Package ‘pathmox’

February 20, 2015

Type Package

Title Pathmox Approach of Segmentation Trees in Partial Least Squares Path Modeling

Version 0.2.0

Date 2013-12-09

Description pathmox, the cousin package of plspm, provides a very interesting solution for handling segmentation variables in PLS Path Modeling: segmentation trees in PLS Path Modeling.

URL http://www.gastonsanchez.com

Depends R (>= 3.0.1), plspm, tester

Suggests plsdepot

License GPL-3

LazyLoad yes

Collate 'fix.pathmox.R' 'fix.techmox.R' 'pathmox-internal.R'
  'pathmox.R' 'plot.bootnodes.R' 'plot.treemox.R'
  'plot.treemox.pls.R' 'plspmox.R' 'print.bootnodes.R'
  'print.treemox.R' 'print.treemox.pls.R' 'techmox.R'
  'treemox.boot.R' 'treemox.pls.R' 'get_fix_xexeloa.R'
  'get_nominal_split.R' 'get_ordinal_split.R' 'get_pls_basic.R'
  'get_xexeloa.R' 'test_sanchez_aluja.R' 'check_pathmox_args.r'

Author Gaston Sanchez [aut, cre],
  Tomas Aluja [aut, ths]

Maintainer Gaston Sanchez <gaston.stat@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-10 13:34:48

R topics documented:

  pathmox-package ......................................................... 2
Description

pathmox is the cousin package of the awesome package plspm; pathmox provides a very interesting solution for handling segmentation variables in PLS Path Modeling: segmentation trees in PLS Path Modeling

Details

Package: pathmox
Type: Package
Version: 0.1.5
Date: 2012-11-17
Depends: R (>= 2.15.1), plspm
Suggests: plsdepot
License: GPL-3
LazyLoad: yes

Author(s)

Authors: Gaston Sanchez, Tomas Aluja
Maintainer: Gaston Sanchez <gaston.stat@gmail.com>

References

Description
This data set contains the variables from a customer satisfaction study of a Spanish credit institution.

Usage
data(csibank)

Format
A data frame with 1707 observations on the following 32 variables. The first five variables are segmentation variables. The rest of the variables refer to seven latent concepts: 1) IMAG=Image, 2) EXPE=Expectations, 3) QUAL=Quality, 4) VAL=Value, 5) SAT=Satisfaction, and 6) LOTH=Loyalty.

IMAG: Includes variables such as reputation, trustworthiness, seriousness, solidity, and caring about customer’s needs.
EXPE: Includes variables such as products and services provided, customer service, providing solutions, and expectations for the overall quality.
QUAL: Includes variables such as reliable products and services, range of products and services, personal advice, and overall perceived quality.
VAL: Includes variables such as beneficial services and products, valuable investments, quality relative to price, and price relative to quality.
SAT: Includes variables such as overall rating of satisfaction, fulfillment of expectations, satisfaction relative to other banks, and performance relative to customer’s ideal bank.
LOTH: Includes variables such as propensity to choose the same bank again, propensity to switch to other bank, intention to recommend the bank to friends, and sense of loyalty.

Gender  a factor with levels Female Male
Age  a factor with levels <=25 >25 26-35 36-45 46-55 56-65
Education  a factor with levels Elementary Graduated Highschool Undergrad Unfinished
Occupation  a factor with levels Manager MediumEmplo Notemploy OwnFreelan Retired
Region  a factor with levels Center East North
imag1  First MV of the block Image
imag2  Second MV of the block Image
imag3  Third MV of the block Image
imag4  Fourth MV of the block Image
imag5  Fifth MV of the block Image
imag6  Sixth MV of the block Image
expe1  First MV of the block Expectations
expe2  Second MV of the block Expectations
expe3 Third MV of the block Expectations
expe4 Fourth MV of the block Expectations
qua11 First MV of the block Quality
qua12 Second MV of the block Quality
qua13 Third MV of the block Quality
qua14 Fourth MV of the block Quality
qua15 Fifth MV of the block Quality
qua16 Sixth MV of the block Quality
qua17 Seventh MV of the block Quality
val11 First MV of the block Value
val12 Second MV of the block Value
val13 Third MV of the block Value
val14 Fourth MV of the block Value
sat11 First MV of the block Satisfaction
sat12 Second MV of the block Satisfaction
sat13 Third MV of the block Satisfaction
loy11 First MV of the block Loyalty
loy12 Second MV of the block Loyalty
loy13 Third MV of the block Loyalty

Source
Laboratory of Information Analysis and Modeling (LIAM). Facultat d’Informatica de Barcelona, Universitat Politecnica de Catalunya.

Examples
data(csibank)

csimobile | CSI Mobile Phone dataset

Description
This table contains data from a Customer Satisfaction Index CSI on Mobile Phone

Usage
data(csimobile)
Format

A data frame with 87 observations on 33 variables. The first seven variables are segmentation variables. The rest of the variables refer to seven latent concepts: 1) IMAG=Image, 2) EXPE=Expectations, 3) QUAL=Quality, 4) VAL=Value, 5) SAT=Satisfaction, 6) COM=Complaints, and 7) LOY=Loyalty.

IMAG: Includes variables such as trustworthiness, dynamic, solidness, innovation, and caring about customer’s needs.
EXPE: Includes variables such as products and services provided and expectations for the overall quality.
QUAL: Includes variables such as reliable products and services, range of products and services, and overall perceived quality.
VAL: Includes variables such as service and products, quality relative to price, and price relative to quality.
SAT: Includes variables such as overall rating of satisfaction, fulfillment of expectations, satisfaction relative to other phone providers.
COM: Includes one variable defining how well or poorly customer’s complaints were handled.
LOY: Includes variables such as propensity to choose the same phone provider again, intention to recommend the phone provider to friends.

Age  a factor with levels <=25 >25
Gender a factor with levels female male
Occupation a factor with levels employee other student
Education a factor with levels basic highschool university
Operator  a factor with levels A B C
Contract  a factor with levels contract prepay
SwitchOp  a factor with levels No  Yes
image1  First MV of the block Image
image2  Second MV of the block Image
image3  Thrid MV of the block Image
image4  Fourth MV of the block Image
image5  Fifth MV of the block Image
expe1  First MV of the block Expectations
expe2  Second MV of the block Expectations
expe3  Third MV of the block Expectations
expe4  Fourth MV of the block Expectations
qua11  First MV of the block Quality
qua12  Second MV of the block Quality
qua13  Third MV of the block Quality
qua14  Fourth MV of the block Quality
qua15  Fifth MV of the block Quality
qua16  Sixth MV of the block Quality
value1 First MV of the block Value
value2 Second MV of the block Value
value3 Third MV of the block Value
satis1 First MV of the block Satisfaction
satis2 Second MV of the block Satisfaction
satis3 Third MV of the block Satisfaction
comp1 First MV of the block Complaints
comp2 Second MV of the block Complaints
comp3 Third MV of the block Complaints
loyal1 First MV of the block Loyalty
loyal2 Second MV of the block Loyalty

Source
Laboratory of Information Analysis and Modeling (LIAM). Facultat d'Informatica de Barcelona, Universitat Politecnica de Catalunya.

Examples
data(csimobile)

Description
The function pathmox calculates a binary segmentation tree for PLS Path Models following the PATHMOX algorithm. In contrast, fix.pathmox obtains a supervised PATHMOX tree in the sense of allowing the user to interactively fix the partitions along the construction process of the tree.

Usage
pathmox(pls, EXEV, X = NULL, signif = 0.05, size = 0.1, deep = 2, tree = TRUE)

Arguments
pls An object of class "plspm" returned by plspm.
EXEV A data frame of factors containing the segmentation variables.
X Optional dataset (matrix or data frame) used when argument dataset=NULL inside pls.
signif A numeric value indicating the significance threshold of the F-statistic. Must be a decimal number between 0 and 1.
size A numeric value indicating the minimum size of elements inside a node.
deep An integer indicating the depth level of the tree. Must be an integer greater than 1.
tree A logical value indicating if the tree should be displayed (TRUE by default).

Details

The argument EXEV must be a data frame containing segmentation variables as factors (see factor). The number of rows in EXEV must be the same as the number of rows in the data used in pls.

The argument size can be defined as a decimal value (i.e. proportion of elements inside a node), or as an integer (i.e. number of elements inside a node).

When the object pls does not contain a data matrix (i.e. pls$data=NULL), the user must provide the data matrix or data frame in X.

Value

An object of class "treemox". Basically a list with the following results:

MX Data frame with the results of the segmentation tree
FT Data frame containing the results of the F-test for each node partition
candidates List of data frames containing the candidate splits of each node partition
list.nodes List of elements for each node

Author(s)

Gaston Sanchez

References


See Also

technox, plot.treemox, treemox.pls.

Examples

## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
data(csimobile)

# select manifest variables
data_mobile = csimobile[,8:33]
# define path matrix (inner model)
IMAG = c(0, 0, 0, 0, 0,
EXPE = c(1, 0, 0, 0, 0,
QUAL = c(0, 1, 0, 0, 0,
VAL = c(0, 1, 0, 0, 0,
SAT = c(1, 1, 1, 0, 0,
COM = c(0, 0, 0, 1, 0,
LOY = c(1, 0, 0, 0, 1,

# blocks of indicators (outer model)
mob_modes = rep(HA", 7)

# apply plsrm
mob_pls = plsrm(data_mobile, mob_path, mob_blocks, modes = mob_modes,

# re-ordering those segmentation variables with ordinal scale
# (Age and Education)
csimobile$Education = factor(csimobile$Education,
levels = c(Hbasic"", "highschool","university"),
ordered = TRUE)

# select the segmentation variables
seg_vars = csimobile[1:7]

# Pathmox Analysis
mob_pathmox = pathmox(mob_pls, seg_vars, signif = .1, size = .1, deep = 2)

## End(Not run)

---

**plot.bootnodes**

*Plot bootstrap results of terminal nodes*

**Description**

Plot method for objects of class "bootnodes"

The function `plot.bootnodes` displays the value of the selected path coefficient for the root node and the terminal nodes of PATHMOX and TECHMOX trees. In addition, the value of the mean bootstrap as well as the values of the confidence interval are also shown.

**Usage**

```r
## S3 method for class 'bootnodes'
plot(x, pc = 1, ...)
```
Arguments

x object of class "bootnodes"
pc integer indicating the number of path coefficient to be plotted
...
Further arguments are ignored

plot.treemox  Plot PATHMOX and TECHMOX trees

Description

The function plot.treemox allows to display binary trees of PATHMOX and TECHMOX analyses. If shadow.size=0, no shadows are drawn.

Usage

```r
## S3 method for class 'treemox'
plot(x, root.col = "#eeeee",
     root.bor = "#cccccc", root.txt = "#757575",
     root.cex = 0.8, root.lwd = 3, root.shadow = "gray40",
     node.col = "#feb769", node.bor = "#FE9929",
     node.txt = "#555555", node.cex = 0.7, node.lwd = 3,
     node.shadow = "gray30", leaf.col = "#93c4e5",
     leaf.bor = "#5a99c5", leaf.txt = "#555555",
     leaf.cex = 0.7, leaf.lwd = 3, leaf.shadow = "gray30",
     shadow.size = 0, arr.lwd = 3, lcol = "#dddddbbb",
     arr.col = "gray95", seg.cex = 0.7, seg.col = "#2cb0a7",
     cat.cex = 0.8, cat.col = "#555555", show.pval = TRUE,
     pval.col = "#2cb0a7", main = NULL, cex.main = 1,
     col.main = "gray50", ...)
```

Arguments

x An object of class "treemox" returned by pathmox or techmox.
root.col Fill color of root node.
root.bor Border color of root node.
root.txt Text color of root node.
root.cex magnification to be used for text in root node.
root.lwd Line width of border in the root node.
root.shadow Color of shadow of root node.
node.col Fill color of child nodes.
node.bor Border color of child nodes.
node.txt Text color of child nodes.
node.cex magnification to be used for text in child nodes.
node.lwd           Line width of border in child nodes.
node.shadow        Color of shadow of child nodes.
leaf.col           Fill color of leaf nodes.
leaf.bor           Border color of leaf nodes.
leaf.txt           Text color of leaf nodes.
leaf.cex           magnification to be used for text in leaf nodes.
leaf.lwd           Line width of border in leaf nodes.
leaf.shadow        Color of shadow of leaf nodes.
shadow.size        Relative size of shadows.
arr.lwd            Line width of the tree branches.
lcol               color of lines
arr.col            color of arrows
seg.cex            A numerical value indicating the magnification to be used for plotting text.
seg.col            The color to be used for the labels of the segmentation variables.
cat.cex            magnification to be used for the categories
cat.col            The color to be used for the labels of the categories
show.pval          Logical value indicating whether the p-values should be plotted.
pval.col           The color to be used for the labels of the p-values.
main               A main title for the plot.
cex.main           The magnification to be used for the main title.
col.main           Color to be used for the main title
...
Further arguments are ignored.

Examples

```r
## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
dataHcsimobileI

# select manifest variables
data_mobile = csimobile[,8:33]

# define path matrix (inner model)
IMAG = c(0, 0, 0, 0, 0, 0, 0)
EXPE = c(1, 0, 0, 0, 0, 0, 0)
QUAL = c(0, 1, 0, 0, 0, 0, 0)
VAL = c(0, 1, 1, 0, 0, 0, 0)
SAT = c(1, 1, 1, 0, 0, 0, 0)
COM = c(0, 0, 0, 0, 1, 0, 0)
LOY = c(1, 0, 0, 1, 0, 0, 0)
mob_path = rbind(IMAG, EXPE, QUAL, VAL, SAT, COM, LOY)

# blocks of indicators (outer model)
plot.treemox.pls

```r
mob_outer = list(1:5, 6:9, 10:15, 16:18, 19:21, 22:24, 25:26)
mob_modes = rep("A", 7)

# apply plspsm
mob_pls = plspm(data_mobile, mob_path, mob_blocks, modes = mob_modes,
    scheme = "factor", scaled = FALSE)

# re-ordering those segmentation variables with ordinal scale
# (Age and Education)
csimobile$Education = factor(csimobile$Education,
    levels=c("basic","highschool","university"),
    ordered=TRUE)

# select the segmentation variables
seg_vars = csimobile[,1:7]

# Pathmox Analysis
mob_pathmox = pathmox(mob_pls, seg_vars, signif=.10, size=.20, deep=2)

# plot pathmox tree
plot(mob_pathmox, root.col="lightblue", node.col="turquoise", leaf.col="skyblue3",
    shadow.size=0, seg.col="blue2", pval.col="magenta")
```

```
## End(Not run)
```

---

**plot.treemox.pls**

Comparative plot between nodes from a PATHMOX or TECHMOX tree

**Description**

Plot method for objects of class "treemox.pls". Barplots of path coefficients of terminal nodes with respect to those of the global (root) model

**Usage**

```r
## S3 method for class 'treemox.pls'
plot(x, comp.by = "nodes",
    nodes.names = NULL, ordered = TRUE, decreasing = FALSE,
    color = NULL, show.box = TRUE, border = NA,
    cex.names = 0.75, cex.axis = 0.75, short.labs = TRUE,
    short.min = NULL, ...)
```

**Arguments**

- `x` An object of class "treemox.pls" returned by `treemox.pls`.
- `comp.by` One of "nodes" or "latents". This argument indicates the type of barplots comparison.
nodes.names: Optional vector of names for the terminal nodes (must be a vector of length equal to the number of terminal nodes).

ordered: A logical value indicating whether the barplots are shown in increasing or decreasing order.

decreasing: A logical value indicating if the sort order should be increasing or decreasing.

color: Optional vector of colors for the bars. When color=NULL rainbow colors are used.

show.box: A logical value indicating whether a box is drawn around each barplot.

border: The color to be used for the border of the bars. Use border=NA to omit borders.

cex.names: Expansion factor for axis names (bar labels).

cex.axis: Expansion factor for numeric axis labels.

short.labs: Logical value indicating if the labels of the barplots should be abbreviated (TRUE by default).

short.min: Integer number indicating the minimum length of the abbreviations for the labels. Only used when short.labs=TRUE.

Arguments to be passed to/from other methods.

Details

This function aims to visualize the comparison between path coefficients of the terminal nodes against the path coefficients of the global model in the root node.

When comp.by="nodes" a graphic window is displayed for each endogenous latent variable of the PLS model, and barplots of nodes are shown.

When comp.by="latents" a graphic window is displayed for each endogenous relationship of the PLS model, and barplots of independent latent variables are shown.

See Also

treemox.pls, pathmox, techmox

Examples

## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
data(csimobile)

# select manifest variables
data_mobile = csimobile[,8:33]

# define path matrix (inner model)
IMAG = c(0, 0, 0, 0, 0, 0, 0)
EXPE = c(1, 0, 0, 0, 0, 0, 0)
QUAL = c(0, 1, 0, 0, 0, 0, 0)
VAL = c(0, 1, 1, 0, 0, 0, 0)
SAT = c(1, 1, 1, 0, 0, 0, 0)
COM = c(0, 0, 0, 0, 1, 0, 0)
LOY = c(1, 0, 0, 0, 1, 1, 0)
```r
mob_path = rbind(IMAG, EXPE, QUAL, VAL, SAT, COM, LOY)

# blocks of indicators (outer model)
mob_blocks = list(1:5, 6:9, 10:15, 16:18, 19:21, 22:24, 25:26)
mob_modes = rep("A", 7)

# apply pls
mob_pls = plspm(data_mobile, mob_path, mob_blocks, modes = mob_modes,
                 scheme = "factor", scaled = FALSE)

# re-ordering those segmentation variables with ordinal scale (Age and Education)
csimobile$education = factor(csimobile$education,
                               levels=c("basic","highschool","university"),
                               ordered=TRUE)

# select the segmentation variables
seg_vars = csimobile[,1:7]

# Pathmox Analysis
mob_pathmox = pathmox(mob_pls, seg_vars, signif=.10, size=.10, deep=2)

# applying function treemox.pls
mob_nodes <- treemox.pls(mob_pls, mob_pathmox)

# default plot
# comparative barplots of endogenous latent variables between nodes
plot(mob_nodes, comp.by="nodes")

# comparative barplots of nodes between latent regressors
plot(mob_nodes, comp.by="latents", decreasing=TRUE)

## End(Not run)
```

---

**plspmox**

*PLS Path Model of a node from a PATHMOX or TECHMOX tree*

---

**Description**

Calculates a PLS Path Model on a selected node from a pathmox or techmox tree

**Usage**

```r
plspmox(pls, treemox, X = NULL, node, boot.val = FALSE,
        br = NULL, dataset = FALSE)
```

**Arguments**

- `pls` An object of class "plspm"
- `treemox` An object of class "treemox"
X Optional argument for data table
node An integer value indicating the number of the node. Must be an integer larger than 1.
boot.val A logical value indicating whether bootstrap validation is performed (FALSE by default).
br An integer indicating the number bootstrap resamples. Used only when boot.val=TRUE.
dataset A logical value indicating whether the data matrix should be included in the list of results (FALSE by default).

Details
Performs a PLS-PM analysis with the elements contained in node, by calling the function \texttt{plspm}.
The rest of the parameters to perform the PLS-PM analysis (i.e. inner, outer, modes, scheme, scaled, tol, iter) are inherited from the object in argument \texttt{pls}.
When the object \texttt{pls} does not contain a data matrix (i.e. \texttt{pls$data=NULL}), the user must provide the data matrix or data frame in \texttt{X}.

Value
An object of class "plspm"

Author(s)
Gaston Sanchez

See Also
See Also as \texttt{plspm}, \texttt{pathmox}

Examples
```
## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
data(csimobile)

data_mobile = csimobile[,8:33]

# define path matrix (inner model)
IMAG = c(0, 0, 0, 0, 0, 0, 0)
EXPE = c(1, 0, 0, 0, 0, 0, 0)
QUAL = c(0, 1, 0, 0, 0, 0, 0)
VAL = c(0, 1, 1, 0, 0, 0, 0)
SAT = c(1, 1, 1, 1, 0, 0, 0)
COM = c(0, 0, 0, 1, 0, 0, 0)
LOY = c(1, 0, 0, 0, 1, 1, 0)
mob_path = rbind(IMAG, EXPE, QUAL, VAL, SAT, COM, LOY)

# blocks of indicators (outer model)
**TECHMOX Algorithm: Segmentation Trees in PLS Path Modeling**

**Description**

The function `techmox` calculates a binary segmentation tree following the TECHMOX algorithm. In contrast, `fixNtechmox` obtains a supervised TECHMOX tree in the sense of allowing the user to interactively fix the partitions along the construction process of the tree.

**Usage**

```
 techmox(pls, EXEV, X = NULL, signif = 0.05, size = 0.1, 
 deep = 2, tree = TRUE)
```

**Arguments**

- **pls**: An object of class "plspm" returned by `plspm`.
- **EXEV**: A data frame of factors containing the segmentation variables.
- **X**: Optional dataset (matrix or data frame) used when argument `dataset=NULL` inside `pls`.
- **signif**: A numeric value indicating the significance threshold of the F-statistic. Must be a decimal number between 0 and 1.
- **size**: A numeric value indicating the minimum size of elements inside a node.
deep  An integer indicating the depth level of the tree. Must be an integer greater than 1.

tree  A logical value indicating if the tree should be displayed (TRUE by default).

Details
The argument EXEV must be a data frame containing segmentation variables as factors (see factor). The number of rows in EXEV must be the same as the number of rows in the data used in pls.

The argument size can be defined as a decimal value (i.e. proportion of elements inside a node), or as an integer (i.e. number of elements inside a node).

When the object pls does not contain a data matrix (i.e. pls$data]$null), the user must provide the data matrix or data frame in x.

Value
An object of class “treemox”. Basically a list with the following results:

MOX  Data frame containing the results of the segmentation tree.
FT  Data frame containing the results of the F-tests for each node partition.
candidates  List of data frames containing the candidate splits of each node partition.
list.nodes  List of elements for each node.

Author(s)
Gaston Sanchez

References

See Also
pathmox, plot.treemox treemox.pls.

Examples
## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
data(csimobile)

# select manifest variables
data_mobile = csimobile[,8:33]

# define path matrix (inner model)
treemox.boot

Bootstrapping validation for PATHMOX or TECHMOX trees

Description

Performs bootstrapping validation on path coefficients of terminal nodes from a PATHMOX or TECHMOX tree

Usage

treemox.boot(pls, treemox, X = NULL, br = 100)

Arguments

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pls</strong></td>
<td>An object of class &quot;plspm&quot; returned by plspm.</td>
</tr>
<tr>
<td><strong>treemox</strong></td>
<td>An object of class &quot;treemox&quot; returned by either pathmox or techmox.</td>
</tr>
<tr>
<td><strong>X</strong></td>
<td>Optional dataset (matrix or data frame) used when argument dataset=NULL inside pls.</td>
</tr>
<tr>
<td><strong>br</strong></td>
<td>An integer indicating the number bootstrap resamples (br=100 by default).</td>
</tr>
</tbody>
</table>
Details
The default number of re-samples is 100. However, br can be specified in a range from 50 to 500. When the object pls does not contain a data matrix (i.e. pls$data=NULL), the user must provide the data matrix or data frame in X.

Value
An object of class "bootnodes". Basically a list with the following results:

- **PC**: Matrix of original path coefficients for the root node and the terminal nodes.
- **PMB**: Matrix of bootstrap path coefficients (mean value) for the root node and the terminal nodes.
- **PSB**: Matrix of bootstrap standard errors of path coefficients for the root node and the terminal nodes.
- **PP05**: Matrix of 0.05 bootstrap percentile of path coefficients for the root node and the terminal nodes.
- **PP95**: Matrix of 0.95 bootstrap percentile of path coefficients for the root node and the terminal nodes.

Author(s)
Gaston Sanchez

See Also
pathmox, techmox, treemox.pls.

Examples
```r
## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
data(csimobile)

# select manifest variables
data_mobile = csimobile[,8:33]

# define path matrix (inner model)
imag = c(0, 0, 0, 0, 0, 0, 0)
expe = c(1, 0, 0, 0, 0, 0, 0)
qual = c(0, 1, 0, 0, 0, 0, 0)
val = c(0, 1, 1, 0, 0, 0, 0)
sat = c(1, 1, 1, 0, 0, 0, 0)
com = c(0, 0, 0, 1, 0, 0, 0)
loy = c(1, 0, 0, 0, 1, 1, 0)
mob_path = rbind(imag, expe, qual, val, sat, com, loy)

# blocks of indicators (outer model)
mob_blocks = list(1:5, 6:9, 10:15, 16:18, 19:21, 22:24, 25:26)
mob_modes = rep("A", 7)
```
# apply plspm
mob_pls = plspm(data_mobile, mob_path, mob_blocks, modes = mob_modes, 
    scheme = "factor", scaled = FALSE)

# re-ordering those segmentation variables with ordinal scale
# (Age and Education)
csimobile$Education = factor(csimobile$Education, 
    levels = c("basic", "highschool", "university"),
    ordered = TRUE)

# select the segmentation variables
seg_vars = csimobile[,1:7]

# Pathmox Analysis
mob_pathmox = pathmox(mob_pls, seg_vars, signif=.1, size=.1, deep=2)

# applying function treemox.pls
mob_nodes_boot = treemox.boot(mob_pls, mob_pathmox)

# plot of results for path coefficient number 12
plot(mob_nodes_boot, pc=12)

## End(Not run)

---

treemox.pls  

**PLS-PM results of terminal nodes from a PATHMOX or TECHMOX tree**

---

**Description**

Calculates basic PLS-PM results for the terminal nodes of PATHMOX and TECHMOX trees

**Usage**

treemox.pls(pls, treemox, X = NULL)

**Arguments**

- **pls**: An object of class "plspm" returned by `plspm`.
- **treemox**: An object of class "treemox" returned by `pathmox` or `techmox`.
- **X**: Optional dataset (matrix or data frame) used when argument dataset=NULL inside pls.

**Details**

The argument `pls` must be the same used for calculating the `treemox` object. When the object `pls` does not contain a data matrix (i.e. `pls$data=NULL`), the user must provide the data matrix or data frame in `X`. 
Value

An object of class "treemox.pls". Basically a list with the following results:

weights       Matrix of outer weights for each terminal node
loadings      Matrix of loadings for each terminal node
paths         Matrix of path coefficients for each terminal node
r2            Matrix of r-squared coefficients for each terminal node

Author(s)

Gaston Sanchez

See Also

pathmox, techmox, plot.treemox.

Examples

## Not run:
## example of PLS-PM in customer satisfaction analysis
## model with seven LVs and reflective indicators
data(csimobile)

# select manifest variables
data_mobile = csimobile[,8:33]

# define path matrix (inner model)
IMAG = c(0, 0, 0, 0, 0, 0)
EXPE = c(1, 0, 0, 0, 0, 0)
QUAL = c(0, 1, 0, 0, 0, 0)
VAL = c(0, 1, 1, 0, 0, 0)
SAT = c(1, 1, 1, 1, 0, 0)
COM = c(0, 0, 0, 1, 0, 0)
LOY = c(1, 0, 0, 1, 1, 0)

mob_path = rbind(IMAG, EXPE, QUAL, VAL, SAT, COM, LOY)

# blocks of indicators (outer model)
mob_blocks = list(1:5, 6:9, 10:15, 16:18, 19:21, 22:24, 25:26)
mob_modes = rep("A", 7)

# apply pls

mob_pls = pls(csimobile, mob_path, mob_blocks, modes = mob_modes,
              scheme = "factor", scaled = FALSE)

# re-ordering those segmentation variables with ordinal scale (Age and Education)
csimobile$Education = factor(csimobile$Education,
                              levels=c("basic","highschool","university"),
                              ordered=TRUE)

# select the segmentation variables
seg_vars = csimobile[,1:7]
# Pathmox Analysis
mob_pathmox = pathmox(mob_pls, seg_vars, signif=.10, size=.10, deep=2)

# applying function treemox.pls
mob_nodes = treemox.pls(mob_pls, mob_pathmox)

# comparative barplots
plot(mob_nodes)

## End(Not run)
Index

*Topic **datasets**
   csibank, 3
   csimobile, 4

*Topic **package**
   pathmox-package, 2

csibank, 3
   csimobile, 4

factor, 7, 16
fix.pathmox (pathmox), 6
fix.techmox (techmox), 15

pathmox, 6, 9, 12, 14, 16–20
pathmox-package, 2
plot.bootnodes, 8
plot.treemox, 7, 9, 16, 20
plot.treemox.pls, 11
plspm, 6, 14, 15, 17, 19
plspmox, 13

technox, 7, 9, 12, 15, 17–20

treemox.boot, 17
treemox.pls, 7, 11, 12, 16, 18, 19