Principal component analysis

Description

A simple function for principal component analysis.

Usage

```
PCA(Y, stand=FALSE, cumfit.obj=TRUE, cumfit.var=TRUE, ...)
# Write total variance, eigenvalues, relative eigenvalues, cumulative rel. eigenvalues to window
name_of_output_object   or   print(name_of_output_object, kk=5, ...)
# Plot the scaling type 1 (distance) biplot or scaling type 2 (correlation) biplot
biplot(name_of_output_object, scaling=1, plot.axes=c(1,2), color.obj="black",
color.var="red", ...)
```

Arguments

`Y`	Data matrix
`stand`	FALSE: center the data by column (variable); do not divide by standard deviation. TRUE: center the data by column (variable) and divide by standard deviation.
`cumfit.obj`	TRUE: compute the table of cumulative fit of the objects
`cumfit.var`	TRUE: compute the table of cumulative fit per variable
`kk`	Number of axes for cumulative fit tables. Default: kk=5.
`scaling`	=1: distance biplot (default). =2: correlation biplot.
`plot.axes`	The axes to be plotted (default: axes 1 and 2).
`color`	Color of the object symbols and variable arrows, and their labels, in the biplots. Defaults: color.obj="black", color.var="red".
`...`	Other parameters passed to `print` or `biplot` functions.

Details

Principal component analysis (PCA) of a data table producing scaling 1 and scaling 2 biplots. The variables must be standardized (stand=TRUE) if they are not all expressed in the same physical dimensions. The default is stand=FALSE.

Scaling type 1 biplot uses matrices F for objects and U for variables; notation as in Legendre and Legendre (1998, Section 9.1). This projection preserves the Euclidean distances among objects.

Scaling type 2 biplot uses matrices G for objects and U2 for variables. This projection preserves the correlations among variables.

Value

Function PCA returns a list containing the following results and matrices:

total.var	Total variance in matrix Y, possibly after standardization.
eigenvalues	PCA eigenvalues.
rel.eigen	Relative eigenvalues.
rel.cum.eigen	Cumulative sum of the relative eigenvalues.
U, F, U2, G	Matrices required to produce the biplot.
cumulative.fit.var	Table of "Cumulative fit per variable" (R^2). The maximum value is 1.
cumulative.fit.obj	Table of "Cumulative fit of the objects". The maximum value is 1.
stand	Logical (TRUE, FALSE) indicating if the variables were standardized.

obj.names, var.names: additional elements needed to produce the biplot.

Reference

Legendre, P. and Legendre, L. 1998. *Numerical Ecology*. 2nd English ed. Elsevier, Amsterdam.

Author

Pierre Legendre, Departement de Sciences Biologiques, Universite de Montreal.

Examples

```
# Example: data from Fig. 9.2 of Legendre and Legendre (1998)

table = matrix(c(2,3,5,7,9,1,4,0,6,2), 5, 2)
rownames(table) = c("Obj1", "Obj2", "Obj3", "Obj4", "Obj5")
colnames(table) = c("Var1", "Var2")
res = PCA(table)
res                       # Print out the summary results
biplot(res, scaling=2)    # Produce a scaling 2 biplot in the graphics window
summary(res)              # Print the structure of the output object
res$U                     # Matrix of eigenvectors of cov(Y) (variable arrows in scaling 1 biplot)
res$F                     # Matrix of principal components (object points in scaling 1 biplot)
res$U2                    # Matrix of variable arrow positions in scaling 2 biplot
res$G                     # Matrix of object point positions in scaling 2 biplot
res$cumulative.fit.var    # Table of cumulative fit per variable
res$cumulative.fit.obj    # Table of cumulative fit of the objects

# Example: the spider data of Aart and Smeenk-Enserink (1975), available in library mvpart.
# The spider data frame has 28 rows and 18 columns. The first 12 columns are abundances of
different species of spiders and the next 6 are environmental data.

library(mvpart)
data(spider)              # Note : this file does not contain site names
```

```
# Hellinger transformation of species data prior to PCA, using function decostand (library vegan)
library(vegan)
spider.hel = decostand(spider[,1:12], "hellinger")
res = PCA(spider.hel)
res                              # Print out the summary results
biplot(res, color.obj="blue")  # Produce a scaling 1 biplot in the graphics window
summary(res)                     # Print the structure of the output object
```