Package 'explore'

January 30, 2022

Type Package

Title Simplifies Exploratory Data Analysis

Version 0.8.0

Author Roland Krasser

Maintainer Roland Krasser <roland.krasser@gmail.com>

Description Interactive data exploration with one line of code or use an easy
to remember set of tidy functions for exploratory data analysis.
Introduces three main verbs. explore() to graphically explore a variable or
table, describe() to describe a variable or table and report() to create an
automated report.

License GPL-3

Encoding UTF-8

URL

Imports assertthat, broom, dplyr, DT, forcats, ggplot2 (>= 3.0.0),
gridExtra, magrittr, MASS, rlang, rmarkdown, rpart, rpart.plot,
shiny, stats, stringr, tibble, tidyr

RoxygenNote 7.1.2

Suggests knitr, palmerpenguins, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-30 15:00:02 UTC

R topics documented:

---

add_random_var *Add a random variable to dataset*

Description

Add a random variable to dataset

Usage

```
add_random_var(
  data,
  name = "random",
  min_val = 0,
  max_val = 1,
  overwrite = TRUE
)
```

Arguments

data	A dataset
name	Name of new variable (as string)
min_val	Minimum random integers
max_val	Maximum random integers
overwrite	Can new random variable overwrite an existing variable in dataset?

Value

Dataset containing new random variable

Examples

```
add_random_var(iris)
add_random_var(iris, name = "random_var")
add_random_var(iris, min_val = 1, max_val = 10)
add_random_var(iris, min_val = 1, max_val = 100, overwrite = FALSE)
```

---

balance_target *Balance target variable*

Description

Balances the target variable in your dataset. Target must be 0/1, FALSE/TRUE ore no/yes

Usage

```
balance_target(data, target, min_prop = 0.1)
```

Arguments

data	A dataset
target	Target variable (0/1, TRUE/FALSE, yes/no)
min_prop	Minimum proportion of one of the target categories

Value

Data

Examples

```
iris$is_versicolor <- ifelse(iris$Species == "versicolor", 1, 0)
balanced <- balance_target(iris, target = is_versicolor, min_prop = 0.5)
describe(balanced, is_versicolor)
```

---

clean_var	*Clean variable*

Description

Clean variable (replace NA values, set min_val and max_val)

Usage

```
clean_var(
  data,
  var,
  na = NA,
  min_val = NA,
  max_val = NA,
  max_cat = NA,
  rescale01 = FALSE,
  simplify_text = FALSE,
  name = NA
)
```

Arguments

data	A dataset
var	Name of variable
na	Value that replaces NA
min_val	All values < min_val are converted to min_val (var numeric or character)
max_val	All values > max_val are converted to max_val (var numeric or character)
max_cat	Maximum number of different factor levels for categorical variable (if more, .OTHER is added)

rescale01	Rescale into value between 0 and 1 (var must be numeric)
simplify_text	if TRUE, a character variable is simplified (trim, upper, ...)
name	New name of variable (as string)

Value

Dataset

Examples

```
clean_var(iris, Sepal.Width, max_val = 3.5, name = "sepal_width")
```

---

count_pct	*Adds percentage to dplyr::count()*

Description

Adds variables total and pct (percentage) to dplyr::count()

Usage

```
count_pct(data, ...)
```

Arguments

data	A dataset
...	Other parameters passed to count()

Value

Dataset

Examples

```
count_pct(iris, Species)
```

create_fake_data   *Create fake data*

Description

Fake data that can be used for unit-testing or demonstration

Usage

```
create_fake_data(
  obs = 1000,
  target_name = "target_ind",
  factorise_target = FALSE,
  target1_prob = 0.5,
  add_extreme = TRUE,
  flip_gender = FALSE,
  add_id = FALSE,
  seed = 123
)
```

Arguments

obs	Number of observations
target_name	Variable name of target
factorise_target	
	Should target variable be factorised? (from 0/1 to facotr no/yes)?
target1_prob	Probability that buy = 1
add_extreme	Add an obervation with extreme values?
flip_gender	Should Male/Female be flipped in data?
add_id	Add an id-variable to data?
seed	Seed for randomization

Details

Variables in dataset:

- id = Identifier
- period = Year & Month (YYYYMM)
- city_ind = Indicating if customer is residing in a city (1 = yes, 0 = no)
- female_ind = Gender of customer is female (1 = yes, 0 = no)
- fixedvoice_ind = Customer has a fixed voice product (1 = yes, 0 = no)
- fixeddata_ind = Customer has a fixed data product (1 = yes, 0 = no)
- fixedtv_ind = Customer has a fixed tv product (1 = yes, 0 = no)

- mobilevoice_ind = Customer has a mobile voice product (1 = yes, 0 = no)
- mobiledata_ind = Customer has a mobile data product (1 = yes, 0 = no)
- bbi_speed_ind = Customer has a Broadband Internet (BBI) with speed
- bbi_usg_gb = Broadband Internet (BBI) usage in Gigabyte (GB) last month
- hh_single = Expected to be a Single Household (1 = yes, 0 = no)

Target in dataset:

- target_ind (may be renamed) = Did customer buy a new product in next month? (1 = yes, 0 = no)

Value

A dataframe

---

create_random_data *Create random data*

Description

Random data that can be used for unit-testing or demonstration

Usage

```
create_random_data(
  obs = 1000,
  vars = 10,
  target_name = "target_ind",
  factorise_target = FALSE,
  target1_prob = 0.5,
  add_id = TRUE,
  seed = 123
)
```

Arguments

obs	Number of observations
vars	Number of variables
target_name	Variable name of target
factorise_target	
	Should target variable be factorised? (from 0/1 to facotr no/yes)?
target1_prob	Probability that buy = 1
add_id	Add an id-variable to data?
seed	Seed for randomization

Details

Variables in dataset:

- id = Identifier
- var_X = variable containing values between 0 and 100

Target in dataset:

- target_ind (may be renamed) = random values (1 = yes, 0 = no)

Value

A dataframe

---

data_dict_md	*Create a data dictionary Markdown file*

Description

Create a data dictionary Markdown file

Usage

```
data_dict_md(
  data,
  title = "",
  description = NA,
  output_file = "data_dict.md",
  output_dir
)
```

Arguments

data	A dataframe (data dictionary for all variables)
title	Title of the data dictionary
description	Detailed description of variables in data (dataframe with columns 'variable' and 'description')
output_file	Output filename for Markdown file
output_dir	Directory where the Markdown file is saved

Value

Create Markdown file

Examples

```
# Data dictionary of a dataframe
data_dict_md(iris,
             title = "iris flower data set",
             output_dir = tempdir())

# Data dictionary of a dataframe with additional description of variables
description <- data.frame(
                  variable = c("Species"),
                  description = c("Species of Iris flower"))
data_dict_md(iris,
             title = "iris flower data set",
             description = description,
             output_dir = tempdir())
```

---

decrypt   *decrypt text*

Description

decrypt text

Usage

```
decrypt(text, codeletters = c(toupper(letters), letters, 0:9), shift = 18)
```

Arguments

text   A text (character)

codeletters   A string of letters that are used for decryption

shift   Number of elements shifted

Value

Decrypted text

Examples

```
decrypt("zw336 E693v")
```

---

describe   *Describe a dataset or variable*

Description

Describe a dataset or variable (depending on input parameters)

Usage

```
describe(data, var, n, target, out = "text", ...)
```

Arguments

data	A dataset
var	A variable of the dataset
n	Weights variable for count-data
target	Target variable (0/1 or FALSE/TRUE)
out	Output format ("text"|"list") of variable description
...	Further arguments

Value

Description as table, text or list

Examples

```
# Load package
library(magrittr)

# Describe a dataset
iris %>% describe()

# Describe a variable
iris %>% describe(Species)
iris %>% describe(Sepal.Length)
```

---

describe_all *Describe all variables of a dataset*

Description

Describe all variables of a dataset

Usage

```
describe_all(data = NA, out = "large")
```

Arguments

data	A dataset
out	Output format ("small"|"large")

Value

Dataset (tibble)

Examples

```
describe_all(iris)
```

---

describe_cat *Describe categorical variable*

Description

Describe categorical variable

Usage

```
describe_cat(data, var, n, max_cat = 10, out = "text", margin = 0)
```

Arguments

data	A dataset
var	Variable or variable name
n	Weights variable for count-data
max_cat	Maximum number of categories displayed
out	Output format ("text"|"list")
margin	Left margin for text output (number of spaces)

Value

Description as text or list

Examples

```
describe_cat(iris, Species)
```

---

describe_num   *Describe numerical variable*

Description

Describe numerical variable

Usage

```
describe_num(data, var, n, out = "text", margin = 0)
```

Arguments

data	A dataset
var	Variable or variable name
n	Weights variable for count-data
out	Output format ("text"|"list")
margin	Left margin for text output (number of spaces)

Value

Description as text or list

Examples

```
describe_num(iris, Sepal.Length)
```

---

describe_tbl *Describe table*

Description

Describe table (e.g. number of rows and columns of dataset)

Usage

```
describe_tbl(data, n, target, out = "text")
```

Arguments

data	A dataset
n	Weigts variable for count-data
target	Target variable (binary)
out	Output format ("text"|"list")

Value

Description as text or list

Examples

```
describe_tbl(iris)

iris$is_virginica <- ifelse(iris$Species == "virginica", 1, 0)
describe_tbl(iris, is_virginica)
```

---

dwh_connect *connect to DWH*

Description

connect to datawarehouse (DWH) using ODBC

Usage

```
dwh_connect(dsn, user = NA, pwd = NA, pwd_crypt = FALSE, ...)
```

Arguments

dsn	DSN string
user	user name
pwd	password of user
pwd_crypt	is password encryption used?
...	Further arguments to be passed to DBI::dbConnect()

Value

connection

Examples

```
## Not run:
con <- dwh_connect(dsn = "DWH1", user = "u12345")

## End(Not run)
```

dwh_disconnect   *disconnect from DWH*

Description

disconnect from datawarehouse (DWH) using a ODBC connection

Usage

```
dwh_disconnect(connection, ...)
```

Arguments

connection	channel (ODBC connection)
...	Further arguments to be passed to DBI::dbDisconnect()

Examples

```
## Not run:
dwh_disconnect(con)

## End(Not run)
```

dwh_fastload   *write data to a DWH table*

Description

write data fast to a DWH table using a ODBC connection Function uses packages DBI/odbc to write data faster than RODBC Connects, writes data and disconnects

Usage

```
dwh_fastload(data, dsn, table, overwrite = FALSE, append = FALSE, ...)
```

Arguments

data	dataframe
dsn	DSN string
table	table name (character string)
overwrite	Overwrite table if already exist
append	Append data to table
...	Further arguments to be passed to DBI::dbConnect()

Value

status

Examples

```
## Not run:
dwh_fastload(data, "DWH", "database.table_test")

## End(Not run)
```

---

dwh_read_data   *read data from DWH*

Description

read data from DWH using a ODBC connection

Usage

```
dwh_read_data(connection, sql, names_lower = TRUE, ...)
```

Arguments

connection	DWH connection
sql	sql (character string)
names_lower	convert field names to lower (default = TRUE)
...	Further arguments to be passed to DBI::dbGetQuery()

Value

dataframe containing table data

Examples

```
## Not run:
dwh_read_data(con, "select * from database.table_test")

## End(Not run)
```

---

dwh_read_table   *read a table from DWH*

Description

read a table from DWH using a ODBC connection

Usage

```
dwh_read_table(connection, table, names_lower = TRUE, ...)
```

Arguments

connection	DWH connection
table	table name (character string)
names_lower	convert field names to lower (default = TRUE)
...	Further arguments to be passed to DBI::dbGetQuery()

Value

dataframe containing table data

Examples

```
## Not run:
dwh_read_table(con, "database.table_test")

## End(Not run)
```

---

encrypt   *encrypt text*

Description

encrypt text

Usage

```
encrypt(text, codeletters = c(toupper(letters), letters, 0:9), shift = 18)
```

Arguments

text	A text (character)
codeletters	A string of letters that are used for encryption
shift	Number of elements shifted

Value

Encrypted text

Examples

```
encrypt("hello world")
```

| explain_logreg | *Explain a binary target using a logistic regression (glm). Model cho-sen by AIC in a Stepwise Algorithm (MASS::stepAIC).* |

Description

Explain a binary target using a logistic regression (glm). Model chosen by AIC in a Stepwise Algorithm (MASS::stepAIC).

Usage

```
explain_logreg(data, target, out = "tibble", ...)
```

Arguments

data	A dataset
target	Target variable (binary)
out	Output of the function: "tibble"	"model"
...	Further arguments

Value

Dataset with results (term, estimate, std.error, z.value, p.value) or the model (if out = "model")

Examples

```
data <- iris
data$is_versicolor <- ifelse(iris$Species == "versicolor", 1, 0)
data$Species <- NULL
explain_logreg(data, target = is_versicolor)
```

---

explain_tree   *Explain a target using a simple decision tree (classification or regres-
                              sion)*

Description

Explain a target using a simple decision tree (classification or regression)

Usage

```
explain_tree(
  data,
  target,
  n,
  max_cat = 10,
  max_target_cat = 5,
  maxdepth = 3,
  minsplit = NA,
  cp = 0,
  weights = NA,
  size = 0.7,
  out = "plot",
  ...
)
```

Arguments

data	A dataset
target	Target variable
n	weigths variable (for count data)
max_cat	Drop categorical variables with higher number of levels
max_target_cat	Maximum number of categories to be plotted for target (except NA)
maxdepth	Maximal depth of the tree (rpart-parameter)
minsplit	The minimum number of observations that must exist in a node to split.
cp	Complexity parameter (rpart-parameter)
weights	Vector containing weight of each observation (rpart-parameter). Can not be used in combination with parameter n (variable containing weight for count-data)
size	Textsize of plot
out	Output of function: "plot"	"model"
...	Further arguments

Value

Plot or additional the model (if out = "model")

Examples

```
data <- iris
data$is_versicolor <- ifelse(iris$Species == "versicolor", 1, 0)
data$Species <- NULL
explain_tree(data, target = is_versicolor)
```

---

explore   *Explore a dataset or variable*

Description

Explore a dataset or variable

Usage

```
explore(
  data,
  var,
  var2,
  n,
  target,
  split,
  min_val = NA,
  max_val = NA,
  auto_scale = TRUE,
  na = NA,
  ...
)
```

Arguments

data	A dataset
var	A variable
var2	A variable for checking correlation
n	A Variable for number of observations (count data)
target	Target variable (0/1 or FALSE/TRUE)
split	Split by target variable (FALSE/TRUE)
min_val	All values < min_val are converted to min_val
max_val	All values > max_val are converted to max_val
auto_scale	Use 0.2 and 0.98 quantile for min_val and max_val (if min_val and max_val are not defined)
na	Value to replace NA
...	Further arguments (like flip = TRUE/FALSE)

Value

Plot object

Examples

```
## Launch Shiny app (in interactive R sessions)
if (interactive())  {
    explore(iris)
}

## Explore grafically

# Load library
library(magrittr)

# Explore a variable
iris %>% explore(Species)
iris %>% explore(Sepal.Length)
iris %>% explore(Sepal.Length, min_val = 4, max_val = 7)

# Explore a variable with a target
iris$is_virginica <- ifelse(iris$Species == "virginica", 1, 0)
iris %>% explore(Species, target = is_virginica)
iris %>% explore(Sepal.Length, target = is_virginica)

# Explore correlation between two variables
iris %>% explore(Species, Petal.Length)
iris %>% explore(Sepal.Length, Petal.Length)

# Explore correlation between two variables and split by target
iris %>% explore(Sepal.Length, Petal.Length, target = is_virginica)
```

---

explore_all   *Explore all variables*

Description

Explore all variables of a dataset (create plots)

Usage

```
explore_all(data, n, target, ncol = 2, split = TRUE)
```

Arguments

data	A dataset
n	Weights variable (only for count data)
target	Target variable (0/1 or FALSE/TRUE)

ncol	Layout of plots (number of columns)
split	Split by target (TRUE|FALSE)

Value

Plot

Examples

```
explore_all(iris)

iris$is_virginica <- ifelse(iris$Species == "virginica", 1, 0)
explore_all(iris, target = is_virginica)
```

---

explore_bar	*Explore categorical variable using bar charts*

Description

Create a barplot to explore a categorical variable. If a target is selected, the barplot is created for all levels of the target.

Usage

```
explore_bar(
  data,
  var,
  target,
  flip = NA,
  title = "",
  numeric = NA,
  max_cat = 30,
  max_target_cat = 5,
  legend_position = "right",
  label,
  label_size = 2.7,
  ...
)
```

Arguments

data	A dataset
var	variable
target	target (can have more than 2 levels)
flip	Should plot be flipped? (change of x and y)
title	Title of the plot (if empty var name)

numeric	Display variable as numeric (not category)
max_cat	Maximum number of categories to be plotted
max_target_cat	Maximum number of categories to be plotted for target (except NA)
legend_position	
	Position of the legend ("bottom"|"top"|"none")
label	Show labels? (if empty, automatic)
label_size	Size of labels
...	Further arguments

Value

Plot object (bar chart)

---

explore_cor   *Explore the correlation between two variables*

Description

Explore the correlation between two variables

Usage

```
explore_cor(
  data,
  x,
  y,
  target,
  bins = 8,
  min_val = NA,
  max_val = NA,
  auto_scale = TRUE,
  title = NA,
  color = "grey",
  ...
)
```

Arguments

data	A dataset
x	Variable on x axis
y	Variable on y axis
target	Target variable (categorical)
bins	Number of bins
min_val	All values < min_val are converted to min_val

max_val	All values > max_val are converted to max_val
auto_scale	Use 0.2 and 0.98 quantile for min_val and max_val (if min_val and max_val are not defined)
title	Title of the plot
color	Color of the plot
...	Further arguments

Value

Plot

Examples

```
explore_cor(iris, x = Sepal.Length, y = Sepal.Width)
```

---

explore_count	*Explore count data (categories + frequency)*

Description

Create a plot to explore count data (categories + freuency) Variable named 'n' is auto detected as Frequency

Usage

```
explore_count(
  data,
  cat,
  n,
  target,
  pct = FALSE,
  split = TRUE,
  title = NA,
  numeric = FALSE,
  max_cat = 30,
  max_target_cat = 5,
  flip = NA
)
```

Arguments

data	A dataset (categories + frequency)
cat	Numerical variable
n	Number of observations (frequency)
target	Target variable

pct	Show as percent?
split	Split by target?
title	Title of the plot
numeric	Display variable as numeric (not category)
max_cat	Maximum number of categories to be plotted
max_target_cat	Maximum number of categories to be plotted for target (except NA)
flip	Flip plot? (for categorical variables)

Value

Plot object

Examples

```
library(dplyr)
iris %>%
  count(Species) %>%
  explore_count(Species)
```

---

explore_density   *Explore density of variable*

Description

Create a density plot to explore numerical variable

Usage

```
explore_density(
  data,
  var,
  target,
  title = "",
  min_val = NA,
  max_val = NA,
  color = "grey",
  auto_scale = TRUE,
  max_target_cat = 5,
  ...
)
```

Arguments

`data`	A dataset
`var`	Variable
`target`	Target variable (0/1 or FALSE/TRUE)
`title`	Title of the plot (if empty var name)
`min_val`	All values < min_val are converted to min_val
`max_val`	All values > max_val are converted to max_val
`color`	Color of plot
`auto_scale`	Use 0.02 and 0.98 percent quantile for min_val and max_val (if min_val and max_val are not defined)
`max_target_cat`	Maximum number of levels of target shown in the plot (except NA).
`...`	Further arguments

Value

Plot object (density plot)

Examples

```
explore_density(iris, "Sepal.Length")
iris$is_virginica <- ifelse(iris$Species == "virginica", 1, 0)
explore_density(iris, Sepal.Length, target = is_virginica)
```

`explore_shiny`	*Explore dataset interactive*

Description

Launches a shiny app to explore a dataset

Usage

```
explore_shiny(data, target)
```

Arguments

`data`	A dataset
`target`	Target variable (0/1 or FALSE/TRUE)

Examples

```
# Only run examples in interactive R sessions
if (interactive()) {
   explore_shiny(iris)
}
```

---

explore_targetpct   *Explore variable + binary target (values 0/1)*

Description

Create a plot to explore relation between a variable and a binary target as target percent. The target variable is choosen automatically if possible (name starts with 'target')

Usage

```
explore_targetpct(
  data,
  var,
  target = NULL,
  title = NULL,
  min_val = NA,
  max_val = NA,
  auto_scale = TRUE,
  na = NA,
  flip = NA,
  ...
)
```

Arguments

data	A dataset
var	Numerical variable
target	Target variable (0/1 or FALSE/TRUE)
title	Title of the plot
min_val	All values < min_val are converted to min_val
max_val	All values > max_val are converted to max_val
auto_scale	Use 0.2 and 0.98 quantile for min_val and max_val (if min_val and max_val are not defined)
na	Value to replace NA
flip	Flip plot? (for categorical variables)
...	Further arguments

Value

Plot object

Examples

```
iris$target01 <- ifelse(iris$Species == "versicolor",1,0)
explore_targetpct(iris)
```

---

explore_tbl *Explore table*

Description

Explore a table. Plots variable types, variables with no variance and variables with NA

Usage

```
explore_tbl(data, n)
```

Arguments

data	A dataset
n	Weight variable for count data

Examples

```
explore_tbl(iris)
```

---

format_num_auto *Format number as character string (auto)*

Description

Formats a number depending on the value as number with space, scientific or big number as k (1 000), M (1 000 000) or B (1 000 000 000)

Usage

```
format_num_auto(number = 0, digits = 1)
```

Arguments

number	A number (integer or real)
digits	Number of digits

Value

Formated number as text

Examples

```
format_num_kMB(5500, digits = 2)
```

---

format_num_kMB   *Format number as character string (kMB)*

Description

Formats a big number as k (1 000), M (1 000 000) or B (1 000 000 000)

Usage

```
format_num_kMB(number = 0, digits = 1)
```

Arguments

number	A number (integer or real)
digits	Number of digits

Value

Formated number as text

Examples

```
format_num_kMB(5500, digits = 2)
```

---

format_num_space   *Format number as character string (space as big.mark)*

Description

Formats a big number using space as big.mark (1000 = 1 000)

Usage

```
format_num_space(number = 0, digits = 1)
```

Arguments

number	A number (integer or real)
digits	Number of digits

Value

Formated number as text

Examples

```
format_num_space(5500, digits = 2)
```

---

format_target   *Format target*

Description

Formats a target as a 0/1 variable. If target is numeric, 1 = above average.

Usage

```
format_target(target)
```

Arguments

target   Variable as vector

Value

Formated target

Examples

```
iris$is_virginica <- ifelse(iris$Species == "virginica", "yes", "no")
iris$target <- format_target(iris$is_virginica)
table(iris$target)
```

---

format_type   *Format type description*

Description

Format type description of varable to 3 letters (int|dbl|lgl|chr|dat)

Usage

```
format_type(type)
```

Arguments

type   Type description ("integer", "double", "logical", character", "date")

Value

Formated type description (int|dbl|lgl|chr|dat)

Examples

```
format_type(typeof(iris$Species))
```

| get_nrow | *Get number of rows for a grid plot (deprecated, use total_fig_height()* |
| | *instead)* |

Description

Get number of rows for a grid plot (deprecated, use total_fig_height() instead)

Usage

```
get_nrow(varnames, exclude = 0, ncol = 2)
```

Arguments

varnames	List of variables to be plotted
exclude	Number of variables that will be excluded from plot
ncol	Number of columns (default = 2)

Value

Number of rows

Examples

```
get_nrow(names(iris), ncol = 2)
```

| get_type | *Return type of variable* |

Description

Return value of typeof, except if variable contains <hide>, then return "other"

Usage

```
get_type(var)
```

Arguments

| var | A vector (dataframe column) |

Value

Value of typeof or "other"

Examples

```
get_type(iris$Species)
```

---

get_var_buckets | *Put variables into "buckets" to create a set of plots instead one large plot*

Description

Put variables into "buckets" to create a set of plots instead one large plot

Usage

```
get_var_buckets(data, bucket_size = 100, var_name_target = NA, var_name_n = NA)
```

Arguments

data   A dataset

bucket_size   Maximum number of variables in one bucket

var_name_target

                Name of the target variable (if defined)

var_name_n   Name of the weight (n) variable (if defined)

Value

Buckets as a list

Examples

```
get_var_buckets(iris)
get_var_buckets(iris, bucket_size = 2)
get_var_buckets(iris, bucket_size = 2, var_name_target = "Species")
```

---

guess_cat_num | *Return if variable is categorial or nomerical*

Description

Guess if variable is categorial or numerical based on name, type and values of variable

Usage

```
guess_cat_num(var, descr)
```

Arguments

var   A vector (dataframe column)

descr   A description of the variable (optional)

Value

"cat" (categorial), "num" (numerical) or "oth" (other)

Examples

```
guess_cat_num(iris$Species)
```

---

plot_legend_targetpct *Plots a legend that can be used for explore_all with a binary target*

Description

Plots a legend that can be used for explore_all with a binary target

Usage

```
plot_legend_targetpct(border = TRUE)
```

Arguments

border   Draw a border?

Value

Base plot '@importFrom graphics legend par plot

Examples

```
plot_legend_targetpct(border = TRUE)
```

---

plot_text   *Plot a text*

Description

Plots a text (base plot) and let you choose text-size and color

Usage

```
plot_text(text = "hello world", size = 1.2, color = "black")
```

Arguments

text   Text as string
size   Text-size
color   Text-color

Value

Plot

Examples

```
plot_text("hello", size = 2, color = "red")
```

---

plot_var_info   *Plot a variable info*

Description

Creates a ggplot with the variable-name as title and a text

Usage

```
plot_var_info(data, var, info = "")
```

Arguments

data	A dataset
var	Variable
info	Text to plot

Value

Plot (ggplot)

---

replace_na_with   *Replace NA*

Description

Replace NA values of a variable in a dataframe

Usage

```
replace_na_with(data, var_name, with)
```

Arguments

data	A dataframe
var_name	Name of variable where NAs are replaced
with	Value instead of NA

Value

Updated dataframe

Examples

```
data <- data.frame(nr = c(1,2,3,NA,NA))
replace_na_with(data, "nr", 0)
```

---

report   *Generate a report of all variables*

Description

Generate a report of all variables If target is defined, the relation to the target is reported

Usage

```
report(data, n, target, split = TRUE, output_file, output_dir)
```

Arguments

data	A dataset
n	Weights variable for count data
target	Target variable (0/1 or FALSE/TRUE)
split	Split by target? (TRUE/FALSE)
output_file	Filename of the html report
output_dir	Directory where to save the html report

Examples

```
if (rmarkdown::pandoc_available("1.12.3"))   {
  report(iris, output_dir = tempdir())
}
```

---

rescale01 *Rescales a numeric variable into values between 0 and 1*

Description

Rescales a numeric variable into values between 0 and 1

Usage

```
rescale01(x)
```

Arguments

x   numeric vector (to be rescaled)

Value

vector with values between 0 and 1

Examples

```
rescale01(0:10)
```

---

simplify_text *Simplifies a text string*

Description

A text string is converted into a simplified version by trimming, converting to upper case, replacing german Umlaute, dropping special characters like comma and semicolon and replacing multiple spaces with one space.

Usage

```
simplify_text(text)
```

Arguments

text   text string

Value

text string

Examples

```
simplify_text(" Hello  World !, ")
```

---

target_explore_cat   *Explore categorical variable + target*

Description

Create a plot to explore relation between categorical variable and a binary target

Usage

```
target_explore_cat(
  data,
  var,
  target = "target_ind",
  min_val = NA,
  max_val = NA,
  flip = TRUE,
  num2char = TRUE,
  title = NA,
  auto_scale = TRUE,
  na = NA,
  max_cat = 30,
  legend_position = "bottom"
)
```

Arguments

data	A dataset
var	Categorical variable
target	Target variable (0/1 or FALSE/TRUE)
min_val	All values < min_val are converted to min_val
max_val	All values > max_val are converted to max_val
flip	Should plot be flipped? (change of x and y)
num2char	If TRUE, numeric values in variable are converted into character
title	Title of plot
auto_scale	Not used, just for compatibility
na	Value to replace NA
max_cat	Maximum numbers of categories to be plotted
legend_position	
	Position of legend ("right"|"bottom"|"non")

Value

Plot object

---

target_explore_num   *Explore categorical variable + target*

Description

Create a plot to explore relation between numerical variable and a binary target

Usage

```
target_explore_num(
  data,
  var,
  target = "target_ind",
  min_val = NA,
  max_val = NA,
  flip = TRUE,
  title = NA,
  auto_scale = TRUE,
  na = NA,
  legend_position = "bottom"
)
```

Arguments

data	A dataset
var	Numerical variable
target	Target variable (0/1 or FALSE/TRUE)
min_val	All values < min_val are converted to min_val
max_val	All values > max_val are converted to max_val
flip	Should plot be flipped? (change of x and y)
title	Title of plot
auto_scale	Use 0.02 and 0.98 quantile for min_val and max_val (if min_val and max_val are not defined)
na	Value to replace NA
legend_position	
	Position of legend ("right"|"bottom"|"non")

Value

Plot object

---

total_fig_height   *Get fig.height for RMarkdown-junk using explore_all()*

Description

Get fig.height for RMarkdown-junk using explore_all()

Usage

```
total_fig_height(
  data,
  var_name_n,
  var_name_target,
  nvar = NA,
  ncol = 2,
  size = 3
)
```

Arguments

data	A dataset
var_name_n	Weights variable for count data? (TRUE / MISSING)
var_name_target	
	Target variable (TRUE / MISSING)
nvar	Number of variables to plot
ncol	Number of columns (default = 2)
size	fig.height of 1 plot (default = 3)

Value

Number of rows

Examples

```
total_fig_height(iris)
total_fig_height(iris, var_name_target = "Species")
total_fig_height(nvar = 5)
```

---

weight_target *Weight target variable*

Description

Create weights for the target variable in your dataset so that are equal weiths for target = 0 and target = 1. Target must be 0/1, FALSE/TRUE ore no/yes

Usage

```
weight_target(data, target)
```

Arguments

data   A dataset

target   Target variable (0/1, TRUE/FALSE, yes/no)

Value

Weights for each observation (as a vector)

Examples

```
iris$is_versicolor <- ifelse(iris$Species == "versicolor", 1, 0)
weights <- weight_target(iris, target = is_versicolor)
summary(weights)
```

Index