Package 'inspectdf'

August 26, 2019

Title Inspection, Comparison and Visualisation of Data Frames

Version 0.0.5

Maintainer Alastair Rushworth <alastairmrushworth@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 2019-08-26 16:50:02 UTC

R topics documented:

`inspect_cat` *Summarise and compare the levels for each categorical feature in one or two dataframes.*

Description

Summarise and compare the levels for each categorical feature in one or two dataframes.

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 Superseded by the function `show_plot()` and will be dropped in a future version.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

When `df2 = NULL`, a tibble containing summaries of the categorical features in `df1` is returned:

- `col_name` character vector containing column names of `df1`.
- `cnt` integer column containing count of unique levels found in each column, including `NA`.
- `common` character column containing the name of the most common level.
- `common_pcnt` percentage of each column occupied by the most common level shown in `common`.
- `levels` names list containing relative frequency tibbles for each feature.

When `df1` and `df2` are specified, a comparison of the relative frequencies of levels in common columns is performed. In particular, Jensen-Shannon divergence and Fisher's exact test are returned as part of the comparison.

- `col_name` character vector containing names of columns appearing in both `df1` and `df2`.
- `jsd` 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` p-value corresponding to Fisher's exact test. A small p indicates evidence that the the two sets of relative frequencies are actually different.
- `lvls_1`, `lvls_2` relative frequency of levels in each of `df1` and `df2`.

Value

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

Examples

```
data("starwars", package = "dplyr")
inspect_cat(starwars)
# compare the levels in two data frames
inspect_cat(starwars, starwars[1:20, ])
```

`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

```
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, co1_2 character vectors containing names of numeric columns in df1.
- corr the calculated correlation coefficient.
- lower, upper lower and upper values of the confidence interval for the correlations.
- p_value p-value associated with a test where the null hypothesis is that the numeric pair have 0 correlation.

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, co1_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

```
data("starwars", package = "dplyr")
# correlations in numeric columns
inspect_cor(starwars)
# only show correlations with 'mass' column
inspect_cor(starwars, with_col = "mass")
# compare correlations with a different data frame
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()
```

`inspect_imb` *Summarise and compare columnwise imbalance for non-numeric columns in one or two dataframes.*

Description

Summarise and compare columnwise imbalance for non-numeric columns in one or two dataframes.

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

When `df2 = NULL`, a tibble containing a summary of columnwise imbalance is returned, with columns:

- `col_name` character vector containing column names of `df1`.
- `value` 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`.

When both `df1` and `df2` are specified, the most common levels in features common to both `df1` and `df2` is returned. A simple test of the null hypothesis that the relative frequencies of a common level is the same in both dataframes is performed. The resulting tibble has columns

- `col_name` character vector containing names of the unique columns in `df1` and `df2`.
- `value` 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`.

Value

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

Examples

```
data("starwars", package = "dplyr")
# get tibble of most common levels
inspect_imb(starwars)
# compare imbalance
inspect_imb(starwars, starwars[1:10, -3])
```

`inspect_mem` *Summarise and compare the memory usage in one or two dataframes.*

Description

Summarise and compare the memory usage in one or two dataframes.

Usage

```
inspect_mem(df1, df2 = NULL, show_plot = FALSE)
```

Arguments

`df1`	A data frame.
`df2`	An optional second data frame with which to comparing memory usage. 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

When `df1` is specified and `df2 = NULL`, a tibble summarising columnwise memory usage in descending order of size is returned:

- `col_name` character vector containing column names of `df1`.
- `size` character vector containing display-friendly memory usage of each column.
- `pcnt` the percentage of the dataframe's total memory footprint used by each column.

When both `df1` and `df2` are specified, column memory usages are jointly tabulated for both data frames. Rows are sorted in descending order of size as they appear in `df1`:

- `col_name` character vector containing column names of `df1` and `df2`.
- `size_1`, `size_2` 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`.

Value

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

Examples

```
data("starwars", package = "dplyr")
# get tibble of column memory usage for the starwars data
inspect_mem(starwars)
# compare memory usage
inspect_mem(starwars, starwars[1:10, -3])
```

`inspect_na` *Summarise and compare the rate of missingness in one or two dataframes.*

Description

Summarise and compare the rate of missingness in one or two dataframes.

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

When `df1` is specified and `df2 = NULL`, a tibble containing columnwise summaries of missing values is returned, with columns:

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

When both `df1` and `df2` are specified, missingness is compared across all columns in both dataframes. A test of the null hypothesis that the rate of missingness is the same across the same column in either dataframe.

- `col_name` the name of the columns occurring in either `df1` or `df2`.
- `cnt_1`, `cnt_2` pair of integer vectors containing counts of missing entries for each column in `df1` and `df2`.
- `pcnt_1`, `pcnt_2` pair of columns containing percentage of missing entries for each column in `df1` and `df2`.
- `p_value` 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`.

Value

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

Examples

```
data("starwars", package = "dplyr")
# inspect missingness in starwars data
inspect_na(starwars)
# compare two dataframes
inspect_na(starwars, starwars[1:30, ])
```

`inspect_num` *Summarise and compare the numeric variables within one or two dataframes*

Description

Summarise and compare the numeric variables within one or two dataframes

Usage

```
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. `hist(...,breaks)`. See `?hist` for more details.

Details

If only `df1` is specified, `inspect_num()` returns a tibble with columns

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

If both df1 and df2 are specified, the tibble has columns

- col_name character vector containing the column names in df1 and df2
- hist_1, hist_2 list column for histograms of each of df1 and df2. Where a column appears in both dataframe, the bins used for df1 are reused to calculate histograms for df2.
- jsd 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 p-value corresponding to Fisher's exact test. A small p indicates evidence that the two histograms are actually different.

Value

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

Examples

```
data("starwars", package = "dplyr")
# show summary statistics for starwars
inspect_num(starwars)
# with a visualisation too - try to limit number of bins
inspect_num(starwars, breaks = 10)
# compare two data frames
inspect_num(starwars, starwars[-c(1:10), ], breaks = 10)
```

---

inspect_types *Summarise and compare column types in one or two dataframes.*

Description

Summarise and compare column types in one or two dataframes.

Usage

```
inspect_types(df1, df2 = NULL, show_plot = FALSE)
```

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

When df2 = NULL, a tibble is returned with the columns

- type 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.

When a second data frame df2 is specified, column type summaries are tabulated for both data frames to enable comparison of contents. The resulting tibble has the columns

- type 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.

Value

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

Examples

```
data("starwars", package = "dplyr")
# get tibble of column types for the starwars data
inspect_types(starwars)
# compare two data frames
inspect_types(starwars, starwars[, -1])
```

---

show_plot *Visualise summaries and comparisons of one or two dataframes.*

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‘: a [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

```
# 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

```
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