

Package ‘bpca’

May 25, 2026

Version 1.5-0

Date 2026-05-25

Title Biplot Analysis for Multivariate Data Using Principal Components

Maintainer José C. Faria <joseclaudio.faria@gmail.com>

Depends R (>= 4.1.0)

Imports scatterplot3d, rgl, xtable

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Description Provides tools for creating 2D and 3D biplots of multivariate data based on principal components analysis, together with diagnostics for reduction quality and enhanced visualization of variables and objects.

License GPL-2

URL <https://github.com/jcfaria/bpca>

BugReports <https://github.com/jcfaria/bpca/issues>

Encoding UTF-8

LazyData true

NeedsCompilation no

Author José C. Faria [aut, cre],
Ivan B. Allaman [aut],
Clarice G. B. Demétrio [aut]

Repository CRAN

Date/Publication 2026-05-25 17:30:02 UTC

Contents

bpca-package	2
bpca	6
dt.tools	12

gabriel1971	14
gge2003	15
marina	16
ontario	17
plot	18
print.xtable	30
qbPCA	32
summary.bpca	36
var.rbf	37
var.rdf	38
xtable.bpca	41

Index	44
--------------	-----------

bpca-package	<i>Biplot of Multivariate Data Based on Principal Component Analysis</i>
--------------	--

Description

Provides 2D and 3D biplot methods and diagnostic tools to evaluate the quality of dimensionality reduction.

Author(s)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

References

- Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453–467.
- Galindo Vilardón, M. P. (1986) Una alternativa de representación simultánea: HJ-Biplot. *Qüestió*, 10(1):13–23.
- Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.
- Gower, J. C. and Hand, D. J. (1996) *Biplots*. Chapman & Hall.
- 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)
##

dev.new(w=6, h=6)
```

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

## Not run:
# 2D
plot(bpca(iris[-5]),
     var.pos=c(4,
              2,
              3,
              1),
     var.offset=.3,
     var.cex=.7,
     obj.names=FALSE,
     obj.cex=1.5,
     obj.col=c('red',
              'green3',
              'blue')[as.numeric(iris$Species)],
     obj.pch=c('+',
              '* ',
              '-')[as.numeric(iris$Species)])

# 3D static
plot(bpca(iris[-5],
         d=1:3),
     var.color=c('blue', 'red'),
     var.cex=1,
     obj.names=FALSE,
     obj.cex=1,
     obj.col=c('red',
              'green3',
              'blue')[as.numeric(iris$Species)],
     obj.pch=c('+',
              '* ',
              '-')[as.numeric(iris$Species)])

# 3D dynamic
plot(bpca(iris[-5],
         method='hj',
         d=1:3),
     rgl.use=TRUE,
     var.col=c('blue', 'red'),
     var.cex=1.2,
     obj.names=FALSE,
     obj.cex=.8,
     obj.col=c('red',
              'green3',
              'orange')[as.numeric(iris$Species)],
     simple.axes=FALSE,
     box=TRUE)

## End(Not run)

##
## New plotting options
```

```
##
plot(bpca(ontario))

# Labels for all objects
(obj.lab <- paste('g',
                 1:18,
                 sep=''))

# Set obj.labels
plot(bpca(ontario),
     obj.labels=obj.lab)

# Evaluate an object (1 is the default)
plot(bpca(ontario),
     type='eo',
     obj.cex=1)

plot(bpca(ontario),
     type='eo',
     obj.id=7,
     obj.cex=1)

# Set obj.labels
plot(bpca(ontario),
     type='eo',
     obj.labels=obj.lab,
     obj.id=7,
     obj.cex=1)

# The same as above
plot(bpca(ontario),
     type='eo',
     obj.labels=obj.lab,
     obj.id='g7',
     obj.cex=1)

# Evaluate a variable (1 is the default)
plot(bpca(ontario),
     type='ev',
     var.cex=1)

plot(bpca(ontario),
     type='ev',
     var.id='E7',
     obj.labels=obj.lab,
     var.cex=1)

# A complete plot
c1 <- 1:3
plot(bpca(iris[-5]),
     type='ev',
     var.id=1,
     obj.names=FALSE,
```

```
obj.col=c1[as.numeric(iris$Species)],
obj.cex=1)

legend('topleft',
      legend=levels(iris$Species),
      text.col=c1,
      pch=19,
      col=c1,
      cex=.9,
      box.lty=0)

# Compare two objects (1 and 2 are the default)
plot(bpca(ontario),
     type='co')

plot(bpca(ontario),
     type='co',
     obj.labels=obj.lab)

plot(bpca(ontario),
     type='co',
     obj.labels=obj.lab,
     obj.id=13:14)

plot(bpca(ontario),
     type='co',
     obj.labels=obj.lab,
     obj.id=c('g7', 'g13'))

# Compare two variables
plot(bpca(ontario),
     type='cv')

# Which won where/what
plot(bpca(ontario),
     type='ww')

# Discriminativeness vs. representativeness
plot(bpca(ontario),
     type='dv')

# Means vs. stability
plot(bpca(ontario),
     type='ms')

# Rank objects with reference to the ideal variable
plot(bpca(ontario),
     type='ro')

# Rank variables with reference to the ideal object
plot(bpca(ontario),
     type='rv')
```

```
## Not run:
plot(bpca(iris[-5]),
     type='eo',
     obj.id=42,
     obj.cex=1)

plot(bpca(iris[-5]),
     type='ev',
     var.id='Sepal.Width')

plot(bpca(iris[-5]),
     type='ev',
     var.id='Sepal.Length')

## End(Not run)

devAskNewPage(oask)
```

bpca

Biplot of Multivariate Data Based on Principal Component Analysis

Description

Computes biplot reductions for `data.frame`, `matrix`, and `prcomp` objects, returning an object of class `bpca`.

Usage

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

Arguments

<code>x</code>	A <code>data.frame</code> , <code>matrix</code> or <code>prcomp</code> object.
<code>d</code>	A vector indicating the first and last dimensions to be used in the biplot reduction. For example, <code>d=1:3</code> or <code>d=c(1,3)</code> for a 3D biplot. The default is <code>d=1:2</code> .
<code>center</code>	Numeric. The type of centering to be performed: '0' - no centering '1' - global-centered = <code>sweep(x, 1, mean(x))</code> '2' - column-centered = <code>sweep(x, 2, apply(x, 2, mean))</code> '3' - double-centered = <code>sweep(sweep(x, 1, apply(x, 1, mean)), 2, apply(x, 2, mean)) + mean(x)</code> The default is 2.
<code>scale</code>	Logical. Indicates whether variables are scaled to unit variance before analysis: FALSE - no scaling; TRUE - scaling enabled.
<code>method</code>	A character value indicating the factorization method: 'hj' - 'HJ' ('symmetric', Galindo Villardón (1986)); 'sqrt' - 'SQRT' ('square root - symmetric', Gabriel (1971)); 'jk' - 'JK' ('row metric preserving', Gabriel (1971)); 'gh' - 'GH' ('column metric preserving', Gabriel (1971)).
<code>iec</code>	Logical. If TRUE, the matrix of eigenvalues and the coordinates of objects and variables are inverted. The default is FALSE.
<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>	Numeric threshold (percentage) used to define poor variable representation.
<code>...</code>	Additional parameters. Required for S3 method dispatch.

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, `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 *tripplot* (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

$$svd({}_n Y_p)$$

- 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 ' $nu > 0$ '. Dimension ' $c(n, nu)$ ';
- v : A matrix whose columns contain the right singular vectors of Y , present if ' $nv > 0$ '. Dimension $c(p, nv)$.

and also,

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

$$n = n_objects(Y)$$

it is possible to obtain 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 - symmetric*, Galindo Villardón (1986):

$$g = u * s^2$$

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

- *SQRT - square root symmetric*, 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, method `bpca.prcomp` recovers values from the `prcomp` object. In other words, it regenerates (or computes) the inverse of the svd decomposition of the given data

$${}_n Y_p$$

After that, it calls `bpca.default` with the appropriate 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)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

References

- Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453–467.
- Galindo Vilardón, M. P. (1986) Una alternativa de representación simultánea: HJ-Biplot. *Qüestió*, 10(1):13–23.
- Johnson, R. A. and Wichern, D. W. (1988) *Applied multivariate statistical analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 6 ed.
- Gower, J. C. and Hand, D. J. (1996) *Biplots*. Chapman & Hall.
- 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
## Compute and plot a bpca object with base graphics (2D)
##

bp <- bpca(gabriel1971)

dev.new(w=6, h=6)
oask <- devAskNewPage(dev.interactive(orNone=TRUE))
plot(bp)

# Explore the object created by 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

## Not run:
##
## Example 2
## Compute and plot a bpca object with scatterplot3d (3D)
##

bp <- bpca(gabriel1971,
            d=2:4)

plot(bp)

# Explore the object created by 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
## Compute and plot a bpca object with rgl (3D)
##

plot(bpca(gabriel1971,
          d=1:3),
      rgl.use=TRUE)

# Tip: interact with the graphic using the mouse
# left button: click and drag to rotate;
# right button: click and drag to zoom.

##
## Example 4
## Group objects using different symbols and colors (2D and 3D)
##

# 2D
plot(bpca(iris[-5]),
     var.cex=.7,
     obj.names=FALSE,
     obj.cex=1.5,
     obj.col=c('red',
               'green3',
               'blue')[as.numeric(iris$Species)],
     obj.pch=c('+',
                'x',
                '-') [as.numeric(iris$Species)])

# 3D static
plot(bpca(iris[-5],
          d=1:3),
     var.color=c('blue', 'red'),
     var.cex=1,
     obj.names=FALSE,
     obj.cex=1,
     obj.col=c('red',
               'green3',
               'blue')[as.numeric(iris$Species)],
     obj.pch=c('+',
                'x',
                '-') [as.numeric(iris$Species)])

# 3D dynamic
plot(bpca(iris[-5],
          method='hj',
          d=1:3),
```

```

    rgl.use=TRUE,
    var.col=c('blue', 'red'),
    var.cex=1.2,
    obj.names=FALSE,
    obj.cex=.8,
    obj.col=c('red',
              'green3',
              'orange')[as.numeric(iris$Species)],
    simple.axes=FALSE,
    box=TRUE)

## End(Not run)

devAskNewPage(oask)

```

dt.tools

Data Tools for Multivariate Analysis

Description

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

Usage

```

dt.tools(x,
         center=2,
         scale=TRUE)

```

Arguments

x	A <code>data.frame</code> or <code>matrix</code> object.
center	Numeric. The type of centering to be performed: '0' - no centering; '1' - global-centered = <code>sweep(x, 1, mean(x))</code> ; '2' - column-centered = <code>sweep(x, 2, apply(x, 2, mean))</code> ; '3' - double-centered = <code>sweep(sweep(x, 1, apply(x, 1, mean)), 2, apply(x, 2, mean)) + mean(x)</code> . The default is 2.
scale	Logical. A value indicating whether the variables should be scaled to have unit variance before the analysis takes place: FALSE - no scaling; TRUE - scaling enabled.

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 (`center=2`), 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)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

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 lengths, angles between variable vectors,
## and variable correlations from data.frame or matrix objects (n x p)
## n = rows (objects)
## p = columns (variables)
##

dt <- dt.tools(iris,
               2) # Non-numeric columns are ignored internally.

# Explore the object created by dt.tools()
class(dt)
names(dt)
str(dt)

dt$length
dt$angle
dt$r
dt
```

```
# Checking the determinations
(iris.tools <- round(dt.tools(iris,
                           center=2)$r,
                    5))

(iris.obsv <- round(cor(iris[-5]),
                    5))

all(iris.tools == iris.obsv)
```

gabriel1971	<i>Percentages of households having various facilities and appliances in East Jerusalem Arab areas, by quarters of the town</i>
-------------	---

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 indicates its overall prevalence.

Usage

```
data(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(*, "dimnames")=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)

dev.new(w=6, h=6)
plot(bp)
```

```
# Explore the object created by 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 didactic matrix of genotypes (rows) and environments (columns)

Description

A didactic 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

```
data(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 and Kang (2003), GGE biplot analysis
## for breeders, geneticists, and agronomists
##

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

as.dist(bp$var.rb)

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

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

plot(bpca(t(gge2003)),
     main='Rows as variables',
     var.col=1,
     obj.col=c(2:4, 2),
     obj.cex=.8)

par(op)
```

marina

Films shown at five festivals in Brazil from 2007 to 2011

Description

A data frame containing films shown at five festivals in Brazil from 2007 to 2011.

Usage

```
data(marina)
```

Format

The format is:

```
'data.frame': 25 obs. of 6 variables:
 year: int  2011 2011 2011 2011 2011 2010 2010 2010 2010 2010 ...
 regE: Factor w/ 5 levels "CO","N","NE",...: 1 5 4 3 2 1 5 4 3 2 ...
  F  : int  84 55 63 44 25 40 54 37 49 27 ...
  D  : int  26 13 19 16 7 9 14 11 19 6 ...
 MD : int  22 9 13 14 5 6 11 10 13 3 ...
```

```

WD : int  4 4 6 2 2 3 3 1 6 3 ...
#
# Description
#
year: Year in which the film was shown
regE: Region where the film was shown
F   : Total number of films
D   : Number of documentaries
MD  : Documentary directed by men
WD  : Documentaries directed by women

```

Examples

```
data(marina)
```

```
marina
```

```
ontario          Ontario winter wheat (1993)
```

Description

Sample yields from the 1993 Ontario winter wheat (*Triticum aestivum* L.) performance trials, in which 18 cultivars were tested at nine locations (Yan and Kang, 2003).

Usage

```
data(ontario)
```

Format

A data frame with 18 observations on the following 10 variables.

Source

Yan W, Kang MS (2003). GGE Biplot Analysis: A Graphical Tool for Breeders, Geneticists, and Agronomists. CRC Press, Boca Raton, FL, USA.

Examples

```
data(ontario)
```

```
# 2D
plot(bpca(ontario,
         d=1:2))
```

```
# 3D
plot(bpca(ontario,
```

```

    d=1:3),
    rgl.use=TRUE)

```

plot

Biplot of Multivariate Data Based on Principal Component Analysis

Description

These are methods for objects of class `bPCA.2d`, `bPCA.3d` and `qBPCA`.

These functions provide a 2D or 3D graphical representation of the objects and variables from a `bPCA` object. It allows for manual or automatic positioning of labels to improve visualization and avoid overlaps.

Usage

```

## S3 method for class 'bPCA.2d'
plot(x,
     type=c('bp',
            'eo',
            'ev',
            'co',
            'cv',
            'ww',
            'dv',
            'ms',
            'ro',
            'rv'),
     c.color='darkgray',
     c.lwd=1,
     c.number=5,
     c.radio=1,
     obj.id=1:2,
     var.id=1,
     base.color='red3',
     base.lty='dotted',
     proj.color='gray',
     proj.lty='dotted',
     a.color='blue',
     a.lty='solid',
     a.lwd=2,
     a.length=.1,
     ref.lines=TRUE,
     ref.color='navy',
     ref.lty='dotted',
     var.factor=NULL,
     var.color='red3',
     var.lty='solid',

```

```
var.pch=20,  
var.cex=.6,  
var.pos=NULL,  
var.offset=.2,  
obj.factor=1,  
obj.color='black',  
obj.pch=20,  
obj.pos=4,  
obj.cex=.6,  
obj.offset=.2,  
obj.names=TRUE,  
obj.labels,  
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=NULL,  
  var.color='red3',  
  var.lty=ifelse(rgl.use, NA, 'solid'),  
  var.pch=ifelse(rgl.use, NULL, 20),  
  var.cex=ifelse(rgl.use, .8, .6),  
  var.pos=ifelse(rgl.use, 0, NA),  
  var.offset=ifelse(rgl.use, 0.2, 0.2),  
  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,  
  obj.identify=FALSE,  
  box=FALSE,  
  angle=ifelse(rgl.use, NULL, 40),  
  xlim,  
  ylim,  
  zlim,  
  xlab,
```

```

      ylab,
      zlab, ...)

## S3 method for class 'qbPCA'
plot(x,
      limit=10,
      highlight.col='gray75',
      highlight.lty='dashed',
      highlight.pad=0.05,
      highlight.width=0.25,
      highlight.guides=TRUE,
      pair.labels=TRUE,
      label.max.nchar=NULL,
      label.side='bottom',
      label.angle=45,
      label.cex=0.6,
      label.offset=0.02,
      xlab='',
      ylab='r',
      pch=c(1,8),
      col=c(4,2), ...)

```

Arguments

<code>x</code>	An object of class <code>bpca.2d</code> , <code>bpca.3d</code> , or <code>qbPCA</code> .
<code>type</code>	Type of biplot: 'bp' - conventional biplot; 'eo' - evaluate an object; 'ev' - evaluate a variable; 'co' - compare two objects; 'cv' - compare (relationship among) variables; 'ww' - which won where/what; 'dv' - discriminativeness vs. representativeness; 'ms' - mean vs. stability; 'ro' - rank objects with reference to an "ideal" variable; 'rv' - rank variables with reference to an "ideal" object.
<code>c.color</code>	The color of circles.
<code>c.lwd</code>	The lwd of circles.
<code>c.number</code>	The number of circles.
<code>c.radio</code>	The radius of circles.
<code>obj.id</code>	An object(s) number(s) or name(s). It is used with reprojections to identify the object(s) when the <code>type</code> option is 'eo' or 'co'. The default is '1' for 'eo' and 1:2 for 'co'.
<code>var.id</code>	A variable number or name. It is used with reprojections to identify the variable when the <code>type</code> option is 'ev'.
<code>base.color</code>	The color for the base lines. It is used only with reprojections.

<code>base.lty</code>	The 'lty' for the base lines. It is used only with reprojections.
<code>proj.color</code>	The color for the projections lines. It is used only with reprojections.
<code>proj.lty</code>	The 'lty' for the projections lines. It is used only with reprojections.
<code>a.color</code>	The color for the arrow. It is used only with reprojections.
<code>a.lty</code>	The 'lty' for the arrow. It is used only with reprojections.
<code>a.lwd</code>	The 'lwd' for the arrow. It is used only with reprojections.
<code>a.length</code>	The 'length' for the arrow. It is used only with reprojections.
<code>rgl.use</code>	A logical value. If TRUE, the 3D scatter is drawn with <code>rgl</code> ; otherwise, <code>scatterplot3d</code> is 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. Clears (or not) a 3D biplot before drawing 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.cex</code>	Character expansion for the variables.
<code>var.pos</code>	A vector or a single integer (1, 2, 3, or 4) to specify the position of the variable labels relative to the points. The convention is: 1 = below, 2 = left, 3 = above, and 4 = right. If NULL (2D or 3D static) or 0 (3D dynamic), an automatic radial adjustment is used. A vector can be used to specify different positions for each variable to avoid overlaps.
<code>var.offset</code>	A numeric value to specify the distance (offset) of the label from the variable point. In 3D dynamic mode (<code>rgl</code>), it is internally scaled to match the visual output of the 2D and 3D static modes.
<code>obj.factor</code>	Factor of expansion/reduction of length lines of the objects.
<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 (offset) 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 TRUE, objects can be identified interactively 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 $180 < angle < 360$ 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>limit</code>	Numeric threshold (percentage) used to mark pairs with poor representation in <code>plot.qbpca</code> . Pairs where $100 * \text{abs}(r.\text{obs} - r.\text{rb}) > \text{limit}$ are highlighted with dashed rectangles around the pair of points. Default is 10.
<code>highlight.col</code>	Color used for the dashed rectangle border in <code>plot.qbpca</code> . Default is 'gray75'.
<code>highlight.lty</code>	Line type used for the dashed rectangle border in <code>plot.qbpca</code> . Default is 'dashed'.
<code>highlight.pad</code>	Vertical margin added above and below the two points in each highlighted pair. Default is 0.05.
<code>highlight.width</code>	Half-width of the dashed highlight rectangle around each flagged pair along the x-axis. Must be in $(0, 0.5]$. Default is 0.25.
<code>highlight.guides</code>	Logical. If TRUE, draws dashed guide segments from the lower edge of each highlighted rectangle down to the x-axis to facilitate pair identification. Default is TRUE.
<code>pair.labels</code>	Logical. If TRUE, variable-pair labels from <code>qbpca</code> row names are drawn on the x-axis. When labels are enabled, <code>plot.qbpca</code> automatically adjusts the corresponding plot margin to fit long rotated labels and restores the original device margin afterwards. Default is TRUE.
<code>label.max.nchar</code>	Optional maximum number of characters for each variable-pair label. Labels longer than this value are truncated and suffixed with <code>...</code> . Set NULL to keep full labels. Default is NULL.
<code>label.side</code>	Side where pair labels are drawn when <code>pair.labels=TRUE</code> . Allowed values are 'bottom' and 'top'. Default is 'bottom'.
<code>label.angle</code>	Numeric angle (degrees) used to rotate x-axis pair labels when <code>pair.labels=TRUE</code> . Default is 45.
<code>label.cex</code>	Character expansion for rotated x-axis pair labels. Default is 0.6.
<code>label.offset</code>	Proportional vertical offset used to place rotated x-axis labels below or above the plotting region. Default is 0.02.
<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>bcpa.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>bcpa.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 method `plot.bpca.2d` draws a 2D biplot (PC1 and PC2 on axis 'x' and 'y', respectively). It uses the traditional graphics system.

The method `plot.bpca.3d` draws a 3D biplot (PC1, PC2 and PC3 on axis 'x', 'y' and 'z', respectively) in two ways:

1. `static`: It uses the package `scatterplot3d` based on traditional graphic system. Use the parameter `'rgl.use=FALSE'` for it. It is the default.
2. `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 `plot.qbpca` draws a scatterplot of observed (in the data) and projected (under the biplot) related to the variables. It uses the traditional graphics system. `qbpca` is a simple (and graphical) measure of the quality of the biplot reduction associated to the variables.

Author(s)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

References

- Gabriel, K. R. (1971) The biplot graphical display of matrices with application to principal component analysis. *Biometrika* **58**, 453-467.
- Galindo Vilardón, M. P. (1986) Una alternativa de representación simultánea: HJ-Biplot. *Qüestió*, 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.
- Gower, J.C. and Hand, D. J. (1996) *Biplots*. Chapman & Hall.
- 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

```
# To avoid overlap in a 2D biplot by manually positioning labels:
# Suppose we have 4 variables and want to position them differently
plot(bpca(ontario),
      var.pos = c(1,
                 3,
                 4,
```

```

        2),
    var.offset = 0.5)

# For 3D dynamic plots with custom offsets:
if(interactive()) {
  plot(bpca(ontario, d=1:3),
       rgl.use = TRUE,
       var.pos = 3,
       var.offset = 0.8)
}

##
## Example 1
## Computing and plotting a bpca object with base graphics (2D)
##

bp <- bpca(gabriel1971)

dev.new(w=6, h=6)
oask <- devAskNewPage(dev.interactive(orNone=TRUE))
plot(bp)

# To avoid overlap in a 2D biplot by manually positioning labels
plot(bp,
     var.pos=c(1,
              3,
              rep(4, 7)),
     var.offset=.3)

# Additional graphical parameters (nonsense)
plot(bpca(gabriel1971,
         meth='sqrt'),
     main='gabriel1971 - sqrt',
     sub='The graphical parameters are working fine!',
     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 (3D)
##

bp <- bpca(gabriel1971,
           d=1:3)

plot(bp,
     var.col=rainbow(9))

```

```

# Additional graphical parameters (nonsense)
plot(bpca(gabriel1971,
         d=1:3,
         meth='jk'),
     main='gabriel1971 - jk',
     sub='The graphical parameters are working fine!',
     var.pch='+',
     var.cex=.6,
     var.col=rainbow(ncol(gabriel1971)),
     obj.pch='*',
     obj.cex=.8,
     obj.col=rainbow(nrow(gabriel1971)),
     ref.lty='dotted',
     ref.col=gray(.6),
     angle=70)

##
## Example 3
## Computing and plotting a bpca object with rgl (3D)
##

plot(bpca(gabriel1971,
         d=1:3),
     rgl.use=TRUE)

# Tip: interact with the graphic using the mouse
# left button: click and drag to rotate;
# right button: click and drag to zoom.

## Not run:
##
## Example 4
## Grouping objects with different symbols and colors (2D and 3D)
##

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

plot(bpca(iris[-5]),
     var.cex=.7,
     var.pos=c(4,
              2,
              3,

```

```

        1),
var.offset=.3,
obj.names=FALSE,
obj.cex=1.5,
obj.col=c('red',
          'green3',
          'blue')[as.numeric(iris$Species)],
obj.pch=c('+',
          '*',
          '-') [as.numeric(iris$Species)]

# 3D static
plot(bpca(iris[-5],
         d=1:3),
     var.color=c('blue', 'red'),
     var.cex=1,
     obj.names=FALSE,
     obj.cex=1,
     obj.col=c('red',
               'green3',
               'blue')[as.numeric(iris$Species)],
     obj.pch=c('+',
               '*',
               '-') [as.numeric(iris$Species)])

# 3D dynamic
plot(bpca(iris[-5],
         method='hj',
         d=1:3),
     rgl.use=TRUE,
     var.col=c('blue', 'red'),
     var.cex=1.2,
     obj.names=FALSE,
     obj.cex=.8,
     obj.col=c('red',
               'green3',
               'orange')[as.numeric(iris$Species)])

##
## Example 5
## Computing and plotting a bpca object with `obj.identify=TRUE` (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` (3D)
##

bp <- bpca(gabriel1971,
           d=1:3)

# 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)

##
## New plotting options
##
plot(bpca(ontario))

# Labels for all objects
(obj.lab <- paste('g',
                 1:18,
                 sep=''))

# Set obj.labels
plot(bpca(ontario),
     obj.labels=obj.lab)
```

```
# Evaluate an object (1 is the default)
plot(bpca(ontario),
     type='eo',
     obj.cex=1)

plot(bpca(ontario),
     type='eo',
     obj.cex=1)

plot(bpca(ontario),
     type='eo',
     obj.id=7,
     obj.cex=1)

# Set obj.labels
plot(bpca(ontario),
     type='eo',
     obj.labels=obj.lab,
     obj.id=7,
     obj.cex=1)

# The same as above
plot(bpca(ontario),
     type='eo',
     obj.labels=obj.lab,
     obj.id='g7',
     obj.cex=1)

# Evaluate a variable (1 is the default)
plot(bpca(ontario),
     type='ev',
     var.cex=1)

plot(bpca(ontario),
     type='ev',
     var.id='E7',
     obj.labels=obj.lab,
     var.cex=1)

# A complete plot
cl <- 1:3
plot(bpca(iris[-5]),
     type='ev',
     var.id=1,
     var.fac=.3,
     obj.names=FALSE,
     obj.cex=.9,
     obj.col=cl[as.numeric(iris$Species)])

legend('topleft',
       legend=levels(iris$Species),
       text.col=cl,
```

```
        pch=19,
        col=c1,
        cex=.9,
        box.lty=0)

# Compare two objects (1 and 2 are the default)
plot(bpca(ontario),
     type='co',
     c.radio=.4,
     c.color='blue',
     c.lwd=2)

plot(bpca(ontario),
     type='co',
     obj.labels=obj.lab,
     c.radio=.5,
     c.color='blue',
     c.lwd=2)

plot(bpca(ontario),
     type='co',
     obj.labels=obj.lab,
     obj.id=13:14)

plot(bpca(ontario),
     type='co',
     obj.labels=obj.lab,
     obj.id=c('g7',
              'g13'))

# Compare two variables
plot(bpca(ontario),
     type='cv',
     c.number=3,
     c.radio=1.5)

# Which won where/what
plot(bpca(ontario),
     type='ww')

# Discriminativeness vs. representativeness
plot(bpca(ontario),
     type='dv')

plot(bpca(ontario),
     type='dv',
     c.number=4,
     c.radio=1)

# Means vs. stability
plot(bpca(ontario),
     type='ms')
```

```

plot(bpca(ontario),
     type='ms',
     c.number=3)

# Rank objects with reference to the ideal variable
plot(bpca(ontario),
     type='ro')

plot(bpca(ontario),
     type='ro',
     c.number=6,
     c.radio=.5)

# Rank variables with reference to the ideal object
plot(bpca(ontario),
     type='rv')

plot(bpca(ontario),
     type='rv',
     c.number=6,
     c.radio=.5)

plot(bpca(iris[-5]),
     type='eo',
     obj.id=42,
     obj.cex=1)

plot(bpca(iris[-5]),
     type='ev',
     var.id='Sepal.Width')

plot(bpca(iris[-5]),
     type='ev',
     var.id='Sepal.Width',
     var.factor=.3)

## End(Not run)

devAskNewPage(oask)

```

print.xtable

Print Method for xtable.bpca Objects

Description

Formatted printing for `xtable.bpca` objects produced by `xtable::xtable(bpca(...))`. By default (`type = "latex"`) the output matches the biplot-style LaTeX table (multirow headers and eigenvalues in column titles). With `type = "html"`, a simpler HTML table is produced for R Markdown and web publishing; column headings include numeric eigenvalues.

Usage

```
## S3 method for class 'xtable.bpca'
print(x,
      hline.after = getOption("xtable.hline.after", NULL),
      include.colnames = getOption("xtable.include.colnames", FALSE),
      add.to.row = getOption("xtable.add.to.row", NULL),
      sanitize.text.function = getOption("xtable.sanitize.text.function", NULL),
      sanitize.rownames.function = getOption("xtable.sanitize.rownames.function",
                                             sanitize.text.function),
      sanitize.colnames.function = getOption("xtable.sanitize.colnames.function",
                                             sanitize.text.function),...)
```

Arguments

<code>x</code>	An object of class <code>xtable</code> or <code>xtable.bpca</code> .
<code>hline.after</code>	When <code>type="latex"</code> , a vector of numbers between -1 and <code>nrow(x)</code> , inclusive, indicating the rows after which a horizontal line should appear. Default value is <code>NULL</code> , which means drawing a line before and after the column names, draw a line before variables and at the end of the table.
<code>include.colnames</code>	If <code>TRUE</code> , column names are printed. The formal argument defaults to <code>getOption("xtable.include.colnames", FALSE)</code> . When <code>type = "html"</code> is requested and <code>include.colnames</code> is <i>not</i> supplied in the call, the method defaults to <code>TRUE</code> so PC columns (with eigenvalues in the header) are shown; set <code>include.colnames = FALSE</code> explicitly to omit them.
<code>add.to.row</code>	A list of two components. The first component (which should be called 'pos') is a list that contains the position of rows on which extra commands should be added at the end. The second component (which should be called 'command') is a character vector of the same length as the first component, which contains the command that should be added at the end of the specified rows. Default value is <code>NULL</code> .
<code>sanitize.text.function</code>	All non-numeric entries (except row and column names) are sanitized in an attempt to remove characters which have special meaning for the output format. If <code>sanitize.text.function</code> is not <code>NULL</code> , it should be a function taking a character vector and returning one, and will be used for the sanitization instead of the default internal function. Default value is <code>NULL</code> .
<code>sanitize.rownames.function</code>	Like the <code>sanitize.text.function</code> , but applicable to row names. The default uses the <code>sanitize.text.function</code> .
<code>sanitize.colnames.function</code>	Like the <code>sanitize.text.function</code> , but applicable to column names. The default uses the <code>sanitize.text.function</code> .
<code>...</code>	Other arguments passed to <code>xtable::print.xtable</code> , in particular <code>type</code> : use <code>type = "html"</code> for HTML tables (e.g. R Markdown HTML output) and the default <code>"latex"</code> (or <code>getOption("xtable.type")</code>) for the biplot-style LaTeX table. For HTML, row continuations do not use the LaTeX & prefix; custom <code>sanitize.rownames.function</code> should return plain text suitable for HTML if you use this type.

Author(s)

Faria, J. C.
 Allaman, I. B.
 Demétrio C. G. B.

Examples

```
## Example 1: Principal labels in Portuguese
library(xtable)

bp2 <- bpca(gabriel1971)
tbl <- xtable(bp2)
rownames(tbl) <- gsub('Eigenvectors', 'Autovetores', rownames(tbl))
rownames(tbl) <- c(rownames(tbl)[1:9], 'Autovalores', 'Variância retida', 'Variância acumulada')
dimnames(tbl)[[2]] <- c('CP 1', 'CP 2')

print(tbl)

## Example 2: With bold in the column
tbl1 <- xtable(bp2)
bold <- function(x){
  paste('\textbf{',
        x,
        '}')
}

print(tbl1,
      sanitize.colnames.function = bold)

# Example 3: With italic row labels
tbl2 <- xtable(bp2)
italic <- function(x){
  paste('& \textit{',
        x,
        '}')
} # The "&" keeps the correct number of table columns.

print(tbl2,
      sanitize.rownames.function = italic)

## HTML table (e.g. R Markdown HTML): pass type via print()
print(tbl,
      type = "html")
```

Description

This function returns an object of 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

<code>x</code>	A <code>data.frame</code> or <code>matrix</code> object.
<code>bpca</code>	An object of class <code>bpca</code> .

Details

This function combines 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:

<code>obs</code>	A vector of the observed correlations for all variables.
<code>var.rb</code>	A vector of projected correlations for all variables determined under biplot reduction.

Row names are generated automatically as variable-pair labels (e.g., V1 vs. V2).

Author(s)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

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 the `var.rb=TRUE` parameter as a quality measure (2D)
##

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

## Differences between methods of factorization
# SQR T
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',
            var.rb=TRUE)

qbp4 <- qbpca(gabriel1971,
              bp4)
```

```
plot(qbp4,
     main='hj - 2D \n (good)')

##
## Example 2
## Example of the `var.rb=TRUE` parameter as a quality measure (3D)
##

## Differences between methods of factorization
# SQR
bp1 <- bpca(gabriel1971,
            meth='sqrt',
            d=1:3,
            var.rb=TRUE)

qbp1 <- qbpca(gabriel1971,
              bp1)

plot(qbp1,
     main='sqrt - 3D \n (poor)')

# JK
bp2 <- bpca(gabriel1971,
            meth='jk',
            d=1:3,
            var.rb=TRUE)

qbp2 <- qbpca(gabriel1971,
              bp2)

plot(qbp2,
     main='jk - 3D \n (very poor)')

# GH
bp3 <- bpca(gabriel1971,
            meth='gh',
            d=1:3,
            var.rb=TRUE)

qbp3 <- qbpca(gabriel1971,
              bp3)

plot(qbp3,
     main='gh - 3D \n (wow!)')

# HJ
bp4 <- bpca(gabriel1971,
            meth='hj',
            d=1:3,
```

```
var.rb=TRUE)

qbp4 <- qbpca(gabriel1971,
             bp4)

plot(qbp4,
     main='hj - 3D \n (wow!)')

devAskNewPage(oask)
```

summary.bpca

Summary Method for bpca Objects

Description

summary.bpca computes and returns a summary of the dimensionality reduction for bpca (bpca.2d and bpca.3d) objects.

print.summary.bpca displays the summary on the console, following the standard S3 convention where summary() returns an object and print() handles the presentation. It replaces the former presentation = TRUE argument of summary.bpca.

Usage

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

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

Arguments

object	An object of class bpca (bpca.2d or bpca.3d).
x	An object of class summary.bpca, as returned by summary.bpca.
...	Potential further arguments (required by the generic).

Author(s)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

See Also

[bpca](#), [xtable.bpca](#)

Examples

```
##
## Example 1
## bpca - 2D
##
bp <- bpca(gabriel1971)

# Returns a summary object.
sm <- summary(bp)
sm

# Prints the summary in a formatted layout.
print(sm)

##
## Example 2
## bpca - 3D
##
bp <- bpca(gabriel1971,
           d=1:3)

sm <- summary(bp)
str(sm)
print(sm)
```

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 An object of class 'bpca.2d' or '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)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

See Also

[bpca](#)

Examples

```
##
## Direct use
##

bp1 <- bpca(gabriel1971)
bp1$var.rb # NA

# Compute correlations of all variables under the biplot projection
(res <- var.rbf(bp1$coord$var))

##
## Typical 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	An object of class <code>data.frame</code> or <code>matrix</code> .
var.rb	A matrix with projected variable correlations from the biplot.
limit	Numeric threshold (percentage) used to define poor variable representation.

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)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

See Also

[bpca](#)

Examples

```
##  
## Example 1  
## Diagnostic of representation quality for the gabriel1971 dataset  
##  
oask <- devAskNewPage(dev.interactive(orNone=TRUE))  
  
bp1 <- bpca(gabriel1971,  
            meth='hj',  
            var.rb=TRUE)  
  
(res <- var.rdf(gabriel1971,  
                bp1$var.rb,  
                lim=3))  
class(res)  
  
##  
## Example 2  
## Diagnostic of representation quality using `var.rd`  
##  
bp2 <- bpca(gabriel1971,  
            meth='hj',  
            var.rb=TRUE,  
            var.rd=TRUE,  
            limit=3)  
  
plot(bp2,  
      var.factor=2)  
  
bp2$var.rd
```

```

bp2$eigenvectors

# Graphical visualization of variable importance not represented
# in the selected reduction
plot(bpca(gabriel1971,
          meth='hj',
          d=3:4),
      main='hj')

# Interpretation:
# RUR followed by CRISTIAN contains information in dimensions not captured
# by the 2D biplot reduction (mainly PC3).
# Among all variables, RUR and CRISTIAN are the most poorly represented
# in a 2D biplot.

## Not run:
##
## Example 3
## Diagnostic of iris representation quality using `var.rd`
##

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

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

bp3$var.rd
bp3$eigenvectors

# Graphical diagnostic
plot(bpca(iris[-5],
          d=3:4),
     obj.col=c('red',
               'green3',
               'blue')[as.numeric(iris$Species)],
     obj.names=FALSE)

# Interpretation:
# Sepal.Length followed by Petal.Width contains information in dimensions
# (mainly PC3) that is not fully captured by the PC1-PC2 reduction.
# Therefore, among all variables, these are the most poorly represented
# by a 2D biplot.

bp4 <- bpca(iris[-5],
            d=1:3,
            var.rb=TRUE,

```

```

        var.rd=TRUE,
        limit=2)

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

## End(Not run)

devAskNewPage(oask)

```

xtable.bpca

LaTeX Table for Biplot Results

Description

Returns a LaTeX table from bpca objects.

Usage

```

## S3 method for class 'bpca'
xtable(x,
       caption = NULL,
       label = NULL,
       align = NULL,
       digits = NULL,
       display = NULL,
       auto = FALSE, ...)

```

Arguments

x	An object of class <code>bpca</code> .
caption	Character vector of length 1 or 2 containing the table's caption or title. If length is 2, the second item is the "short caption" used when LaTeX generates a "List of Tables". Set to NULL to suppress the caption. Default value is NULL.

label	Character vector of length 1 containing the LaTeX <code>\label</code> or HTML anchor. Set to NULL to suppress the label. Default value is NULL.
align	Character vector of length equal to the number of columns of the resulting table, indicating the alignment of the corresponding columns. Also, "l" may be used to produce vertical lines between columns in LaTeX tables, but these are effectively ignored when considering the required length of the supplied vector. If a character vector of length one is supplied, it is split as <code>strsplit(align, "")[[1]]</code> before processing. Since the row names are printed in the first column, the length of align is one greater than <code>ncol(x)</code> if <code>x</code> is a <code>data.frame</code> . Use "l", "r", and "c" to denote left, right, and center alignment, respectively. For a LaTeX column of the specified width. For HTML output the "p" alignment is interpreted as "l", ignoring the width request. Default depends on the class of <code>x</code> .
digits	Numeric vector of length equal to one (in which case it will be replicated as necessary) or to the number of columns of the resulting table or <code>matrix</code> of the same size as the resulting table, indicating the number of digits to display in the corresponding columns. Since the row names are printed in the first column, the length of the vector <code>digits</code> or the number of columns of the <code>matrix</code> <code>digits</code> is one greater than <code>ncol(x)</code> if <code>x</code> is a <code>data.frame</code> . Default depends on the class of <code>x</code> . If values of <code>digits</code> are negative, the corresponding values of <code>x</code> are displayed in scientific format with <code>abs(digits)</code> digits.
display	Character vector of length equal to the number of columns of the resulting table, indicating the format for the corresponding columns. Since the row names are printed in the first column, the length of <code>display</code> is one greater than <code>ncol(x)</code> if <code>x</code> is a <code>data.frame</code> . These values are passed to the <code>formatC</code> function. Use "d" (for integers), "f", "e", "E", "g", "G", "fg" (for reals), or "s" (for strings). "f" gives numbers in the usual <code>xxx.xxx</code> format; "e" and "E" give <code>n.ddde+nn</code> or <code>n.dddE+nn</code> (scientific format); "g" and "G" put <code>x[i]</code> into scientific format only if it saves space to do so. "fg" uses fixed format as "f", but digits as number of significant digits. Note that this can lead to quite long result strings. Default depends on the class of <code>x</code> .
auto	Logical, indicating whether to apply automatic format when no value is passed to <code>align</code> , <code>digits</code> , or <code>display</code> . This autoformat (based on <code>xalign</code> , <code>xdigits</code> , and <code>xdisplay</code>) can be useful to quickly format a typical <code>matrix</code> or <code>data.frame</code> . Default value is FALSE.
...	Additional arguments. (Currently ignored.)

Details

This function extracts tabular information from `x` and returns an object of class `xtable.bpca`, `xtable`, or `data.frame`.

To enable all formatting resources, include the LaTeX package `'multirow'` in the Sweave/Rnoweb preamble.

Value

An object of the class `xtable.bpca`.

Author(s)

Faria, J. C.
Allaman, I. B.
Demétrio C. G. B.

References

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

Examples

```
## Example 1: Simplest use
library(xtable)

bp <- bpca(iris[-5],
          d=1:3)

xtable::xtable(bp)

## Example 2: With caption and label
bp2 <- bpca(gabriel1971)

xtable::xtable(bp2,
               caption='Biplot gabriel1971',
               label='example_2')
```

Index

- * **bpca**
 - summary.bpca, 36
 - xtable.bpca, 41
 - * **datasets**
 - gabriel1971, 14
 - gge2003, 15
 - marina, 16
 - ontario, 17
 - * **latex**
 - xtable.bpca, 41
 - * **movie**
 - marina, 16
 - * **multivariate**
 - bpca, 6
 - bpca-package, 2
 - dt.tools, 12
 - plot, 18
 - print.xtable, 30
 - qbpca, 32
 - summary.bpca, 36
 - var.rbf, 37
 - var.rdf, 38
 - xtable.bpca, 41
 - * **package**
 - bpca-package, 2
 - * **summary**
 - summary.bpca, 36
 - * **table**
 - xtable.bpca, 41
- biplot.prcomp, 9, 23
bpca, 6, 13, 33, 36, 38, 39, 41
bpca-package, 2
- dt.tools, 12
- gabriel1971, 14
gge2003, 15
- marina, 16
- ontario, 17
- plot, 18
print.summary.bpca (summary.bpca), 36
print.xtable, 30
- qbpca, 32
- summary.bpca, 36
- var.rbf, 37
var.rdf, 38
- xtable.bpca, 36, 41