Package 'visdat'

July 11, 2017

Title Preliminary Data Visualisation

Version 0.1.0

Description Create preliminary exploratory data visualisations of an entire dataset to identify problems or unexpected features using 'ggplot2'.

Depends R (>= 3.2.2)

License MIT + file LICENSE

LazyData true

RoxygenNote 6.0.1.9000

Imports ggplot2, tidyr, dplyr, purrr, magrittr, stats

URL https://github.com/njtierney/visdat/

BugReports https://github.com/njtierney/visdat/issues

Suggests plotly (>= 4.5.6), testthat, knitr, rmarkdown, vdiffr

VignetteBuilder knitr

NeedsCompilation no

Author Nicholas Tierney [aut, cre]

Maintainer Nicholas Tierney <nicholas.tierney@gmail.com>

Repository CRAN

Date/Publication 2017-07-11 16:51:27 UTC

R topics documented:

`add_vis_dat_pal` *Add a specific palette to a visdat plot*

Description

Add a specific palette to a visdat plot

Usage

```
add_vis_dat_pal(vis_plot, palette)
```

Arguments

`vis_plot`	visdat plot created using vis_gather_, vis_extract_value and vis_create_
`palette`	character "default", "qual" or "cb_safe". "default" (the default) provides the stock ggplot scale for separating the colours. "qual" uses an experimental qualitative colour scheme for providing distinct colours for each Type. "cb_safe" is a set of colours that are appropriate for those with colourblindness. "qual" and "cb_safe" are drawn from http://colorbrewer2.org/.

Value

a visdat plot with a particular palette

Examples

```
## Not run:
# see internal use inside vis_guess and vis_dat

## End(Not run)
```

fingerprint *A utility function for* vis_dat

Description

fingerprint is an internal function that takes the "fingerprint" of a dataframe, and (currently) replaces the contents (x) with the class of a given object, unless it is missing (coded as NA), in which case it leaves it as NA. The name "fingerprint" is taken from the csv-fingerprint, of which this package is based.

Usage

```
fingerprint(x)
```

Arguments

x	a vector

label_col_missing_pct *(Internal) Create labels for the columns containing the % missing data*

Description

(Internal) Create labels for the columns containing the % missing data

Usage

```
label_col_missing_pct(x, col_order_index)
```

Arguments

x	data.frame
col_order_index	the order of the columns

Value

data.frame containing the missingness percent down to 0.1 percent

miss_guide_label *Label the legend with the percent of missing data*

Description

`miss_guide_label` is an internal function for vis_miss to label the legend.

Usage

```
miss_guide_label(x)
```

Arguments

`x`	is a dataframe passed from vis_miss(x).

Value

a data_frame with two columns `p_miss_lab` and `p_pres_lab`, containing the labels to use for present and missing. A dataframe is returned because I think it is a good style habit compared to a list.

typical_data *A small toy dataset of imaginary people*

Description

A dataset containing information about some randomly generated people, created using the excellent `wakefield` package. It is created as deliberately messy dataset.

Usage

```
typical_data
```

Format

A data frame with 5000 rows and 11 variables:

ID Unique identifier for each individual, a sequential character vector of zero-padded identification numbers (IDs). see ?wakefield::id

Race Race for each individual, "Black", "White", "Hispanic", "Asian", "Other", "Bi-Racial", "Native", and "Hawaiin", see ?wakefield::race

Age Age of each individual, see ?wakefield::age

Sex Male or female, see ?wakefield::sex

Height(cm) Height in centimeters, see ?wakefield::height

IQ vector of intelligence quotients (IQ), see ?wakefield::iq

Smokes whether or not this person smokes, see ?wakefield::smokes

Income Yearly income in dollars, see ?wakefield::income

Died Whether or not this person has died yet., see ?wakefield::died

---

`typical_data_large` *A small toy dataset of imaginary people*

Description

A wider dataset than `typical_data` containing information about some randomly generated people, created using the excellent `wakefield` package. It is created as deliberately odd / eclectic dataset.

Usage

```
typical_data_large
```

Format

A data frame with 300 rows and 49 variables:

Age Age of each individual, see ?wakefield::age for more info

Animal A vector of animals, see ?wakefield::animal

Answer A vector of "Yes" or "No"

Area A vector of living areas "Suburban", "Urban", "Rural"

Car names of cars - see ?mtcars

Children vector of number of children - see ?wakefield::children

Coin character vector of "heads" and "tails"

Color vector of vectors from "colors()"

Date vector of "important" dates for an individual

Death TRUE / FALSE for whether this person died

Dice 6 sided dice result

DNA vector of GATC nucleobases

DOB birth dates

Dummy a 0/1 dummy var

Education education attainment level

Employment employee status

Eye eye colour

Grade percent grades

Grade_Level favorite school grade

Group control or treatment

hair hair colours - "brown", "black", "blonde", or "red"

Height height in cm

Income yearly income

Browser choice of internet browser

IQ intelligence quotient

Language random language of the world

Level levels between 1 and 4

Likert likert response - "strongly agree", "agree", and so on

Lorem_Ipsum lorem ipsum text

Marital marital status- "married", "divorced", "widowed", "separated", etc

Military miliary branch they are in

Month their favorite month

Name their name

Normal a random normal number

Political their favorite political party

Race their race

Religion their religion

SAT their SAT score

Sentence an uttered sentence

Sex_1 sex of their first child

Sex_2 sex of their second child

Smokes do they smoke

Speed their median speed travelled in a car

State the last state they visited in the USA

String a random string they smashed out on the keyboard

Upper the last key they hit in upper case

Valid TRUE FALSE answer to a question

Year significant year to that individuals

Zip a zip code they have visited

visdat *visdat*

Description

visdat is a package that helps with the preliminary visualisation of data. visdat makes it easy to visualise your whole dataset so that you can visually identify problems.

See Also

It's main functions are:

- vis_dat()
- vis_miss()

Learn more about visdat at www.njtierney.com/visdat/articles/using_visdat.html

vis_create_ *(Internal) Create a boilerplate for visualisations of the vis_ family*

Description

(Internal) Create a boilerplate for visualisations of the vis_ family

Usage

```
vis_create_(x)
```

Arguments

x	a dataframe in longformat as transformed by vis_gather_ and vis_extract_value.

Value

a ggplot object

vis_dat *Visualises a data.frame to tell you what it contains.*

Description

vis_dat gives you an at-a-glance ggplot object of what is inside a dataframe. Cells are coloured according to what class they are and whether the values are missing. As vis_dat returns a ggplot object, it is very easy to customize and change labels, and customize the plot

Usage

```
vis_dat(x, sort_type = TRUE, palette = "default", warn_large_data = TRUE,
  large_data_size = 9e+05)
```

Arguments

x	a data.frame object
sort_type	logical TRUE/FALSE. When TRUE (default), it sorts by the type in the column to make it easier to see what is in the data
palette	character "default", "qual" or "cb_safe". "default" (the default) provides the stock ggplot scale for separating the colours. "qual" uses an experimental qualitative colour scheme for providing distinct colours for each Type. "cb_safe" is a set of colours that are appropriate for those with colourblindness. "qual" and "cb_safe" are drawn from http://colorbrewer2.org/.
warn_large_data	logical default is TRUE
large_data_size	integer default is 900000, this can be changed.

Value

ggplot2 object displaying the type of values in the data frame and the position of any missing values.

See Also

vis_miss()

Examples

```
vis_dat(airquality)

# experimental colourblind safe palette
vis_dat(airquality, palette = "cb_safe")
vis_dat(airquality, palette = "qual")
```

vis_extract_value_ *(Internal) Add values of each row as a column*

Description

This adds information about each row, so that when called by plotly, the values are made visible on hover. Warnings are suppressed because tidyr gives a warning about type coercion, which is fine.

Usage

```
vis_extract_value_(x)
```

Arguments

x	dataframe created from `vis_gather_`

Value

the x dataframe with the added column `value`.

vis_gather_ *(Internal) Gather rows into a format appropriate for grid visualisation*

Description

(Internal) Gather rows into a format appropriate for grid visualisation

Usage

```
vis_gather_(x)
```

Arguments

x	a dataframe

Value

data.frame gathered to have columns "variables", "valueType", and a row id called "rows".

vis_miss *Visualise a data.frame to display missingness.*

Description

vis_miss provides an at-a-glance ggplot of the missingness inside a dataframe, colouring cells according to missingness, where black indicates a missing cell and grey indicates a present cell. As it returns a ggplot object, it is very easy to customize and change labels.

Usage

```
vis_miss(x, cluster = FALSE, sort_miss = FALSE, show_perc = TRUE,
  show_perc_col = TRUE, large_data_size = 9e+05, warn_large_data = TRUE)
```

Arguments

x	a data.frame
cluster	logical. TRUE specifies that you want to use hierarchical clustering (mcquitty method) to arrange rows according to missingness. FALSE specifies that you want to leave it as is.
sort_miss	logical. TRUE arranges the columns in order of missingness
show_perc	logical. TRUE now adds in the % of missing/complete data in the whole dataset into the legend. Default value is TRUE.
show_perc_col	logical. TRUE adds in the % missing data in a given column into the x axis. Can be disabled with FALSE
large_data_size	integer default is 900000, this can be changed.
warn_large_data	logical default is TRUE

Value

ggplot2 object displaying the position of missing values in the dataframe, and the percentage of values missing and present.

See Also

vis_dat()

Examples

```
vis_miss(airquality)

vis_miss(airquality, cluster = TRUE)

vis_miss(airquality, sort_miss = TRUE)
```

Index