A Better .docx Template for Quarto

Miles D. Williams[[1]](#footnote-20)

2024-05-20

**Abstract**: This document was produced using a Quarto (.qmd) template. It’s goal is to provide a better looking and functioning Word template for researchers to use when weaving together reports with their text and embedded code in Quarto.

***Keywords***: Quarto, Word, Templates

## 1 Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

While I much prefer creating .pdf files with Quarto, .docx files remain ubiquitous. Many academic publishers still prefer .docx submissions over .pdf as well. It therefore makes sense to have the best possible template at our disposal.

## 2 Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. However, for writing professional reports, you don’t want to have the content of your embedded code appear in your .docx file. This is why in the YAML of this Quarto file, the option echo: false has been set. There are a few other custom settings in the YAML as well.

## 3 Reporting Analysis Output

Data visualization, data tables, and regression tables can be a bear when rendering to .docx files, but I’ve come up with some strategies, both in the .docx template in this repository, and with the code examples in the Quarto file that will help.

These strategies make it possible to produce decent looking data viz alongside your text (see Figure 1).

|  |
| --- |
| Figure 1: Here’s a data visualization produced with ggplot2 |

It also makes it possible to produce passable summary statistics tables (see Table 1) and half-way decent looking regression tables (see Table 2).

Table 1: A table of summary statistics

|  | Mean | Median | SD | Min. | Max. |
| --- | --- | --- | --- | --- | --- |
| Miles per Gallon | 20.09 | 19.20 | 6.03 | 10.40 | 33.90 |
| Weight | 3.22 | 3.33 | 0.98 | 1.51 | 5.42 |
| Horse Power | 146.69 | 123.00 | 68.56 | 52.00 | 335.00 |

Table 2: Here is a regression table

|  | Model 1 | Model 2 | Model 3 |
| --- | --- | --- | --- |
| Control | 37.285\*\*\* (2.195) | 30.099\*\*\* (2.077) | 37.227\*\*\* (2.037) |
| Weight | -5.344\*\*\* (0.654) |  | -3.878\*\*\* (0.651) |
| Horse Power |  | -0.068\*\*\* (0.014) | -0.032\*\*\* (0.007) |
| N | 32 | 32 | 32 |
| Adj. R2 | 0.74 | 0.59 | 0.81 |

## 4 Appearance of Equations and Mathematical Operators

When writing a manuscript for a quantitative study, it’s always a good idea to be explicit about your analysis choices, especially any regression model specifications. Thankfully, rendering to .docx handles this pretty well. Here’s an example regression equation for Model 3 in Table 2:

And here’s one for Model 2:

Notice that they’re numbered! Unlike with rendering to .pdf documents, the Latex \tag{} command won’t work for numbering your equations, but you can use {#eq-name} instead without missing a beat: https://quarto.org/docs/authoring/cross-references.html#equations

## 5 A Code Appendix

So that it is easier for others to replicate your work, it may be a good idea to create a “Code Appendix” where all the code embedded in your report can be shown. This can be done with a very simple code block (you can see what it looks like in the Quarto file used to render this .docx file). With it, you can show all the underlying code you used in one central place. (A good tip when doing this is to make sure your embedded code is labeled in a helpful way so that readers can know what code produced what output.)

# Here's the code to produce Figure 1 ----  
  
## First open {ggplot2}  
library(ggplot2)  
  
## Now make a plot  
ggplot(mtcars) +  
 aes(x = wt, y = mpg) +  
 geom\_point(col = "gray") +  
 geom\_smooth(  
 method = "gam",  
 col = "steelblue",  
 se = F  
 ) +  
 labs(  
 x = "Weight",  
 y = "Miles per Gallon",  
 title = "Heavier cars are less fuel efficient"  
 )  
  
# Here's the code to produce Table 1 ----  
  
## First open {modelsummary}  
library(modelsummary)  
  
## Then produce a table of summary stats  
datasummary(  
 (`Miles per Gallon` = mpg) +   
 (Weight = wt) +   
 (`Horse Power` = hp) ~   
 (Mean = mean) +   
 (Median = median) +   
 (SD = sd) +   
 (Min. = min) +   
 (Max. = max),  
 mtcars  
)  
  
# Here's the code to produce Table 2 ----  
  
## Fit some regression models and save in a list  
list(  
 "Model 1" = lm(mpg ~ wt, mtcars),  
 "Model 2" = lm(mpg ~ hp, mtcars),  
 "Model 3" = lm(mpg ~ wt + hp, mtcars)  
) -> models  
  
## Make a regression table  
modelsummary(  
 models,  
 estimate = "{estimate}{stars} ({std.error})",  
 statistic = NULL,  
 vcov = "stata",  
 coef\_map = c(  
 "(Intercept)" = "Control",  
 "wt" = "Weight",  
 "hp" = "Horse Power"  
 ),  
 gof\_map = list(  
 list(  
 raw = "nobs",  
 clean = "N",  
 fmt = \(x) format(x, big.marks = ",")  
 ),  
 list(  
 raw = "adj.r.squared",  
 clean = "Adj. R2",  
 fmt = \(x) round(x, 2)  
 )  
 )  
)

1. Denison University, Granville, OH USA. Contact at mdwilliams@denison.edu. [↑](#footnote-ref-20)