R Markdown Example of MS Word

Jason Thomas

01/14/2020

# Introduction

This document shows how to use R Markdown to generate an MS Word document. First, you will need to install pandoc, which can be downloaded at <https://pandoc.org/installing.html>. Several R packages are also needed to create the final .docx file. They can be installed by running the following code in R Studio:

install.packages(c('rmarkdown', 'summarytools', 'furniture'))

# Markdown

Surround text with two asterisks, \*\*like this\*\*, to make text **bold**. Use a single asterisk, \*like this\*, to put text in *italics*. Here is a nested list (indentation is used for sub-levels):

* Point A
  + A.1
    - A.1.1
    - A.1.2
      * A.1.2.1
  + A.2
* Point B

You can include footnotes.[[1]](#footnote-23) If you want use a font that looks like computer code[[2]](#footnote-25), surround your text with back ticks, i.e., `like this`. Finally, if you want to insert a new line, simply include \ at the end of the line, as I do here  
and you can see that this second sentence fragment is on a new line  
and this third section is also on a new line.

## R Markdown

R Markdown is a special type of markdown that weaves R output directly into the final document.[[3]](#footnote-27) Use code chunks to include R output, which are created using three back ticks and some options surrounded by curly braces, like this:

```{r cars}

summary(cars)

```

This r code chunk (labeled cars) produces the following output in the MS Word document:

summary(cars)

speed dist

Min. : 4.0 Min. : 2.00  
1st Qu.:12.0 1st Qu.: 26.00  
Median :15.0 Median : 36.00  
Mean :15.4 Mean : 42.98  
3rd Qu.:19.0 3rd Qu.: 56.00  
Max. :25.0 Max. :120.00

Note that each code chunk must have a unique name (or R will complain about ‘duplicate labels’) and the document will fail to compile. We can also include R output in a line. For example, the third observation of the speed variable is 7. Now we will replace it with a random value

cars$speed[3] <- rnorm(1)  
cars$speed[3]

[1] -1.208879

and show that it has changed to -1.2088788.

# Tables

There are several options. The first is to use markdown syntax, but this is tedious and only practical for very basic tables. R packages, e.g., [summarytools](https://cran.r-project.org/package=summarytools) and [furniture](https://cran.r-project.org/package=furniture) are quite useful for making tables, as illustrated below.

## Markdown syntax

|  |  |  |
| --- | --- | --- |
| Col 1 | *Col 2* | **Column 3** |
| left justified | center | right justified |

or something like this

|  |  |  |
| --- | --- | --- |
| **Countries** | **Visited** | **Good Football team?** |
| G. Britain | Yes | Sometimes :) |
| India | Yes | No Asian Cup trophies yet |
| Zambia | Not yet | 1 Cup of Nations Title |

But this is a pain!

## R packages

This section includes examples with [kable](https://www.rdocumentation.org/packages/knitr/versions/1.26/topics/kable), and [furniture](https://cran.r-project.org/package=furniture).

There are some other options as well (but I think kable() and the furniture tools are best for MS Word)

* [summarytools](https://cran.r-project.org/package=summarytools)
  + [vignette](https://cran.r-project.org/web/packages/summarytools/vignettes/Recommendations-rmarkdown.html)
* [pander](https://cran.r-project.org/package=pander)
  + [vignette](http://rapporter.github.io/pander/knitr.html)

### *Example with kable*

This examples uses the Iris dataset that is included with the base installation of R.

# create table from iris data set (150 observations of plant species  
# with variables sepal length and width, and pedal length and names.  
names(iris)

[1] “Sepal.Length” “Sepal.Width” “Petal.Length” “Petal.Width” “Species”

table(iris$Species)

setosa versicolor virginica   
 50 50 50

# create a matrix of means  
tab1 <- aggregate(. ~ Species, data = iris, FUN = mean)  
specieNames <- gsub("\\.", " ", names(iris))  
  
# use kable() to format the table  
kable(tab1, digits = 2, col.names = specieNames, align = c('l', rep('c', 4)),  
 caption = "Table 1: Variable means for iris data.")

Table 1: Variable means for iris data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sepal Length | Sepal Width | Petal Length | Petal Width | Species |
| setosa | 5.01 | 3.43 | 1.46 | 0.25 |
| versicolor | 5.94 | 2.77 | 4.26 | 1.33 |
| virginica | 6.59 | 2.97 | 5.55 | 2.03 |

[This link](https://www.rdocumentation.org/packages/knitr/versions/1.26/topics/kable) contains the help file for the kable() function.

### *Example with furniture*

First, we’ll create a table of descriptive statistics using table1()

table1(mtcars, type = c("simple", "condensed"),  
 output = "pandoc")

|  |  |
| --- | --- |
|  | Mean/Count (SD/%) |
|  | n = 32 |
| mpg | 20.1 (6.0) |
| cyl | 6.2 (1.8) |
| disp | 230.7 (123.9) |
| hp | 146.7 (68.6) |
| drat | 3.6 (0.5) |
| wt | 3.2 (1.0) |
| qsec | 17.8 (1.8) |
| vs | 0.4 (0.5) |
| am | 0.4 (0.5) |
| gear | 3.7 (0.7) |
| carb | 2.8 (1.6) |

Another example, but for a subset of variables and different summary functions.

vNames <- c("Miles per gallon", "Number of cylinders",  
 "Gross horsepower")  
table1(mtcars, mpg, cyl, hp, var\_names = vNames,  
 caption = "\*\*Means and SD for subset of variables\*\*",  
 type = c("simple", "condensed"),  
 output = "pandoc")

**Means and SD for subset of variables**

|  |  |
| --- | --- |
|  | Mean/Count (SD/%) |
|  | n = 32 |
| Miles per gallon | 20.1 (6.0) |
| Number of cylinders | 6.2 (1.8) |
| Gross horsepower | 146.7 (68.6) |

vNames <- c("Miles per gallon", "Number of cylinders",  
 "Gross horsepower")  
table1(mtcars, mpg, cyl, hp, var\_names = vNames,  
 splitby = ~am,  
 caption = "\*\*Means and SD for subset of variables\*\*",  
 type = c("simple", "condensed"),  
 output = "pandoc")

**Means and SD for subset of variables**

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
|  | n = 19 | n = 13 |
| Miles per gallon | 17.1 (3.8) | 24.4 (6.2) |
| Number of cylinders | 6.9 (1.5) | 5.1 (1.6) |
| Gross horsepower | 160.3 (53.9) | 126.8 (84.1) |

crossTab <- tableX(mtcars, cyl, am, type = "count")  
kable(crossTab)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | Total |
| 4 | 3 | 8 | 11 |
| 6 | 4 | 3 | 7 |
| 8 | 12 | 2 | 14 |
| Total | 19 | 13 | 32 |

### *Example with regression tables*

model1 <- lm(mpg ~ hp, data = mtcars)  
model2 <- lm(mpg ~ hp + drat, data = mtcars)  
model3 <- lm(mpg ~ hp + drat + factor(gear), data = mtcars)  
tab2 <- regTab(model1, model2, model3)  
kable(tab2, align = c('l', rep('c', 3)))

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Model 1** | **Model 2** | **Model 3** |
| intercept | 30.099 | 10.790 | 16.306 |
| hp | -0.068 | -0.052 | -0.064 |
| drat |  | 4.698 | 3.510 |
| gear:4 |  |  | -0.276 |
| gear:5 |  |  | 3.761 |

# Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

1. To format footnotes (e.g., set the font size) you probably have to use a template. [Here](https://rmarkdown.rstudio.com/articles_docx.html) is an example. [↑](#footnote-ref-23)
2. I think it is called Consolas in MS Word. [↑](#footnote-ref-25)
3. More specifically, it actually call on another R package, knitr, to incorporate the R output. [↑](#footnote-ref-27)