Package ‘PCADSC’

April 19, 2017

Type Package
Title Tools for Principal Component Analysis-Based Data Structure Comparisons
Version 0.8.0
Description A suite of non-parametric, visual tools for assessing differences in data structures for two datasets that contain different observations of the same variables. These tools are all based on Principal Component Analysis (PCA) and thus effectively address differences in the structures of the covariance matrices of the two datasets. The PCASDC tools consist of easy-to-use, intuitive plots that each focus on different aspects of the PCA decompositions. The cumulative eigenvalue (CE) plot describes differences in the variance components (eigenvalues) of the deconstructed covariance matrices. The angle plot presents the information loss when moving from the PCA decomposition of one dataset to the PCA decomposition of the other. The chroma plot describes the loading patterns of the two datasets, thereby presenting the relative weighting and importance of the variables from the original dataset.

Depends R (>= 3.2.2)
License GPL-2
Encoding UTF-8
LazyData true
Imports reshape2, methods, pander, ggplot2, Matrix
RoxygenNote 5.0.1
URL https://github.com/annepetersen1/PCADSC
BugReports https://github.com/annepetersen1/PCADSC/issues
NeedsCompilation no
Author Anne H. Petersen [aut, cre], Bo Markussen [aut]
Maintainer Anne H. Petersen <ahpe@sund.ku.dk>
Repository CRAN
Date/Publication 2017-04-19 10:07:43 UTC
R topics documented:

anglePlot
CEPlot
chromaPlot
doAngle
doCE
doChroma
PCADSC

Index

anglePlot Angle plot

Description

Produce an angle plot from a full or partial PCADSC object, as obtained from a call to PCADSC. In either case, this PCADSC object must have a non-NULL angleInfo slot (see examples). The angle plot compares the eigenvalue- and loading patterns from PCA performed on two datasets that consist of different observations of the same variables.

Usage

anglePlot(x)

Arguments

x A PCADSC or angleInfo object, as produced by PCADSC or doAngle, respectively.

See Also

PCADSC, doAngle

Examples

#load iris data
data(iris)

#Define grouping variable, grouping the observations by whether their species is Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

## Not run:
#make a full PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group")
CEPlot

Cumulative eigenvalue plot

Description

Produce a cumulative eigenvalue (CE) plot from a full or partial PCADSC object, as obtained from a call to PCADSC. In either case, this PCADSC object must have a non-NULL CEInfo slot (see examples). The CE plot compares the eigenvalues obtained from PCA performed separately and jointly on two datasets that consist of different observations of the same variables.

Usage

CEPlot(x, nDraw = NULL)

Arguments

- **x**: A PCADSC or angleInfo object, as produced by PCADSC or doAngle, respectively.
- **nDraw**: A positive integer. The number of simulated cumulative eigenvalue curves that should be added to the plot.

Details

In the x-coordinates, cumulative differences in eigenvalues are shown, while the y-coordinates are the cumulative sum of the joint eigenvalues. The plot is annotated with Kolmogorov-Smirnov and Cramer-von Mises tests evaluated by permutation tests, testing the null hypothesis of no difference in eigenvalues. The plot also features a number of cumulative simulated cumulative eigenvalue curves as dashed lines. Moreover, a shaded area presents pointwise 95% confidence bands for the cumulative difference, also obtained using the permutation test.

See Also

PCADSC, doCE
Examples

```r
# load iris data
data(iris)

# Define grouping variable, grouping the observations by whether their species is
# Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

# Not run: make a PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group")

# make a partial PCADSC object from iris and fill out CEInfo in the next call
irisPCADSC2 <- PCADSC(iris, "group", doCE = FALSE)
irisPCADSC2 <- doCE(irisPCADSC2)

# make a CE plot
CEPlot(irisPCADSC)
CEPlot(irisPCADSC2)

# End(Not run)

# Only do CE information and use less resamplings for a faster runtime
irisPCADSC_fast <- PCADSC(iris, "group", doAngle = FALSE, doChroma = FALSE, B = 1000)
CEPlot(irisPCADSC_fast)
```

chromaPlot  

**Chroma plot**

Description

Produce a chroma plot from a full or partial PCADSC object, as obtained from a call to PCADSC. In either case, this PCADSC object must have a non-NULL chromaInfo slot (see examples). The chroma plot compares the loading patterns from PCA conducted on two datasets consisting of different observations of the same variables.

Usage

```r
chromaPlot(x, varLabels = NULL, cvCO = 1, splitLabels = NULL,
           varAnnotation = "cum", useComps = NULL)
```

Arguments

- `x` Either a PCADSC object or a chromaInfo object, as produced by PCADSC and doChroma.
The plot consists of one display for each of the two datasets. The two displays both consist of a number of vertical bars. Each vertical bar represents a principal component and the width of each colored section (chroma) within the bar corresponds to the normalized PCA loading vector of that component. The bars can be annotated with the (cumulative) variance contributions of the components (see `varAnnotation`).

See Also

`PCADSC`, `doChroma`

Examples

```r
# load iris data
data(iris)

# Define grouping variable, grouping the observations by whether their species is Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

## Not run:
# make a PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group")

# make a partial PCADSC object from iris and fill out chromaInfo in the next call
irisPCADSC2 <- PCADSC(iris, "group", doChroma = FALSE)
irisPCADSC2 <- doChroma(irisPCADSC2)
```
doAngle

#make a chroma plot
cromaPlot(irisPCADSC)
cromaPlot(irisPCADSC)

#Change the labels of the splitting variable
cromaPlot(irisPCADSC, splitLabels = list("non-setosa" = "Not Setosa",
                                          "setosa" = "Setosa"))

#Only plot components 1 and 4 and remove annotated variances
cromaPlot(irisPCADSC, useComps = c(1,4), varAnnotation = "no")

#Only plot the first components responsible for explaining 80 percent variance
cromaPlot(irisPCADSC, cvCO = 0.8)

#Change variable labels
cromaPlot(irisPCADSC, varLabels = c("Sepal length", "Sepal width", "Petal length",
                                      "Petal width"))

## End(Not run)

#Only do chroma information in order to get a faster runtime:
irisPCADSC_fast <- PCADSC(iris, "group", doCE = FALSE,
                          doAngle = FALSE)
cromaPlot(irisPCADSC_fast)

---

doAngle  

Compute angle information

Description

Computes the information that is needed in order to make an anglePlot from a PCADSC or pcaRes object. Typically, this function is called on a partial PCADSC object in order to add angleInfo (see examples).

Usage

doAngle(x)

Arguments

  x        Either a PCADSC or a pcaRes object.

See Also

  anglePlot, PCADSC
Examples

```r
# load iris data
data(iris)

# Define grouping variable, grouping the observations by whether their species is
# Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

## Not run:
# make a partial PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group", doAngle = FALSE)

# No angleInfo available
irisPCADSC$angleInfo

# Add and show angleInfo
irisPCADSC <- doAngle(irisPCADSC)
irisPCADSC$angleInfo

## End(Not run)

# Make a partial PCADSC object and only add angle information for a
# faster runtime
irisPCADSC_fast <- PCADSC(iris, "group", doAngle = FALSE,
                           doChroma = FALSE, doCE = FALSE)
irisPCADSC_fast <- doAngle(irisPCADSC_fast)
irisPCADSC_fast$angleInfo
```

---

doCE

**Compute cumulative eigenvalue information**

**Description**

Computes the information that is needed in order to make a `CEPlot` from a PCADSC or pcaRes object. Typically, this function is called on a partial PCADSC object in order to add CEInfo (see examples).

**Usage**

```r
doCE(x, ...)
```

**Arguments**

- `x`        Either a PCADSC or a pcaRes object.
- `...`      If doCE is called on a pcaRes object, the full dataset must also be supplied (as data), as well as the number of resampling steps (B).
See Also

CEPlot, PCADSC

Examples

```r
# load iris data
data(iris)

# Define grouping variable, grouping the observations by whether their species is
# Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

## Not run:
# make a partial PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group", doCE = FALSE)

# No CEInfo available
irisPCADSC$CEInfo

# Add and show CEInfo
irisPCADSC <- doCE(irisPCADSC)
irisPCADSC$CEInfo

## End(Not run)

# Make a partial PCADSC object and only add CE information with no
# bootstrapping (and thus no test)
irisPCADSC_fast <- PCADSC(iris, "group", doAngle = FALSE,
    doChroma = FALSE, doCE = FALSE)
irisPCADSC_fast <- doCE(irisPCADSC_fast, B = 100)
irisPCADSC_fast$CEInfo
```

---

**doChroma**

*Compute chroma information*

**Description**

Computes the information that is needed in order to make a `chromaPlot` from a PCADSC or pcaRes object. Typically, this function is called on a partial PCADSC object in order to add `chromaInfo` (see examples).

**Usage**

`doChroma(x)`
Arguments

x Either a PCADSC or a pcaRes object.

See Also

chromaPlot, PCADSC

Examples

# load iris data
data(iris)

# Define grouping variable, grouping the observations by whether their species is
# Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

## Not run:
# make a partial PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group", doChroma = FALSE)

# No chromaInfo available
irisPCADSC$chromaInfo

# Add and show chromaInfo
irisPCADSC <- doChroma(irisPCADSC)
irisPCADSC$chromaInfo

## End(Not run)

# Make a partial PCADSC object and only add chroma information for a
# faster runtime
irisPCADSC_fast <- PCADSC(iris, "group", doAngle = FALSE,
doChroma = FALSE, doCE = FALSE)
irisPCADSC_fast <- doChroma(irisPCADSC_fast)
irisPCADSC_fast$chromaInfo

PCADSC

Compute the elements used for PCADSC

Description

Principal Component Analysis-based Data Structure Comparison tools that prepare a dataset for
various diagnostic plots for comparing data structures. More specifically, PCADSC performs PCA on
two subsets of a dataset in order to compare the structures of these datasets, e.g. to assess whether
they can be analyzed pooled or not. The results of the PCAs are then manipulated in various ways
and stored for easy plotting using the three PCADSC plotting tools, the CEPlot, the anglePlot and
the chromaPlot.
Usage

PCADSC(data, splitBy, vars = NULL, doCE = TRUE, doAngle = TRUE, doChroma = TRUE, B = 100000)

Arguments

data
A dataset, either a data.frame or a matrix with variables in columns and observations in rows. Note that tibbles and data.tables are accepted as input, but they are instantly converted to data.frames. Future releases might include specific implementation for these data representations.

splitBy
The name of a grouping variable with two levels defining the two groups within the dataset whose data structures we wish to compare.

vars
The variable names in data to include in the PCADSC. If NULL (the default), all variables except for splitBy are used.

doce
Logical. Should the cumulative eigenvalue plot information be computed?

doAngle
Logical. Should the angle plot information be computed?

dochroma
Logical. Should the chroma plot information be computed?

B
A positive integer. The number of resampling steps performed in the cumulative eigenvalue step, if relevant.

Details

PCADSC presents a suite of non-parametric, visual tools for comparing the structures of two subsets of a dataset. These tools are all based on PCA (principal component analysis) and thus they can be interpreted as comparisons of the covariance matrices of the two (sub)datasets. PCADSC performs PCA using singular value decomposition for increased numerical precision. Before performing PCA on the full dataset and the two subsets, all variables within each such dataset are standardized.

Value

An object of class PCADSC, which is a named list with the following entries:

pcaRes The results of the PCAs performed on the first subset, the second subset and the full subset and also information about the data splitting.

CEInfo The information needed for making a cumulative eigenvalue plot (see CEPlot).

angleInfo The information needed for making an angle plot (see anglePlot).

chromaInfo The information needed for making a chroma plot (see chromaPlot).

data The original (full) dataset.

splitBy The name of the variable that splits the dataset in two.

vars The names of the variables in the dataset that should be used for PCA.

B The number of resamplings performed for the CEInfo.

See Also

doCE, doAngle, doChroma, CEPlot, anglePlot, chromaPlot
Examples

```r
# load iris data
data(iris)

# Define grouping variable, grouping the observations by whether their species is
# Setosa or not
iris$group <- "setosa"
iris$group[iris$Species != "setosa"] <- "non-setosa"
iris$Species <- NULL

## Not run:
# Make a full PCADSC object, splitting the data by "group"
irisPCADSC <- PCADSC(iris, "group")

# The three plotting functions can now be called on irisPCADSC:
CEPlot(irisPCADSC)
anglePlot(irisPCADSC)
chromaPlot(irisPCADSC)

# Make a partial PCADSC object with no angle plot information and add
# angle plot information afterwards:
irisPCADSC2 <- PCADSC(iris, "group", doAngle = FALSE)
irisPCADSC2 <- doAngle(irisPCADSC)

## End (Not run)

# Make a partial PCADSC object with no plotting (angle/CE/chroma)
# information:
irisPCADSC_minimal <- PCADSC(iris, "group", doAngle = FALSE,
doCE = FALSE, doChroma = FALSE)
```
Index

anglePlot, 2, 6, 9, 10
CEPlot, 3, 7–10
chromaPlot, 4, 8–10
data.frame, 10
data.table, 10
doAngle, 2, 3, 6, 10
doCE, 3, 7, 10
doChroma, 4, 5, 8, 10
PCADSC, 2–6, 8, 9, 9
tibble, 10