Package: gamlss.prepdata (via r-universe)

March 28, 2025

Type Package

Title Prepering Data for Fitting a Generalized Additive Model for Location Scale and Shape

Version 0.1.5

Date 2025-03-06

Description Functions for prepering data to fit a Generalized Additive Models for Location Scale and Shape from the 'gamlss' or `gamlss2` package, Stasinopoulos and Rigby (2007) <doi:10.18637/jss.v023.i07>, using for graphical methods 'ggplot2'.

License GPL-2 | GPL-3

URL https://www.gamlss.com/

BugReports https://github.com/gamlss-dev/gamlss.prepdata/issues

Depends R (>= 3.5.0), gamlss.dist, gamlss (>= 4.3.3), gamlss.foreach

Imports methods, ggridges, ellipse, gamlss.inf, foreach, mgcv, ggplot2, yaImpute, gamlss.ggplots

Suggests glmnet, reshape2, igraph, networkD3, grid, gridExtra

LazyLoad yes

Repository https://gamlss-dev.r-universe.dev

RemoteUrl https://github.com/gamlss-dev/gamlss.prepdata

RemoteRef HEAD

RemoteSha 28cf1d1bfe9f7efb39b01cd40738f6433e63de05

Contents

`gamlss.prepdata-package`

Prepering Data for Fitting a Generalized Additive Model for Location Scale and Shape

---

Description

Functions for prepering data to fit a Generalized Additive Models for Location Scale and Shape from the 'gamlss' or ‘gamlss2‘ package, Stasinopoulos and Rigby (2007) <doi:10.18637/jss.v023.i07>, using for graphical methods 'ggplot2'.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
The following convention has been used to name the functions:

`y_NAME`: plots concerning fitted values from a single fitted model

`data_NAME`: plots concerning residuals from a single fitted model

where `NAME` refer to different characteristics.

Author(s)

Mikis Stasinopoulos [aut, cre, cph] (<https://orcid.org/0000-0003-2407-5704>), Robert Rigby [aut] (<https://orcid.org/0000-0003-3853-1707>), Fernanda De Bastiani [aut] (<https://orcid.org/0000-0001-8532-639X>), Julian Merder [ctb]

Maintainer: Mikis Stasinopoulos <d.stasinopoulos@gre.ac.uk>

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC, doi:10.1201/9780429298547. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, doi:10.18637/jss.v023.i07.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M. D., Rigby, R. A., and De Bastiani F., (2018) GAMLSS: a distributional regression approach, *Statistical Modelling*, Vol. **18**, pp, 248-273, SAGE Publications Sage India: New Delhi, India.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also `https://www.gamlss.com/`).

See Also

`gamlss`, `gamlss.family`

Examples

```
data(rent)
rent |> data_str()
```

`data_cor` *Plotting pairwise linear and partial correlations.*

Description

The function `data_cor` takes a data frame and plots the pairwise Pearson's correlation coefficients of all continuous variables in the data.

The function `data_pcor` takes a data frame and plots the pairwise partial Pearson's correlation coefficients of all continuous variables in the data.

The function `data_association` takes a data frame and plots the pairwise association coefficients of all variables in the data. For contituous against continuous variables it shows the absolute value of the Pearson's correlation coeficient, for categorical agaist categorical it shows Cramer's ϕ, for continuous agaist categorigal it fit an analyis of variance model and reports the square root of the R^2.

The functions `high_val` and `low_val` take the square matrix generated by the above thee functions and shows which paire-wise have a value larger or smaller respectively, than the value specified by the argument by `val`.

Usage

```
data_cor(data,  digits = 3, plot = TRUE, diag.off = TRUE,
        lower.tri.off = FALSE, method = c("square", "circle"),
        type = c("pearson", "kendall", "spearman"),
        outline.color = "gray", colors = c("blue", "white", "red"),
        legend.title = "Corr", title, ggtheme = theme_minimal(),
        tl.cex = 12, tl.col = "black", tl.srt = 45, lab = TRUE,
        lab_col = "black", lab_size = 3, circle.size = 20, seed = 123,
        percentage)

data_pcor(data, digits = 3, plot = TRUE, diag.off = TRUE,
        lower.tri.off = FALSE, method = c("square", "circle"),
        outline.color = "gray", colors = c("blue", "white", "red"),
        legend.title = "Corr", title, ggtheme = theme_minimal(),
        tl.cex = 12, tl.col = "black", tl.srt = 45, lab = TRUE,
        lab_col = "black", lab_size = 3, circle.size = 20, seed = 123,
        percentage)

data_association(data, digits = 3, plot = TRUE, diag.off = TRUE,
        lower.tri.off = FALSE, method = c("square", "circle"),
        outline.color = "gray", colors = c("blue", "white", "red"),
        legend.title = "Assoc", title, ggtheme = ggplot2::theme_minimal(),
        tl.cex = 12, tl.col = "black", tl.srt = 45, lab = TRUE,
       lab_col = "black", lab_size = 3, circle.size = 20, seed = 123, percentage)

high_val(table, val = 0.90, digits = 3, plot = FALSE, igraph = TRUE)

low_val(table, val = 0.05, digits = 3, plot = FALSE, igraph = TRUE)
```

Arguments

`data`	a data frame
`table`	a correlation table obtained by `data_cor(,plot=FALSE)` or `data_pcor(,plot=FALSE)`
`digits`	the digits for printing the correlation coefficients
`plot`	whether to plot or not
`diag.off`	whether to show the diagonal ellements
`lower.tri.off`	whether to show the lower part of the matrix
`method`	plotting in `"square"` or `"cicle"`
`type`	type of correlation c("pearson", "kendall", "spearman")
`outline.color`	the outline colour
`colors`	the range of colours
`legend.title`	title for the legend
`title`	the main tittle
`ggtheme`	the theme for the plot, see package **ggthemes** for more themes
`tl.cex`	the text size for the marginal labels
`tl.col`	the colour of the he marginal labels
`tl.srt`	the angle of the text in the bottom labels of the table
`lab`	whether to show the correlation coefficients in the table
`lab_col`	the colour of the lettering of the correlation coefficients

`lab_size` the size of the lettering of the correlation coefficients, increase (or decrease) if the defaul 3 is not appropriate

`circle.size` the size of the circles, increase (or decrease) if the defaul 20 is not appropriate

`percentage` this is for big data sets. if more tha a milion ony 10% is plotted, if from 100.00 to a milion, 20%, if 50.000 to 100.000, 50% otherwise 100% of the data.

`seed` Setting a seed value for selection of the percantage of data (for big data sets)

`val` the theshold value so if tha actul value is greater than `val` it will be reported in `high_val()`

`igraph` if in `high_val()`, the option `plot=TRUE` is set, then there are two options for plotting; i) `igraph::graph_from_data_frame()` with `igraph=TRUE` or ii) `networkD3::simpleNetwork` with `igraph=FALSE`

Value

creates a correlation matrix plot.

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/9780429298547.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also `https://www.gamlss.com/`).

See Also

mcor

Examples

```
data_cor(rent99)
Pearson.cor <-  data_cor(rent99, plot=FALSE)

data_pcor(rent99)
partial.cor  <-  data_cor(rent99, plot=FALSE)


high_val(partial.cor, val=0.5)
high_val(Pearson.cor, val=0.5)
```

data_dim *Function to get information from data.*

Description

This is a set of function are designed to help the user to deal with new data sets.

`data_dim()`: the class, the dimension and the % NA's in the data

`data_which_na()`: which variables have NA's and how many

`data_omit()`: omit the NA's from the data.

`data_names()`: The names of the variables in the data.

`data_shorter_names()`: abbriviate the names up to specified digits.

`data_rename()` renames some of of the variables.

Usage

```
data_dim(data)

data_which_na(data)

data_omit(data)

data_names(data)

data_shorter_names(data, max = 5, newnames)

data_rename(data, oldnames, newnames)
```

Arguments

data	a data frame
max	the maximum number of characters allowed, with default 5. Make sure that you are using enought characters otherwise you could end up with variables with the some name
newnames	New names if not abbreviation is required, as characters
oldnames	the old names as characters

Details

The function `data_dim()` gives the the class, the dimension and the % NA's in the data.

The function `data_which_na()` gives the number of missing observation for each variable in the data.

The function `data_omit()`: omits the NA's from the data.

The function `data_names()` gives the names of the variables.

The function `data_shorter_names()` takes the current names and abbreviates to `max` characters.

The function `data_rename()` renames variable from the `data`.

Value

The function `data_dim()` after printing gives the originasl data set.

The function `data_which_na()` prints the number of missing observation for each variable in the data and passes the original data set.

The function `data_omit()`: omits the NA's from the data and passes the new data set.

The function `data_names()` prints the names of the variables in the data andpasses the original data

The function `data_shorter_names()` takes the current names and abbreviates to `max` characters and return the data with shorter names.

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

`data_cor`

Examples

```
data_dim(rent)
data_which_na(rent)
data_omit(rent)
data_names(rent)
data_shorter_names(rent)
pp=data_rename(rent, c("R", "Fl"), c("rent", "floor"))
data_names(pp)
```

data_inter *Identifying Pair-Wise Interactions in the Data Frames*

Description

The function `data_inter()` is trying to identify pair-wise interations given the response variable using linear regression methodology. At the moment it works only with continuous reponse variables.

Usage

```
data_inter(data, response, weights, digits = 3, plot = TRUE,
        lower.tri.off = TRUE,
        method = c("circle", "square"), fit.method = c("linear", "nonlinear"),
        outline.color = "gray", colors = c("blue", "white", "red"),
        legend.title = "Inter", title, ggtheme = theme_minimal(),
        tl.cex = 12, tl.col = "black", tl.srt = 45, lab = TRUE,
        lab_col = "black", lab_size = 3, circle.size = 20, seed = 123,
        percentage)
```

Arguments

`data`	a data frame
`response`	the response variable
`weights`	prior weights
`digits`	the number of digits in the plot
`plot`	whether to plot the results
`lower.tri.off`	whether to show the lower part of the matrix
`method`	plotting in `"square"` or `"cicle"`
`fit.method`	whether in `"linear"` or `"nonlinear"`
`outline.color`	the outline colour
`colors`	the range of colours
`legend.title`	title for the legend
`title`	the main tittle
`ggtheme`	the theme for the plot, see package **ggthemes** for more themes
`tl.cex`	the text size for the marginal labels
`tl.col`	the colour of the he marginal labels
`tl.srt`	the angle of the text in the bottom labels of the table
`lab`	whether to show the correlation coefficients in the table
`lab_col`	the colour of the lettering of the correlation coefficients
`lab_size`	the size of the lettering of the correlation coefficients, increase (or decrease) if the defaul 3 is not appropriate

`circle.size` the size of the circles, increase (or decrease) if the defaul 20 is not appropriate

`percentage` the percentage of data to show if the observation number is too big

`seed` Setting a seed value for selection of the percantage of data (for big data sets)

Details

The function `data_inter()` uses the funcion `z_scores()` to standarized the continuous response variable and then uses linear model fits to establish whether the first order interactions between the the x's are singificant or not. It reports the significant level based on Chi-square tests. Note that for large data sets it uses the function `data_cut()` to cut randomnly the size of the data in order to use `ggplo2` graphs to plot it.

Typically for linear model first ortder interaction it fits the models `y~x1+x2` and `y~x1*x2`, respectively, and calculated significant level based on the difference in deviances. Under the H_o hypothesis the difference in deviances follow be a Chi-square distribution with degrees of freedom based on the difference of the degrees of freedom of the two fitted models.

Value

It produce a plot `plot=TRUE` or a square upper triangular table.

Note

The function `data_inter()` works only for continuous responses.

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/9780429298547.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also `https://www.gamlss.com/`).

See Also

`data_cor`

Examples

```
data_inter(rent[,-4,5], response=R)
```

data_leverage *Finding high leverage points*

Description

The function `data_leverage()` uses linear model methodology to identify unusual observations in the explnatory variables.

Usage

```
data_leverage(data, response, weights, quan.val = 0.99, annotate = TRUE,
     line.col = "steelblue4", point.col = "steelblue4", annot.col = "darkred",
     plot = TRUE, title, percentage, seed = 123, ...)
```

Arguments

`data`	a data frame
`response`	the response variable
`weights`	prior weights
`quan.val`	the quantile value to plot the line
`annotate`	whether to annotate the points
`line.col`	the colour of the horozontal line
`point.col`	the colour of the leverage points
`annot.col`	the colour of the annotated points
`plot`	whether to plot
`title`	whether to put a title
`percentage`	for very large data the percentage of points to use in the calculations
`seed`	the seed needed for selecteing a subset of data if only a percentage of points are plotted
`...`	for extra argument

Details

The method used is as follows. Fit a linear model to all explanatory variables in the data and calculate the leverge points.

Value

The function plot the leverge points unless `plot=FALSE` is used where in thos case the leverage points can be saved.

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in `https://www.gamlss.com/`.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, `https://www.jstatsoft.org/v23/i07/`.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also `https://www.gamlss.com/`).

See Also

`data_outliers`

Examples

```
data_leverage(rent, response=R)
```

`data_outliers` *Outlier identification*

Description

Those two function idententify outliers in variables in data

Usage

```
data_outliers(data, value = 4, min.distinct = 50, family = SHASHo)

y_outliers(var, value = 4, family = SHASH)
```

Arguments

`data`	a data frame
`var`	a continues variable
`value`	max value from which the absolute value of the z-scores should be greater to identify outliers
`min.distinct`	if a variable has less distinct values than `min.distinct` is excluded
`family`	the distribution family used for standardization

Details

the continuous variables are power transforemed and then standartised

Value

return a list

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

`data_names`

Examples

```
da <- rent99[,-2]
data_outliers(da)
```

`data_part` *A function to partition a data frame*

Description

The function `data_part()` takes a data set and creates a new identical data set with an extra factor called `partition` which can be used to allocate data in different data sets.

i) if `partition=2L` the factor has two levels `train`, and `test`.

ii) if `partition=3L` the factror has three levels `train`, for training data, `val` for validation data and `test` for test data.

iii) if `partition` > `4L` say `K` then the levela are "1", "2"..."K". The factor then cen be used for `K` cross validation sets.

the function `data_part_list()` in does similar things like the function `data_part()` but instead of adding a factor to the data creates a `list` with ellements the training, test and validation data sets. Note that this function do not allow k-fold cross-validation or bootsrapping indexing.

The function `data_part_index()` takes a `data.frame` and produces a list of indeces which then can be used to select diffetent sections of the data.

The function `data_cut()` takes a `data.frame` and selects randomly specified % of the data. For `data.frames` with more than 50.000 observations is automatically select part of the data. The function `data_cut()` is used a lot in the plotting routines to make sure that plots ate not take for ever.

Usage

```
data_part(data, partition = 2L, probs, setseed = 123, ...)

data_part_list(data, partition = 2, probs, setseed = 123, ...)

data_part_index(data, K = 2, bootstrap = FALSE)

data_cut(data, percentage, seed = 123)
```

Arguments

`data`	a `data.frame`
`partition`	2, 3 or a number less than 20
`K`	the number of partitions
`bootstrap`	whether bootstrap of k-fold cross validation samples
`probs`	probabilities for the random selection
`setseed`	setting the sead so the proccess can be repeated
`percentage`	The percentage of data to keep. If set, i.e. `percentage=0.5` only 50% are kept otherwise for large data set (more that 50.000) only percentage of data are kept.
`seed`	the `set.seed()` argument
`...`	extra arguments

Value

Both function produce a data frame. The function `data_part()` adds a factor `partition` while `data_rm1val()` removes variable with only one value.

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

`data_str`

Examples

```
da <- data_part(rent)
head(da)
mosaicplot(table(da$partition))

da.train <- subset(da, partition=="train")
da.test <- subset(da, partition=="test")
dim(da.train)
dim(da.test)


allda <-  data_part_list(rent)
dim(allda[[1]]) # training data
dim(allda[[2]]) # test data

index <-  data_part_index(rent)
 trdata<- rent[index[[1]],]
 tedata<- rent[index[[2]],]
 dim(trdata)
 dim(tedata)
 dim(trdata)[1]+dim(tedata)[1]
 dim(rent)
```

data_rm *Functions operating on variables in the data*

Description

There are several function operating on a `data.frame` and export a `data.frame`. The functions are

1) `data_rm()`: this function removes the variables specified by `vars` from the `data.frame`. Note that `vars` can take either character names or numbers.

2) `data_rm1val()`: This function looks for varables with a unique distinct value (most likely factors left from a previous `subset()` operation) and remove them form the data.

3) `data_exclude_class()`: This function looks for variable (columns) of a specified 'R' class and remove them from the data. The default class is "factor".

4) `data_only_continuous()`": This function pick up only the continuous variable in the `data.frame`.

5) `data_select()`": This function select only the variables in the `vars` list and save the data.

Usage

```
data_rm(data, vars)

data_rm1val(data)

data_exclude_class(data, class.out = "factor")

data_only_continuous(data)

data_select(data, vars)

data_rmNAvars(data)
```

Arguments

`data`	a data frame
`vars`	selected variables (columns from the data frame)
`class.out`	a specific variable class to be excluded form the data frame

Details

All the above functions can be used for piping i.e. `da |> data_rm1val()`.

Value

returm a `data.frame`

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

data_cor

Examples

```
library(gamlss)
da <- rent |> data_rm( vars=c("Sp", "Sm"))
head(da)

da<- rent |> data_exclude_class()
head(da)

da<- data_only_continuous(rent)
head(da)

da <- rent |> data_select( vars=c("R", "Fl", "A"))
head(da)
```

`data_scale` *Scalling Continuous Variables in Data*

Description

The function `data_scale()` takes a `data.frame` and creates a new data set with all continous variable standarised. The standardization can be

i) scale 0 variance 1 which is equivalent to have the options `scale.to="z-scores"` and `family="NO"`. That is the z-scores after fitting a normal distribution to a continuous variable

ii) A more general `z-score` using say `scale.to="z-scores"` and `family="SHASH"` in which case correction to the skewness and kurtosis is done to the specified variable or

iii) the range is resticted from zero to one, i.e. `scale.to="0to1"`.

Usage

```
data_scale(data, response, position.response = NULL,
          scale.to = c("z-scores", "0to1"), family = "NO",
          scale.response = FALSE)
```

Arguments

`data`	A data frame
`response`	The name of the response variable
`position.response`	or the position of the response variable in the data.
`scale.to`	how to scape by normalization, `scale.to="z-scores"` or range 0 to 1 `scale.to="0to1"`.
`family`	The family used in the standarization, defaul is `family="NO"` but any other continuous family from $-Inf$ to $+Inf$ will do i.e. `family="SHASH"`.
`scale.response`	whether to scale also the response. The default value is `scale.response=FALSE` because in GAMLSS we are hoping to get the right family for the response.

Value

A data frame is return with all continous variables standarised.

Author(s)

Mikis Stasinopolos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

`data_zscores`

Examples

```
rent[, -c(4,5)] |> data_scale(, response=R)|> head()
```

data_str *Function applied to data*

Description

his is a set of function are designed to help the user to deal with the structure of new data sets.

Usage

```
data_str(data, min.values = 100, min.levels = 10)

y_distinct(var)

data_distinct(data, get.distinct = FALSE, print=TRUE)

data_cha2fac(data, show.str = FALSE)

data_few2fac(data, min.levels = 10, show.str = FALSE)

data_int2num(data, min.values = 50, show.str = FALSE)

data_fac2num(data, vars)
```

Arguments

`data`	a data frame
`min.values`	the minimal value distinct values befor warning
`min.levels`	the minimal value distinct levels befor warning
`var`	a vector
`vars`	a character vector with names from the data
`get.distinct`	TRUE if you need to save the values FALSE if not not
`show.str`	whether to show the structure
`print`	TRUE or FALSE

Details

The function `data_str()` gives the structure of the data set.

The function `data_distinct()` gives the distinct values of the vectors in the data set

The function `y_distinct()` gives the distinct values of single vector

The function `data_cha2fac()` tranforms all character vectors to factors

The function data_few2fac() transform all vectors with fewer values than min.levels into factors

The function data_int2num() transform all integer vectors with more values than min.values into numeric

The function data_fac2num() transform sellected variables factors into numeric vectors

Author(s)

Mikis Stasinopoulos, Bob Rigby and Fernanda De Bastiani

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

data_dim

Examples

```
data_str(rent)
data_distinct(rent)
data_cha2fac(rent)
data_few2fac(rent)
data_int2num(rent)
```

---

data_void *Finding the Percentage of Empty Spaces*

Description

The function void() is looking for the % of empty spaces in the direction of two variables x and y.

The function data_void() is looking pair-wise for empty spaces in all the continuous variables in the data set.

Usage

```
data_void(data, digits = 3, plot = TRUE, diag.off = TRUE,
        lower.tri.off = FALSE,
        method = c("square", "circle"),
        outline.color = "gray", colors = c("blue", "white", "red"),
        legend.title = "Void", title, ggtheme = ggplot2::theme_minimal(),
        tl.cex = 12, tl.col = "black", tl.srt = 45, lab = TRUE,
        lab_col = "black", lab_size = 3, circle.size = 20, seed = 123,
        percentage)

void(x, y, plot = TRUE, print = TRUE, table.length)
```

Arguments

`data`	A data frame
`digits`	the digits for printing the correlation coefficients
`plot`	whether to plot or not
`diag.off`	whether to show the diagonal ellements
`lower.tri.off`	whether to show the lower part of the matrix
`method`	plotting in `"square"` or `"cicle"`
`outline.color`	the outline colour
`colors`	the range of colours
`legend.title`	title for the legend
`title`	the main tittle
`ggtheme`	the theme for the plot, see package **ggthemes** for more themes
`tl.cex`	the text size for the marginal labels
`tl.col`	the colour of the he marginal labels
`tl.srt`	the angle of the text in the bottom labels of the table
`lab`	whether to show the correlation coefficients in the table
`lab_col`	the colour of the lettering of the correlation coefficients
`lab_size`	the size of the lettering of the correlation coefficients, increase (or decrease) if the defaul 3 is not appropriate
`circle.size`	the size of the circles, increase (or decrease) if the defaul 20 is not appropriate
`seed`	the `set.seed()` value
`percentage`	the percentage of data to show if the observation number is too big
`x`	the first variable in `void()`
`y`	the second variable in `void()`
`print`	whether to print the results
`table.length`	the table length (if siging is calculated automatically)

Details

The functions void() and data_void() work with discretising the data in the x and y direction and then calculate the % of zeros. By discretising the data we mean cut both variable x variables abd y, at an equal spaced grid of k points and create a (k x k) dimenstional matrix containing the number of data points in the grid. The problem thought, with any attempt to calculated the % of empty spaces is that by increasing k) in the x and y directions would resulst more zeros cells and therefore more % empty spaces. To avoid this we need a way to stop the discretazation at a stage before the data become too sparce. The waythis is done in tjhe current function is the following;

i) If the n points (x,y) are randomly allocated we would expect the number of counts in the cells of the matrix of a discretised two dimestional data set to be Poisson distributed with a probability for zeros equal to $exp(-\mu)$ where μ is the mean of the Poisson distribution. That is, under the null hypothesis that the n points are spead randomly we expect some of the cell to be zero with probability $exp(-\mu)$. Given n the number of obsrvations, we can use this information to find out at which disretation point k we should stop.

ii) To identify at which stage k we should stop for given number of observations say n, we have genarated randomly from a uniform distribution n values for x and y. We use those values to calculate at which point k this will give a probability of zero close to 0.05. We calculate those probabilities using $exp(-xbar)$ where xbar is the mean of the cells. By doing this we found that that the following is holding; $\log k = -0.516 + 0.498 \log n$. This equation provide us with an easy way to calculate k given n.

Value

It produce a value between zero and 1.

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/9780429298547.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC. doi:10.1201/b21973

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

data_cor

Examples

```
void(rent$A, rent$Fl)
data_void(rent)
```

`data_xyplot` *Plotting the response against the explanatory variables*

Description

The function `data_xyplot()` plots the response against all other variables in a given data set.

The function `data_plot()` plots all variables individually.

The function `data_bucket()` plots the bucket plot for all continuous variables.

The function `data_zscores()` calculates and plots the z-scores (obtained after fitting the SHASHo distribution) for all continuous variables.

The function `y_zscores()` calculates and plots the z-scores (obtained after fitting the SHASHo distribution) for a single variable.

The function `data_response()` calculates and plots the z-scores (obtained after fitting the SHASHo distribution) for the response variable.

Usage

```
data_xyplot(data, response, point.size = 0.5, nrow = NULL,
           ncol = NULL, percentage, seed = 123,
           max.levels = 10, plots.per.page = 9, one.by.one = FALSE, title, ...)

data_plot(data, value = 3, hist.col = "black", hist.fill = "white",
           dens.fill = "#FF6666", nrow = NULL, ncol = NULL,
           percentage, seed = 123,
           plot.hist = TRUE, plots.per.page = 9,
           one.by.one = FALSE,
           title, ...)

data_bucket(data, value = 3, max.levels = 20,
           nrow = NULL, ncol = NULL, plots.per.page = 9,
           one.by.one = FALSE, title, percentage, seed = 123,
           ...)

y_zscores(x, family = SHASHo, plot = TRUE, hist = FALSE, ...)

data_zscores(data, plot = TRUE, hist=FALSE, value = 3, family = SHASHo,
          max.levels = 10, hist.col = "black", hist.fill = "white",
          dens.fill = "#FF6666", nrow = NULL, ncol = NULL,
          plots.per.page = 9, one.by.one = FALSE, title, percentage,
          seed = 123,...)

data_response(data, response, plot = TRUE, percentage, seed = 123)
```

Arguments

`data`	a data frame
`x`	a single variable
`family`	a gamlss distribution family (continuous)
`response`	the respose variable should be in the data
`point.size`	the size of points in scatter plots
`nrow`	the number of rows in the plot
`ncol`	the number of columns in the plot
`plots.per.page`	maximu plots per page
`one.by.one`	whether plotted individually
`value`	value to identify outliers if `y_dots` is used i.e. for upper tail an outliers is if it is greater than Q_3+value*IQ
`hist.col`	the colour of lines of the histogram, if `plot.hist=TRUE`
`hist.fill`	the colour of the histogram, if `plot.hist=TRUE`
`dens.fill`	the color of the density plot, if `plot.hist=TRUE`
`plot.hist`	whether to use `y_dots()` or `y_hist()` for the continuous variables
`plot`	whether to plot
`hist`	whether histiogram or dot plot
`max.levels`	excludes from plotting bucket plots for variables with less than `max.levels`, distinct values
`title`	title of the plot
`percentage`	if set, i.e. 0.50, plots a portotion of data otherwise for big data sets greater than 50.000 observartions it plots a porpotion
`seed`	the `set.seed()` argument
`...`	other arguments

Details

The function `data_xyplot()` it takes a data frame and plot all the explanarory variables against the response.

The function `data_plot()` it takes a data frame and plot all variables against the response. The continuous are plotted using `y_dots()` or `y_hist()` while the factors and integer as bar plots.

Value

Plots of the data

Author(s)

Mikis Stasinopoulos

References

Rigby, R. A. and Stasinopoulos D. M. (2005). Generalized additive models for location, scale and shape,(with discussion), *Appl. Statist.*, **54**, part 3, pp 507-554.

Rigby, R. A., Stasinopoulos, D. M., Heller, G. Z., and De Bastiani, F. (2019) *Distributions for modeling location, scale, and shape: Using GAMLSS in R*, Chapman and Hall/CRC. An older version can be found in https://www.gamlss.com/.

Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, https://www.jstatsoft.org/v23/i07/.

Stasinopoulos D. M., Rigby R.A., Heller G., Voudouris V., and De Bastiani F., (2017) *Flexible Regression and Smoothing: Using GAMLSS in R*, Chapman and Hall/CRC.

Stasinopoulos, M.D., Kneib, T., Klein, N., Mayr, A. and Heller, G.Z., (2024). *Generalized Additive Models for Location, Scale and Shape: A Distributional Regression Approach, with Applications* (Vol. **56**). Cambridge University Press.

(see also https://www.gamlss.com/).

See Also

data_names

Examples

```
da <- rent99[,-2]
data_xyplot(da, rent)
data_plot(da)
y_zscores(da$rent)
data_response(da, response=rent)
```

Index