Package 'mbgraphic'

May 13, 2017

Type Package

Title Measure Based Graphic Selection

Version 1.0.0

Date 2017-05-13

Author Katrin Grimm

Maintainer Katrin Grimm <katrin-grimm@web.de>

VignetteBuilder knitr

Description Measure based exploratory data analysis. Some of the functions call interactive apps programmed with the package shiny to provide flexible selection options.

Depends R (>= 3.1)

Imports mgcv, extracat, scagnostics, diptest, hexbin,scales, dplyr, magrittr, shiny, seriation, ggplot2, gridExtra, GGally, Rcpp, stats, graphics, utils, grDevices

Suggests knitr, rmarkdown, png

LinkingTo Rcpp

LazyLoad yes

LazyData yes

License GPL (>= 2)

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-05-13 10:15:21 UTC

R topics documented:

`binning` *Binning of single variables or data frames*

Description

Bins a single vector, a pair of vectors, a matrix or a data frame by equidistant or quantile based binning.

Usage

```
binning(x, y = NULL, b = 20, bin = "equi", anchor = "min")
```

Arguments

`x`	A numeric vector, a numeric matrix or a data frame. In case of a data frame only the numeric variables are used.
`y`	`NULL` (default) or a vector with same length as `x`.
`b`	A positive integer. Number of bins in each variable.
`bin`	A character string giving the binning method. "`equi`" (default) for equdistant binning or "`quant`" for quantile based binning.
`anchor`	A chraracter string or a numeric. How should the anchor point be chosen? "`min`" (default) for the minimum of each variable, "`ggplot`" for the method used in ggplot graphics, "`nice`" for a "pretty"" anchorpoint, or a user specified value.

Value

A data frame giving the mean of each bin and the numbers of counts.

Author(s)

Katrin Grimm

See Also

binningplot

Examples

```
df <- data.frame(
  x = rnorm(1000),
  y = rnorm(1000),
  z = rnorm(1000))

# A common binning for all variables of df with 5 bins in each dimension:
binning(df,b=5)
```

---

binningplot *Two-dimensional binningplot*

Description

Visualisation of a two-dimensional binning based on equidistant or quantile based binning.

Usage

```
binningplot(x, y, b = 10, bin = "equi", anchor = "min")
```

Arguments

x	A numeric vector.
y	A numeric vector.
b	A positive integer. Number of bins in each variable.
bin	A chraracter string. Binning method "equi" (default) for equidistant binning or "quant" for quantile binning.
anchor	A chraracter string or a numeric. How should the anchorpoint be chosen? "min" (default) for the minimum of each variable, "ggplot" for the method used in ggplot graphics, "nice" for a "pretty"" anchor point, or a user specified value.

Value

A ggplot object.

Author(s)

Katrin Grimm

References

H. Wickham (2009) *ggplot2: Elegant Graphics for Data Analysis* New York: Springer

See Also

binning

Examples

```
x <- rnorm(10000)
y <- rnorm(10000)

# equidistant binning with 20 bins in each variable
binningplot(x,y,b=20)

# quantile based binning with 20 bins in each variable
binningplot(x,y,b=20,bin="quant")
```

---

datatype *Description of the datatype of variables*

Description

Simple calculations to describe the datatype of variables and quantify the degree of discreteness.

Usage

```
datatype(x, limit = 0.8)
```

Arguments

x	A vector, a numeric matrix or a data frame.
limit	The limit used to define if a variable is treated as discrete or continous, depending on the value of discrete1d.

Value

A data frame with the following variables:

numeric	A logical value. Is variable of mode "numeric"?
discrete	Value of function discrete1d.
valdisc	A logical value. Is variable discrete under the given definition of discreteness?

Author(s)

Katrin Grimm

See Also

discrete1d

Examples

```
data(Election2005)
datatype(Election2005[,1:10])
```

---

dcor2d *Distance correlation for pairs of variables*

Description

Calculates the bivariate distance correlation for a given pair of variables, a numeric matrix or a data frame.

Usage

```
dcor2d(x, y = NULL, binning = FALSE, b = 50, anchor = "min",parallel=FALSE)
```

Arguments

x	A numeric vector, a numeric matrix or a data frame. In case of a data frame only the numeric variables are used.
y	A numeric vector.
binning	A logical value. Whether or not binning should be used. TRUE, "equi" for equidistant binng, "quant" for quantile based binning or "hexb" for hexagonal binning. Default is FALSE.
b	A positive integer. Number of bins in each variable.
anchor	A chraracter string or a numeric value. How should the anchor point be chosen? "min" (default) for the minimum of each variable, "ggplot" for the method used in ggplot graphics, "nice" for a "pretty"" anchorpoint, or a user specified value.
parallel	A logical value. Whether or not parallelization should be used. Default is FALSE.

Value

A numeric value describing the value of the measure if a pair of vectors is given. Otherwise a data frame with the following variables:

splines2d	Value of the measure.
x1	Number of first variable
x2	Number of second variable.
nx1	Name of first variable (missing if x is not a data frame).
nx2	Name of second variable (missing if x is not a data frame).

Author(s)

Katrin Grimm

References

G. J. Szekely, M. L. Rizzo und N. K. Bakirov (2007) Measuring and testing dependence by correlation of distances.*The Annals of Statistics* **35**(6) 2769–2794.

A. Pilhoefer und A. Unwin (2013) New Approaches in Visualization of Categorical Data: R Package extracat *Journal of Statistical Software* **53**(1) 1–25.

See Also

wdcor in package **extracat**, splines2d

Examples

```
data(Election2005)
## Not run:
# distance correlation for all pairs of variables
dcor <- dcor2d(Election2005)
# put the pairs in decreasing order
o_dcor <- dcor[order(dcor$dcor2d,decreasing=TRUE),]

# Show the 10 pairs with highest values
o_dcor[1:10,]

# Show the 4 scatterplots with highest values
par(mfrow=c(2,2))
for(i in 1:4){
plot(with(Election2005,get(as.character(o_dcor$nx1[i]))),
  with(Election2005,get(as.character(o_dcor$nx2[i]))),
  xlab=paste(o_dcor$nx1[i]),ylab=paste(o_dcor$nx2[i]),pch=19)
}

## End(Not run)
```

discrete1d *Measure for the degree of discreteness*

Description

A simple measure for the degree of discreteness of a variable, of the columns of a numeric matrix or of the variables of a data frame based on the number of unique values.

Usage

```
discrete1d(x)
```

Arguments

x	A numeric vector, a numeric matrix or a data frame.

Value

A single value or a vector including the results of `discrete1d` for each variable (in case of numeric matrices or data frames). If variables from data frames are categorical, 'NA' is returned.

Author(s)

Katrin Grimm

See Also

`multimod1d`, `skew1d`, `iaunivariate`

Examples

```
data(Election2005)

discrete <- discrete1d(Election2005)
maxv <- order(discrete,decreasing=TRUE)[1:4]
par(mfrow=c(2,2))
for(i in 1:4){
  hist(Election2005[,maxv[i]], ,xlab="",breaks=30,
  main=paste(names(Election2005[maxv[i]])))
}

## Not run:
# Explore skewness, multimodality and discreteness
# within an interactive environment.
iaunivariate(Election2005)

## End(Not run)
```

`Election2005` *Election2005 data*

Description

A data set for the German 'Bundestag' election of 2005. It includes information about the elections in 2005 and in 2002 separately for each of the 299 constituencies and also demographic and other information about the constituencies themselves.

Usage

```
data(Election2005)
```

Format

A data frame with 299 observations on 70 variables. The variables of the data set are:

- 1-4: general information about the constituencies (ID and name of district, state)
- 5-40: demographic and economic information
- 41 - 48: general information about the elections in 2005 and 2002
  - `WBerechE`: number of eligible voters
  - `WE`: votes cast
  - `UngZE`: invalid second preferences
  - `GulZE`: valid second preferences
- 49- 70: results of the five biggest parties in 2005 and 2002 (smaller parties are summarized in the variable `Rest`). Everything with a 'V' or 'v' at the end is from the election in 2005. 'ze' and 'zv' at the end refer to second preference votes.

Source

Original source was http://www.bundeswahlleiter.de.

See Also

Election2013

Examples

```
dim(Election2005)
datatype(Election2005)
str(Election2005)
```

`Election2013` *Election2013 data*

Description

A data set for the German 'Bundestag' election of 2013. It includes information about the elections in 2013 and in 2009 separately for each of the 299 constituencies and also demographic and other information about the constituencies themselves.

Usage

```
data(Election2013)
```

Format

A data frame with 299 observations on 114 variables. The variables of the data set are:

- 1-3: general information about the constituencies (Bundesland, ID and name of district)
- 4-42: demographic and economic information
- 43 - 50: general information about the elections in 2013 and 2009
  - `Wahlberechtigte`: number of eligible voters
  - `Waehler`: votes cast
  - `UNG_ZS`: invalid second votes
  - `GUEL_ZS`: valid second votes
- 51- 70: results of the seven biggest parties in 2013 and 2009 (smaller parties are summarized in the variable `Sonstige_ZS`). Everything with a '_VP' at the end is from the election in 2009. '_q' stands for proportion of second votes.

Source

Original source was http://www.bundeswahlleiter.de.

See Also

Election2013

In comparison to the **Election2005** data, there are more smaller parties reported. That produces more variables and more missing values (not all of the parties stand for election in all of the constituencies).

Examples

```
dim(Election2013)
datatype(Election2013)
str(Election2013)
```

groups *Quantifying the influence of a factor variable*

Description

The function calculates a linear model by calling `lm(xi~classvar)` for all numeric variables (`xi`) from `x` and returns the coefficients of determination. The aim is to find numeric variables for which high ratios of the variability can be explained by a factor variable `classvar`.

Usage

```
groups(data, classvar)
```

Arguments

`data`	A data frame.
`classvar`	A factor. The variable which is used to explain the numeric variables from `x`.

Value

If `x` is a numeric vector, the coefficient of determination of the model described by `lm(x~classvar)` is returned. Otherwise a data frame with the following two variables:

`groups`	coefficient of determination.
`variable`	actual variable.

Author(s)

Katrin Grimm

See Also

`groups_maxplot`, `lm`

Examples

```
data(Election2005)

# Define new variable which explains affiliation of the constituencies
# to east Germany, west Germany and Berlin
OstWest <-  ifelse(Election2005$Land %in%
    c("Thueringen","Sachsen","Sachsen-Anhalt",
    "Brandenburg","Mecklenburg-Vorpommern"),  "Ost", "West")
Election2005$OstWestBerlin <-
  as.factor(ifelse(Election2005$Land == "Berlin","Berlin",OstWest))

# Calculate measure groups
groupm <- groups(Election2005, "OstWestBerlin")
# Show highest values
groupm[with(groupm,order(groups,decreasing=TRUE))[1:10],]
```

groups_maxplot *Plots the variables with highest values in groups*

Description

Plots the variables with highest values in groups within histograms.

Usage

```
groups_maxplot(data, classvar, m = 5, samebinsize = FALSE)
```

Arguments

data	A data frame.
classvar	A factor. The variable which is used to explain the numeric variables from x.
m	A positive integer. Number of plots to draw. Maximum is 10.
samebinsize	Logical. Whether or not bins are drawn with the same size. Default is FALSE.

Value

A ggplot object.

Note

You can only use factors with up to 7 different values.

Author(s)

Katrin Grimm

References

H. Wickham (2009) *ggplot2: Elegant Graphics for Data Analysis* New York: Springer

See Also

groups

Examples

```
data(Election2005)

# Define new variable which explains affiliation of the constituencies
# to east Germany, west Germany and Berlin
OstWest <-  ifelse(Election2005$Land %in%
    c("Thueringen","Sachsen","Sachsen-Anhalt",
    "Brandenburg","Mecklenburg-Vorpommern"),  "Ost", "West")
Election2005$OstWestBerlin <-
```

```
  as.factor(ifelse(Election2005$Land == "Berlin","Berlin",OstWest))

# Plot 5 variables with highest values in groups
groups_maxplot(Election2005,"OstWestBerlin")
```

`iacorrgram` *Interactive corrgrams*

Description

Generates an interactive corrgram programmed with the package **shiny**. It uses the package **seriation** for reordering, optimal leaf ordering ('OLO') and reordering based on principal components ('PCA'). If 'OLO' is selected, cluster lines can be drawn by selecting the number of clusters or a minimum correlation (see `varclust` for details). A range of correlation can be chosen, where only pairs of variables with absolute correlation within the range are drawn with colored cells, otherwise the cells are colored white.

Usage

```
iacorrgram(data)
```

Arguments

`data`	A data frame.

Value

A shiny app object.

Author(s)

Katrin Grimm

References

M. Friendly (2002) Corrgrams: Exploratory displays for correlation matrices *The American Statistician* **56**(4), 316–324.

W. Chang, J. Cheng, J. Allaire, Y. Xie and J. McPherson (2016) shiny: Web Application Framework for R. https://cran.r-project.org/package=shiny.

See Also

`varclust`

Examples

```
## Not run:
data(Election2005)
iacorrgram(Election2005)

## End(Not run)
```

`iascaggram` *Interactive scaggrams*

Description

An interactive version of the scaggrams from function `scaggram` programmed with the package **shiny**. Selections within the scaggram can be made by clicking or brushing. A selection implies drawing the corresponding scatterplot(s).

Usage

```
iascaggram(sdfdata)
```

Arguments

`sdfdata`	A list of class `"sdfdata"`.

Details

Reordering is done with the functions `sdf_sort` (options 'Use all scagnostics' and 'Use only chosen scagnostics') or `sdf_quicksort` (option 'Quick'). The glyphs drawn by choosing option 'add -> glyphs' represent all given scagnostics.

Value

A shiny app object.

Note

The scagnostics need to lie between 0 and 1.

`sdf_sort` can be very slow for data sets with a huge number of variables. Use Option 'Quick' in this case for a fast reordering or use `sdf_sort` (with a time limit if necessary) before starting the interactive app.

Adding scatterplots or glyphs is only advisable for a small number of variables.

Author(s)

Katrin Grimm

References

W. Chang, J. Cheng, J. Allaire, Y. Xie and J. McPherson (2016) shiny: Web Application Framework for R. https://cran.r-project.org/package=shiny.

See Also

scaggram, sdf, scag2sdf, sdf_sort, sdf_quicksort

Examples

```
## Not run:
data(Election2005)
# some demographic/economic variables
sdfdata1 <- sdf(Election2005[,5:40])
iascaggram(sdfdata1)

# another smaller subset (for example to explore the glyphs in the scaggram)
sdfdata2 <- sdf(Election2005[,5:19])
iascaggram(sdfdata2)

## End(Not run)
```

---

iascagpcp *Interactive parallel coordinate plots for exploring scagnostics results*

Description

An interactive parallel coordinate plot for exploring scagnostics results programmed with the package **shiny**. If `sdfdata` is generated by function `sdf`, *Outliers* and *Exemplars* can be explored separately. Selections within the parallel coordinate plot can be made by drawing boxes on the axes around the chosen line.

Usage

```
iascagpcp(sdfdata)
```

Arguments

`sdfdata`	A list of class `"sdfdata"`.

Details

For scaling the three options 'std' (every scagnostic individually by subtraction of mean and division by standard deviation), 'uniminmax' (every scagnostic individually to unit interval) and 'globalminmax' (no scaling) can be used. See also `ggparcoord`.

Value

A shiny app object.

Author(s)

Katrin Grimm

References

W. Chang, J. Cheng, J. Allaire, Y. Xie and J. McPherson (2016) shiny: Web Application Framework for R. https://cran.r-project.org/package=shiny.

B. Schloerke et al. (2016) GGally: Extension to ggplot2. https://cran.r-project.org/package=GGally

See Also

sdf, scag2sdf

Examples

```
## Not run:
data(Election2005)
# some demographic/economic variables
sdfdata <- sdf(Election2005[,5:40])
iascagpcp(sdfdata)

## End(Not run)
```

---

iaunivariate *Interactive app for exploring univariate anomalies*

Description

Generates an interactive app programmed with the package **shiny**. It uses the functions discrete1d, skew1d and multimod1d and displays them in histograms.

Which variables from the data set are plotted can be decided interactively by selecting bins within the histograms of the measures. Character variables from the data set can be used for highlighting categories in the plots of the variables.

Usage

```
iaunivariate(data, exp = 1, seed = NULL)
```

Arguments

data	A data frame.
exp	A positive integer. See multimod1d.
seed	An integer. See multimod1d.

Value

A shiny app object.

Author(s)

Katrin Grimm

References

W. Chang, J. Cheng, J. Allaire, Y. Xie and J. McPherson (2016) shiny: Web Application Framework for R. https://cran.r-project.org/package=shiny.

See Also

discrete1d, skew1d and multimod1d

Examples

```
## Not run:
data(Electon2005)
iaunivariate(Election2005)

## End(Not run)
```

---

mbgraphic *Measure Based Graphic Selection*

Description

Measure based exploratory data analysis. Some of the functions call interactive apps programmed with the package **shiny** to provide flexible selection options.

Details

Package:	mbgraphic
Type:	Package
Version:	1.0.0
Date:	2017-05-07
License:	GPL (>= 2)

Most important functions are:

iaunivariate	Interactive app for exploring univariate anomalies
iacorrgram	Interactive Corrgrams
varclust	Clustering of numeric variables

scag2sdf	Calculating (user defined) scagnostics
sdf	Calculating measures from package **scagnostics**
sdf_sort	Reordering a list of class "sdfdata"
sdf_quicksort	Fast reordering of a list of class "sdfdata"
scaggram	Visualization of scagnostics results in matrix graphics
iascaggram	Interactive scaggrams
iascagpcp	Interactive parallel coordinate plots for exploring scagnostics results
selectscat	Selecting a scatterplot matrix based on scagnostics

Author(s)

Katrin Grimm
Department of Computer Oriented Statistics and Data Analysis
University of Augsburg
Germany

Maintainer: Katrin Grimm <katrin-grimm@web.de>

---

mergesdfdata *Merging two lists of class "sdfdata"*

Description

The function merges two lists of class "sdfdata" generated by function sdf or scag2sdf.

Usage

```
mergesdfdata(sdfdata1, sdfdata2)
```

Arguments

sdfdata1	A list of class "sdfdata".
sdfdata2	A list of class "sdfdata".

Value

A list of class "sdfdata".

Author(s)

Katrin Grimm

See Also

sdf, scag2sdf

Examples

```
data(Election2005)
set.seed(345456)
election05_small <- Election2005[,sample(5:70,5)]
scagdf <- sdf(election05_small)
addscag <- scag2sdf(election05_small,
    scagfun.list=list(dcor2d=dcor2d,splines2d=splines2d))
# merge addscag and scagsdf
scagdf2 <- mergesdfdata(scagdf,addscag)
```

`multimod1d` *Measure of one-dimensional multimodality*

Description

A measure of one-dimensional multimodality based on p-values of the diptest. A jittering whose intensity depends on the value of `discrete1d` is used to prevent ties. The function `dip.test` from package **diptest** is used for calculating the p-value.

Usage

```
multimod1d(x, seed = NULL, exp = 1)
```

Arguments

`x`	A numeric vector, a numeric matrix or a data frame.
`seed`	An integer. The seed which is used for the jittering. If `NULL`, a fixed default value is used.
`exp`	A positive integer. Controls the sensitivity of the result with `(1-p-value)^exp` (see details).

Details

The specific result of the function depends on the seed, its default can be changed by setting `seed`. A normally distributed vector, generated by `rnorm` with `mean=0` and `sd=discrete1d(x)*sd(x,na.rm=T)/5`, is added to `x` to break ties.

To control the sensitivity of the measure an exponent can be choosen which influences the value by `(1-p-value)^exp`. Higher values of `exp` are recommended if a data set has a huge number of multimodal variables. Higher values makes it easier to seperate clear multimodal structures from not so clear ones. For example an exponent of `10` implies a value of `0.6` when `(1-p-value)^exp=0.95` (this is the value for `exp=1`).

Value

A single value or a vector including the results of `skew1d` for each variable (in case of numeric matrices or data frames). If variables from data frames are categorical, 'NA' is returned.

Author(s)

Katrin Grimm

References

M. Maechler (2015) diptest: Hartigan's Dip Test Statistic for Unimodality - Corrected. https://CRAN.R-project.org/package=diptest.

J. A. Hartigan and P. M. Hartigan (1985) The Dip Test of Unimodality. *Annals of Statistics* **13**(1), 70–84.

P. M. Hartigan (1985) Algorithm AS 217: Computation of the Dip Statistic to Test for Unimodality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **34**(3), 320–325.

See Also

skew1d, discrete1d, iaunivariate

Examples

```
data(Election2005)

multimod <- multimod1d(Election2005)
maxv <- order(multimod,decreasing=TRUE)[1:4]
par(mfrow=c(2,2))
for(i in 1:4){
  hist(Election2005[,maxv[i]],xlab="",breaks=30,
  main=paste(names(Election2005[maxv[i]])))
}
## Not run:
# Explore skewness, multimodality and discreteness within interactive environment.
iaunivariate(Election2005)

## End(Not run)
```

---

outlier1dv *Outlier measure based on tukey definition*

Description

Creates a matrix of outlier scores for the data set based on Tukey's defnition.

Usage

```
outlier1dv(x)
```

Arguments

x	A numeric vector, a numeric matrix or a data frame. In case of a data frame only the numeric variables are used.

Value

A data frame with same dimension as x minus non-numeric variables.

Author(s)

Katrin Grimm

See Also

outliertuk

Examples

```
data(Election2005)
outlm <- outlier1dv(Election2005)

# order data frame based on absolute row and column sums
abssum <- function(x){
  x <- abs(x)
  return(sum(x))
}
outlm2 <- outlm[order(apply(outlm,1,abssum),decreasing=TRUE),]
# consider only the 50 rows with  hightest values
outlm3 <- outlm2[1:50,1:68]
outlm4 <- outlm3[,order(apply(outlm3,2,abssum),decreasing=TRUE)]

# plot in heatmap
## Not run:
library(gplots)
heatmap.2(as.matrix(outlm4), col=colorpanel(20, "red", "white","blue"),
          breaks=seq(-1,1,2/20),dendrogram="none",
          key=FALSE,trace="none",Rowv=FALSE,Colv=FALSE)

## End(Not run)
```

---

outliertuk *Outlier classification*

Description

Creates a matrix of outlier classification ("outl" or "non") for the data set based on Tukey's defnition.

Usage

```
outliertuk(x)
```

Arguments

x	A numeric vector, a numeric matrix or a data frame. In case of a data frame only the numeric variables are used.

Value

A data frame with same dimension as x minus non-numeric variables.

Author(s)

Katrin Grimm

See Also

outlier1dv

Examples

```
data(Election2005)
outlc <- outliertuk(Election2005)

# sort and plot with function "visid" from package "extracat"
## Not run:
library(extracat)
visid(outlc,sort="b")

## End(Not run)
```

---

scag2sdf *Calculating (user defined) scagnostics*

Description

The function calculates scagnostics for a data frame based on the functions giving by the user and converts them to a list of class "sdfdata".

Usage

```
scag2sdf(data, scagfun.list, arguments.list = NULL)
```

Arguments

data	A data frame. If it contains categorical variables, they are excluded.
scagfun.list	A list of functions (see Details).
arguments.list	A list of arguments (see Details).

Details

scagfun.list is a list of functions. Each of the functions needs to be a function with arguments x (a numeric vector) and y (a numeric vector). The function can also include further parameters, which can be specified in arguments.list.

Value

A list of class "sdfdata" including data frames sdf and data.

sdf	The scagnostics for every single pair of variables for the functions given in scagfun.list.
data	The original data frame minus categorical variables.

Author(s)

Katrin Grimm

See Also

scag2sdf

Examples

```
data(Election2005)
set.seed(345456)
election05_small <- Election2005[,sample(5:70,10)]

# Use correlation, distance correlation and a spline based
# measure as scagnostics.

scagdf <- scag2sdf(election05_small,
    scagfun.list=
      list(cor=cor,dcor2d=dcor2d,splines2d=splines2d))
head(scagdf$sdf)

# Use sperman correlation
scagdf <- scag2sdf(election05_small,
    scagfun.list=
      list(cor=cor,dcor2d=dcor2d, splines2d=splines2d),
    arguments.list =
      list(list(method="spearman"),NULL,NULL))
```

---

scaggram *Visualization of scagnostics results in matrix graphics*

Description

Scaggram are thought as a generalization of corrgrams. It is possible to use up to three arbitrary scagnostics at the same time. Each of the measures is represented by one of the colors red, green, or blue. The resulting color of a cell (representing a pair of variables) in the scaggram is given by the mixture of the (up to) three colors according to the values of the scagnostics.

Usage

```
scaggram(sdfdata,select=1,add=FALSE,label=FALSE,order= FALSE)
```

Arguments

sdfdata	A list of class "sdfdata".
select	Numeric or character vector of length 1, 2 or 3. Give the column numbers of the measures or their names. Default is 1.
add	Logical. Which additional features should be added to the plot? FALSE for none, "splom" for scatterplots and "glyphs" for star glyphs of all scagnostics stored in sdfdata$sdf.
label	Should the names of the variables be plotted? Default is FALSE.
order	Should the variables be reordered? Default is FALSE. Use 2 or "quicksort" for quick reorering with function sdf_quicksort, 3 or "All scagnostics" for re-ordering with function sdf_sort based on all scagnosticsand 4 or "Selected scagnostics" for reordering with sdf_sort based on the selected scagnostics.

Details

Given three scagnostics with values s1, s2, s3 describing a pair of variables, then the color of the respective cell of the scaggram is determined by rgb(s1,s2,s3, alpha = max(s1,s2,s3)).

If glyphs or a splom are added by setting add, the part above the diagonal is used for the additional feature and the color of the cells is drawn with a transparency of max(s1,s2,s3,0.2)).

Note

The scagnostics need to range between 0 and 1.

For a good and informative color result, it is required that the scagnostics used have similar scales. Otherewise the result can be strongly influenced by single measures.

Author(s)

Katrin Grimm

See Also

iascaggram, sdf, scag2sdf, sdf_sort, sdf_quicksort

rgb

Examples

```
data(Election2005)
# Results of the election
## Not run:
sdfres <- sdf(Election2005[,41:70])
# Use scagnostics "Outlying", "Clumpy" and "Monotonic"
scaggram(sdfres,select=c(1,3,9))

## End(Not run)
```

sdf *Calculates measures from package* **scagnostics**.

Description

The function calculates the scagnostics from package **scagnostics** and converts them to a list of class "sdfdata".

Usage

```
sdf(data, selection = "All")
```

Arguments

data	A data frame. If it contains categorical variables, they got excluded.
selection	Numeric or character. Do you want to use all nine measures or only a selection? Give the numbers of the measures or their names. Default is "All".

Value

A list of class "sdfdata" including data frames sdf and data.

sdf	The (up to) 9 scagnostics from the package **scagnostics** and a variable status, which describes if a plot is classified as "Outlier", "Exemplar" or not.
data	The original data frame minus categorical variables.

Note

Uses '1-Convex' instead of 'Convex'.

The nine measures are: "Outlying", "Skewed", "Clumpy", "Sparse", "Striated", "1-Convex", "Skinny

Author(s)

Katrin Grimm

References

L. Wilkinson, A. Anand and R. Grossman (2005) Graph-Theoretic Scagnostics *Proceedings of the 2005 IEEE Symposium on Information Visualization* 157–164.

L. Wilkinson and A. Anand (2012) scagnostics: Compute scagnostics - scatterplot diagnostics. https://cran.r-project.org/package=scagnostics.

See Also

scag2sdf,

scagnostics in package **scagnostics**

Examples

```
data(Election2005)
set.seed(345456)
election05_small <- Election2005[,sample(5:70,10)]

scagdf <- sdf(election05_small)
head(scagdf$sdf)

# Use only "Outlying", "Clumpy" and "1-Convex"
scagdf_sel <- sdf(election05_small,selection=c("Outlying", "Clumpy","1-Convex"))
# the same as
scagdf_sel <- sdf(election05_small,selection=c(1,3,6))
```

---

sdf_quicksort *Fast reordering of a list of class "sdfdata"*

Description

Reorders a list of class "sdfdata" based on hierarchical clustering in connection with optimal leaf ordering for displaying scagnostics results in scaggrams.

Usage

```
sdf_quicksort(sdfdata)
```

Arguments

sdfdata	A list of class "sdfdata" generated by function sdf or scag2sdf.

Details

The reordering is based on clustering of the variables of sdfdata$data.

1-cor(sdfdata$data) is used as a similarity measure. The linkage method is average. Function seriate from package **seriation** is used.

Value

A list of class "sdfdata".

Note

This is a reordering based on similarities of variables not scatterplots. For reordering based on similarities of scatterplots see sdf_sort.

Author(s)

Katrin Grimm

References

Z. Bar-Joseph, E. D. Demaine, D. K. Gifford, and T. Jaakkola (2001) Fast Optimal Leaf Ordering for Hierarchical Clustering *Bioinformatics* **17**(1) 22–29.

M. Hahsler, K. Hornik und C. Buchta (2008) Getting Things in Order: An Introduction to the R Package seriation *Journal of Statistical Software* **25**(3) 1–34.

See Also

sdf, scag2sdf, sdf_sort, scaggram, iascaggram,

seriate in package **seriation**

Examples

```
data(Election2005)
# consider only demographic/economic information
election05_ds <- Election2005[,1:40]
## Not run:
scagdf <- sdf(election05_ds)
# new order
scagdf_o <- sdf_quicksort(scagdf)
# compare scaggrams
par(mfrow=c(1,2))
scaggram(scagdf,select=(c(1,5,9)))
scaggram(scagdf_o,select=(c(1,5,9)))

## End(Not run)
```

---

sdf_sort *Reordering a list of class "sdfdata"*

Description

The function reorders a list of class "sdfdata" for displaying scagnostics results in scaggrams. The reordering algorithm is based on distances between "direct" neighbors. See Details.

Usage

```
sdf_sort(sdfdata, maxt = NULL, select = "All", printmsum = FALSE)
```

Arguments

sdfdata	A list of class "sdfdata".
maxt	NULL or a positive integer giving a time limit. Default is NULL (no time limit).
select	"All" or a vector of integers or strings giving the scagnostics which should be used for reordering.
printmsum	logical. Should the optimality criterion be printed after each iteration? Default is FALSE.

Details

The implemented algorithm to reorder the "sdfdata" list is greedy. In each iteration all variable changes of exactly two variables are considered, that means associated rows and columns within the scaggram are switched. The (p choose 2) different orders are compared by msum. The goal is to minimize msum, which is defined as the sum of distances of all row-wise and all column-wise neighbors. The algorithm stops if no improvement is possible by changing two arbitrary variables or the specified time limit is exceeded.

Value

A list of class "sdfdata".

Author(s)

Katrin Grimm

See Also

sdf, scag2sdf, sdf_quicksort, scaggram, iascaggram

Examples

```
data(Election2005)
# consider only demographic/economic information
election05_ds <- Election2005[,1:40]
## Not run:
scagdf <- sdf(election05_ds)
# ordering based on all scagnostics from sdf
scagdf_o2 <- sdf_sort(scagdf)
# compare scaggrams
par(mfrow=c(1,2))
scaggram(scagdf,select=(c(1,5,9)))
scaggram(scagdf_o2,select=(c(1,5,9)))

## End(Not run)
```

---

selectscat *Selecting a scatterplot matrix based on scagnostics*

Description

Selects a scatterplot matrix from a data frame including the k variables with approximately highest "relevance". If no own measure of relevance is defined, the function uses the maximum of the measures "Outlying", "Clumpy", "Sparse", "Striated", "1-Convex" and "Stringy" from the **scagnostics** package. See Details and Note.

Usage

```
selectscat(data,relmat=NULL,k=5,r=k,plot=TRUE,criteria="maxm")
```

Arguments

data	A data frame or a list of class "sdfdata". If data is a data frame and contains categorical variables, they are excluded.
relmat	NULL or a matrix which can be interpreted as a similarity matrix (m_ii = 1, m_ij = m_ji, 0 <= m_ij <= 1).
k	A positive integer. The number of variables to include in the scatterplot matrix.
r	A positive integer (greater or equal to k). Controls the goodness of the approximation (see Details).
plot	Logical. Should the plot be drawn? Default is TRUE.
criteria	"maxm" or "cor". Use the maximum of the measures ("maxm") or the correlation as the measure of relevance. Ignored if relmat is not equal to NULL.

Details

To make this selection work fast in case of data sets with a huge number of variables, considering all possible combinations needs to be avoided. The implemented algorithm reorders the variables on optimal leafs. That means an average linkage clustering is done based on the criterion of relevance which is interpreted as a similarity measure. The new order of the variables is chosen so that pairs of variables with high values in the criteria are grouped. That allows us to search around the diagonal of the reordered matrix including all variables for the optimal matrix of size k. The size of the area around the diagonal in which the optimal matrix is searched is controlled by r. If r = p (number of numeric variables of the data set) than every possible combination is considered. Otherwise it is not certain that the optimal matrix is found.

Value

A ggpair object (if plot=TRUE) or a character vector including the variable names selected by the function (if plot=FALSE).

Note

When using more than one measure, results can be strongly influenced by differences in the scales of the measures. Make sure that all measures have similar scales.

When using the function defaults, results can strongly depend on the measure "1-Convex".

Author(s)

Katrin Grimm

References

B. Schloerke et al. (2016) GGally: Extension to ggplot2. https://cran.r-project.org/package=GGally

See Also

sdf, scag2sdf

Examples

```
data(Election2005)

## Not run:
# Use whole data set with default settings
selectscat(Election2005)
# 7 variables and a higher chance of finding optimal matrix
selectscat(Election2005,k=7,r=15)

## End(Not run)

# Use correlation as the measure of relevance
selectscat(Election2005,criteria="cor")
# boring for the election data
# same result as
election_num <- Election2005[,sapply(Election2005,is.numeric)]
selectscat(election_num,relmat=cor(election_num),plot=FALSE)

## Not run:
# If a list of class "sdfdata" is already calculated
sdfdf <- sdf(Election2005)
# Use only measure "Outlying"
sdfdf_O <- sdfdf
sdfdf_O$sdf <- sdfdf_O$sdf[,c(1,10,11)]
selectscat(sdfdf_O,k=7,r=15)

## End(Not run)
```

---

skew1d *Measure for one-dimensional skewness*

Description

A measure for one-dimensional skewness based on quantiles.

Usage

```
skew1d(x)
```

Arguments

x	A numeric vector, a numeric matrix or a data frame.

Value

A single value or a vector including the results of skew1d for each variable (in case of numeric matrices or data frames). If variables from data frames are categorical, 'NA' is returned.

Author(s)

Katrin Grimm

See Also

multimod1d, discrete1d, iaunivariate

Examples

```
data(Election2005)

skew <- skew1d(Election2005)
maxv <- order(skew,decreasing=TRUE)[1:4]
par(mfrow=c(2,2))
for(i in 1:4){
  hist(Election2005[,maxv[i]], xlab="",breaks=30,
  main=paste(names(Election2005[maxv[i]])))
}

## Not run:
# Explore skewness, multimodality and discreteness within interactive environment.
iaunivariate(Election2005)

## End(Not run)
```

---

splines2d *Spline-based dependency measure for pairs of variables*

Description

The function calculates a smoothing spline-based measure for quantifying functional dependencies between two variables. The function gam from package **mgcv** is used.

Usage

```
splines2d(x, y = NULL, binning = FALSE, b = 50, anchor = "min", parallel=FALSE)
```

Arguments

x	A numeric vector, a numeric matrix or a data frame. In case of a data frame only the numeric variables are used.
y	A numeric vector.
binning	A logical value. Whether or not binning should be used. TRUE, "equi" for equidistant binng, "quant" for quantile based binning or "hexb" for hexagonal binning. Default is FALSE.
b	A positive integer. Number of bins in each variable.

anchor	A chraracter string or a numeric value. How should the anchorpoint be chosen? "min" (default) for the minimum of each variable, "ggplot" for the method used in ggplot graphics, "nice" for a "pretty" anchorpoint, or a user specified value.
parallel	A logical value. Whether or not parallelization should be used. Default is FALSE.

Details

For each pair of variables x and y a model where x depends on y and a model where y depends on x are calculated. The proportions of the explained variance is calculated for both models and the maximum is returned. "cr" basis is used for faster calculation.

The number of start knots depends on the number of unique values in the independent variable. If the number is smaller than 20, 3 start knots are used, 10 otherwise.

The smoothing parameter is determined by cross validation.

Value

A numeric value decribing the value of the measure if a pair of vectors is given. Otherwise a data frame with the following variables:

splines2d	Value of the measure.
x1	Number of first variable
x2	Number of second variable.
nx1	Name of first variable (missing if x is not a data frame).
nx2	Name of second variable (missing if x is not a data frame).
tarvar	The variable which was use as target variable (delivered higher value in the measure).

Author(s)

Katrin Grimm

References

S. N. Wood (2006) Generalized Additive Models: An Introduction with R. CRC Press, London.

S. N. Wood (2016). mgcv: Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation. https://cran.r-project.org/package=mgcv

See Also

gam in **mgcv**, dcor2d

Examples

```
data(Election2005)
## Not run:
# spline-based measure for all pairs of variables
spl <- splines2d(Election2005)

# order the pairs decreasing
o_spl <- spl[with(spl,order(spl[,1],decreasing=TRUE)),]

# show the 10 pairs with highest values
o_spl[1:10,]

# Show the 4 scatterplots with highest values
par(mfrow=c(2,2))
for(i in 1:4){
plot(with(Election2005,get(as.character(o_spl$nx1[i]))),
  with(Election2005,get(as.character(o_spl$nx2[i]))),
  xlab=paste(o_spl$nx1[i]),ylab=paste(o_spl$nx2[i]),pch=19)
}

## End(Not run)
```

---

varclust *Clustering of numeric variables*

Description

The function clusters variables using the average linkage algorithm.

Usage

```
varclust(data, c = NULL, mincor = NULL)
```

Arguments

data	A data frame.
c	A positive integer. The number of clusters.
mincor	Numeric. The minimal correlation within each of the single clusters.

Details

The number of clusters can be set by specifying `c` or choosing `mincor` (but not both at the same time). If `mincor` is used, the number of clusters is defined by a minimum correlation. That means every single pair of variables in the individual clusters is correlated with a value which is at least `mincor`.

Value

A list with the following object:

`c`	The number of clusters.
`mincor`	The minimal correlation within each of the single clusters.
`clusters`	The cluster allocation for each variable.
`clusrep`	A character vector indicating the variables which are used as representatives for the clusters.
`dfclusrep`	A data frame which only holds the cluster representatives as variables.

Author(s)

Katrin Grimm

Examples

```
data(Election2005)

# Variable clustering based on minimum correlation
vc1 <- varclust(Election2005,mincor=0.8)
vc1$c # The number of clusters is 45

# Variable clustering based on the number of clusters
vc2 <- varclust(Election2005,30)
vc2$mincor # The minimum correlation is 0.63
```

Index