Package 'explore'

August 27, 2019

Type Package

Title Simplifies Exploratory Data Analysis

Version 0.4.4

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 two main verbs. describe() to describe a variable or table, explore() to graphically explore a variable or table.

License GPL-3

Encoding UTF-8

LazyData true

URL http://github.com/rolkra/explore

Imports broom, dplyr, DBI, DT, forcats, ggplot2 (>= 3.0.0), gridExtra, magrittr, MASS, odbc, rlang, rpart, rpart.plot, shiny, rmarkdown

RoxygenNote 6.1.1

Suggests knitr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-27 12:00:12 UTC

R topics documented:

`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, 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)
name	New name of variable (as string)

Value

Dataset

Examples

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

`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, target, out = "text", ...)
```

Arguments

data	A dataset
var	A variable of the dataset
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

Examples

```
describe_all(iris)
```

describe_cat *Describe categorial variable*

Description

Describe categorial variable

Usage

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

Arguments

data	A dataset
var	Variable or variable name
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, out = "text", margin = 0)
```

Arguments

data	A dataset
var	Variable or variable name
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, target, out = "text")
```

Arguments

`data`	A dataset
`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, ...)
```

Arguments

data	dataframe
dsn	DSN string
table	table name (character string)
...	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 logistic regression*

Description

Explain a binary target using logistic regression

Usage

```
explain_logreg(data, target, ...)
```

Arguments

data	A dataset
target	Target variable (binary)
...	Further arguments

Value

Dataset with results (term, estimate, std.error, z.value, p.value)

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 regression)*

Description

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

Usage

```
explain_tree(data, target, max_cat = 10, maxdepth = 3, minsplit = 20,
  cp = 0, size = 0.7, ...)
```

Arguments

data	A dataset
target	Target variable
max_cat	Drop categorical variables with higher number of levels
maxdepth	Maximal depth of the tree (rpart-parameter)
minsplit	The minimum number of observations that must exist in a node in order for a split to be attempted (rpart-parameter)
cp	Complexity parameter (rpart-parameter)
size	Textsize of plot
...	Further arguments

Value

Plot

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, 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
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

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, target, ncol = 2, split = TRUE)
```

Arguments

data	A dataset
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 categorial variable using bar charts*

Description

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

Usage

```
explore_bar(data, var, target, flip = TRUE, title = "", 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)
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, 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)
`color`	Color of the plot
`...`	Further arguments

Value

Plot

Examples

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

`explore_density` *Explore density of variable*

Description

Create a density plot to explore numerical variable

Usage

```
explore_density(data, var, target, 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)
`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_tbl *Explore table*

Description

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

Usage

```
explore_tbl(data)
```

Arguments

data	A dataset

Examples

```
explore_tbl(iris)
```

format_num_kMB *Format number*

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*

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|lg|chr|dat)

Usage

```
format_type(type)
```

Arguments

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

Value

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

Examples

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

get_nrow *Get number of rows for a grid plot*

Description

Get number of rows for a grid plot

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)
```

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)
```

Arguments

var	A vector (dataframe column)

Value

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

Examples

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

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, target, split = TRUE, output_file, output_dir)
```

Arguments

data	A dataset
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())
}
```

target_explore_cat *Explore categorial variable + target*

Description

Create a plot to explore relation between categorial 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	Categorial 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 categorial 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

Index