Custom Statistics in dfSummary

Dominic Comtois

2021-07-30

This document shows how to customize the content of the *Stats / Values* column in data frame summaries generated using `summarytools::dfSummary()`. This feature was introduced in version 1.0.0 of **summarytools**as a response to a feature request that came up several times, in a form or another.

How it works

Two new options were created: `dfSummary.custom.1` and `dfSummary.custom.2`. The first one has a predefined value – it is the one that makes up the fourth row of the cell (showing IQR and CV). The second one is set to `NA` by default. If both options are defined (non-`NA`), the cell will now show 5 lines rather than 4, provided there are no additional line feed occurring within the cell, be it by design or by an "overflow" of one of the custom lines.

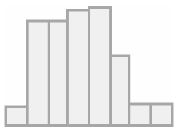
Setup & Baseline

We'll use the first column of *iris* to show results as they are before making any changes. But first, a little bit of setting-up:

```
library(summarytools)
st_options(plain.ascii     = FALSE,
           headings        = FALSE,
           footnote        = NA,
           round.digits    = 1,
           style           = "rmarkdown", # For freq(), descr(), & ctable()
           dfSummary.varnumbers   = FALSE,
           dfSummary.valid.col    = FALSE,
           dfSummary.silent       = TRUE,
           dfSummary.style        = "grid",
           tmp.img.dir            = "img")
```

Now let's show the default output:

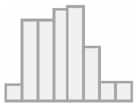
```
iris_subset <- iris[1]
dfSummary(iris_subset, graph.magnif = .45)
```

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Sepal.Length [numeric]	Mean (sd) : 5.8 (0.8) min < med < max: 4.3 < 5.8 < 7.9 IQR (CV) : 1.3 (0.1)	35 distinct values		0 (0.0%)

Example 1 : Removing *IQR (CV)*

Setting `dfSummary.custom.1` to `NA` will remove the last line in *Stats / Values*:

```
st_options(dfSummary.custom.1 = NA)
dfSummary(iris_subset, graph.magnif = .35) # Adjust graph size accordingly
```

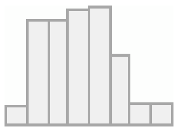
Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Sepal.Length [numeric]	Mean (sd) : 5.8 (0.8) min < med < max: 4.3 < 5.8 < 7.9	35 distinct values		0 (0.0%)

Example 2 : Adding *Q1* & *Q3*

Here we'll create the expression needed to generate new statistics, *Q1* & *Q3*. The expression is evaluated while looping on column data, and we need to refer to that data. The variable name to use is, well, `column_data`. Another variable you can use is `round.digits` (we've set to `1` in the setup chunk on page 1).

```
st_options(
  dfSummary.custom.1 =
    expression(
      paste(
        "Q1 - Q3 :",
        round(
          quantile(column_data,
                   probs = .25,
                   type = 2,
                   names = FALSE,
                   na.rm = TRUE),
          digits = round.digits
        ), " - ",
        round(
          quantile(column_data,
                   probs = .75,
                   type = 2,
                   names = FALSE,
                   na.rm = TRUE),
          digits = round.digits
        )
      )
    )
)

dfSummary(iris_subset, graph.magnif = .45)
```

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Sepal.Length [numeric]	Mean (sd) : 5.8 (0.8) min < med < max: 4.3 < 5.8 < 7.9 Q1 - Q3 : 5.1 - 6.4	35 distinct values		0 (0.0%)

Example 3: Inserting Back *IQR (CV)*

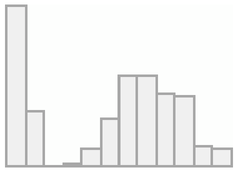
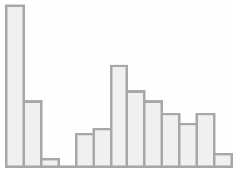
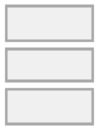
It is always possible to reset the value of `dfSummary.custom.1` to its initial value by using

```
st_options(dfSummary.custom.1 = "default")
```

But let's make things a bit more interesting by actually showing *IQR (CV)* **under** *Q1* & *Q3*. For this, we will use the default expression for `dfSummary.custom.1` to define `dfSummary.custom.2`:

```
st_options(
  dfSummary.custom.2 =
    expression(
      paste(
        paste0(
          trs("iqr"), " (", trs("cv"), ") : "
        ),
        format_number(
          IQR(column_data, na.rm = TRUE),
          round.digits
        ),
        " (",
        format_number(
          sd(column_data, na.rm = TRUE) /
              mean(column_data, na.rm = TRUE),
          round.digits
        ),
        ")",
        collapse = "",
        sep = ""
    )
  )
)

dfSummary(iris[3:5], graph.magnif = .65) # Again, graph size adjusted
```

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Petal.Length [numeric]	Mean (sd) : 3.8 (1.8) min < med < max: 1 < 4.3 < 6.9 Q1 - Q3 : 1.6 - 5.1 IQR (CV) : 3.5 (0.5)	43 distinct values		0 (0.0%)
Petal.Width [numeric]	Mean (sd) : 1.2 (0.8) min < med < max: 0.1 < 1.3 < 2.5 Q1 - Q3 : 0.3 - 1.8 IQR (CV) : 1.5 (0.6)	22 distinct values		0 (0.0%)
Species [factor]	1. setosa 2. versicolor 3. virginica	50 (33.3%) 50 (33.3%) 50 (33.3%)		0 (0.0%)

Don't forget to set `na.rm = TRUE` whenever necessary (most base R statistics use it with `FALSE` as default).

Number Formatting

You may have noticed that instead of `round()`, we used `format_number()`, which is a **summarytools** internal function. It applies not only rounding, but all relevant formatting attributes as well (*nsmall, decimal.mark, big.mark, scientific*,and so on).

Displaying Formatted Expressions

As shown in the Introduction to summarytools vignette, the following bit of code can be used to retrieve and format the expressions stored in the custom options. To achieve good results, the chunk option `results='markup'` was used for this chunk.

```
st_options(dfSummary.custom.1 = "default")
formatR::tidy_source(
  text   = deparse(st_options("dfSummary.custom.1")),
  indent = 2,
  args.newline = TRUE
)
```

```
expression(
  paste(
    paste0(
      trs("iqr"),
      " (", trs("cv"),
      ") : "
    ),
    format_number(
      IQR(column_data, na.rm = TRUE),
      round.digits
    ),
    " (", format_number(
      sd(column_data, na.rm = TRUE)/mean(column_data, na.rm = TRUE),
      round.digits
    ),
    ")", collapse = "", sep = ""
  )
)
```

Useful links

1. Introduction to summarytools (package vignette)
2. Summarytools in R Markdown Documents (package vignette)
3. Data Frame Summaries in PDF's (supplemental documentation)