

Title: Knitting Beautiful Documents in RStudio

Header: Markdown

This document was written using [Markdown](#) in [RStudio](#). RStudio is a wonderful IDE for writing R scripts and keeping track of variables, dataframes, history, packages and much more. Markdown is a simple formatting syntax for authoring web pages. It allows for easy formatting, for example

```
# Header 1
## Header 2
### Header 3
#### Header 4
#### Header 5
```

Header 1

Header 2

Header 3

Header 4

Header 5 Code can be embedded in documents with nice formatting using backticks (symbol below the tilde in the upper left of an American qwerty keyboard). For example, if you were to type the following (without the >)

```
>`{r}
>install.packages("lme4", repos = 'http://cran.us.r-project.org')
>`
```

you could run that block in R and install the lme4 package. In the markdown document it would show up as:

```
{r} install.packages("lme4", repos = 'http://cran.us.r-project.org')
```

You should try this with the [knitr package](#). knitr is a fantastic package for converting the R markdown file (.Rmd) into an HTML document. The design of knitr allows any input languages (e.g. R, Python and awk) and any output markup languages (e.g. LaTeX, HTML, Markdown, AsciiDoc, and reStructuredText). I find Markdown to be the easiest language to use to create documents for coursework/assignments. knitr is dynamic, so it will run your code and print the results below it. The best part is if you get new data or find an error in your code, you just re-knit the file and it's updated a few seconds later.

In RStudio, once you install knitr a button shows up that you can click to knit the markdown file (produce your html document). When you click the **Knit HTML** button a web page will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
{r} summary(cars)
```

These will be embedded and run in the document, but you can also run single lines or blocks (chunks) of code in RStudio to make sure they work before knitting the document. You can also embed plots, for example:

```
{r fig.width=7, fig.height=6} plot(cars)
```

This is great but after knitting you are left with an HTML document. What if you want a PDF or Word Document? There may be other ways to do this, but I use [pandoc](#) to covert the HTML file into other forms such as PDF or MS Word Documents. After installing pandoc, you use it via the command line. I use a Mac, so I just open the Terminal to get to my Bash Shell and `cd` to the folder containing my document, then just run

```
pandoc Knitting_Beautiful_Documents_in_RStudio.Rmd -o Knitting_Beautiful_Documents_in_RStudio.pdf
```

and I have a PDF. Unfortunately, the pandoc default is to have huge margins that look ridiculous. To correct for that you can save a style file in the folder with the document you're converting. To do that you just create a text file with the line:

```
\usepackage[vmargin=1in,hmargin=1in]{geometry}
```

and save it as something like `margins.sty`. Then in the command line you run

```
pandoc Knitting_Beautiful_Documents_in_RStudio.Rmd -o Knitting_Beautiful_Documents_in_RStudio.pdf -H margins.sty
```

Now when you view the PDF it should have 1 inch margins and look much nicer. That is about it for making attractive, dynamic documents using R, RStudio, knitr, and pandoc. It seems like a lot but after doing it once or twice it becomes very quick and easy. I have yet to use it but there is also a package [pander](#), which allows you to skip the command line step and do the file conversion right in R (RStudio).

Using Markdown in RStudio with knitr is wonderful for creating reports, class assignments, and homework solutions (as a student, you'd really impress your instructor), but other people take it even further. [Karthik Ram](#) of [rOpenSci](#) fame has a great [blog post](#) of how to extend this system to include citations and bibliographies that [allow you to ditch word](#).

For a nice Markdown reference list, check out this useful [cheat sheet](#)

Finally, if you want to see my Markdown code and everything that went into making this, you can find in my [GitHub repository](#)