

Comments on R Markdown

MAS 627 - Programming for Data Analytics

8/17/2020

.Rmd vs .R

You will probably use .R (R Script) files way more than you use .Rmd (R Markdown) files. I have friends that have been R programmers for 10+ years and don't even know what R Markdown is.

Use R Markdown when you need to create a document that someone else can read. This document may or may not include R code and output. Things like papers, reports, presentations, summaries of findings, walking the reader through an analysis step-by-step with comments and clarifications. I will have you use R Markdown on most homeworks, so I can see your code and output clearly in a PDF without having to open your code in R and run it myself.

Use an R Script when you are just coding and don't need a document. Most of your work will fall into this category; most of the time you are just cleaning data or fitting models. Forcing this into R Markdown is not impossible but it's simply not necessary.

You make more mistakes in R Markdown when you are new. Your document will crash and you will not understand why. Since you're still learning R, you won't know if you have a mistake in the R code or if it's something Markdown specific. For that reason, I encourage you to spend most of your time in .R scripts, and transition to Markdown when you're comfortable.

Tips for using R Markdown

- **Never copy/paste into an R Markdown document.**
 - This almost never works. Things like quotes will paste in as unrecognized characters. Certain fonts may or may not be recognized. Many students ran into this problem last year in the MAS 637 class. Rather than type out regression equations, e.g. $y = \beta_0 + \beta_1 X + \epsilon$, they tried to copy/paste from the notes. R will not recognize the greek symbols and throw an error when you try to knit the document. Similarly, be careful if you are on an international keyboard, as many fonts/characters may not be recognized.
- **Knit your document all the time.**
 - Knit your document every time you add anything to it. If you add one paragraph, one equation, one code chunk, re-knit the document. If you do this, any mistakes (either in your markdown syntax or R code) will be found quickly. If your document knits, you add a small chunk of R code, and it no longer knits, you know right away that the problem is in the code you just added. If you forget to do this and wait until the end to knit, it will be much harder to find the problem.
- **Your document has to be completely self contained.**
 - Regardless of what is in your R environment, when you click “Knit” markdown only reads, from start to finish, what is inside the document. If you need a dataset in your document, you need to make sure you read it in within the document. If I read it in to my R environment via an R Script or the Console, but not inside the document itself, it won't find that dataset and throw an error message when I knit.
- **Be careful not to include unnecessary output in your document**
 - If you are like me, you like to look at your data every time you make a change to it. For example, every time I modify a column in a dataset, I tend to run `head(...)` to look at the data and make sure it looks as I expect it to. I might run `head(...)` 20+ times while manipulating a dataset. If I left all of these calls to the `head(...)` function in my document, I would end up with tens or hundreds of pages, with the dataset printed out over and over and over. Use some common sense here. Don't do this. I've seen students print entire datasets and submit documents that were over 1,000 pages long. Don't do this.