Package 'VIM'

February 11, 2019

Version 4.8.0

Date 2019-02-07

Title Visualization and Imputation of Missing Values

Depends R (>= 3.1.0),colorspace,grid,data.table(>= 1.9.4)

Imports car, grDevices, robustbase, stats, sp,
vcd,MASS,nnet,e1071,methods,Rcpp,utils,graphics,laeken, ranger

Suggests dplyr, testthat

Description New tools for the visualization of missing and/or imputed values are introduced, which can be used for exploring the data and the structure of the missing and/or imputed values. Depending on this structure of the missing values, the corresponding methods may help to identify the mechanism generating the missing values and allows to explore the data including missing values. In addition, the quality of imputation can be visually explored using various univariate, bivariate, multiple and multivariate plot methods. A graphical user interface available in the separate package VIMGUI allows an easy handling of the implemented plot methods.

LazyData TRUE

ByteCompile TRUE

License GPL (>= 2)

URL https://github.com/statistikat/VIM

Repository CRAN

LinkingTo Rcpp

RoxygenNote 6.1.1

NeedsCompilation yes

Author Matthias Templ [aut, cre],
Alexander Kowarik [aut] (<https://orcid.org/0000-0001-8598-4130>),
Andreas Alfons [aut],
Bernd Prantner [aut]

Maintainer Matthias Templ <matthias.templ@gmail.com>

Date/Publication 2019-02-11 12:33:20 UTC

R topics documented:

VIM-package . . . 3
aggr . . . 4
alphablend . . . 7
barMiss . . . 8
bgmap . . . 10
chorizonDL . . . 11
colormapMiss . . . 15
colSequence . . . 17
countInf . . . 19
gapMiss . . . 20
growdotMiss . . . 21
histMiss . . . 23
hotdeck . . . 25
initialise . . . 27
irmi . . . 28
kNN . . . 30
kola.background . . . 32
mapMiss . . . 33
marginmatrix . . . 34
marginplot . . . 36
matchImpute . . . 38
matrixplot . . . 39
mosaicMiss . . . 41
pairsVIM . . . 43
parcoordMiss . . . 45
pbox . . . 48
prepare . . . 50
print.summary.aggr . . . 51
regressionImp . . . 52
rugNA . . . 53
SBS5242 . . . 54
scattJitt . . . 55
scattmatrixMiss . . . 57
scattMiss . . . 59
sleep . . . 61
spineMiss . . . 62
tao . . . 64
testdata . . . 65
vmGUIenvir . . . 66

Index **68**

VIM-package *Visualization and Imputation of Missing Values*

Description

This package introduces new tools for the visualization of missing or imputed values in , which can be used for exploring the data and the structure of the missing or imputed values. Depending on this structure, they may help to identify the mechanism generating the missing values or errors, which may have happened in the imputation process. This knowledge is necessary for selecting an appropriate imputation method in order to reliably estimate the missing values. Thus the visualization tools should be applied before imputation and the diagnostic tools afterwards.

Details

Detecting missing values mechanisms is usually done by statistical tests or models. Visualization of missing and imputed values can support the test decision, but also reveals more details about the data structure. Most notably, statistical requirements for a test can be checked graphically, and problems like outliers or skewed data distributions can be discovered. Furthermore, the included plot methods may also be able to detect missing values mechanisms in the first place.

A graphical user interface available in the package VIMGUI allows an easy handling of the plot methods. In addition, VIM can be used for data from essentially any field.

Package:	VIM
Version:	3.0.3
Date:	2013-01-09
Depends:	R (>= 2.10),e1071,car, colorspace, nnet, robustbase, tcltk, tkrplot, sp, vcd, Rcpp
Imports:	car, colorspace, grDevices, robustbase, stats, tcltk, sp, utils, vcd
License:	GPL (>= 2)
URL:	http://cran.r-project.org/package=VIM

Author(s)

Matthias Templ, Andreas Alfons, Alexander Kowarik, Bernd Prantner

Maintainer: Matthias Templ <templ@tuwien.ac.at>

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

M. Templ, A. Kowarik, P. Filzmoser (2011) Iterative stepwise regression imputation using standard and robust methods. *Journal of Computational Statistics and Data Analysis*, Vol. 55, pp. 2793-2806.

aggr *Aggregations for missing/imputed values*

Description

Calculate or plot the amount of missing/imputed values in each variable and the amount of missing/imputed values in certain combinations of variables.

Print method for objects of class "aggr".

Summary method for objects of class "aggr".

Usage

```
aggr(x, delimiter = NULL, plot = TRUE, ...)

## S3 method for class 'aggr'
plot(x, col = c("skyblue", "red", "orange"), bars = TRUE,
  numbers = FALSE, prop = TRUE, combined = FALSE, varheight = FALSE,
  only.miss = FALSE, border = par("fg"), sortVars = FALSE,
  sortCombs = TRUE, ylabs = NULL, axes = TRUE, labels = axes,
  cex.lab = 1.2, cex.axis = par("cex"), cex.numbers = par("cex"),
  gap = 4, ...)

## S3 method for class 'aggr'
print(x, digits = NULL, ...)

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

Arguments

x	a vector, matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
plot	a logical indicating whether the results should be plotted (the default is TRUE).
...	for aggr and TKRaggr, further arguments and graphical parameters to be passed to plot.aggr. For plot.aggr, further graphical parameters to be passed down. par("oma") will be set appropriately unless supplied (see par).
col	a vector of length three giving the colors to be used for observed, missing and imputed data. If only one color is supplied, it is used for missing and imputed data and observed data is transparent. If only two colors are supplied, the first one is used for observed data and the second color is used for missing and imputed data.

bars	a logical indicating whether a small barplot for the frequencies of the different combinations should be drawn.
numbers	a logical indicating whether the proportion or frequencies of the different combinations should be represented by numbers.
prop	a logical indicating whether the proportion of missing/imputed values and combinations should be used rather than the total amount.
combined	a logical indicating whether the two plots should be combined. If FALSE, a separate barplot on the left hand side shows the amount of missing/imputed values in each variable. If TRUE, a small version of this barplot is drawn on top of the plot for the combinations of missing/imputed and non-missing values. See "Details" for more information.
varheight	a logical indicating whether the cell heights are given by the frequencies of occurrence of the corresponding combinations.
only.miss	a logical indicating whether the small barplot for the frequencies of the combinations should only be drawn for combinations including missing/imputed values (if bars is TRUE). This is useful if most observations are complete, in which case the corresponding bar would dominate the barplot such that the remaining bars are too compressed. The proportion or frequency of complete observations (as determined by prop) is then represented by a number instead of a bar.
border	the color to be used for the border of the bars and rectangles. Use border=NA to omit borders.
sortVars	a logical indicating whether the variables should be sorted by the number of missing/imputed values.
sortCombs	a logical indicating whether the combinations should be sorted by the frequency of occurrence.
ylabs	if combined is TRUE, a character string giving the y-axis label of the combined plot, otherwise a character vector of length two giving the y-axis labels for the two plots.
axes	a logical indicating whether axes should be drawn.
labels	either a logical indicating whether labels should be plotted on the x-axis, or a character vector giving the labels.
cex.lab	the character expansion factor to be used for the axis labels.
cex.axis	the character expansion factor to be used for the axis annotation.
cex.numbers	the character expansion factor to be used for the proportion or frequencies of the different combinations
gap	if combined is FALSE, a numeric value giving the distance between the two plots in margin lines.
digits	the minimum number of significant digits to be used (see print.default).
object	an object of class "aggr".

Details

Often it is of interest how many missing/imputed values are contained in each variable. Even more interesting, there may be certain combinations of variables with a high number of missing/imputed values.

If `combined` is `FALSE`, two separate plots are drawn for the missing/imputed values in each variable and the combinations of missing/imputed and non-missing values. The barplot on the left hand side shows the amount of missing/imputed values in each variable. In the *aggregation plot* on the right hand side, all existing combinations of missing/imputed and non-missing values in the observations are visualized. Available, missing and imputed data are color coded as given by `col`. Additionally, there are two possibilities to represent the frequencies of occurrence of the different combinations. The first option is to visualize the proportions or frequencies by a small bar plot and/or numbers. The second option is to let the cell heights be given by the frequencies of the corresponding combinations. Furthermore, variables may be sorted by the number of missing/imputed values and combinations by the frequency of occurrence to give more power to finding the structure of missing/imputed values.

If `combined` is `TRUE`, a small version of the barplot showing the amount of missing/imputed values in each variable is drawn on top of the aggregation plot.

The graphical parameter `oma` will be set unless supplied as an argument.

Value

for `aggr`, a list of class `"aggr"` containing the following components: - x the data used. - combinations a character vector representing the combinations of variables. - count the frequencies of these combinations. - percent the percentage of these combinations. - missings a `data.frame` containing the amount of missing/imputed values in each variable. - tabcomb the indicator matrix for the combinations of variables.

a list of class `"summary.aggr"` containing the following components: - missings a `data.frame` containing the amount of missing or imputed values in each variable. - combinations a `data.frame` containing a character vector representing the combinations of variables along with their frequencies and percentages.

Note

Some of the argument names and positions have changed with version 1.3 due to extended functionality and for more consistency with other plot functions in `VIM`. For back compatibility, the arguments `labs` and `names.arg` can still be supplied to `...{}` and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use `ylabs` and `labels` instead.

Author(s)

Andreas Alfons, Matthias Templ, modifications for displaying imputed values by Bernd Prantner

Matthias Templ, modifications by Andreas Alfons and Bernd Prantner

Matthias Templ, modifications by Andreas Alfons

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

print.aggr, summary.aggr

aggr

print.summary.aggr, aggr

Examples

```
data(sleep, package="VIM")
## for missing values
a <- aggr(sleep)
a
summary(a)

## for imputed values
sleep_IMPUTED <- kNN(sleep)
a <- aggr(sleep_IMPUTED, delimiter="_imp")
a
summary(a)


data(sleep, package = "VIM")
a <- aggr(sleep, plot=FALSE)
a


data(sleep, package = "VIM")
summary(aggr(sleep, plot=FALSE))
```

---

alphablend *Alphablending for colors*

Description

Convert colors to semitransparent colors.

Usage

```
alphablend(col, alpha = NULL, bg = NULL)
```

Arguments

col	a vector specifying colors.
alpha	a numeric vector containing the alpha values (between 0 and 1).
bg	the background color to be used for alphablending. This can be used as a workaround for graphics devices that do not support semitransparent colors.

Value

a vector containing the semitransparent colors.

Author(s)

Andreas Alfons

Examples

```
alphablend("red", 0.6)
```

---

barMiss *Barplot with information about missing/imputed values*

Description

Barplot with highlighting of missing/imputed values in other variables by splitting each bar into two parts. Additionally, information about missing/imputed values in the variable of interest is shown on the right hand side.

Usage

```
barMiss(x, delimiter = NULL, pos = 1, selection = c("any", "all"),
  col = c("skyblue", "red", "skyblue4", "red4", "orange", "orange4"),
  border = NULL, main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
  axes = TRUE, labels = axes, only.miss = TRUE, miss.labels = axes,
  interactive = TRUE, ...)
```

Arguments

x	a vector, matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).

pos	a numeric value giving the index of the variable of interest. Additional variables in x are used for highlighting.
selection	the selection method for highlighting missing/imputed values in multiple additional variables. Possible values are "any" (highlighting of missing/imputed values in *any* of the additional variables) and "all" (highlighting of missing/imputed values in *all* of the additional variables).
col	a vector of length six giving the colors to be used. If only one color is supplied, the bars are transparent and the supplied color is used for highlighting missing/imputed values. Else if two colors are supplied, they are recycled.
border	the color to be used for the border of the bars. Use border=NA to omit borders.
main, sub	main and sub title.
xlab, ylab	axis labels.
axes	a logical indicating whether axes should be drawn on the plot.
labels	either a logical indicating whether labels should be plotted below each bar, or a character vector giving the labels.
only.miss	logical; if TRUE, the missing/imputed values in the variable of interest are visualized by a single bar. Otherwise, a small barplot is drawn on the right hand side (see 'Details').
miss.labels	either a logical indicating whether label(s) should be plotted below the bar(s) on the right hand side, or a character string or vector giving the label(s) (see 'Details').
interactive	a logical indicating whether variables can be switched interactively (see 'Details').
...	further graphical parameters to be passed to title and axis.

Details

If more than one variable is supplied, the bars for the variable of interest are split according to missingness/number of imputed missings in the additional variables.

If only.miss=TRUE, the missing/imputed values in the variable of interest are visualized by one bar on the right hand side. If additional variables are supplied, this bar is again split into two parts according to missingness/number of imputed missings in the additional variables.

Otherwise, a small barplot consisting of two bars is drawn on the right hand side. The first bar corresponds to observed values in the variable of interest and the second bar to missing/imputed values. Since these two bars are not on the same scale as the main barplot, a second y-axis is plotted on the right (if axes=TRUE). Each of the two bars are again split into two parts according to missingness/number of imputed missings in the additional variables. Note that this display does not make sense if only one variable is supplied, therefore only.miss is ignored in that case.

If interactive=TRUE, clicking in the left margin of the plot results in switching to the previous variable and clicking in the right margin results in switching to the next variable. Clicking anywhere else on the graphics device quits the interactive session. When switching to a continuous variable, a histogram is plotted rather than a barplot.

Value

a numeric vector giving the coordinates of the midpoints of the bars.

Note

Some of the argument names and positions have changed with version 1.3 due to extended functionality and for more consistency with other plot functions in VIM. For back compatibility, the arguments axisnames, names.arg and names.miss can still be supplied to ...{} and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use labels and miss.labels instead.

Author(s)

Andreas Alfons, modifications to show imputed values by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

spineMiss, histMiss

Examples

```
data(sleep, package = "VIM")
## for missing values
x <- sleep[, c("Exp", "Sleep")]
barMiss(x)
barMiss(x, only.miss = FALSE)

## for imputed values
x_IMPUTED  <- kNN(sleep[, c("Exp", "Sleep")])
barMiss(x_IMPUTED, delimiter = "_imp")
barMiss(x_IMPUTED, delimiter = "_imp", only.miss = FALSE)
```

---

bgmap *Backgound map*

Description

Plot a background map.

Usage

```
bgmap(map, add = FALSE, ...)
```

Arguments

map	either a matrix or data.frame with two columns, a list with components x and y, or an object of any class that can be used for maps and provides its own plot method (e.g., "SpatialPolygons" from package sp). A list of the previously mentioned types can also be provided.
add	a logical indicating whether map should be added to an already existing plot (the default is FALSE).
...	further arguments and graphical parameters to be passed to plot and/or lines.

Author(s)

Andreas Alfons

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

growdotMiss, mapMiss

Examples

```
data(kola.background, package = "VIM")
bgmap(kola.background)
```

---

chorizonDL *C-horizon of the Kola data with missing values*

Description

This data set is the same as the chorizon data set in package mvoutlier, except that values below the detection limit are coded as NA.

Format

A data frame with 606 observations on the following 110 variables.

***ID** a numeric vector

XCOO a numeric vector

YCOO a numeric vector

Ag a numeric vector

Ag_INAA a numeric vector

Al a numeric vector

Al2O3 a numeric vector

As a numeric vector

As_INAA a numeric vector

Au_INAA a numeric vector

B a numeric vector

Ba a numeric vector

Ba_INAA a numeric vector

Be a numeric vector

Bi a numeric vector

Br_IC a numeric vector

Br_INAA a numeric vector

Ca a numeric vector

Ca_INAA a numeric vector

CaO a numeric vector

Cd a numeric vector

Ce_INAA a numeric vector

Cl_IC a numeric vector

Co a numeric vector

Co_INAA a numeric vector

EC a numeric vector

Cr a numeric vector

Cr_INAA a numeric vector

Cs_INAA a numeric vector

Cu a numeric vector

Eu_INAA a numeric vector

F_IC a numeric vector

Fe a numeric vector

Fe_INAA a numeric vector

Fe2O3 a numeric vector

Hf_INAA a numeric vector

Hg a numeric vector

Hg_INAA a numeric vector

Ir_INAA a numeric vector

K a numeric vector

K2O a numeric vector

La a numeric vector

La_INAA a numeric vector

Li a numeric vector

LOI a numeric vector

Lu_INAA a numeric vector

wt_INAA a numeric vector

Mg a numeric vector

MgO a numeric vector

Mn a numeric vector

MnO a numeric vector

Mo a numeric vector

Mo_INAA a numeric vector

Na a numeric vector

Na_INAA a numeric vector

Na2O a numeric vector

Nd_INAA a numeric vector

Ni a numeric vector

Ni_INAA a numeric vector

NO3_IC a numeric vector

P a numeric vector

P2O5 a numeric vector

Pb a numeric vector

pH a numeric vector

PO4_IC a numeric vector

Rb a numeric vector

S a numeric vector

Sb a numeric vector

Sb_INAA a numeric vector

Sc a numeric vector

Sc_INAA a numeric vector

Se a numeric vector

Se_INAA a numeric vector

Si a numeric vector

SiO2 a numeric vector

Sm_INAA a numeric vector

Sn_INAA a numeric vector

SO4_IC a numeric vector

Sr a numeric vector

Sr_INAA a numeric vector

SUM_XRF a numeric vector

Ta_INAA a numeric vector

Tb_INAA a numeric vector

Te a numeric vector

Th a numeric vector

Th_INAA a numeric vector

Ti a numeric vector

TiO2 a numeric vector

U_INAA a numeric vector

V a numeric vector

W_INAA a numeric vector

Y a numeric vector

Yb_INAA a numeric vector

Zn a numeric vector

Zn_INAA a numeric vector

ELEV a numeric vector

***COUN** a numeric vector

***ASP** a numeric vector

TOPC a numeric vector

LITO a numeric vector

Al_XRF a numeric vector

Ca_XRF a numeric vector

Fe_XRF a numeric vector

K_XRF a numeric vector

Mg_XRF a numeric vector

Mn_XRF a numeric vector

Na_XRF a numeric vector

P_XRF a numeric vector

Si_XRF a numeric vector

Ti_XRF a numeric vector

Note

For a more detailed description of this data set, see `chorizon` in package `mvoutlier`.

Source

Kola Project (1993-1998)

References

Reimann, C., Filzmoser, P., Garrett, R.G. and Dutter, R. (2008) *Statistical Data Analysis Explained: Applied Environmental Statistics with R*. Wiley.

See Also

chorizon

Examples

```
data(chorizonDL, package = "VIM")
summary(chorizonDL)
```

---

colormapMiss *Colored map with information about missing/imputed values*

Description

Colored map in which the proportion or amount of missing/imputed values in each region is coded according to a continuous or discrete color scheme. The sequential color palette may thereby be computed in the *HCL* or the *RGB* color space.

Usage

```
colormapMiss(x, region, map, imp_index = NULL, prop = TRUE,
  polysRegion = 1:length(x), range = NULL, n = NULL, col = c("red",
  "orange"), gamma = 2.2, fixup = TRUE, coords = NULL, numbers = TRUE,
  digits = 2, cex.numbers = 0.8, col.numbers = par("fg"), legend = TRUE,
  interactive = TRUE, ...)

colormapMissLegend(xleft, ybottom, xright, ytop, cmap, n = 1000,
  horizontal = TRUE, digits = 2, cex.numbers = 0.8,
  col.numbers = par("fg"), ...)
```

Arguments

x	a numeric vector.
region	a vector or factor of the same length as x giving the regions.
map	an object of any class that contains polygons and provides its own plot method (e.g., "SpatialPolygons" from package sp).
imp_index	a logical-vector indicating which values of 'x' have been imputed. If given, it is used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).

prop	a logical indicating whether the proportion of missing/imputed values should be used rather than the total amount.
polysRegion	a numeric vector specifying the region that each polygon belongs to.
range	a numeric vector of length two specifying the range (minimum and maximum) of the proportion or amount of missing/imputed values to be used for the color scheme.
n	for colormapMiss, the number of equally spaced cut-off points for a discretized color scheme. If this is not a positive integer, a continuous color scheme is used (the default). In the latter case, the number of rectangles to be drawn in the legend can be specified in colormapMissLegend. A reasonably large number makes it appear continuously.
col	the color range (start end end) to be used. RGB colors may be specified as character strings or as objects of class "RGB". HCL colors need to be specified as objects of class "polarLUV". If only one color is supplied, it is used as end color, while the start color is taken to be transparent for RGB or white for HCL.
gamma	numeric; the display *gamma* value (see hex).
fixup	a logical indicating whether the colors should be corrected to valid RGB values (see hex).
coords	a matrix or data.frame with two columns giving the coordinates for the labels.
numbers	a logical indicating whether the corresponding proportions or numbers of missing/imputed values should be used as labels for the regions.
digits	the number of digits to be used in the labels (in case of proportions).
cex.numbers	the character expansion factor to be used for the labels.
col.numbers	the color to be used for the labels.
legend	a logical indicating whether a legend should be plotted.
interactive	a logical indicating whether more detailed information about missing/imputed values should be displayed interactively (see 'Details').
...	further arguments to be passed to plot.
xleft	left *x* position of the legend.
ybottom	bottom *y* position of the legend.
xright	right *x* position of the legend.
ytop	top *y* position of the legend.
cmap	a list as returned by colormapMiss that contains the required information for the legend.
horizontal	a logical indicating whether the legend should be drawn horizontally or vertically.

Details

The proportion or amount of missing/imputed values in x of each region is coded according to a continuous or discrete color scheme in the color range defined by col. In addition, the proportions or numbers can be shown as labels in the regions.

If interactive is TRUE, clicking in a region displays more detailed information about missing/imputed values on the console. Clicking outside the borders quits the interactive session.

Value

colormapMiss returns a list with the following components: - nmiss a numeric vector containing the number of missing/imputed values in each region. - nobs a numeric vector containing the number of observations in each region. - pmiss a numeric vector containing the proportion of missing values in each region. - prop a logical indicating whether the proportion of missing/imputed values have been used rather than the total amount. - range the range of the proportion or amount of missing/imputed values corresponding to the color range. - n either a positive integer giving the number of equally spaced cut-off points for a discretized color scheme, or NULL for a continuous color scheme. - start the start color of the color scheme. - end the end color of the color scheme. - space a character string giving the color space (either "rgb" for RGB colors or "hcl" for HCL colors). - gamma numeric; the display *gamma* value (see hex). - fixup a logical indicating whether the colors have been corrected to valid RGB values (see hex).

Note

Some of the argument names and positions have changed with versions 1.3 and 1.4 due to extended functionality and for more consistency with other plot functions in VIM. For back compatibility, the arguments cex.text and col.text can still be supplied to ...{} and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use cex.numbers and col.numbers instead.

Author(s)

Andreas Alfons, modifications to show imputed values by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

colSequence, growdotMiss, mapMiss

---

colSequence *HCL and RGB color sequences*

Description

Compute color sequences by linear interpolation based on a continuous color scheme between certain start and end colors. Color sequences may thereby be computed in the *HCL* or *RGB* color space.

Usage

```
colSequence(p, start, end, space = c("hcl", "rgb"), ...)

colSequenceRGB(p, start, end, fixup = TRUE, ...)

colSequenceHCL(p, start, end, fixup = TRUE, ...)
```

Arguments

p	a numeric vector in $$[0, 1]$$ giving values to be used for interpolation between the start and end color (0 corresponds to the start color, 1 to the end color).
start, end	the start and end color, respectively. For HCL colors, each can be supplied as a vector of length three (hue, chroma, luminance) or an object of class "polarLUV". For RGB colors, each can be supplied as a character string, a vector of length three (red, green, blue) or an object of class "RGB".
space	character string; if start and end are both numeric, this determines whether they refer to HCL or RGB values. Possible values are "hcl" (for the HCL space) or "rgb" (for the RGB space).
...	for colSequence, additional arguments to be passed to colSequenceHCL or colSequenceRGB. For colSequenceHCL and colSequenceRGB, additional arguments to be passed to hex.
fixup	a logical indicating whether the colors should be corrected to valid RGB values (see hex).

Value

A character vector containing hexadecimal strings of the form "#RRGGBB".

Author(s)

Andreas Alfons

References

Zeileis, A., Hornik, K., Murrell, P. (2009) Escaping RGBland: Selecting colors for statistical graphics. *Computational Statistics & Data Analysis*, **53 (9)**, 1259–1270.

See Also

hex, sequential_hcl

Examples

```
p <- c(0, 0.3, 0.55, 0.8, 1)

## HCL colors
colSequence(p, c(0, 0, 100), c(0, 100, 50))
colSequence(p, polarLUV(L=90, C=30, H=90), c(0, 100, 50))

## RGB colors
colSequence(p, c(1, 1, 1), c(1, 0, 0), space="rgb")
colSequence(p, RGB(1, 1, 0), "red")
```

---

countInf *Count number of infinite or missing values*

Description

Count the number of infinite or missing values in a vector.

Usage

```
countInf(x)
```

Arguments

x	a vector.

Value

`countInf` returns the number of infinite values in `x`. `countNA` returns the number of missing values in `x`.

Author(s)

Andreas Alfons

Examples

```
data(sleep, package="VIM")
countInf(log(sleep$Dream))
countNA(sleep$Dream)
```

gapMiss *Missing value gap statistics*

Description

Computes the average missing value gap of a vector.

Usage

```
gapMiss(x, what = mean)
```

Arguments

x	a numeric vector
what	default is the arithmetic mean. One can include an own function that returns a vector of lenght 1 (e.g. median)

Details

The length of each sequence of missing values (gap) in a vector is calculated and the mean gap is reported

Value

The gap statistics

Author(s)

Matthias Templ based on a suggestion and draft from Huang Tian Yuan.

Examples

```
v <- rnorm(20)
v[3] <- NA
v[6:9] <- NA
v[13:17] <- NA
v
gapMiss(v)
gapMiss(v, what = median)
gapMiss(v, what = function(x) mean(x, trim = 0.1))
gapMiss(v, what = var)
```

`growdotMiss` *Growing dot map with information about missing/imputed values*

Description

Map with dots whose sizes correspond to the values in a certain variable. Observations with missing/imputed values in additional variables are highlighted.

Usage

```
growdotMiss(x, coords, map, pos = 1, delimiter = NULL,
  selection = c("any", "all"), log = FALSE, col = c("skyblue", "red",
  "skyblue4", "red4", "orange", "orange4"), border = par("bg"),
  alpha = NULL, scale = NULL, size = NULL, exp = c(0, 0.95, 0.05),
  col.map = grey(0.5), legend = TRUE, legtitle = "Legend",
  cex.legtitle = par("cex"), cex.legtext = par("cex"), ncircles = 6,
  ndigits = 1, interactive = TRUE, ...)
```

Arguments

`x`	a vector, matrix or `data.frame`.
`coords`	a matrix or `data.frame` with two columns giving the spatial coordinates of the observations.
`map`	a background map to be passed to `bgmap`.
`pos`	a numeric value giving the index of the variable determining the dot sizes.
`delimiter`	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, `x` needs to have `colnames`). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see `col`).
`selection`	the selection method for highlighting missing/imputed values in multiple additional variables. Possible values are `"any"` (highlighting of missing/imputed values in *any* of the additional variables) and `"all"` (highlighting of missing/imputed values in *all* of the additional variables).
`log`	a logical indicating whether the variable given by `pos` should be log-transformed.
`col`	a vector of length six giving the colors to be used in the plot. If only one color is supplied, it is used for the borders of non-highlighted dots and the surface area of highlighted dots. Else if two colors are supplied, they are recycled.
`border`	a vector of length four giving the colors to be used for the borders of the growing dots. Use `NA` to omit borders.
`alpha`	a numeric value between 0 and 1 giving the level of transparency of the colors, or `NULL`. This can be used to prevent overplotting.
`scale`	scaling factor of the map.

size	a vector of length two giving the sizes for the smallest and largest dots.
exp	a vector of length three giving the factors that define the shape of the exponential function (see 'Details').
col.map	the color to be used for the background map.
legend	a logical indicating whether a legend should be plotted.
legtitle	the title for the legend.
cex.legtitle	the character expansion factor to be used for the title of the legend.
cex.legtext	the character expansion factor to be used in the legend.
ncircles	the number of circles displayed in the legend.
ndigits	the number of digits displayed in the legend. Note that \ this is just a suggestion (see format).
interactive	a logical indicating whether information about certain observations can be displayed interactively (see 'Details').
...	for growdotMiss, further arguments and graphical parameters to be passed to bgmap. For bubbleMiss, the arguments to be passed to growdotMiss.

Details

The smallest dots correspond to the 10% quantile and the largest dots to the 99% quantile. In between, the dots grow exponentially, with exp defining the shape of the exponential function. Missings/imputed missings in the variable of interest will be drawn as rectangles.

If interactive=TRUE, detailed information for an observation can be printed on the console by clicking on the corresponding point. Clicking in a region that does not contain any points quits the interactive session.

Note

The function was renamed to growdotMiss in version 1.3. bubbleMiss is a (deprecated) wrapper for growdotMiss for back compatibility with older versions. However, due to extended functionality, some of the argument positions have changed.

The code is based on bubbleFIN from package StatDA.

Author(s)

Andreas Alfons, Matthias Templ, Peter Filzmoser, Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

bgmap, mapMiss, colormapMiss

Examples

```
data(chorizonDL, package = "VIM")
data(kola.background, package = "VIM")
coo <- chorizonDL[, c("XCOO", "YCOO")]
## for missing values
x <- chorizonDL[, c("Ca","As", "Bi")]
growdotMiss(x, coo, kola.background, border = "white")

## for imputed values
x_imp <- kNN(chorizonDL[,c("Ca","As","Bi" )])
growdotMiss(x_imp, coo, kola.background, delimiter = "_imp", border = "white")
```

---

histMiss *Histogram with information about missing/imputed values*

Description

Histogram with highlighting of missing/imputed values in other variables by splitting each bin into two parts. Additionally, information about missing/imputed values in the variable of interest is shown on the right hand side.

Usage

```
histMiss(x, delimiter = NULL, pos = 1, selection = c("any", "all"),
  breaks = "Sturges", right = TRUE, col = c("skyblue", "red", "skyblue4",
  "red4", "orange", "orange4"), border = NULL, main = NULL, sub = NULL,
  xlab = NULL, ylab = NULL, axes = TRUE, only.miss = TRUE,
  miss.labels = axes, interactive = TRUE, ...)
```

Arguments

x	a vector, matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
pos	a numeric value giving the index of the variable of interest. Additional variables in x are used for highlighting.
selection	the selection method for highlighting missing/imputed values in multiple additional variables. Possible values are "any" (highlighting of missing/imputed values in *any* of the additional variables) and "all" (highlighting of missing/imputed values in *all* of the additional variables).

breaks	either a character string naming an algorithm to compute the breakpoints (see hist), or a numeric value giving the number of cells.
right	logical; if TRUE, the histogram cells are right-closed (left-open) intervals.
col	a vector of length six giving the colors to be used. If only one color is supplied, the bars are transparent and the supplied color is used for highlighting missing/imputed values. Else if two colors are supplied, they are recycled.
border	the color to be used for the border of the cells. Use border=NA to omit borders.
main, sub	main and sub title.
xlab, ylab	axis labels.
axes	a logical indicating whether axes should be drawn on the plot.
only.miss	logical; if TRUE, the missing/imputed values in the first variable are visualized by a single bar. Otherwise, a small barplot is drawn on the right hand side (see 'Details').
miss.labels	either a logical indicating whether label(s) should be plotted below the bar(s) on the right hand side, or a character string or vector giving the label(s) (see 'Details').
interactive	a logical indicating whether the variables can be switched interactively (see 'Details').
...	further graphical parameters to be passed to title and axis.

Details

If more than one variable is supplied, the bins for the variable of interest will be split according to missingness/number of imputed missings in the additional variables.

If only.miss=TRUE, the missing/imputed values in the variable of interest are visualized by one bar on the right hand side. If additional variables are supplied, this bar is again split into two parts according to missingness/number of imputed missings in the additional variables.

Otherwise, a small barplot consisting of two bars is drawn on the right hand side. The first bar corresponds to observed values in the variable of interest and the second bar to missing/imputed values. Since these two bars are not on the same scale as the main barplot, a second y-axis is plotted on the right (if axes=TRUE). Each of the two bars are again split into two parts according to missingness/number of imputed missings in the additional variables. Note that this display does not make sense if only one variable is supplied, therefore only.miss is ignored in that case.

If interactive=TRUE, clicking in the left margin of the plot results in switching to the previous variable and clicking in the right margin results in switching to the next variable. Clicking anywhere else on the graphics device quits the interactive session. When switching to a categorical variable, a barplot is produced rather than a histogram.

Value

a list with the following components: - breaks the breakpoints. - counts the number of observations in each cell. - missings the number of highlighted observations in each cell. - mids the cell midpoints.

Note

Some of the argument names and positions have changed with version 1.3 due to extended functionality and for more consistency with other plot functions in `VIM`. For back compatibility, the arguments `axisnames` and `names.miss` can still be supplied to `...{}` and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use `miss.labels` instead.

Author(s)

Andreas Alfons, Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

`spineMiss`, `barMiss`

Examples

```
data(tao, package = "VIM")
## for missing values
x <- tao[, c("Air.Temp", "Humidity")]
histMiss(x)
histMiss(x, only.miss = FALSE)

## for imputed values
x_IMPUTED <- kNN(tao[, c("Air.Temp", "Humidity")])
histMiss(x_IMPUTED, delimiter = "_imp")
histMiss(x_IMPUTED, delimiter = "_imp", only.miss = FALSE)
```

---

hotdeck *Hot-Deck Imputation*

Description

Implementation of the popular Sequential, Random (within a domain) hot-deck algorithm for imputation.

Usage

```
hotdeck(data, variable = NULL, ord_var = NULL, domain_var = NULL,
  makeNA = NULL, NAcond = NULL, impNA = TRUE, donorcond = NULL,
  imp_var = TRUE, imp_suffix = "imp")
```

Arguments

data	data.frame or matrix
variable	variables where missing values should be imputed
ord_var	variables for sorting the data set before imputation
domain_var	variables for building domains and impute within these domains
makeNA	list of length equal to the number of variables, with values, that should be converted to NA for each variable
NAcond	list of length equal to the number of variables, with a condition for imputing a NA
impNA	TRUE/FALSE whether NA should be imputed
donorcond	list of length equal to the number of variables, with a donorcond condition for the donors e.g. ">5"
imp_var	TRUE/FALSE if a TRUE/FALSE variables for each imputed variable should be created show the imputation status
imp_suffix	suffix for the TRUE/FALSE variables showing the imputation status

Value

the imputed data set.

Note

If the sequential hotdeck does not lead to a suitable, a random donor in the group will be used.

Author(s)

Alexander Kowarik

References

A. Kowarik, M. Templ (2016) Imputation with R package VIM. *Journal of Statistical Software*, 74(7), 1-16.

Examples

```
data(sleep)
sleepI <- hotdeck(sleep)
sleepI2 <- hotdeck(sleep,ord_var="BodyWgt",domain_var="Pred")

set.seed(132)
nRows <- 1e3
# Generate a data set with nRows rows and several variables
x<-data.frame(x=rnorm(nRows),y=rnorm(nRows),z=sample(LETTERS,nRows,replace=TRUE),
   d1=sample(LETTERS[1:3],nRows,replace=TRUE),d2=sample(LETTERS[1:2],nRows,replace=TRUE),
    o1=rnorm(nRows),o2=rnorm(nRows),o3=rnorm(100))
origX <- x
```

```
x[sample(1:nRows,nRows/10),1] <- NA
x[sample(1:nRows,nRows/10),2] <- NA
x[sample(1:nRows,nRows/10),3] <- NA
x[sample(1:nRows,nRows/10),4] <- NA
xImp <- hotdeck(x,ord_var = c("o1","o2","o3"),domain_var="d2")
```

`initialise` *Initialization of missing values*

Description

Rough estimation of missing values in a vector according to its type.

Usage

```
initialise(x, mixed, method = "kNN", mixed.constant = NULL)
```

Arguments

`x`	a vector.
`mixed`	a character vector containing the names of variables of type mixed (semi-continous).
`method`	Method used for Initialization (median or kNN)
`mixed.constant`	vector with length equal to the number of semi-continuous variables specifying the point of the semi-continuous distribution with non-zero probability

Details

Missing values are imputed with the mean for vectors of class `"numeric"`, with the median for vectors of class `"integer"`, and with the mode for vectors of class `"factor"`. Hence, `x` should be prepared in the following way: assign class `"numeric"` to numeric vectors, assign class `"integer"` to ordinal vectors, and assign class `"factor"` to nominal or binary vectors.

Value

the initialized vector.

Note

The function is used internally by some imputation algorithms.

Author(s)

Matthias Templ, modifications by Andreas Alfons

irmi *Iterative robust model-based imputation (IRMI)*

Description

In each step of the iteration, one variable is used as a response variable and the remaining variables serve as the regressors.

Usage

```
irmi(x, eps = 5, maxit = 100, mixed = NULL, mixed.constant = NULL,
  count = NULL, step = FALSE, robust = FALSE, takeAll = TRUE,
  noise = TRUE, noise.factor = 1, force = FALSE, robMethod = "MM",
  force.mixed = TRUE, mi = 1, addMixedFactors = FALSE, trace = FALSE,
  init.method = "kNN", modelFormulas = NULL, multinom.method = "multinom",
  imp_var = TRUE, imp_suffix = "imp")
```

Arguments

x	data.frame or matrix
eps	threshold for convergency
maxit	maximum number of iterations
mixed	column index of the semi-continuous variables
mixed.constant	vector with length equal to the number of semi-continuous variables specifying the point of the semi-continuous distribution with non-zero probability
count	column index of count variables
step	a stepwise model selection is applied when the parameter is set to TRUE
robust	if TRUE, robust regression methods will be applied
takeAll	takes information of (initialised) missings in the response as well for regression imputation.
noise	irmi has the option to add a random error term to the imputed values, this creates the possibility for multiple imputation. The error term has mean 0 and variance corresponding to the variance of the regression residuals.
noise.factor	amount of noise.
force	if TRUE, the algorithm tries to find a solution in any case, possible by using different robust methods automatically.
robMethod	regression method when the response is continuous.
force.mixed	if TRUE, the algorithm tries to find a solution in any case, possible by using different robust methods automatically.
mi	number of multiple imputations.
addMixedFactors	if TRUE add additional factor variable for each mixed variable as X variable in the regression

`trace`	Additional information about the iterations when trace equals TRUE.
`init.method`	Method for initialization of missing values (kNN or median)
`modelFormulas`	a named list with the name of variables for the rhs of the formulas, which must contain a rhs formula for each variable with missing values, it should look like list(y1=c("x1","x2"),y2=c("x1","x3"))
	if factor variables for the mixed variables should be created for the regression models
`multinom.method`	
	Method for estimating the multinomial models (current default and only available method is multinom)
`imp_var`	TRUE/FALSE if a TRUE/FALSE variables for each imputed variable should be created show the imputation status
`imp_suffix`	suffix for the TRUE/FALSE variables showing the imputation status

Details

The method works sequentially and iterative. The method can deal with a mixture of continuous, semi-continuous, ordinal and nominal variables including outliers.

A full description of the method can be found in the mentioned reference.

Value

the imputed data set.

Author(s)

Matthias Templ, Alexander Kowarik

References

M. Templ, A. Kowarik, P. Filzmoser (2011) Iterative stepwise regression imputation using standard and robust methods. *Journal of Computational Statistics and Data Analysis*, Vol. 55, pp. 2793-2806.

A. Kowarik, M. Templ (2016) Imputation with R package VIM. *Journal of Statistical Software*, 74(7), 1-16.

See Also

`mi`

Examples

```
data(sleep)
irmi(sleep)

data(testdata)
imp_testdata1 <- irmi(testdata$wna,mixed=testdata$mixed)
```

```
# mixed.constant != 0 (-10)
testdata$wna$m1[testdata$wna$m1==0] <- -10
testdata$wna$m2 <- log(testdata$wna$m2+0.001)
imp_testdata2 <- irmi(testdata$wna,mixed=testdata$mixed,mixed.constant=c(-10,log(0.001)))
imp_testdata2$m2 <- exp(imp_testdata2$m2)-0.001

#example with fixed formulas for the variables with missing
form=list(
NonD=c("BodyWgt","BrainWgt"),
Dream=c("BodyWgt","BrainWgt"),
Sleep=c("BrainWgt"),
Span=c("BodyWgt"),
Gest=c("BodyWgt","BrainWgt")
)
irmi(sleep,modelFormulas=form,trace=TRUE)

# Example with ordered variable
td <- testdata$wna
td$c1 <- as.ordered(td$c1)
irmi(td)
```

---

kNN *k-Nearest Neighbour Imputation*

Description

k-Nearest Neighbour Imputation based on a variation of the Gower Distance for numerical, categorical, ordered and semi-continous variables.

Usage

```
kNN(data, variable = colnames(data), metric = NULL, k = 5,
  dist_var = colnames(data), weights = NULL, numFun = median,
  catFun = maxCat, makeNA = NULL, NAcond = NULL, impNA = TRUE,
  donorcond = NULL, mixed = vector(), mixed.constant = NULL,
  trace = FALSE, imp_var = TRUE, imp_suffix = "imp", addRF = FALSE,
  onlyRF = FALSE, addRandom = FALSE, useImputedDist = TRUE,
  weightDist = FALSE)
```

Arguments

data	data.frame or matrix
variable	variables where missing values should be imputed
metric	metric to be used for calculating the distances between
k	number of Nearest Neighbours used
dist_var	names or variables to be used for distance calculation

weights	weights for the variables for distance calculation. If weights = "auto" weights will be selected based on variable importance from random forest regression, using function ranger. Weights are calculated for each variable seperately.
numFun	function for aggregating the k Nearest Neighbours in the case of a numerical variable
catFun	function for aggregating the k Nearest Neighbours in the case of a categorical variable
makeNA	list of length equal to the number of variables, with values, that should be converted to NA for each variable
NAcond	list of length equal to the number of variables, with a condition for imputing a NA
impNA	TRUE/FALSE whether NA should be imputed
donorcond	condition for the donors e.g. list(">5"), must be NULL or a list of same length as variable
mixed	names of mixed variables
mixed.constant	vector with length equal to the number of semi-continuous variables specifying the point of the semi-continuous distribution with non-zero probability
trace	TRUE/FALSE if additional information about the imputation process should be printed
imp_var	TRUE/FALSE if a TRUE/FALSE variables for each imputed variable should be created show the imputation status
imp_suffix	suffix for the TRUE/FALSE variables showing the imputation status
addRF	TRUE/FALSE each variable will be modelled using random forest regression (ranger) and used as additional distance variable.
onlyRF	TRUE/FALSE if TRUE only additional distance variables created from random forest regression will be used as distance variables.
addRandom	TRUE/FALSE if an additional random variable should be added for distance calculation
useImputedDist	TRUE/FALSE if an imputed value should be used for distance calculation for imputing another variable. Be aware that this results in a dependency on the ordering of the variables.
weightDist	TRUE/FALSE if the distances of the k nearest neighbours should be used as weights in the aggregation step

Details

The function sampleCat samples with probabilites corresponding to the occurrence of the level in the NNs. The function maxCat chooses the level with the most occurrences and random if the maximum is not unique. The function gowerD is used by kNN to compute the distances for numerical, factor ordered and semi-continous variables.

Value

the imputed data set.

Author(s)

Alexander Kowarik, Statistik Austria

References

A. Kowarik, M. Templ (2016) Imputation with R package VIM. *Journal of Statistical Software*, 74(7), 1-16.

Examples

```
data(sleep)
kNN(sleep)
library(laeken)
kNN(sleep, numFun = weightedMean, weightDist=TRUE)
```

---

kola.background *Background map for the Kola project data*

Description

Coordinates of the Kola background map.

Source

Kola Project (1993-1998)

References

Reimann, C., Filzmoser, P., Garrett, R.G. and Dutter, R. (2008) *Statistical Data Analysis Explained: Applied Environmental Statistics with R*. Wiley, 2008.

Examples

```
data(kola.background, package = "VIM")
bgmap(kola.background)
```

mapMiss *Map with information about missing/imputed values*

Description

Map of observed and missing/imputed values.

Usage

```
mapMiss(x, coords, map, delimiter = NULL, selection = c("any", "all"),
  col = c("skyblue", "red", "orange"), alpha = NULL, pch = c(19, 15),
  col.map = grey(0.5), legend = TRUE, interactive = TRUE, ...)
```

Arguments

x	a vector, matrix or data.frame.
coords	a data.frame or matrix with two columns giving the spatial coordinates of the observations.
map	a background map to be passed to bgmap.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
selection	the selection method for displaying missing/imputed values in the map. Possible values are "any" (display missing/imputed values in *any* variable) and "all" (display missing/imputed values in *all* variables).
col	a vector of length three giving the colors to be used for observed, missing and imputed values. If a single color is supplied, it is used for all values.
alpha	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
pch	a vector of length two giving the plot characters to be used for observed and missing/imputed values. If a single plot character is supplied, it will be used for both.
col.map	the color to be used for the background map.
legend	a logical indicating whether a legend should be plotted.
interactive	a logical indicating whether information about selected observations can be displayed interactively (see 'Details').
...	further graphical parameters to be passed to bgmap and points.

Details

If `interactive=TRUE`, detailed information for an observation can be printed on the console by clicking on the corresponding point. Clicking in a region that does not contain any points quits the interactive session.

Author(s)

Matthias Templ, Andreas Alfons, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

`bgmap`, `bubbleMiss`, `colormapMiss`

Examples

```
data(chorizonDL, package = "VIM")
data(kola.background, package = "VIM")
coo <- chorizonDL[, c("XCOO", "YCOO")]
## for missing values
x <- chorizonDL[, c("As", "Bi")]
mapMiss(x, coo, kola.background)

## for imputed values
x_imp <- kNN(chorizonDL[, c("As", "Bi")])
mapMiss(x_imp, coo, kola.background, delimiter = "_imp")
```

---

marginmatrix *Marginplot Matrix*

Description

Create a scatterplot matrix with information about missing/imputed values in the plot margins of each panel.

Usage

```
marginmatrix(x, delimiter = NULL, col = c("skyblue", "red", "red4",
  "orange", "orange4"), alpha = NULL, ...)
```

Arguments

x	a matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
col	a vector of length five giving the colors to be used in the marginplots in the off-diagonal panels. The first color is used for the scatterplot and the boxplots for the available data, the second/fourth color for the univariate scatterplots and boxplots for the missing/imputed values in one variable, and the third/fifth color for the frequency of missing/imputed values in both variables (see 'Details'). If only one color is supplied, it is used for the bivariate and univariate scatterplots and the boxplots for missing/imputed values in one variable, whereas the boxplots for the available data are transparent. Else if two colors are supplied, the second one is recycled.
alpha	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
...	further arguments and graphical parameters to be passed to pairsVIM and marginplot. par("oma") will be set appropriately unless supplied (see par).

Details

marginmatrix uses pairsVIM with a panel function based on marginplot.

The graphical parameter oma will be set unless supplied as an argument.

Author(s)

Andreas Alfons, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

marginplot, pairsVIM, scattmatrixMiss

Examples

```
data(sleep, package = "VIM")
## for missing values
x <- sleep[, 1:5]
x[,c(1,2,4)] <- log10(x[,c(1,2,4)])
```

```
marginmatrix(x)

## for imputed values
x_imp <- kNN(sleep[, 1:5])
x_imp[,c(1,2,4)] <- log10(x_imp[,c(1,2,4)])
marginmatrix(x_imp, delimiter = "_imp")
```

`marginplot` *Scatterplot with additional information in the margins*

Description

In addition to a standard scatterplot, information about missing/imputed values is shown in the plot margins. Furthermore, imputed values are highlighted in the scatterplot.

Usage

```
marginplot(x, delimiter = NULL, col = c("skyblue", "red", "red4", "orange",
  "orange4"), alpha = NULL, pch = c(1, 16), cex = par("cex"),
  numbers = TRUE, cex.numbers = par("cex"), zeros = FALSE, xlim = NULL,
  ylim = NULL, main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
  ann = par("ann"), axes = TRUE, frame.plot = axes, ...)
```

Arguments

`x`	a `matrix` or `data.frame` with two columns.
`delimiter`	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, `x` needs to have `colnames`). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see `col`).
`col`	a vector of length five giving the colors to be used in the plot. The first color is used for the scatterplot and the boxplots for the available data. In case of missing values, the second color is taken for the univariate scatterplots and boxplots for missing values in one variable and the third for the frequency of missing/imputed values in both variables (see 'Details'). Otherwise, in case of imputed values, the fourth color is used for the highlighting, the frequency, the univariate scatterplot and the boxplots of mputed values in the first variable and the fifth color for the same applied to the second variable. A black color is used for the highlighting and the frequency of imputed values in both variables instead. If only one color is supplied, it is used for the bivariate and univariate scatterplots and the boxplots for missing/imputed values in one variable, whereas the boxplots for the available data are transparent. Else if two colors are supplied, the second one is recycled.

alpha	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
pch	a vector of length two giving the plot symbols to be used for the scatterplot and the univariate scatterplots. If a single plot character is supplied, it is used for the scatterplot and the default value will be used for the univariate scatterplots (see 'Details').
cex	the character expansion factor to be used for the bivariate and univariate scatterplots.
numbers	a logical indicating whether the frequencies of missing/imputed values should be displayed in the lower left of the plot (see 'Details').
cex.numbers	the character expansion factor to be used for the frequencies of the missing/imputed values.
zeros	a logical vector of length two indicating whether the variables are semi-continuous, i.e., contain a considerable amount of zeros. If TRUE, only the non-zero observations are used for drawing the respective boxplot. If a single logical is supplied, it is recycled.
xlim, ylim	axis limits.
main, sub	main and sub title.
xlab, ylab	axis labels.
ann	a logical indicating whether plot annotation (main, sub, xlab, ylab) should be displayed.
axes	a logical indicating whether both axes should be drawn on the plot. Use graphical parameter "xaxt" or "yaxt" to suppress only one of the axes.
frame.plot	a logical indicating whether a box should be drawn around the plot.
...	further graphical parameters to be passed down (see par).

Details

Boxplots for available and missing/imputed data, as well as univariate scatterplots for missing/imputed values in one variable are shown in the plot margins.

Imputed values in either of the variables are highlighted in the scatterplot.

Furthermore, the frequencies of the missing/imputed values can be displayed by a number (lower left of the plot). The number in the lower left corner is the number of observations that are missing/imputed in both variables.

Note

Some of the argument names and positions have changed with versions 1.3 and 1.4 due to extended functionality and for more consistency with other plot functions in VIM. For back compatibility, the argument cex.text can still be supplied to ...{} and is handled correctly. Nevertheless, it is deprecated and no longer documented. Use cex.numbers instead.

Author(s)

Andreas Alfons, Matthias Templ, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

scattMiss

Examples

```
data(tao, package = "VIM")
data(chorizonDL, package = "VIM")
## for missing values
marginplot(tao[,c("Air.Temp", "Humidity")])
marginplot(log10(chorizonDL[,c("CaO", "Bi")]))

## for imputed values
marginplot(kNN(tao[,c("Air.Temp", "Humidity")]), delimiter = "_imp")
marginplot(kNN(log10(chorizonDL[,c("CaO", "Bi")])), delimiter = "_imp")
```

---

matchImpute *Fast matching/imputation based on categorical variable*

Description

Suitable donors are searched based on matching of the categorical variables. The variables are dropped in reversed order, so that the last element of 'match_var' is dropped first and the first element of the vector is dropped last.

Usage

```
matchImpute(data, variable = colnames(data)[!colnames(data) %in% match_var],
  match_var, imp_var = TRUE, imp_suffix = "imp")
```

Arguments

data	data.frame, data.table, survey object or matrix
variable	variables to be imputed
match_var	variables used for matching
imp_var	TRUE/FALSE if a TRUE/FALSE variables for each imputed variable should be created show the imputation status
imp_suffix	suffix for the TRUE/FALSE variables showing the imputation status

Details

The method works by sampling values from the suitable donors.

Value

the imputed data set.

Author(s)

Johannes Gussenbauer, Alexander Kowarik

See Also

hotdeck

Examples

```
data(sleep,package="VIM")
imp_data <- matchImpute(sleep,variable=c("NonD","Dream","Sleep","Span","Gest"),
  match_var=c("Exp","Danger"))

data(testdata,package="VIM")
imp_testdata1 <- matchImpute(testdata$wna,match_var=c("c1","c2","b1","b2"))

dt <- data.table(testdata$wna)
imp_testdata2 <- matchImpute(dt,match_var=c("c1","c2","b1","b2"))
```

---

matrixplot *Matrix plot*

Description

Create a matrix plot, in which all cells of a data matrix are visualized by rectangles. Available data is coded according to a continuous color scheme, while missing/imputed data is visualized by a clearly distinguishable color.

Usage

```
matrixplot(x, delimiter = NULL, sortby = NULL, col = c("red", "orange"),
  fixup = TRUE, xlim = NULL, ylim = NULL, main = NULL, sub = NULL,
  xlab = NULL, ylab = NULL, axes = TRUE, labels = axes, xpd = NULL,
  interactive = TRUE, ...)
```

Arguments

x	a matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
sortby	a numeric or character value specifying the variable to sort the data matrix by, or NULL to plot without sorting.
col	the colors to be used in the plot. RGB colors may be specified as character strings or as objects of class "RGB". HCL colors need to be specified as objects of class "polarLUV". If only one color is supplied, it is used for missing and imputed data and a greyscale is used for available data. If two colors are supplied, the first is used for missing and the second for imputed data and a greyscale for available data. If three colors are supplied, the first is used as end color for the available data, while the start color is taken to be transparent for RGB or white for HCL. Missing/imputed data is visualized by the second/third color in this case. If four colors are supplied, the first is used as start color and the second as end color for the available data, while the third/fourth color is used for missing/imputed data.
fixup	a logical indicating whether the colors should be corrected to valid RGB values (see hex).
xlim, ylim	axis limits.
main, sub	main and sub title.
xlab, ylab	axis labels.
axes	a logical indicating whether axes should be drawn on the plot.
labels	either a logical indicating whether labels should be plotted below each column, or a character vector giving the labels.
xpd	a logical indicating whether the rectangles should be allowed to go outside the plot region. If NULL, it defaults to TRUE unless axis limits are specified.
interactive	a logical indicating whether a variable to be used for sorting can be selected interactively (see 'Details').
...	for matrixplot and iimagMiss, further graphical parameters to be passed to plot.window, title and axis. For TKRmatrixplot, further arguments to be passed to matrixplot.

Details

In a *matrix plot*, all cells of a data matrix are visualized by rectangles. Available data is coded according to a continuous color scheme. To compute the colors via interpolation, the variables are first scaled to the interval

$$[0, 1]$$

. Missing/imputed values can then be visualized by a clearly distinguishable color. It is thereby possible to use colors in the *HCL* or *RGB* color space. A simple way of visualizing the magnitude

of the available data is to apply a greyscale, which has the advantage that missing/imputed values can easily be distinguished by using a color such as red/orange. Note that `-Inf` and `Inf` are always assigned the begin and end color, respectively, of the continuous color scheme.

Additionally, the observations can be sorted by the magnitude of a selected variable. If `interactive` is `TRUE`, clicking in a column redraws the plot with observations sorted by the corresponding variable. Clicking anywhere outside the plot region quits the interactive session.

Note

This is a much more powerful extension to the function `imagmiss` in the former CRAN package `dprep`.

`iimagMiss` is deprecated and may be omitted in future versions of `VIM`. Use `matrixplot` instead.

Author(s)

Andreas Alfons, Matthias Templ, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

Examples

```
data(sleep, package = "VIM")
## for missing values
x <- sleep[, -(8:10)]
x[,c(1,2,4,6,7)] <- log10(x[,c(1,2,4,6,7)])
matrixplot(x, sortby = "BrainWgt")

## for imputed values
x_imp <- kNN(sleep[, -(8:10)])
x_imp[,c(1,2,4,6,7)] <- log10(x_imp[,c(1,2,4,6,7)])
matrixplot(x_imp, delimiter = "_imp", sortby = "BrainWgt")
```

---

mosaicMiss *Mosaic plot with information about missing/imputed values*

Description

Create a mosaic plot with information about missing/imputed values.

Usage

```
mosaicMiss(x, delimiter = NULL, highlight = NULL, selection = c("any",
  "all"), plotvars = NULL, col = c("skyblue", "red", "orange"),
  labels = NULL, miss.labels = TRUE, ...)
```

Arguments

x	a matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
highlight	a vector giving the variables to be used for highlighting. If NULL (the default), all variables are used for highlighting.
selection	the selection method for highlighting missing/imputed values in multiple highlight variables. Possible values are "any" (highlighting of missing/imputed values in *any* of the highlight variables) and "all" (highlighting of missing/imputed values in *all* of the highlight variables).
plotvars	a vector giving the categorical variables to be plotted. If NULL (the default), all variables are plotted.
col	a vector of length three giving the colors to be used for observed, missing and imputed data. If only one color is supplied, the tiles corresponding to observed data are transparent and the supplied color is used for highlighting.
labels	a list of arguments for the labeling function labeling_border.
miss.labels	either a logical indicating whether labels should be plotted for observed and missing/imputed (highlighted) data, or a character vector giving the labels.
...	additional arguments to be passed to mosaic.

Details

Mosaic plots are graphical representations of multi-way contingency tables. The frequencies of the different cells are visualized by area-proportional rectangles (tiles). Additional tiles are be used to display the frequencies of missing/imputed values. Furthermore, missing/imputed values in a certain variable or combination of variables can be highlighted in order to explore their structure.

Value

An object of class "structable" is returned invisibly.

Note

This function uses the highly flexible strucplot framework of package vcd.

Author(s)

Andreas Alfons, modifications by Bernd Prantner

References

Meyer, D., Zeileis, A. and Hornik, K. (2006) The `strucplot` framework: Visualizing multi-way contingency tables with **vcd**. *Journal of Statistical Software*, **17 (3)**, 1–48.

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

`spineMiss`, `mosaic`

Examples

```
data(sleep, package = "VIM")
## for missing values
mosaicMiss(sleep, highlight = 4,
    plotvars = 8:10, miss.labels = FALSE)

## for imputed values
mosaicMiss(kNN(sleep), highlight = 4,
    plotvars = 8:10, delimiter = "_imp", miss.labels = FALSE)
```

---

pairsVIM *Scatterplot Matrices*

Description

Create a scatterplot matrix.

Usage

```
pairsVIM(x, ..., delimiter = NULL, main = NULL, sub = NULL,
  panel = points, lower = panel, upper = panel, diagonal = NULL,
  labels = TRUE, pos.labels = NULL, cex.labels = NULL,
  font.labels = par("font"), layout = c("matrix", "graph"), gap = 1)
```

Arguments

x	a matrix or data.frame.
...	further arguments and graphical parameters to be passed down. par("oma") will be set appropriately unless supplied (see par).
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
main, sub	main and sub title.
panel	a function(x, y, ...{}), which is used to plot the contents of each off-diagonal panel of the display.
lower, upper	separate panel functions to be used below and above the diagonal, respectively.
diagonal	optional function(x, ...{}) to be applied on the diagonal panels.
labels	either a logical indicating whether labels should be plotted in the diagonal panels, or a character vector giving the labels.
pos.labels	the vertical position of the labels in the diagonal panels.
cex.labels	the character expansion factor to be used for the labels.
font.labels	the font to be used for the labels.
layout	a character string giving the layout of the scatterplot matrix. Possible values are "matrix" (a matrix-like layout with the first row on top) and "graph" (a graph-like layout with the first row at the bottom).
gap	a numeric value giving the distance between the panels in margin lines.

Details

This function is the workhorse for marginmatrix and scattmatrixMiss.

The graphical parameter oma will be set unless supplied as an argument.

A panel function should not attempt to start a new plot, since the coordinate system for each panel is set up by pairsVIM.

Note

The code is based on pairs. Starting with version 1.4, infinite values are no longer removed before passing the x and y vectors to the panel functions.

Author(s)

Andreas Alfons, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

marginmatrix, scattmatrixMiss

Examples

```
data(sleep, package = "VIM")
x <- sleep[, -(8:10)]
x[,c(1,2,4,6,7)] <- log10(x[,c(1,2,4,6,7)])
pairsVIM(x)
```

---

parcoordMiss *Parallel coordinate plot with information about missing/imputed values*

Description

Parallel coordinate plot with adjustments for missing/imputed values. Missing values in the plotted variables may be represented by a point above the corresponding coordinate axis to prevent disconnected lines. In addition, observations with missing/imputed values in selected variables may be highlighted.

Usage

```
parcoordMiss(x, delimiter = NULL, highlight = NULL, selection = c("any",
  "all"), plotvars = NULL, plotNA = TRUE, col = c("skyblue", "red",
  "skyblue4", "red4", "orange", "orange4"), alpha = NULL, lty = par("lty"),
  xlim = NULL, ylim = NULL, main = NULL, sub = NULL, xlab = NULL,
  ylab = NULL, labels = TRUE, xpd = NULL, interactive = TRUE, ...)
```

Arguments

x	a matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
highlight	a vector giving the variables to be used for highlighting. If NULL (the default), all variables are used for highlighting.
selection	the selection method for highlighting missing/imputed values in multiple highlight variables. Possible values are "any" (highlighting of missing/imputed values in *any* of the highlight variables) and "all" (highlighting of missing/imputed values in *all* of the highlight variables).

plotvars	a vector giving the variables to be plotted. If NULL (the default), all variables are plotted.
plotNA	a logical indicating whether missing values in the plot variables should be represented by a point above the corresponding coordinate axis to prevent disconnected lines.
col	if plotNA is TRUE, a vector of length six giving the colors to be used for observations with different combinations of observed and missing/imputed values in the plot variables and highlight variables (vectors of length one or two are recycled). Otherwise, a vector of length two giving the colors for non-highlighted and highlighted observations (if a single color is supplied, it is used for both).
alpha	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
lty	if plotNA is TRUE, a vector of length four giving the line types to be used for observations with different combinations of observed and missing/imputed values in the plot variables and highlight variables (vectors of length one or two are recycled). Otherwise, a vector of length two giving the line types for non-highlighted and highlighted observations (if a single line type is supplied, it is used for both).
xlim, ylim	axis limits.
main, sub	main and sub title.
xlab, ylab	axis labels.
labels	either a logical indicating whether labels should be plotted below each coordinate axis, or a character vector giving the labels.
xpd	a logical indicating whether the lines should be allowed to go outside the plot region. If NULL, it defaults to TRUE unless axis limits are specified.
interactive	a logical indicating whether interactive features should be enabled (see 'Details').
...	for parcoordMiss, further graphical parameters to be passed down (see par). For TKRparcoordMiss, further arguments to be passed to parcoordMiss.

Details

In parallel coordinate plots, the variables are represented by parallel axes. Each observation of the scaled data is shown as a line. Observations with missing/imputed values in selected variables may thereby be highlighted. However, plotting variables with missing values results in disconnected lines, making it impossible to trace the respective observations across the graph. As a remedy, missing values may be represented by a point above the corresponding coordinate axis, which is separated from the main plot by a small gap and a horizontal line, as determined by plotNA. Connected lines can then be drawn for all observations. Nevertheless, a caveat of this display is that it may draw attention away from the main relationships between the variables.

If interactive is TRUE, it is possible switch between this display and the standard display without the separate level for missing values by clicking in the top margin of the plot. In addition, the variables to be used for highlighting can be selected interactively. Observations with missing/imputed values in any or in all of the selected variables are highlighted (as determined by selection). A variable can be added to the selection by clicking on a coordinate axis. If a variable is already

selected, clicking on its coordinate axis removes it from the selection. Clicking anywhere outside the plot region (except the top margin, if missing/imputed values exist) quits the interactive session.

Note

Some of the argument names and positions have changed with versions 1.3 and 1.4 due to extended functionality and for more consistency with other plot functions in `VIM`. For back compatibility, the arguments `colcomb` and `xaxlabels` can still be supplied to `...{}` and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use `highlight` and `labels` instead.

Author(s)

Andreas Alfons, Matthias Templ, modifications by Bernd Prantner

References

Wegman, E. J. (1990) Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association* **85 (411)**, 664–675.

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

pbox

Examples

```
data(chorizonDL, package = "VIM")
## for missing values
parcoordMiss(chorizonDL[,c(15,101:110)],
    plotvars=2:11, interactive = FALSE)
legend("top", col = c("skyblue", "red"), lwd = c(1,1),
    legend = c("observed in Bi", "missing in Bi"))

## for imputed values
parcoordMiss(kNN(chorizonDL[,c(15,101:110)]), delimiter = "_imp" ,
    plotvars=2:11, interactive = FALSE)
legend("top", col = c("skyblue", "orange"), lwd = c(1,1),
    legend = c("observed in Bi", "imputed in Bi"))
```

pbox *Parallel boxplots with information about missing/imputed values*

Description

Boxplot of one variable of interest plus information about missing/imputed values in other variables.

Usage

```
pbox(x, delimiter = NULL, pos = 1, selection = c("none", "any", "all"),
  col = c("skyblue", "red", "red4", "orange", "orange4"), numbers = TRUE,
  cex.numbers = par("cex"), xlim = NULL, ylim = NULL, main = NULL,
  sub = NULL, xlab = NULL, ylab = NULL, axes = TRUE,
  frame.plot = axes, labels = axes, interactive = TRUE, ...)
```

Arguments

x	a vector, matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
pos	a numeric value giving the index of the variable of interest. Additional variables in x are used for grouping according to missingness/number of imputed missings.
selection	the selection method for grouping according to missingness/number of imputed missings in multiple additional variables. Possible values are "none" (grouping according to missingness/number of imputed missings in every other variable that contains missing/imputed values), "any" (grouping according to missingness/number of imputed missings in *any* of the additional variables) and "all" (grouping according to missingness/number of imputed missings in *all* of the additional variables).
col	a vector of length five giving the colors to be used in the plot. The first color is used for the boxplots of the available data, the second/fourth are used for missing/imputed data, respectively, and the third/fifth color for the frequencies of missing/imputed values in both variables (see 'Details'). If only one color is supplied, it is used for the boxplots for missing/imputed data, whereas the boxplots for the available data are transparent. Else if two colors are supplied, the second one is recycled.
numbers	a logical indicating whether the frequencies of missing/imputed values should be displayed (see 'Details').
cex.numbers	the character expansion factor to be used for the frequencies of the missing/imputed values.

xlim, ylim	axis limits.
main, sub	main and sub title.
xlab, ylab	axis labels.
axes	a logical indicating whether axes should be drawn on the plot.
frame.plot	a logical indicating whether a box should be drawn around the plot.
labels	either a logical indicating whether labels should be plotted below each box, or a character vector giving the labels.
interactive	a logical indicating whether variables can be switched interactively (see 'Details').
...	for pbox, further arguments and graphical parameters to be passed to boxplot and other functions. For TKRpbox, further arguments to be passed to pbox.

Details

This plot consists of several boxplots. First, a standard boxplot of the variable of interest is produced. Second, boxplots grouped by observed and missing/imputed values according to selection are produced for the variable of interest.

Additionally, the frequencies of the missing/imputed values can be represented by numbers. If so, the first line corresponds to the observed values of the variable of interest and their distribution in the different groups, the second line to the missing/imputed values.

If interactive=TRUE, clicking in the left margin of the plot results in switching to the previous variable and clicking in the right margin results in switching to the next variable. Clicking anywhere else on the graphics device quits the interactive session.

Value

a list as returned by boxplot.

Note

Some of the argument names and positions have changed with version 1.3 due to extended functionality and for more consistency with other plot functions in VIM. For back compatibility, the arguments names and cex.text can still be supplied to ...{} and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use labels and cex.numbers instead.

Author(s)

Andreas Alfons, Matthias Templ, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

parcoordMiss

Examples

```
data(chorizonDL, package = "VIM")
## for missing values
pbox(log(chorizonDL[, c(4,5,8,10,11,16:17,19,25,29,37,38,40)]))

## for imputed values
pbox(kNN(log(chorizonDL[, c(4,8,10,11,17,19,25,29,37,38,40)])),
     delimiter = "_imp")
```

`prepare` *Transformation and standardization*

Description

This function is used by the `VIM` GUI for transformation and standardization of the data.

Usage

```
prepare (x, scaling = c("none","classical","MCD","robust","onestep"),
    transformation = c("none","minus","reciprocal","logarithm",
  "exponential","boxcox","clr","ilr","alr"),
     alpha = NULL, powers = NULL, start = 0, alrVar)
```

Arguments

`x`	a vector, matrix or `data.frame`.
`scaling`	the scaling to be applied to the data. Possible values are `"none"`, `"classical"`, `MCD`, `"robust"` and `"onestep"`.
`transformation`	the transformation of the data. Possible values are `"none"`, `"minus"`, `"reciprocal"`, `"logarithm"`, `"exponential"`, `"boxcox"`, `"clr"`, `"ilr"` and `"alr"`.
`alpha`	a numeric parameter controlling the size of the subset for the *MCD* (if `scaling="MCD"`). See `covMcd`.
`powers`	a numeric vector giving the powers to be used in the Box-Cox transformation (if `transformation="boxcox"`). If `NULL`, the powers are calculated with function `powerTransform`.
`start`	a constant to be added prior to Box-Cox transformation (if `transformation="boxcox"`).
`alrVar`	variable to be used as denominator in the additive logratio transformation (if `transformation="alr"`).

Details

Transformation:

"none": no transformation is used.

"logarithm": compute the the logarithm (to the base 10).

"boxcox": apply a Box-Cox transformation. Powers may be specified or calculated with the function powerTransform.

Standardization:

"none": no standardization is used.

"classical": apply a z-Transformation on each variable by using function scale.

"robust": apply a robustified z-Transformation by using median and MAD.

Value

Transformed and standardized data.

Author(s)

Matthias Templ, modifications by Andreas Alfons

See Also

scale, powerTransform

Examples

```
data(sleep, package = "VIM")
x <- sleep[, c("BodyWgt", "BrainWgt")]
prepare(x, scaling = "robust", transformation = "logarithm")
```

---

print.summary.aggr *Print method for objects of class summary.aggr*

Description

Print method for objects of class "summary.aggr".

Usage

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

Arguments

x	an object of class "summary.aggr".
...	Further arguments, currently ignored.

Author(s)

Andreas Alfons, modifications by Bernd Prantner

See Also

summary.aggr, aggr

Examples

```
data(sleep, package = "VIM")
s <- summary(aggr(sleep, plot=FALSE))
s
```

---

regressionImp *Regression Imputation*

Description

Impute missing values based on a regression model.

Usage

```
regressionImp(formula, data, family = "AUTO", robust = FALSE,
  imp_var = TRUE, imp_suffix = "imp", mod_cat = FALSE)
```

Arguments

formula	model formula to impute one variable
data	A data.frame or survey object containing the data
family	family argument for "glm" ("AUTO" tries to choose automatically, only really tested option!!!)
robust	TRUE/FALSE if robust regression should be used
imp_var	TRUE/FALSE if a TRUE/FALSE variables for each imputed variable should be created show the imputation status
imp_suffix	suffix used for TF imputation variables
mod_cat	TRUE/FALSE if TRUE for categorical variables the level with the highest prediction probability is selected, otherwise it is sampled according to the probabilities.

Details

"lm" is used for family "normal" and glm for all other families. (Robust=TRUE: lmrob, glmrob)

Value

the imputed data set.

Author(s)

Alexander Kowarik

References

A. Kowarik, M. Templ (2016) Imputation with R package VIM. *Journal of Statistical Software*, 74(7), 1-16.

Examples

```
data(sleep)
sleepImp1 <- regressionImp(Dream+NonD~BodyWgt+BrainWgt,data=sleep)
sleepImp2 <- regressionImp(Sleep+Gest+Span+Dream+NonD~BodyWgt+BrainWgt,data=sleep)

data(testdata)
imp_testdata1 <- regressionImp(b1+b2~x1+x2,data=testdata$wna)
imp_testdata3 <- regressionImp(x1~x2,data=testdata$wna,robust=TRUE)
```

---

rugNA *Rug representation of missing/imputed values*

Description

Add a rug representation of missing/imputed values in only one of the variables to scatterplots.

Usage

```
rugNA(x, y, ticksize = NULL, side = 1, col = "red", alpha = NULL,
  miss = NULL, lwd = 0.5, ...)
```

Arguments

x, y	numeric vectors.
ticksize	the length of the ticks. Positive lengths give inward ticks.
side	an integer giving the side of the plot to draw the rug representation.
col	the color to be used for the ticks.
alpha	the alpha value (between 0 and 1).

miss	a data.frame or matrix with two columns and logical values. If NULL, x and y are searched for missing values, otherwise, the first column of miss is used to determine the imputed values in x and the second one for the imputed values in y.
lwd	the line width to be used for the ticks.
...	further arguments to be passed to Axis.

Details

If side is 1 or 3, the rug representation consists of values available in x but missing/imputed in y. Else if side is 2 or 4, it consists of values available in y but missing/imputed in x.

Author(s)

Andreas Alfons, modifications by Bernd Prantner

Examples

```
data(tao, package = "VIM")
## for missing values
x <- tao[, "Air.Temp"]
y <- tao[, "Humidity"]
plot(x, y)
rugNA(x, y, side = 1)
rugNA(x, y, side = 2)

## for imputed values
x_imp <- kNN(tao[, c("Air.Temp","Humidity")])
x <- x_imp[, "Air.Temp"]
y <- x_imp[, "Humidity"]
miss <- x_imp[, c("Air.Temp_imp","Humidity_imp")]
plot(x, y)
rugNA(x, y, side = 1, col = "orange", miss = miss)
rugNA(x, y, side = 2, col = "orange", miss = miss)
```

SBS5242 *Synthetic subset of the Austrian structural business statistics data*

Description

Synthetic subset of the Austrian structural business statistics (SBS) data, namely NACE code 52.42 (retail sale of clothing).

Details

The Austrian SBS data set consists of more than 320.000 enterprises. Available raw (unedited) data set: 21669 observations in 90 variables, structured according NACE revision 1.1 with 3891 missing values.

We investigate 9 variables of NACE 52.42 (retail sale of clothing).

From these confidential raw data set a non-confidential, close-to-reality, synthetic data set was generated.

Source

http://www.statistik.at

Examples

```
data(SBS5242)
aggr(SBS5242)
```

---

scattJitt *Bivariate jitter plot*

Description

Create a bivariate jitter plot.

Usage

```
scattJitt(x, delimiter = NULL, col = c("skyblue", "red", "red4", "orange",
  "orange4"), alpha = NULL, cex = par("cex"), col.line = "lightgrey",
  lty = "dashed", lwd = par("lwd"), numbers = TRUE,
  cex.numbers = par("cex"), main = NULL, sub = NULL, xlab = NULL,
  ylab = NULL, axes = TRUE, frame.plot = axes, labels = c("observed",
  "missing", "imputed"), ...)
```

Arguments

x	a data.frame or matrix with two columns.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).

col	a vector of length five giving the colors to be used in the plot. The first color will be used for complete observations, the second/fourth color for missing/imputed values in only one variable, and the third/fifth color for missing/imputed values in both variables. If only one color is supplied, it is used for all. Else if two colors are supplied, the second one is recycled.
alpha	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
cex	the character expansion factor for the plot characters.
col.line	the color for the lines dividing the plot region.
lty	the line type for the lines dividing the plot region (see par).
lwd	the line width for the lines dividing the plot region.
numbers	a logical indicating whether the frequencies of observed and missing/imputed values should be displayed (see 'Details').
cex.numbers	the character expansion factor to be used for the frequencies of the observed and missing/imputed values.
main, sub	main and sub title.
xlab, ylab	axis labels.
axes	a logical indicating whether both axes should be drawn on the plot. Use graphical parameter "xaxt" or "yaxt" to suppress just one of the axes.
frame.plot	a logical indicating whether a box should be drawn around the plot.
labels	a vector of length three giving the axis labels for the regions for observed, missing and imputed values (see 'Details').
...	further graphical parameters to be passed down (see par).

Details

The amount of observed and missing/imputed values is visualized by jittered points. Thereby the plot region is divided into up to four regions according to the existence of missing/imputed values in one or both variables. In addition, the amount of observed and missing/imputed values can be represented by a number.

Note

Some of the argument names and positions have changed with version 1.3 due to extended functionality and for more consistency with other plot functions in VIM. For back compatibility, the argument cex.text can still be supplied to ...{} and is handled correctly. Nevertheless, it is deprecated and no longer documented. Use cex.numbers instead.

Author(s)

Matthias Templ, modifications by Andreas Alfons and Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

Examples

```
data(tao, package = "VIM")
## for missing values
scattJitt(tao[, c("Air.Temp", "Humidity")])

## for imputed values
scattJitt(kNN(tao[, c("Air.Temp", "Humidity")]), delimiter = "_imp")
```

`scattmatrixMiss` *Scatterplot matrix with information about missing/imputed values*

Description

Scatterplot matrix in which observations with missing/imputed values in certain variables are highlighted.

Usage

```
scattmatrixMiss(x, delimiter = NULL, highlight = NULL,
  selection = c("any", "all"), plotvars = NULL, col = c("skyblue", "red",
  "orange"), alpha = NULL, pch = c(1, 3), lty = par("lty"),
  diagonal = c("density", "none"), interactive = TRUE, ...)
```

Arguments

`x`	a matrix or `data.frame`.
`delimiter`	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, `x` needs to have `colnames`). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see `col`).
`highlight`	a vector giving the variables to be used for highlighting. If `NULL` (the default), all variables are used for highlighting.
`selection`	the selection method for highlighting missing/imputed values in multiple highlight variables. Possible values are `"any"` (highlighting of missing/imputed values in *any* of the highlight variables) and `"all"` (highlighting of missing/imputed values in *all* of the highlight variables).
`plotvars`	a vector giving the variables to be plotted. If `NULL` (the default), all variables are plotted.
`col`	a vector of length three giving the colors to be used in the plot. The second/third color will be used for highlighting missing/imputed values.
`alpha`	a numeric value between 0 and 1 giving the level of transparency of the colors, or `NULL`. This can be used to prevent overplotting.

pch	a vector of length two giving the plot characters. The second plot character will be used for the highlighted observations.
lty	a vector of length two giving the line types for the density plots in the diagonal panels (if diagonal="density"). The second line type is used for the highlighted observations. If a single value is supplied, it is used for both non-highlighted and highlighted observations.
diagonal	a character string specifying the plot to be drawn in the diagonal panels. Possible values are "density" (density plots for non-highlighted and highlighted observations) and "none".
interactive	a logical indicating whether the variables to be used for highlighting can be selected interactively (see 'Details').
...	for scattmatrixMiss, further arguments and graphical parameters to be passed to pairsVIM. par("oma") will be set appropriately unless supplied (see par). For TKRscattmatrixMiss, further arguments to be passed to scattmatrixMiss.

Details

scattmatrixMiss uses pairsVIM with a panel function that allows highlighting of missing/imputed values.

If interactive=TRUE, the variables to be used for highlighting can be selected interactively. Observations with missing/imputed values in any or in all of the selected variables are highlighted (as determined by selection). A variable can be added to the selection by clicking in a diagonal panel. If a variable is already selected, clicking on the corresponding diagonal panel removes it from the selection. Clicking anywhere else quits the interactive session.

The graphical parameter oma will be set unless supplied as an argument.

TKRscattmatrixMiss behaves like scattmatrixMiss, but uses tkrplot to embed the plot in a *Tcl/Tk* window. This is useful if the number of variables is large, because scrollbars allow to move from one part of the plot to another.

Note

Some of the argument names and positions have changed with version 1.3 due to a re-implementation and for more consistency with other plot functions in VIM. For back compatibility, the argument colcomb can still be supplied to ...{} and is handled correctly. Nevertheless, it is deprecated and no longer documented. Use highlight instead. The arguments smooth, reg.line and legend.plot are no longer used and ignored if supplied.

Author(s)

Andreas Alfons, Matthias Templ, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

pairsVIM, marginmatrix

Examples

```
data(sleep, package = "VIM")
## for missing values
x <- sleep[, 1:5]
x[,c(1,2,4)] <- log10(x[,c(1,2,4)])
scattmatrixMiss(x, highlight = "Dream")

## for imputed values
x_imp <- kNN(sleep[, 1:5])
x_imp[,c(1,2,4)] <- log10(x_imp[,c(1,2,4)])
scattmatrixMiss(x_imp, delimiter = "_imp", highlight = "Dream")
```

---

scattMiss *Scatterplot with information about missing/imputed values*

Description

In addition to a standard scatterplot, lines are plotted for the missing values in one variable. If there are imputed values, they will be highlighted.

Usage

```
scattMiss(x, delimiter = NULL, side = 1, col = c("skyblue", "red",
  "orange", "lightgrey"), alpha = NULL, lty = c("dashed", "dotted"),
  lwd = par("lwd"), quantiles = c(0.5, 0.975), inEllipse = FALSE,
  zeros = FALSE, xlim = NULL, ylim = NULL, main = NULL, sub = NULL,
  xlab = NULL, ylab = NULL, interactive = TRUE, ...)
```

Arguments

x	a matrix or data.frame with two columns.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
side	if side=1, a rug representation and vertical lines are plotted for the missing/imputed values in the second variable; if side=2, a rug representation and horizontal lines for the missing/imputed values in the first variable.

col	a vector of length four giving the colors to be used in the plot. The first color is used for the scatterplot, the second/third color for the rug representation for missing/imputed values. The second color is also used for the lines for missing values. Imputed values will be highlighted with the third color, and the fourth color is used for the ellipses (see 'Details'). If only one color is supplied, it is used for the scatterplot, the rug representation and the lines, whereas the default color is used for the ellipses. Else if a vector of length two is supplied, the default color is used for the ellipses as well.
alpha	a numeric value between 0 and 1 giving the level of transparency of the colors, or NULL. This can be used to prevent overplotting.
lty	a vector of length two giving the line types for the lines and ellipses. If a single value is supplied, it will be used for both.
lwd	a vector of length two giving the line widths for the lines and ellipses. If a single value is supplied, it will be used for both.
quantiles	a vector giving the quantiles of the chi-square distribution to be used for the tolerance ellipses, or NULL to suppress plotting ellipses (see 'Details').
inEllipse	plot lines only inside the largest ellipse. Ignored if quantiles is NULL or if there are imputed values.
zeros	a logical vector of length two indicating whether the variables are semi-continuous, i.e., contain a considerable amount of zeros. If TRUE, only the non-zero observations are used for computing the tolerance ellipses. If a single logical is supplied, it is recycled. Ignored if quantiles is NULL.
xlim, ylim	axis limits.
main, sub	main and sub title.
xlab, ylab	axis labels.
interactive	a logical indicating whether the side argument can be changed interactively (see 'Details').
...	further graphical parameters to be passed down (see par).

Details

Information about missing values in one variable is included as vertical or horizontal lines, as determined by the side argument. The lines are thereby drawn at the observed x- or y-value. In case of imputed values, they will additionally be highlighted in the scatterplot. Supplementary, percentage coverage ellipses can be drawn to give a clue about the shape of the bivariate data distribution.

If interactiveis TRUE, clicking in the bottom margin redraws the plot with information about missing/imputed values in the first variable and clicking in the left margin redraws the plot with information about missing/imputed values in the second variable. Clicking anywhere else in the plot quits the interactive session.

Note

The argument zeros has been introduced in version 1.4. As a result, some of the argument positions have changed.

Author(s)

Andreas Alfons, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

marginplot

Examples

```
data(tao, package = "VIM")
## for missing values
scattMiss(tao[,c("Air.Temp", "Humidity")])

## for imputed values
scattMiss(kNN(tao[,c("Air.Temp", "Humidity")]), delimiter = "_imp")
```

---

sleep *Mammal sleep data*

Description

Sleep data with missing values.

Format

A data frame with 62 observations on the following 10 variables.

BodyWgt a numeric vector

BrainWgt a numeric vector

NonD a numeric vector

Dream a numeric vector

Sleep a numeric vector

Span a numeric vector

Gest a numeric vector

Pred a numeric vector

Exp a numeric vector

Danger a numeric vector

Source

Allison, T. and Chichetti, D. (1976) Sleep in mammals: ecological and constitutional correlates. *Science* **194 (4266)**, 732–734.

The data set was imported from GGobi.

Examples

```
data(sleep, package = "VIM")
summary(sleep)
aggr(sleep)
```

---

spineMiss *Spineplot with information about missing/imputed values*

Description

Spineplot or spinogram with highlighting of missing/imputed values in other variables by splitting each cell into two parts. Additionally, information about missing/imputed values in the variable of interest is shown on the right hand side.

Usage

```
spineMiss(x, delimiter = NULL, pos = 1, selection = c("any", "all"),
  breaks = "Sturges", right = TRUE, col = c("skyblue", "red", "skyblue4",
  "red4", "orange", "orange4"), border = NULL, main = NULL, sub = NULL,
  xlab = NULL, ylab = NULL, axes = TRUE, labels = axes,
  only.miss = TRUE, miss.labels = axes, interactive = TRUE, ...)
```

Arguments

x	a vector, matrix or data.frame.
delimiter	a character-vector to distinguish between variables and imputation-indices for imputed variables (therefore, x needs to have colnames). If given, it is used to determine the corresponding imputation-index for any imputed variable (a logical-vector indicating which values of the variable have been imputed). If such imputation-indices are found, they are used for highlighting and the colors are adjusted according to the given colors for imputed variables (see col).
pos	a numeric value giving the index of the variable of interest. Additional variables in x are used for highlighting.
selection	the selection method for highlighting missing/imputed values in multiple additional variables. Possible values are "any" (highlighting of missing/imputed values in *any* of the additional variables) and "all" (highlighting of missing/imputed values in *all* of the additional variables).

breaks	if the variable of interest is numeric, breaks controls the breakpoints (see hist for possible values).
right	logical; if TRUE and the variable of interest is numeric, the spinogram cells are right-closed (left-open) intervals.
col	a vector of length six giving the colors to be used. If only one color is supplied, the bars are transparent and the supplied color is used for highlighting missing/imputed values. Else if two colors are supplied, they are recycled.
border	the color to be used for the border of the cells. Use border=NA to omit borders.
main, sub	main and sub title.
xlab, ylab	axis labels.
axes	a logical indicating whether axes should be drawn on the plot.
labels	if the variable of interest is categorical, either a logical indicating whether labels should be plotted below each cell, or a character vector giving the labels. This is ignored if the variable of interest is numeric.
only.miss	logical; if TRUE, the missing/imputed values in the variable of interest are also visualized by a cell in the spineplot or spinogram. Otherwise, a small spineplot is drawn on the right hand side (see 'Details').
miss.labels	either a logical indicating whether label(s) should be plotted below the cell(s) on the right hand side, or a character string or vector giving the label(s) (see 'Details').
interactive	a logical indicating whether the variables can be switched interactively (see 'Details').
...	further graphical parameters to be passed to title and axis.

Details

A spineplot is created if the variable of interest is categorial and a spinogram if it is numerical. The horizontal axis is scaled according to relative frequencies of the categories/classes. If more than one variable is supplied, the cells are split according to missingness/number of imputed values in the additional variables. Thus the proportion of highlighted observations in each category/class is displayed on the vertical axis. Since the height of each cell corresponds to the proportion of highlighted observations, it is now possible to compare the proportions of missing/imputed values among the different categories/classes.

If only.miss=TRUE, the missing/imputed values in the variable of interest are also visualized by a cell in the spine plot or spinogram. If additional variables are supplied, this cell is again split into two parts according to missingness/number if imputed values in the additional variables.

Otherwise, a small spineplot that visualizes missing/imputed values in the variable of interest is drawn on the right hand side. The first cell corresponds to observed values and the second cell to missing/imputed values. Each of the two cells is again split into two parts according to missingness/number of imputed values in the additional variables. Note that this display does not make sense if only one variable is supplied, therefore only.miss is ignored in that case.

If interactive=TRUE, clicking in the left margin of the plot results in switching to the previous variable and clicking in the right margin results in switching to the next variable. Clicking anywhere else on the graphics device quits the interactive session.

Value

a table containing the frequencies corresponding to the cells.

Note

Some of the argument names and positions have changed with version 1.3 due to extended functionality and for more consistency with other plot functions in VIM. For back compatibility, the arguments xaxlabels and missaxlabels can still be supplied to ...{} and are handled correctly. Nevertheless, they are deprecated and no longer documented. Use labels and miss.labels instead.

The code is based on the function spineplot by Achim Zeileis.

Author(s)

Andreas Alfons, Matthias Templ, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

See Also

histMiss, barMiss, mosaicMiss

Examples

```
data(tao, package = "VIM")
data(sleep, package = "VIM")
## for missing values
spineMiss(tao[, c("Air.Temp", "Humidity")])
spineMiss(sleep[, c("Exp", "Sleep")])

## for imputed values
spineMiss(kNN(tao[, c("Air.Temp", "Humidity")]), delimiter = "_imp")
spineMiss(kNN(sleep[, c("Exp", "Sleep")]), delimiter = "_imp")
```

---

tao *Tropical Atmosphere Ocean (TAO) project data*

Description

A small subsample of the Tropical Atmosphere Ocean (TAO) project data, derived from the GGOBI project.

Format

A data frame with 736 observations on the following 8 variables.

Year a numeric vector
Latitude a numeric vector
Longitude a numeric vector
Sea.Surface.Temp a numeric vector
Air.Temp a numeric vector
Humidity a numeric vector
UWind a numeric vector
VWind a numeric vector

Details

All cases recorded for five locations and two time periods.

Source

http://www.pmel.noaa.gov/tao/

Examples

```
data(tao, package = "VIM")
summary(tao)
aggr(tao)
```

---

testdata *Simulated data set for testing purpose*

Description

2 numeric, 2 binary, 2 nominal and 2 mixed (semi-continous) variables

Format

The format is: List of 4 $ wna :'data.frame': 500 obs. of 8 variables: ..$ x1: num [1:500] 10.87 9.53 7.83 8.53 8.67 ... ..$ x2: num [1:500] 10.9 9.32 7.68 8.2 8.41 ... ..$ c1: Factor w/ 4 levels "a","b","c","d": 3 2 2 1 2 2 1 3 3 2 ... ..$ c2: Factor w/ 4 levels "a","b","c","d": 2 3 2 2 2 2 2 4 2 2 ... ..$ b1: Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 1 1 ... ..$ b2: Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 2 2 2 ... ..$ m1: num [1:500] 0 8.29 9.08 0 0 ... ..$ m2: num [1:500] 10.66 9.39 7.8 8.11 7.33 ... $ wona :'data.frame': 500 obs. of 8 variables: ..$ x1: num [1:500] 10.87 9.53 7.83 8.53 8.67 ... ..$ x2: num [1:500] 10.9 9.32 7.68 8.2 8.41 ... ..$ c1: Factor w/ 4 levels "a","b","c","d": 3 2 2 1 2 2 1 3 3 2 ... ..$ c2: Factor w/ 4 levels "a","b","c","d": 2 3 2 2 2 2 2 4 2 2 ... ..$ b1: Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 1 1 ... ..$ b2: Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 2 2 2 ... ..$ m1: num [1:500] 0 8.29 9.08 0 0 ... ..$ m2: num [1:500] 10.66 9.39 7.8 8.11 7.33 ... $ mixed : chr [1:2] "m1" "m2" $ outlierInd: NULL

Examples

```
data(testdata)
```

`vmGUIenvir` *Environment for the GUI for Visualization and Imputation of Missing Values*

Description

Location were everything from package VIM and VIMGUI is stored.

Usage

```
vmGUIenvir

putVm(x, value)

getVm(x, mode = "any")

existsVm(x, mode = "any")

rmVm(...)
```

Arguments

`x`	object name
`value`	value to be assigned to x
`mode`	see 'exists'
`...`	see 'rm'

Format

An object of class `environment` of length 0.

Details

Internal information regarding the VIM GUI is stored in the environment `vmGUIenvir`.

Author(s)

Andreas Alfons, based on an initial design by Matthias Templ, modifications by Bernd Prantner

References

M. Templ, A. Alfons, P. Filzmoser (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Online first. DOI: 10.1007/s11634-011-0102-y.

Index

∗Topic **color**
alphablend, 7
colSequence, 17
rugNA, 53
∗Topic **datasets**
chorizonDL, 11
kola.background, 32
SBS5242, 54
sleep, 61
tao, 64
testdata, 65
∗Topic **hplot**
aggr, 4
barMiss, 8
bgmap, 10
colormapMiss, 15
growdotMiss, 21
histMiss, 23
mapMiss, 33
marginmatrix, 34
marginplot, 36
matrixplot, 39
mosaicMiss, 41
pairsVIM, 43
parcoordMiss, 45
pbox, 48
scattJitt, 55
scattmatrixMiss, 57
scattMiss, 59
spineMiss, 62
vmGUIenvir, 66
∗Topic **manip**
gapMiss, 20
hotdeck, 25
initialise, 27
irmi, 28
kNN, 30
matchImpute, 38
prepare, 50
regressionImp, 52
∗Topic **multivariate**
vmGUIenvir, 66
∗Topic **package**
VIM-package, 3
∗Topic **print**
aggr, 4
print.summary.aggr, 51
∗Topic **utilities**
countInf, 19

aggr, 4, *7*, *52*
alphablend, 7
Axis, *54*
axis, *9*, *24*, *40*, *63*

barMiss, 8, *25*, *64*
bgmap, 10, *21, 22*, *33, 34*
boxplot, *49*
bubbleFIN, *22*
bubbleMiss, *34*
bubbleMiss *(*growdotMiss*)*, 21

chorizon, *11*, *14, 15*
chorizonDL, 11
colnames, *4*, *8*, *21*, *23*, *33*, *35, 36*, *40*, *42*, *44, 45*, *48*, *55*, *57*, *59*, *62*
colormapMiss, 15, *22*, *34*
colormapMissLegend *(*colormapMiss*)*, 15
colSequence, *17*, 17
colSequenceHCL *(*colSequence*)*, 17
colSequenceRGB *(*colSequence*)*, 17
countInf, 19
countNA *(*countInf*)*, 19
covMcd, *50*

existsVm *(*vmGUIenvir*)*, 66

format, *22*

gapMiss, 20

getVm *(vmGUIenvir)*, 66
gowerD *(kNN)*, 30
growdotMiss, *11*, *17*, 21

hex, *16–18*, *40*
hist, *24*, *63*
histMiss, *10*, 23, *64*
hotdeck, 25, *39*

iimagMiss *(matrixplot)*, 39
initialise, 27
irmi, 28

kNN, 30
kola.background, 32

labeling_border, *42*
lines, *11*

mapMiss, *11*, *17*, *22*, 33
marginmatrix, 34, *44, 45*, *59*
marginplot, *35*, 36, *61*
matchImpute, 38
matrixplot, 39
maxCat *(kNN)*, 30
mi, *29*
mosaic, *42, 43*
mosaicMiss, 41, *64*

pairs, *44*
pairsVIM, *35*, 43, *58, 59*
par, *4*, *35*, *37*, *44*, *46*, *56*, *58*, *60*
parcoordMiss, 45, *49*
pbox, *47*, 48
plot.aggr, *4*
plot.aggr *(aggr)*, 4
plot.window, *40*
points, *33*
polarLUV, *16*, *18*, *40*
powerTransform, *50, 51*
prepare, 50
print.aggr, *7*
print.aggr *(aggr)*, 4
print.default, *5*
print.summary.aggr, *7*, 51
putVm *(vmGUIenvir)*, 66

ranger, *31*
regressionImp, 52
RGB, *16*, *18*, *40*
rmVm *(vmGUIenvir)*, 66
rugNA, 53

sampleCat *(kNN)*, 30
SBS5242, 54
scale, *51*
scattJitt, 55
scattmatrixMiss, *35*, *44, 45*, 57
scattMiss, *38*, 59
sequential_hcl, *18*
sleep, 61
spineMiss, *10*, *25*, *43*, 62
spineplot, *64*
summary.aggr, *7*, *52*
summary.aggr *(aggr)*, 4

tao, 64
testdata, 65
title, *9*, *24*, *40*, *63*
TKRmatrixplot *(matrixplot)*, 39
tkrplot, *58*

VIM *(VIM-package)*, 3
VIM-package, 3
vmGUIenvir, 66