My First R Markdown Document

Author: Your Name

Last update: 24 May, 2016

Render `Rmd` script

An R Markdown script can be evaluated and rendered with the following `render` command or by pressing the `knit` button in RStudio. The `output_format` argument defines the format of the output (*e.g.* `html_document`). The setting `output_format="all"` will generate all supported output formats. Alternatively, one can specify several output formats in the metadata section as shown in the above example.

```
rmarkdown::render("input.Rmd", clean=TRUE, output_format="html")
```

The following shows how to run the rendering from the command-line.

```
$ echo "rmarkdown::render('input.Rmd', clean=TRUE, output_format='html')" | R --slave
```

R code chunks

R Code Chunks can be embedded in an R Markdown script by using three backticks at the beginning of a new line along with a certain modifier syntax to initialize a code chunk and another three to terminate a code chunk also at the beginning of a new line. The following shows an example of such a code chunk. Note the backslashes are not part of it. They have been added to include the syntax in this document.

```
    ```\{r code_chunk_name, eval=FALSE\}
    x <- 1:10
    ```
```

The code chunk options are provided within the braces of the first line. The following explains the meaning of the most important options:

- `r`: specifies language for code chunk, here R
- `chode_chunk_name`: name of code chunk; this name needs to be unique
- `eval`: if assigned `TRUE` the code will be evaluated
- `warning`: if assigned `FALSE` warnings will not be shown
- `message`: if assigned `FALSE` messages will not be shown
- `cache`: if assigned `TRUE` results will be cached to reuse in future rendering instances
- `fig.height`: allows to specify height of figures in inches
- `fig.width`: allows to specify width of figures in inches

For more details on code chunk options see here.

Learning Markdown

The basic syntax of Markdown and derivatives like kramdown is extremely easy to learn. Rather than providing another introduction on this topic, here are some useful sites for learning Markdown:

- Markdown Intro on GitHub
- Markdown Cheet Sheet
- Markdown Basics from RStudio
- R Markdown Cheat Sheet
- kramdown Syntax

Tables

There are several ways to render tables. First, they can be printed within the R code chunks. Second, nice formatted table can be generated with the functions `kable`, `pander` or `xtable`. The following example uses `kable` from the `knitr` package.

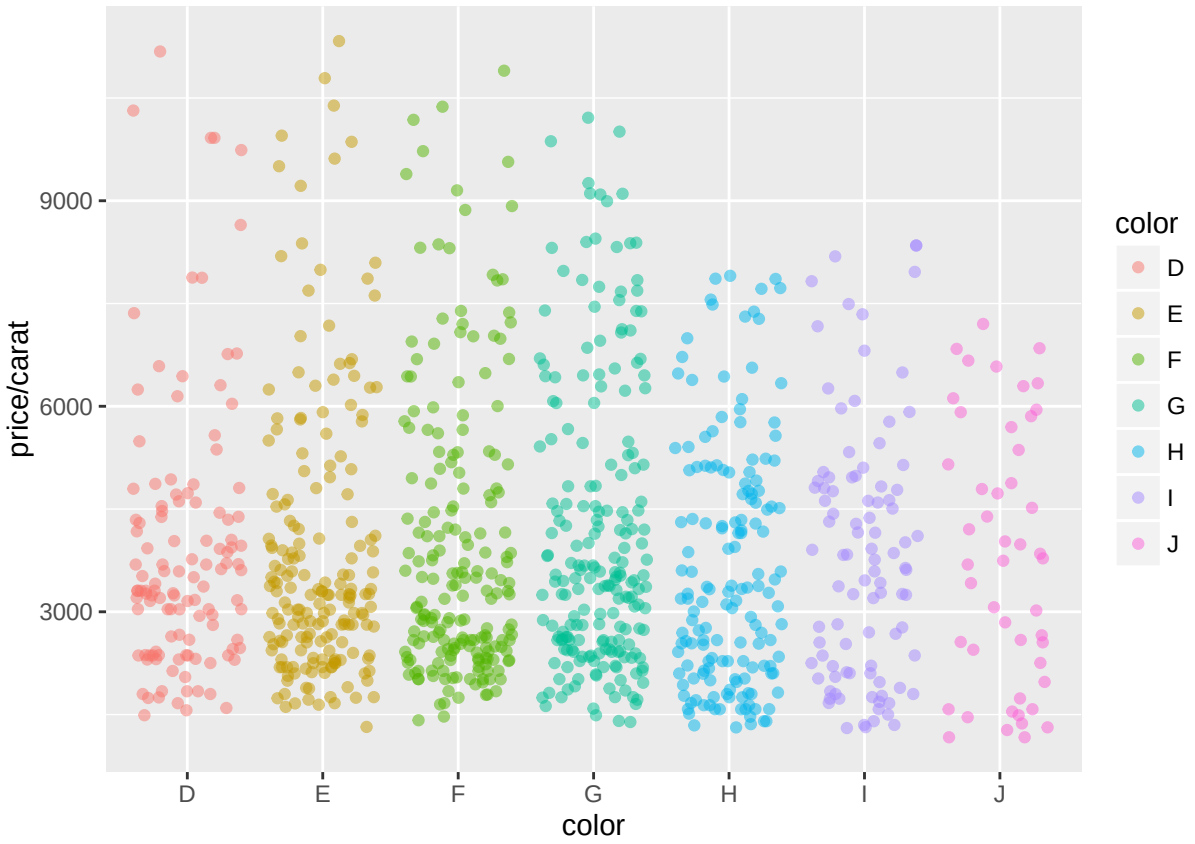
```
library(knitr)
kable(iris[1:12,])
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa

Figures

Plots generated by the R code chunks in an R Markdown document can be automatically inserted in the output file. The size of the figure can be controlled with the `fig.height` and `fig.width` arguments.

```
library(ggplot2)
dsmall <- diamonds[sample(nrow(diamonds), 1000), ]
ggplot(dsmall, aes(color, price/carat)) + geom_jitter(alpha = I(1 / 2), aes(color=color))
```



Sometimes it can be useful to explicitly write an image to a file and then insert that image by referencing its file name. For instance, this can be useful for time consuming analyses. The following code will generate a file named `myplot.png`. To insert the file in the rendered document, one can use standard Markdown or HTML syntax, *e.g.*: `<img src="myplot.png"/>`.

```
png("myplot.png")
ggplot(dsmall, aes(color, price/carat)) + geom_jitter(alpha = I(1 / 2), aes(color=color))
dev.off()
```

```
## pdf
##   2
```

Inline R code

To evaluate R code inline, one can enclose an R expression with a single back-tick followed by `r` and then the actual expression. For instance, the back-ticked version of 'r 1 + 1' evaluates to 2 and 'r pi' evaluates to 3.1415927.

Mathematical expressions

To render mathematical equations, one can use standard Latex syntax. When expressions are enclosed with single `$` signs then they will be shown inline, while enclosing them with double `$$` will show them in display mode. For instance, the following Latex syntax `d(X,Y) = \sqrt[]{ \sum_{i=1}^{n}{(x_{i}-y_{i})^2} }` renders in display mode as follows:

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Citations and bibliographies

Citations and bibliographies can be autogenerated in R Markdown in a similar way as in Latex/Bibtex. Reference collections should be stored in a separate file in Bibtex or other supported formats. To cite a publication in an R Markdown script, one uses the syntax `[@<id1>]` where `<id1>` needs to be replaced with a reference identifier present in the Bibtex database listed in the metadata section of the R Markdown script (*e.g.* `bibtex.bib`). For instance, to cite Lawrence et al. (2013), one uses its reference identifier (*e.g.* `Lawrence2013-kt`) as `<id1>` (Lawrence et al. 2013). This will place the citation inline in the text and add the corresponding reference to a reference list at the end of the output document. For the latter a special section called `References` needs to be specified at the end of the R Markdown script. To fine control the formatting of citations and reference lists, users want to consult this the corresponding R Markdown page. Also, for general reference management and outputting references in Bibtex format Paperpile can be very helpful.

Session Info

```
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  utils     datasets  grDevices methods   base
##
## other attached packages:
## [1] ggplot2_2.1.0 knitr_1.13
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5      digest_0.6.9     plyr_1.8.3       grid_3.3.0
##  [5] gtable_0.2.0     formatR_1.4      magrittr_1.5     evaluate_0.9
##  [9] scales_0.4.0     highr_0.6        stringi_1.0-1    rmarkdown_0.9.6
## [13] labeling_0.3     tools_3.3.0      stringr_1.0.0    munsell_0.4.3
## [17] yaml_2.1.13      colorspace_1.2-6 htmltools_0.3.5
```

References

Lawrence, Michael, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T Morgan, and Vincent J Carey. 2013. "Software for Computing and Annotating Genomic Ranges." *PLoS Comput. Biol.* 9 (8): e1003118. doi:10.1371/journal.pcbi.1003118.