Package 'SCOUTer'

October 12, 2022

Title Simulate Controlled Outliers

Version 1.0.0

Description Using principal component analysis as a base model, 'SCOUTer'
offers a new approach to simulate outliers in a simple and precise way.
The user can generate new observations defining them by a pair of well-known
statistics: the Squared Prediction Error (SPE) and the Hotelling's T^2 (T^2)
statistics. Just by introducing the target values of the SPE and T^2, 'SCOUTer'
returns a new set of observations with the desired target properties.
Authors: Alba González, Abel Folch-Fortuny, Francisco Arteaga and
Alberto Ferrer (2020).

License GPL-3

Encoding UTF-8

LazyData true

Maintainer Alba Gonzalez Cebrian <algonceb@upv.es>

RoxygenNote 7.1.1

Depends R (>= 3.5.0), ggplot2, ggpubr, stats

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Alba Gonzalez Cebrian [aut, cre],
Abel Folch-Fortuny [aut],
Francisco Arteaga [aut],
Alberto Ferrer [aut]

Repository CRAN

Date/Publication 2020-06-30 09:30:03 UTC

R topics documented:

---

barwithucl *barwithucl*

Description

Single bar plot with Upper Control Limis. Customized title and labels. Y-Axis limits are fixed according to the range of the values in x.

Usage

```
barwithucl(
  x,
  iobs,
  ucl,
  plotname = "",
  ylabelname = "",
  xlabelname = "Obs. Index"
)
```

Arguments

x	vector with the values of the statistic.
iobs	index of the observations whose value will be displayed.
ucl	Upper Control Limit of the statistic.
plotname	string with the title of the plot. Set to `""` by default.
ylabelname	string with the y-axis label. Set to `""` by default.
xlabelname	string with the y-axis label. Set to `"Obs. Index"` by default.

Value

ggplot object with the individual value of a variable as a geom_col with an horizontal line reference.

Examples

```
barwithucl(c(1:10), 6, 5)
barwithucl(c(1:10), 6, 5, plotname = "Plot title", ylabelname = "Y label",
xlabelname= "X label")
```

---

custombar   *custombar*

Description

Bar plot with customized title and labels. Y-Axis limits are fixed according to the range of the values in X.

Usage

```
custombar(X, iobs, plotname = "", ylabelname = "Contribution", xlabelname = "")
```

Arguments

X	matrix with observations as row vectors.
iobs	index of the observations whose value will be displayed.
plotname	string with the title of the plot. Set to "" by default.
ylabelname	string with the y-axis label. Set to "Contribution" by default.
xlabelname	string with the y-axis label. Set to "" by default.

Value

ggplot object with the values of a vector with a customized geom_col layer.

Examples

```
X <- as.matrix(X)
custombar(X, 2)
custombar(X, 2, plotname = "Observation 2", ylabelname = bquote(x.["j"]),
xlabelname= "Variables")
```

---

distplot   *distplot*

Description

Returns the distance plot providing a dataset and a Principal Component Analysis model.

Usage

```
distplot(
  X,
  pcaref,
  obstag = matrix(0, nrow(X), 1),
  plottitle = "Distance plot\n"
)
```

Arguments

X	data matrix with observations to be displayed in the distance plot.
pcaref	list with the information of the PCA model.
obstag	Optional column vector of integers indicating the group of each observation (0 or 1). Default value set to `matrix(0, nrow(X), 1)`.
plottitle	Optional string with the plot title. Set to `"Distance plot"` by default.

Details

Coordinates are expressed in terms of the Hotelling's T^2 (x-axis) and the Squared Prediction Error (y-axis) obtained projecting X on the provided model. Observations can be identified by the obstag input argument.

Value

ggplot object with the distance plot.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 2, 0.05, "cent")
distplot(X, pcamodel.ref)
tags <- dotag(X[1:40,], X[-c(1:40),])
distplot(X, pcamodel.ref, obstag = tags, plottitle = "D plot title")
```

---

distplotsimple   *displotsimple*

Description

Returns the distance plot directly providing the coordinates and Upper Control Limits.

Usage

```
distplotsimple(
  T2,
  SPE,
  lim.t2,
  lim.spe,
  ncomp,
  obstag = matrix(0, length(T2), 1),
  alpha = 0.05,
  plottitle = "Distance plot\n"
)
```

Arguments

T2	Vector with the Hotelling's T^2 values for each observation.
SPE	Vector with the SPE values for each observation.
lim.t2	Value of the Upper Control Limit for the T^2 statistic.
lim.spe	Value of the Upper Control Limit for the SPE.
ncomp	An integer indicating the number of PCs.
obstag	Optional column vector of integers indicating the group of each observation (0 or 1). Default value set to matrix(0, nrow(X), 1).
alpha	Optional number between 0 and 1 expressing the type I risk assumed in the computation of the Upper Control Limits (UCL) set to 0.05 (5 %) by default.
plottitle	Optional string with the plot title, "Distance plot" by default.

Details

Coordinates are expressed in terms of the Hotelling's T^2 (T^2, x-axis) and the Squared Prediction Error (SPE, y-axis). Observations can be identified by the obstag input argument.

Value

distplotobj ggplot object with the generated distance plot.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 2, 0.05, "cent") # PCA-MB with first 40
# observations
pcaproj <- pcame(X[-c(1:40),], pcamodel.ref) # Project last observations
distplotsimple(pcaproj$T2, pcaproj$SPE, pcamodel.ref$limt2, pcamodel.ref$limspe,
pcamodel.ref$ncomp)

pcaproj <- pcame(X, pcamodel.ref) # Project all observations
tags <- dotag(X[1:40,], X[-c(1:40),]) # 0's for observations used in PCA-MB
distplotsimple(pcaproj$T2, pcaproj$SPE, pcamodel.ref$limt2, pcamodel.ref$limspe,
pcamodel.ref$ncomp, obstag = tags)
```

---

dotag   *dotag*

Description

Returns the tag vector to identify two different data sets

Usage

```
dotag(X.zeros = NA, X.ones = NA)
```

Arguments

X.zeros	Matrix with the tag 0.
X.ones	Matrix with the tag 1.

Value

tag.all vector with 0 tags for observations in X.zeros and 1 tags for observations in X.ones.

Examples

```
X <- as.matrix(X)
dotag(X[1:40,], X[-c(1:40),])
```

---

dscplot	*dscplot*

Description

Returns the distance plot and the score plot providing a data matrix and a Principal Component Analysis (PCA) model. Observations can be identified by the obstag input argument.

Usage

```
dscplot(
  X,
  pcamodel,
  obstag = matrix(0, nrow(X), 1),
  pcx = 1,
  pcy = 2,
  alpha = 0.05,
  nrow = 1,
  ncol = 2,
  legpos = "bottom"
)
```

Arguments

X	Matrix with the data to be displayed.
pcamodel	List with the PCA model elements.
obstag	Optional column vector of integers indicating the group of each observation (0 or 1). Default value set to matrix(0, nrow(X), 1).
pcx	Optional integer with the number of the PC in the horizontal axis. Set to 1 by default.
pcy	Optional integer with the number of the PC in the vertical axis. Set to 2 by default.
alpha	Optional number between 0 and 1 expressing the type I risk assumed in the computation of the confidence ellipse, set to 0.05 (5 %) by default.
nrow	Optional number of rows the plot layout. Set to 1 by default.
ncol	Optional number of columns the plot layout. Set to 2 by default.
legpos	Optional string with the position of the legend. Set to "bottom" by default.

Value

ggplot object with the generated score plot.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 3, 0.05, "cent")
dscplot(X, pcamodel.ref)
dscplot(X, pcamodel.ref, nrow = 2, ncol = 1)
tags <- dotag(X[1:40,], X[-c(1:40),])
dscplot(X, pcamodel.ref, obstag = tags, pcy = 3)
```

---

ht2info   *ht2info*

Description

Returns information about the Hotelling's T^2 statistic for an observation. Two subplots show the information of an observation regarding its T^2 statistic, i.e.: a bar plot indicating the value of the statistic for the observation, and a bar plot with the contribution that each component had for the T^2 value. The term T^2_A makes reference to the T^2 for a model with A principal components (PCs).

Usage

```
ht2info(HT2, T2matrix, limht2, iobs = NA)
```

Arguments

HT2	A vector with values of the Hotelling's T^2_A statistic.
T2matrix	A matrix with the contributions of each PC (A columns) for each observation (rows) to the Hotelling's T^2_A statistic.
limht2	Upper Control Limit for the Hotelling's T^2_A statistic, at a certain confidence level (1-alpha)*100 %.
iobs	Integer with the index of the observation of interest. Default value set to NA.

Value

ggplot object with the generated bar plots.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 2, 0.05, "cent") # PCA-MB with first 40
# observations
pcaproj <- pcame(X[-c(1:40),], pcamodel.ref) # Project last observations
ht2info(pcaproj$T2, pcaproj$T2matrix, pcamodel.ref$limt2, 2) # Information about
# the T^2 of the row #2
```

---

obscontribpanel   *obscontribpanel*

Description

Information about the Hotelling's T^2 and the Squared Predidiction Error (SPE) of an observation. The term T^2_A makes reference to the T^2 for a model with A principal components (PCs).

Usage

```
obscontribpanel(pcax, pcaref, obsid = NA)
```

Arguments

pcax	A list with the elements of the PCA model that will be displayed: SPE, T^2_A and their constributions (E and T2matrix).
pcaref	A list with the PCA model according to which the distance and contributions are expressed.
obsid	Integer with the index of the observation of interest. Default set to NA.

Value

ggplot object with the generated bar plots in a 1 x 4 subplots layout.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 2, 0.05, "cent") # PCA-MB with first
# 40 observations
pcaproj <- pcame(X[-c(1:40),], pcamodel.ref) # Project last observations
obscontribpanel(pcaproj, pcamodel.ref, 2) # Information about the SPE and T^2
# of the row #2
```

---

pcamb_classic   *pcamb_classic*

Description

Principal Component Analysis (PCA) model fitting according to a matrix X using singular value decomposition (svd)

Usage

```
pcamb_classic(X, ncomp, alpha, prepro)
```

Arguments

X	Matrix with observations that will used to fit the PCA model.
ncomp	An integer indicating the number of PCs that the model will have.
alpha	A number between 0 and 1 indicating the type I risk assumed to calculate the Upper Control Limits (UCLs) for the Squared Prediction Error (SPE), the Hotelling's T^2_A and the scores. The confidence level of these limits will be (1-alpha)*100.
prepro	A string indicating the preprocessing to be performed on X. Its possible values are: "none", for any preprocessing, "cent", for a mean-centering, or "autosc", for a mean-centering and unitary variance scaling (autoscaling).

Value

list with elements containing information about PCA model:

- m: mean vector.

- s: standard deviation vector.

- P: loading matrix with the loadings of each PC stored as columns.

- Pfull: full loading matrix obtained by the svd,

- lambda: vector with the variance of each PC.

- limspe: Upper Control Limit for the SPE with a confidence level (1-alpha)*100 %.

- limt2: Upper Control Limit for the T^2_A with a confidence level (1-alpha)*100 %.

- limits_t: Upper control Limits for the scores with a confidence level (1-alpha)*100 %.

- prepro: string indicating the type of preprocessing performed on X.

- ncomp: number of PCs of the PCA model, A.

- alpha: value of the type I risk assumed to calculate the Upper Control Limits of the SPE, T^2_A and scores.

- n: dimension of the number of rows in X.

- S: covariance matrix of X.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
pcamodel.ref <- pcamb_classic(X[1:40,], 2, 0.05, "cent") # PCA-MB with first 40
# observations
```

---

pcame	*pcame*

Description

Projection of X onto a Principal Component Analysis (PCA) model.

Usage

```
pcame(X, pcaref)
```

Arguments

X   Matrix with observations that will be projected onto the PCA model.

pcaref   A list with the elemements of a PCA model:

- m: mean.
- s: standard deviation.
- prepro: preprocessing: "none", "cent" or "autosc".
- P: loading matrix.
- lambda: vector with variances of each PC.

Details

pcame performs the projection of the data in X onto the PCA model stored as a list of parameters. It returns the projection of the observations in X, along with the Squared Prediction Errors (SPE), Hotelling's T^2_A, contribution elements and the reconstruction of X obtained by the PCA model.

Value

list with elements containing information about X in the PCA model:

- Xpreprocessed: matrix X preprocessed.
- Tscores: score matrix with the projection of X on each one of the A PCs.
- E: error matrix with the par of X not explained by the PCA model.
- SPE: vector with the SPE for each observation of X.
- T2: vector with the T^_A for each observation of X.
- T2matrix: matrix with the contributions of each PC to the T^2_A for each observation of X.
- Xrec: matrix with the reconstructed part of X, i.e. the part of X explained by the PCA model.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
pcame(X, pcamodel.ref) # Project all observations onto PCA model of pcamodel.ref

pcamodel.ref <- pcamb_classic(X[1:40,], 2, 0.05, "cent") # PCA-MB with first 40 observations
pcame(X[-c(1:40),], pcamodel.ref) # Project observations not used in PCA-MB onto PCA model
# of pcamodel.ref
```

---

scoreplot   *scoreplot*

Description

Returns the score plot providing a dataset and a pca model. Observations can be identified by the obstag input argument.

Usage

```
scoreplot(
  X,
  pcamodel,
  obstag = matrix(0, nrow(X), 1),
  pcx = 1,
  pcy = 2,
  alpha = 0.05,
  plottitle = "Score plot\n"
)
```

Arguments

X	Matrix with the data to be displayed.
pcamodel	List wiht the PCA model elements.
obstag	Optional column vector of integers indicating the group of each observation (0 or 1). Default value set to `matrix(0, nrow(X), 1)`.
pcx	Optional integer with the number of the PC in the horizontal axis. Set to 1 by default.
pcy	Optional integer with the number of the PC in the vertical axis. Set to 2 by default.
alpha	Optional number between 0 and 1 expressing the type I risk assumed in the compuatation of the confidence ellipse, set to `0.05` (5 %) by default.
plottitle	Optional string with the plot title. Set to `"Score plot"` by default.

Value

ggplot object with the generated score plot.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 3, 0.05, "cent")
scoreplot(X, pcamodel.ref)
tags <- dotag(X[1:40,], X[-c(1:40),])
scoreplot(X, pcamodel.ref, obstag = tags, pcx = 2, pcy = 3, alpha = 0.1, plottitle = "T-plot")
```

---

scoreplotsimple *scoreplotsimple*

Description

Returns the score plot providing the scores matrix, **T**. Observations can be identified by the obstag input argument.

Usage

```
scoreplotsimple(
  Tscores,
  pcx = 1,
  pcy = 2,
  obstag = matrix(0, nrow(Tscores), 1),
  alpha = 0.05,
  varT = stats::var(Tscores),
  plottitle = "Score plot\n"
)
```

Arguments

Tscores	Matrix with the scores to be displayed, with the information of each Principal Component (PC) stored by columns.
pcx	Optional integer with the number of the PC in the horizontal axis. Set to 1 by default.
pcy	Optional integer with the number of the PC in the vertical axis. Set to 2 by default.
obstag	Optional column vector of integers indicating the group of each observation (0 or 1). Default value set to matrix(0, nrow(X), 1).
alpha	Optional number between 0 and 1 expressing the type I risk assumed in the computation of the confidence ellipse, set to 0.05 (5 %) by default.
varT	Optional parameter expressing the variance of each PC. Set to var(Tscores) by default.
plottitle	Optional string with the plot title. Set to "Score plot" by default.

Value

ggplot object with the generated score plot.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 3, 0.05, "cent")
pcaproj <- pcame(X, pcamodel.ref) # Project last observations
scoreplotsimple(pcaproj$Tscores)

pcaproj <- pcame(X[-c(1:40),], pcamodel.ref) # Project all observations
tags <- dotag(X[1:40,], X[-c(1:40),]) # 0's for observations used in PCA-MB
scoreplotsimple(pcaproj$Tscores, pcx = 2, pcy = 3, obstag = tags)
```

---

scout	*scout*

Description

Shift of an observation following a selected pattern.

Usage

```
scout(
  X,
  pcaref,
  T2.y = NA,
  SPE.y = NA,
  nsteps = 1,
  nsteps.spe = 1,
  nsteps.t2 = 1,
  gspe = 1,
  gt2 = 1,
  mode = "simple"
)
```

Arguments

X	Matrix with observations that will be shifted as rows.
pcaref	List with the elements of a PCA model:

- m: mean.
- s: standard deviation.
- prepro: preprocessing: "none", "cent" or "autosc".
- P: loading matrix.
- lambda: vector with variances of each PC.

T2.y	A number indicating the target value for the Hotelling's T^2_A after the shift. Set to NA by default.
SPE.y	A number indicating the target value for the Squared Prediction Error after the shift. Set to NA by default.

nsteps	A number indicating the number of steps between the reference and target values of the SPE and the T^2. Set to 1 by default.
nsteps.spe	An integer indicating the number of steps in which the shift from the reference to the target value of the SPE will be performed. Set to 1 by default.
nsteps.t2	An integer indicating the number of steps in which the shift from the reference to the target value of the T^2_A will be performed. Set to 1 by default.
gspe	A number indicating the term that will tune the spacing between steps for the SPE. Set to 1 by default (linear spacing).
gt2	A number indicating the term that will tune the spacing between steps for the SPE. Set to 1 by default (linear spacing).
mode	A character indicating the type of shift that will be performed: "simple", "steps" or "grid".

Value

list with elements:

- X: matrix with the new and shifted data.
- SPE: SPE of each one of the generated outliers in the list element X.
- T2: T^2 of each one of the generated outliers in the list element X.
- step.spe: step of each observation according to the shift of the SPE.
- step.t2: step of each observation according to the shift of the T^2.
- tag: is a vector of ones as long as the number of generated observations.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
# Shift the first observation:
outscout <- scout(X[1,], pcamodel.ref, T2.y = 40, SPE.y = 50, nsteps.spe = 3, nsteps.t2 = 2,
gspe = 3, gt2 = 0.5, mode = "grid")

# Shift a set of observations increasing only the T^2 in one step:
outscout <- scout(X, pcamodel.ref, T2.y = matrix(40, nrow(X), 1), mode = "simple")
```

---

scoutgrid	*scoutgrid*

Description

Shift of an array following a grid pattern.

Usage

```
scoutgrid(
  X,
  pcaref,
  T2.target = NA,
  SPE.target = NA,
  nsteps.t2 = 1,
  nsteps.spe = 1,
  gspe = 1,
  gt2 = 1
)
```

Arguments

X	Matrix with observations that will be shifted as rows.
pcaref	List with the elements of a PCA model:

- m: mean.
- s: standard deviation.
- prepro: preprocessing: ″none″, ″cent″ or ″autosc″.
- P: loading matrix.
- lambda: vector with variances of each PC.

T2.target	A number indicating the target value for the T^2_A after the shift. Set to NA by default.
SPE.target	A number indicating the target value for the SPE after the shift. Set to NA by default.
nsteps.t2	An integer indicating the number of steps in which the shift from the reference to the target value of the T^2_A will be performed. Set to 1 by default.
nsteps.spe	An integer indicating the number of steps in which the shift from the reference to the target value of the SPE will be performed. Set to 1 by default.
gspe	A number indicating the term that will tune the spacing between steps for the SPE. Set to 1 by default (linear spacing).
gt2	A number indicating the term that will tune the spacing between steps for the SPE. Set to 1 by default (linear spacing).

Value

list with elements:

- X: matrix with the new and shifted data.
- SPE: SPE of each one of the generated outliers in the list element X.
- T2: T^2 of each one of the generated outliers in the list element X.
- step.spe: step of each observation according to the shift of the SPE.
- step.t2: step of each observation according to the shift of the T^2.
- tag: is a vector of ones as long as the number of generated observations.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
# Shift a set of observations increasing the T^2  and the SPE in 3 and 2 linear and
# non-linear steps respectively:
outgrid <- scoutgrid(X, pcamodel.ref, T2.target = matrix(40, nrow(X), 1),
SPE.target = matrix(50, nrow(X), 1), nsteps.t2 = 3, nsteps.spe = 2, gspe = 4)
```

---

scoutsimple	*scoutsimple*

Description

Shift of an array with a single step.

Usage

```
scoutsimple(X, pcaref, T2.target = NA, SPE.target = NA)
```

Arguments

X	Matrix with observations that will be shifted as rows.
pcaref	List with the elements of a PCA model:
	- m: mean. - s: standard deviation. - prepro: preprocessing: "none", "cent" or "autosc". - P: loading matrix. - lambda: vector with variances of each PC.
T2.target	A number indicating the target value for the T^2_A after the shift. Set to NA by default.
SPE.target	A number indicating the target value for the SPE after the shift. Set to NA by default.

Value

list with elements:

- X: matrix with the new and shifted data.
- SPE: SPE of each one of the generated outliers in the list element X.
- T2: T^2 of each one of the generated outliers in the list element X.
- step.spe: step of each observation according to the shift of the SPE.
- step.t2: step of each observation according to the shift of the T^2.
- tag: is a vector of ones as long as the number of generated observations.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
# Shift a set of observations increasing only the T^2 in one step:
outsimple <- scoutsimple(X, pcamodel.ref, T2.target = matrix(40, nrow(X), 1))
```

---

scoutsteps	*scoutsteps*

Description

Shift of an array following a step-wise pattern.

Usage

```
scoutsteps(
  X,
  pcaref,
  T2.target = NA,
  SPE.target = NA,
  nsteps = 1,
  gspe = 1,
  gt2 = 1
)
```

Arguments

X	Matrix with observations that will be shifted as rows.
pcaref	List with the elements of a PCA model:
	- m: mean. - s: standard deviation. - prepro: preprocessing: "none", "cent" or "autosc". - P: loading matrix. - lambda: vector with variances of each PC.
T2.target	A number indicating the target value for the Hotelling's T^2_A after the shift. Set to NA by default.
SPE.target	A number indicating the target value for the Squared Prediction Error after the shift. Set to NA by default.
nsteps	A number indicating the number of steps between the reference and target values of the SPE and the T^2. Set to 1 by default.
gspe	A number indicating the term that will tune the spacing between steps for the SPE. Set to 1 by default (linear spacing).
gt2	A number indicating the term that will tune the spacing between steps for the SPE. Set to 1 by default (linear spacing).

Value

list with elements:

- X: matrix with the new and shifted data.

- SPE: SPE of each one of the generated outliers in the list element X.

- T2: T^2 of each one of the generated outliers in the list element X.

- step.spe: step of each observation according to the shift of the SPE.

- step.t2: step of each observation according to the shift of the T^2.

- tag: is a vector of ones as long as the number of generated observations.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
# Shift a set of observations increasing the T^2  and the SPE in 4 linear steps:
outsteps <- scoutsteps(X, pcamodel.ref, T2.target = matrix(40, nrow(X), 1),
SPE.target = matrix(50, nrow(X), 1), nsteps = 4)
# Shift a set of observations increasing the SPE in 4 non-linear steps:
outsteps <- scoutsteps(X, pcamodel.ref, SPE.target = matrix(50, nrow(X), 1), nsteps = 4,
gspe = 0.3)
```

---

speinfo	*speinfo*

Description

Information about the Squared Prediction Error (SPE) of an observation. Two subplots show the information of an observation regarding its SPE statistic, i.e.: a bar plot indicating the value of the statistic for the observation, and a bar plot with the contribution that each variable had for the SPE value

Usage

```
speinfo(SPE, E, limspe, iobs = NA)
```

Arguments

SPE	Vector with values of the SPE statistic.
E	Matrix with the contributions of each variable (columns) for each observation (rows) to the SPE. It is the error term obtained from the unexplained part of X by the PCA model.
limspe	Upper Control Limit (UCL) for the SPE, at a certain confidence level (1-alpha)*100 %.
iobs	Integer with the index of the observation of interest. Default value set to NA.

Value

ggplot object with the generated bar plots.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X[1:40,], 2, 0.05, "cent") # PCA-MB with first 40 observations
pcaproj <- pcame(X[-c(1:40),], pcamodel.ref) # Project last observations
speinfo(pcaproj$SPE, pcaproj$E, pcamodel.ref$limspe, 2) # Information about the SPE of the
# row #2
```

X	*Demo dataset*

Description

It is a small data set to use as a demo for the SCOUTer package. It consists of normally distributed variables, with two Principal Components explaining an 80% of the total variance.

Usage

```
X
```

Format

A matrix data frame with 50 rows and 5 normally distributed variables.

---

xshift	*xshift*

Description

Shift of an observation. The performed operation results as a combination of two main directions: the direction of maximum gradient for the SPE (weighted by the parameter b) and the direction of the projection of the observation on the model (weighted by the parameter a).

Usage

```
xshift(X, P, a, b)
```

Arguments

X	Matrix with observations that will be shifted.
P	Loading matrix of the PCA model according to which the shift will be performed.
a	A number or vector tuning the shift in the direction of its projection.
b	A number or vector tuning the shift in the direction of its residual.

Value

Matrix with shifted observation as rows, keeping the order of the input matrix X.

Examples

```
X <- as.matrix(X)
pcamodel.ref <- pcamb_classic(X, 3, 0.1, "autosc") # PCA-MB with all observations
# Shift observation #10 increasing by a factor of 2 and 4 its T^2 and its SPE respectively
x.new <- xshift(X[10,], pcamodel.ref$P, sqrt(2) - 1, sqrt(4) - 1)
```

Index