Package 'tabbycat'

August 22, 2023

Type Package

Title Tabulate and Summarise Categorical Data

Version 0.18.0

Maintainer Oliver Hawkins <oli@olihawkins.com>

Description Functions for tabulating and summarising categorical variables. Most functions are designed to work with dataframes, and use the 'tidyverse' idiom of taking the dataframe as the first argument so they work within pipelines. Equivalent functions that operate directly on vectors are also provided where it makes sense. This package aims to make exploratory data analysis involving categorical variables quicker, simpler and more robust.

License MIT + file LICENSE

Depends R (>= 3.4.0)

Imports dplyr (>= 1.0.0), janitor, magrittr, purrr, rlang, stringr, tibble, tidyr

Encoding UTF-8

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Oliver Hawkins [aut, cre]

Repository CRAN

Date/Publication 2023-08-22 19:30:02 UTC

R topics documented:

cat_compare *Calculate the frequency of discrete values in one categorical variable for each group within another categorical variable*

Description

This function crosstabulates the frequencies of one categorical variable within the groups of another. The results are sorted on the values of the variable whose distribution is shown in each column i.e. the variable specified with `row_cat`. If this variable is a character vector it will be sorted alphabetically. If it is a factor it will be sorted in the order of its levels.

Usage

```
cat_compare(
  data,
  row_cat,
  col_cat,
  na.rm.row = FALSE,
  na.rm.col = FALSE,
  na.rm = NULL,
  only = "",
  clean_names = getOption("tabbycat.clean_names"),
  na_label = getOption("tabbycat.na_label")
)
```

Arguments

`data`	A dataframe containing the two variables of interest.
`row_cat`	The column name of a categorical variable whose distribution will be calculated for each group in `col_cat`.
`col_cat`	The column name of a categorical variable which will be split into groups and the distrubtion of `row_cat` calulated for each group.
`na.rm.row`	A boolean indicating whether to exclude NAs from the row results. The default is FALSE.
`na.rm.col`	A boolean indicating whether to exclude NAs from the column results. The default is FALSE.
`na.rm`	A boolean indicating whether to exclude NAs from both row and column results. This argument is provided as a convenience. It allows you to set `na.rm.row` and `na.rm.col` to the same value without having to specify them separately. If the value of `na.rm` is NULL, the argument is ignored. If it is not NULL it takes precendence. default is NULL.
`only`	A string indicating that only one set of frequency columns should be returned in the results. If `only` is either "n" or "number", only the number columns are returned. If `only` is either "p" or "percent", only the percent columns are returned. If `only` is any other value, both sets of columns are shown. The default value is an empty string, which means both sets of columns are shown.

clean_names — A boolean indicating whether the column names of the results tibble should be cleaned, so that any column names produced from data are converted to snake_case. The default is TRUE, but this can be changed with `options(tabbycat.clean_names = FALSE)`.

na_label — A string indicating the label to use for the columns that contain data for missing values. The default value is "na", but use this argument to set a different value if the default value collides with data in your dataset.

Value

A tibble showing the distribution of `row_cat` within each group in `col_cat`.

---

cat_contrast *Calculate the frequency of discrete values in one categorical variable for each of two mutually exclusive groups within another categorical variable*

Description

This function shows the distrbution of values within given a categorical variable for one group within another categorical variable, and compares it with the distribution among all observations not in that group. Its purpose is to let you see quickly whether the distribution within that group differs from the distribution for the rest of the observations. The results are sorted in descending order of frequency for the named group i.e. the group named in `col_group`.

Usage

```
cat_contrast(
  data,
  row_cat,
  col_cat,
  col_group,
  na.rm.row = FALSE,
  na.rm.col = FALSE,
  na.rm = NULL,
  only = "",
  clean_names = getOption("tabbycat.clean_names"),
  na_label = getOption("tabbycat.na_label"),
  other_label = getOption("tabbycat.other_label")
)
```

Arguments

data — A dataframe containing the two variables of interest.

row_cat — The column name of a categorical variable whose distribution should be calculated for each exclusive group in `col_cat`.

col_cat	The column name of a categorical variable that will be split into two exclusive groups, one containing observations with a particular value of that variable, and another containing all other observations.
col_group	The name of the group within col_cat that is used to split the observations into two exclusive groups: those that are in the group and those that are not in the group.
na.rm.row	A boolean indicating whether to exclude NAs from the row results. The default is FALSE.
na.rm.col	A boolean indicating whether to exclude NAs from the column results. The default is FALSE.
na.rm	A boolean indicating whether to exclude NAs from both row and column results. This argument is provided as a convenience. It allows you to set na.rm.row and na.rm.col to the same value without having to specify them separately. If the value of na.rm is NULL, the argument is ignored. If it is not NULL it takes precendence. default is NULL.
only	A string indicating that only one set of frequency columns should be returned in the results. If only is either "n" or "number", only the number columns are returned. If only is either "p" or "percent", only the percent columns are returned. If only is any other value, both sets of columns are shown. The default value is an empty string, which means both sets of columns are shown.
clean_names	A boolean indicating whether the column names of the results tibble should be cleaned, so that any column names produced from data are converted to snake_case. The default is TRUE, but this can be changed with options(tabbycat.clean_names = FALSE).
na_label	A string indicating the label to use for the columns that contain data for missing values. The default value is "na", but use this argument to set a different value if the default value collides with data in your dataset.
other_label	A string indicating the label to use for the columns that contain data for observations not in the named group. The default value is "other", but use this argument to set a different value if the default value collides with data in your dataset.

Value

A tibble showing the distribution of row_cat within each of the two exclusive groups in col_cat.

---

cat_count *Count the frequency of discrete values in the column of a dataframe*

Description

This function differs from cat_vcount in that it operates on columns in dataframes rather than directly on vectors, which means it is more useful in pipelines but handles a narrower range of inputs. The results are sorted in descending order of frequency.

Usage

```
cat_count(
  data,
  cat,
  na.rm = FALSE,
  only = "",
  clean_names = getOption("tabbycat.clean_names")
)
```

Arguments

data	A dataframe containing a categorical vector for which frequencies will be calculated.
cat	The column name of the categorical variable for which frequencies will be calculated.
na.rm	A boolean indicating whether to exclude NAs from the results. The default is FALSE.
only	A string indicating that only one of the frequency columns should be returned in the results. If only is either "n" or "number", only the number column is returned. If only is either "p" or "percent", only the percent column is returned. If only is any other value, both columns are shown. The default value is an empty string, which means both columns are shown.
clean_names	A boolean indicating whether the column names of the results tibble should be cleaned, so that any column names produced from data are converted to snake_case. The default is TRUE, but this can be changed with options(tabbycat.clean_names = FALSE).

Value

A tibble showing the frequency of each value in cat.

---

cat_summarise *Summarise the values of a numerical variable for each group within a categorical variable*

Description

The results are sorted on the values of the categorical variable i.e. the variable specified with cat. If this variable is a character vector it will be sorted alphabetically. If it is a factor it will be sorted in the order of its levels. This function can be called as either cat_summarise or cat_summarize.

Usage

```
cat_summarise(
  data,
  cat,
  num,
  na.rm = FALSE,
  clean_names = getOption("tabbycat.clean_names")
)

cat_summarize(
  data,
  cat,
  num,
  na.rm = FALSE,
  clean_names = getOption("tabbycat.clean_names")
)
```

Arguments

data	A dataframe containing a categorical variable and numerical variable to summarise.
cat	The name of a column in data which is a categorical vector of discrete values for which summaries will be calculated.
num	The name of a column in data which is a numerical vector that will be summarised for each group.
na.rm	A boolean indicating whether to exclude NAs from the row results. Note that NAs are **always** ignored in calculating the summary statistics for num shown in each row, and the number of NAs that exist in num for each group in cat is shown in the na column of the results table. This argument controls whether a row of summary statistics is shown for observations that are NA in cat. The default is FALSE.
clean_names	A boolean indicating whether the column names of the results tibble should be cleaned, so that any column names produced from data are converted to snake_case. The default is TRUE, but this can be changed with options(tabbycat.clean_names = FALSE).

Value

A tibble showing summary statistics for num for each group in cat.

---

cat_vcount *Count the frequency of discrete values in a categorical vector*

Description

This function differs from `cat_count` in that it operates directly on vectors, rather than on columns in dataframes, which means it is less useful in pipelines but can handle a wider range of inputs. The results are sorted in descending order of frequency.

Usage

```
cat_vcount(
  cat,
  na.rm = FALSE,
  only = "",
  clean_names = getOption("tabbycat.clean_names")
)
```

Arguments

`cat`	A categorical vector for which frequencies will be calculated.
`na.rm`	A boolean indicating whether to exclude NAs from the results. The default is FALSE.
`only`	A string indicating that only one of the frequency columns should be returned in the results. If `only` is either "n" or "number", only the number column is returned. If `only` is either "p" or "percent", only the percent column is returned. If `only` is any other value, both columns are shown. The default value is an empty string, which means both columns are shown.
`clean_names`	A boolean indicating whether the column names of the results tibble should be cleaned, so that any column names produced from data are converted to snake_case. The default is TRUE, but this can be changed with `options(tabbycat.clean_names = FALSE)`.

Value

A tibble showing the frequency of each value in `cat`.

Index