Introduction to R Markdown

Taylor N. Carlson

March 21, 2018

Contents

What is R Markdown?

R Markdown is a fantastic tool that allows you to write your code and analysis all in one document. Creating data reports to help you get to know your data is much simpler, largely because it helps keep everything organized. It also helps you be more transparent in your research since it allows you to show the code you used to generate your results right next to them if desired. If you make a coding error (we've all been there . . .) you don't have to go back and redo every single table, figure, or in-text description. R Markdown will update it all for you!

With Markdown, you can write out to HTML, PDF, or MS Word. The syntax is a little bit different for each output. My personal preference is PDF. You will need to have LaTeX installed for this to work. If you don't have LaTeX installed, you should still be able to follow along and have your output be HTML instead of a PDF.

The Basics

Header / Preamble

What you see at the top of this .rmd file is a preamble. It essentially tells us how we want our final document to be formatted. For example, we can see the title, author, and date of the document up at the top. After that, we can see how big we want our margins to be (2cm in this case). We can also see a bunch of features under "output." Note on the PDF output that we can choose whether to have a table of contents (toc) or not, as well as how many levels of that table of contents we want to have (3 in our case). The table of contents is automatically hyperlinked, which is awesome.

Writing R Code

Any time you want to start writing your R code, you'll create a "code chunk" by using three tick marks (above the tab key) and curly brackets like this:

```
# Here's where I'd start my R Code
# I'm using the pound sign (or "hashtag" for you millennials)
# to -- just like I would if I were writing in a .R file
# You end the code chunk with three more tick marks
```

The R Markdown syntax is fairly straightforward. Here's what's going on inside the curly brackets in the example above:

- r — tells Markdown that we're going to be writing R Code
- intro, — this is optional, but it just gives the codechunk a name. This is helpful for when you start making figures and tables, so you can refer to them in the text. This can only be one word, but you can use underscores or periods as separators (e.g. intro_chunk,). You should end with a comma.
- echo=TRUE — this tells us that we want our pdf to "echo" (include) our code. If echo=TRUE, our PDF will include everything in the code chunk. This is useful if you want to show someone the code you used to generate a result (e.g. on a problem set, when working with a collaborator, when having trouble with your code and you want help from someone). Usually, you will probably not want this code in your data report, so most of the time you'll want echo=FALSE.
- eval=TRUE — this tells us that we want to actually run the code in that chunk. Sometimes you end up writing code to generate a figure that you end up wanting to cut from the paper, but don't want to delete the code altogether. You can just set eval=FALSE to tell R to ignore that whole code chunk.
- You can add other information between these curly brackets for more advanced options. For example:
  - results="asis" — this is useful when you generate a table with xtable() or stargazer().
  - fig.cap="blah blah blah" — this is where you'd type in the caption to a figure (presumably you'd write something more informative than blah blah blah, but you do you)
  - fig.height=5 — this is how you can customize the height of a figure (here I've set it to 5 inches)
  - fig.width=5 — this is how you can customize the width of a figure (here I've set it to 5 inches)

Example Loading in Data

```
## [1] 3649  106
```

Now if we want to describe our data a bit, we **could** just type the numbers we saw in our R Console in like this: There are 3,649 rows and 106 columns. But, what if we later realize that some of these rows need to be dropped? Or that some columns were included in error? We'd have to go back and re-type the correct numbers. What a pain! R Markdown can help fix this by letting you write flexible results. We can write: There are 3649 rows and 106 columns in our dataset. Here, we've used a single tick mark around our code in the text. We have the letter r to tell Markdown we're writing R code. Finally, we just write our code as normal. Much cleaner, eh?

Notice also that I used asterisks to **bold** something.

Creating Figures

We could have a whole session on how to create figures in R. Instead of spending time doing that, the purpose of this section is to show you how to generate figures in R Markdown that are embedded within your data report or paper. Just as before, we begin by creating a code chunk with three tick marks. In this case, we're going to add the arguments referenced above to set the height and width (in inches) of the figure we're going to produce. In this specific example, I know that my figure is going to be wide, so I'm going to set the width to be longer than the height. I also add a caption for my figure, keeping it in quotes.

Next I write my R code just as I normally would. In this case, I'm creating a barplot that shows the average feeling thermometer toward Trump (my DV) for each value of strength of partisanship (my IV).

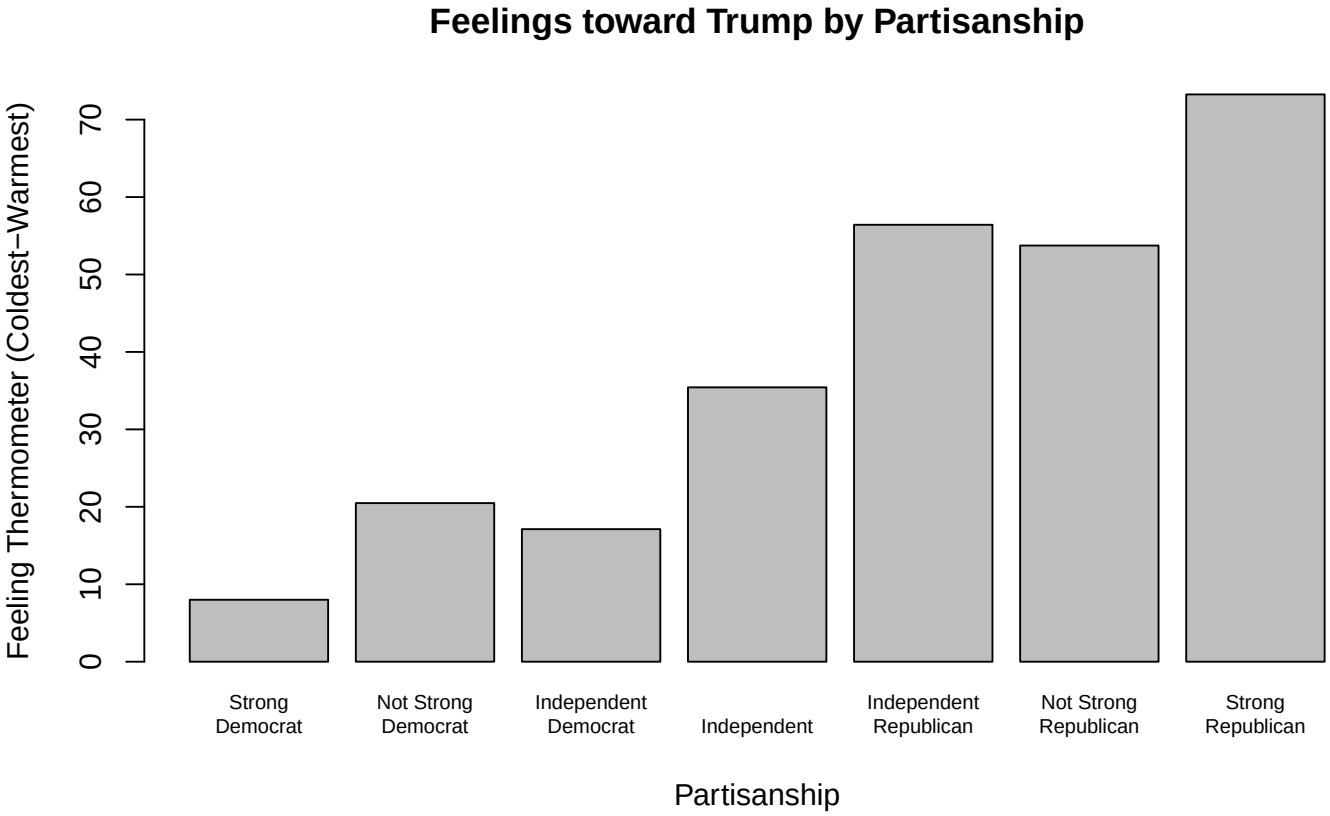

Figure 1: Feeling thermometer toward President Trump by strength of party ID.

Now if I want to reference the figure I just created, I'm going to write: Figure 1. If you're reading the .pdf, the code I typed was [r figr("feeling_pid", type="Figure")], but with a tick mark replacing each bracket. I use "feeling_pid" because that's what I named that code chunk. Now I can write things like "As shown in Figure 1, Strong Republicans had much warmer feelings toward President Trump than Strong Democrats."

Creating Tables

Tables are also a useful way to show your results. There are some awesome R packages that help create pretty tables for you: stargazer and xtable. If you don't have these packages installed, you should install them by running the following R code:

install.packages("stargazer")

install.packages("xtable")

If you already have them installed, you'll just need to load them, which I've already done in our first code chunk up at the top.

Tables with xtable

We can then refer to our table in much the same way we referred to our figure before. The key difference now is that instead of writing type="Figure", we're going to write type="Table". This is to make sure that when R Markdown is automatically numbering everything for us (thanks!) it doesn't combine or mix up tables and figures. If we had left this table as type="Figure", it would list it as Figure 2, even though it's Table 1. So, if I want to refer to my table, I will write: As shown in Table 1.

Table 1: Party ID by Gender

	1. Male	2. Female
1. Strong Democrat	0.17	0.24
2. Not very strong Democract	0.11	0.16
3. Independent-Democrat	0.12	0.11
4. Independent	0.14	0.12
5. Independent-Republican	0.15	0.10
6. Not very strong Republican	0.13	0.11
7. Strong Republican	0.18	0.15

Tables with Stargazer

We might also want to include a regression table in our results. To include a regression table that looks nice, we'll use the stargazer package. You'll need to store your regression in an object and then use stargazer on that object (see code below).

Table 2:

	Dependent variable:
	V17
V16	-0.672^{***}
	(0.013)
Constant	64.969^{***}
	(0.691)
Observations	3,579
R^2	0.438
Adjusted R^2	0.438
Residual Std. Error	26.015 (df = 3577)
F Statistic	$2{,}791.580^{***}$ (df = 1; 3577)
Note:	*p<0.1; **p<0.05; ***p<0.01

This looks a little sloppy. Our variable names don't make any sense to the average reader, there isn't a clear title. This is not going to be helpful. See the next code chunk for how we can add in variable names and titles.

Table 3: Relationship Between Feelings toward Clinton and Feelings toward Trump

	Dependent variable:
	Feelings toward Trump
Feelings toward Clinton	-0.672^{***}
	(0.013)
Constant	64.969^{***}
	(0.691)
Observations	3,579
R^2	0.438
Adjusted R^2	0.438
Residual Std. Error	26.015 (df = 3577)
F Statistic	$2{,}791.580^{***}$ (df = 1; 3577)
Note:	*p<0.1; **p<0.05; ***p<0.01

This looks **much** nicer! But, now let's say we want to do a multivariate regression in which we control for feelings toward Obama (V15) and feelings toward Big Business (V75B).

Table 4: Relationship Between Feelings toward Clinton and Feelings toward Trump

	Dependent variable:
	Feelings toward Trump
Feelings toward Clinton	-0.222^{***}
	(0.020)
Feelings toward Obama	-0.489^{***}
	(0.019)
Feelings toward Big Business	0.223^{***}
	(0.018)
Constant	60.826^{***}
	(1.224)
Observations	3,493
R^2	0.553
Adjusted R^2	0.552
Residual Std. Error	23.188 (df = 3489)
F Statistic	$1{,}436.851^{***}$ (df = 3; 3489)
Note:	*p<0.1; **p<0.05; ***p<0.01

Let's say we wanted to put our bivariate regression in the same table as our multivariate regression. See the code chunk below.

Table 5: Relationship Between Feelings toward Clinton and Feelings toward Trump

	Dependent variable:	
	Feelings toward Trump	
	(1)	(2)
Feelings toward Clinton	-0.672^{***}	-0.222^{***}
	(0.013)	(0.020)
Feelings toward Obama		-0.489^{***}
		(0.019)
Feelings toward Big Business		0.223^{***}
		(0.018)
Constant	64.969^{***}	60.826^{***}
	(0.691)	(1.224)
Observations	3,579	3,493
R^2	0.438	0.553
Adjusted R^2	0.438	0.552
Residual Std. Error	26.015 (df = 3577)	23.188 (df = 3489)
F Statistic	$2{,}791.580^{***}$ (df = 1; 3577)	$1{,}436.851^{***}$ (df = 3; 3489)
Note:		*p<0.1; **p<0.05; ***p<0.01

Resources

- https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf
- https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf
- https://rmarkdown.rstudio.com/lesson-1.html