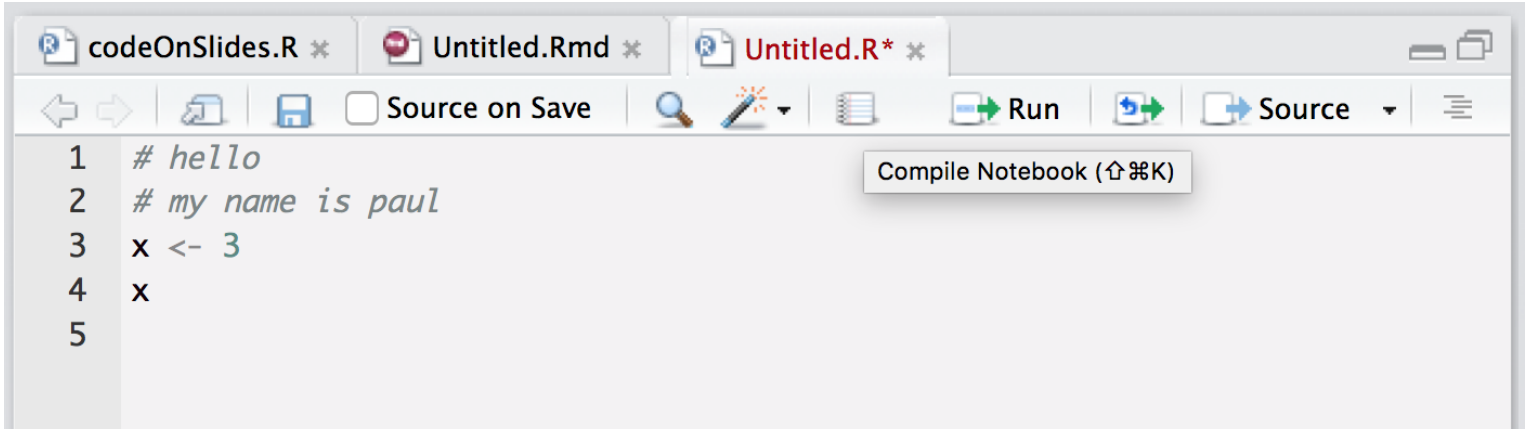
Section 3

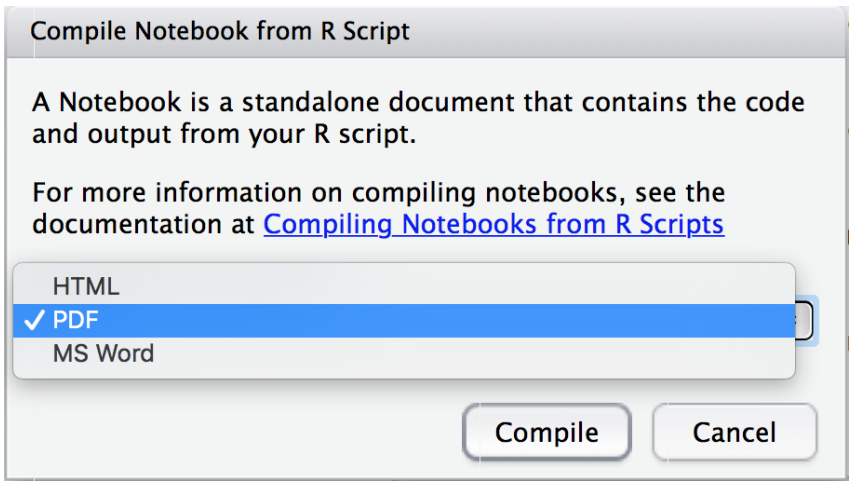
RMarkdown

Technical Reporting & Presentation Tools

- There are many ways to generate pdf- and web-based technical reports and presentations
  1. LaTeX(pronounced *lay-tech*)
  2. Lyx
  3. Markdown (RMarkdown and other variations)
  4. ...
- The (arguably) default, most generic and most flexible technical report/presentation tool is LaTeX
- When dealing with exclusively with R code and R output, RMarkdown is a versatile and easy way to embed code and graphical output in a report or presentation
- Alternate R-based packages exist for report generation

The Simplest RMarkdown Execution





The Simplest RMarkdown Execution

- Rmd source file
- pdf output

Untitled.R

Paul

Tue Jul 5 10:51:43 2016

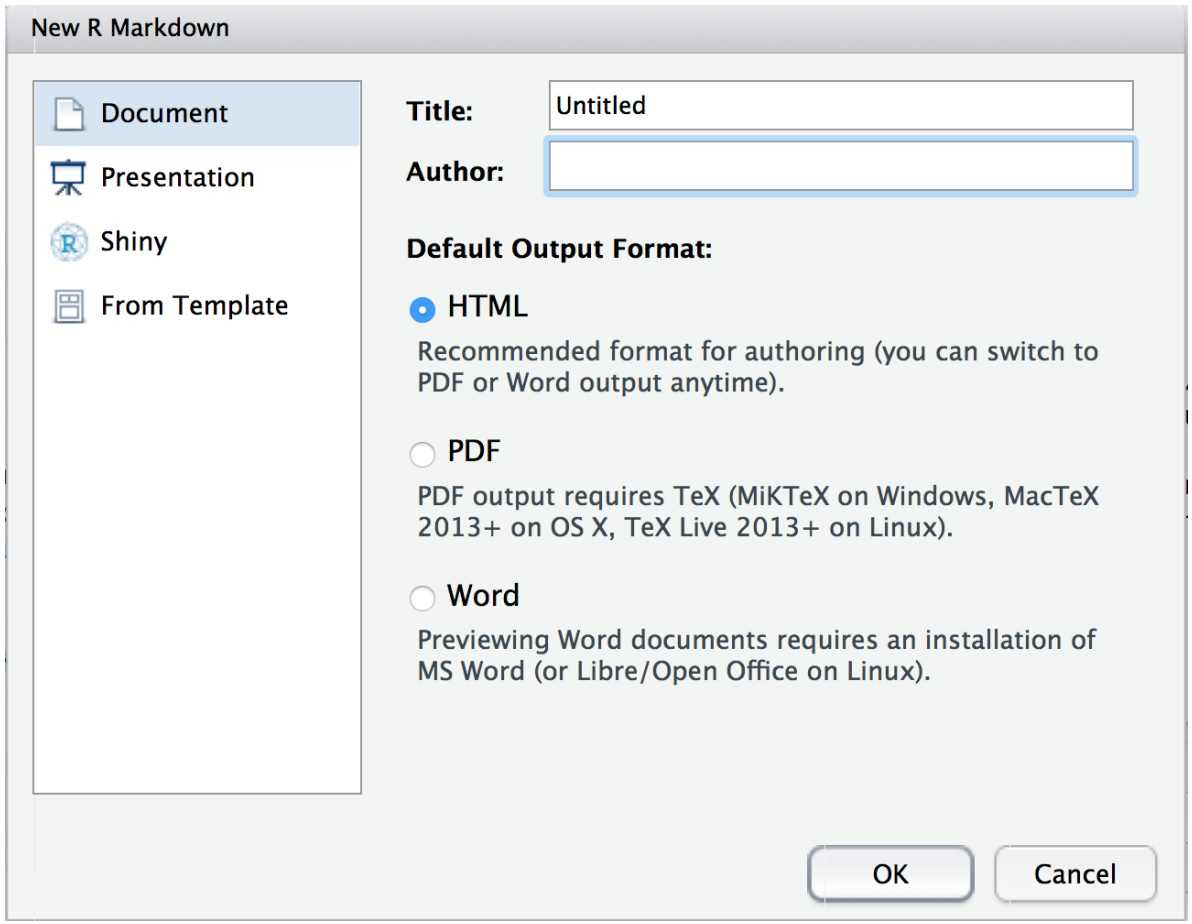
```
# hello
# my name is paul
x <- 3
x
```

```
## [1] 3
```

A Smarter RMarkdown Document

Create a new RMarkdown file

`File` ⇒ `New File` ⇒ `RMarkdown...`



MacTex is Required for RMarkdown pdf Output

RMarkdown: The Header

- The header begins and ends with three dashes ---
- There are many header options, we will examine a few basic options

```
---
title: "My Title"
author: "Paul Intrevado"
date: "July 5, 2017"
output: pdf_document
---
```

RMarkdown: Body Text

- How do you write plain text?

```
Just like this
```

- How do you comment out a line of text?

```
[//]: # comment goes here
```

- You can create sections / section headers using the "#" symbol

```
# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6
```

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

RMarkdown: Body Text [CONT'D]

- Inline equations are similar (identical?) to LaTeXsyntax
  - $e^{i\pi} - 1 = 0$ is written `$e^{i \pi} - 1 = 0$`

RMarkdown: Body Text [CONT'D]

- Inline equations are similar (identical?) to LaTeXsyntax
  - $e^{i\pi} - 1 = 0$ is written `$e^{i \pi} - 1 = 0$`
- Bold and italicized statements are written using `**myBoldText**` and `*myItalicizedText*`

RMarkdown: Body Text [CONT'D]

- Inline equations are similar (identical?) to LaTeXsyntax
  - $e^{i\pi} - 1 = 0$ is written `$e^{i \pi} - 1 = 0$`
- Bold and italicized statements are written using `**myBoldText**` and `*myItalicizedText*`
- In-line code that is **not** executed can be included in backticks (over tilde)

CODE To assign a value to a variable: `` `myVar <- 1` ``

OUTPUT To assign a value to a variable: `myVar <- 1`

RMarkdown: Body Text [CONT'D]

- Inline equations are similar (identical?) to LaTeXsyntax
  - $e^{i\pi} - 1 = 0$ is written `$e^{i \pi} - 1 = 0$`
- Bold and italicized statements are written using `**myBoldText**` and `*myItalicizedText*`
- In-line code that is **not** executed can be included in backticks (over tilde)

CODE To assign a value to a variable: `` `myVar <- 1` ``
OUTPUT To assign a value to a variable: `myVar <- 1`

- In-line code that **is** executed can be included as `` `r <insert code here>` ``

CODE The product of 2 + 3 is `` `r sum(c(2, 3))` ``
OUTPUT The product of 2 + 3 is 5

RMarkdown: Code Chunks

- At the heart of RMarkdown are the code chunks, which allow for great flexibility when including raw code as well as results, from simple computations to complex graphs and analyses

```
```{r <sectionTitle>, <options>}
<include code here>
```
```

- Use ctrl + option + I as a shortcut to include code chunk
- <sectionTitle> is the *unique* name of the code chunk
- <options> are a sequence of options separated by commas

n.b. all labels and code chunk options must be on the same line

RMarkdown: Selected Code Chunk Options

- `eval = F`: prevents code from being evaluated
- `echo = F`: prevents code, but not results, from appearing in final output document
- `include = F`: runs code but doesn't show code **or** results in final output document
- `message = F / warning = F`: prevents messages or warning from appearing in final output document
- `cache = T`: will store the results of a code chunk in cache, so subsequent knits of the document don't need to re-execute computationally expensive code chunks (use only for static data)

RMarkdown: Selected Global Chunk Options

- To set global options for all code chunks, include the following code chunk after the header

````
```{r <sectionTitle>, include = FALSE}
knitr::opts_chunk$set(<options>)
```
````

- `include = FALSE` is included so that the code is evaluated but the code chunk nor the results are printed to the output document

E.g. To have all code chunks in an RMarkdown document be suppressed, include `include = FALSE` in the output document

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

Including Non-R Code in Code Chunks

- RMarkdown is not limited to R code
- knitr can run code from a variety of other languages including but not limited to Python, Ruby and Bash
- To include non-R code in a code chunk, set the engine code chunk to tell knitr which language you are using

E.g. To include Python code

````
```{r engine = 'python'}
print "Hello World"
```
````

- Additional non-R programming language interpretation is available using the highlighter package

Tables in RMarkdown

```
dplyr::slice(mtcars, 1:5)
```

```
## # A tibble: 5 x 11
##     mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21.0     6   160   110  3.90 2.620 16.46     0     1     4     4
## 2  21.0     6   160   110  3.90 2.875 17.02     0     1     4     4
## 3  22.8     4   108    93  3.85 2.320 18.61     1     1     4     1
## 4  21.4     6   258   110  3.08 3.215 19.44     1     0     3     1
## 5  18.7     8   360   175  3.15 3.440 17.02     0     0     3     2
```

```
knitr:: kable(dplyr::slice(mtcars, 1:5))
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

RMarkdown Resources

- Reproducible Research with R and RStudio by Christopher Gandrud
  - This is a less technical, more pragmatic approach to RMarkdown
- Dynamic Documents with R and *knitr* by Yihui Xie
  - A more technical, detailed and rigorous treatment of RMarkdown and *knitr*

RMarkdown Resources [July 2016]

- RMarkdown Cheat Sheet

  http://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

- RMarkdown Reference Guide

  http://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf

- RMarkdown PDF Documents: Overview

  http://rmarkdown.rstudio.com/pdf_document_format.html

IMPORTANT: Style Guide

Now that Introduction to RMarkdown is complete, be sure to **thoroughly read** the Style Guide (Chapter 5), in Hadley Wickham's `Advanced R`. You will be held to that standard in your coding style moving forward.

http://adv-r.had.co.nz/Style.html

LAB

RMarkdown: Titanic Case Study

Paul Intrevado

July 12, 2018

This is my first R Markdown document for **MSDS 593**. I am required to submit all **MSDS 593** homework in RMarkdown.

Firstly, I am going to generate an `html` document as output, set in the YAML header section of this document. Secondly, I am **NOT** hardcoding the date, but rather using a function which will automatically print the current date on the day the document is knit.

Now, I am going to import a dataset about passengers from the Titanic, using the following line of code:

```
read.csv("~/Desktop/titanic.csv")
```

This fails for a few reasons, namely, I read in the file and stored it no where. So I wasted my time waiting for R to read in the file, and then when it