for `xtable.mlpsa`.

Usage

```
## S3 method for class 'xmlpsa'
print(x,
      tabular.environment = "longtable", floating = FALSE,
      ...)
```

Arguments

`x` result of `xtable.mlpsa`
`tabular.environment` see `print.xtable`.
`floating` see `print.xtable`.
`...` other parameters passed to `print.xtable`

`psrange` *Estimates models with increasing number of comparison subjects starting from 1:1 to using all available comparison group subjects.*

Description

Estimates models with increasing number of comparison subjects starting from 1:1 to using all available comparison group subjects.

Usage

```
psrange(df, treatvar, formula, nsteps = 10, nboot = 10,
        samples, ...)
```

Arguments

`df` data frame with variables to pass to `glm`
`treatvar` vector representing treatment placement. Should be coded as 0s (for control) and 1s (for treatment).
`formula` formula for logistic regression model
`nsteps` number of steps to estimate from 1:1 to using all control records.
`nboot` number of models to execute for each step.
`samples` the sample sizes to draw from control group for each step.
`...` other parameters passed to `glm`.

Value

a class of psrange that contains a summary data frame, a details data frame, and a list of each individual result from glm.

summary.mlpsa	<i>Provides a summary of a mlpsa class.</i>
---------------	---

Description

Provides a summary of a mlpsa class.

Usage

```
## S3 method for class 'mlpsa'
summary(object,
  overall.label = "Overall", ...)
```

Arguments

object	the mlpsa object.
overall.label	the label to place in the strata column for the overall results.
...	unused.

summary.psrange	<i>Prints the summary results of psrange.</i>
-----------------	---

Description

Prints the summary results of psrange.

Usage

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

Arguments

object	psrange to print summary of.
...	currently unused.

tree.plot	<i>Heat map representing variables used in a conditional inference tree across level 2 variables.</i>
-----------	---

Description

This figure provides a summary of the covariates used within each level two cluster along with their relative importance. Covariates are listed on the y-axis and level two clusters along the x-axis. Cells that are shaded indicate that that covariate was present in the conditional. The shade of the color represents the highest level within the tree that covariate appeared. That is, the darkest color, or depth 1, corresponds to the covariate used at the root of the tree, or the first split.

Usage

```
tree.plot(x, colNames, level2Col, colLabels = NULL,
          color.high = "azure", color.low = "steelblue",
          color.na = "white", ...)
```

Arguments

x	the results of <code>mlpsa.ctree</code>
colNames	the columns to include in the graphic
level2Col	the name of the level 2 column.
colLabels	column labels to use. This is a data frame with two columns, the first column should match the values in trees and the second column the description that will be used for labeling the variables.
color.low	color for variables with greater relative importance as determined by occurring sooner in the tree (closer to the root split).
color.high	color for variables with less relative importance as determined by occurring later in the tree (further from the root split).
color.na	color for variables that do not occur in the tree.
...	currently unused.

Value

a `ggplot2` expression

See Also

`plot.mlpsa`

Examples

```
## Not run:
require(party)
data(pisana)
data(pisa.colnames)
data(pisa.psa.cols)
mlctree = mlpsa.ctree(pisana[,c('CNT', 'PUBPRIV', pisa.psa.cols)], formula=PUBPRIV ~ ., level2='CNT')
student.party = getStrata(mlctree, pisana, level2='CNT')
tree.plot(mlctree, level2Col=pisana$CNT)

## End(Not run)
```

xtable.mlpsa	<i>Prints the results of <code>mlpsa</code> as a LaTeX table.</i>
--------------	---

Description

This function implements the `xtable` method for `mlpsa`.

Usage

```
## S3 method for class 'mlpsa'
xtable(x, caption, label, align,
       digits = 2, display, include.note = TRUE, ...)
```

Arguments

<code>x</code>	results of <code>mlpsa</code>
<code>digits</code>	number of digits to print.
<code>include.note</code>	include a table note indicating how many rows were removed due to insufficient data within a strata.
<code>caption</code>	passed through to <code>xtable</code> .
<code>label</code>	passed through to <code>xtable</code> .
<code>align</code>	Not used.
<code>display</code>	passed through to <code>xtable</code> .
<code>...</code>	other parameters passed to <code>summary.mlpsa</code>

Index

- *Topic **analysis**
 - multilevelPSA-package, 2
- *Topic **datasets**
 - pisa.colnames, 15
 - pisa.countries, 16
 - pisa.psa.cols, 16
 - pisana, 16
- *Topic **graphics**
 - multilevelPSA-package, 2
- *Topic **multilevel**
 - multilevelPSA-package, 2
- *Topic **propensity**
 - multilevelPSA-package, 2
- *Topic **psa**
 - multilevelPSA-package, 2
- *Topic **score**
 - multilevelPSA-package, 2
- align.plots, 3
- difftable.plot, 4
- geom_rug, 5
- geom_rug_alt, 4
- geom_smooth, 7
- getPropensityScores, 5
- getStrata, 6, 13
- is.mlpsa, 6
- loess.plot, 7
- lsos, 8
- missing.plot, 8
- mlpsa, 3, 4, 9, 11, 13, 14, 17, 20, 23
- mlpsa.circ.plot, 10, 17
- mlpsa.ctree, 3, 6, 10, 12, 22
- mlpsa.difference.plot, 13
- mlpsa.distribution.plot, 14, 17
- mlpsa.logistic, 3, 5, 10, 15
- multilevelPSA (multilevelPSA-package), 2
- multilevelPSA-package, 2
- pisa.colnames, 15
- pisa.countries, 16
- pisa.psa.cols, 16
- pisana, 15, 16
- plot.mlpsa, 3, 17
- plot.psrangle, 18
- print.mlpsa, 19
- print.psrangle, 19
- print.xmlpsa, 20
- print.xtable, 20
- psrangle, 20
- stat_smooth, 7
- stepAIC, 15
- summary.mlpsa, 21, 23
- summary.psrangle, 21
- tree.plot, 13, 22
- xtable, 23
- xtable.mlpsa, 20, 23