Getting started with R Markdown

497 / 597 Reproducible Research

Richard Layton

Rose-Hulman Institute of Technology
Fall 2018

Image from Wickham & Grolemund, *R for Data Science*, http://r4ds.had.co.nz/r-markdown.html

Tasks to complete before starting a report file

installed locally

obtained an account, created a project 1 repo

created an R Project for project 1

created your directory for project 1

pushed to the repo

Plan to create separate Rproj/repos for each project

497 students

```
me497_reproducible_research
  |-- practice_work/
  |-- project_1/
  `-- project_2/
```

597 students

```
me597_reproducible_research
  |-- practice_work/
  |-- project_1/
  |-- project_2/
  `-- project_3/
```

- the top level is the course folder
- each project folder is an Rproj/repo

  `practice_work/`

  `project_1/`

  `project_2/`

  `project_3/`

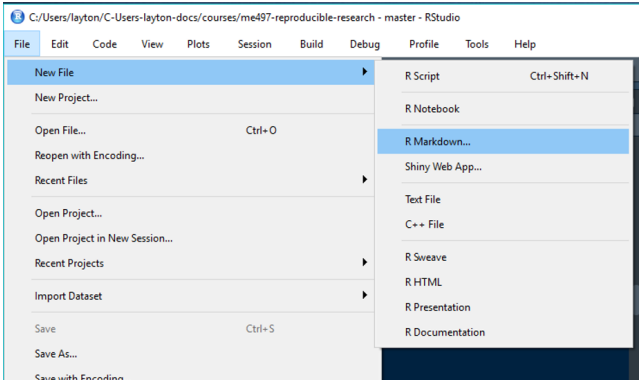
Launch your Rproj and open a new Rmd file

R Markdown is installed with RStudio

In RStudio, launch the project, for example, **`project_1.Rproj`**

Open an new Rmd file:

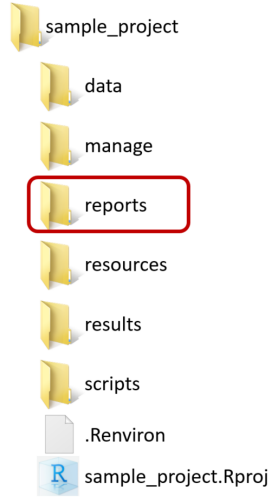
File > New File > R Markdown > OK



Save the Rmd file to your reports directory

Select a file name consistent with your file naming scheme, e.g.,

`301_report_2018-09-04.Rmd`



An R Markdown script has three types of text

- YAML header, surrounded by ---

```
---
title: "Sample report"
author: "Richard Layton"
date: "September 1, 2016"
output: word_document
---
```

- Prose mixed with markup text, e.g.,

```
# Section heading
Prose with markup for *italics*, **bold**, `inline R code`,  etc.
```

- Chunks of R code surrounded by ```

````
```{r setup, echo=FALSE}
# a hashtag in a R code chunk is a comment
library(ggplot2)
library(dplyr)
smaller <- diamonds %>% filter(carat <= 2.5)
```
````

The default Rmd file illustrates common mark-ups

- YAML header
- knitr setup
- section heading
- URL active link
- ****bold****
- code chunks
- **`` `inline code` ``**

````
---
title: "Untitled"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for
authoring HTML, PDF, and MS Word documents. For more details on using R
Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document 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 cars}
summary(cars)
```

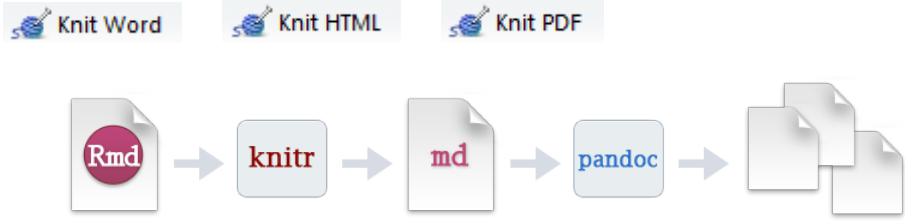
## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent
printing of the R code that generated the plot.
````

Knit the document any time



- R Markdown sends the .Rmd file to knitr
- knitr executes the code chunks and creates a markdown (.md) document that includes the code and its output
- pandoc process the .md file to create the output file

Image from Wickham & Grolemund, *R for Data Science*, http://r4ds.had.co.nz/r-markdown.html

Writing an R Markdown script...

script

````
---
title: "Sample report"
author: "Richard Layton"
date: "September 1, 2016"
output: word_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.
````

Writing an R Markdown script...and knitting to Word

script

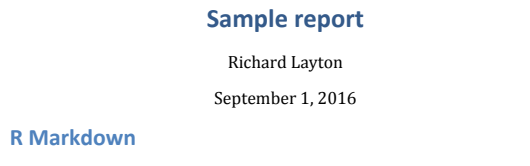
````
---
title: "Sample report"
author: "Richard Layton"
date: "September 1, 2016"
output: word_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting
syntax for authoring HTML, PDF, and MS Word documents. For more
details on using R Markdown see <http://rmarkdown.rstudio.com>.
````

output



Sample report

Richard Layton

September 1, 2016

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Add a code chunk

script

```
When you click the **Knit** button
a document 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)
```
```

Add a code chunk

script

```
When you click the **Knit** button
a document 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)
```
```

output

When you click the **Knit** button a document 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)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Add a graph

script

```
## Including Plots

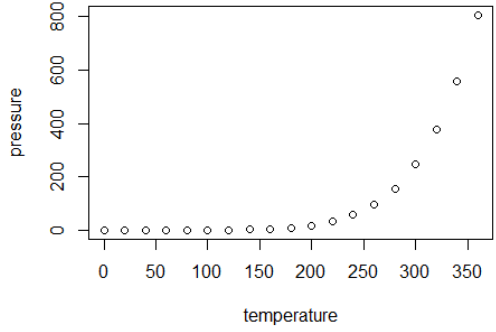
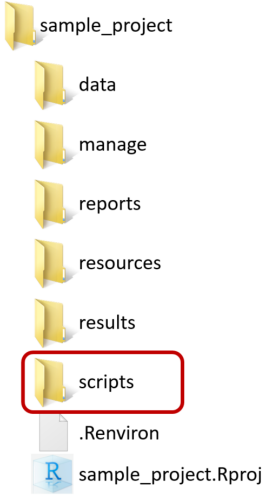
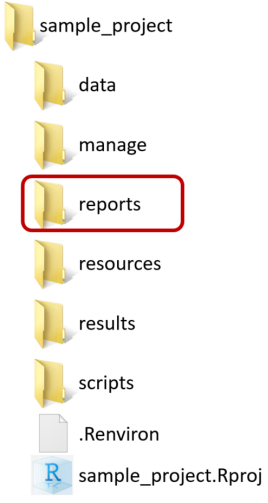
You can also embed plots, for example:

```{r echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter
was added to the code chunk to prevent
printing of the R code that generated
the plot.
```

Add a graph

script

````
## Including Plots

You can also embed plots, for example:

```{r echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter
was added to the code chunk to prevent
printing of the R code that generated
the plot.
````

output

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

We typically write our code in stand-alone R scripts



401_data-tidy.R	reads raw data, writes it in tidy format
402_data-wide.R	reads tidy data, writes it in wide format
403_data-graph.R	reads tidy data, edits it for graph
404_calibr-graph.R	reads graph data, creates the figure

Then execute those scripts from the Rmd script




````
---
output: word_document
---

```{r echo=FALSE}
library(knitr)
opts_knit$set(root.dir="../")
```

# Introduction
Prose to explain the context of the report

# Data
Prose to explain the data

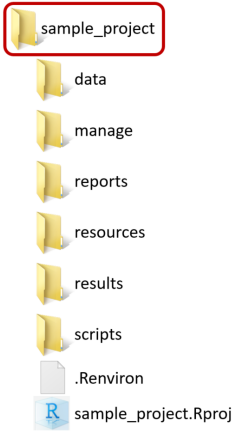
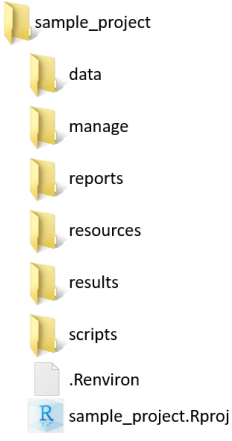
```{r echo=FALSE}
# create the data in its 3 forms
source("scripts/401_data-tidy.R")
source("scripts/402_data-wide.R")
source("scripts/403_data-graph.R")
```

More prose
````

Rmd script (continued)

````
Prose to discuss the data table

```{r echo=FALSE}
# import and print tabulated data
df <- readRDS("results/402_data-wide.rds")
kable(df)
```

# Results
Prose to explain the results

```{r echo=FALSE}
# create the graph
source("scripts/404_calibr-graph.R")

# import the graph
knitr::include_graphics("../results/404_calibr-graph.png")
```

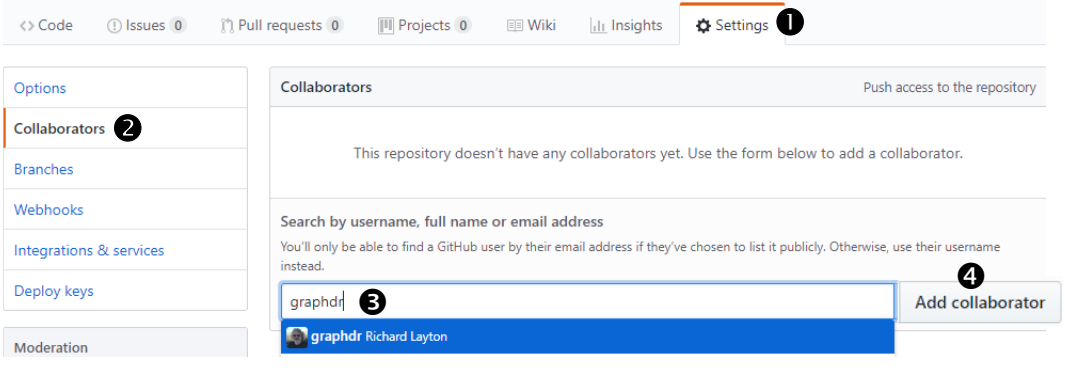
# Conclusion
Prose to explain the conclusions
````

File paths are relative to the R working directory



The *working directory* is where R, by default,

- looks for files you ask it to run
- saves file you write to disk

When you run an RStudio Project,

- the project directory is the working directory
- file paths are relative to the working directory

Relative file paths support reproducibility



- Cloning your repo to my computer, I can run your code because all file paths are relative
- Absolute file paths are non-reproducible

Relative file paths are how we link files explicitly

The Rmd script includes code chunks that run other R scripts, for example,

- read the raw data file, tidy the data, and save it to file,

```
```{r}
source("scripts/401_data-tidy.R")
```
```

- read the tidy data, manipulate it for graphing, and write it to file,

```
```{r}
source("scripts/403_data-graph.R")
```
```

- read the graph-ready data, create the graph, and write it to file,

```
```{r}
source("scripts/404_calibr-graph.R")
```
```

Some tools assume the file folder is the working directory

- *YAML* arguments: from Rmd, up to project directory, down to the file

```
---
bibliography: "../resources/portfolio.bib"
---
```

- *knitr:* reset the knitr root directory one level up

````
```{r}
knitr::opts_knit$set(root.dir="../")
```
````

- importing graphics using knitr syntax

````
```{r}
include_graphics("../results/404_calibr-graph.png")
```
````

- importing graphics using Rmd syntax

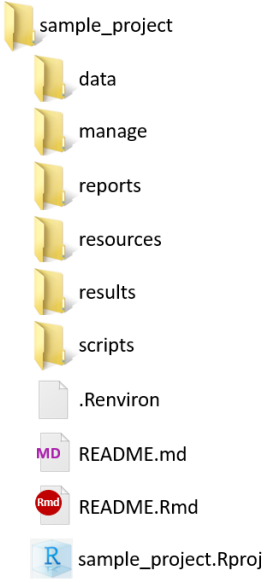
```
![](../results/404_calibr-graph.png)
```

Next steps for your project

From your GitHub repo, invite me (graphdr) to be a collaborator



Edit the README file



- README is the landing page of your repo
- README introduces the reader to the project
  - project context and summary
  - what the project looks like in action
  - how the reader might use the project
- README.Rmd resides at the top level of the directory
- Knit README.Rmd using YAML GitHub output

  `output: github_document`
- Stage, commit, push

For more information on writing a README file, see https://github.com/noffle/art-of-readme