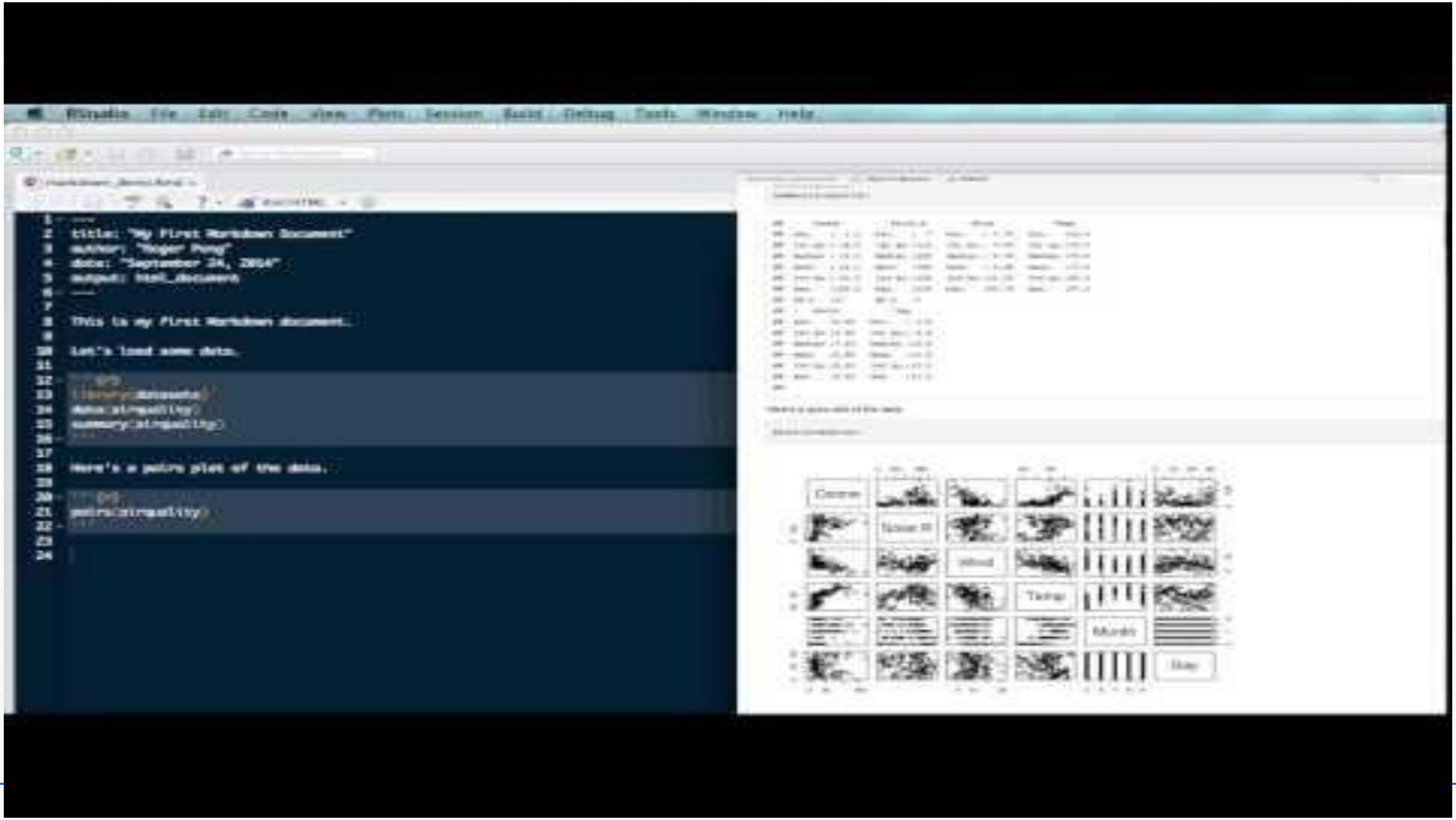
R Markdown

R Markdown tutorial

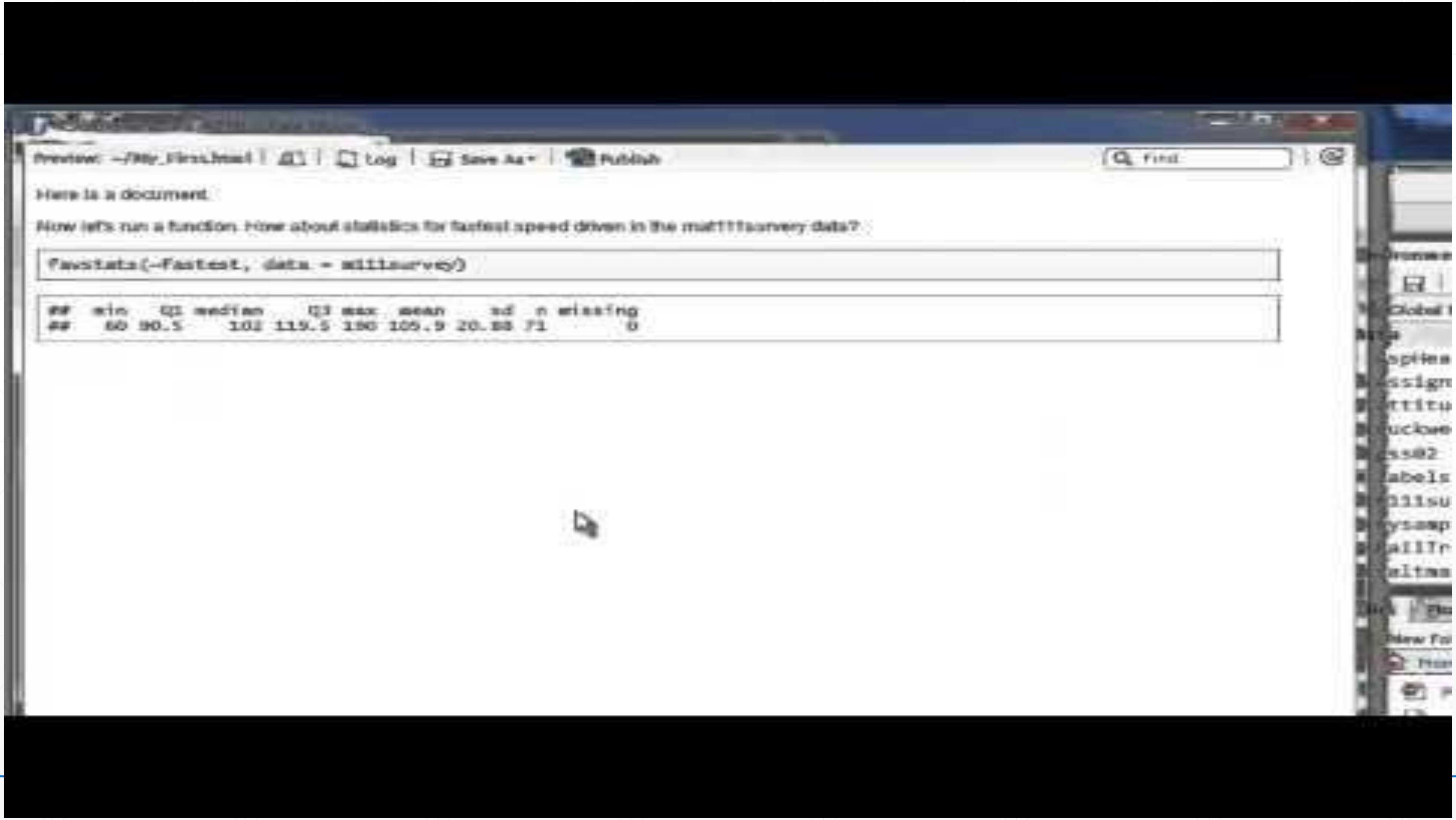
- R Markdown with RStudio
  - https://www.youtube.com/watch?v=DNS7i2m4sB0
- Introduction to R Markdown
  - https://www.youtube.com/watch?v=-apyD5f9nwg
- Getting started with R Markdown
  - http://rmarkdown.rstudio.com/lesson-1.html
- Markdown: Dingus
  - http://daringfireball.net/projects/markdown/dingus

R Markdown with RStudio



~~~
---
title: "My First Markdown Document"
author: "Roger Peng"
date: "September 24, 2014"
output: html_document
---

This is my first Markdown document.

Let's load some data.

```{r}
library(datasets)
data(airquality)
summary(airquality)
```

Here's a pairs plot of the data.

```{r}
pairs(airquality)
```

~~~

Introduction to R Markdown



Here is a document.

Now let's run a function. How about statistics for fastest speed driven in the m111survey data?

```
favstats(~Fastest, data = m111survey)
```

```
##  min   Q1 median    Q3 max  mean    sd  n missing
##   60 90.5    102 115.5 190 105.9 20.88 71       0
```

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

RStudio

.Rmd files

An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

Reproducible Research

At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents

You can choose to export the finished report as a html, pdf, MS Word, ODT, RTF, or markdown document; or as a html or pdf based slide show.

Workflow

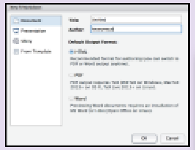
1 Open a new .Rmd file at File ▸ New File ▸ R Markdown. Use the wizard that opens to pre-populate the file with a template

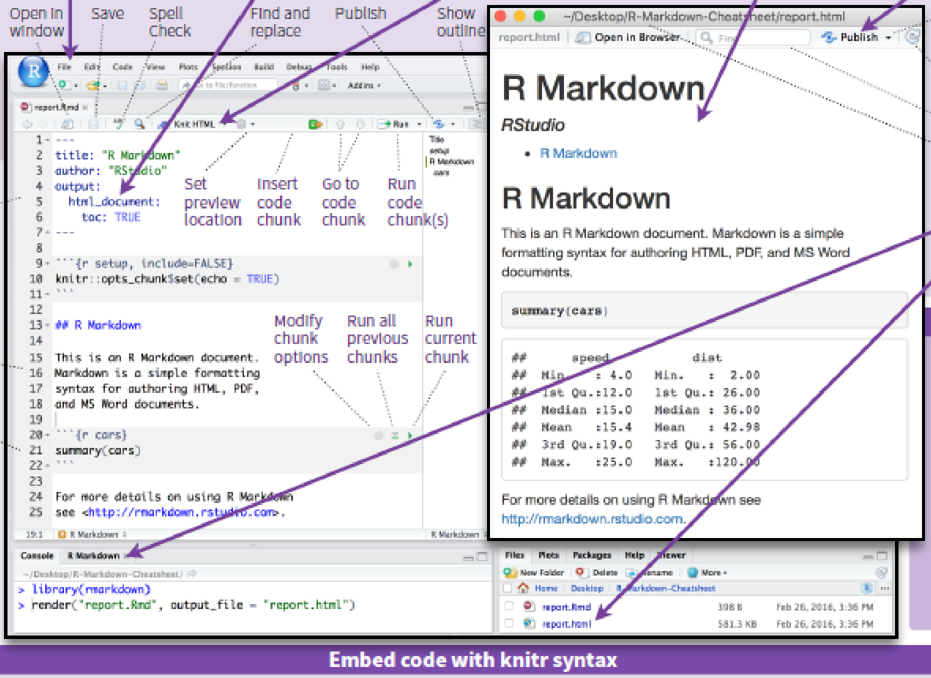
2 Write document by editing template

3 Knit document to create report Use knit button or **render()** to knit

4 Preview Output in IDE window

5 Publish (optional) to web or server





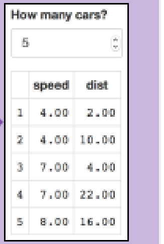
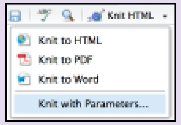
~~~
---
title: "R Markdown"
author: "RStudio"
output:
  html_document:
    toc: TRUE
---

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

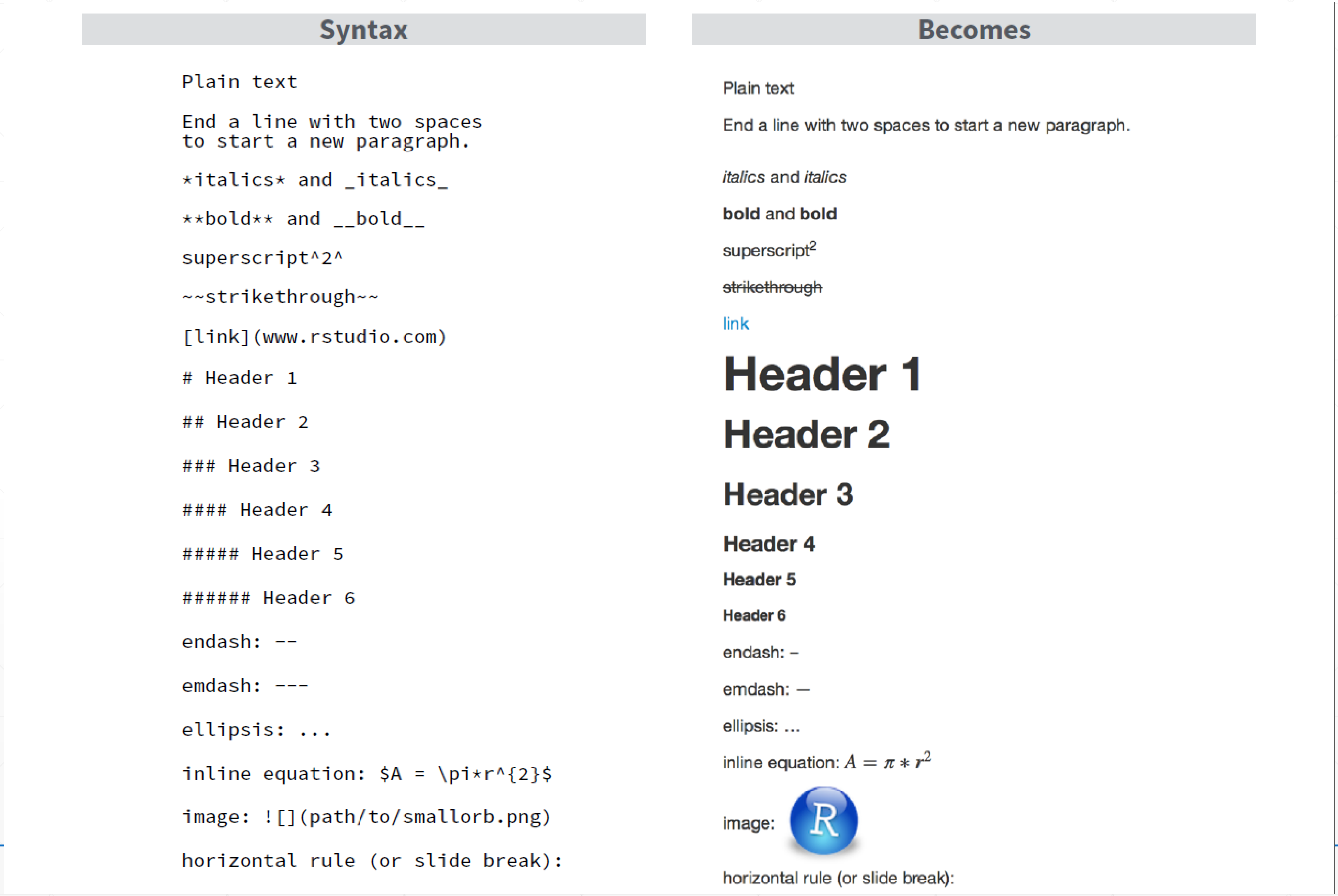
R Markdown

This is an R Markdown document.
Markdown is a simple formatting
syntax for authoring HTML, PDF,
and MS Word documents.

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

For more details on using R Markdown
see <http://rmarkdown.rstudio.com>.
```

```
> library(rmarkdown)
> render("report.Rmd", output_file = "report.html")
```

R Markdown

RStudio

- R Markdown

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
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
```

For more details on using R Markdown see http://rmarkdown.rstudio.com.

report.Rmd	398 B	Feb 26, 2016, 3:36 PM
report.html	581.3 KB	Feb 26, 2016, 3:36 PM

Synch publish button to accounts at
- rpubs.com,
- shinyapps.io
- RStudio Connect

Reload document

Find in document

File path to output document

6 Examine build log in R Markdown console

7 Use output file that is saved alongside .Rmd

.Rmd structure

YAML Header
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).
- At start of file
- Between lines of - - -

Text
Narration formatted with markdown, mixed with:

Code chunks
Chunks of embedded code. Each chunk:
- Begins with ```{r}
- ends with ```

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the **working directory**

render()

Use rmarkdown::**render()** to render/knit at cmd line. Important args:

input - file to render

output_format

output_options - List of render options (as in YAML)

output_file

output_dir

params - list of params to use

envir - environment to evaluate code chunks in

encoding - of input file

Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

1 **Add runtime: shiny** to the YAML header.

2 Call Shiny **input** functions to embed input objects.

3 Call Shiny **render** functions to embed reactive output.

4 Render with rmarkdown::**run** or click **Run Document in RStudio IDE**

```
---
output: html_document
runtime: shiny
---

```{r, echo = FALSE}
numericInput("n",
  "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```
```



How many cars?

5

	speed	dist
1	4.00	2.00
2	4.00	10.00
3	7.00	4.00
4	7.00	22.00
5	8.00	16.00

Embed a complete app into your document with shiny::**shinyAppDir()**

* *Your report will rendered as a Shiny app, which means you must choose an html output format, like **html_document**, and serve it with an active R Session.*

Embed code with knitr syntax

Inline code

Insert with `` `r <code>` ``. Results appear as text without code.

```
Built with
`r getRversion()`
```

Built with 3.2.3

Code chunks

One or more lines surrounded with ```{r} and ```. Place chunk options within curly braces, after r. Insert with

```
```{r echo=TRUE}
getRversion()
```
```

```
getRversion()
```

```
## [1] '3.2.3'
```

Global options

Set with knitr::**opts_chunk$set()**, e.g.

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

Important chunk options

cache - cache results for future knits (default = FALSE)

cache.path - directory to save cached results in (default = "cache/")

child - file(s) to knit and then include (default = NULL)

collapse - collapse all output into single block (default = FALSE)

comment - prefix for each line of results (default = '##')

dependson - chunk dependencies for caching (default = NULL)

echo - Display code in output document (default = TRUE)

engine - code language used in chunk (default = 'R')

error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)

eval - Run code in chunk (default = TRUE)

fig.align - 'left', 'right', or 'center' (default = 'default')

fig.cap - figure caption as character string (default = NULL)

fig.height, fig.width - Dimensions of plots in inches

highlight - highlight source code (default = TRUE)

include - Include chunk in doc after running (default = TRUE)

message - display code messages in document (default = TRUE)

results (default = 'markup')
'asis' - passthrough results
'hide' - do not display results
'hold' - put all results below all code

tidy - tidy code for display (default = FALSE)

warning - display code warnings in document (default = TRUE)

Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts

Parameters

Parameterize your documents to reuse with different inputs (e.g., data sets, values, etc.)

1 **Add parameters**
Create and set parameters in the header as sub-values of **params**

```
---
params:
  n: 100
  d: !r Sys.Date()
---
```

Indent 2 spaces

2 **Call parameters**
Call parameter values in code as **params$<name>**

```
Today's date
is `r params$d`
```

3 **Set parameters**
Set values wth **Knit with parameters** or the params argument of render():



render("doc.Rmd",
  params = list(n = 1, d = as.Date("2015-01-01"))

More cheat sheets at http://www.rstudio.com/resources/cheatsheets/ Learn more at **rmarkdown.rstudio.com** • RStudio IDE 0.99.879 • Updated: 02/16
```

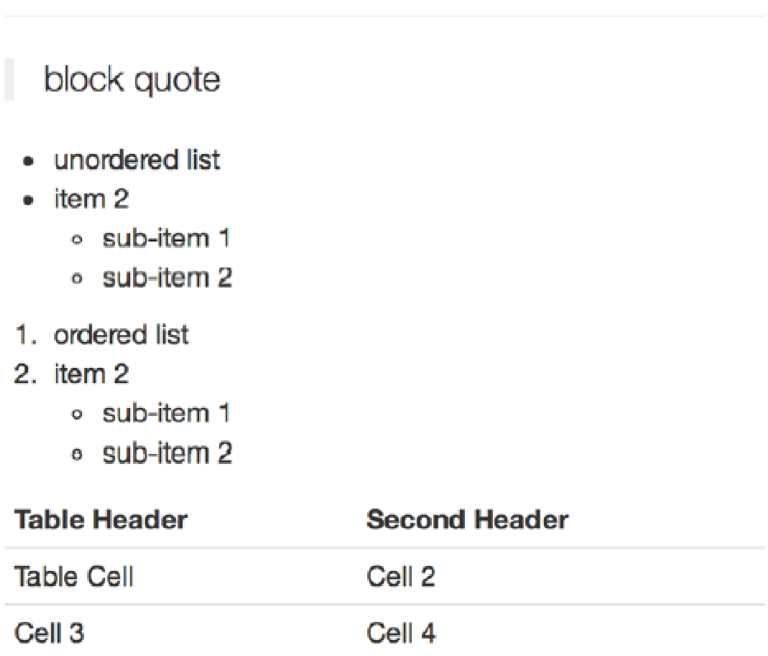


| Syntax | Becomes |
|---|---|
| `Plain text` | Plain text |
| `End a line with two spaces`<br>`to start a new paragraph.` | End a line with two spaces to start a new paragraph. |
| `*italics* and _italics_` | *italics* and *italics* |
| `**bold** and __bold__` | **bold** and **bold** |
| `superscript^2^` | superscript$^2$ |
| `~~strikethrough~~` | ~~strikethrough~~ |
| `[link](www.rstudio.com)` | link |
| `# Header 1` | **Header 1** |
| `## Header 2` | **Header 2** |
| `### Header 3` | **Header 3** |
| `#### Header 4` | **Header 4** |
| `##### Header 5` | **Header 5** |
| `###### Header 6` | **Header 6** |
| `endash: --` | endash: – |
| `emdash: ---` | emdash: — |
| `ellipsis: ...` | ellipsis: ... |
| `inline equation: $A = \pi*r^{2}$` | inline equation: $A = \pi * r^2$ |
| `image: ![](path/to/smallorb.png)` | image: |
| `horizontal rule (or slide break):` | horizontal rule (or slide break): |

## Syntax

```
***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header  | Second Header
------------- | -------------
Table Cell    | Cell 2
Cell 3        | Cell 4
```

Becomes

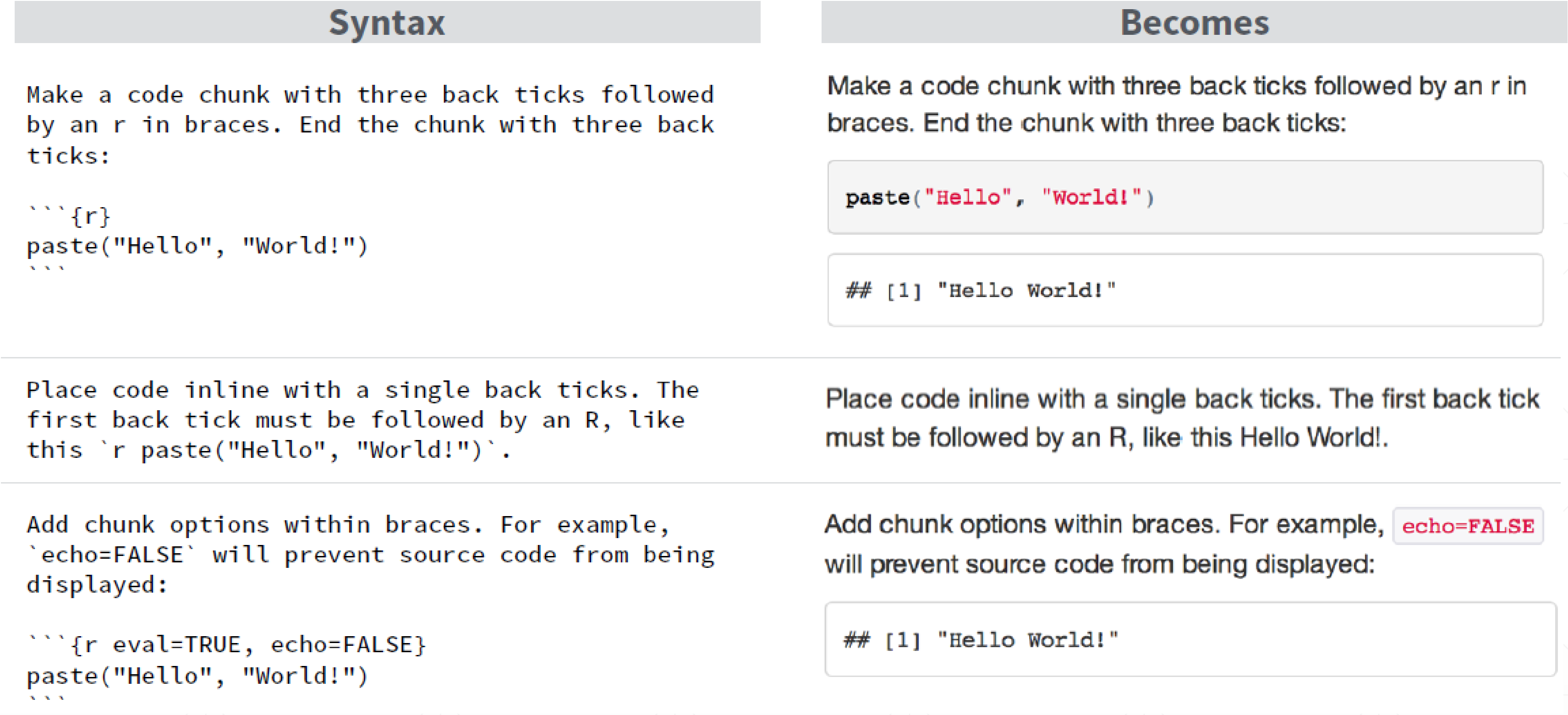


> block quote

- unordered list
- item 2
    - sub-item 1
    - sub-item 2

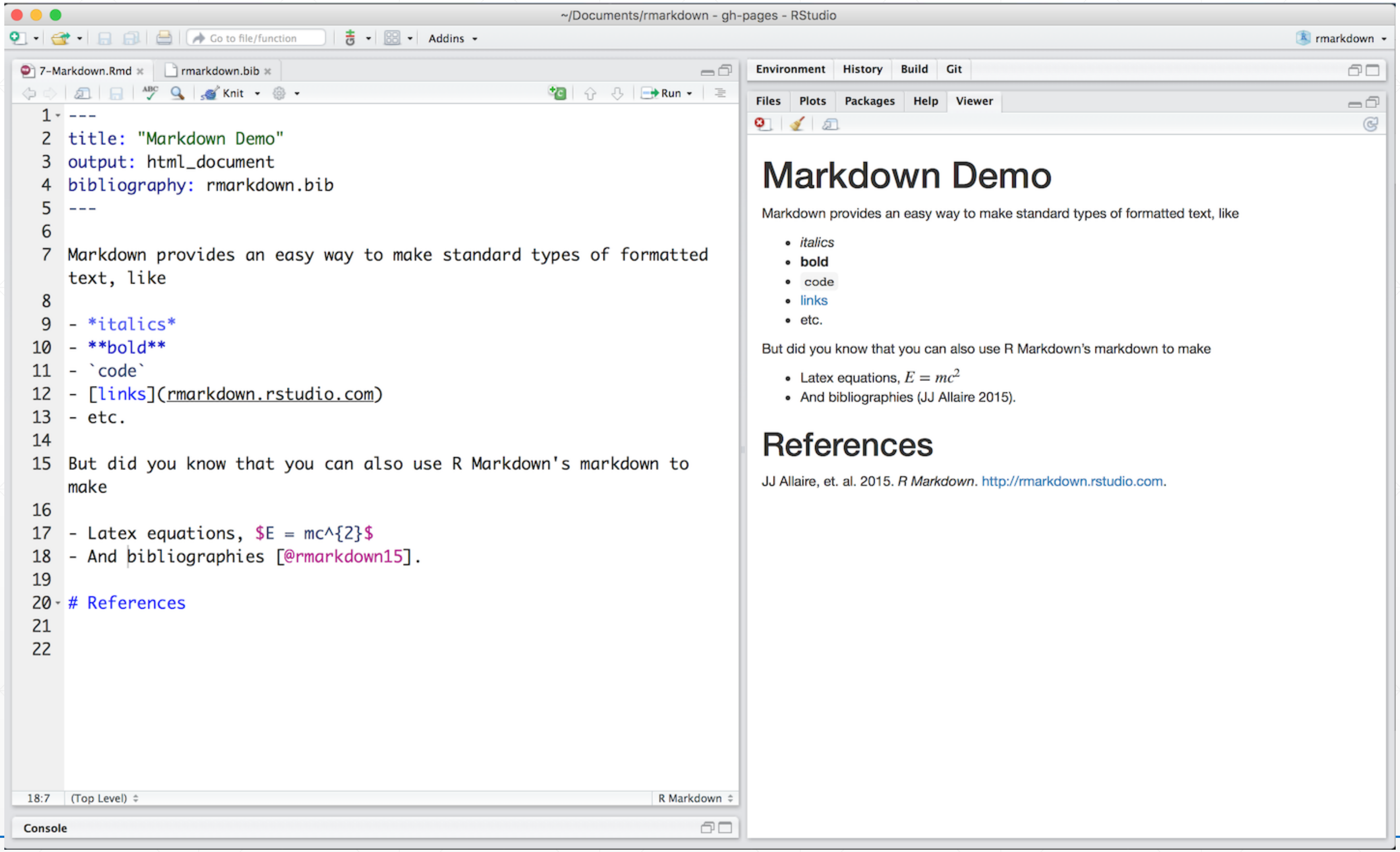
1. ordered list
2. item 2
    - sub-item 1
    - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4



<table>
<tr><th>Syntax</th><th>Becomes</th></tr>
<tr><td>Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:
<pre>
```{r}
paste("Hello", "World!")
```
</pre></td><td>Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:
<pre>
paste("Hello", "World!")
</pre>
<pre>
## [1] "Hello World!"
</pre></td></tr>
<tr><td>Place code inline with a single back ticks. The first back tick must be followed by an R, like this `r paste("Hello", "World!")`.</td><td>Place code inline with a single back ticks. The first back tick must be followed by an R, like this Hello World!.</td></tr>
<tr><td>Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:
<pre>
```{r eval=TRUE, echo=FALSE}
paste("Hello", "World!")
```
</pre></td><td>Add chunk options within braces. For example, <code>echo=FALSE</code> will prevent source code from being displayed:
<pre>
## [1] "Hello World!"
</pre></td></tr>
</table>

Markdown Basics



```
---
title: "Markdown Demo"
output: html_document
bibliography: rmarkdown.bib
---

Markdown provides an easy way to make standard types of formatted text, like

- *italics*
- **bold**
- `code`
- [links](rmarkdown.rstudio.com)
- etc.

But did you know that you can also use R Markdown's markdown to make

- Latex equations, $E = mc^{2}$
- And bibliographies [@rmarkdown15].

# References


```

Markdown Demo

Markdown provides an easy way to make standard types of formatted text, like

- *italics*
- **bold**
- `code`
- links
- etc.

But did you know that you can also use R Markdown's markdown to make

- Latex equations, $E = mc^2$
- And bibliographies (JJ Allaire 2015).

References

JJ Allaire, et. al. 2015. *R Markdown*. http://rmarkdown.rstudio.com.

http://rmarkdown.rstudio.com/lesson-8.html

knitr package

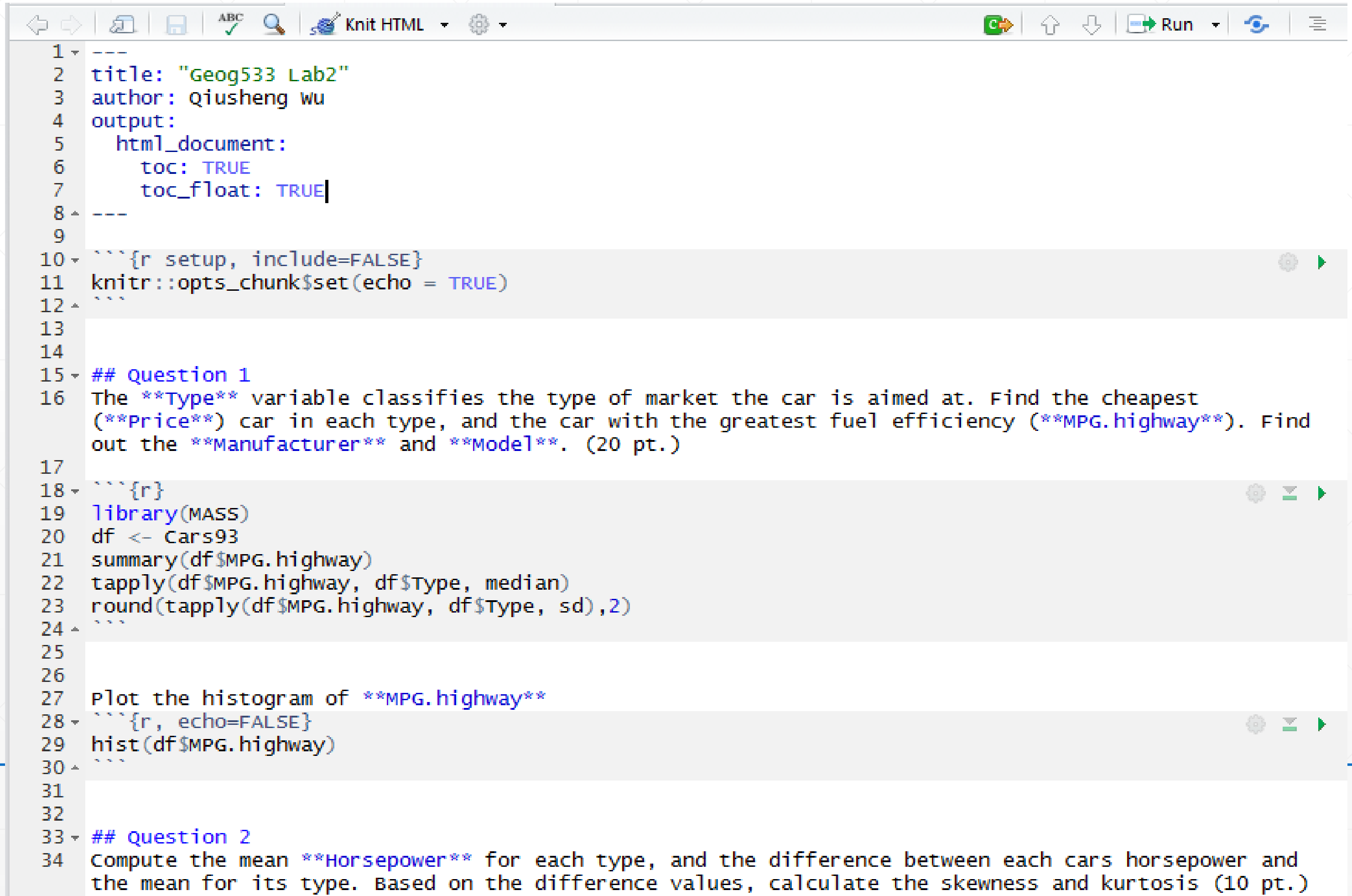
- **if (!require("knitr")) install.packages("knitr")**
- library(knitr)
- knit("your-document.Rmd") # compiles a document

Chunk options

- echo=FALSE Don't include the code
- results="hide" Don't include the output
- include=FALSE Don't show code or output
- eval=FALSE Don't evaluate the code at all
- warning=FALSE Don't show R warnings
- message=FALSE Don't show R messages
- fig.width=# Width of figure
- fig.height=# Height of figure
- fig.path="Figs/" Path for figure files

YAML header

- output: html_document
- output: html_notebook



~~~
---
title: "Geog533 Lab2"
author: Qiusheng Wu
output:
  html_document:
    toc: TRUE
    toc_float: TRUE
---

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

Question 1
The **Type** variable classifies the type of market the car is aimed at. Find the cheapest
(**Price**) car in each type, and the car with the greatest fuel efficiency (**MPG.highway**). Find
out the **Manufacturer** and **Model**. (20 pt.)

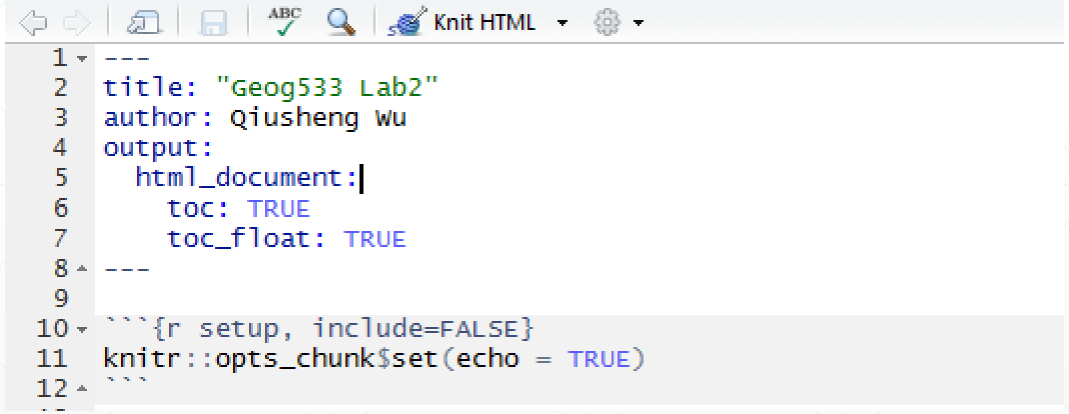
```{r}
library(MASS)
df <- Cars93
summary(df$MPG.highway)
tapply(df$MPG.highway, df$Type, median)
round(tapply(df$MPG.highway, df$Type, sd),2)
```

Plot the histogram of **MPG.highway**
```{r, echo=FALSE}
hist(df$MPG.highway)
```

Question 2
Compute the mean **Horsepower** for each type, and the difference between each cars horsepower and
the mean for its type. Based on the difference values, calculate the skewness and kurtosis (10 pt.)
```
```

Global chunk options

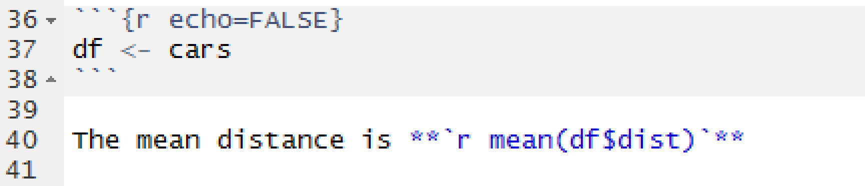
- Use global chunk options rather than repeat the same options over and over.
- You can override the global values in specific chunks.
- opts_chunk$set



~~~
---
title: "Geog533 Lab2"
author: Qiusheng Wu
output:
  html_document:
    toc: TRUE
    toc_float: TRUE
---

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

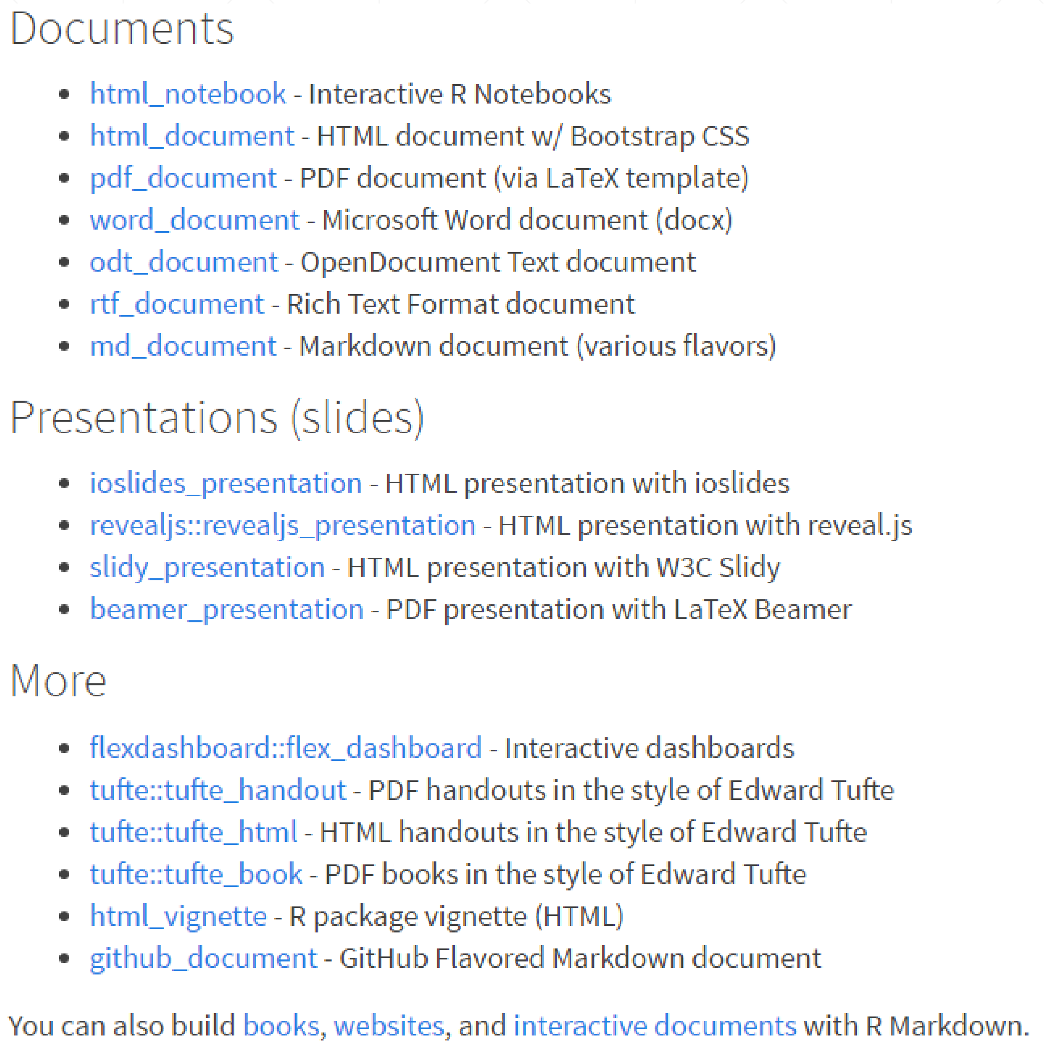
In-line code

- Each bit of in-line code needs to be within one line; they can't span across lines.
- I'll often precede a paragraph with a code chunk with include=FALSE, defining various variables, to simplify the in-line code.
- Never hard-code a result or summary statistic again!



~~~
```{r echo=FALSE}
df <- cars
```

The mean distance is **`r mean(df$dist)`**

~~~

Output Formats



Documents

- html_notebook - Interactive R Notebooks
- html_document - HTML document w/ Bootstrap CSS
- pdf_document - PDF document (via LaTeX template)
- word_document - Microsoft Word document (docx)
- odt_document - OpenDocument Text document
- rtf_document - Rich Text Format document
- md_document - Markdown document (various flavors)

Presentations (slides)

- ioslides_presentation - HTML presentation with ioslides
- revealjs::revealjs_presentation - HTML presentation with reveal.js
- slidy_presentation - HTML presentation with W3C Slidy
- beamer_presentation - PDF presentation with LaTeX Beamer

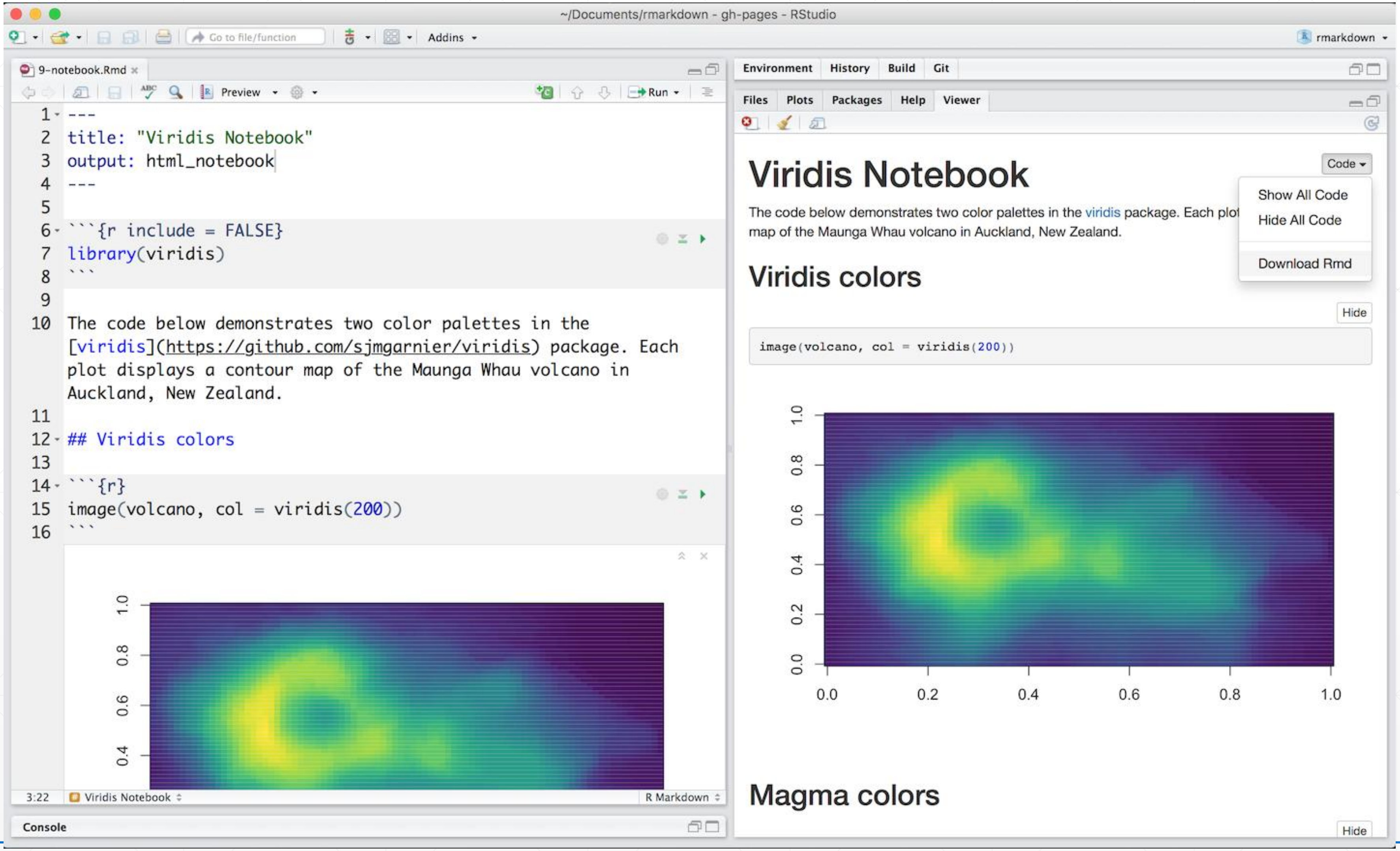
More

- flexdashboard::flex_dashboard - Interactive dashboards
- tufte::tufte_handout - PDF handouts in the style of Edward Tufte
- tufte::tufte_html - HTML handouts in the style of Edward Tufte
- tufte::tufte_book - PDF books in the style of Edward Tufte
- html_vignette - R package vignette (HTML)
- github_document - GitHub Flavored Markdown document

You can also build books, websites, and interactive documents with R Markdown.

http://rmarkdown.rstudio.com/lesson-9.html

Notebooks



~~~
---
title: "Viridis Notebook"
output: html_notebook
---

```{r include = FALSE}
library(viridis)
```

The code below demonstrates two color palettes in the [viridis](https://github.com/sjmgarnier/viridis) package. Each plot displays a contour map of the Maunga Whau volcano in Auckland, New Zealand.

## Viridis colors

```{r}
image(volcano, col = viridis(200))
```
~~~

Viridis Notebook

The code below demonstrates two color palettes in the viridis package. Each plot map of the Maunga Whau volcano in Auckland, New Zealand.

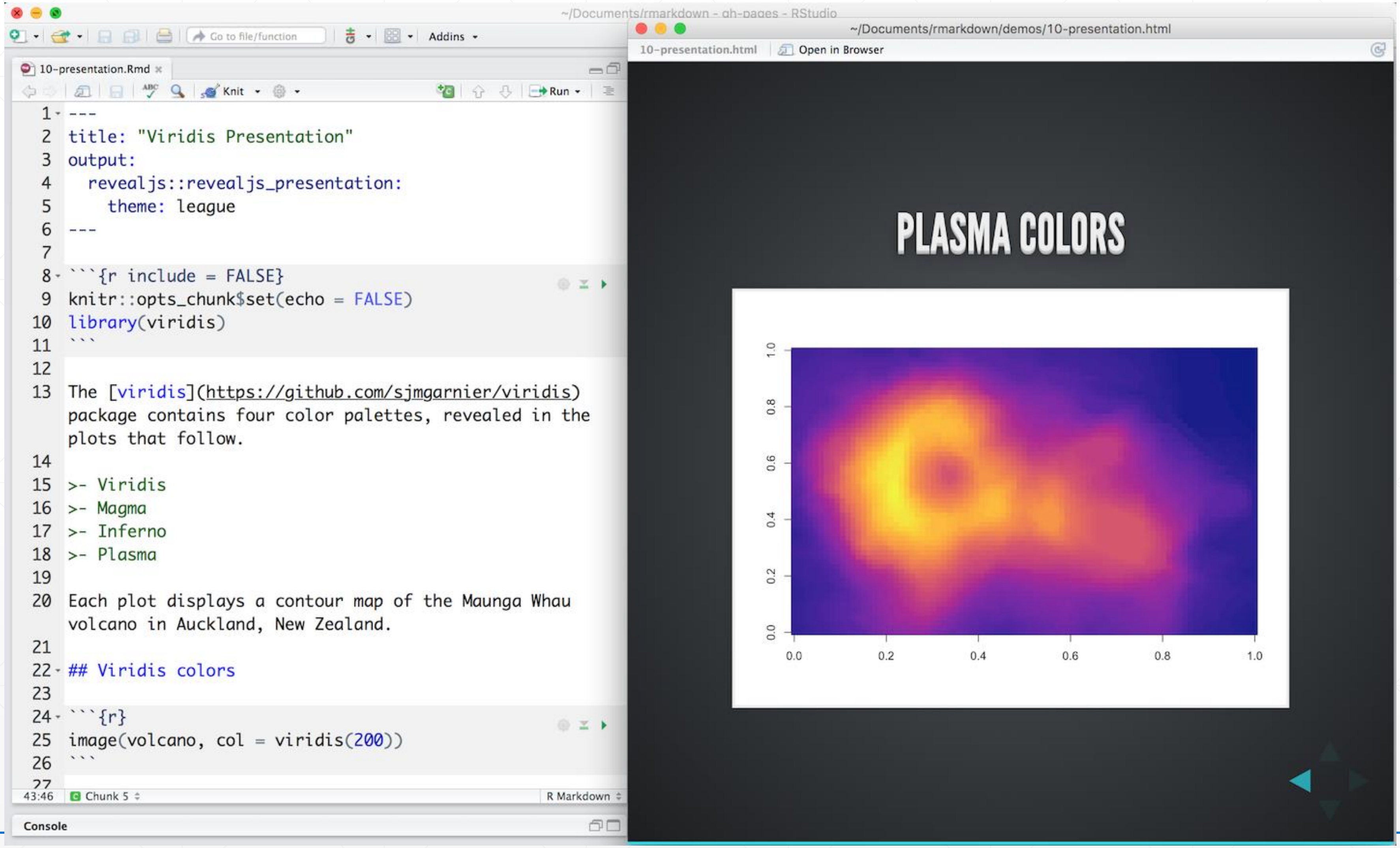
Viridis colors

```
image(volcano, col = viridis(200))
```

Magma colors

http://rmarkdown.rstudio.com/lesson-10.html

Slide Presentations



~~~
---
title: "Viridis Presentation"
output:
  revealjs::revealjs_presentation:
    theme: league
---

```{r include = FALSE}
knitr::opts_chunk$set(echo = FALSE)
library(viridis)
```

The [viridis](https://github.com/sjmgarnier/viridis)
package contains four color palettes, revealed in the
plots that follow.

>- Viridis
>- Magma
>- Inferno
>- Plasma

Each plot displays a contour map of the Maunga Whau
volcano in Auckland, New Zealand.

## Viridis colors

```{r}
image(volcano, col = viridis(200))
```
~~~

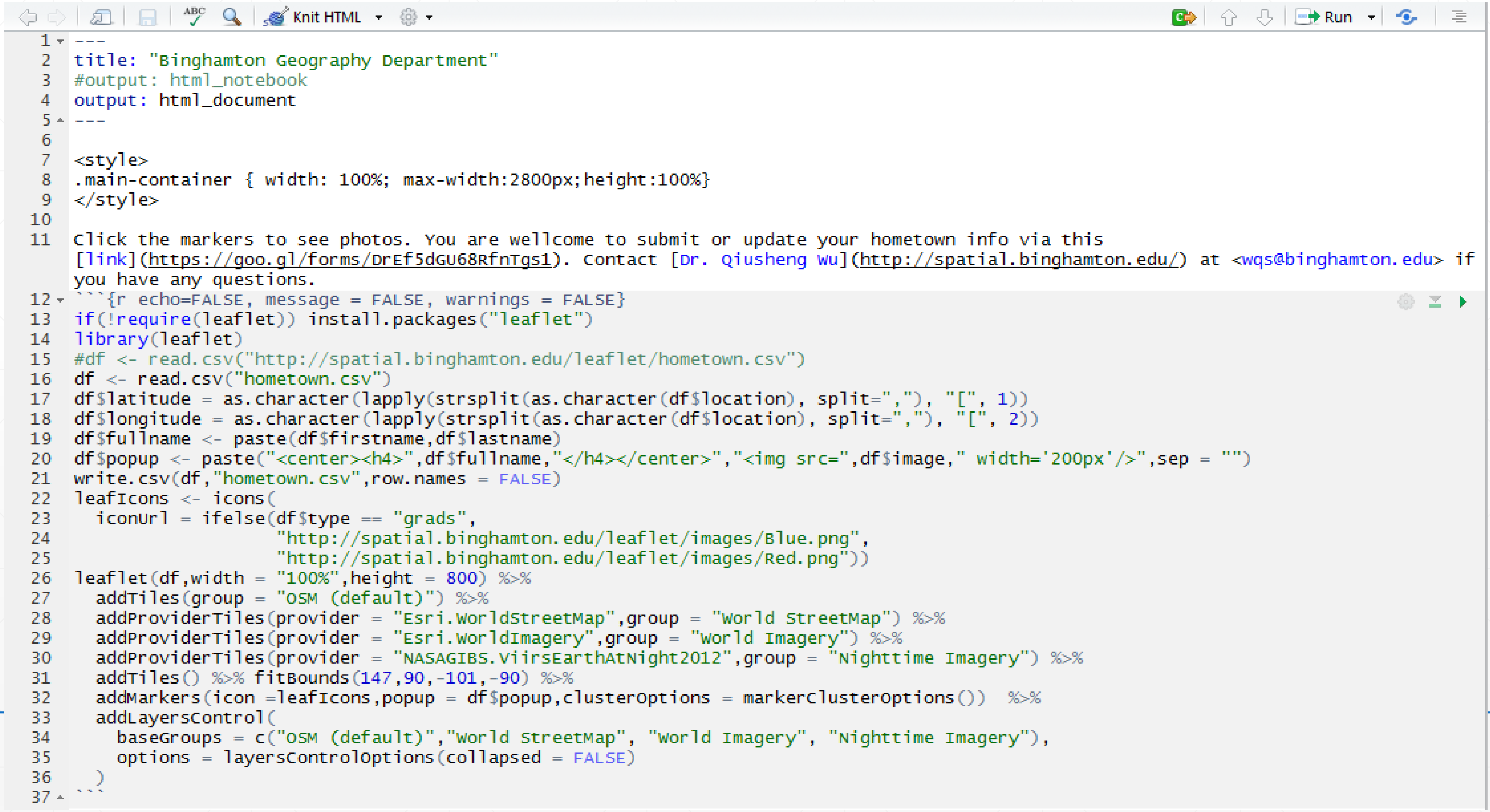
http://rmarkdown.rstudio.com/lesson-11.html

Books

- Bookdown: Authoring Books with R Markdown
  - https://bookdown.org/yihui/bookdown/
- Authoring Books with R Markdown
  - https://www.rstudio.com/resources/webinars/introducing-bookdown/

Websites

- http://spatial.binghamton.edu/leaflet/hometown.nb.html



~~~
---
title: "Binghamton Geography Department"
#output: html_notebook
output: html_document
---

<style>
.main-container { width: 100%;  max-width:2800px;height:100%}
</style>

Click the markers to see photos. You are wellcome to submit or update your hometown info via this
[link](https://goo.gl/forms/DrEf5dGU68RfnTgs1). Contact [Dr. Qiusheng Wu](http://spatial.binghamton.edu/) at <wqs@binghamton.edu> if
you have any questions.
```{r echo=FALSE, message = FALSE, warnings = FALSE}
if(!require(leaflet)) install.packages("leaflet")
library(leaflet)
#df <- read.csv("http://spatial.binghamton.edu/leaflet/hometown.csv")
df <- read.csv("hometown.csv")
df$latitude = as.character(lapply(strsplit(as.character(df$location), split=","), "[", 1))
df$longitude = as.character(lapply(strsplit(as.character(df$location), split=","), "[", 2))
df$fullname <- paste(df$firstname,df$lastname)
df$popup <- paste("<center><h4>",df$fullname,"</h4></center>","<img src=",df$image," width='200px'/>",sep = "")
write.csv(df,"hometown.csv",row.names = FALSE)
leafIcons <- icons(
  iconUrl = ifelse(df$type == "grads",
                   "http://spatial.binghamton.edu/leaflet/images/Blue.png",
                   "http://spatial.binghamton.edu/leaflet/images/Red.png"))
leaflet(df,width = "100%",height = 800) %>%
  addTiles(group = "OSM (default)") %>%
  addProviderTiles(provider = "Esri.WorldStreetMap",group = "World StreetMap") %>%
  addProviderTiles(provider = "Esri.WorldImagery",group = "World Imagery") %>%
  addProviderTiles(provider = "NASAGIBS.ViirsEarthAtNight2012",group = "Nighttime Imagery") %>%
  addTiles() %>% fitBounds(147,90,-101,-90) %>%
  addMarkers(icon =leafIcons,popup = df$popup,clusterOptions = markerClusterOptions())  %>%
  addLayersControl(
    baseGroups = c("OSM (default)","World StreetMap", "World Imagery", "Nighttime Imagery"),
    options = layersControlOptions(collapsed = FALSE)
  )
```
~~~