Example APA7 Manuscript Made with Quarto and apaquarto

Natalie Dowling

MA Program in the Social Sciences, University of Chicago

Author Note

This example is based on the apaquarto template, created by W. Joel Schneider as part of the apaquarto extension for the Quarto publishing system. This manuscript and all associated materials are a demonstration of how to use these systems for the purposes of the course "From Data to Manuscript in R." The author is grateful Dr. Marisa Casillas for creating the original version of this course and to the teaching assistants and students who have contributed to its development in the years since. Author roles were classified using the Contributor Role Taxonomy (CRediT; https://credit.niso.org/) as follows: Natalie Dowling: conceptualization, writing

Correspondence concerning this article should be addressed to Natalie Dowling, MA Program in the Social Sciences, University of Chicago, 1155 E 60th St., Chicago, IL 60637, USA, Email: ndowling@uchicago.edu

Abstract

This document is a template demonstrating the apaquarto format. It includes examples of how to create figures and tables, as well as how to reference them in the text. The document is written in Quarto, a system for creating documents with R Markdown. The apaquarto extension provides a template for creating APA7-formatted manuscripts.

Keywords: R programming, ggplot2, data communication

Example APA7 Manuscript Made with Quarto and apaquarto

This document is a template demonstrating the apaquarto format. It includes examples of how to create figures and tables, as well as how to reference them in the text. The document is written in Quarto, a system for creating documents with R Markdown. The apaquarto extension provides a template for creating APA7-formatted manuscripts.

When rendered into html, pdf, or Word, this example produces an APA styled manuscript. Although the contents of the manuscript are not what you would expect in a psychology journal article, the formatting should demonstrate both the capabilities of the apaquarto extension and the basic template an actual manuscript would follow.

You can learn more about APA style in the APA Style Manual. Details about creating documents using the apaquarto extension in the documentation.

The example demonstrates the mechanics of markdown manuscripts using Quarto and the apaquarto extension, specifically. While many of the topics covered are the same in other markdown systems, like the older R Markdown or the `papaja` package, the specific syntax and options may differ.

Literature Review

Methods

Tables

In this section, we will go over how to create, render, and reference tables in apaquarto documents.

Results

Figures can be created in R chunks and rendered as figures in the text. They can be referenced in the text and will be numbered in the order they are rendered.

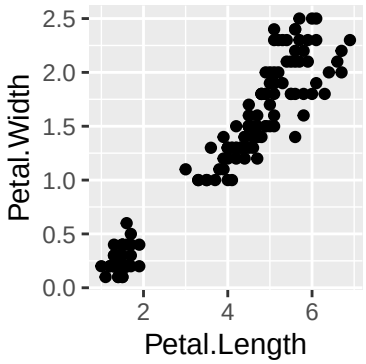
Figure chunk options

Figure chunks should use the quarto-preferred "comment style" chunk option settings. Minimally, they should include a label and a caption. The label should begin with "fig-" to be

recognized as a figure. The caption should be a string in title case. The image (file or object) should be rendered in the chunk.

Figure 1

The "Caption" (aka Title) of a Rendered Plot



In the chunk above, I produce Figure 1 a scatterplot of the `iris` dataset. It renders from the chunk (it is not saved to an object) at the location of the chunk in the manuscript. I can reference this plot in the text as `@fig-iris-rendered-plot`, which will render as "Figure 1" in the rendered document.

Figure 1, when rendered, has three components:

1. The plot itself
2. The figure *label*, which is the word "Figure" and the generated number in the order it was rendered
3. The figure caption, which is the title of the plot

Take note that the `fig-cap` option that assigned the "caption" is actually assigning with APA7 would refer to as the "title" of the figure. The thing you probably think of when you hear caption – the explanatory text below the figure – is called a "note." There is an option to include a note, but it is not required or included in this minimal example.

There are several important things to note about the simple figure chunk above:

1. The chunk options are in the "comment style" format. They are preceded by `#|` and are within the chunk, not the chunk header (the "`{r}` part).

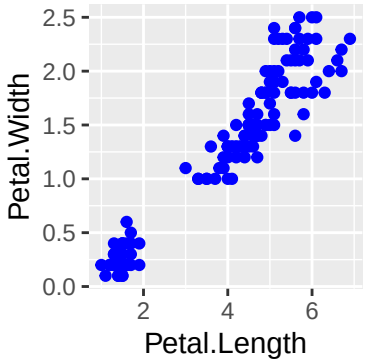
2. The first chunk option, label, is in the line *immediately following the chunk header*. This is important for the figure to be recognized as a figure. If there is anything above the label, including comments and whitespace, the figure will not render as a figure and the document may not render at all without error
3. The label begins with "fig-", which tells apaquarto that this is a figure. If the chunk has any other name, it will still render images in the chunk (including generated plots), but they will not be recognized as figures.
4. The caption in the `fig-cap` option. As discussed above, this "caption" is the title of the figure and should be in title case, with no period.

Figure 1 is rendered by actually creating a ggplot in the chunk. However, you can also create a ggplot object elsewhere and render it as a figure. This is useful if you want to use the same plot in multiple places in the document, or if you want to create a very complicated plot in a sourced script.

Creating a chunk that assigns a plot to an object does not render the plot, since R code that assigns an object does not return anything. It only creates the object `iris_plot`. To render the plot, you need to call the object in a chunk with the figure options.

Figure 2

The Caption of a Rendered Plot Object



I can render the plot in a separate chunk by calling the plot object `iris_plot` in the chunk. This will render the plot as a figure in the text. I can reference this plot as Figure 2.

Quarto, and the apaquarto extension, can accept many additional chunk options. Like the label and caption options, these are set in the comment style within the chunk. Here are some additional options that can be used in figure chunks:

1. `apa-note`: A note that appears below the figure caption. This is the explanatory text that you might think of as a "caption" in other contexts. Unlike the options that follow, this is a feature of apaquarto specifically, not markdown or Quarto.
2. `fig-scap`: A short caption that appears in the list of figures. This is useful for long figure captions that are unwieldy in a list of figures.
3. `fig-alt`: Alt text for accessibility. This will appear if you render as HTML and is useful for screen readers.
4. `fig-align`: The alignment of the figure. This can be "left", "right", or "center" (the default).
5. `fig-width` and `fig-height`: The width and height of the figure in inches. This is useful for controlling the size of the figure in the rendered document. The default is 7 inches wide and 5 inches tall.

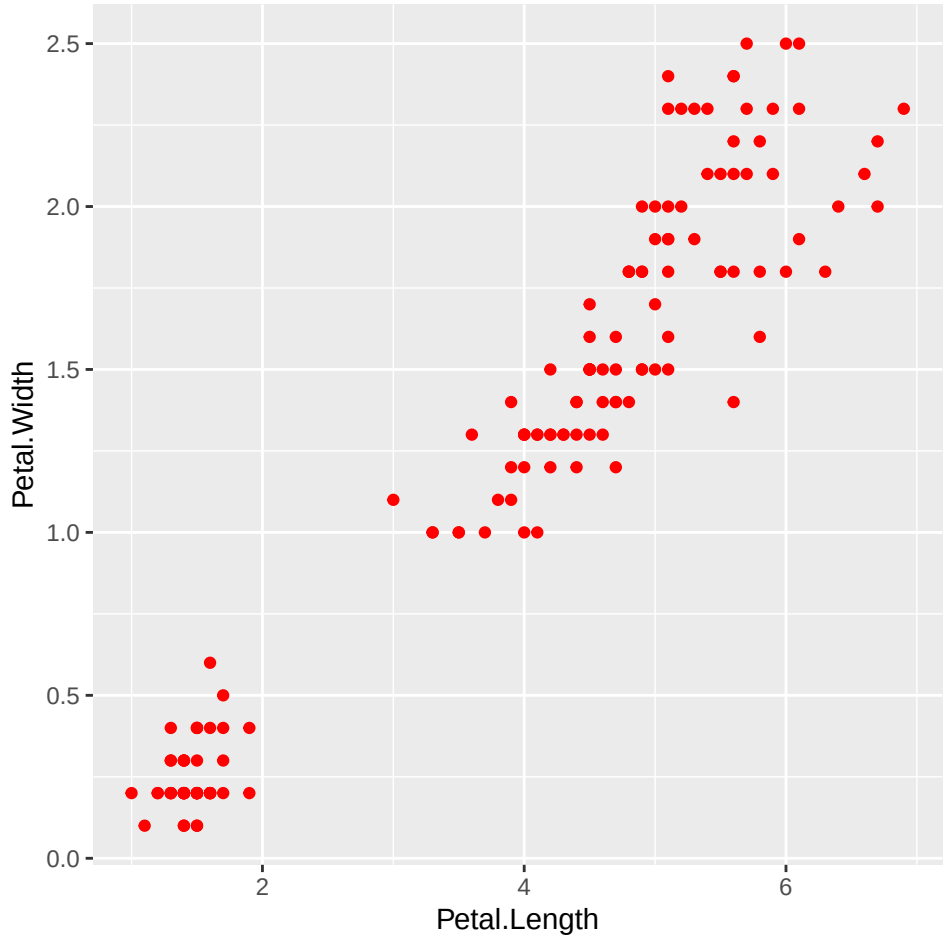
You can see all the effects of these chunk options in Figure 3. The figure is aligned to the right, has a width of 5 inches and a height of 6 inches, and has a note below the caption. The short caption is what will appear in the list of figures, and the alt text will appear when a reader mouses over the figure in an HTML document (it does not do anything in pdf or Word documents).

Referencing figures. As seen in the text above, figures can be referenced in text using the `@` symbol followed by the figure label. This will render as "Figure X" in the rendered document, where X is the order in which the figure was rendered. The figure label should be unique and begin with "fig-" to be recognized as a figure.

Plots are the only thing that count as figures. In APA documents, all images are typically treated as figures. You can include images as figures in your document by rendering them in a chunk using the `include_graphics()` function from the knitr package. Like with any other

Figure 3

A Plot With More Chunk Options



Note. A note appearing below the figure.

figure, the chunk required a label beginning with "fig-" and a caption and may take additional chunk options.

For images, it's usually better to use `out-width` or `out-height` rather than `fig-width` and `fig-height`. The image files already have an inherent size (unlike rendered plots), which can create problems when you try to give them new absolute dimensions (with the `fig-` options). The `out-` options let you use a relative sizing as a percentage, which is usually more reliable.

Figure 4

Types of Scientific Papers

Note. From xkcd (#2456) by Randall Monroe

Why not take the opportunity to look at a few more xkcd comics?

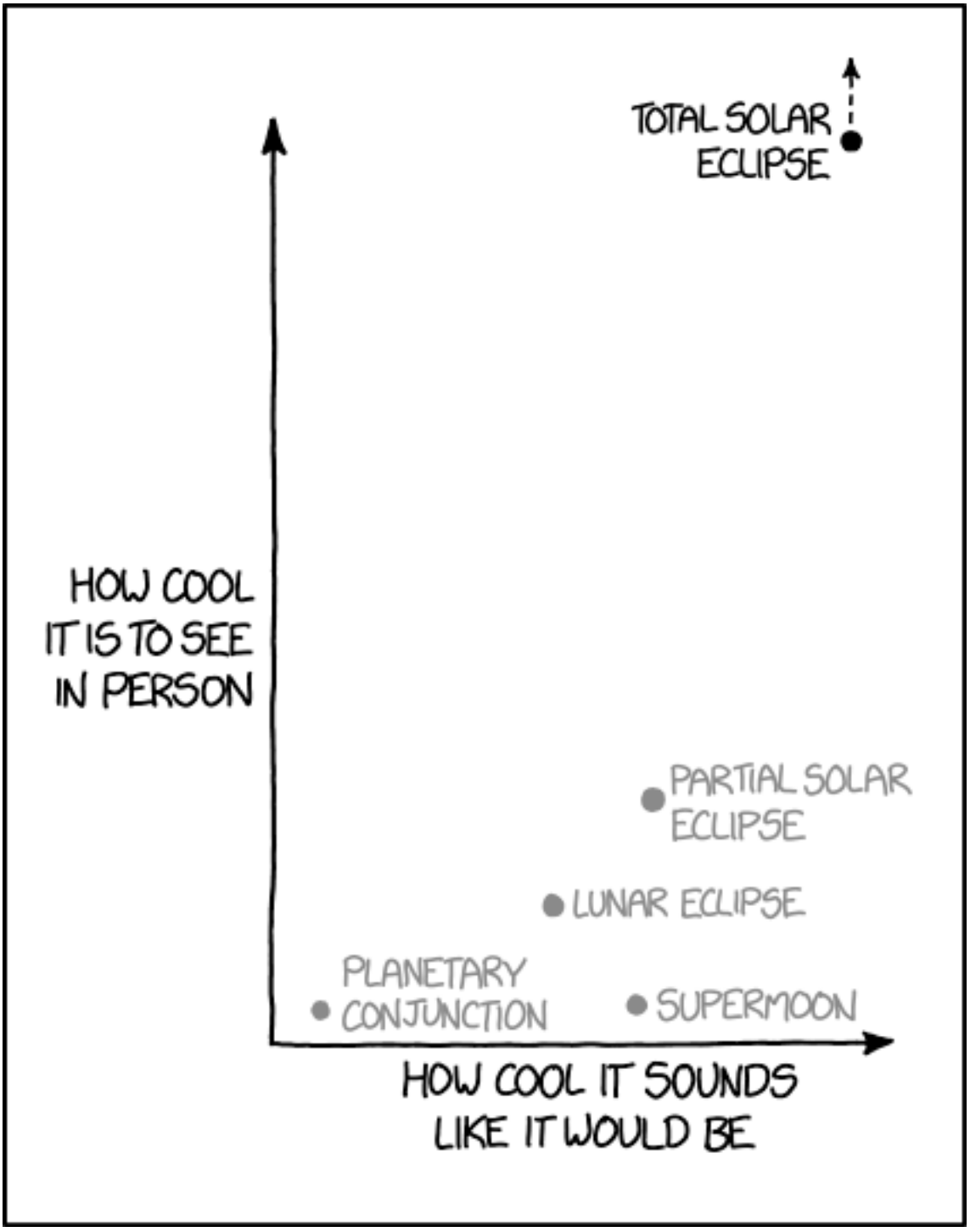
When you knit this document to a pdf, knitr/Quarto will try to pick the best location to include it. If your image is very large, or if there is very little text between images, the location it chooses may not be precisely where you put the chunk. You can force the image to render exactly

where the chunk is with the option `fig-pos: "H"`. Either way–the default or the forced hold–can produce unexpected consequences, so keep an eye out for issues and try out alternatives as needed.

Back to xkcd. Here's a good one about types of eclipses:

Figure 5

Types of Eclipses



Note. From xkcd (#1880) by Randall Monroe

And of course there's this classic about literally everyone's experience using git:

Figure 6

The Tragedy of git

Note. From xkcd (#1597) by Randall Monroe

Love a good xkcd. They're all great, but right now Figure 6 really speaks to me. (You too, maybe?) Figure 5 is good and it's got a plot, but it's not as relevant to our class as Figure 4 or Figure 6.

I have a point. Notice that the numbers assigned to each figure are based on the order in

which they are rendered, not the order in which they are referenced.

Analyses

In the next section, we will talk about running and referencing statistical analyses.