



# **Data Understanding: Advanced Reporting**

**Patrick Mathias**

May 9, 2022

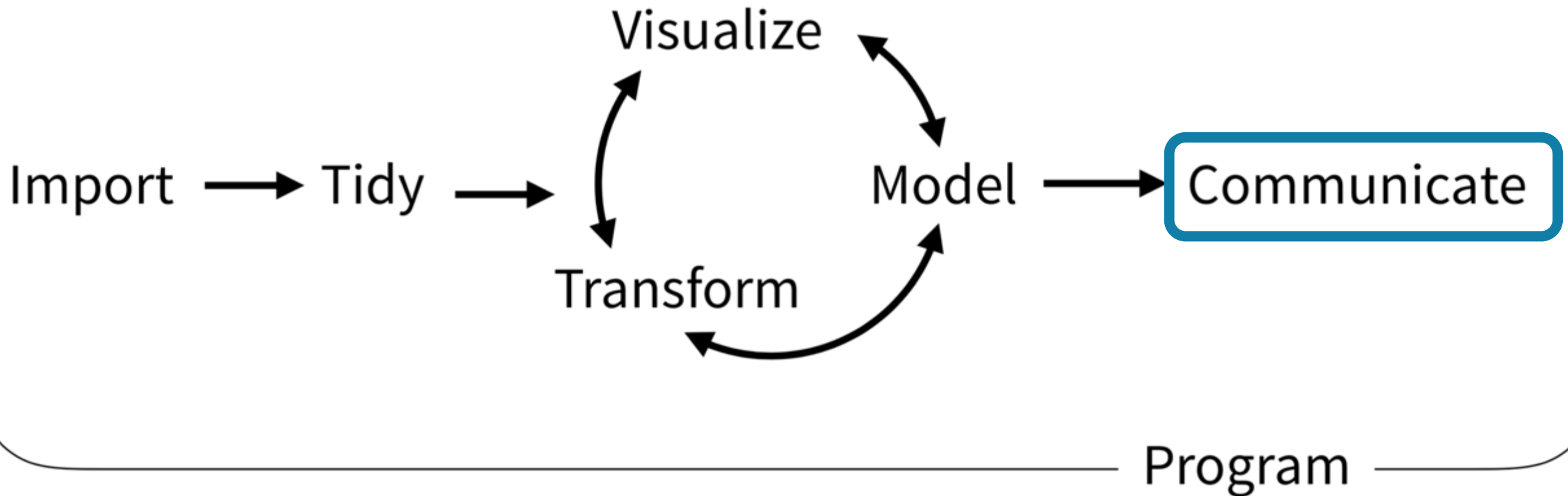
# Goals

1. Build R Markdown reports using formatting outputs beyond standard document formats

# Objectives

1. Format a flexdashboard to improve display of multiple plots
2. Convert a static plot into an interactive plot

# Typical Data Science Pipeline



# Refresher Quiz

You need to install 3 new packages you've never used before. What function do you run if you need to install the *flexdashboard*, *plotly*, and *DT* packages?

# Refresher: Installing packages

- Installing a package

```
install.packages(c("flexdashboard", "plotly", "DT"))
```

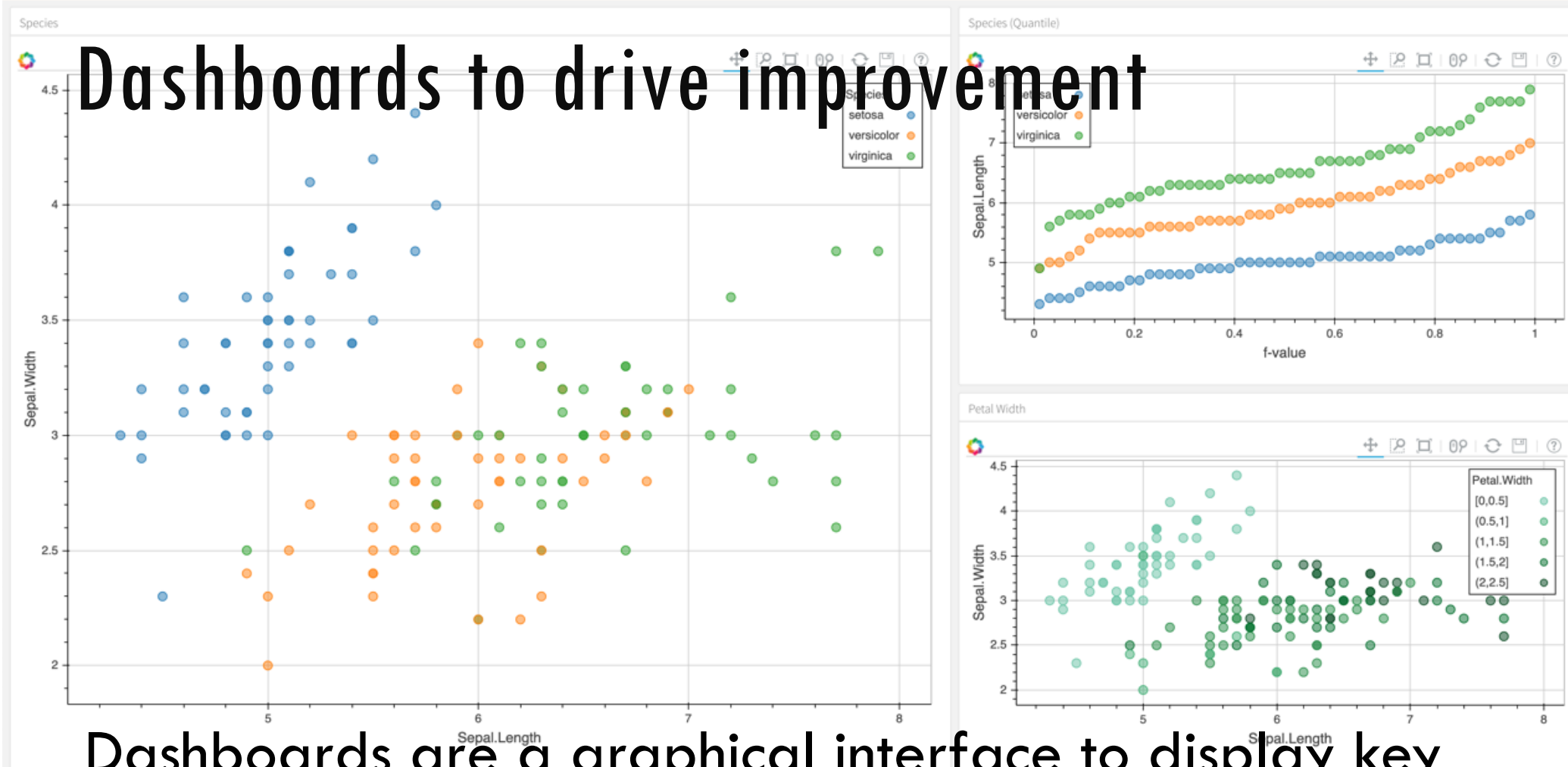
- Loading into your environment

```
library(flexdashboard)  
library(plotly)  
library(DT)
```

The flexdashboard and plotly packages were already installed in your Rstudio Cloud instance. To install them locally use `install.packages`.



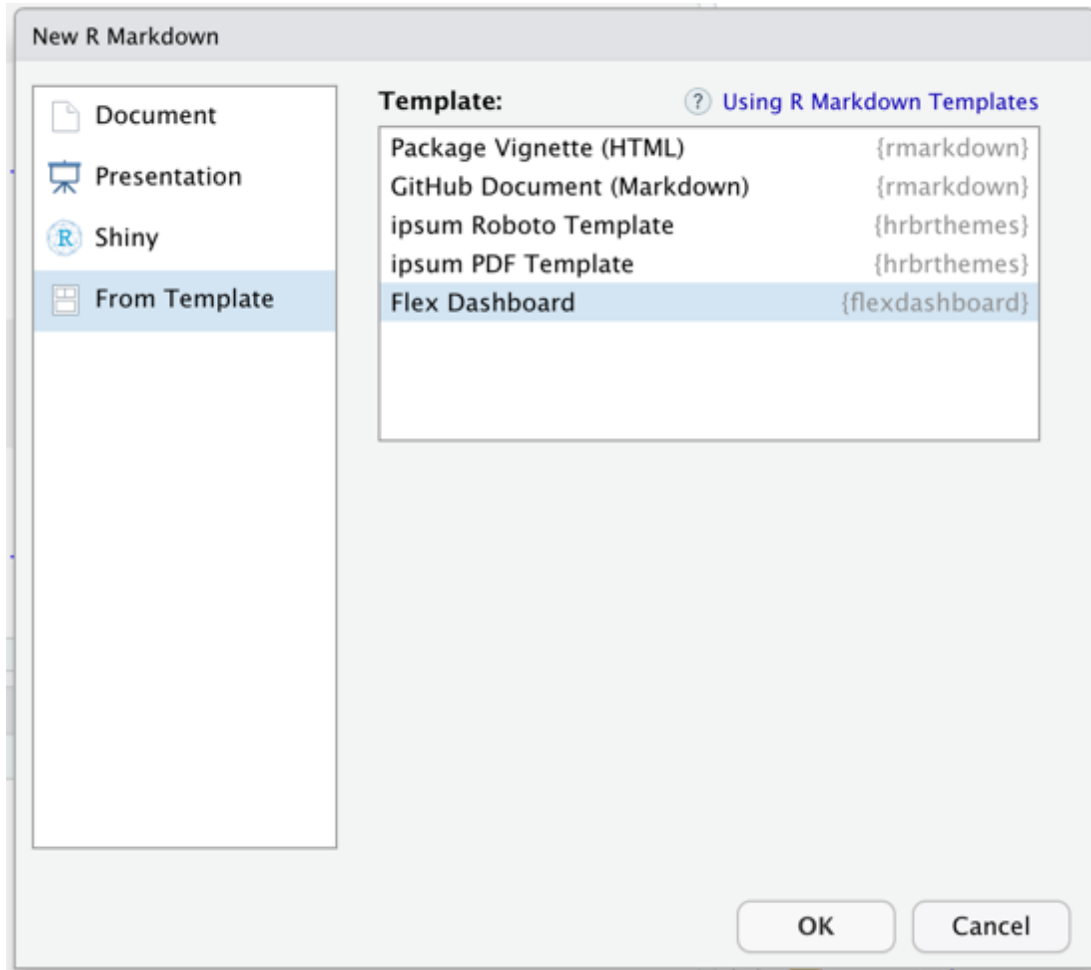
# From R Markdown to Quick and Painless Dashboards



Dashboards are a graphical interface to display key performance indicators or other metrics

Intended to represent multiple pieces of information at a glance

# flexdashboard provides easy dashboard templates for reporting



Produces HTML file that can be opened on web browsers

Or deployed on existing web server

Provides row or column based layouts

Get started on your desktop by running:

```
install.packages("flexdashboard")
```

<https://rmarkdown.rstudio.com/flexdashboard/>



Untitled1 x

←

→

📄

📁

ABC

🔍

🌐 Knit

⚙️

➕ C Insert

↑

↓

🏃 Run

🔄

☰

```
1 |---
2 title: "Untitled"
3 output:
4   flexdashboard::flex_dashboard: ← flexdashboard output format
5     orientation: columns ← layout page by columns
6     vertical_layout: fill
7   ---
8
9   ```{r setup, include=FALSE}
10  library(flexdashboard)
11  ```
12
13  Column {data-width=650} define width
14  -----
15
16  ### Chart A title for chart
17
18  ```{r}
19
20  ```
21
22  Column {data-width=350}
23  -----
24
25  ### Chart B
```

1:1 # Untitled ↕ R Markdown ↕

```
1 ---
2 title: "Column Orientation"
3 output: flexdashboard::flex_dashboard
4 ---
5
6 Column
7 -----
8 |
9 ### Chart 1
10
11 ```{r}
12 ```
13
14 Column
15 -----
16
17 ### Chart 2
18
19 ```{r}
20 ```
21
22 ### Chart 3
23
24 ```{r}
25 ```
26
```

Chart 1

Chart 2

Chart 3

```
1 |--
2 title: "Row Orientation"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6   ---
7
8   Row
9   -----
10
11  ### Chart 1
12
13  ```${r}```
14  ```
15
16  Row
17  -----
18
19  ### Chart 2
20
21  ```${r}```
22  ```
23
24  ### Chart 3
25
26  ```${r}```
27  ```
28
```

**Chart 1**

**Chart 2**

**Chart 3**

```
1 |---
2 |title: "Chart Stack (Scrolling)"
3 |output:
4 |  flexdashboard::flex_dashboard:
5 |    vertical_layout: scroll
6 |---
7 |
8 |### Chart 1
9 |
10| ```${r}```
11| ```
12|
13|### Chart 2
14|
15| ```${r}```
16| ```
17|
18|### Chart 3
19|
20| ```${r}```
21| ```
22|
23|
24|
25|
```

**Chart 1**

**Chart 2**

**Chart 3**

# Your Turn #1

1. Open “06 - Advanced Reporting.Rmd” to work with a draft COVID-19 flexdashboard and run the setup chunk. Knit the document to see the dashboard output.
2. The “Test Volumes Over Time” plot could show additional information regarding positive tests. Add fill to your barplot to show the result field in addition to overall test volume by day. Run that code chunk to see the output.
3. Too much information is crunched on the right side. Change the layout from columns to a row orientation. The 2<sup>nd</sup> and 3<sup>rd</sup> plots (Turnaround Times and Cycle Thresholds) should appear on the 2<sup>nd</sup> row.

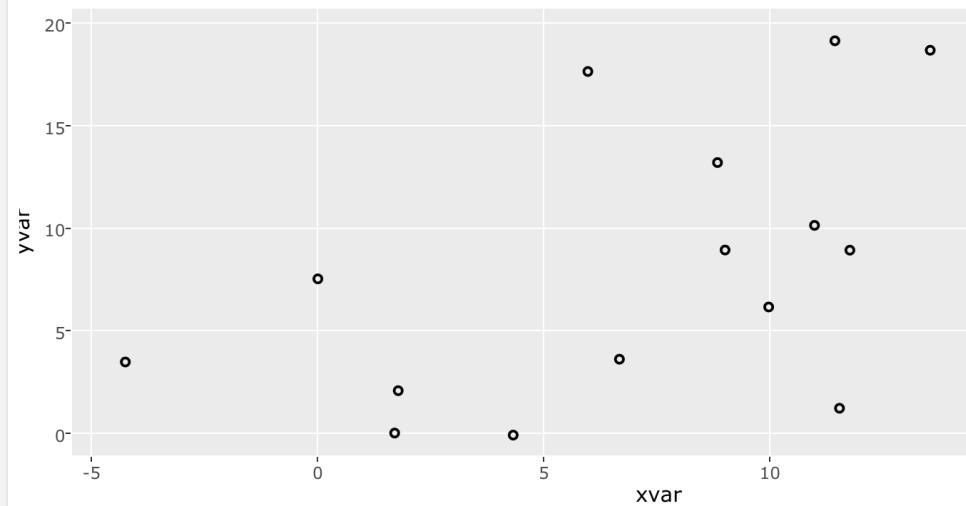
# Customization

ggplotly geoms

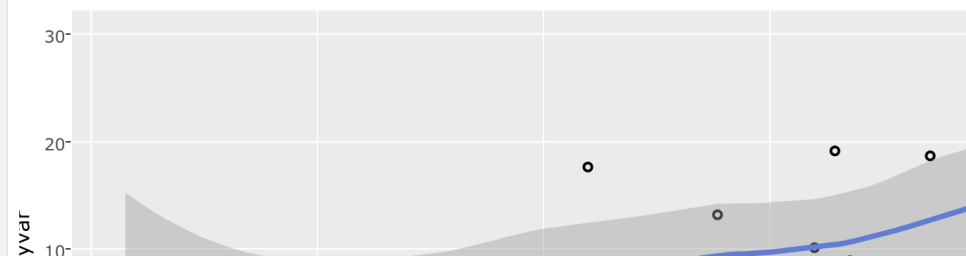
geom\_point

geom\_density

Scatter Chart with geom\_point



geom\_smooth with Loess Smoothed Fit



Bootswatch Themes ▾ Download ▾ Help Blog

## Cerulean

A calm blue sky

Primary

Secondary

Success

Info

Warning

## Cerulean

A calm blue sky

PREVIEW

DOWNLOAD ▾

Bootswatch Themes ▾ Download ▾ Help Blog

## Darkly

Flatly in night mode

Primary

Secondary

Success

Info

Warning

## Darkly

Flatly in night mode

PREVIEW

DOWNLOAD ▾

Bootswatch Themes ▾ Download ▾ Help Blog

## Cosmo

An ode to Metro

Primary

Secondary

Success

Info

Warning

Danger

## Cosmo

An ode to Metro

PREVIEW

DOWNLOAD ▾

Bootswatch Themes ▾ Download ▾ Help Blog

## Flatly

Flat and modern

Primary

Secondary

Success

Info

Warning

## Flatly

Flat and modern

PREVIEW

DOWNLOAD ▾



# **Making plots interactive**



# htmlwidgets for R support interactive visuals

Packages using htmlwidgets use R code to call Javascript visualization libraries (<http://www.htmlwidgets.org/>)

Use one line of code to convert a static plot into an interactive one



# Plotly package converts ggplot with a simple command

To use Plotly on your desktop install the plotly package using the following command:

```
install.packages("plotly")
```

Examples of visualizations at Plotly website:

<https://plotly.com/r/>

# Store plot as object and add one line to make interactive

```
plot_name <- ggplot(data = data_frame) +  
  geom_function(mapping = aes(mappings))  
ggplotly(plot_name)
```

# Your Turn #2

1. Load the plotly package in your setup chunk
2. Convert each of the plots into an interactive plot by storing the ggplot in an object and using the `ggplotly()` function.
3. Knit the dashboard and hover over the interactive plots.

# Other options for interactive plots

Other interactive plot packages:

- rbokeh
- Highcharter

Time series graphs with dygraphs package

Maps with leaflet package

# Interactive tables with one line

DataTables library quickly converts tables into interactive element

DT package in R – to install use:

```
install.packages("DT")
```

Use `datatable()` function on a data frame to allow:






- Filter number of entries
- Search entries
- Sort by column

# datatable example

```
datatable(head(iris), class = 'cell-border stripe')
```

Show  entries

Search:

	Sepal.Length 	Sepal.Width 	Petal.Length 	Petal.Width 	Species 
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

Showing 1 to 6 of 6 entries

Previous

1

Next

# Your Turn #3

We are going to replace one of our panels of content with an interactive table.

1. Load the DataTables package in your setup chunk
2. Rename the cycle threshold distribution panel to “Positive Result Details”. Use `filter()` to create a dataframe that only includes positive results.
3. Display an interactive table that includes the content from the positive result dataframe you created.
4. Knit the dashboard. Search “lannister” in the search box to confirm the interactive table is working.

## Your Turn #4

Customize your dashboard. Use any of the data available in your covid testing dataset to generate new plots or tables that provide insight into the underlying data.

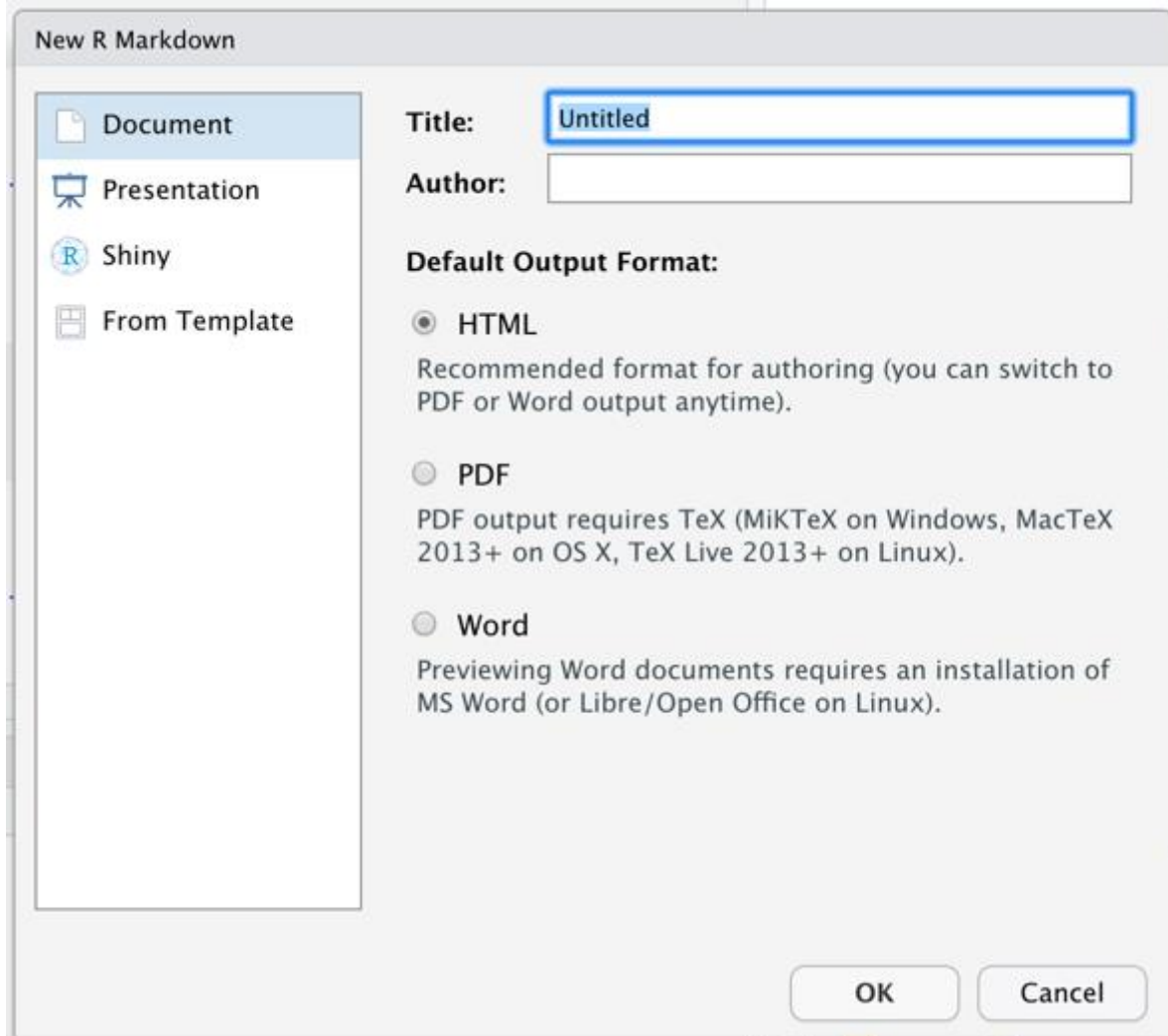




**What Else?**

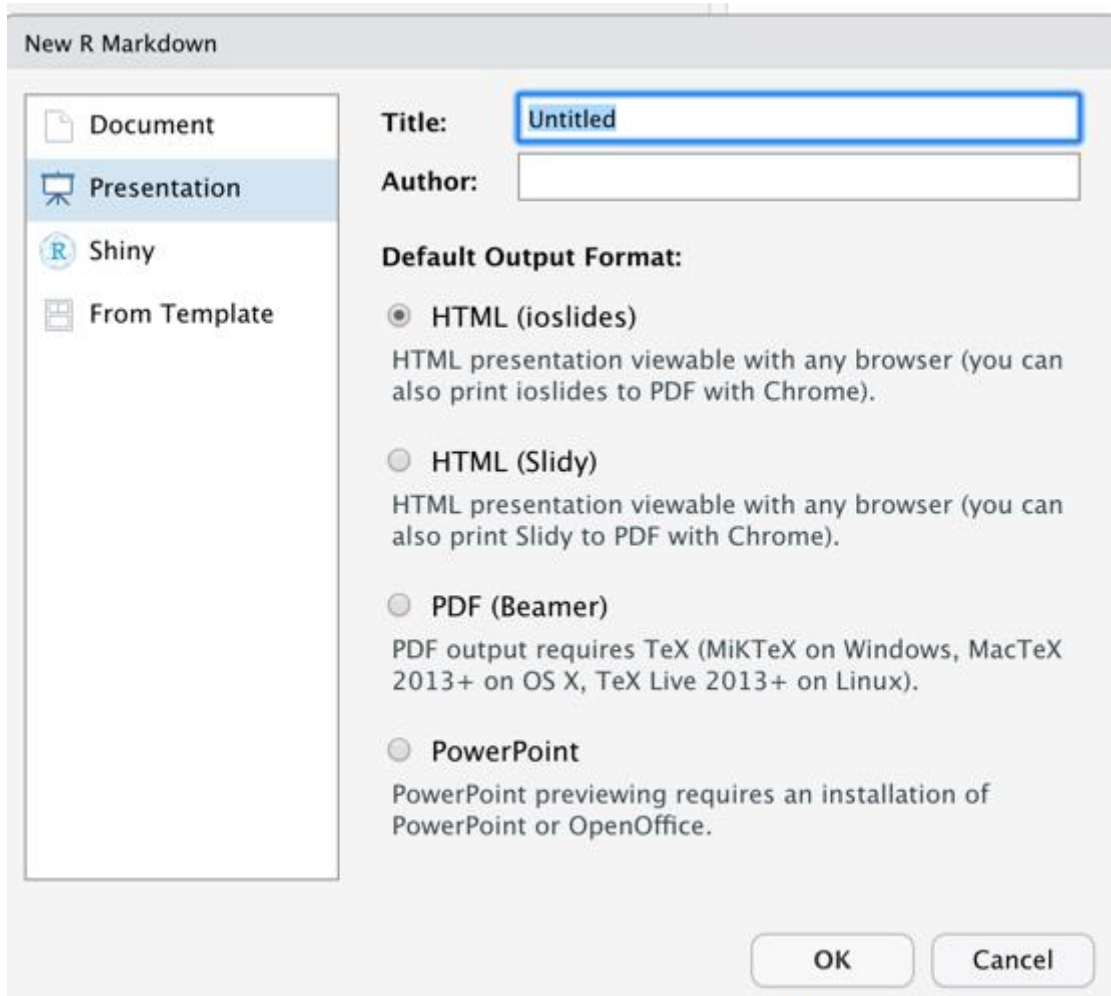


# Standard Markdown Reporting Formats



- HTML file - open with any web browser
- PDF – requires LaTeX dependencies
  - `install.packages('tinytex')`
  - `tinytex::install_tinytex()`
- Word – default format for collaborating with those who aren't familiar with R

# Formats to go straight from code to slides



Multiple HTML formats create webpage that's advanceable like slides

PDF presentation uses LaTeX in the background

Powerpoint produces simple slides

```
1 ---
2 title: "Untitled"
3 output: powerpoint_presentation
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = FALSE)
8 ```
9
10 ## R Markdown
11
12 This is an R Markdown presentation. Markdown is a simple formatting syntax for authoring HTML, PDF,
13 and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
14
15 When you click the Knit button a document will be generated that includes both content as well as
16 the output of any embedded R code chunks within the document.
17
18 ## Slide with Bullets
19
20 - Bullet 1
21 - Bullet 2
22 - Bullet 3
23
24 ## Slide with R Output
25
26 ```{r cars, echo = TRUE}
27 summary(cars)
28 ```
29
30 ## Slide with Plot
31
32 ```{r pressure}
33 plot(pressure)
34 ```
```

Output format

Each slide has its own header

# R Markdown

This is an R Markdown presentation. 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.

Example output slide

# Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

Example output slide

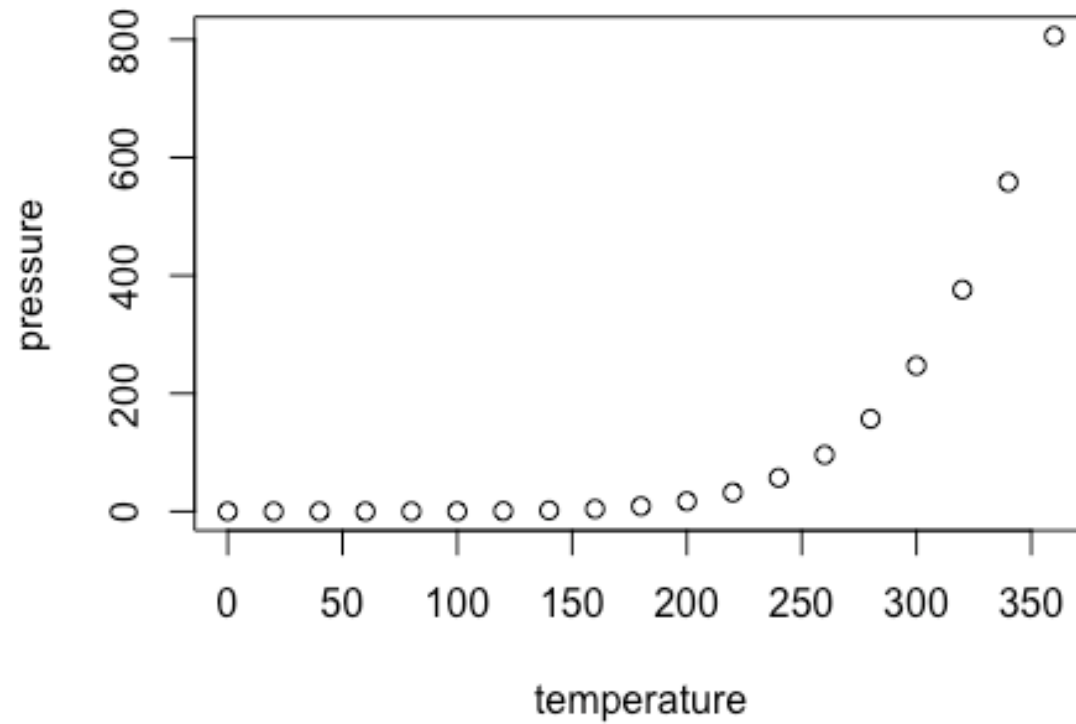
# Slide with R Output

```
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
```

**Example output slide**

# Slide with Plot



Example output slide



# Books and longer documents also generated from R Markdown

Can generate printer ready books and ebooks

Supports LaTeX features such as equations

Generates blog formatted websites



<https://github.com/rstudio/bookdown>

<https://bookdown.org/yihui/bookdown/>

<https://bookdown.org/yihui/blogdown/>

# Goals

1. Build R Markdown reports using formatting outputs beyond standard document formats

# Objectives

1. Format a flexdashboard to improve display of multiple plots
2. Convert a static plot into an interactive plot