Package ‘genpathmox’

February 19, 2015

Title Generalized PATHMOX Algorithm for PLS-PM, LS and LAD Regression

Version 0.2

Description genpathmox provides a very interesting solution for handling segmentation variables in complex statistical methodology. It contains an extended version of the PATHMOX algorithm in the context of partial least square path modeling (Sanchez, 2009) including the F-block test (to detect the responsible latent endogenous equations of the difference), the F-coefficient (to detect the path coefficients responsible of the difference) and the invariance test (to realize a comparison between the sub-models’ latent variables). Furthermore, the package contains a generalized version of the PATHMOX algorithm to approach different methodologies: linear regression and least absolute regression models.

Depends R (>= 3.1.1), plspm, quantreg, mice, diagram, methods

License GPL-3

LazyData true

Author `Giuseppe Lamberti` [aut, cre]

Maintainer `Giuseppe Lamberti` <giusepplamb@hotmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-11-27 20:55:35

R topics documented:

  fibtele ................................................................. 2
  fibtelereg ............................................................. 4
  info.pls_class ....................................................... 5
  info.reg_class ....................................................... 5
  invariance_test ...................................................... 5
  node-class ........................................................... 8
  node.reg_class ....................................................... 8
  plot.treemodel ....................................................... 8
  plot.treemodelreg ................................................... 10
Description

Fibtele

Usage

fibtele

Format

A data frame with 147 observations on the following 35 variables. The first ten variables are segmentation variables. The rest of the variables refer to five latent concepts: 1) Image=Image, 2) Qual.spec=Specific Quality, 3) Qual.gen=Generic Quality, 4) Value=Value, 5) Satis=Satisfaction.

Variables description

• Image: Generic students perception of ICT schools: (internationally recognized, ranges of courses, leader in research).
• Qual.spec: Perception about the achieved quality on the specific skills in the school.
• Qual.gen: Perception about achieved quality on the generic skills in the school (abilities in solving problem, communication skills).
• Value: The advantage or profit that the alumni may draw from the school degree (well paid job, motivated job, prospects in improvement and promotion).
• Satis: Degree of alumni satisfaction about the formation in school respect to their actual work conditions.

Manifest variables description

• ima1MV: It is the best college to study IE
• ima2MV: It is internationally recognized
• ima3MV: It has a wide range of courses
• ima4MV: The Professors are good
• ima5MV: Facilities and equipment are good
• ima6MV: It is leader in research
• ima7MV: It is well regarded by the companies
• ima8MV: It is oriented to new needs and technologies
• quaf1MV: Basic skills
• quaf2MV: Specific Technic skills
• quaf3MV: Applied skills
• qutr1MV: Achieved abilities in solving problem
• qutr2MV: Training in business management
• qutr3MV: The written and oral communication skills
• qutr4MV: Planning and time management acquired
• qutr5MV: Team-work skills
• va11MV: It has allowed me to find a well paid job
• va12MV: I have good prospectives in improvement and promotion
• va13MV: It has allowed me to find a job that motivates me
• va14MV: The training received is the basis on which I will develope my career
• sat1MV: I am satisfied with the training received
• sat2MV: I am satisfied with my current situation
• sat3MV: I think I will have a good career
• sat4MV: What do you think is the prestige of your work

Segmentation Variables description

• Career a factor with levels E1 ETS TEL
• Gender a factor with levels female male
• Age a factor with levels 25-26 years 27-28 years 29-30 years 31 years+
• Studying a factor with levels no stud yes stud
• Contract a factor with levels fix cont other cont temp cont
• Salary a factor with levels 18k>45k 25k 35k 45k
• Firm type a factor with levels priva publi
• Accgrade a factor with levels 7 accnote accnote<7 accnote>8
• Grade a factor with levels <6.5 note >7.5 note 6.5-7 note 7-7.5 note
• Startwork a factor with levels after grad before grad

Source

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

References

Description

Fibtelereg dataset

Usage

fibtelereg

Format

A data frame with 147 observations on the following 18 variables. The first ten variables are segmentation variables. The rest of the variables refer to five variables 1) Image = Image, 2) Exp.spec = Specific Expectation, 3) Exp.gen = Generic Expectation, 4) Qual.spec = Specific Quality, 5) Qual.gen = Generic Quality, 6) Value = Value, 7) Satis = Satisfaction. Variables description

- Image: Generic students perception of ICT schools: (internationally recognized, ranges of courses, leader in research).
- Exp.spec: Specific Expectation on specific skills (technic or applied skills).
- Exp.gen: Generic Expectation on generic skills (abilities in problem solving, communication skills).
- Qual.spec: Perception about the achieved quality on the specific skills in the school.
- Qual.gen: Perception about achieved quality on the generic skills in the school (abilities in solving problem, communication skills).
- Value: The advantage or profit that the alumni may draw from the school degree (well paid job, motivated job, perspectives in improvement and promotion).
- Satis: Degree of alumni satisfaction about the formation in school respect to their actual work conditions.

Segmentation Variables description

- Career: a factor with levels EI ETS TEL
- Gender: a factor with levels female male
- Age: a factor with levels 25-26years 27-28years 29-30years 31years+
- Studying: a factor with levels no.stud yes.stud
- Contract: a factor with levels fix.cont other.cont temp.cont
- Salary: a factor with levels 18k >45k 25k 35k 45k
- Firma: a factor with levels priva publi
- Acc grade: a factor with levels 7-8 accnote accnote<7 accnote>8
- Grade: a factor with levels <6.5 note >7.5 note 6.5-7 note 7-7.5 note
- Startwork: a factor with levels after.grad before.grad
Source

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

References


---

**info.pls_class**  
**info.pls class**

**Description**

info.pls is a S4 class that contains info on the variable and his levels that provides the best binary split and the the the Fischers statistics: F-global, F-coefficientes

---

**info.reg_class**  
**info.reg class**

**Description**

info.pls is a S4 class that contains info on the variable and his levels that provides the best binary split and the the the Fischers statistics: F-global, F-coefficientes

---

**invariance_test**  
**Invariance Test**

**Description**

The invariance test is a test that allows to verify the existence of common weights for the different local PLS-PM models identified by one or more segmentation variable.

**Usage**

invariance_test(x, nodes, inner, outer, mode, scheme, scaling, scaled)
Arguments

x  Matrix or data frame containing the manifest variables.

nodes  List of vectors. Each vector contains the position of the individual that belongs to a specific node.

inner  A square (lower triangular) boolean matrix representing the inner model (i.e. the path relationships between latent variables).

outer  list of vectors with column indices or column names from Data indicating the sets of manifest variables forming each block (i.e. which manifest variables correspond to each block).

mode  character vector indicating the type of measurement for each block. Possible values are: "A", "B", "newA", "PLScore", "PLScow". The length of mode must be equal to the length of outer.

scheme  string indicating the type of inner weighting scheme. Possible values are "centroid", "factorial", or "path".

scaling  optional list of string vectors indicating the type of measurement scale for each manifest variable specified in blocks. scaling must be specified when working with non-metric variables. Possible values: "num" (numeric), "raw", "nom" (nominal), and "ord" (ordinal).

scaled  whether manifest variables should be standardized. Only used when scaling = NULL. When (TRUE, data is scaled to standardized values (mean=0 and variance=1).

Details

The "x" refers to a matrix or a data.frame that contains all individuals used for the global PLS-PM estimation. The "nodes" is a list of vectors. Each vector contains the position of the individual that belongs to a specific node. The position is identified by the number of row. For example, the row 4 corresponds to the individual 4. The other parameters are the classical parameter of the function "plspm".

Value

An data.frame res. Basically a list with the following results:

chisq.statistic  A Number; X^2 statistic

p.value  A Number; p-value

dfH0  A Number; degree of freedom null Hypothesis

dfH1  A Number; degree of freedom alternative Hypothesis

avg.weights  data frame of the common weights if they exist

test  data frame with summary information of the invariance test

Author(s)

Giuseppe Lamberti, Tomas Aluja
References


Examples

```r
## Not run:
## example of PLS-PM in alumni satisfaction

data(fibtele)

data.fib <- fibtele[,12:35]

# select manifest variables
data.fib <- fibtele[,12:35]

# define inner model matrix
Image = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen = rep(0,5)
Value = c(1,1,1,0,0)
Satis = c(1,1,1,1,0)
inner.fib <- rbind(Image,Qual.spec, Qual.gen, Value, Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8:9:11,12:16,17:20,21:24)
modes.fib = rep("A", 5)

# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, modes.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib$Age = factor(seg.fib$Age, ordered=T)
seg.fib$Salary = factor(seg.fib$Salary, levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
seg.fib$Accgrade = factor(seg.fib$Accgrade, levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
seg.fib$Grade = factor(seg.fib$Grade, levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

# Pathmox Analysis
fib.pathmox=pls.pathmox(pls.fib,seg.fib,signif=0.05,deep=2,size=0.2,n.node=20)

# Select the terminal nodes
ls(fib.pathmox)
terminal.nodes=fib.pathmox$terminal

# Invariance test
inv.test=invariance_test(data.fib,terminal.nodes,inner.fib, outer.fib,modes.fib,scheme="centroid",scale=T)
```
inv.test

## End(Not run)

---

<table>
<thead>
<tr>
<th>node-class</th>
<th>node class</th>
</tr>
</thead>
</table>

**Description**

node is a S4 class that contains info on each node of the binary segmentation tree

---

<table>
<thead>
<tr>
<th>node.reg_class</th>
<th>node.reg class</th>
</tr>
</thead>
</table>

**Description**

info.pls is a S4 class that contains element of the node class

---

<table>
<thead>
<tr>
<th>plot.treemodel</th>
<th>Comparative plot between nodes from the Pathmox Segmentation Trees: PLS-PM</th>
</tr>
</thead>
</table>

**Description**

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

**Usage**

```r
## S3 method for class 'treemodel'
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, cex.main = 1, ...)
```
Arguments

- **x**: An object of class "treemodel" returned by `pls.treemodel`.
- **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 ordered.
- **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`.
- **cex.main**: Allows to fix the size of the main. Equal to 1 to default
- **...**: Further arguments are ignored.

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.

Author(s)

Giuseppe Lamberti

References

Examples

```r
## Not run:
## example of PLS-PM in alumni satisfaction

data(fibtele)

# select manifest variables
data.fib <- fibtele[,12:35]

# define inner model matrix
Image = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen = rep(0,5)
Value = c(1,1,1,0,0)
Satis = c(1,1,1,1,0)
inner.fib <- rbind(Image,Qual.spec, Qual.gen, Value, Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8,9:11,12:16,17:20,21:24)
modes.fib = rep("A", 5)

# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, modes.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib = fibtele[,2:11]
seg.fib$Age = factor(seg.fib$Age, ordered=T)
seg.fib$Salary = factor(seg.fib$Salary,
levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
seg.fib$Accgrade = factor(seg.fib$Accgrade,
levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
seg.fib$Grade = factor(seg.fib$Grade,
levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

# Pathmox Analysis
fib.pathmox=pls.pathmox(pls.fib,seg.fib,signif=0.05,
deep=2,size=0.2,n.node=20)

fib.comp=pls.treemodel(pls.fib,fib.pathmox)
plot(fib.comp)

## End(Not run)
```

---

```
plot.treemodelreg

Comparative plot between nodes from or the Pathmox Segmentation Trees: linear and LAD regression
```
Description

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

Usage

```r
## S3 method for class 'treemodelreg'
plot(x, main.node = FALSE, names.nodes = NULL,
     eti = FALSE, lab.vec = NULL, short.min = NULL, cex.names = 1,
     cex.axis = 1.2, cex.main = 1, lim = c(-0.5, 0.5), short.labs = TRUE,
     ...)```

Arguments

- `x`: An object of class "treemodelreg" returned by `reg.treemodel`
- `main.node`: It is string. If equal to TRUE you have to indicate the main of each barplot in "names.nodes".
- `names.nodes`: Optional vector of names for each the terminal node (must be a vector of length equal to the number of terminal nodes).
- `eti`: is string. If it is TRUE the label of each coefficients for all the terminal nodes must be specify in "lab.vec". If it is false the labels are defined by the program.
- `lab.vec`: Optional vector of names for each coefficient of the terminal nodes (must be a vector of length equal to the number of coefficients).
- `short.min`: Integer number indicating the minimum length of the.
- `cex.names`: Allows to fix the size of coefficient labels. Equal to 1 to default.
- `cex.axis`: Allows to fix the size of axes. Equal to 1.2 to default.
- `cex.main`: Allows to fix the size of the main. Equal to 1 to default.
- `lim`: Allows to fix the axes interval. Equal to (-0.5,0.5) to default.
- `short.labs`: Logical value indicating if the labels of the barplots.
- `...`: Further arguments passed on to `plot.treemodelreg`

Details

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

Author(s)

Giuseppe Lamberti

References

Examples

## Not run:
# example of LM in alumni satisfaction

data(fibtelereg)
segvar = fibtelereg[,2:11]
data.fib=fibtelereg[,12:18]
segvar$Age = factor(segvar$Age, ordered=T)
segvar$Salary = factor(segvar$Salary,
levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
segvar$Accgrade = factor(segvar$Accgrade,
levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
segvar$Grade = factor(segvar$Grade,
levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

fib.reg.pathmox = reg.pathmox(Satisfact~, .data=data.fib, segvar,
signif=0.05, deep=2, method="lm", size=0.15)

# terminal nodes comparison
fib.node.comp = reg.treemodel(fib.reg.pathmox)
plot(fib.node.comp)

## End(Not run)

---

**plot.xtree.pls**

*Plot function for the Pathmox Segmentation Trees: PLS-PM*

**Description**

The function `plot.xtree.pls` allows to draw PATHMOX tree for PLS-PM

**Usage**

```r
## S3 method for class 'xtree.pls'
plot(x, root.col = "grey", node.col = "orange",
     leaf.col = "green2", shadow.size = 0.003, node.shadow = "red",
     leaf.shadow = "darkgreen", cex = 0.7, seg.col = "blue3", lwd = 1,
     show.pval = TRUE, pval.col = "blue", main = NULL, cex.main = 1, ...)
```

**Arguments**

- `x` An object of class "xtree.pls" returned by `pls.pathmox`
- `root.col` Fill color of root node.
node.col Fill color of child nodes.
leaf.col Fill color of leaf.
shadow.size Relative size of shadows.
node.shadow Color of shadow of child nodes.
leaf.shadow Color of shadow of leaf nodes.
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.
lwd The line width, a positive number, defaulting to 1
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.
... Further arguments passed on to plot.xtree.pls.

Examples

## Not run:
## example of PLS-PM in alumni satisfaction

data(fibtele)

# select manifest variables
data.fib <- fibtele[,12:35]

# define inner model matrix
Image = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen = rep(0,5)
Value = c(1,1,1,0,0)
Satis = c(1,1,1,1,0)
inner.fib <- rbind(Image, Qual.spec, Qual.gen, Value, Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8,9:11,12:16,17:20,21:24)
modes.fib = rep("A", 5)

# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, modes.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib$Age = factor(seg.fib$Age, ordered=T)
seg.fib$Salary = factor(seg.fib$Salary, levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
seg.fib$Accgrade = factor(seg.fib$Accgrade,
levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
seg.fib$Grade = factor(seg.fib$Grade, 
levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

# Pathmox Analysis
fib.pathmox=pls.pathmox(pls.fib,seg.fib,signif=0.05, 
deep=2,size=0.2,n.node=20)

# plot pathmox tree
plot(pls.fib)

## End(Not run)

---

**plot.xtree.reg**

*Plot function for the Pathmox Segmentation Trees: linear regression and LAD*

**Description**

The function `plot.xtree.reg` allows to draw PATHMOX tree for linear and LAD regression

**Usage**

```r
## S3 method for class 'xtree.reg'
plot(x, root.col = "grey", node.col = "orange", 
     leaf.col = "green2", shadow.size = 0.003, node.shadow = "red", 
     leaf.shadow = "darkgreen", cex = 0.7, seg.col = "blue3", lwd = 1, 
     show.pval = TRUE, pval.col = "blue", main = NULL, cex.main = 1, ...
)
```

**Arguments**

- **x**: An object of class "xtree.reg" returned by `reg.pathmox`
- **root.col**: Fill color of root node.
- **node.col**: Fill color of child nodes.
- **leaf.col**: Fill color of leaf nodes.
- **shadow.size**: Relative size of shadows.
- **node.shadow**: Color of shadow of child nodes.
- **leaf.shadow**: Color of shadow of leaf nodes.
- **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.
- **lwd**: The line width, a positive number, defaulting to 1.
- **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.
- **...**: Further arguments are ignored.
Examples

```r
# Not run:
# example of LM in alumni satisfaction

data(fibtelereg)

segvar = fibtelereg[,2:11]

data.fib= fibtelereg[,12:18]

segvar$Age  = factor(segvar$Age, ordered=T)
segvar$Salary = factor(segvar$Salary, levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
segvar$Accgrade = factor(segvar$Accgrade, levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
segvar$Grade  = factor(segvar$Grade, levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

fib.reg.pathmox = reg.pathmox(Satisfact~., data=data.fib, segvar, 
signif=0.05, deep=2, method="lm", size=0.15)

plot(fib.reg.pathmox)
```

## End(Not run)

---

**pls.pathmox**

**PATHMOX-PLS: Extended Segmentation Trees in Partial Least Squares Path Modeling**

**Description**

The function `pathmox.pls` calculates a binary segmentation tree in the context PLS-PM following the PATHMOX algorithm. This function extends the pathmox algorithm introduced by Sanchez in 2009 including the two new test: the F-block test (to detect the responsible latent endogenous equations of the difference), the F-coefficient test (to detect the path coefficients responsible of the difference). The F-tests used in the split process are implemented following the classic least square estimation. An implementation of the tests following the LAD regression also are proposed to overcome the parametric hypothesis of the F-test.

**Usage**

```r
pls.pathmox(xpls, SVAR, signif, deep, method = "lm", size, X = NULL, 
tree = TRUE, n.node = 30, ...)
```

**Arguments**

- `xpls` : An object of class "plspm" returned by `plspm`. 
- `SVAR`, `signif`, `deep`, `method`, `size`, `X`, `tree`, `n.node` are arguments of `plspm`.

---
SVAR  A data frame of factors containing the segmentation variables.
signif  A numeric value indicating the significance threshold of the F-statistic. Must be a decimal number between 0 and 1.
depth  An integer indicating the depth level of the tree. Must be an integer greater than 1.
method  A string indicating the criterion used to calculate the test can be equal to "lm" or "lad".
size  A numeric value indicating the minimum size of elements inside a node.
X  Optional dataset (matrix or data frame) used when argument dataset=NULL inside pls.
tree  A logical value indicating if the tree should be displayed (TRUE by default).
n.node  It is the minimum number of individuals to consider a candidate partition (30 by default).
...  Further arguments passed on to pls.pathmox.

Details

The argument xpls is object of class "plspm" returned by plspm.
The argument SVAR must be a data frame containing segmentation variables as factors. The number of rows in SVAR must be the same as the number of rows in the data used in pls.
The argument signif represent the p-value level takes as reference to stop the tree partitions.
The argument depth represent the depth level of the tree takes as reference to stop the tree partitions.
The argument method is a string containig the criterion used to calculate the tests; if method="lm" the classic least square approach is used to perform the tests; if method="lad" the LAD (least absolute deviation regression) is used.
The argument size is defined as a decimal value (i.e. proportion of elements inside a node).
The argument n.node is the minimum number of individuals to consider a candidate partition. If the candidate split produces a partition where the number of individuals is less then n.node, the partition is not considered.

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 "xtree.pls". Basically a list with the following results:

MOX  Data frame with the results of the segmentation tree
root  List of elements contained in the root node
terminal  List of elements contained in terminal nodes
nodes  List of elements contained in all nodes: terminal and intermediate
candidates  List of data frames containing the candidate splits of each node partition
Fg.r  Data frame containing the results of the F-global test for each node partition
**Fb.r**  List of data frames containing the results of the F-block test for each node partition

**Fc.r**  A list of data frames containing the results of the F-coefficients test for each node partition

**model**  Information about the internal parameters

**Author(s)**

Giuseppe Lamberti

**References**


**Examples**

```r
## Not run:
## example of PLS-PM in alumni satisfaction
data(fibtele)

data.fib <- fibtele[,12:35]

# select manifest variables
data.fib <- fibtele[,12:35]

# define inner model matrix
Image   = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen   = rep(0,5)
Value   = c(1,1,1,0,0)
Satis   = c(1,1,1,1,0)
inner.fib <- rbind(Image,Qual.spec,Qual.gen,Value,Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8,9:11,12:16,17:20,21:24)
modes.fib = rep("A", 5)

# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, modes.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib$Age = factor(seg.fib$Age, ordered=T)
seg.fib$Salary = factor(seg.fib$Salary,
levels=c("<18k" ,"25k" ,"35k" ,"45k" ,">45k" ), ordered=T)
seg.fib$Accgrade = factor(seg.fib$Accgrade,
levels=c("accnote<7" ,"7-8accnote" ,"accnote>8" ), ordered=T)
```
pls.treemodel

PLS-PM results of terminal nodes from the Pathmox Segmentation Trees

Description

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

Usage

pls.treemodel(xpls, xtree, X = NULL, alpha = 0.05, terminal = TRUE,
               scaled = FALSE, label = FALSE, label.nodes = NULL, ...)

Arguments

- **xpls**: An object of class "plspm" returned by plspm.
- **xtree**: An object of class "xtree.pls" returned by pls.pathmox.
- **X**: Optional dataset (matrix or data frame) used when argument dataset=NULL inside xpls.
- **alpha**: is numeric value indicating the significance threshold of the invariance test
- **terminal**: is string, if equal to TRUE, just the terminal nodes are considered for the output results. when it is equal to FALSE, the PLS-PM results are generated for all nodes of the tree
- **scaled**: to standardize the latent variables or not
- **label**: is a string. It is false for defect. If it is TRUE, label.nodes has to be fix.
- **label.nodes**: is a vector with the name of the nodes. It is null for defect.
- **...**: Further arguments passed on to pls.treemodel.

Details

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

The argument xtree is an object of class "xtree.pls" returned by pls.pathmox.
Value

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

- **inner**: Matrix of the inner relationship between latent variables of the PLS-PM model
- **invariance.test**: A data frame containing the results of the invariance test
- **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
- **sign**: list of matrix with the significance for each terminal node

Author(s)

Giuseppe Lamberti

References


See Also

- `pls.pathmox`, `plot.xtree.pls`

Examples

```r
# Not run:
# example of PLS-PM in alumni satisfaction

data(fibtele)

# select manifest variables
data.fib < fibtele[,12:35]

# define inner model matrix
Image  = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen  = rep(0,5)
Value    = c(1,1,1,0,0)
Satis    = c(1,1,1,1,0)
inner.fib <- rbind(Image,Qual.spec,Qual.gen,Value,Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8,9:11,12:16,17:20,21:24)
modes.fib = rep("A", 5)
```
# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, modes.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib$fibtele[2:1]

seg.fib$Age = factor(seg.fib$Age, ordered=T)
seg.fib$Salary = factor(seg.fib$Salary,
levels=c("<18k", "25k", "35k", "45k",">45k"), ordered=T)
seg.fib$Accgrade = factor(seg.fib$Accgrade,
levels=c("<acnote<7", "7-8acnote", "acnote>8"), ordered=T)
seg.fib$Grade = factor(seg.fib$Grade,
levels=c("<6.5note", "6.5-7note", "7-7.5note",">7.5note"), ordered=T)

# Pathmox Analysis
fib.pathmox=pls.pathmox(pls.fib, seg.fib, signif=0.05,
deep=2, size=0.2, n.node=20)

fib.comp=pls.treemodel(pls.fib, fib.pathmox)

## End(Not run)
Examples

## Not run:
## example of PLS-PM in alumni satisfaction

data(fibtele)

# select manifest variables
data.fib <- fibtele[,1:35]

# define inner model matrix
Image = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen = rep(0,5)
Value = c(1,1,1,0,0)
Satis = c(1,1,1,1,0)
inner.fib <- rbind(Image, Qual.spec, Qual.gen, Value, Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8, 9:11, 12:16, 17:20, 21:24)
modes.fib = rep("A", 5)

# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, modes.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib <- fibtele[,2:11]
seg.fib$Age = factor(seg.fib$Age, ordered=T)
seg.fib$Salary = factor(seg.fib$Salary, levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
seg.fib$Accgrade = factor(seg.fib$Accgrade, levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
seg.fib$Grade = factor(seg.fib$Grade, levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

# Pathmox Analysis
fib.pathmox = pls.pathmox(pls.fib, seg.fib, signif=0.05, deep=2, size=0.2, n.node=20)

print(fib.pathmox)

## End(Not run)
Description

The function `print.xtree.reg` print the `reg.pathmox` tree

Usage

```r
## S3 method for class 'xtree.reg'
print(x, ...)
```

Arguments

- `x` An object of class "xtree.reg".
- `...` Further arguments are ignored.

Author(s)

Giuseppe Lamberti

References


`summary.xtree.pls`.

Examples

```r
## Not run:
# example of LM in alumni satisfaction

data(fibtelereg)

segvar= fibtelereg[,2:11]

data.fib=fibtelereg[,]2:18]

segvar$Age  = factor(segvar$Age, ordered=T)
segvar$Salary = factor(segvar$Salary, levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
segvar$Accgrade = factor(segvar$Accgrade, levels=c("acnote<7","7-8acnote","acnote>8"), ordered=T)
segvar$Grade  = factor(segvar$Grade, levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

fib.reg.pathmox=reg.pathmox(Satisfact~.,data=data.fib,segvar,
signif=0.05,deep=2,method="lm",size=0.15)

print(fib.reg.pathmox)

## End(Not run)
Description

The function `regNpathmox` calculates a binary segmentation tree in the context of linear regression following the PATHMOX algorithm. This function generalizes the Pathmox algorithm introduced by Sanchez in 2009 to the context of linear and LAD regression.

Usage

```r
regNpathmox(formula, svar, signif, deep, method, size, tree = TRUE, data = NULL, ...)
```

Arguments

- `formula`: An object of class "formula".
- `svar`: A data frame of factors containing the segmentation variables.
- `signif`: A numeric value indicating the significance threshold of the F-statistic. Must be a decimal number between 0 and 1.
- `deep`: An integer indicating the depth level of the tree. Must be an integer greater than 1.
- `method`: A string indicating the criterion used to calculate the test; can be equal to "lm" or "lad" node.
- `size`: A numeric value indicating the minimum size of elements inside a node.
- `tree`: A logical value indicating if the tree should be displayed (TRUE by default).
- `data`: An optional data frame.
- `...`: Further arguments passed on to `regNpathmox`.

Details

The argument `formula` is an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.

The argument `svar` must be a data frame containing segmentation variables as factors. The number of rows in `svar` must be the same as the number of rows in the data.

The argument `signif` represents the p-value level that takes as reference to stop the tree partitions.

The argument `deep` represents the p-value level that takes as reference to stop the tree partitions.

The argument `method` is a string containing the criterion used to calculate the test; if `method="lm"` the classic least square approach is used to perform the test; if `method="lad"` the LAD (least absolute deviation) is used.

The argument `size` has defined as a decimal value (i.e. proportion of elements inside a node).
Value

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

- **MOX**: Data frame with the results of the segmentation tree
- **root**: element of containing in the root node
- **terminal**: element of containing in the terminal nodes
- **nodes**: element of containing in all nodes terminal and intermediate
- **candidates**: List of data frames containing the candidate splits of each node partition
- **Fg.r**: Data frame containing the results of the F-global test for each node partition
- **Fc.r**: A list of Data frames containing the results of the F-coefficients test for each node partition
- **model**: Information about the internal parameters

Author(s)

Giuseppe Lamberti

References


Examples

```r
## Not run:
# example of LM in alumni satisfaction

data(fibtelereg)

segvar = fibtelereg[,2:11]
data.fib=fibtelereg[,12:18]

segvar$Age = factor(segvar$Age, ordered=T)
segvar$Salary = factor(segvar$Salary, 
  levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
segvar$Accgrade = factor(segvar$Accgrade, 
  levels=c("acnote<7","7-8acnote","acnote>8"), ordered=T)
segvar$Grade  = factor(segvar$Grade, 
  levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

fib.reg.pathmox=reg.pathmox(Satisfact~.,data=data.fib,segvar, 
  signif=0.05,deep=2,method="lm",size=0.15)
```

## End(Not run)
**reg.treemodel**

| reg.treemodel | Regression results of terminal nodes from the Pathmox Segmentation Trees |

**Description**

Calculates basic regression results for the terminal nodes of Pathmox Segmentation Trees: linear regression and LAD trees

**Usage**

```r
greg.treemodel(xtree.reg, terminal = TRUE, intercept = FALSE, label = FALSE, label.nodes = NULL, ...)
```

**Arguments**

- **xtree.reg**: An object of class "xtree.reg" returned by `reg.pathmox`.
- **terminal**: is string, if equal to `TRUE`, just the terminal nodes are considered for the output results. when it is equal to `FALSE`, the regression results are generated for all nodes of the tree.
- **intercept**: if equal to `TRUE` also the intercept is considered in the estimation.
- **label**: is a boolean. `T` is false for defect. If it is `TRUE`, `label.nodes` has to be fix.
- **label.nodes**: is a vector with the name of the nodes. It is null for defect.
- **...**: Further arguments passed on to `reg.treemodel`.

**Details**

The argument `xtree.reg` is an object of class "xtree.reg" returned by `reg.pathmox`.

**Value**

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

- **inner**: Matrix of the inner relationship between latent variables of the PLS-PM model.
- **method**: A string containing the used method ("lm" or "lad").
- **coefficients**: Matrix coefficients for each terminal node.
- **std**: Matrix of standard deviation of coefficients for each terminal node.
- **pval.coef**: Matrix of p-value significance for each terminal node.
- **r2**: Matrix of r-squared coefficients for each terminal node.

**Author(s)**

Giuseppe Lamberti
Summary function for the Pathmox Segmentation Trees: PLS-PM

The function `summary.xtree.pls` returns the most important results obtained by the function `pls.pathmox`. In order, it provides the parameters `algorithm` (threshold significance, node size limit, tree depth level and the method used for the split partition), the basic characteristics of the tree (deep and number of terminal nodes), the basic characteristics of the nodes and the F-global the F-block and F-coefficients results. For the test results the significance level is indicated.

References

See Also
`pls.pathmox`, `plot.xtree.pls`

Examples
```r
## Not run:
# example of LM in alumni satisfaction
data(fibtelereg)

segvar = fibtelereg[,2:11]
data.fib = fibtelereg[,12:18]

segvar$Age = factor(segvar$Age, ordered=T)
segvar$Salary = factor(segvar$Salary, levels=c(<18k","25k","35k","45k",">45k"), ordered=T)
segvar$Accgrade = factor(segvar$Accgrade, levels=c("accnote<7","7-8accnote","accnote>8"), ordered=T)
segvar$Grade = factor(segvar$Grade, levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

fib.reg.pathmox = reg.pathmox(Satisfact~.,data=data.fib,segvar, signif=0.05,deep=2,method="lm",size=0.15)

# terminal nodes comparison
fib.node.comp = reg.treemodel(fib.reg.pathmox)

## End(Not run)
```
Usage

```r
## S3 method for class 'xtree pls'
summary(object, ...)
```

Arguments

- `object`: An object of class "xtree pls".
- `...`: Further arguments are ignored.

Author(s)

Giuseppe Lamberti

References


Examples

```r
## Not run:
## example of PLS-PM in alumni satisfaction

data(fibtele)

# select manifest variables
data.fib <- fibtele[,12:35]

# define inner model matrix
Image = rep(0,5)
Qual.spec = rep(0,5)
Qual.gen = rep(0,5)
Value = c(1,1,1,1,0)
Satis = c(1,1,1,1,0)
inner.fib <- rbind(Image, Qual.spec, Qual.gen, Value, Satis)
colnames(inner.fib) <- rownames(inner.fib)

# blocks of indicators (outer model)
outer.fib <- list(1:8,9:11,12:16,17:20,21:24)
model.fib <- rep("A", 5)

# apply plspm
pls.fib <- plspm(data.fib, inner.fib, outer.fib, model.fib)

# re-ordering those segmentation variables with ordinal scale
seg.fib$Age <- factor(seg.fib$Age, ordered=T)
seg.fib$Salary <- factor(seg.fib$Salary,
levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
```
summary.xtree.reg

Summary function for the Pathmox Segmentation Trees: linear regression and LAD

Description

The function summary.xtree.reg returns the most important results obtained by the function reg.pathmox. In order, it provides the parameters algorithm (threshold significance, node size limit, tree depth level and the method used for the split partition), the basic characteristics of the tree (deep and number of terminal nodes), the basic characteristics of the nodes and the F-global and F-coefficients results. For the test results the significance level is indicated.

Usage

## S3 method for class 'xtree.reg'
summary(object, ...)

Arguments

  object An object of class "xtree.reg".

  ... Further arguments are ignored.

Author(s)

Giuseppe Lamberti

References


summary.xtree.pls, reg.pathmox.
Examples

## Not run:

```r
# example of LM in alumni satisfaction

data(fibtelereg)

segvar = fibtelereg[,2:11]

data.fib = fibtelereg[,12:18]

segvar$Age = factor(segvar$Age, ordered=T)
segvar$Salary = factor(segvar$Salary, levels=c("<18k","25k","35k","45k",">45k"), ordered=T)
segvar$Accgrade = factor(segvar$Accgrade, levels=c("<accnote=7","7-8accnote","accnote>8"), ordered=T)
segvar$Grade = factor(segvar$Grade, levels=c("<6.5note","6.5-7note","7-7.5note",">7.5note"), ordered=T)

fib.reg.pathmox = reg.pathmox(Satisfact~., data=data.fib, segvar, signif=0.05, deep=2, method="lm", size=0.15)

summary(fib.reg.pathmox)
```

## End(Not run)

Description

tree is a S4 class that contains info on the binary segmentation tree
Index

*Topic datasets
  fibtele, 2
  fibtelereg, 4

fibtele, 2
fibtelereg, 4

info.pls_class, 5
info.reg_class, 5
invariance_test, 5

node-class, 8
node.reg_class, 8

plot.treemodel, 8
plot.treemodelreg, 10, 11
plot.xtree.pls, 12, 13, 19, 26
plot.xtree.reg, 14
pls.pathmox, 12, 15, 16, 18, 19, 26, 27
pls.treemodel, 9, 18, 18
plspm, 15, 16, 18
print.xtree.pls, 20
print.xtree.reg, 21

reg.pathmox, 11, 14, 23, 23, 25, 28
reg.treemodel, 11, 25, 25

summary.xtree.pls, 20, 22, 26, 28
summary.xtree.reg, 28

tree-class, 29