Package ‘inspectdf’

November 5, 2019

Title Inspection, Comparison and Visualisation of Data Frames
Version 0.0.7
Maintainer Alastair Rushworth <alastairrushworth@gmail.com>
Description A collection of utilities for columnwise summary, comparison and visualisation of data frames. Functions report missingness, categorical levels, numeric distribution, correlation, column types and memory usage.
Language en_GB
LinkingTo Rcpp
LazyLoad yes
LazyData true
ByteCompile yes
Encoding UTF-8
Depends R (>= 3.1.0)
Imports dplyr, ggplot2, magrittr, progress, Rcpp, tibble, tidyr, ggfittext (>= 0.8.0)
Suggests testthat
License GPL-2
URL https://alastairrushworth.github.io/inspectdf/
BugReports http://github.com/alastairrushworth/inspectdf/issues
RoxygenNote 6.1.1
NeedsCompilation yes
Author Alastair Rushworth [aut, cre],
David Wilkins [ctb]
Repository CRAN
Date/Publication 2019-11-05 20:10:02 UTC
R topics documented:

inspect_cat ...................................................... 2
inspect_coe ......................................................... 4
inspect_imb ........................................................ 5
inspect_mem ........................................................ 7
inspect_na ........................................................... 8
inspect_num .......................................................... 10
inspect_types ......................................................... 12
show_plot ............................................................ 13
techn ................................................................. 14

Index 16

inspect_cat  Summary and comparison of the levels in categorical columns

Description

For a single dataframe, summarise the levels of each categorical column. If two dataframes are supplied, compare the levels of categorical features that appear in both dataframes. For grouped dataframes, summarise the levels of categorical features separately for each group.

Usage

inspect_cat(df1, df2 = NULL, show_plot = FALSE)

Arguments

df1 A dataframe.
df2 An optional second data frame for comparing categorical levels. Defaults to NULL.
show_plot (Deprecated) Logical flag indicating whether a plot should be shown. Superseded by the function show_plot() and will be dropped in a future version.

Details

For a single dataframe, the tibble returned contains the columns:

- col_name, character vector containing column names of df1.
- cnt integer column containing count of unique levels found in each column, including NA.
- common, a character column containing the name of the most common level.
- common_pcnt, the percentage of each column occupied by the most common level shown in common.
- levels, a named list containing relative frequency tibbles for each feature.
For a **pair of dataframes**, the tibble returned contains the columns:

- **col_name**, character vector containing names of columns appearing in both df1 and df2.
- **jsd**, a numeric column containing the Jensen-Shannon divergence. This measures the difference in relative frequencies of levels in a pair of categorical features. Values near to 0 indicate agreement of the distributions, while 1 indicates disagreement.
- **fisher_p**, the p-value corresponding to Fisher’s exact test. A small p indicates evidence that the two sets of relative frequencies are actually different.
- **lvls_1, lvls_2**, the relative frequency of levels in each of df1 and df2.

For a **grouped dataframe**, the tibble returned is as for a single dataframe, but where the first k columns are the grouping columns. There will be as many rows in the result as there are unique combinations of the grouping variables.

**Value**

A tibble summarising or comparing the categorical features in one or a pair of dataframes.

**Author(s)**

Alastair Rushworth

**See Also**

`inspect_imb, show_plot`

**Examples**

```r
# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_cat(starwars)

# Paired dataframe comparison
inspect_cat(starwars, starwars[1:20, ])

# Grouped dataframe summary
starwars %>% group_by(gender) %>% inspect_cat()
```
inspect_cor

**Summarise and compare Pearson’s correlation coefficients for numeric columns in one or two dataframes.**

**Description**

Summarise and compare Pearson’s correlation coefficients for numeric columns in one or two dataframes.

**Usage**

```r
inspect_cor(df1, df2 = NULL, method = "pearson", with_col = NULL, alpha = 0.05, show_plot = FALSE)
```

**Arguments**

- `df1`: A data frame.
- `df2`: An optional second data frame for comparing correlation coefficients. Defaults to NULL.
- `method`: a character string indicating which type of correlation coefficient to use, one of "pearson", "kendall", or "spearman", which can be abbreviated.
- `with_col`: Character vector of column names to calculate correlations with all other numeric features. The default `with_col = NULL` returns all pairs of correlations.
- `alpha`: Alpha level for correlation confidence intervals. Defaults to 0.05.
- `show_plot`: (Deprecated) Logical flag indicating whether a plot should be shown. Superseded by the function `show_plot()` and will be dropped in a future version.

**Details**

When `df2 = NULL`, a tibble containing correlation coefficients for `df1` is returned:

- `col_1`, `col_2` character vectors containing names of numeric columns in `df1`.
- `corr` the calculated correlation coefficient.
- `p_value` p-value associated with a test where the null hypothesis is that the numeric pair have 0 correlation.
- `lower`, `upper` lower and upper values of the confidence interval for the correlations.
- `pcnt_nna` the number of pairs of observations that were non missing for each pair of columns.

The correlation calculation used by `inspect_cor()` uses only pairwise complete observations.

If `df1` has class `grouped_df`, then correlations will be calculated within the grouping levels and the tibble returned will have an additional column corresponding to the group labels.

When both `df1` and `df2` are specified, the tibble returned contains a comparison of the correlation coefficients across pairs of columns common to both dataframes.
• col_1, col_2 character vectors containing names of numeric columns in either df1 or df2.
• corr_1, corr_2 numeric columns containing correlation coefficients from df1 and df2, respectively.
• p_value p-value associated with the null hypothesis that the two correlation coefficients are the same. Small values indicate that the true correlation coefficients differ between the two dataframes.

Note that confidence intervals for kendall and spearman assume a normal sampling distribution for the Fisher z-transform of the correlation.

Value
A tibble summarising and comparing the correlations for each numeric column in one or a pair of data frames.

Examples

# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_cor(starwars)
# Only show correlations with 'mass' column
inspect_cor(starwars, with_col = "mass")

# Paired dataframe summary
inspect_cor(starwars, starwars[1:10, ])

# NOT RUN - change in correlation over time
# library(dplyr)
# tech_grp <- tech %>%
#   group_by(year) %>%
#   inspect_cor()
# tech_grp %>% show_plot()
Usage

inspect_imb(df1, df2 = NULL, show_plot = FALSE, include_na = FALSE)

Arguments

df1
A dataframe.
df2
An optional second data frame for comparing columnwise imbalance. Defaults to NULL.
show_plot
(Deprecated) Logical flag indicating whether a plot should be shown. Superseded by the function show_plot() and will be dropped in a future version.
include_na
Logical flag, whether to include missing values as a unique level. Default is FALSE - to ignore NA values.

Details

#' For a **single dataframe**, the tibble returned contains the columns:

- col_name, a character vector containing column names of df1.
- value, a character vector containing the most common categorical level in each column of df1.
- pcnt, the relative frequency of each column’s most common categorical level expressed as a percentage.
- cnt, the number of occurrences of the most common categorical level in each column of df1.

For a **pair of dataframes**, the tibble returned contains the columns:

- col_name, a character vector containing names of the unique columns in df1 and df2.
- value, a character vector containing the most common categorical level in each column of df1.
- pcnt, the percentage of each column’s entries occupied by the level in value column.
- cnt, the number of occurrences of the most common categorical level in each column of df1 and df2.

For a **grouped dataframe**, the tibble returned is as for a single dataframe, but where the first k columns are the grouping columns. There will be as many rows in the result as there are unique combinations of the grouping variables.

Value

A tibble summarising and comparing the imbalance for each non-numeric column in one or a pair of data frames.

Author(s)

Alastair Rushworth
inspect_mem

See Also
inspect_cat, show_plot

Examples
# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_imb(starwars)

# Paired dataframe comparison
inspect_imb(starwars, starwars[1:20, ])

# Grouped dataframe summary
starwars %>% group_by(gender) %>% inspect_imb()

inspect_mem(A data frame.)

Arguments
df1
A data frame.
df2
An optional second data frame with which to comparing memory usage. De-
defaults to NULL.
show_plot
(Deprecated) Logical flag indicating whether a plot should be shown. Super-
se ded by the function show_plot() and will be dropped in a future version.

Details
For a single dataframe, the tibble returned contains the columns:

- col_name, a character vector containing column names of df1.
- bytes, integer vector containing the number of bytes in each column of df1.
- size, a character vector containing display-friendly memory usage of each column.
- pcnt, the percentage of the dataframe’s total memory footprint used by each column.
For a pair of dataframes, the tibble returned contains the columns:

- `col_name`, a character vector containing column names of df1 and df2.
- `size_1, size_2`, a character vector containing memory usage of each column in each of df1 and df2.
- `pcnt_1, pcnt_2`, the percentage of total memory usage of each column within each of df1 and df2.

For a grouped dataframe, the tibble returned is as for a single dataframe, but where the first k columns are the grouping columns. There will be as many rows in the result as there are unique combinations of the grouping variables.

**Value**

A tibble summarising and comparing the columnwise memory usage for one or a pair of data frames.

**Author(s)**

Alastair Rushworth

**See Also**

`show_plot`

**Examples**

```r
# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_mem(starwars)

# Paired dataframe comparison
inspect_mem(starwars, starwars[1:20, ])

# Grouped dataframe summary
starwars %>% group_by(gender) %>% inspect_mem()
```

---

**inspect_na**

*Summary and comparison of the rate of missingness across dataframe columns*

**Description**

For a single dataframe, summarise the rate of missingness in each column. If two dataframes are supplied, compare missingness for columns appearing in both dataframes. For grouped dataframes, summarise the rate of missingness separately for each group.
Usage

inspect_na(df1, df2 = NULL, show_plot = FALSE)

Arguments

df1  A data frame

df2  An optional second data frame for making columnwise comparison of missingness. Defaults to NULL.

show_plot  (Deprecated) Logical flag indicating whether a plot should be shown. Superseded by the function show_plot() and will be dropped in a future version.

Details

For a single dataframe, the tibble returned contains the columns:

- col_name, a character vector containing column names of df1.
- cnt, an integer vector containing the number of missing values by column.
- pcnt, the percentage of records in each columns that is missing.

For a pair of dataframes, the tibble returned contains the columns:

- col_name, the name of the columns occurring in either df1 or df2.
- cnt_1, cnt_2, a pair of integer vectors containing counts of missing entries for each column in df1 and df2.
- pcnt_1, pcnt_2, a pair of columns containing percentage of missing entries for each column in df1 and df2.
- p_value, the p-value associated with test of equivalence of rates of missingness. Small values indicate evidence that the rate of missingness differs for a column occurring in both df1 and df2.

For a grouped dataframe, the tibble returned is as for a single dataframe, but where the first k columns are the grouping columns. There will be as many rows in the result as there are unique combinations of the grouping variables.

Value

A tibble summarising the count and percentage of columnwise missingness for one or a pair of data frames.

Author(s)

Alastair Rushworth

See Also

show_plot
Examples

```r
# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_na(starwars)

# Paired dataframe comparison
inspect_na(starwars, starwars[1:20, ])

# Grouped dataframe summary
starwars %>% group_by(gender) %>% inspect_na()
```

---

**inspect_num**

*Summary and comparison of numeric columns*

**Description**

For a single dataframe, summarise the numeric columns. If two dataframes are supplied, compare numeric columns appearing in both dataframes. For grouped dataframes, summarise numeric columns separately for each group.

**Usage**

```r
inspect_num(df1, df2 = NULL, breaks = 20, include_int = TRUE, show_plot = FALSE)
```

**Arguments**

- `df1`: A dataframe.
- `df2`: An optional second dataframe for comparing categorical levels. Defaults to NULL.
- `breaks`: Integer number of breaks used for histogram bins, passed to `graphics::hist()`.
  Defaults to 20.
- `include_int`: Logical flag, whether to include integer columns in numeric summaries. Defaults to TRUE.
- `show_plot`: (Deprecated) Logical flag indicating whether a plot should be shown. Superseded by the function `show_plot()` and will be dropped in a future version.

**Details**

For a **single dataframe**, the tibble returned contains the columns:

- `col_name`, a character vector containing the column names in `df1`
inspect_num

- min, q1, median, mean, q3, max and sd, the minimum, lower quartile, median, mean, upper quartile, maximum and standard deviation for each numeric column.
- pcnt_na, the percentage of each numeric feature that is missing
- hist, a named list of tibbles containing the relative frequency of values in a falling in bins determined by breaks.

For a pair of dataframes, the tibble returned contains the columns:

- col_name, a character vector containing the column names in df1 and df2
- hist_1, hist_2, a list column for histograms of each of df1 and df2. Where a column appears in both dataframes, the bins used for df1 are reused to calculate histograms for df2.
- jsd, a numeric column containing the Jensen-Shannon divergence. This measures the difference in distribution of a pair of binned numeric features. Values near to 0 indicate agreement of the distributions, while 1 indicates disagreement.
- fisher_p, the p-value corresponding to Fisher’s exact test. A small p indicates evidence that the two histograms are actually different.

For a grouped dataframe, the tibble returned is as for a single dataframe, but where the first k columns are the grouping columns. There will be as many rows in the result as there are unique combinations of the grouping variables.

Value

A tibble containing statistical summaries of the numeric columns of df1, or comparing the histograms of df1 and df2.

Author(s)

Alastair Rushworth

See Also

show_plot

Examples

# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_num(starwars)

# Paired dataframe comparison
inspect_num(starwars, starwars[1:20, ])

# Grouped dataframe summary
starwars %>% group_by(gender) %>% inspect_num()
inspect_types  

**Summary and comparison of column types**

**Description**
For a single dataframe, summarise the column types. If two dataframes are supplied, compare column type composition of both dataframes.

**Usage**

```r
going_to_show_plot <- function() {
  message("inspect_types() is deprecated, please use inspect_types() instead")
}
{
  <- NULL
  <- TRUE
}
inspect_types <- function(df1, df2 = going_to_show_plot(), show_plot = FALSE) {
  if (!is.null(df2)) {
    return(inspect_type_table(df1, show_plot = show_plot, df2 = df2))
  } else {
    return(inspect_type_table(df1, show_plot = show_plot))
  }
}
```

**Arguments**

- `df1`: A dataframe.
- `df2`: An optional second dataframe for comparison.
- `show_plot`: (Deprecated) Logical flag indicating whether a plot should be shown. Superseded by the function `show_plot()` and will be dropped in a future version.

**Details**

For a **single dataframe**, the tibble returned contains the columns:

- `type`, a character vector containing the column types in `df1`.
- `cnt`, integer counts of each type.
- `pcnt`, the percentage of all columns with each type.
- `col_name`, the names of columns with each type.

For a **pair of dataframes**, the tibble returned contains the columns:

- `type`, a character vector containing the column types in `df1` and `df2`.
- `cnt_1`, `cnt_2`, pair of integer columns containing counts of each type - in each of `df1` and `df2`.
- `pcnt_1`, `pcnt_2`, pair of columns containing the percentage of columns with each type - the data frame name are appended.

For a **grouped dataframe**, the tibble returned is as for a single dataframe, but where the first k columns are the grouping columns. There will be as many rows in the result as there are unique combinations of the grouping variables.

**Value**

A tibble summarising the count and percentage of different column types for one or a pair of data frames.
show_plot

Author(s)
Alastair Rushworth

See Also
show_plot

Examples

# Load dplyr for starwars data & pipe
library(dplyr)

# Single dataframe summary
inspect_types(starwars)

# Paired dataframe comparison
inspect_types(starwars, starwars[1:20, ])

show_plot(x, text_labels = TRUE, alpha = 0.05, high_cardinality = 0,
plot_layout = NULL, col_palette = 0, plot_type = "bar",
label_thresh = 0.1)

Description

Visualise summaries and comparisons of one or two dataframes.

Usage

show_plot(x, text_labels = TRUE, alpha = 0.05, high_cardinality = 0,
plot_layout = NULL, col_palette = 0, plot_type = "bar",
label_thresh = 0.1)

Arguments

x          Dataframe resulting from a call to an 'inspect_' function.
text_labels Whether to show text annotation on plots (when show_plot = T).
alpha       Alpha level for performing significance tests. Defaults to 0.05.
high_cardinality Minimum number of occurrences of category to be shown as a distinct segment in the plot (inspect_cat only). Default is 0. This can help when some columns contain many unique or near-unique levels that take a long time to render.
plot_layout Vector specifying the number of rows and columns in the plotting grid. For example, 3 rows and 2 columns would be specified as plot_layout = c(3,2). Default is TRUE.
col_palette Integer indicating the colour palette to use.
- '0': (default) ‘ggplot2’ color palette
- '1': [colorblind friendly palette](http://www.cookbook-r.com/Graphs/Colors_(ggplot2/)
- '2': [80s theme](https://www.color-hex.com/color-palette/25888)
- '3': [rainbox theme](https://www.color-hex.com/color-palette/79261)
- '4': [mario theme](https://www.color-hex.com/color-palette/78663)
- '5': [pokemon theme](https://www.color-hex.com/color-palette/78664)

plot_type String determining the type of plot to show. Defaults to “bar”.

label_thresh Minimum percentage frequency of category for a text label to be shown. Defaults to 0.1. Smaller values will show potentially smaller labels, but at the expense of longer rendering time.

Examples

```r
# Load 'starwars' data
data("starwars", package = "dplyr")

# categorical plot
x <- inspect_cat(starwars)
show_plot(x)

# correlations in numeric columns
x <- inspect_cor(starwars)
show_plot(x)

# feature imbalance bar plot
x <- inspect_imb(starwars)
show_plot(x)

# memory usage barplot
x <- inspect_mem(starwars)
show_plot(x)

# missingness barplot
x <- inspect_na(starwars)
show_plot(x)

# histograms for numeric columns
x <- inspect_num(starwars)
show_plot(x)

# barplot of column types
x <- inspect_types(starwars)
show_plot(x)
```

---

**tech**

Tech stocks closing prices
**Description**

Daily closing stock prices of the three tech companies Microsoft, Apple and IBM between 2007 and 2019.

**Usage**

`data(tech)`

**Format**

A dataframe with 3158 rows and 6 columns.

**Source**

Data gathered using the `quantmod` package.

**Examples**

```r
data(tech)
head(tech)
# NOT RUN - change in correlation over time
# library(dplyr)
# tech_grp <- tech %>%
#   group_by(year) %>%
#   inspect_cor()
# tech_grp %>% show_plot()
```
Index

*Topic datasets
  tech, 14

inspect_cat, 2, 7
inspect_cor, 4
inspect_imb, 3, 5
inspect_mem, 7
inspect_na, 8
inspect_num, 10
inspect_types, 12

show_plot, 3, 7–9, 11, 13, 13

technologies. 14