Package 'plsdepot'

July 23, 2025

Type Package

Title Partial Least Squares (PLS) Data Analysis Methods

Version 0.2.0

Date 2023-03-12

Author Frederic Bertrand [cre] (ORCID:
<https://orcid.org/0000-0002-0837-8281>),
Gaston Sanchez [aut]

Maintainer Frederic Bertrand <frederic.bertrand@utt.fr>

Description Different methods for PLS analysis of
one or two data tables such as Tucker's Inter-Battery, NIPALS,
SIMPLS, SIMPLS-CA, PLS Regression, and PLS Canonical Analysis.
The main reference for this software is the awesome book (in
French) 'La Regression PLS: Theorie et Pratique' by Michel
Tenenhaus.

Depends R (>= 2.15.1)

Suggests FactoMineR

License GPL-3

LazyLoad yes

Collate 'nipals.R' 'plot.nipals.R' 'plot.plsca.R' 'plot.plsreg1.R'
'plot.plsreg2.R' 'plsca.R' 'plsreg1.R' 'plsreg2.R'
'print.nipals.R' 'print.plsca.R' 'print.plsreg1.R'
'print.plsreg2.R' 'interbat.R' 'plot.interbat.R'
'print.interbat.R' 'print.simpls.R' 'print.simplsca.R'
'simpls.R' 'simplsca.R' 'plot.simpls.R' 'plot.simplsca.R'

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Repository CRAN

Date/Publication 2023-03-13 10:00:02 UTC

Contents

`carscomplete` *carscomplete data set*

Description

Data set with six variables measured on 24 cars (no missing values)

Usage

```
data(carscomplete)
```

Format

A data frame with 6 variables measured on the 24 cars.

Num	*Variable*	*Description*
1	Cylindree	Engine
2	Puissance	Power
3	Vitese	Speed
4	Poids	Weight
5	Longueur	Length
6	Largeur	Height

Details

This is data with no missing values

Source

Tenenhaus M. (1998) *La Regression PLS: theorie et pratique*. Table 4, page 67

References

Tenenhaus M. (1998) *La Regression PLS: theorie et pratique* Paris: Editions Technip

Examples

```
data(carscomplete)
head(carscomplete)
```

`carsmissing` *carsmissing data set*

Description

Data set with six variables measured on 24 cars

Usage

```
data(carsmissing)
```

Format

A data frame with 6 variables measured on the 24 cars (contains missing values).

Num	*Variable*	*Description*
1	Cylindree	Engine
2	Puissance	Power
3	Vitese	Speed
4	Poids	Weight
5	Longueur	Length
6	Largeur	Height

Details

This is data contains one missing value in each row

Source

Tenenhaus M. (1998) *La Regression PLS: theorie et pratique*. Table 5, page 69

References

Tenenhaus M. (1998) *La Regression PLS: theorie et pratique* Paris: Editions Technip

Examples

```
data(carsmissing)
head(carsmissing)
```

`cornell` *Cornell data set*

Description

Data describing the composition of 12 octane mixtures (units measured in proportions)

Usage

```
data(cornell)
```

Format

A data frame with 12 observations on the following 8 variables.

Num	*Variable*	*Description*
1	Distillation	Direct Distillation
2	Reformed	Reformed
3	NaphthaTher	Thermal cracking Naphtha
4	NaphthaCat	Catalytic cracking Naphtha
5	Polymer	Polymer
6	Alkylat	Alkylat
7	NatEssence	Natural Essence
8	Octane	Octane

Source

Tenenhaus M. (1998) Table 6, page 78

References

Tenenhaus M. (1998) *La Regression PLS: theorie et pratique* Paris: Editions Technip

Examples

```
data(cornell)
head(cornell)
```

`interbat` *Tucker's Inter-Battery Method of Factor Analysis*

Description

The function `interbat` performs Tucker's Inter-Battery method of factor analysis as described in Michel Tenenhaus book *La Regression PLS*, chapter 3

Usage

```
interbat(X, Y, scaled = TRUE)
```

Arguments

`X`	Numeric matrix or data frame with two or more columns (X-block).
`Y`	Numeric matrix or data frame with two or more columns (Y-block).
`scaled`	Logical value indicating whether to scale the data (`TRUE` by default).

Details

Arguments `X` and `Y` must contain more than one variable. No missing data are allowed.

Value

An object of class `"interbat"`, basically a list with the following elements:

`values`	The extracted eigenvalues
`x.scores`	scores of the X-block (also known as T components)
`x.wgs`	weights of the X-block
`y.scores`	scores of the Y-block (also known as U components)
`y.wgs`	weights of the Y-block
`cor.xt`	correlations between X and T
`cor.yu`	correlations between Y and U
`cor.xu`	correlations between X and U
`cor.yt`	correlations between Y and T
`cor.tu`	correlations between T and U
`R2X`	explained variance of X by T
`R2Y`	explained variance of Y by U
`com.xu`	communality of X with U
`com.yt`	communality of Y with T
`statistic`	Phi statistic values for assessing the number of relevant components

Author(s)

Gaston Sanchez

References

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique.* Paris: Editions TECHNIP.

Tucker, L.R. (1958) An inter-battery method of factor analysis. *Psychometrika*, 23(2): 111-136.

See Also

`plot.interbat`, `plsca`

Examples

```
## Not run:
 # load data linnerud
 data(linnerud)

 # apply inter-battery method
 ib = interbat(linnerud[,1:3], linnerud[,4:6])

 # plot variables
 plot(ib, what="variables")

 # plot observations
 plot(ib, what="observations", comps=c(1,1), where=c("t","u"))

## End(Not run)
```

`linnerud` *Linnerud data set*

Description

Thris data set contains three physiological and three exercise variables measured on 20 men

Usage

```
data(linnerud)
```

Format

A data frame with 20 observations on the following 6 variables.

Num	*Variable*	*Description*
1	Weight	Weight
2	Waist	Waist size
3	Pulse	Pulse

4	Pulls	Pull-ups
5	Squats	Squats
6	Jumps	Jumps

Source

Tenenhaus M. (1998) Table 1, page 15

References

Tenenhaus M. (1998) *La Regression PLS: theorie et pratique* Paris: Editions Technip

Examples

```
data(linnerud)
head(linnerud)
```

`nipals` *NIPALS: Non-linear Iterative Partial Least Squares*

Description

Principal Components Analysis with NIPALS algorithm

Usage

```
nipals(Data, comps = 2, scaled = TRUE)
```

Arguments

`Data`	A numeric matrix or data frame (which may contain missing values).
`comps`	Number of components to be calculated (by default 2)
`scaled`	A logical value indicating whether to scale the data (`TRUE` by default).

Details

The function `nipals` performs Principal Components Analysis of a data matrix that may contain missing values.

Value

An object of class "nipals", basically a list with the following elements:

When the analyzed data contain missing values, the help interpretation tools (e.g. cor.xt, disto, contrib, cos, dmod) may not be meaningful, that is to say, some of the results may not be coherent.

values	The pseudo eigenvalues
scores	The extracted scores (i.e. components)
loadings	The loadings
cor.xt	Correlations between the variables and the scores
disto	Squared distance of the observations to the origin
contrib	Contributions of the observations (rows)
cos	Squared cosinus
dmod	Distance to the Model

Author(s)

Gaston Sanchez

References

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique.* Paris: Editions TECHNIP.

Tenenhaus, M. (2007) *Statistique. Methodes pour decrire, expliquer et prevoir*. Paris: Dunod.

See Also

plot.nipals, plsreg1

Examples

```
## Not run:
 # load datasets carscomplete and carsmissing
 data(carscomplete) # complete data
 data(carsmissing)  # missing values

 # apply nipals
 my_nipals1 = nipals(carscomplete)
 my_nipals2 = nipals(carsmissing)

 # plot variables (circle of correlations)
 plot(my_nipals1, what="variables", main="Complete data")
 plot(my_nipals2, what="variables", main="Missing data")

 # plot observations with labels
 plot(my_nipals1, what="observations", show.names=TRUE, main="Complete data")
 plot(my_nipals2, what="observations", show.names=TRUE, main="Missing data")

 # compare results between my_nipals1 and my_nipals2
```

```
plot(my_nipals1$scores[,1], my_nipals2$scores[,1], type="n")
title("Scores comparison: my_nipals1  -vs-  my_nipals2", cex.main=0.9)
abline(a=0, b=1, col="gray85", lwd=2)
points(my_nipals1$scores[,1], my_nipals2$scores[,1], pch=21,
       col="#5592e3", bg = "#5b9cf277", lwd=1.5)

## End(Not run)
```

`plot.interbat` *Plot inter-battery basic results*

Description

Plot method for objects of class `"interbat"`. This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
 ## S3 method for class 'interbat'
plot(x, what = "variables",
   comps = c(1, 2), where = c("t", "t"), cex = 1,
   col.xlabels = "#5592e3", col.ylabels = "#e3a654",
   yfont = 1, pos = NULL, offset = 0.1,
   col.xarrows = "#5b9cf255", col.yarrows = "#e3a65455",
   lwd = 3, length = 0, angle = 0, col.points = "#5592e3",
   pch = 21, pt.bg = "#5b9cf255", show.names = FALSE,
   xpd = TRUE, xlab = NULL, ylab = NULL, main = NULL,
   col.main = "gray35", cex.main = 1.2,
   col.axis = "gray40", show.grid = TRUE,
   col.grid = "gray95", ...)
```

Arguments

`x`	An object of class `"interbat"`.
`what`	What to plot. Options are `"variables"` and `"observations"`.
`comps`	An integer vector of length two to indicate which components to plot.
`where`	Where to plot the observations. A character vector of length two to indicate which components to use when plotting observations. This parameter will take into account the values in `comps`. Possible options are: `c("t","u")` for using x-y components, `c("t","t")`, for using x components, and `c("u","u")` for using y components. Default `c("t","t")`.
`cex`	Character expansion for labels and points.
`col.xlabels`	Color for labels of X-block variables.
`col.ylabels`	Color for labels of Y-block variables.

yfont	Integer for specifying which font to use for Y-block labels. See font in graphical parameters par.
pos	Position for the text (see graphical paramaters par)
offset	When pos is specified, this value gives the offset of the labels.
col.xarrows	Color for the X-block arrows.
col.yarrows	Color for the Y-block arrows.
lwd	The line width of arrows.
length	Length of the edges of the arrow head (in inches).
angle	Angle from the shaft of the arrow to the edge of the arrow head.
col.points	Color for the points when what="observations".
pch	Plotting character symbol to use (see par).
pt.bg	Background (fill) color for the points given by pch=21:25.
show.names	Logical indicating whether to show labels of points. Only used when what="observations"
xpd	Logical for controlling clipping region of names and labels.
xlab	A title for the x axis.
ylab	A title for the y axis.
main	Main title of the plot.
col.main	Color of main title.
cex.main	Character expansion of main title.
col.axis	Color of axis annotations (tick marks and labels).
show.grid	Logical indicating whether to show grid lines.
col.grid	Color of grid lines. Only used when show.grid=TRUE.
...	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations of each block of variables with the components of the other block: $cor.xu -vs- $cor.yt.

Author(s)

Gaston Sanchez

See Also

interbat

Examples

```
## Not run:
 # load dataset linnerud
 data(linnerud)

 # apply nipals
 ib = interbat(linnerud[,1:3], linnerud[,4:6])

 # plot variables (circle of correlations)
 plot(ib, what="variables")

 # plot observations (as points) using components (t1,u1)
 plot(ib, what="observations", comps=c(1,1), where=c("t","u"))

 # plot observations with names using components (t1,u1)
 plot(ib, what="observations", comps=c(1,1),
      where=c("t","u"), show.names=TRUE)

 # plot observations (as points) using components (t1,t2)
 plot(ib, what="observations", comps=c(1,2), where=c("t","t"))

 # plot observations (as points) using components (u1,u2)
 plot(ib, what="observations", comps=c(1,2), where=c("u","u"))

## End(Not run)
```

`plot.nipals` *Plot NIPALS basic results*

Description

Plot method for objects of class `"nipals"`. This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
 ## S3 method for class 'nipals'
plot(x, what = "variables",
   comps = c(1, 2), cex = 1, col.labels = "#5592e3",
   pos = NULL, offset = 0.1, col.arrows = "#5b9cf255",
   lwd = 3.5, length = 0, angle = 0,
   col.points = "#5592e3", pch = 21, pt.bg = "#5b9cf255",
   show.names = FALSE, xpd = TRUE, xlab = NULL,
   ylab = NULL, main = NULL, col.main = "gray35",
   cex.main = 1.2, col.axis = "gray40", show.grid = TRUE,
   col.grid = "gray95", ...)
```

Arguments

x	An object of class "nipals".
what	What to plot. Options are "variables" and "observations"
comps	An integer vector of length two to indicate which components to plot
cex	Character expansion for labels and points.
col.labels	Color for labels of variables.
pos	Position for the labels text (see par).
offset	When pos is specified, this value gives the offset of the labels.
col.arrows	Color for the arrows when plotting variables.
lwd	The line width of arrows.
length	Length of the edges of the arrow head (in inches).
angle	Angle from the shaft of the arrow to the edge of the arrow head.
col.points	Color for the points when what="observations".
pch	Plotting character symbol to use (see par).
pt.bg	Background (fill) color for the points given by pch=21:25.
show.names	Logical indicating whether to show names of points. Only used when what="observations"
xpd	Logical for controlling clipping region of labels and names.
xlab	Title for the x axis.
ylab	Title for the y axis.
main	Main title of the plot.
col.main	Color of main title.
cex.main	Character expansion of main title.
col.axis	Color of axis annotations (tick marks and labels).
show.grid	Logical indicating whether to show grid lines.
col.grid	Color of grid lines. Only used when show.grid=TRUE.
...	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations in $cor.xt.

Author(s)

Gaston Sanchez

See Also

nipals

Examples

```
## Not run:
 # load data climbing ropes
 data(ropes)

 # apply nipals with 3 components
 nip1 = nipals(ropes[,-1], comps=3)

 # plot variables (correlations)
 plot(nip1)

 # plot observations
 plot(nip1, what="obs")

 # plot observations with names
 plot(nip1, what="obs", show.names=TRUE)

## End(Not run)
```

`plot.plsca` *Plot PLS-CA basic results*

Description

Plot method for objects of class `"plsca"`. This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
 ## S3 method for class 'plsca'
plot(x, what = "variables",
   comps = c(1, 2), where = c("t", "t"), cex = 1,
   col.xlabels = "#5592e3", col.ylabels = "#fe9429",
   yfont = 2, pos = NULL, offset = 0.1,
   col.xarrows = "#5b9cf255", col.yarrows = "#fe942955",
   lwd = 3, length = 0, angle = 0, col.points = "#5592e3",
   pch = 21, pt.bg = "#5b9cf255", show.names = FALSE,
   xpd = TRUE, xlab = NULL, ylab = NULL, main = NULL,
   col.main = "gray35", cex.main = 1.2,
   col.axis = "gray40", show.grid = TRUE,
   col.grid = "gray95", ...)
```

Arguments

`x`	An object of class `"plsca"`.
`what`	What to plot. Options are `"variables"` and `"observations"`.
`comps`	An integer vector of length two to indicate which components to plot.

`where`	Where to plot the observations. A character vector of length two to indicate which components to use when plotting observations. This parameter will take into account the values in `comps`. Possible options are: `c("t","u")` for using x-y components, `c("t","t")`, for using x components, and `c("u","u")` for using y components. Default `c("t","t")`.
`cex`	Character expansion for labels and points.
`col.xlabels`	Color for labels of X-block variables.
`col.ylabels`	Color for labels of Y-block variables.
`yfont`	Integer for specifying which font to use for Y-block labels. See `font` in graphical parameters `par`.
`pos`	Position for the text (see graphical paramaters `par`).
`offset`	When `pos` is specified, this value gives the offset of the labels.
`col.xarrows`	Color for the X-block arrows.
`col.yarrows`	Color for the Y-block arrows.
`lwd`	The line width of arrows.
`length`	Length of the edges of the arrow head (in inches).
`angle`	Angle from the shaft of the arrow to the edge of the arrow head.
`col.points`	Color for the points when `what="observations"`.
`pch`	Plotting character symbol to use (see `par`).
`pt.bg`	Background (fill) color for the points given by `pch=21:25`.
`show.names`	Logical indicating whether to show labels of points. Only used when `what="observations"`.
`xpd`	Logical for controlling clipping region of names and labels.
`xlab`	A title for the x axis.
`ylab`	A title for the y axis.
`main`	Main title of the plot.
`col.main`	Color of main title.
`cex.main`	Character expansion of main title.
`col.axis`	Color of axis annotations (tick marks and labels).
`show.grid`	Logical indicating whether to show grid lines.
`col.grid`	Color of grid lines. Only used when `show.grid=TRUE`.
`...`	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations of each block of variables with its set of components: `$cor.xt` and `$cor.yu`.

Author(s)

Gaston Sanchez

See Also

plsca

Examples

```
## Not run:
 # load dataset linnerud
 data(linnerud)

 # apply plsca
 my_plsca = plsca(linnerud[,1:3], linnerud[,4:6])

 # plot variables (circle of correlations)
 plot(my_plsca, what="variables")

 # plot observations (as points) using components (t1,u1)
 plot(my_plsca, what="observations", comps=c(1,1), where=c("t","u"))

 # plot observations with names using components (t1,u1)
 plot(my_plsca, what="observations", comps=c(1,1),
      where=c("t","u"), show.names=TRUE)

 # plot observations (as points) using components (t1,t2)
 plot(my_plsca, what="observations", comps=c(1,2), where=c("t","t"))

 # plot observations (as points) using components (u1,u2)
 plot(my_plsca, what="observations", comps=c(1,2), where=c("u","u"))

## End(Not run)
```

---

plot.plsreg1 *Plot PLS-R1 basic results*

Description

Plot method for objects of class "plsreg1". This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
 ## S3 method for class 'plsreg1'
plot(x, what = "variables",
   comps = c(1, 2), where = c("t", "t"), cex = 1,
   col.xlabels = "#5592e3", col.ylabel = "#fe9429",
   yfont = 2, pos = NULL, offset = 0.1,
   col.xarrows = "#5b9cf255", col.yarrows = "#fe942955",
   lwd = 3, length = 0, angle = 0, col.points = "#5592e3",
   pch = 21, pt.bg = "#5b9cf255", show.names = FALSE,
```

```
xpd = TRUE, xlab = NULL, ylab = NULL, main = NULL,
col.main = "gray35", cex.main = 1.2,
col.axis = "gray40", show.grid = TRUE,
col.grid = "gray95", ...)
```

Arguments

x	An object of class "plsreg1".
what	What to plot. Options are "variables" and "observations".
comps	An integer vector of length two to indicate which components to plot.
where	Where to plot the observations. A character vector of length two to indicate which components to use when plotting observations. This parameter will take into account the values in comps. Possible options are: c("t","u") for using x-y components, c("t","t"), for using x components, and c("u","u") for using y components. Default c("t","t").
cex	Character expansion for labels and points.
col.xlabels	Color for labels of X-block variables.
col.ylabel	Color for labels of Y-block variables.
yfont	Integer for specifying which font to use for Y-block labels. See font in graphical parameters par.
pos	Position for the text (see graphical paramaters par).
offset	When pos is specified, this value gives the offset of the labels.
col.xarrows	Color for the X-block arrows.
col.yarrows	Color for the Y-block arrows.
lwd	The line width of arrows.
length	Length of the edges of the arrow head (in inches).
angle	Angle from the shaft of the arrow to the edge of the arrow head.
col.points	Color for the points when what="observations".
pch	Plotting character symbol to use (see par).
pt.bg	Background (fill) color for the points given by pch=21:25.
show.names	Logical indicating whether to show labels of points. Only used when what="observations".
xpd	Logical for controlling clipping region of names and labels.
xlab	A title for the x axis.
ylab	A title for the y axis.
main	Main title of the plot.
col.main	Color of main title.
cex.main	Character expansion of main title.
col.axis	Color of axis annotations (tick marks and labels).
show.grid	Logical indicating whether to show grid lines.
col.grid	Color of grid lines. Only used when show.grid=TRUE.
...	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations in `$cor.xyt`.

Author(s)

Gaston Sanchez

See Also

`plsreg1`

Examples

```
## Not run:
 # load dataset cornell
 data(cornell)

 # apply plsreg1
 myplsr1 = plsreg1(cornell[,1:7], cornell[,8,drop=FALSE])

 # plot variables (circle of correlations)
 plot(myplsr1, what="variables")

 # plot observations (as points) using components (t1,u1)
 plot(myplsr1, what="observations", comps=c(1,1), where=c("t","u"))

 # plot observations with names using components (t1,u1)
 plot(myplsr1, what="observations", comps=c(1,1),
      where=c("t","u"), show.names=TRUE)

 # plot observations (as points) using components (t1,t2)
 plot(myplsr1, what="observations", comps=c(1,2), where=c("t","t"))

 # plot observations (as points) using components (u1,u2)
 plot(myplsr1, what="observations", comps=c(1,2), where=c("u","u"))

## End(Not run)
```

---

plot.plsreg2 *Plot PLS-R2 basic results*

Description

Plot method for objects of class `"plsreg2"`. This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
 ## S3 method for class 'plsreg2'
plot(x, what = "variables",
   comps = c(1, 2), where = c("t", "t"), cex = 1,
   col.xlabels = "#5592e3", col.ylabels = "#fe9429",
   yfont = 2, pos = NULL, offset = 0.1,
   col.xarrows = "#5b9cf255", col.yarrows = "#fe942955",
   lwd = 3, length = 0, angle = 0, col.points = "#5592e3",
   pch = 21, pt.bg = "#5b9cf255", show.names = FALSE,
   xpd = TRUE, xlab = NULL, ylab = NULL, main = NULL,
   col.main = "gray35", cex.main = 1.2,
   col.axis = "gray40", show.grid = TRUE,
   col.grid = "gray95", ...)
```

Arguments

x	An object of class "plsreg2".
what	What to plot. Options are "variables" and "observations".
comps	An integer vector of length two to indicate which components to plot.
where	Where to plot the observations. A character vector of length two to indicate which components to use when plotting observations. This parameter will take into account the values in comps. Possible options are: c("t","u") for using x-y components, c("t","t"), for using x components, and c("u","u") for using y components. Default c("t","t").
cex	Character expansion for labels and points.
col.xlabels	Color for labels of X-block variables.
col.ylabels	Color for labels of Y-block variables.
yfont	Integer for specifying which font to use for Y-block labels. See font in graphical parameters par.
pos	Position for the text (see graphical paramaters par).
offset	When pos is specified, this value gives the offset of the labels.
col.xarrows	Color for the X-block arrows.
col.yarrows	Color for the Y-block arrows.
lwd	The line width of arrows.
length	Length of the edges of the arrow head (in inches).
angle	Angle from the shaft of the arrow to the edge of the arrow head.
col.points	Color for the points when what="observations".
pch	Plotting character symbol to use (see par).
pt.bg	Background (fill) color for the points given by pch=21:25.
show.names	Logical indicating whether to show labels of points. Only used when what="observations".
xpd	Logical for controlling clipping region of names and labels.
xlab	A title for the x axis.

ylab	A title for the y axis.
main	Main title of the plot.
col.main	Color of main title.
cex.main	Character expansion of main title.
col.axis	Color of axis annotations (tick marks and labels).
show.grid	Logical indicating whether to show grid lines.
col.grid	Color of grid lines. Only used when show.grid=TRUE.
...	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations of each block of variables with its set of components: $cor.xt and $cor.yt.

Author(s)

Gaston Sanchez

See Also

plsreg2

Examples

```
## Not run:
 # load dataset vehicles
 data(vehicles)

 # apply plsreg2
 pls2 = plsreg2(vehicles[,1:12], vehicles[,13:16])

 # plot variables (circle of correlations)
 plot(pls2, what="variables")

 # plot observations (as points)
 plot(pls2, what="observations")

 # plot observations with labels
 plot(pls2, what="observations", show.names=TRUE)

## End(Not run)
```

`plot.simpls` *Plot simpls basic results*

Description

Plot method for objects of class `"simpls"`. This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
  ## S3 method for class 'simpls'
plot(x, what = "variables",
   comps = c(1, 2), cex = 1, col.xlabels = "#5592e3",
   col.ylabels = "#fe9429", yfont = 2, pos = NULL,
   offset = 0.1, col.xarrows = "#5b9cf255",
   col.yarrows = "#FE992955", lwd = 3, length = 0,
   angle = 0, col.points = "#5592e3", pch = 21,
   pt.bg = "#5b9cf255", show.names = FALSE, xpd = TRUE,
   xlab = NULL, ylab = NULL, main = NULL,
   col.main = "gray35", cex.main = 1.2,
   col.axis = "gray40", show.grid = TRUE,
   col.grid = "gray95", ...)
```

Arguments

`x`	An object of class `"simpls"`.
`what`	What to plot. Options are `"variables"` and `"observations"`
`comps`	An integer vector of length two to indicate which components to plot
`cex`	Character expansion for labels and points.
`col.xlabels`	Color for labels of X-block variables.
`col.ylabels`	Color for labels of Y-block variables.
`yfont`	Integer for specifying which font to use for Y-block labels. See `font` in graphical parameters `par`.
`pos`	Position for the text (see graphical paramaters `par`).
`offset`	When `pos` is specified, this value gives the offset of the label.
`col.xarrows`	Color for the X-block arrows.
`col.yarrows`	Color for the Y-block arrows.
`lwd`	The line width of arrows.
`length`	Length of the edges of the arrow head (in inches).
`angle`	Angle from the shaft of the arrow to the edge of the arrow head.
`col.points`	Color for the points when `what="observations"`.

pch	Plotting character symbol to use (see par).
pt.bg	Background (fill) color for the points given by pch=21:25.
show.names	Logical indicating whether to show labels of points. Only used when what="observations"
xpd	Logical for controlling clipping region of names and labels.
xlab	A title for the x axis.
ylab	A title for the y axis.
main	Main title of the plot.
col.main	Color of main title.
cex.main	Character expansion of main title.
col.axis	Color of axis annotations (tick marks and labels).
show.grid	Logical indicating whether to show grid lines.
col.grid	Color of grid lines. Only used when show.grid=TRUE.
...	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations of each block of variables with the X-components: $cor.xt and $cor.yt.

Author(s)

Gaston Sanchez

See Also

simpls

Examples

```
## Not run:
 # load dataset linnerud
 data(linnerud)

 # apply simpls
 sim = simpls(linnerud[,1:3], linnerud[,4:6])

 # plot variables (circle of correlations)
 plot(sim, what="variables")

 # plot observations (as points)
 plot(sim, what="observations")

 # plot observations with names
 plot(sim, what="observations", show.names=TRUE)

## End(Not run)
```

`plot.simplsca` *Plot SIMPLS-CA basic results*

Description

Plot method for objects of class `"simplsca"`. This function plots either the variables or the observations, on the selected components (i.e. scores). Variables are plotted inside the circle of correlations. Observations are plotted on a scatter plot.

Usage

```
  ## S3 method for class 'simplsca'
plot(x, what = "variables",
   comps = c(1, 2), where = c("t", "t"), cex = 1,
   col.xlabels = "#5592e3", col.ylabels = "#fe9429",
   yfont = 2, pos = NULL, offset = 0.1,
   col.xarrows = "#5b9cf255", col.yarrows = "#fe942955",
   lwd = 3, length = 0, angle = 0, col.points = "#5592e3",
   pch = 21, pt.bg = "#5b9cf255", show.names = FALSE,
   xpd = TRUE, xlab = NULL, ylab = NULL, main = NULL,
   col.main = "gray35", cex.main = 1.2,
   col.axis = "gray40", show.grid = TRUE,
   col.grid = "gray95", ...)
```

Arguments

`x`	An object of class `"simplsca"`.
`what`	What to plot. Options are `"variables"` and `"observations"`.
`comps`	An integer vector of length two to indicate which components to plot.
`where`	Where to plot the observations. A character vector of length two to indicate which components to use when plotting observations. This parameter will take into account the values in `comps`. Possible options are: `c("t","u")` for using x-y components, `c("t","t")`, for using x components, and `c("u","u")` for using y components. Default `c("t","t")`.
`cex`	Character expansion for labels and points.
`col.xlabels`	Color for labels of X-block variables.
`col.ylabels`	Color for labels of Y-block variables.
`yfont`	Integer for specifying which font to use for Y-block labels. See `font` in graphical parameters `par`.
`pos`	Position for the text (see graphical paramaters `par`)
`offset`	When `pos` is specified, this value gives the offset of the label
`col.xarrows`	Color for the X-block arrows.
`col.yarrows`	Color for the Y-block arrows.

lwd	The line width of arrows.
length	Length of the edges of the arrow head (in inches).
angle	Angle from the shaft of the arrow to the edge of the arrow head.
col.points	Color for the points when what="observations".
pch	Plotting character symbol to use (see par).
pt.bg	Background (fill) color for the points given by pch=21:25.
show.names	Logical indicating whether to show labels of points. Only used when what="observations".
xpd	Logical for controlling clipping region of names and labels.
xlab	A title for the x axis.
ylab	A title for the y axis.
main	Main title of the plot.
col.main	Color of main title.
cex.main	Character expansion of main title.
col.axis	Color of axis annotations (tick marks and labels).
show.grid	Logical indicating whether to show grid lines.
col.grid	Color of grid lines. Only used when show.grid=TRUE.
...	Further arguments are passed to labels or points.

Details

Variables are displayed using the correlations of each block of variables with its set of components: $cor.xt and $cor.yu.

Author(s)

Gaston Sanchez

See Also

simplsca

Examples

```
## Not run:
 # load dataset linnerud
 data(linnerud)

 # apply simplsca
 simca = simplsca(linnerud[,1:3], linnerud[,4:6])

 # plot variables (circle of correlations)
 plot(simca, what="variables")

 # plot observations (as points) using components (t1,u1)
 plot(simca, what="observations", comps=c(1,1), where=c("t","u"))
```

```
# plot observations with names using components (t1,u1)
plot(simca, what="observations", comps=c(1,1),
     where=c("t","u"), show.names=TRUE)

# plot observations (as points) using components (t1,t2)
plot(simca, what="observations", comps=c(1,2), where=c("t","t"))

# plot observations (as points) using components (u1,u2)
plot(simca, what="observations", comps=c(1,2), where=c("u","u"))

## End(Not run)
```

`plsca` *PLS-CA: Partial Least Squares Canonical Analysis*

Description

Performs partial least squares canonical analysis for two blocks of data. Compared to PLSR2, the blocks of variables in PLS-CA play a symmetric role (i.e. there is neither predictors nor responses)

Usage

```
plsca(X, Y, comps = NULL, scaled = TRUE)
```

Arguments

`X`	A numeric matrix or data frame (X-block) with more than one variable. No missing data are allowed
`Y`	A numeric matrix or data frame (Y-block) with more than one variable. No missing data are allowed
`comps`	The number of extracted PLS components (`NULL` by default) When `comps=NULL` the number of components is determined by taking the minimum between the number of columns from `X` and `Y`.
`scaled`	A logical value indicating whether scaling data should be performed (`TRUE` by default). #'When `scaled=TRUE` the data is scaled to standardized values (mean=0, variance=1). Otherwise the data will only be centered (mean=0).

Value

An object of class `"plsca"`, basically a list with the following elements:

`x.scores`	scores of the X-block (also known as T components)
`x.wgs`	weights of the X-block
`x.loads`	loadings of the X-block
`y.scores`	scores of the Y-block (also known as U components)

y.wgs	weights of the Y-block
y.loads	loadings of the Y-block
cor.xt	correlations between X and T
cor.yu	correlations between Y and U
cor.tu	correlations between T and U
cor.xu	correlations between X and U
cor.yt	correlations between Y and T
R2X	explained variance of X by T
R2Y	explained variance of Y by U
com.xu	communality of X with U
com.yt	communality of Y with T

Author(s)

Gaston Sanchez

References

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique*. Editions TECHNIP, Paris.

See Also

plot.plsca

Examples

```
## Not run:
 ## example of PLSCA with the vehicles dataset
 data(vehicles)

 # apply plsca
 my_plsca = plsca(vehicles[,1:12], vehicles[,13:16])
 my_plsca

 # plot variables
 plot(my_plsca)

## End(Not run)
```

plsreg1 *PLS-R1: Partial Least Squares Regression 1*

Description

The function plsreg1 performs Partial Least Squares Regression for the univariate case (i.e. one response variable)

Usage

```
plsreg1(predictors, response, comps = 2, crosval = TRUE)
```

Arguments

predictors	A numeric matrix or data frame with the predictor variables (which may contain missing data).
response	A numeric vector for the reponse variable. No missing data allowed.
comps	The number of extracted PLS components (2 by default).
crosval	Logical indicating whether cross-validation should be performed (TRUE by default). No cross-validation is done if there is missing data or if there are less than 10 observations.

Details

The minimum number of PLS components (comps) to be extracted is 2.

The data is scaled to standardized values (mean=0, variance=1).

The argument crosval gives the option to perform cross-validation. This parameter takes into account how comps is specified. When comps=NULL, the number of components is obtained by cross-validation. When a number of components is specified, cross-validation results are calculated for each component.

Value

An object of class "plsreg1", basically a list with the following elements:

x.scores	PLS components (also known as T-components)
x.loads	loadings of the predictor variables
y.scores	scores of the response variable (also known as U-components)
y.loads	loadings of the response variable
cor.xyt	Correlations between the variables and the PLS components
raw.wgs	weights to calculate the PLS scores with the deflated matrices of predictor variables
mod.wgs	modified weights to calculate the PLS scores with the matrix of predictor variables

std.coefs	Vector of standardized regression coefficients
reg.coefs	Vector of regression coefficients (used with the original data scale)
R2	Vector of PLS R-squared
R2Xy	explained variance of variables by PLS-components
y.pred	Vector of predicted values
resid	Vector of residuals
T2	Table of Hotelling T2 values (used to detect atypical observations)
Q2	Table with the cross validation results. Includes: PRESS, RSS, Q2, and cummulated Q2. Only available when crosval=TRUE

Author(s)

Gaston Sanchez

References

Geladi, P., and Kowalski, B. (1986) Partial Least Squares Regression: A Tutorial. *Analytica Chimica Acta*, **185**, pp. 1-17.

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique*. Editions TECHNIP, Paris.

Tenenhaus, M., Gauchi, J.-P., and Menardo, C. (1995) Regression PLS et applications. *Revue de statistique appliquee*, **43**, pp. 7-63.

See Also

plot.plsreg1, plsreg2.

Examples

```
## Not run:
 ## example of PLSR1 with the vehicles dataset
 # predictand variable: price of vehicles
 data(vehicles)

 # apply plsreg1 extracting 2 components (no cross-validation)
 pls1_one = plsreg1(vehicles[,1:12], vehicles[,13,drop=FALSE], comps=2, crosval=FALSE)

 # apply plsreg1 with selection of components by cross-validation
 pls1_two = plsreg1(vehicles[,1:12], vehicles[,13,drop=FALSE], comps=NULL, crosval=TRUE)

 # apply plsreg1 extracting 5 components with cross-validation
 pls1_three = plsreg1(vehicles[,1:12], vehicles[,13,drop=FALSE], comps=5, crosval=TRUE)

 # plot variables
 plot(pls1_one)

## End(Not run)
```

plsreg2 *PLS-R2: Partial Least Squares Regression 2*

Description

The function plsreg2 performs partial least squares regression for the multivariate case (i.e. more than one response variable)

Usage

```
plsreg2(predictors, responses, comps = 2, crosval = TRUE)
```

Arguments

predictors	A numeric matrix or data frame containing the predictor variables.
responses	A numeric matrix or data frame containing the response variables.
comps	The number of extracted PLS components (2 by default)
crosval	Logical indicating whether cross-validation should be performed (TRUE by default). No cross-validation is done if there is missing data or if there are less than 10 observations.

Details

The minimum number of PLS components comps to be extracted is 2.

The data is scaled to standardized values (mean=0, variance=1).

The argument crosval gives the option to perform cross-validation. This parameter takes into account how comps is specified. When comps=NULL, the number of components is obtained by cross-validation. When a number of components is specified, cross-validation results are calculated for each component.

Value

An object of class "plsreg2", basically a list with the following elements:

x.scores	components of the predictor variables (also known as T-components)
x.loads	loadings of the predictor variables
y.scores	components of the response variables (also known as U-components)
y.loads	loadings of the response variables
cor.xt	correlations between X and T
cor.yt	correlations between Y and T
cor.xu	correlations between X and U
cor.yu	correlations between Y and U
cor.tu	correlations between T and U

raw.wgs	weights to calculate the PLS scores with the deflated matrices of predictor variables
mod.wgs	modified weights to calculate the PLS scores with the matrix of predictor variables
std.coefs	Vector of standardized regression coefficients (used with scaled data)
reg.coefs	Vector of regression coefficients (used with the original data)
y.pred	Vector of predicted values
resid	Vector of residuals
expvar	table with R-squared coefficients
VIP	Variable Importance for Projection
Q2	table of Q2 indexes (i.e. leave-one-out cross validation)
Q2cum	table of cummulated Q2 indexes

Author(s)

Gaston Sanchez

References

Geladi, P., and Kowlaski, B. (1986) Partial Least Squares Regression: A Tutorial. *Analytica Chimica Acta*, **185**, pp. 1-17.

Hoskuldsson, A. (1988) PLS Regression Methods. *Journal of Chemometrics*, **2**, pp. 211-228.

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique*. Editions TECHNIP, Paris.

See Also

plot.plsreg2, plsreg1.

Examples

```
## Not run:
 ## example of PLSR2 with the vehicles dataset
 data(vehicles)

 # apply plsreg2 extracting 2 components (no cross-validation)
 pls2_one = plsreg2(vehicles[,1:12], vehicles[,13:16], comps=2, crosval=FALSE)

 # apply plsreg2 with selection of components by cross-validation
 pls2_two = plsreg2(vehicles[,1:12], vehicles[,13:16], comps=NULL, crosval=TRUE)

 # apply plsreg2 extracting 5 components with cross-validation
 pls2_three = plsreg2(vehicles[,1:12], vehicles[,13:16], comps=5, crosval=TRUE)

 # plot variables
 plot(pls2_one)

## End(Not run)
```

ropes *Climbing Ropes data set*

Description

This dataset gives the measurements of 101 climbing ropes available in the market by spring 2011. The data was collected from the brands websites.

Usage

```
data(ropes)
```

Format

A data frame with 101 observations on the following 7 variables.

Num	*Variable*	*Description*
1	brand	a factor with the brand names
2	diameter	the diameter of the ropes measured in mm
3	weight	the weight measured in grams per meter
4	falls	the number of UIAA falls
5	imp.force	the impact force value
6	stat.elong	the static elongation value
7	dyn.elong	the dynamic elongation value

Source

Personal collection of data. (Gaston Sanchez)

Examples

```
data(ropes)
summary(ropes)
```

simpls *SIMPLS: Alternative Approach to PLS Regression*

Description

The function `simpls` performs the SIMPLS Algorithm as described in Michel Tenenhaus book *La Regression PLS*, chapter 5.

Usage

```
simpls(X, Y, comps = 2)
```

Arguments

X	Numeric matrix or data frame with two or more columns (X-block).
Y	Numeric matrix or data frame with two or more columns (Y-block).
comps	Number of components to be extracted. (TRUE by default).

Details

No missing data are allowed.

Value

An object of class "simpls", basically a list with the following elements:

x.scores	scores of the X-block (also known as T components)
x.wgs	weights of the X-block
y.wgs	weights of the Y-block
cor.xt	correlations between X and T
cor.yt	correlations between Y and T
R2X	explained variance of X by T
R2Y	explained variance of Y by T

Author(s)

Gaston Sanchez

References

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique.* Paris: Editions TECHNIP.

de Jong, S. (1993) SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18: 251-263.

See Also

plot.simpls, simplsca

Examples

```
## Not run:
 # load data linnerud
 data(linnerud)

 # apply inter-battery method
 my_simpls = simpls(linnerud[,1:3], linnerud[,4:6])

 # plot variables
 plot(my_simpls, what="variables")

## End(Not run)
```

simplsca *SIMPLS-CA: SIMPLS Canonical Analysis*

Description

The function simplsca performs the SIMPLS Canonical Analysis algorithm as described in Michel Tenenhaus book *La Regression PLS*, chapter 5.

Usage

```
simplsca(X, Y, comps = 2)
```

Arguments

X	Numeric matrix or data frame with two or more columns (X-block).
Y	Numeric matrix or data frame with two or more columns (Y-block).
comps	Number of components to be extracted. (TRUE by default).

Details

No missing data are allowed.

Value

An object of class "simplsca", basically a list with the following elements:

x.scores	scores of the X-block (also known as T components)
x.wgs	weights of the X-block
y.scores	scores of the Y-block (also known as U components)
y.wgs	weights of the Y-block
cor.xt	correlations between X and T
cor.yu	correlations between Y and U
cor.xu	correlations between X and U
cor.yt	correlations between Y and T
cor.tu	correlations between T and U
R2XT	explained variance of X by T
R2YT	explained variance of Y by T
R2YU	explained variance of Y by U
R2XU	explained variance of X by U

Author(s)

Gaston Sanchez

References

Tenenhaus, M. (1998) *La Regression PLS. Theorie et Pratique.* Paris: Editions TECHNIP.

See Also

plot.simplsca, simpls

Examples

```
## Not run:
 # load data linnerud
 data(linnerud)

 # apply inter-battery method
 my_simca = simplsca(linnerud[,1:3], linnerud[,4:6])

 # plot variables
 plot(my_simca, what="variables")

## End(Not run)
```

---

vehicles *Vehicles data set*

Description

These data are the specification of 30 vehicles in terms of various characteristics.

Format

A data frame with 30 observations and 16 variables.

Num	*Variable*	*Description*
1	diesel	Diesel fuel-type
2	turbo	Turbo aspiration
3	two.doors	Vechicles with two doors
4	hatchback	Hatchback body-style
5	wheel.base	Wheel base
6	length	Length
7	width	Width
8	height	Height
9	curb.weight	Curb weight
10	eng.size	Engine size
11	horsepower	Horsepower
12	peak.rpm	Peak revolutions per minute
13	price	Price in dollars
14	symbol	Insurance risk rating
15	city.mpg	Fuel consume in city

16	highway.mpg	Fuel consume in highway

Source

1) 1985 Model Import Car and Truck Specifications, 1985 Ward's Automotive Yearbook.
2) Personal Auto Manuals, Insurance Services Office, 160 Water Street, New York, NY 10038.
3) Insurance Collision Report, Insurance Institute for Highway Safety, Watergate 600, Washington, DC 20037.

Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Automobile

Examples

```
data(vehicles)
vehicles
```

Index