Advanced Reporting

Session 6

Patrick Mathias

July 16, 2021

July 16 2020	**Session**	**Instructor**
1:00 pm - 1:30 pm	Instructor Introductions, Introduction to technology	Amrom Obstfeld
1:30 pm - 2:15 pm	Introduction to R and RStudio	Joe Rudolf
2:30 pm - 3:15 pm	Reproducible Reporting	Patrick Mathias
3:30 pm - 5:00 pm	Data Visualization	Stephan Kadauke
July 17 2020		
1:00 pm - 2:30 pm	Data Transformation	Amrom Obstfeld
2:45 pm - 4:15 pm	Statistical Analysis	Dan Herman
4:30 pm - 5:00 pm	Advanced Reporting	Patrick Mathias

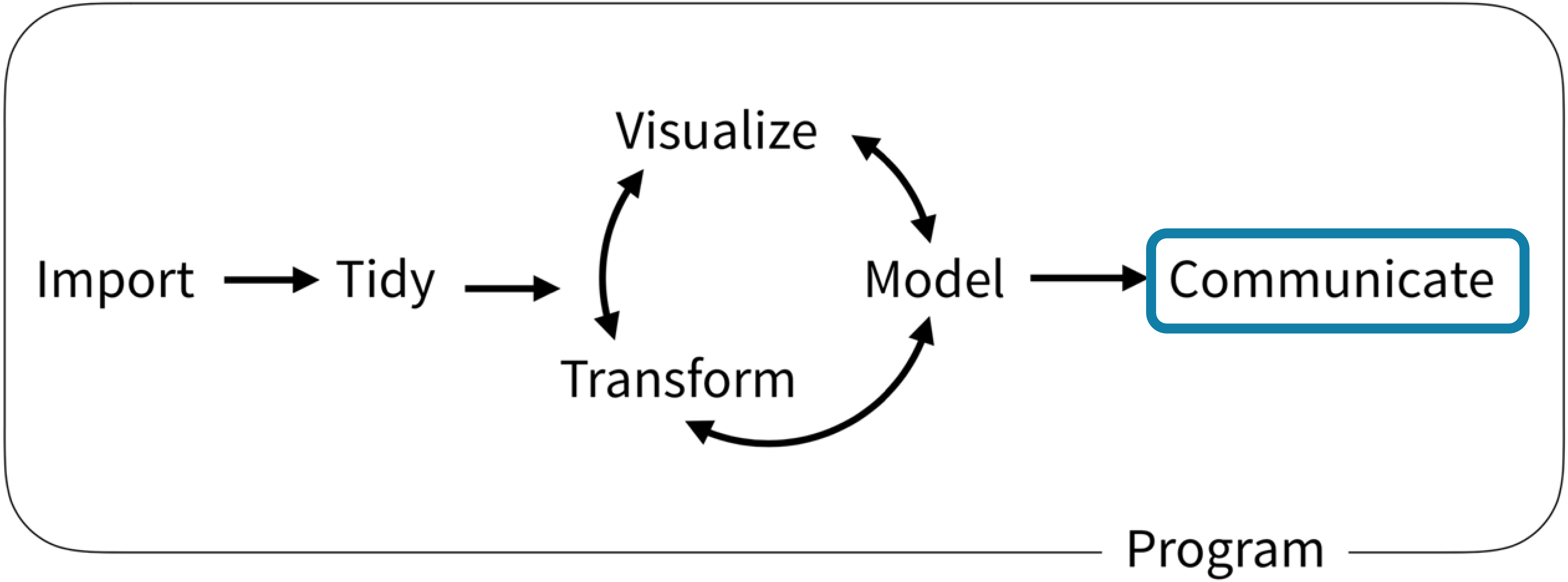
Goals

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



From *R for Data Science* (https://r4ds.had.co.nz/introduction.html)

Refresher Quiz

You need to install 2 new packages you’ve never used before. What function do you run if you need to install the *flexdashboard* and *plotly* packages?

Refresher: Installing packages

- Installing a package

```
install.packages(c(“flexdashboard”, “plotly”),  dependencies = TRUE)
```

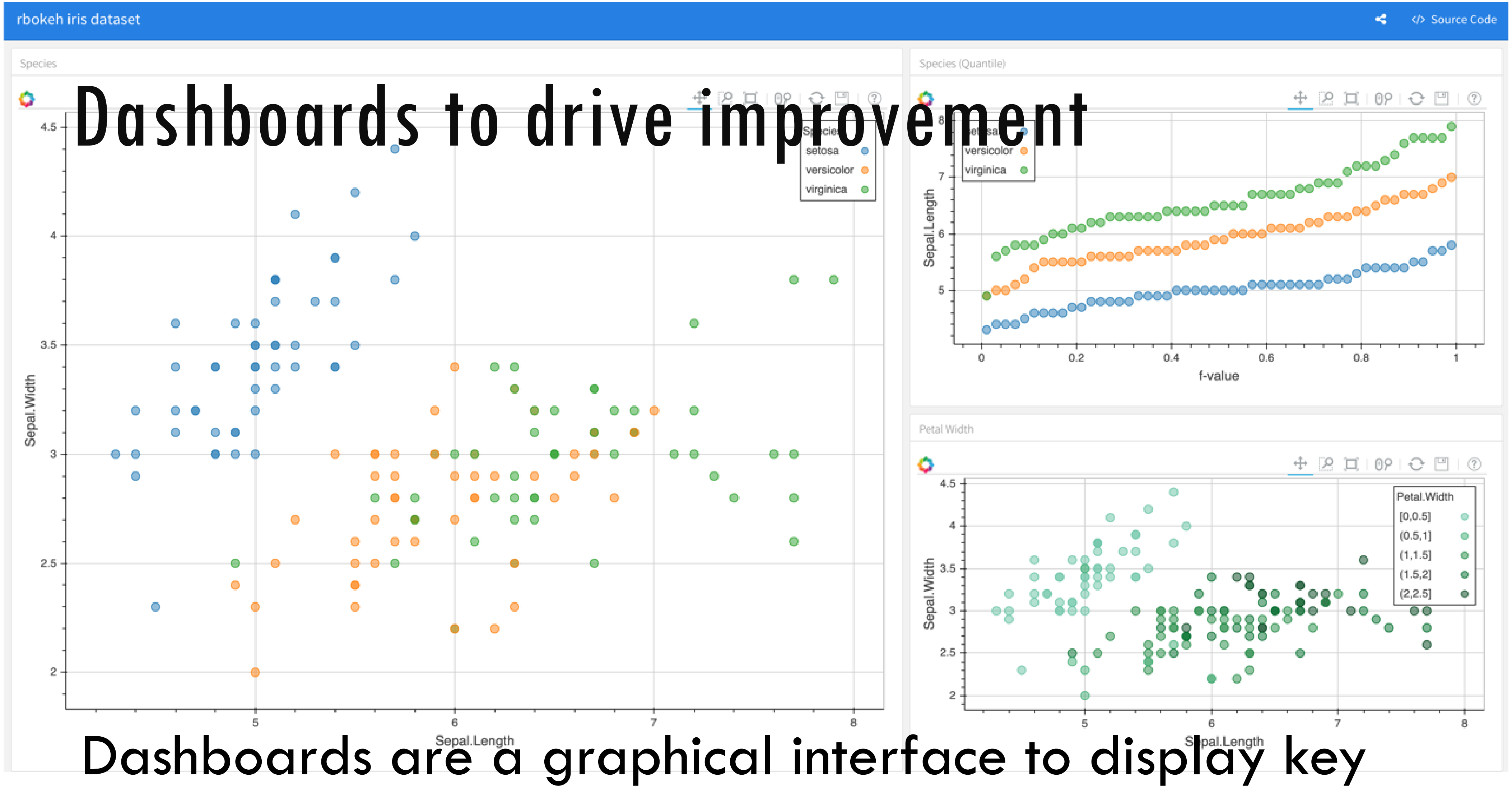
- Loading into your environment

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

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

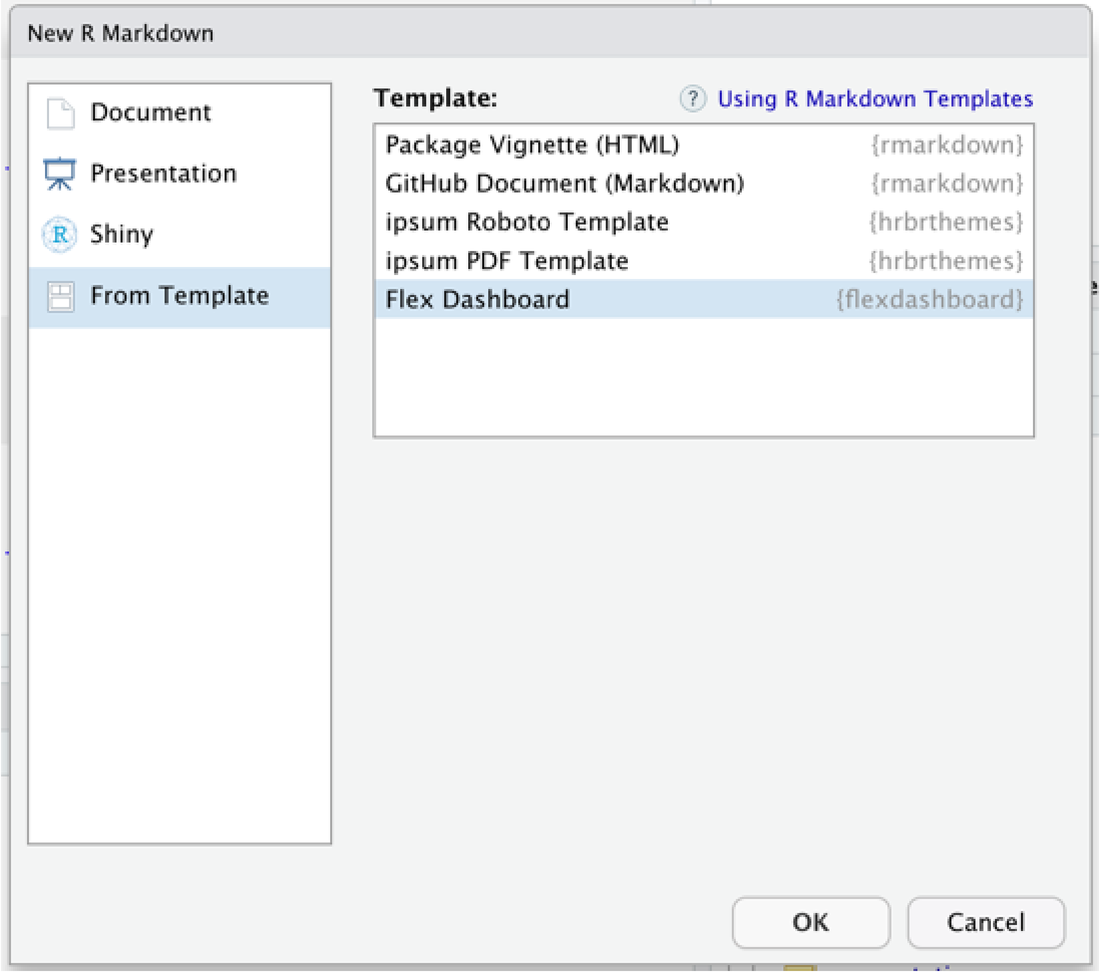
Dashboards to drive improvement



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



Template:

Package Vignette (HTML)	{rmarkdown}
GitHub Document (Markdown)	{rmarkdown}
ipsum Roboto Template	{hrbrthemes}
ipsum PDF Template	{hrbrthemes}
Flex Dashboard	{flexdashboard}

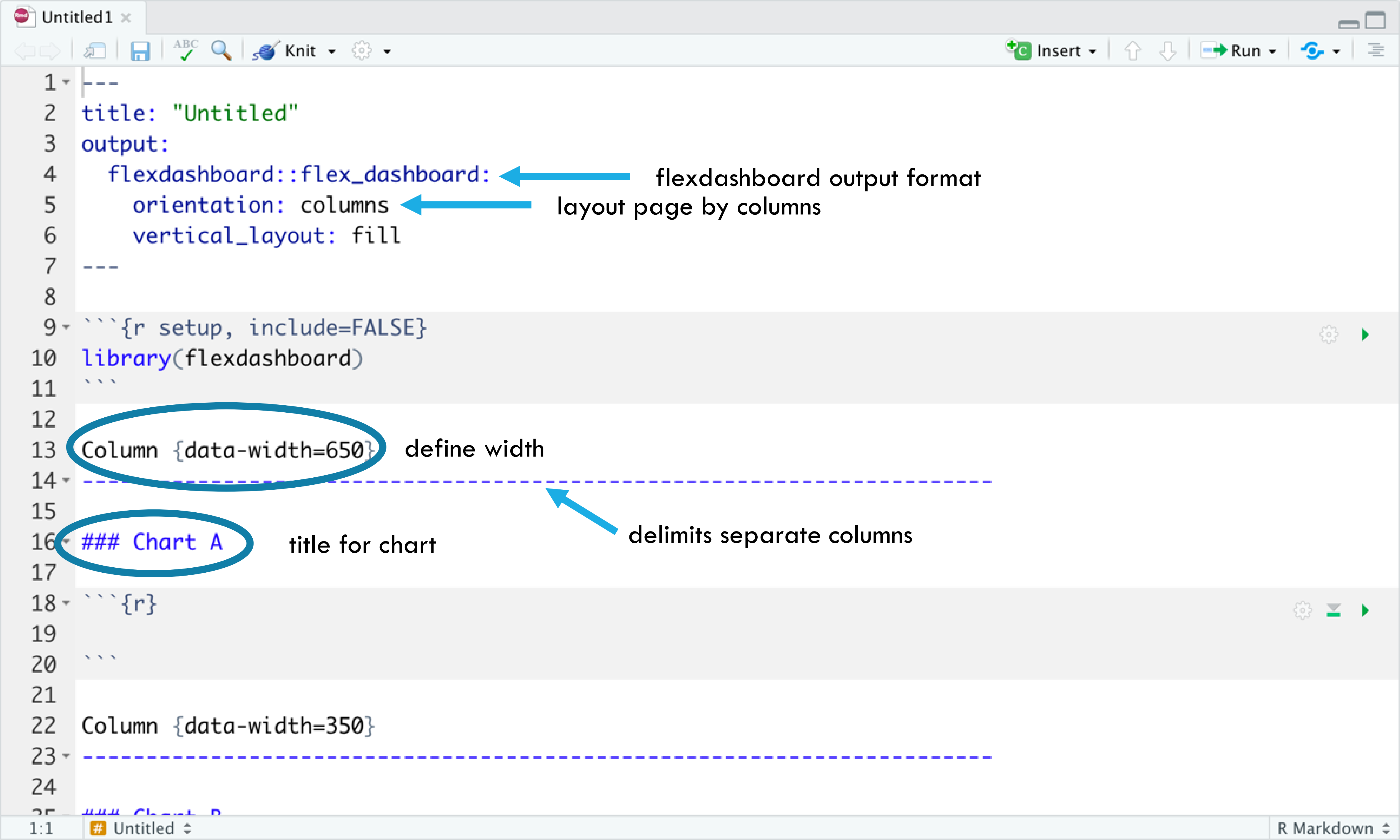
Produces HTML file that can be opened on web browsers

Or deployed on existing web server

Provides row or column based layouts

Get started by running: install.packages("flexdashboard")

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



~~~
---
title: "Untitled"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
---

```{r setup, include=FALSE}
library(flexdashboard)
~~~

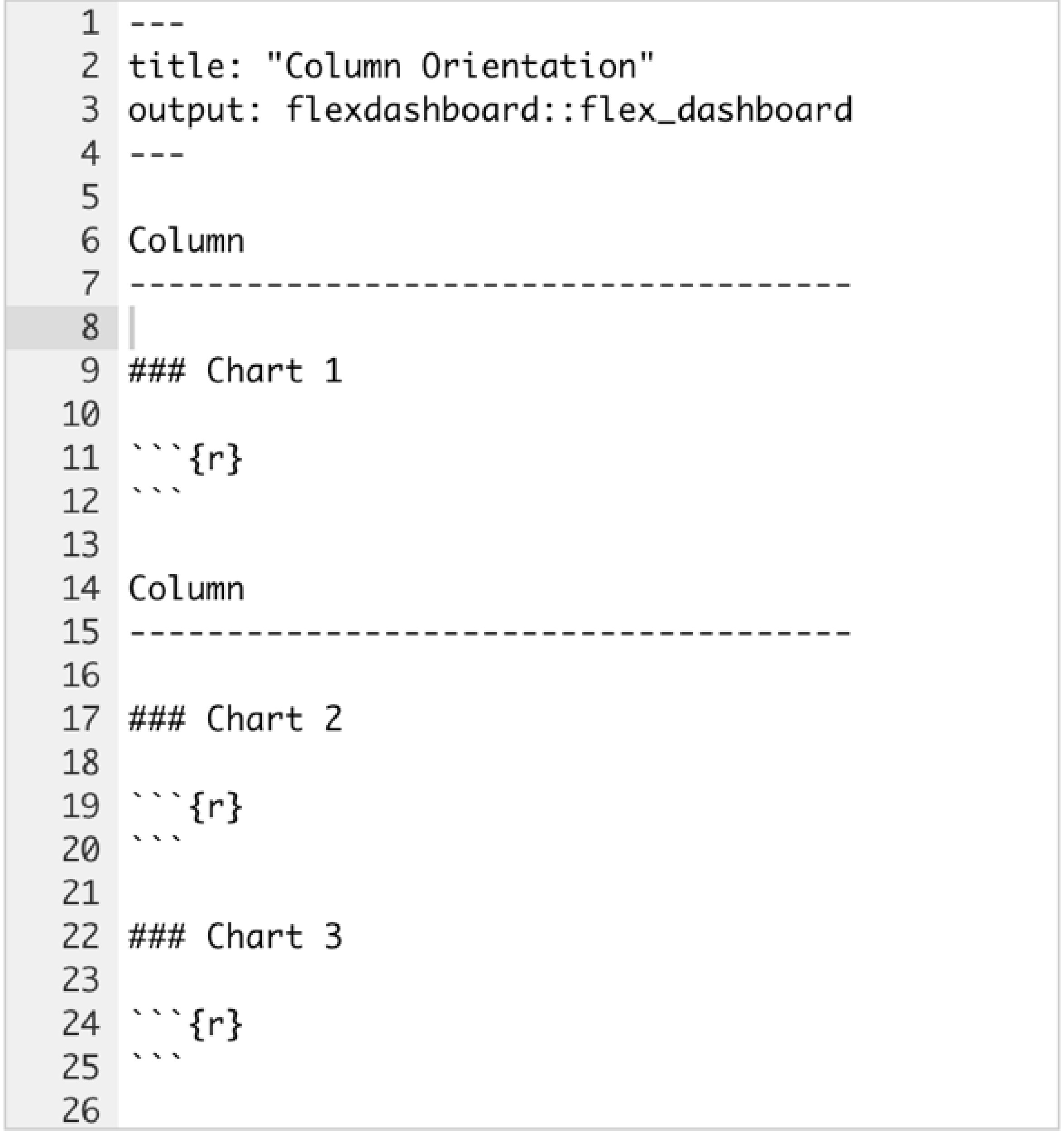
Column {data-width=650}

Chart A

```{r}

```

Column {data-width=350}

Chart B
```

flexdashboard output format

layout page by columns

define width

delimits separate columns

title for chart
```



~~~
---
title: "Column Orientation"
output: flexdashboard::flex_dashboard
---

Column
-------------------------------------

### Chart 1

```{r}
~~~

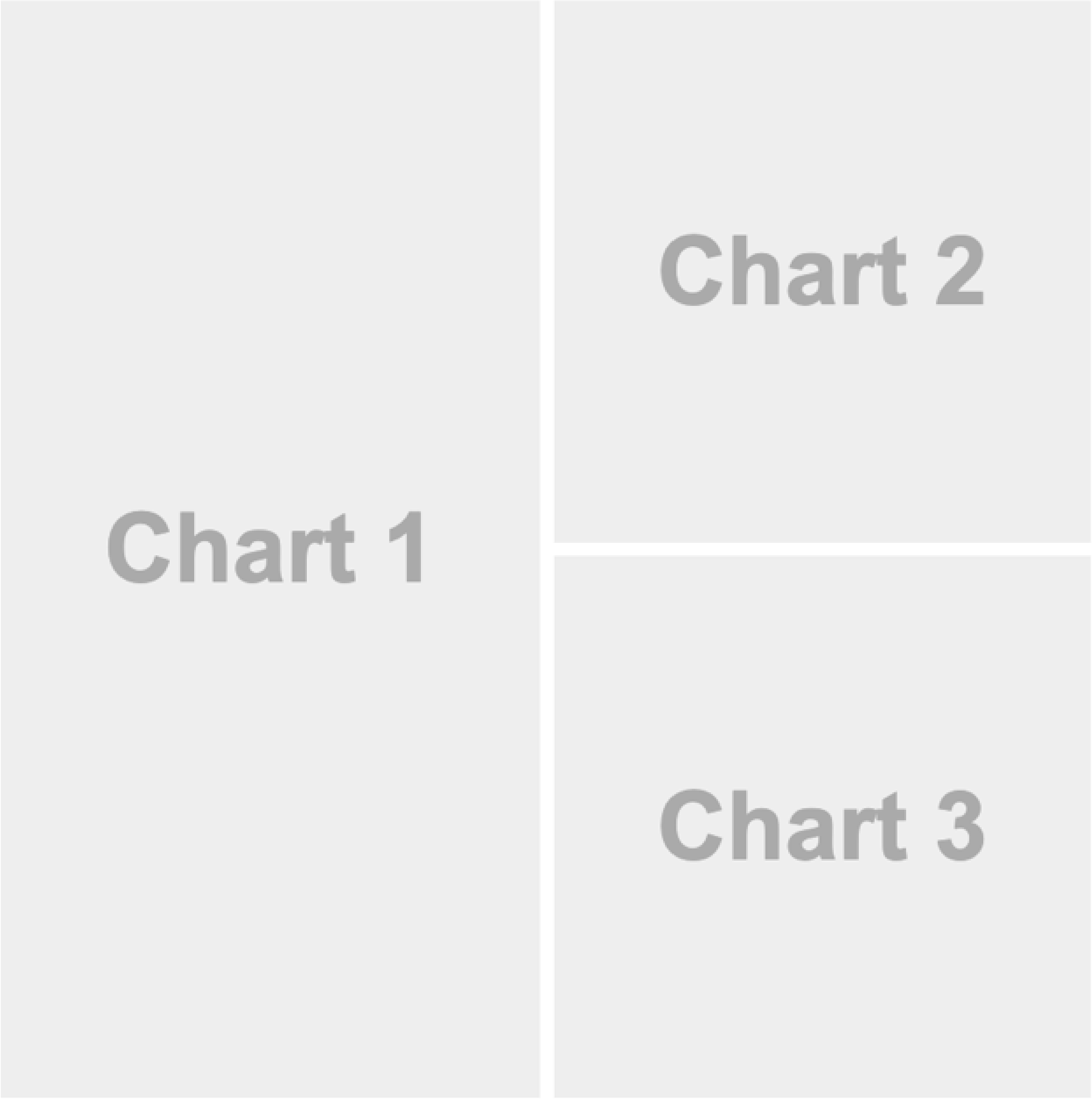
Column
-------------------------------------

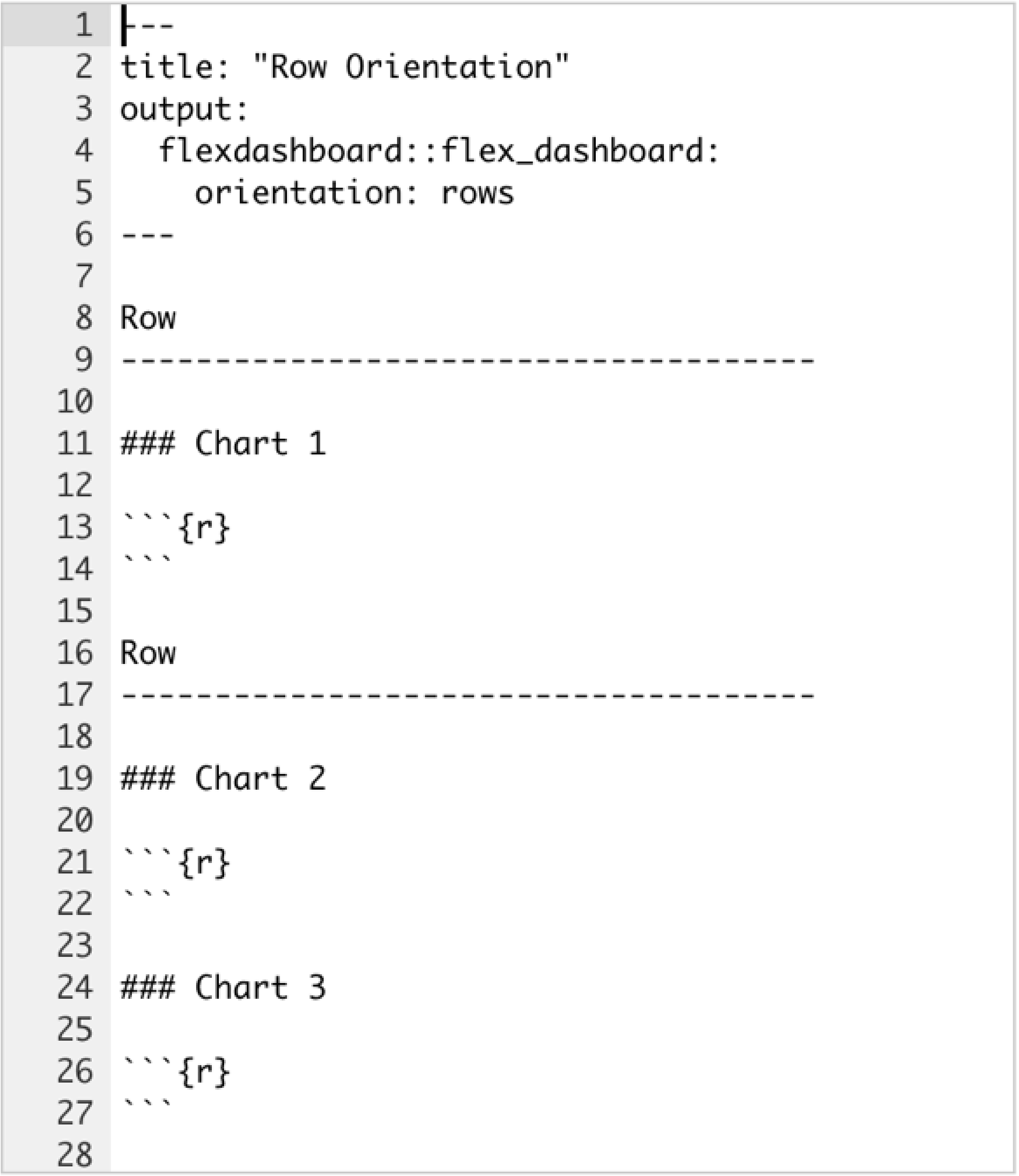
### Chart 2

```{r}
```

Chart 3

```{r}
```

```


```



~~~
---
title: "Row Orientation"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
---

Row
-------------------------------------

### Chart 1

```{r}
~~~

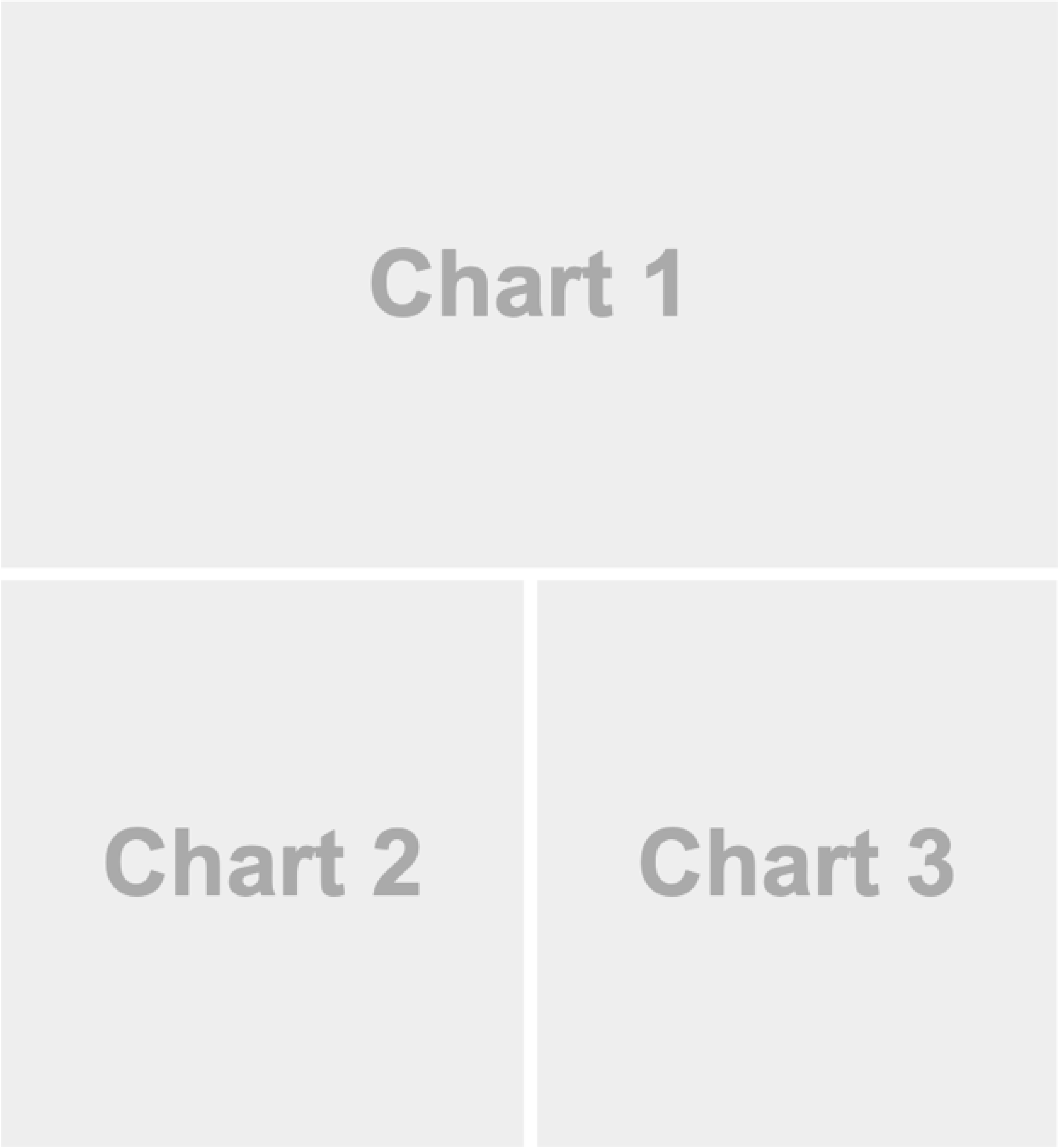
Row

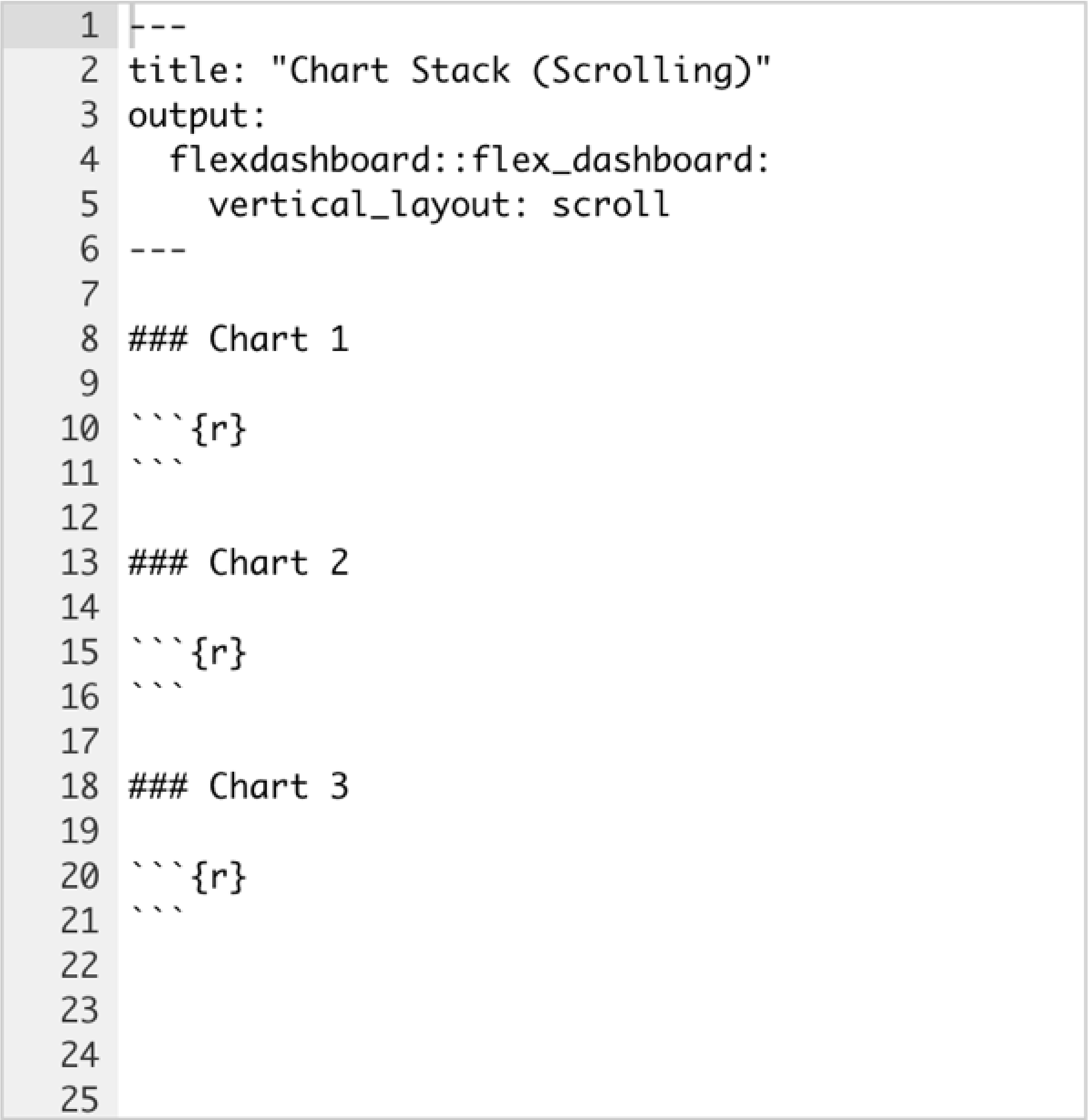
Chart 2

```{r}
```

Chart 3

```{r}
```

```


```

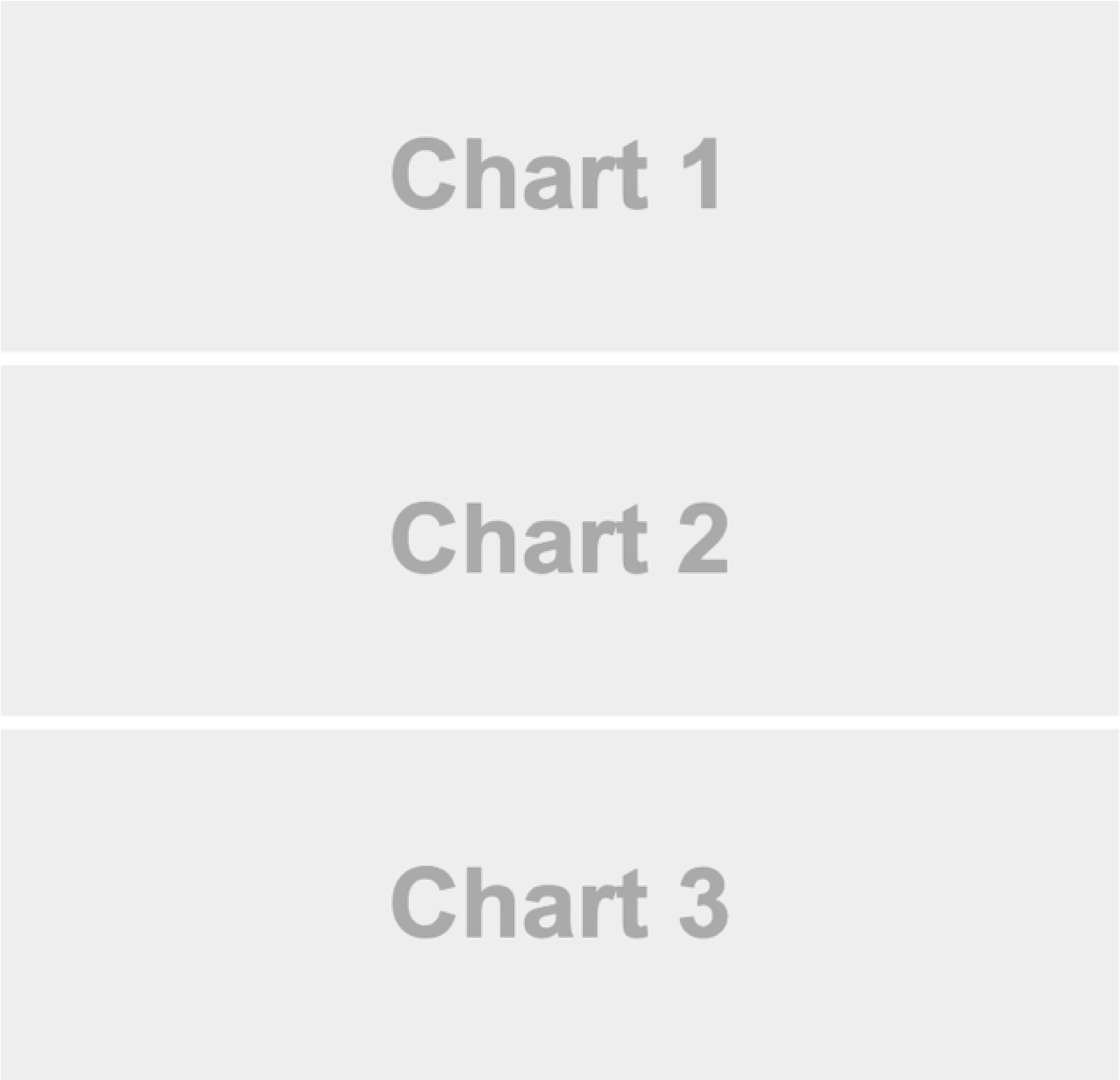


~~~
---
title: "Chart Stack (Scrolling)"
output:
  flexdashboard::flex_dashboard:
    vertical_layout: scroll
---

### Chart 1

```{r}
~~~

Chart 2

```{r}
```

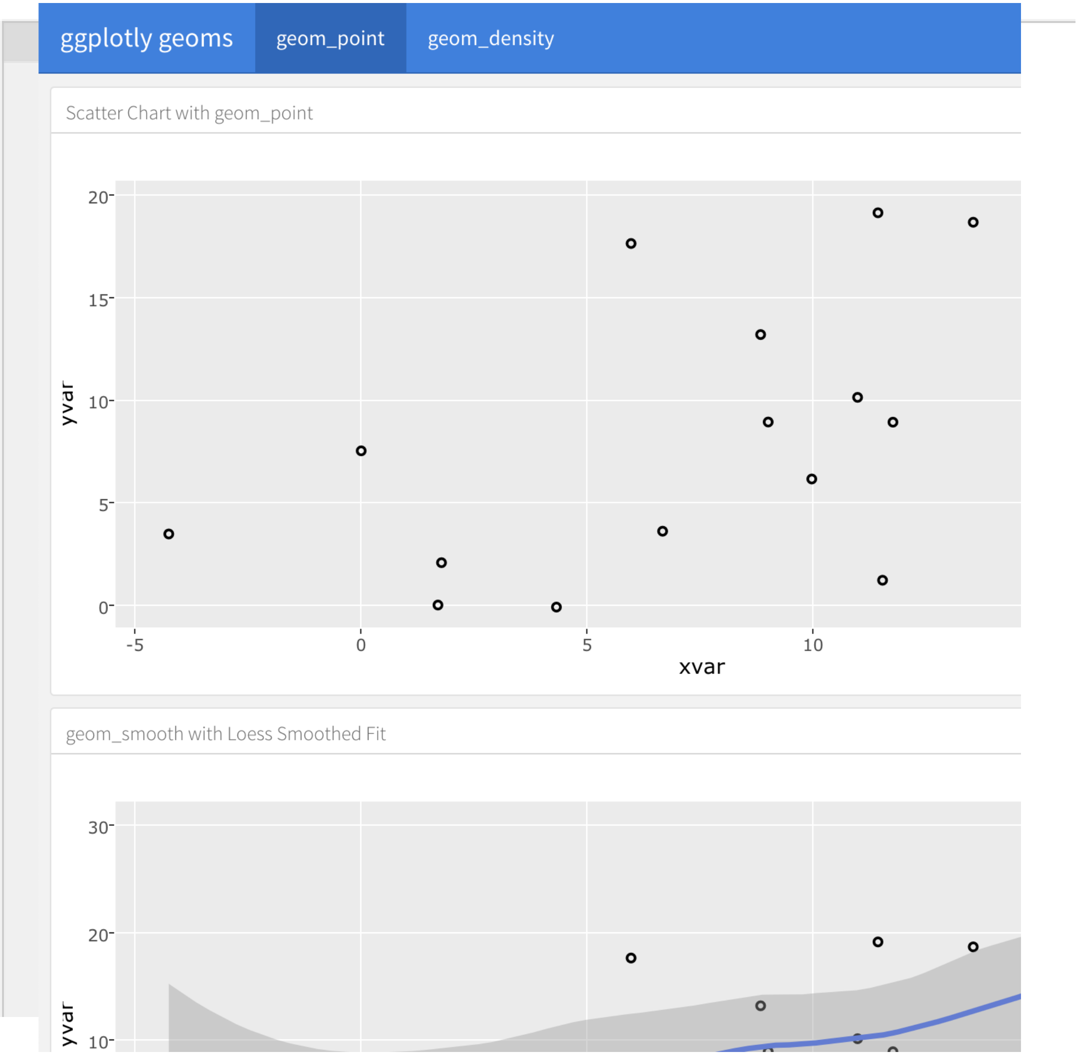
Chart 3

```{r}
```
```


```

Your Turn #1

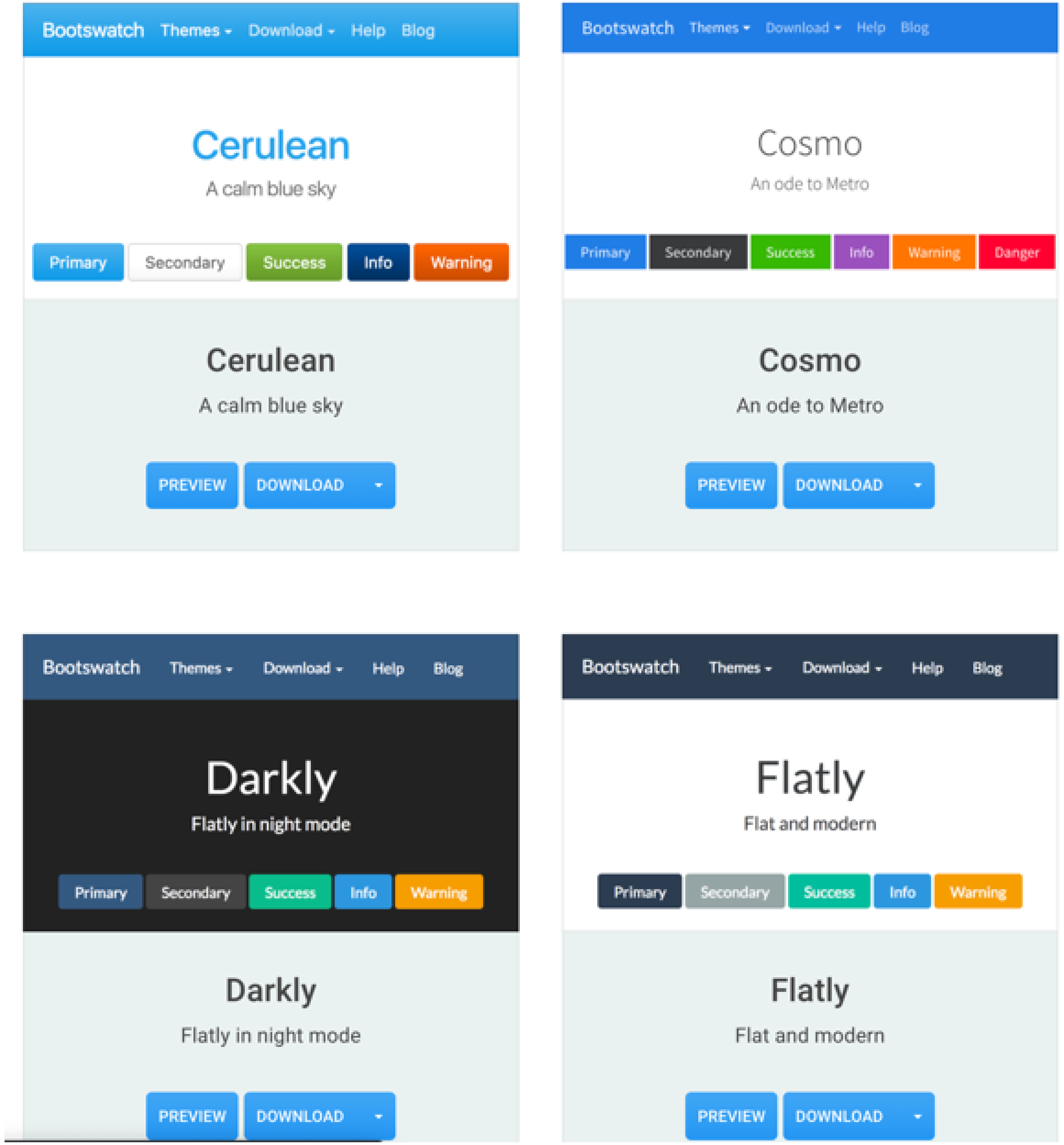
1. Open “06 - Advanced Reporting.Rmd” to work with a draft COVID-19 flexdashboard and run the setup chunk. Knit the document to see the dashboard output.
2. The ”Test Volumes Over Time” plot could show additional information regarding positive tests. Add fill to your barplot to show the result field 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 2nd and 3rd plots (Turnaround Times and Cycle Thresholds) should appear on the 2nd row.

Customization



Scatter Chart with geom_point

geom_smooth with Loess Smoothed Fit



Cerulean

A calm blue sky

Cosmo

An ode to Metro

Darkly

Flatly in night mode

Flatly

Flat and modern

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 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

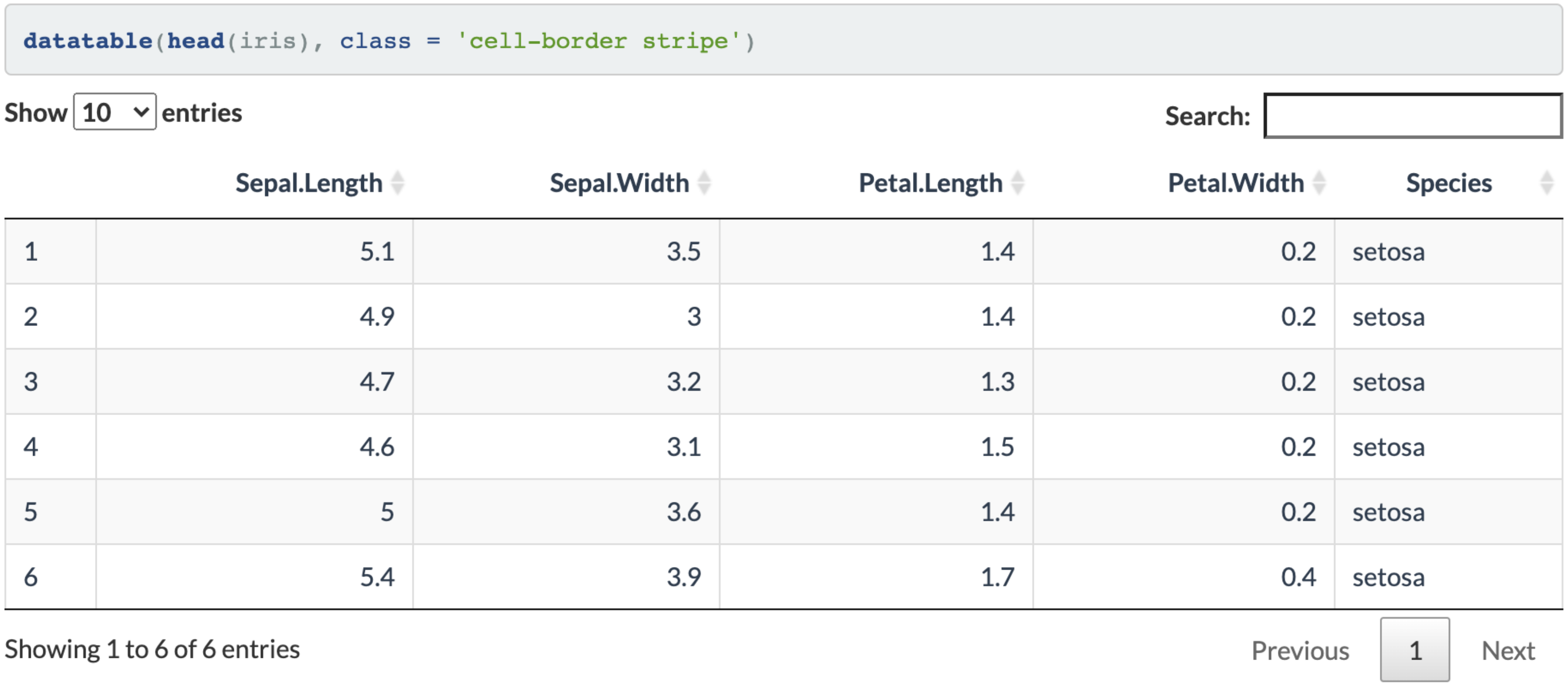
DataTables library quickly converts tables into interactive element

DT package in R

Use datatable() function on a data frame

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

datatable example



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

	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

Your Turn #3

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

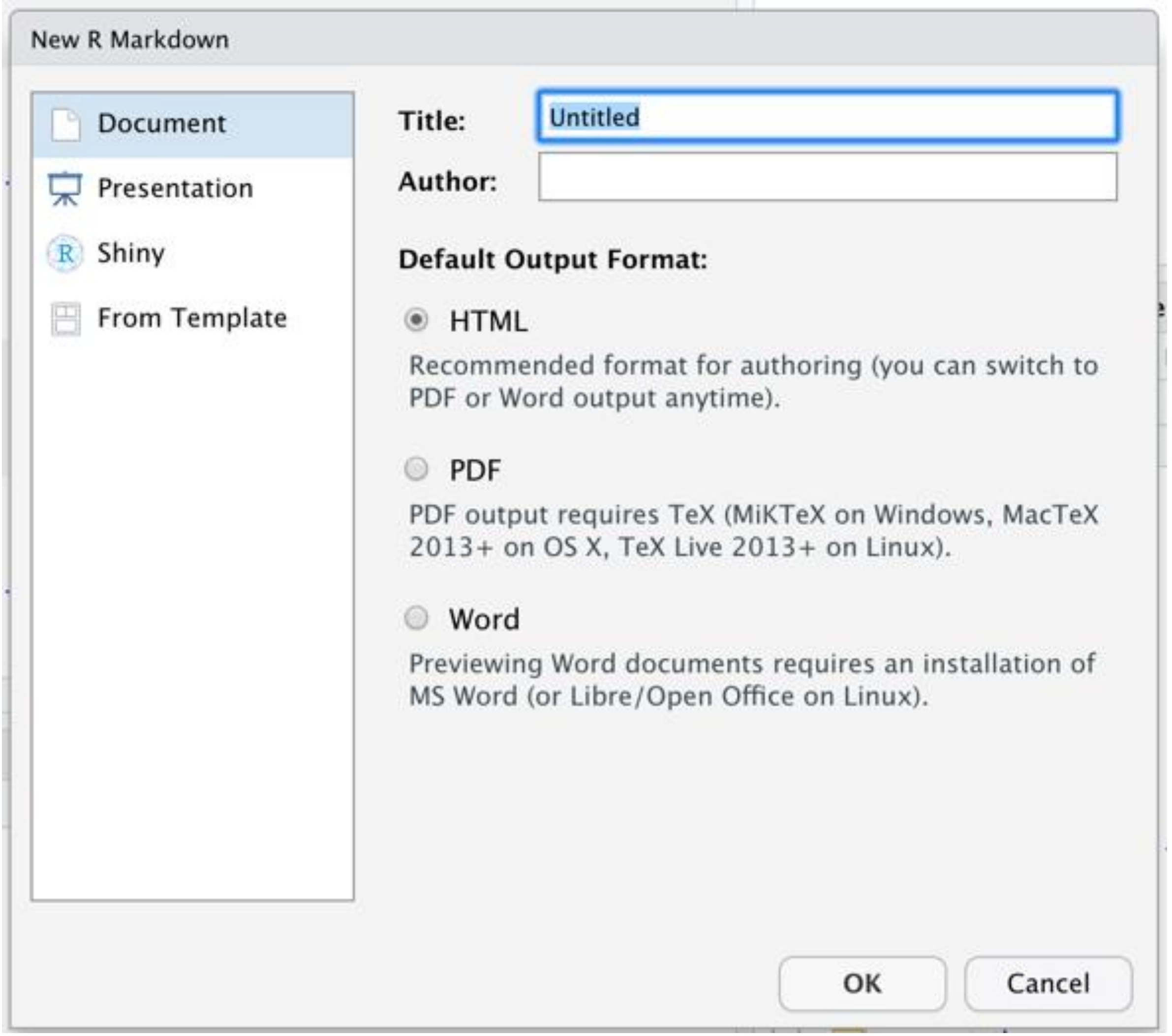
1. Load the DataTables package in your setup chunk
2. Rename the cycle threshold distribution panel to “Positive Result Details”. Use filter() to create a dataframe that only includes positive results.
3. Display an interactive table that includes the content from the positive result dataframe you created.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



Title: Untitled

Author:

Default Output Format:

● HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF

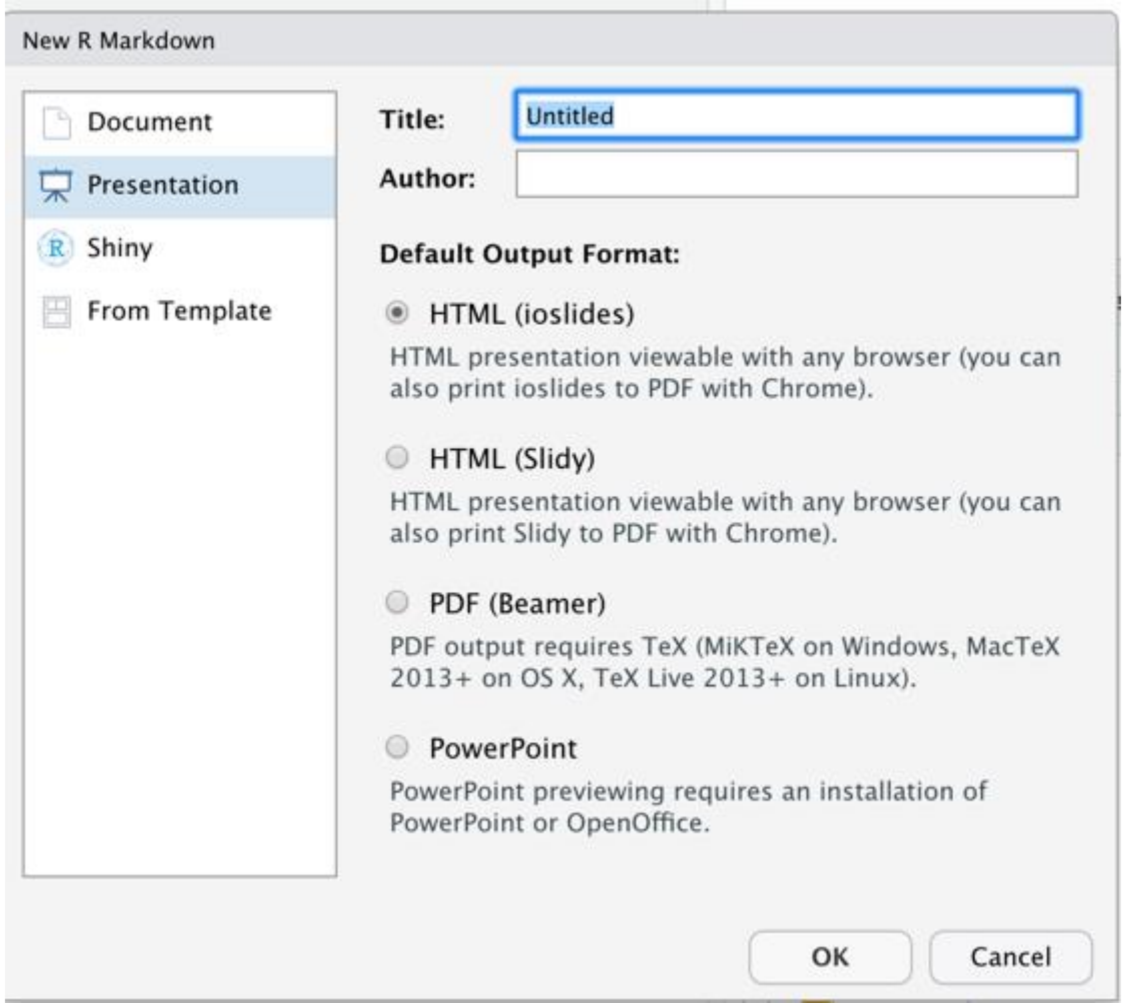
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

- 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



Title: Untitled

Author:

Default Output Format:

● HTML (ioslides)

HTML presentation viewable with any browser (you can also print ioslides to PDF with Chrome).

○ HTML (Slidy)

HTML presentation viewable with any browser (you can also print Slidy to PDF with Chrome).

○ PDF (Beamer)

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

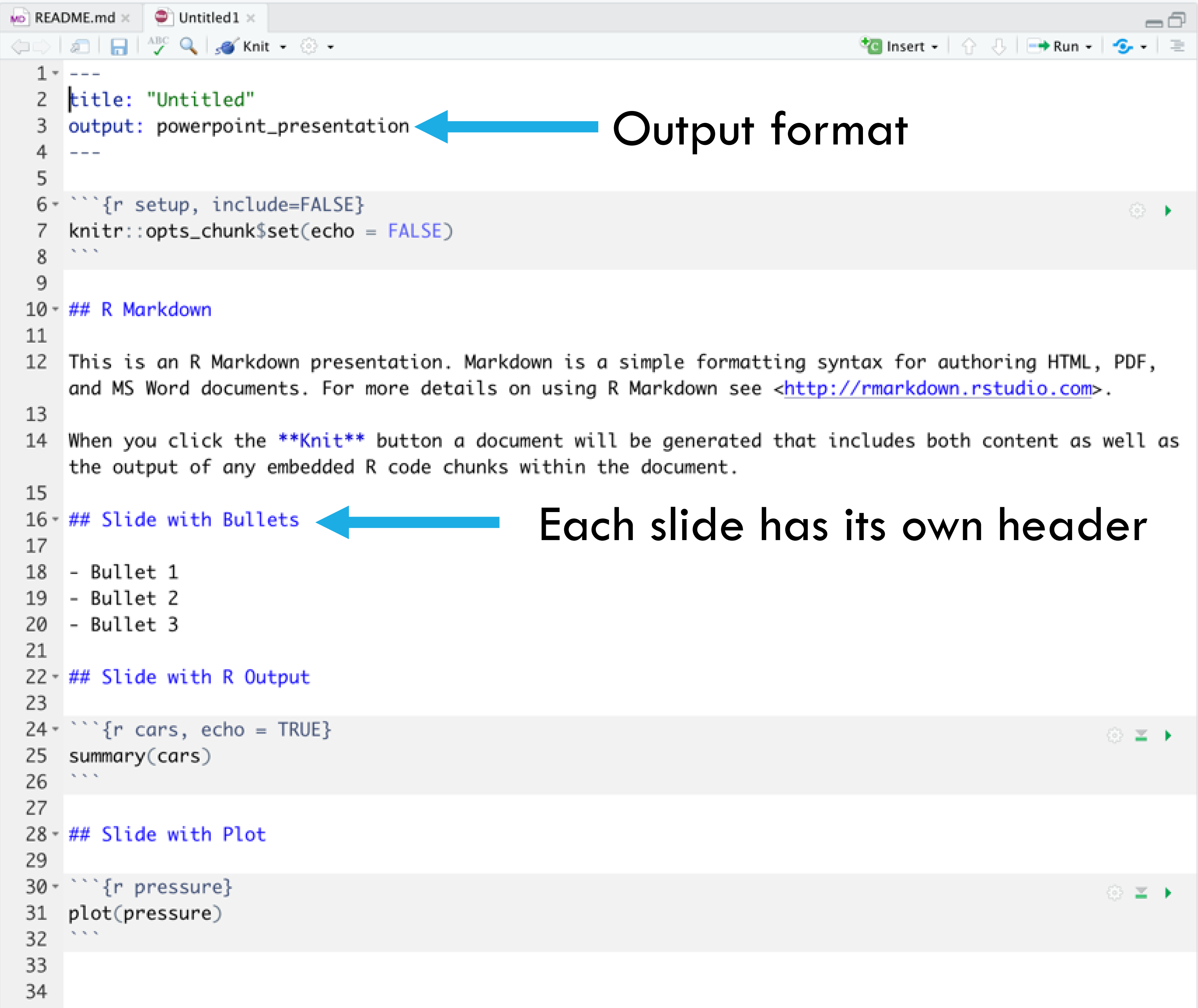
○ PowerPoint

PowerPoint previewing requires an installation of PowerPoint or OpenOffice.

Multiple HTML formats create webpage that’s advanceable like slides

PDF presentation uses LaTeX in the background

Powerpoint produces simple slides



~~~
---
title: "Untitled"
output: powerpoint_presentation
---

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

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.

Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

Slide with R Output

```{r cars, echo = TRUE}
summary(cars)
```

Slide with Plot

```{r pressure}
plot(pressure)
```
```

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

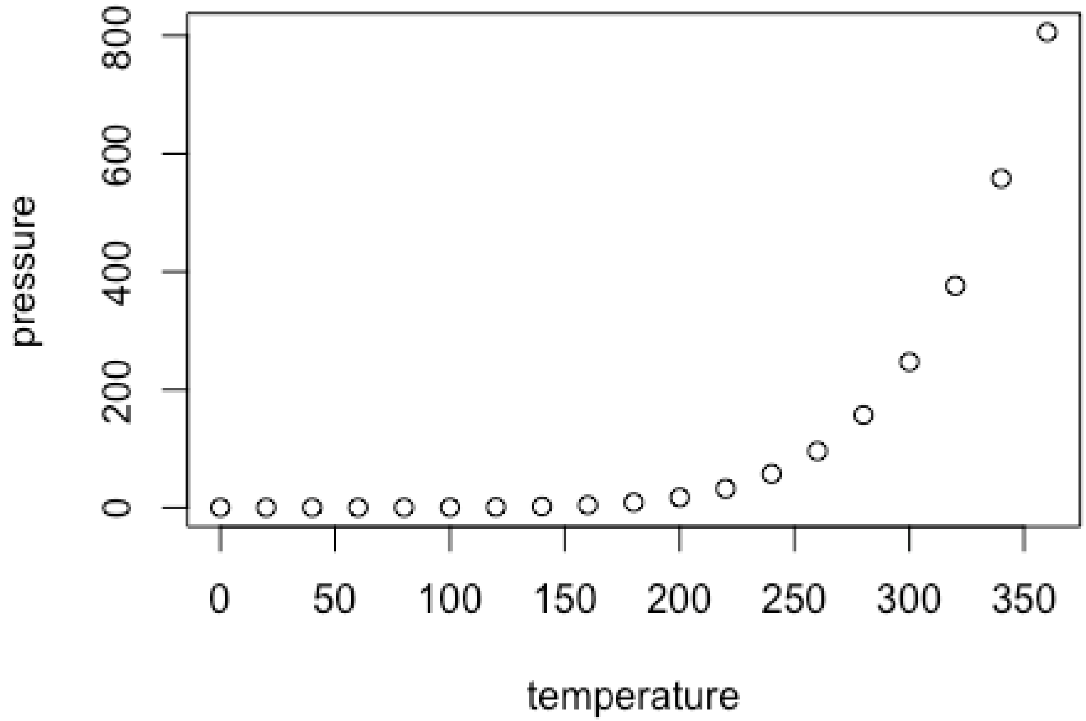
Example output slide

# 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



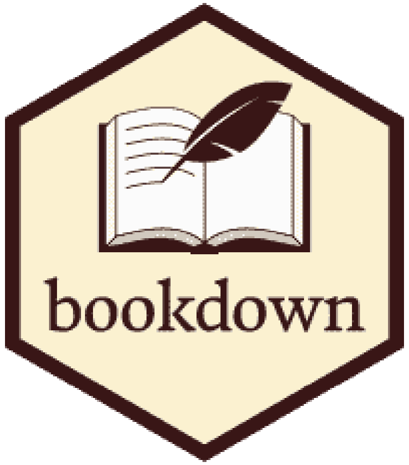
Example output slide

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