Advanced R Part I

Justin Post

August 15-16, 2019

Course Schedule

Daily agenda:

- 9:30-10:40 Session
- 10-minute break
- 10:50-12:00 Session
- 12:00-1:15 Lunch
- 1:15-2:25 Session
- 10-minute break
- 2:35-3:45 Session

What do we want to be able to do?

- Communicate findings effectively
- Document findings
- Make process reproducible
- Share process

Where do we start?

- R Markdown Basics
  - Code Chunks
  - Images/Equations/Misc.
- R Markdown Options
  - Documents: PDF, HTML
  - Presentations: Slides
  - Interactive Components
- R Shiny Applications/Presentations

RStudio

Nice functionality for

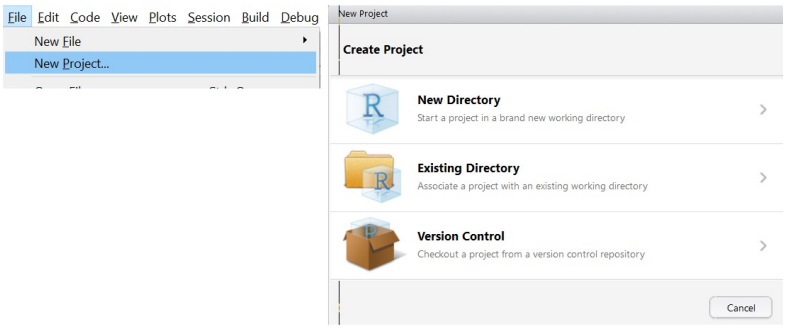
- Markdown
- Shiny
- Databases
- Git/Github

RStudio - Project

- Often have many files associated with each analysis
- Keeping different undertakings separate can be difficult!
- Can use "Project" feature in R Studio
- Provides straightforward way to divide your work into multiple contexts. Each with their own:
  - Working directory
  - Workspace
  - History
  - Source documents

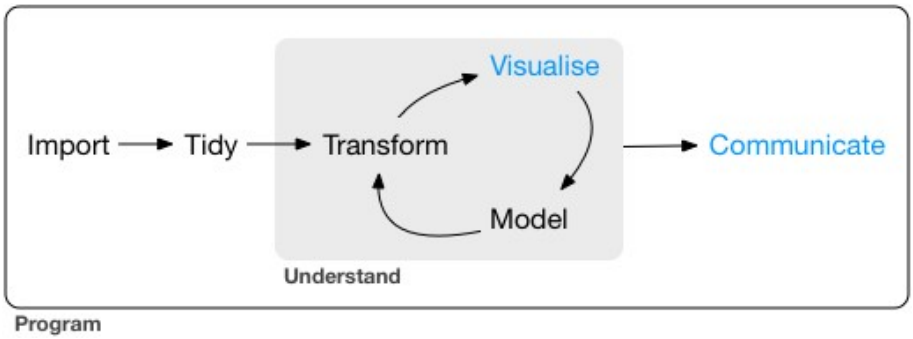
R Studio - Project

- Easy to create!



- Can save workspace, etc. and pick up right where you left off!
- Let's create one for today!
- Place the TextActivityPlainText.Rmd file in the project folder

R Markdown Basics



(From R for Data Science)

R Markdown Basics

- Can read data into R
- Ability to manipulate it
- Likely know best ways to model and visualize it
- Doesn't matter how great your analysis is unless you can explain it to others :)
- Need to communicate results effectively

(Traditional) Basic Use of R

- Usually want to keep code for later use
- Write code in a 'script'
- Save code script
- Send lines of code to console via:
  - "Run" button (runs current line)
  - CTRL+Enter (PC) or Command+Enter (MAC)
  - Highlight section and do above

Using a Notebook Instead

- Communication and reproducibility vital!
- May have heard of JUPYTER notebooks
- R Markdown - built in notebook for R studio
- Intro video

R Markdown Basics

What is Markdown?

- Formatting syntax for authoring HTML, PDF, slide shows, books, and more.
- Digital "Notebook": Program that weaves word processing and code. Example

Designed to be used in three ways (R for Data Science)

- Communicating to decision makers (focus on conclusions not code)
- Collaborating with other data scientists (including future you!)
- As environment to do data science (documents what you did and what you were thinking)
- Examples of markdown documents

R Markdown Basics

Verbage

- Most have heard of HTML (HyperText Mark-up Language)
    - Write plain text that the browser interprets and renders

R Markdown Basics

Verbage

- Most have heard of HTML (HyperText Mark-up Language)
  - Write plain text that the browser interprets and renders
- Markdown is a specific markup language
  - Easier syntax
  - Not as powerful
- Any plain text file with .Rmd extension can be used

R Markdown Basics

R Markdown Basics

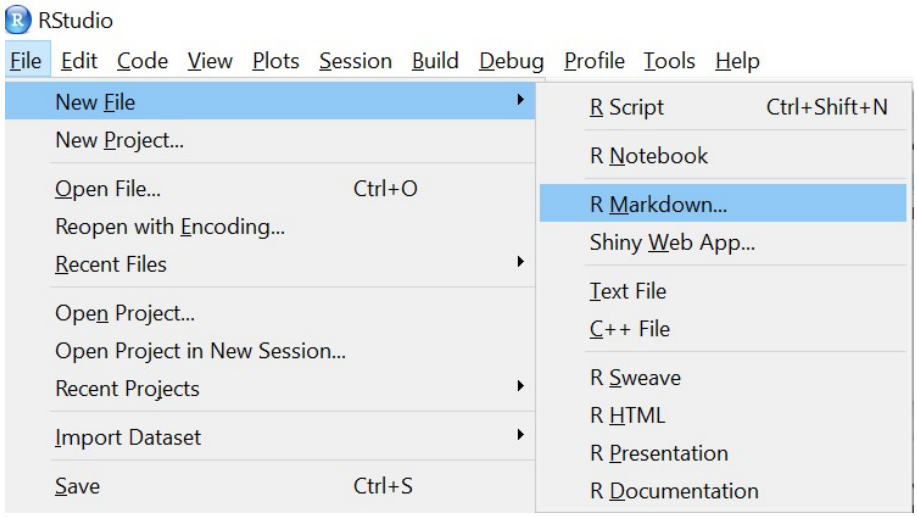
R Markdown file contains three important types of content:

1. (Optional) YAML header surrounded by ---s
2. Chunks of R code surrounded by ```
3. Text mixed with simple text formatting like # heading and *italics*

R Markdown Basics

Creating an R Markdown Document

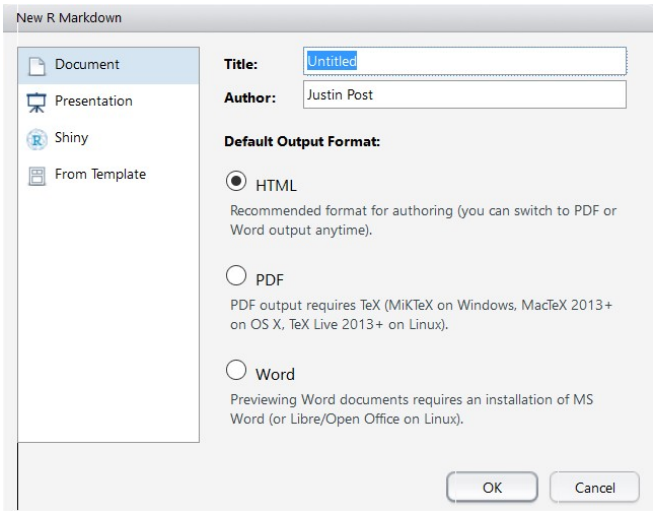
- R Studio makes it easy!



R Markdown Basics

Creating an R Markdown Document

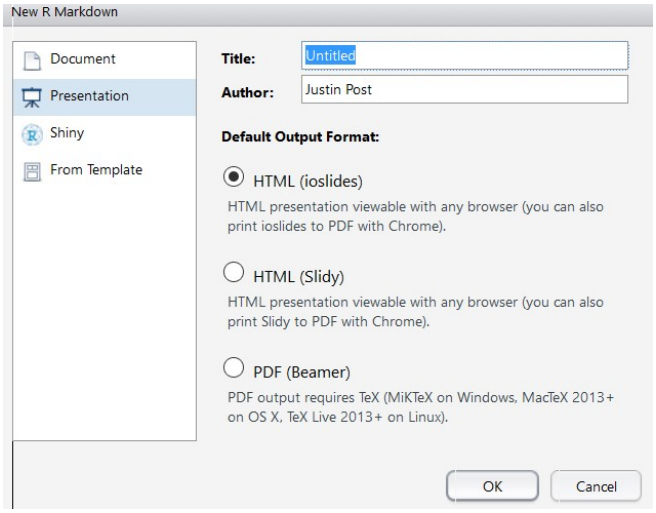
- Commonly used document types can be created



R Markdown Basics

Creating an R Markdown Document

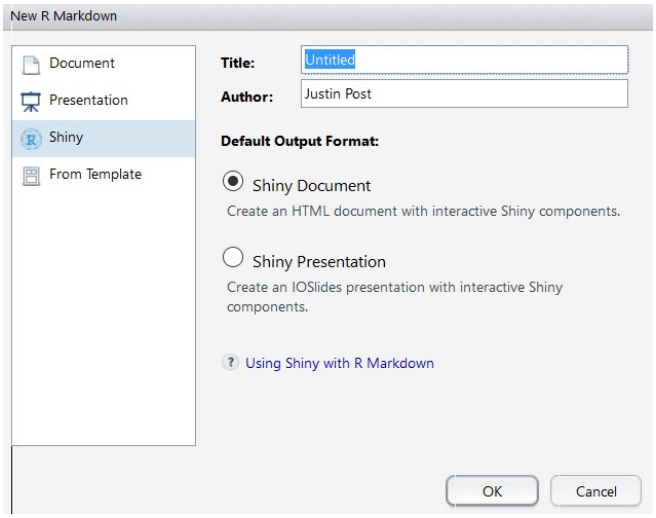
- Slide presentations



R Markdown Basics

Creating an R Markdown Document

- Truly Interactive Documents/Pages (requires R backend)



R Markdown Basics

- Create an HTML Markdown document!

```
---
title: "Untitled"
author: "Justin Post"
date: "August 10, 2017"
output: html_document
---
```

- Top section: YAML header
- Define settings for document
- Author, Title, etc.
- Output type/Options

R Markdown Basics

- Below YAML header: 'r chunk'

````
```{r ggplot,eval=FALSE}
select(iris, Sepal.Width)
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length)) +
geom_point()
```
````

- Start code chunk by typing ```{r} out or with CTRL/CMD + Alt + I
- Code will be executed when document is created
- Can specify options on individual code chunks

R Markdown Basics

- Below code chunk is plain text with markdown sytnax

```
## 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.
```

- When file created, "##" becomes a header, "<…>" a link, and `**Knit**` bold font

R Markdown Basics

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.

Where do we go from here?

- Figure out markdown syntax
- Look at "Notebook" feature
- Check options for code chunks
- Change type of output
- Work with interactivity (shiny)

R Markdown Syntax Can Include…

- Plain text
- End a line with two spaces to start a new paragraph
  - Line breaks are not always added when you return!
  - Two spaces and a return drop marked up text down.
  - Can specify `<br>` as a line break
- `*italics*` *and* `_italics_`
- `**bold**` **and** `__bold__`
- `superscript^2^` becomes superscript^2
- `~~strikethrough~~` becomes ~~strikethrough~~
- Modify text in your .Rmd file and click Knit -> Knit to HTML (at the top)

R Markdown Syntax

- `[link](https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf)` becomes link
- `# Header 1` becomes a large font header
- `## Header 2` becomes a slightly smaller font header
- Goes to 6 headers
- Use of headers can automatically create a Table of Contents!
- Include an image: `![](path/to/file.png)`
- `` `code` `` becomes `code`
- Modify text in your .Rmd file and click Knit -> Knit to HTML (at the top)

R Markdown Syntax

- Can do lists: be sure to end each line with two spaces!
- Indent sub lists two spaces (I often do four for both)

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

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

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

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

R Markdown Syntax

- Can customize things if you know HTML (a little)

```
<div style = "float: left; width: 50%">

* unordered list

</div>

<div style = "float: right; width: 50%">
```

- unordered list

```
</div>
```

R Markdown Syntax

- Can include nice tables

```
Table Header  | Second Header | Col 3
------------- | ------------- | -----------
Table Cell    | Cell (1, 2)   | Cell (1, 3)
Cell (2, 1)   | Cell (2, 2)   | Cell (2, 3)
```

Table Header	Second Header	Col 3
Table Cell	Cell (1, 2)	Cell (1, 3)
Cell (2, 1)	Cell (2, 2)	Cell (2, 3)

- Add a list and a table in your .Rmd file and click Knit -> Knit to HTML (at the top)

Activity

- **Formatting Text Activity** instructions available on web
- Ask questions! TAs and I will float about the room
- Feel free to ask questions about anything you didn't understand as well!

What do we want to be able to do?

- Communicate findings effectively
- Document findings
- Make process reproducible
- Share process

Where are we at?

- R Markdown Basics
  - **Code Chunks**
  - **Images/Equations/Misc.**
- R Markdown Options
  - Documents: PDF, HTML
  - Presentations: Slides
  - Interactive Components
- R Shiny Applications/Presentations

Code Chunks

We've already seen how to include an R code chunk:

```
```{r ggplot,eval=FALSE}
select(iris, Sepal.Width)
ggplot(iris, aes(x = Sepal.Width, y = Sepal.Length)) +
geom_point()
```
```

- Add chunk via typing
  ```{r}
  code
  ```
- or `Ctrl/Cmd + Alt + I`
- Any R code can go into the chunk
- Code can be added in line: Ex: Iris has 150 observations
- Added by beginning with back-tick `r` and ending with a back-tick: Iris has `r length(iris$Sepal.Length)`

Notebook Functionality

Data science notebook - virtual notebook environment used for literate programming

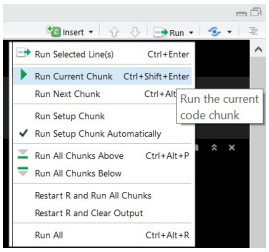
- Pairs the functionality of word processing software with a programming language
- R Markdown brings together the console and the script editor too!
- Rendered markdown document captures R code and process

NHL Example html, NHL Example Rmd

Notebook Functionality

Within a chunk:

- Execute code with `Cmd/Ctrl + Shift + Enter` or with "Run"



- Results show up in editor!

Notebook Functionality

- Allows for quick iteration within a chunk: editing and re-executing - when you are happy, you move on and start a new chunk.
- Can run all code chunks with `Ctrl/Cmd + Alt + R`
- Can develop code and record your thoughts - similar to classic lab notebook in the physical sciences
- Go back to markdown template document, execute code chunk in-line and all code chunks. Add an inline code piece.

Back to Code Chunks

- Many options depending on chunk purpose!
- Can hide/show code with `echo = FALSE/TRUE`
- Can choose if code is evaluated with `eval = TRUE/FALSE`
- `Include = FALSE` is equivalent to `echo = FALSE, eval = TRUE`
- `message = TRUE/FALSE` and `warning = TRUE/FALSE` can turn on/off displaying messages/warnings
- `error = TRUE` allows file to be created with code that has an error

Back to Code Chunks

- Options useful for set-up code (usually first chunk after YAML header)
- Can set global options for all chunks
- Allows for easy change of audience!

Back to Code Chunks

- Options useful for set-up code (usually first chunk after YAML header)
- Can set global options for all chunks
- Allows for easy change of audience!

For a higher up

```
opts_chunk$set(echo = FALSE, eval = TRUE, warning = FALSE)
```

For a collaborator

```
opts_chunk$set(echo = TRUE, eval = TRUE, warning = FALSE)
```

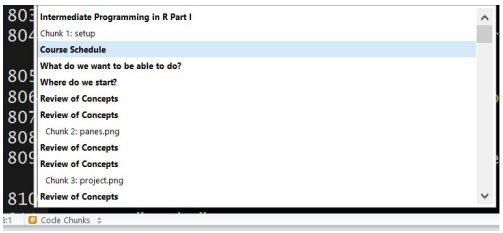
Code Chunks

Can name code chunks to help organization!

- When calling a chunk, add name after `r`

```
```{r name-of-chunk, options...}
code
```
```

Code Chunks

Can name code chunks to help organization!

- When calling a chunk, add name after `r`

```
```{r name-of-chunk, options...}
code
```
```

- TOC type menu in bottom left of notebook editor!



Code Chunks

In a large analysis it may take a long time to run code chunks/knit your document

- Can "Cache" results! Code will only rerun if it has changed.
- Option to set up code dependencies using chunk names
- Use `cache = TRUE` in code chunk definition
- Delete folders created to rerun everything

Code Chunks

In a large analysis it may take a long time to run code chunks/knit your document

- Can "Cache" results! Code will only rerun if it has changed.
- Option to set up code dependencies using chunk names
- Use `cache = TRUE` in code chunk definition
- Delete folders created to rerun everything
- Modify *global options* in `setup` R chunk after your YAML header. Make changes and see differences! (Local settings overwrite global.) (Add `library(tidyverse)` to a code chunk to have messages that would pop up!)

```
opts_chunk$set(cache = TRUE)
```

Images/Equations and Misc.

Adding images in markdown: `![](path/to/file)`

- Not ideal… difficult to control size/scale
- Better way to add images use R code!
- `knitr` package has `include_graphics` function
- Use knitr or code chunk options to control size/scale!
- Ex:
  ```{r graphics, out.width = "800px", echo = FALSE}
  knitr::include_graphics(path/to/file)
  ```

Images/Equations and Misc.

Adding Equations

- Inline equation: `$A = \pi*r^{2}$` becomes $A = \pi * r^2$
- Block equation `$$A = \pi*r^{2}$$` becomes

$$A = \pi * r^2$$

- Outputting equations for HTML is done through MathJax (javascript)
- For PDFs it is done through LaTeX (may need to install)

Images/Equations and Misc.

Outputting data tables better with `kable` from `knitr` package

```
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
```

```
kable(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

Activity

- **Using Notebook Activity** instructions available on web
- Ask questions! TAs and I will float about the room
- Feel free to ask questions about anything you didn't understand as well!

What do we want to be able to do?

- Communicate findings effectively
- Document findings
- Make process reproducible
- Share process

Where are we at?

- R Markdown Basics
  - Code Chunks
  - Images/Equations/Misc.
- R Markdown Options
  - Documents: PDF, HTML
  - Presentations: Slides
  - Interactive Components
- R Shiny Applications/Presentations

Common Outputs

R Markdown really flexible!

Common Outputs

Change output type in the YAML header:

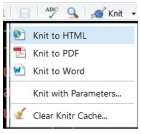
- HTML (a web page)

```
output:  html_document
```

Use code explicity:

```
rmarkdown::render("file.Rmd", output_format = "word_document")
```

Use Knit menu:



Common Outputs

For HTML can include Table of Contents with options

```
output:
  html_document:
    toc: true
    toc_float: true
```

Common Outputs

For HTML can include Table of Contents with options

```
output:
  html_document:
    toc: true
    toc_float: true
```

For html_documents another option is to make the code chunks hidden by default, but visible with a click:

```
output:
  html_document:
    code_folding: hide
```

Common Outputs

- PDF
- May want to install LaTeX for equations

```
output: pdf_document
```

Common Outputs

- PDF
- May want to install LaTeX for equations

```
output: pdf_document
```

- Word

```
output: word_document
```

- Go back to template, change output type and knit. Add some HTML options (careful to drop down lines and use spacing as done in documentation).

Common Outputs

Presentations/Slides (new slides with `##`)

- `output: ioslides_presentation` - HTML presentation
- `slidy_presentation` - HTML presentation
- `beamer_presentation` - PDF presentation with LaTeX Beamer

Common Outputs

Presentations/Slides (new slides with `##`)

- `output: ioslides_presentation` - HTML presentation
- `slidy_presentation` - HTML presentation
- `beamer_presentation` - PDF presentation with LaTeX Beamer
- Shiny (covered later) slides

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

or

```
runtime: shiny
output: ioslides_presentation
```

Interactivity

HTML documents inherently interactive

- Widgets can be included

```
library(leaflet)
leaflet() %>%
  setView(174.764, -36.877, zoom = 16) %>%
  addTiles() %>%
  addMarkers(174.764, -36.877, popup = "Maungawhau")
```

Interactivity

```
## Warning: package 'leaflet' was built under R version 3.5.3
```

Leaflet

Interactivity

Interactive tables with `DT` library

```
library(DT)
datatable(iris)
```

Interactivity

Interactivity

- 3d scatterplots with `rthreejs` package

```
if(!require("devtools")) install.packages("devtools")
devtools::install_github("bwlewis/rthreejs")

library(threejs)

scatterplot3js(x = iris$Sepal.Width, y = iris$Sepal.Length,
                              z = iris$Petal.Width, color =
                                 c(rep("blue", 50), rep("red", 50),
                                     rep("green", 50)),
                              size = 0.5)
```

Interactivity

Interactivity

Previous interactivity happened in the browser

- Great because anyone can access with a browser
- Bad because you can't have as much functionality as you want…
- Shiny allows for interactivity with R!
- Only con: Requires R running somewhere
- Examples: Shiny Showcase, Shiny Gallery

Creating PDFs

- Can change type of output to PDF
- Many PDF options
- Rendering to PDF can be a bit difficult to set up if you don't do latex
- Go through this document (we'll come help!) to set things up
- **Outputs and Interactivity Activity** instructions available on web

What do we want to be able to do?

- Communicate findings effectively
- Document findings
- Make process reproducible
- Share process

Where are we at?

- R Markdown Basics
  - Code Chunks
  - Images/Equations/Misc.
- R Markdown Options
  - Documents: PDF, HTML
  - Presentations: Slides
  - Interactive Components
- R Shiny Applications/Presentations

Parameters (Advanced)

Parameters can be added to the YAML header

- Can help to automate reports!

```
title: "NFL Reports"
author: "Justin Post"
date: "August 2018"
output: html_document
params:
      team: "Pittsburgh Steelers"
```

- Access via `params$team`
- Can 'Knit with parameters'
- Example: Let's open up the NFL.Rmd document

Parameters (Advanced)

- Create data frame for each class (here team)

```
scoreData <- read_csv("https://github.com/jbpost2/
        IntermediateR/blob/master/datasets/scoresFull.csv?raw=true")


teamIDs <- unique(scoreData$awayTeam)
reports <- tibble(
  teamIDs = teamIDs,
  filename = paste0("TeamID-", teamIDs, ".html")
,
  params = lapply(teamIDs, FUN = function(x){list(team = x)})
)
```

Parameters (Advanced)

```
reports
```

```
## # A tibble: 32 x 3
##   teamIDs             filename                        params
##   <chr>               <chr>                           <list>
## 1 San Francisco 49ers TeamID-San Francisco 49ers.html <list [1]>
## 2 Minnesota Vikings   TeamID-Minnesota Vikings.html   <list [1]>
## 3 New Orleans Saints  TeamID-New Orleans Saints.html  <list [1]>
## 4 New York Jets       TeamID-New York Jets.html       <list [1]>
## 5 Arizona Cardinals   TeamID-Arizona Cardinals.html   <list [1]>
## # ... with 27 more rows
```

Parameters (Advanced)

Now knit via the following code:

```
reports %>%
  select(output_file = filename, params) %>%
  purrr::pwalk(rmarkdown::render, input = "NFL.Rmd")
#pwalk applies the function render with the arguments given
```