

Package ‘bpca’

January 2, 2012

Type Package

Title Biplot of Multivariate Data Based on Principal Components Analysis

Version 1.0-8

Date 2011-04-21

Author Jose Claudio Faria & Clarice Garcia Borges Demetrio

Maintainer Jose Claudio Faria <joseclaudio.faria@gmail.com>

Depends R (>= 2.6.0), scatterplot3d, rgl

Description Implements biplot (2d and 3d) of multivariate data based on principal components analysis and diagnostic tools of the quality of the reduction.

License GPL (>= 2)

LazyData yes

Repository CRAN

Date/Publication 2011-04-24 06:06:59

R topics documented:

bpca-package	2
bpca	3
dt.tools	8
gabriel1971	10
gge2003	11
plot	12
qbpc	17
summary	19
var.rbf	20
var.rdf	21
Index	24

bpca-package

Biplot of Multivariate Data Based on Principal Components Analysis

Description

Implements biplot (2d and 3d) and diagnostic tools of the quality of the reduction.

Details

Package: bpca
Type: Package
Version: 1.0-8
Date: 2011-04-21
License: GPL (>= 2)

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

References

- Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453-467.
- Gower, J.C. and Hand, D. J. (1996) *Biplots*. Chapman & Hall.
- Galindo, M. P. (1986) Una alternativa de representacion simultanea: HJ-Biplot. *Questio*, 10(1):13-23, 1986.
- Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.
- Yan, B. W. and Kang, M. S. (2003) *GGE biplot analysis : a graphical tool for breeders, geneticists, and agronomists*. CRC Press, New York, 288p.

Examples

```
##  
## Grouping objects with different symbols and colors - 2d and 3d  
##  
  
x11(w=6, h=6)  
oask <- devAskNewPage(dev.interactive(orNone = TRUE))  
  
# 2d
```

```

plot(bpca(iris[-5]),
     var.factor=.3, var.cex=.7,
     obj.names=FALSE, obj.cex=1.5,
     obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)],
     obj.pch=c('+', '*', '-')[unclass(iris$Species)])

# 3d static
plot(bpca(iris[-5], lambda.end=3),
     var.factor=.2, var.color=c('blue', 'red'), var.cex=1,
     obj.names=FALSE, obj.cex=1,
     obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)],
     obj.pch=c('+', '*', '-')[unclass(iris$Species)])

# 3d dinamic
plot(bpca(iris[-5], method='hj', lambda.end=3), rgl.use=TRUE,
     var.col='brown', var.factor=.3, var.cex=1.2,
     obj.names=FALSE, obj.cex=.8,
     obj.col=c('red', 'green3', 'orange')[unclass(iris$Species)],
     simple.axes=FALSE, box=TRUE)

devAskNewPage(oask)

```

bpca

*Biplot of Multivariate Data Based on Principal Components Analysis***Description**

Computes biplot reduction on data.frame, matrix or prcomp objects and returns a bpca object.

Usage

```

bpca(x, ...)
## Default S3 method:
bpca(x, lambda.ini=1, lambda.end=2, var.position=2,
     var.center=TRUE, var.scale=TRUE, method=c('hj', 'sqrt', 'jk', 'gh'),
     var.rb=FALSE, var.rd=FALSE, limit=10, ...)
## S3 method for class 'prcomp'
bpca(x, lambda.ini=1, lambda.end=2, ...)

```

Arguments

x	A data.frame, matrix or prcomp object.
lambda.ini	A vector of length 1 giving the first eigenvalue to be considered by the biplot reduction.
lambda.end	A vector of length 1 giving the last eigenvalue to be considered by the biplot reduction.
var.position	A vector of length 1 giving the position of the elements to be considered as variables on the given object. They will be presented as vectors in the graphical representation. 1 indicates <i>rows</i> and 2 indicates <i>columns</i> .

<code>var.center</code>	A logical value indicating whether the variables should be shifted to be zero centered. Alternatively, a vector of length equal to the number of columns of <code>x</code> can be supplied. The value is passed to <code>scale</code> .
<code>var.scale</code>	A logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place. Alternatively, a vector of length equal to the number of columns of <code>x</code> can be supplied. The value is passed to <code>scale</code> .
<code>method</code>	A vector of character strings that indicates the method of factorization. 'hj' to HJ (<i>simetric</i> , Galindo (1986)), 'sqrt' to SQR (<i>squared root - simetric</i> , Gabriel (1971)), 'jk' to JK (<i>row metric preserving</i> , Gabriel (1971)) and 'gh' to GH (<i>column metric preserving</i> , Gabriel (1971)).
<code>var.rb</code>	A logical value. If TRUE, all correlation coefficients for all variables (<i>under the biplot projection</i>) will be computed.
<code>var.rd</code>	A logical value. If TRUE, the diagnostic of the representation of variables projected by the biplot will be computed. If <code>var.rd</code> is TRUE the <code>var.rb</code> parameter must be also TRUE.
<code>limit</code>	A vector giving the percentual limit to define poor representation of variables.
<code>...</code>	Additional parameters. It is necessary to be S3 method.

Details

The biplot is a multivariate method for graphing row and column elements using a single plot (Gabriel, 1971).

The biplot of a matrix

$${}_n Y_p$$

projects on the same plot: *rows* (associated with n objects) and *columns* (associated with p variables), markers from data that forms a two-way table (`data.frame` or `matrix` object). The markers are computed from the singular value decomposition, $\text{svd}(Y)$, and subsequent factorization.

The *bi* refers to the kind of information contained in a data set disposed in a two-way table. If the data are a *tri*-dimensional array the method will be called *triplet* (not still contemplated in the `bpca` package).

The basic idea behind the biplot method was to add the information about the variables to the principal component graph (Johnson & Wichern, 1988).

Considering the results of

$$\text{svd}({}_n Y_p)$$

- d : A vector containing the singular values of Y , of length $\min(n, p)$;
- u : A matrix whose columns contain the left singular vectors of Y , present if $n_u > 0$. Dimension $c(n, n_u)$;
- v : A matrix whose columns contain the right singular vectors of Y , present if $n_v > 0$. Dimension $c(p, n_v)$.

and also,

$$s^2 = \text{diag}(d)$$

$$n = n_{\text{objects}}(Y)$$

it is possible an approximation of Y:

$${}_n Y_p \approx Y_m = g \cdot h'$$

in various ways. The methods of factorization computed by the bpca function are:

- *HJ - simetric*, Galindo (1986):

$$g = u * s^2$$

$$h = s^2 * v'$$

- *SQRT - squared root simetric*, Gabriel (1971):

$$g = u * \sqrt{s^2}$$

$$h = \sqrt{s^2} * v'$$

- *JK - row metric preserving*, Gabriel (1971):

$$g = u * s^2$$

$$h = v'$$

- *GH - column metric preserving*, Gabriel (1971):

$$g = \sqrt{n-1} * u$$

$$h = \frac{1}{\sqrt{n-1}} * s^2 * v'$$

Considering

$${}_n Y_p \approx Y_m$$

it is possible to deduce that if the rank (r) of the matrix

$${}_n Y_p$$

is bigger than 'm', the biplot representation of Y will be an approximation, and accurate only in the case of \$r=m\$.

Due to the need of different methods of factorization, if x is a prcomp object, the method bpca.prcomp will go back from the prcom function. In other words, it will regenerates, or computes, the inverse of the svd decomposition of the given data

$${}_n Y_p$$

After this, it will make a call to the method bpca.default with the adequated parameters.

The biplot is used with many multivariate methods to display relationships between objects, variables and the interrelationship between objects and variables (as prevalence, importance). There are many variations of biplots (see the references).

Value

The function `bpca` returns an object of class `bpca.2d` or `bpca.3d`. Both are list objects with the slots:

<code>call</code>	The call used.
<code>eigenvalues</code>	A vector of the eigenvalues.
<code>eigenvectors</code>	A vector of the eigenvectors.
<code>numer</code>	A vector of the number of eigenvalues considered in the reduction.
<code>importance</code>	A matrix with the <i>general</i> and <i>partial</i> variation explained by the reduction.
<code>coord</code>	A list with the coordinates of the two components: objects and variables.
<code>var.rb</code>	A matrix of all correlation coefficients for all variables under the biplot projection.
<code>var.rd</code>	A matrix of the diagnostic of the poor projection of variable correlations by the biplot reduction.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

References

- Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453-467.
- Gower, J.C. and Hand, D. J. (1996) *Biplots*. Chapman & Hall.
- Galindo, M. P. (1986) Una alternativa de representacion simultanea: HJ-Biplot. *Questio*, 10(1):13-23, 1986.
- Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.
- Yan, B. W. and Kang, M. S. (2003) *GGE biplot analysis : a graphical tool for breeders, geneticists, and agronomists*. CRC Press, New York, 288p.

See Also

[biplot.prcomp](#)

Examples

```
##
## Example 1
## Computing and plotting a bpca object with 'graphics' package - 2d
##

bp <- bpca(gabriel1971)
```

```
x11(w=6, h=6)
oask <- devAskNewPage(dev.interactive(orNone = TRUE))
plot(bp, var.factor=2)

# Exploring the object 'bp' created by the function 'bpca'
class(bp)
names(bp)
str(bp)

summary(bp)
bp$call
bp$eigenval
bp$eigenvec
bp$numb
bp$import
bp$coord
bp$coord$obj
bp$coord$var
bp$var.rb
bp$var.rd

##
## Example 2
## Computing and plotting a bpca object with 'scatterplot3d' package - 3d
##

bp <- bpca(gabriel1971, lambda.end=3)

plot(bp, var.factor=3)

# Exploring the object 'bp' created by the function 'bpca'
class(bp)
names(bp)
str(bp)

summary(bp)
bp$call
bp$eigenval
bp$eigenvec
bp$numb
bp$import
bp$coord
bp$coord$obj
bp$coord$var
bp$var.rb
bp$var.rd

##
## Example 3
## Computing and plotting a bpca object with 'rgl' package - 3d
##

plot(bpca(gabriel1971, lambda.end=3),
```

```

rgl.use=TRUE, var.factor=2)

# Suggestion: Interact with the graphic with the mouse
# left button: press, maintain and movement it to interactive rotation;
# right button: press, maintain and movement it to interactive zoom.
# Enjoy it!

##
## Example 4
## Grouping objects with different symbols and colors - 2d and 3d
##

# 2d
plot(bpca(iris[-5]),
var.factor=.3, var.cex=.7,
obj.names=FALSE, obj.cex=1.5,
obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)],
obj.pch=c('+', '*', '-')[unclass(iris$Species)])

# 3d static
plot(bpca(iris[-5], lambda.end=3),
var.factor=.2, var.color=c('blue', 'red'), var.cex=1,
obj.names=FALSE, obj.cex=1,
obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)],
obj.pch=c('+', '*', '-')[unclass(iris$Species)])

# 3d dinamic
plot(bpca(iris[-5], method='hj', lambda.end=3), rgl.use=TRUE,
var.col='brown', var.factor=.3, var.cex=1.2,
obj.names=FALSE, obj.cex=.8,
obj.col=c('red', 'green3', 'orange')[unclass(iris$Species)],
simple.axes=FALSE, box=TRUE)

devAskNewPage(oask)

```

dt.tools

Data Tools for Multivariate

Description

Calculates vector variable lengths, angles between vector variables and variable correlations from 'data.frame' or 'matrix' objects

Usage

```
dt.tools(x, var.position=2, var.center=TRUE, var.scale=FALSE)
```

Arguments

<code>x</code>	A <code>data.frame</code> or <code>matrix</code> object.
<code>var.position</code>	A vector of length 1 giving the position of the elements to be considered as variables on the given object. They will be represented as vectors in the graphical representation. 1 indicates <i>rows</i> and 2 indicates <i>columns</i> .
<code>var.center</code>	A logical value indicating whether the variables should be shifted to be zero centered. Alternatively, a vector of length equal to the number of columns of <code>x</code> can be supplied. The value is passed to <code>scale</code> .
<code>var.scale</code>	A logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place. Alternatively, a vector of length equal to the number of columns of <code>x</code> can be supplied. The value is passed to <code>scale</code> .

Details

This function computes: vector variable lengths, angles between vector variables and variable correlations from `data.frame` or `matrix` objects.

If the data are centered (`var.center=TRUE`), the correlations are the same as those obtained by the `cor` function.

Value

An *list* with the components:

<code>length</code>	A vector of the lengths.
<code>angle</code>	A matrix of the angles.
<code>r</code>	A matrix of the observed correlations.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

References

Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.

See Also

[bpca](#)

Examples

```
##
## Computes: vector variable lengths, angles between vector variables and
## variable correlations from dataframe or matrix objects (n x p)
## n = rows (objects)
## p = columns (variables)
##

dt <- dt.tools(iris[-5], 2)

# Exploring the object 'bp' created by the function 'var.tools'
class(dt)
names(dt)
str(dt)

dt$length
dt$angle
dt$r
dt

# Checking the determinations
iris.tools <- round(dt.tools(iris[-5], var.center=TRUE)$r, 5); iris.tools
iris.obsv <- round(cor(iris[-5]), 5); iris.obsv
all(iris.tools == iris.obsv)
```

 gabriel1971

Percentages of households having various facilities and appliances in East Jerusalem Arab areas, by quarters of the town

Description

Percentages of households having various facilities and appliances in East Jerusalem Arab areas, by quarters of the town. The average percentages in each quarter indicate the standard of living of that area and the average percentage of each facility or appliance its over-all prevalence.

Usage

```
gabriel1971
```

Format

The format is: num [1:8, 1:9] 98.2 78.8 14.4 86.2 32.9 73 4.6 29.2 97.2 81 ... - attr(*, "dim-names")=List of 2 ..\$: chr [1:8] "toilet" "kitchen" "bath" "eletricity"\$: chr [1:9] "CRISTIAN" "ARMENIAN" "JEWISH" "MOSLEM" ...

References

Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453-467.

Examples

```
##
## A simple example
##
data(gabriel1971)
bp <- bpca(gabriel1971)

x11(w=6, h=6)
plot(bp, var.factor=2)

# Exploring the object 'bp' created by the function 'bpca'
class(bp)
names(bp)
str(bp)

summary(bp)
bp$call
bp$eigenval
bp$eigenvec
bp$numb
bp$import
bp$coord
bp$coord$obj
bp$coord$var
bp$var.rb
bp$var.rd
```

gge2003

A didatic matrix of genotypes (rows) and environments (columns)

Description

A didatic matrix of genotypes (rows) and environments (columns) proposed by Weikai Yan and Manjit S. Kang in *GGE biplot analysis: a graphical tool for breeders, geneticists, and agronomists*.

Usage

```
gge2003
```

Format

The format is: num [1:4, 1:3] 20 6 -10 8 -9 12 -6 -12 6 -15 ... - attr(*, "dimnames")=List of 2 ..\$: chr [1:4] "G1" "G2" "G3" "G4" ..\$: chr [1:3] "E1" "E2" "E3"

References

Yan, B. W. and Kang, M. S. (2003) *GGE biplot analysis : a graphical tool for breeders, geneticists, and agronomists*. CRC Press, New York, 288p.

Examples

```
##
## Example from YAN, W & KANG, M.S. GGE biplot analysis : a graphical tool
## for breeders, geneticists, and agronomists
##

data(gge2003)
bp <- bpca(t(gge2003), var.rb=TRUE)

as.dist(bp$var.rb)

x11(w=8, h=4)
op = par(no.readonly=TRUE)
par(mfrow=c(1,2))

plot(bpca(gge2003, var.pos=2),
main='Columns as variables \n (var.pos=2)',
var.col=1,
obj.col=c(2:4, 2), obj.cex=.8)

plot(bpca(gge2003, var.pos=1),
main='Rows as variables \n (var.pos=1)',
var.col=1,
obj.col=2:4, obj.cex=.8)

par(op)
```

plot

Biplot of Multivariate Data Based on Principal Components Analysis

Description

These are methods for objects of class `bpca.2d`, `bpca.3d` and `qbpca`.

Usage

```
## S3 method for class 'bpca.2d'
plot(x, ref.lines=TRUE, ref.color='navy', ref.lty='dotted',
var.factor=1, var.color='red3', var.lty='solid', var.pch=20,
var.pos=4, var.cex=.6, var.offset=.2,
obj.color='black', obj.pch=20, obj.pos=4, obj.cex=.6, obj.offset=.2,
obj.names=TRUE, obj.labels=rownames(x$coord$objects), obj.identify=FALSE,
xlim, ylim, xlab, ylab, ...)

## S3 method for class 'bpca.3d'
plot(x, rgl.use=FALSE,
ref.lines=TRUE, ref.color='navy', ref.lty=ifelse(rgl.use, NA, 'dotted'),
clear3d=ifelse(rgl.use, TRUE, NULL), simple.axes=ifelse(rgl.use, TRUE, NULL),
```

```

aspect=ifelse(rgl.use, c(1, 1, 1), NULL),
var.factor=1, var.color='red3', var.lty=ifelse(rgl.use, NA, 'solid'),
var.pch=ifelse(rgl.use, NULL, 20), var.pos=ifelse(rgl.use, 0, 4),
var.cex=ifelse(rgl.use, .8, .6), var.offset=ifelse(rgl.use, NULL, .2),
obj.color='black', obj.pch=ifelse(rgl.use, NULL, 20),
obj.pos=ifelse(rgl.use, 0, 4), obj.cex=ifelse(rgl.use, .8, .6),
obj.offset=ifelse(rgl.use, NULL, .2), obj.names=TRUE,
obj.labels=rownames(x$coord$objects), obj.identify=FALSE,
box=FALSE, angle=ifelse(rgl.use, NULL, 40), xlim, ylim, zlim, xlab, ylab,
zlab, ...)

## S3 method for class 'qbpca'
plot(x, xlab='Index', ylab='r', pch=c(1,8), col=c(4,2), ...)

```

Arguments

<code>x</code>	A <code>bpca.2d</code> , <code>bpca.3d</code> or <code>qbpca</code> object.
<code>rgl.use</code>	A logical value. If TRUE the 3d scatter will be done with the <code>rgl</code> environment, in another way the <code>scatterplot3d</code> will be used.
<code>ref.lines</code>	A logical value. If TRUE, the reference lines will be added to the biplot.
<code>ref.color</code>	Line color for reference lines.
<code>ref.lty</code>	Line type of the reference lines.
<code>clear3d</code>	A logical value. It clears (or not) a 3d biplot before making a new one. Used only if <code>rgl.use=TRUE</code> .
<code>simple.axes</code>	A logical value to draw simple axes. Used only if <code>rgl.use=TRUE</code> .
<code>aspect</code>	A vector of the apparent ratios of the x, y, and z axes of the bounding box. Used only if <code>rgl.use=TRUE</code> .
<code>var.factor</code>	Factor of expansion/reduction of length lines of the variables.
<code>var.color</code>	A vector of colors for the variables representation.
<code>var.lty</code>	Line type for the variables. Used only if <code>rgl.use=FALSE</code> .
<code>var.pch</code>	A vector of plotting symbols or characters for the variables. If too short, the values are recycled. Used only if <code>rgl.use=FALSE</code> .
<code>var.pos</code>	Position of labels for the variables.
<code>var.cex</code>	Character expansion for the variables.
<code>var.offset</code>	The distance (in character widths) which separates the label from identified points of variables. Negative values are allowed. Used only if <code>rgl.use=FALSE</code> .
<code>obj.color</code>	A vector of colors for the objects representation.
<code>obj.pch</code>	A vector of plotting symbols or characters for objects. If too short, the values are recycled. Used only if <code>rgl.use=FALSE</code> .
<code>obj.pos</code>	Position of labels for objects.
<code>obj.cex</code>	Character expansion for objects.
<code>obj.offset</code>	The distance (in character widths) which separates the label from identified points of objects. Negative values are allowed. Used only if <code>rgl.use=FALSE</code> .

<code>obj.names</code>	A logical value to represent objects as spheres or points.
<code>obj.identify</code>	A logical value. If <code>TRUE</code> , it is possible, in interactive way, to identify the objects with the mouse.
<code>obj.labels</code>	A vector of labels for objects.
<code>box</code>	A logical value to whether to draw a box. Used only if <code>rgl.use=TRUE</code> .
<code>angle</code>	Angle between x and y axis (Attention: result depends on scaling). For $180 < \text{angle} < 360$ the returned functions <code>xyz.convert</code> and <code>points3d</code> will not work properly. Used only if <code>rgl.use=FALSE</code> .
<code>pch</code>	A vector of plotting symbols or characters.
<code>col</code>	A vector of colors.
<code>xlab</code>	A label for the x axis.
<code>ylab</code>	A label for the y axis.
<code>zlab</code>	A label for the z axis (<code>bPCA.3d</code> only).
<code>xlim</code>	The x limits of the plot.
<code>ylim</code>	The y limits of the plot.
<code>zlim</code>	The z limits of the plot (<code>bPCA.3d</code> only).
<code>...</code>	Other graphical parameters may also be passed as arguments to these functions.

Details

A biplot aims to represent both the observations and variables of a matrix of multivariate data on the same plot.

The methods `plot.bPCA.2d` draw a 2d biplot (PC1 and PC2 on axis x and y, respectively). It uses the traditional graphics system.

The methods `plot.bPCA.3d` draw a 3d biplot (PC1, PC2 and PC3 on axis x, y and z, respectively) in two ways:

- *static*: It uses the package `scatterplot3d` based on traditional graphic system. Use the parameter `rgl.use=FALSE` for it.
- *dynamic*: It uses the package `rgl`: a 3D real-time rendering device driver system for R. Use the parameter `rgl.use=TRUE` for it.

The method `qb.pca` draws a scatterplot of the correlations observed (in the data) and projected (under the biplot) related to the variables. It uses the traditional graphics system. `qb.pca` is a simple (and graphical) measure of the quality of the biplot reduction associated to the variables.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>)

and

Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

References

- Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453-467.
- Gower, J.C. and Hand, D. J. (1996) *Biplots*. Chapman & Hall.
- Galindo, M. P. (1986) Una alternativa de representacion simultanea: HJ-Biplot. *Questiio*, 10(1):13-23, 1986.
- Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.
- Yan, B. W. and Kang, M. S. (2003) *GGE biplot analysis : a graphical tool for breeders, geneticists, and agronomists*. CRC Press, New York, 288p.

See Also

[biplot.prcomp](#)

Examples

```
##
## Example 1
## Computing and plotting a bpca object with 'graphics' package - 2d
##

bp <- bpca(gabriel1971)

x11(w=6, h=6)
oask <- devAskNewPage(dev.interactive(orNone = TRUE))
plot(bp, var.factor=2)

# Additional graphical parameters (nonsense)
plot(bpca(gabriel1971, meth='sqrt'),
main='gabriel1971 - sqrt', sub='The graphical parameters are working fine!',
var.factor=2, var.cex=.6, var.col=rainbow(9), var.pch='v',
obj.pch='o', obj.cex=.5, obj.col=rainbow(8), obj.pos=1, obj.offset=.5)

##
## Example 2
## Computing and plotting a bpca object with 'scatterplot3d' package - 3d
##

bp <- bpca(gabriel1971, lambda.end=3)

plot(bp, var.factor=3)

# Additional graphical parameters (nonsense)
plot(bpca(gabriel1971, lambda.end=3, meth='jk'),
main='gabriel1971 - jk', sub='The graphical parameters are working fine!',
var.factor=6, var.pch='+', var.cex=.6, var.col='green4',
obj.pch='*', obj.cex=.8, obj.col=1:8,
ref.lty='solid', ref.col='red', angle=70)
```

```

##
## Example 3
## Computing and plotting a bpca object with 'rgl' package - 3d
##

plot(bpca(gabriel1971, lambda.end=3),
      rgl.use=TRUE, var.factor=2)

# Suggestion: Interact with the graphic with the mouse
# left button: press, maintain and movement it to interactive rotation;
# right button: press, maintain and movement it to interactive zoom.
# Enjoy it!

##
## Example 4
## Grouping objects with different symbols and colors - 2d and 3d
##

# 2d
plot(bpca(iris[-5]),
      var.factor=.3, var.cex=.7,
      obj.names=FALSE, obj.cex=1.5,
      obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)],
      obj.pch=c('+', '*', '-')[unclass(iris$Species)])

# 3d static
plot(bpca(iris[-5], lambda.end=3),
      var.factor=.2, var.color=c('blue', 'red'), var.cex=1,
      obj.names=FALSE, obj.cex=1,
      obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)],
      obj.pch=c('+', '*', '-')[unclass(iris$Species)])

# 3d dinamic
plot(bpca(iris[-5], method='hj', lambda.end=3), rgl.use=TRUE,
      var.col='brown', var.factor=.3, var.cex=1.2,
      obj.names=FALSE, obj.cex=.8,
      obj.col=c('red', 'green3', 'orange')[unclass(iris$Species)],
      simple.axes=FALSE, box=TRUE)

##
## Example 5
## Computing and plotting a bpca object with 'obj.identify=TRUE' parameter - 2d
##

bp <- bpca(gabriel1971)

# Normal labels
if(interactive())
plot(bp, obj.names=FALSE, obj.identify=TRUE)

# Alternative labels
if(interactive())
plot(bp, obj.names=FALSE,

```

```
obj.labels=c('toi', 'kit', 'bat', 'ele', 'wat', 'rad', 'tv', 'ref'),
obj.identify=TRUE)

##
## Example 6
## Computing and plotting a bpca object with 'obj.identify=TRUE' parameter - 3d
##

bp <- bpca(gabriel1971, lambda.end=3)

# Normal labels
if(interactive())
plot(bp, obj.names=FALSE, obj.identify=TRUE)

# Alternative labels
if(interactive())
plot(bp, obj.names=FALSE,
obj.labels=c('toi', 'kit', 'bat', 'ele', 'wat', 'rad', 'tv', 'ref'),
obj.identify=TRUE)

devAskNewPage(oask)
```

qbpca

Quality of the Representation of Variables by Biplot

Description

This function returns an object of the class `qbpca`. It is a simple measure of the quality of biplot representation of the variables. The observed (in the data) and projected (under biplot reduction) correlations are computed.

Usage

```
qbpca(x, bpca)
```

Arguments

`x` A `data.frame` or `matrix` object.

`bpca` A object of the class `bpca`.

Details

This function binds the vectors of observed (from the matrix or `data.frame`) and projected (under biplot reduction) correlations for all variables.

Value

An object of class *qbpca and data.frame* with two columns:

obs	A vector of the observed correlations for all variables.
var.rb	A vector of the projected correlations for all variables determined under biplot reduction).

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

References

Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.

See Also

[bpca](#)

Examples

```
##
## Example 1
## Example of 'var.rb=TRUE' parameter as a measure of the quality of the biplot - 2d
##

oask <- devAskNewPage(dev.interactive(orNone = TRUE))

## Differences between methods of factorization
# SQR
bp1 <- bpca(gabriel1971, meth='sqrt', var.rb=TRUE)
qbp1 <- qbpca(gabriel1971, bp1)
plot(qbp1, main='sqrt - 2d \n (poor)')

# JK
bp2 <- bpca(gabriel1971, meth='jk', var.rb=TRUE)
qbp2 <- qbpca(gabriel1971, bp2)
plot(qbp2, main='jk - 2d \n (very poor)')

# GH
bp3 <- bpca(gabriel1971, meth='gh', var.rb=TRUE)
qbp3 <- qbpca(gabriel1971, bp3)
plot(qbp3, main='gh - 2d \n (good)')

# HJ
bp4 <- bpca(gabriel1971, meth='hj', lambda.end=2, var.rb=TRUE)
qbp4 <- qbpca(gabriel1971, bp4)
plot(qbp4, main='hj - 2d \n (good)')
```

```

##
## Example 2
## Example of 'var.rb=TRUE' parameter as a measure of the quality of the biplot - 3d
##

## Differences between methods of factorization
# SQR
bp1 <- bpca(gabriel1971, meth='sqr', lambda.end=3, var.rb=TRUE)
qbp1 <- qbpca(gabriel1971, bp1)
plot(qbp1, main='sqr - 3d \n (poor)')

# JK
bp2 <- bpca(gabriel1971, meth='jk', lambda.end=3, var.rb=TRUE)
qbp2 <- qbpca(gabriel1971, bp2)
plot(qbp2, main='jk - 3d \n (very poor)')

# GH
bp3 <- bpca(gabriel1971, meth='gh', lambda.end=3, var.rb=TRUE)
qbp3 <- qbpca(gabriel1971, bp3)
plot(qbp3, main='gh - 3d \n (whow!)')

# HJ
bp4 <- bpca(gabriel1971, meth='hj', lambda.end=3, var.rb=TRUE)
qbp4 <- qbpca(gabriel1971, bp4)
plot(qbp4, main='hj - 3d \n (whow!)')

devAskNewPage(oask)

```

summary

Summary Method for bpca Objects

Description

Returns (and prints) a summary list for bpca (bpca.2d and bpca.3d) objects.

Usage

```

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

```

Arguments

object	A given object of the class bpca (bpca.2d or bpca.3d).
...	Potential further arguments (require by generic).

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

See Also[bpca](#)**Examples**

```
##
## Example 1
## bpca - 2d
##

bp <- bpca(gabriel1971)
bp

summary(bp)

##
## Example 2
## bpca - 3d
##

bp <- bpca(gabriel1971, lambda.end=3)
bp

summary(bp)
```

`var.rbf`*Projected Correlations by bpca*

Description

Computes the matrix of graphical correlations represented by biplot for a matrix of variable coordinates.

Usage

```
var.rbf(x)
```

Arguments

`x` A given object of the classes *bpca.2d* and *bpca.3d*.

Value

A matrix of graphical correlations represented by biplot.

Note

This function is mainly for internal use in the *bpca* package, and may not remain available (unless we see a good reason).

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

See Also

[bpca](#)

Examples

```
##  
## Particular use  
##  
  
bp1 <- bpca(gabriel1971)  
bp1$var.rb # NA  
  
res <- var.rbf(bp1$coord$var) # Computes the correlations of all variables under  
# the biplot projection  
res  
  
##  
## Common use  
##  
  
bp2 <- bpca(gabriel1971,  
var.rb=TRUE)  
  
bp2$var.rb
```

var.rdf

Diagnostic of Projected Correlations

Description

Computes the diagnostic of poor graphical correlations projected by biplot according to an arbitrary limit.

Usage

```
var.rdf(x, var.rb, limit)
```

Arguments

x	A given object of the classe data.frame or matrix.
var.rb	A given object of the class matrix with the projected correlations by biplot.
limit	A vector giving the percentual limit to define poor representation of variables.

Value

A data.frame of poor graphical correlations projected by biplot.

Note

This function is mainly for internal use in the bpca package, and may not remain available (unless we see a good reason).

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
and
Clarice Garcia Borges Demetrio (<clarice@esalq.usp.br>)

See Also

[bpca](#).

Examples

```
##
## Example 1
## Diagnostic of gabriel1971 dataset representation
##

oask <- devAskNewPage(dev.interactive(orNone = TRUE))

bp1 <- bpca(gabriel1971, meth='hj', var.rb=TRUE)

res <- var.rdf(gabriel1971, bp1$var.rb, lim=3)
res
class(res)

##
## Example 2
## Diagnostic of gabriel1971 dataset representation with var.rd parameter
##

bp2 <- bpca(gabriel1971, meth='hj', lambda.end=2,
var.rb=TRUE, var.rd=TRUE, limit=3)

plot(bp2, var.factor=2)

bp2$var.rd
bp2$eigenvectors

# Graphical visualization of the importance of the variables not contemplated
# in the reduction
plot(bpca(gabriel1971, meth='hj', lambda.ini=3, lambda.end=4), main='hj')

# Interpretation:
```

```

# RUR followed by CRISTIAN contains information dimensions that
# wasn't contemplated by the biplot reduction (PC3).
# Between all, RUR followed by CRISTIAN, variables are the most poor represented
# by a 2d biplot.

##
## Example 3
## Diagnostic of iris dataset representation with var.rd parameter
##

bp3 <- bpca(iris[-5], var.rb=TRUE, var.rd=TRUE, limit=3)

plot(bp3, obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)], var.factor=.3)

bp3$var.rd
bp3$eigenvectors

# Graphical diagnostic
plot(bpca(iris[-5], lambda.ini=3, lambda.end=4),
obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)], var.factor=.6)

# Interpretation:
# Sepal.length followed by Petal.Width contains information in dimensions
# (PC3 - the PC3 is, essentially, a contrast among both) that wasn't fully
# contemplated by the biplot reduction (PC1 and PC2) .
# Therefore, between all variables, they have the most poor representation by a
# 2d biplot.

bp4 <- bpca(iris[-5], lambda.end=3, var.rb=TRUE, var.rd=TRUE, limit=2)

plot(bp4, obj.names=FALSE,
obj.pch=c('+', '-', '*')[unclass(iris$Species)],
obj.col=c('red', 'green3', 'blue')[unclass(iris$Species)], obj.cex=1)

bp4$var.rd
bp4$eigenvectors
round(bp3$var.rb, 2)
round(cor(iris[-5]), 2)

# Good representation of all variables with a 3d biplot!

devAskNewPage(oask)

```

Index

*Topic **datasets**

gabriel1971, [10](#)

gge2003, [11](#)

*Topic **multivariate**

bpca, [3](#)

bpca-package, [2](#)

dt.tools, [8](#)

plot, [12](#)

qbpca, [17](#)

summary, [19](#)

var.rbf, [20](#)

var.rdf, [21](#)

*Topic **package**

bpca-package, [2](#)

biplot.prcomp, [6](#), [15](#)

bpca, [3](#), [9](#), [17](#), [18](#), [20–22](#)

bpca-package, [2](#)

dt.tools, [8](#)

gabriel1971, [10](#)

gge2003, [11](#)

plot, [12](#)

qbpca, [17](#)

summary, [19](#)

var.rbf, [20](#)

var.rdf, [21](#)