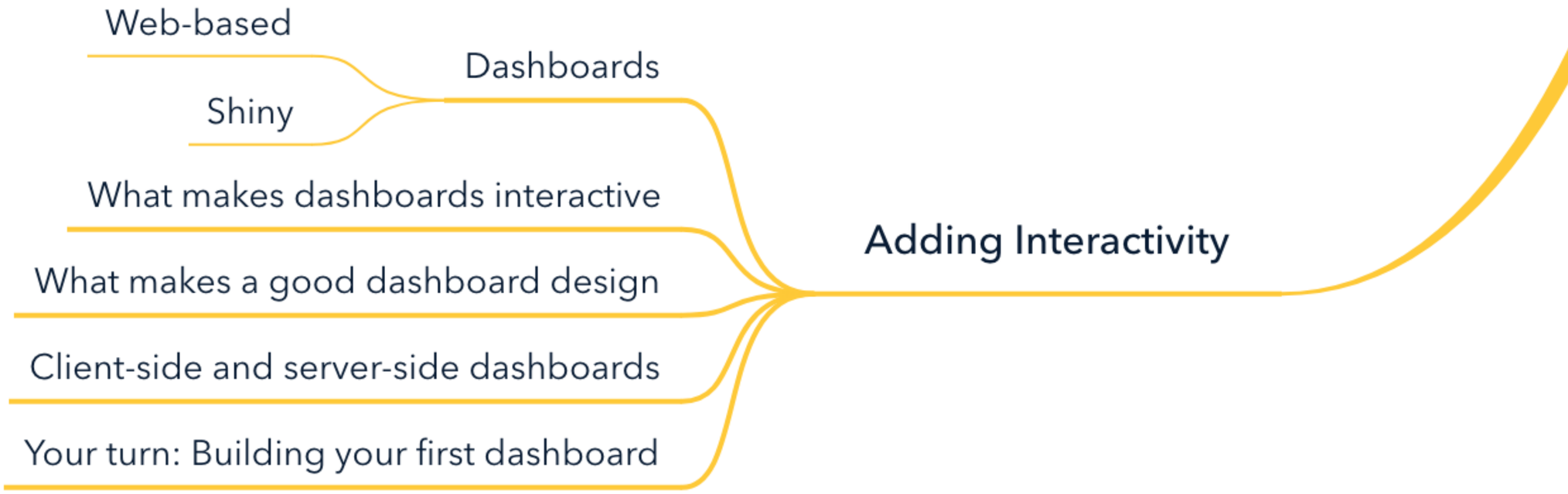
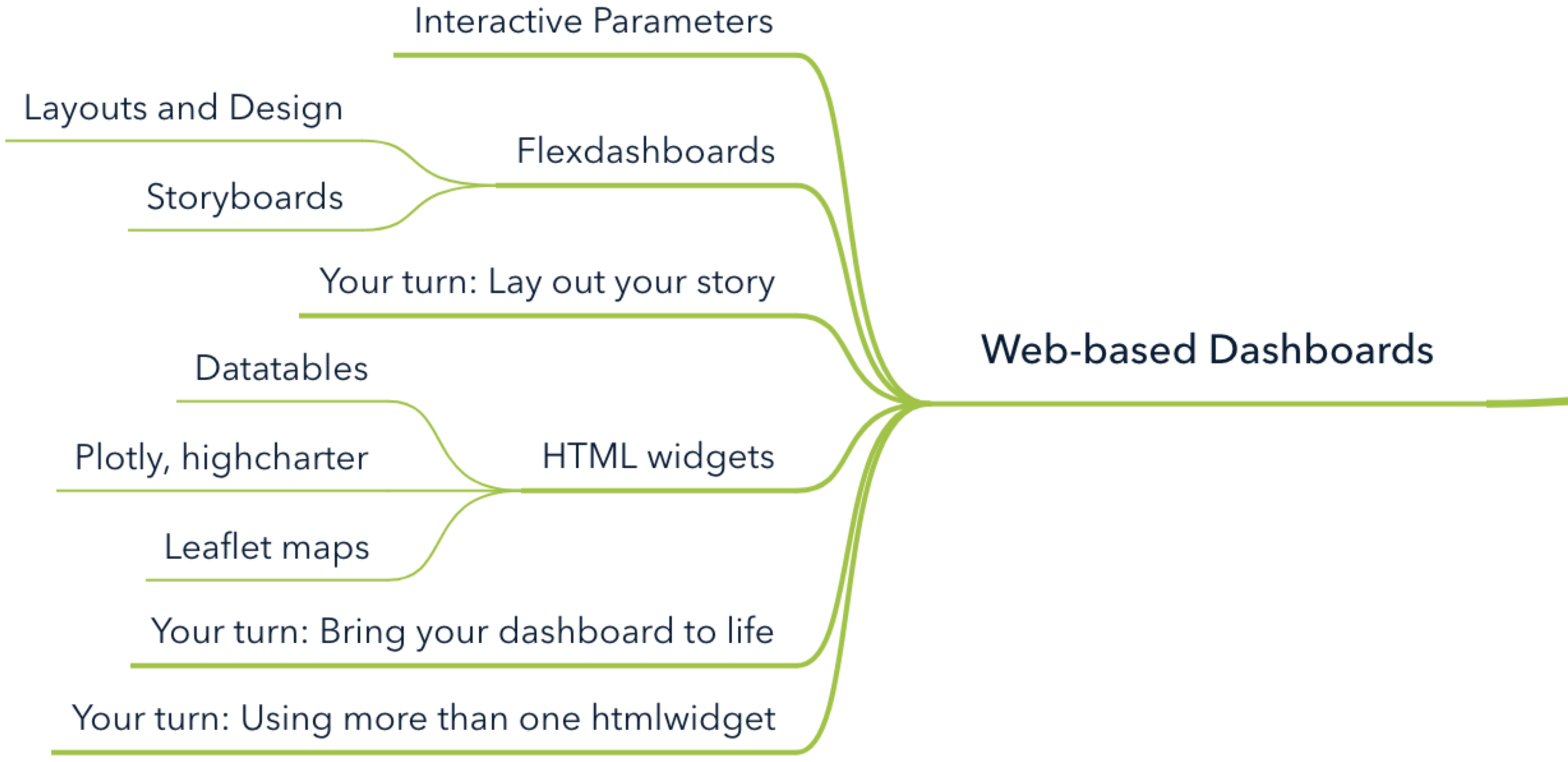

Adding Interactivity
Dashboards
Web-based
Shiny
What makes dashboards interactive
What makes a good dashboard design
Client-side and server-side dashboards
Your turn: Building your first dashboard


Web-based Dashboards
Interactive Parameters
Flexdashboards
Layouts and Design
Storyboards
Your turn: Lay out your story
HTML widgets
Datatables
Plotly, highcharter
Leaflet maps
Your turn: Bring your dashboard to life
Your turn: Using more than one htmlwidget

RStudio Education

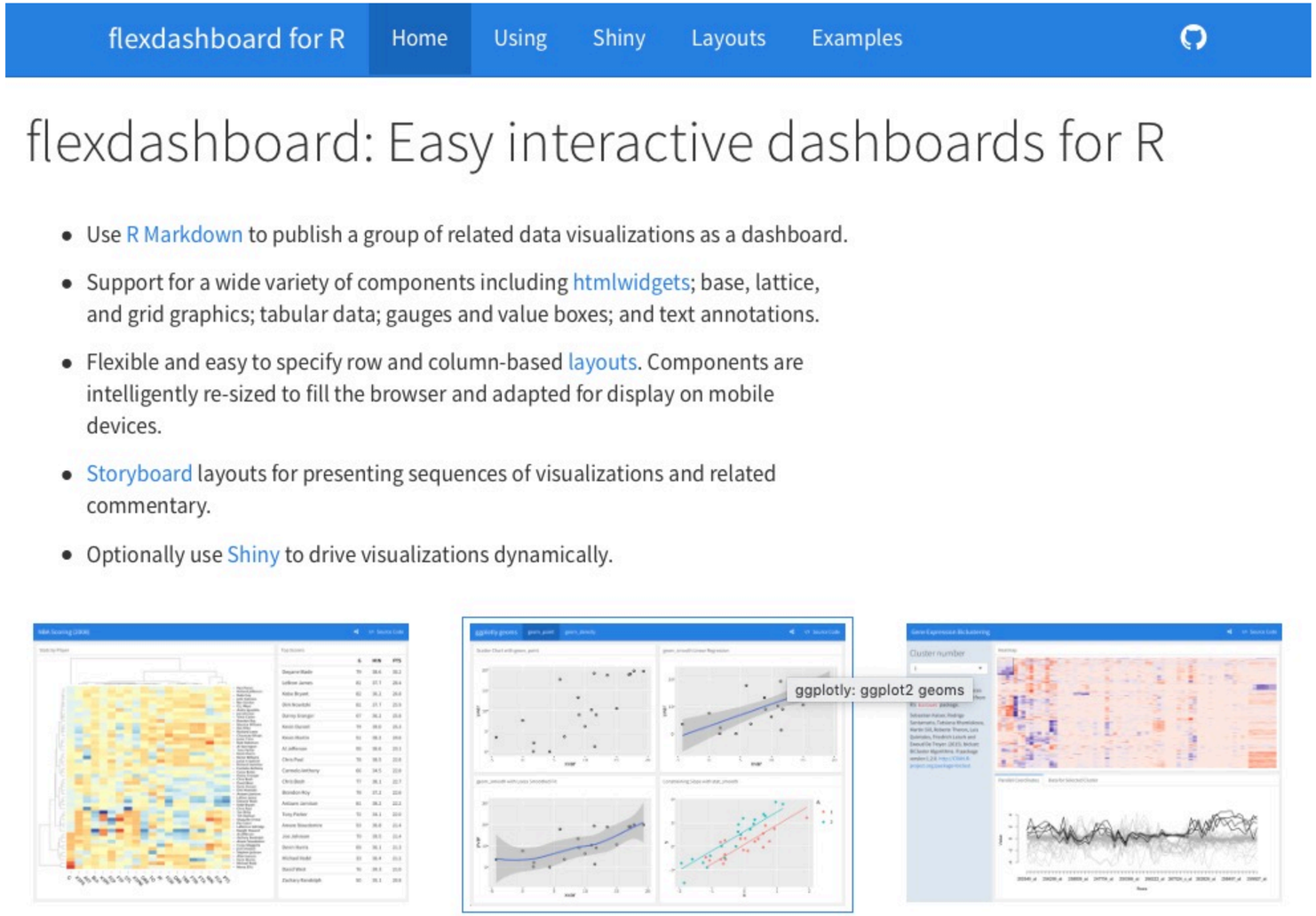
education.rstudio.com

R Markdown and Interactive Dashboards
Adding Interactivity

rstd.io/RMAID

FLEXDASHBOARD REFERENCE

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



rstd.io/RMAID

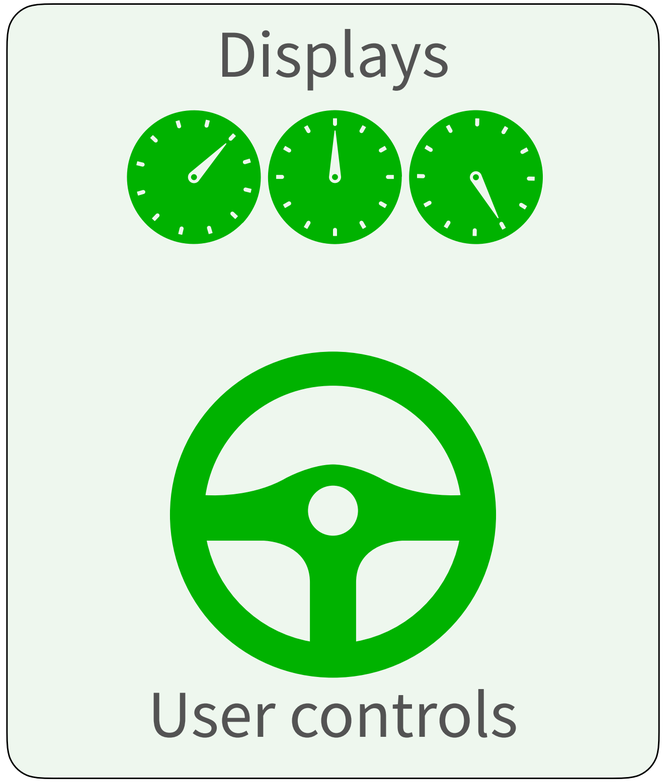
Dashboard Components

What are the common building blocks of a dashboard?

Hint: the fact that these two dashboards look different is also a building block.

05:00

Dashboards Are Composed Of...



Layout

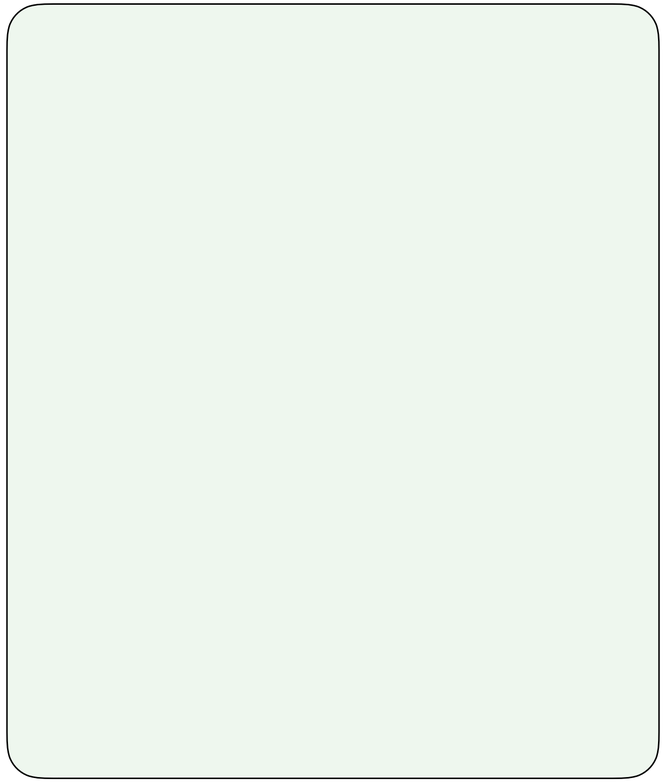
Dashboards have three building blocks:

1. Controls

2. Displays

3. Layouts

Let's Start With...



Layout

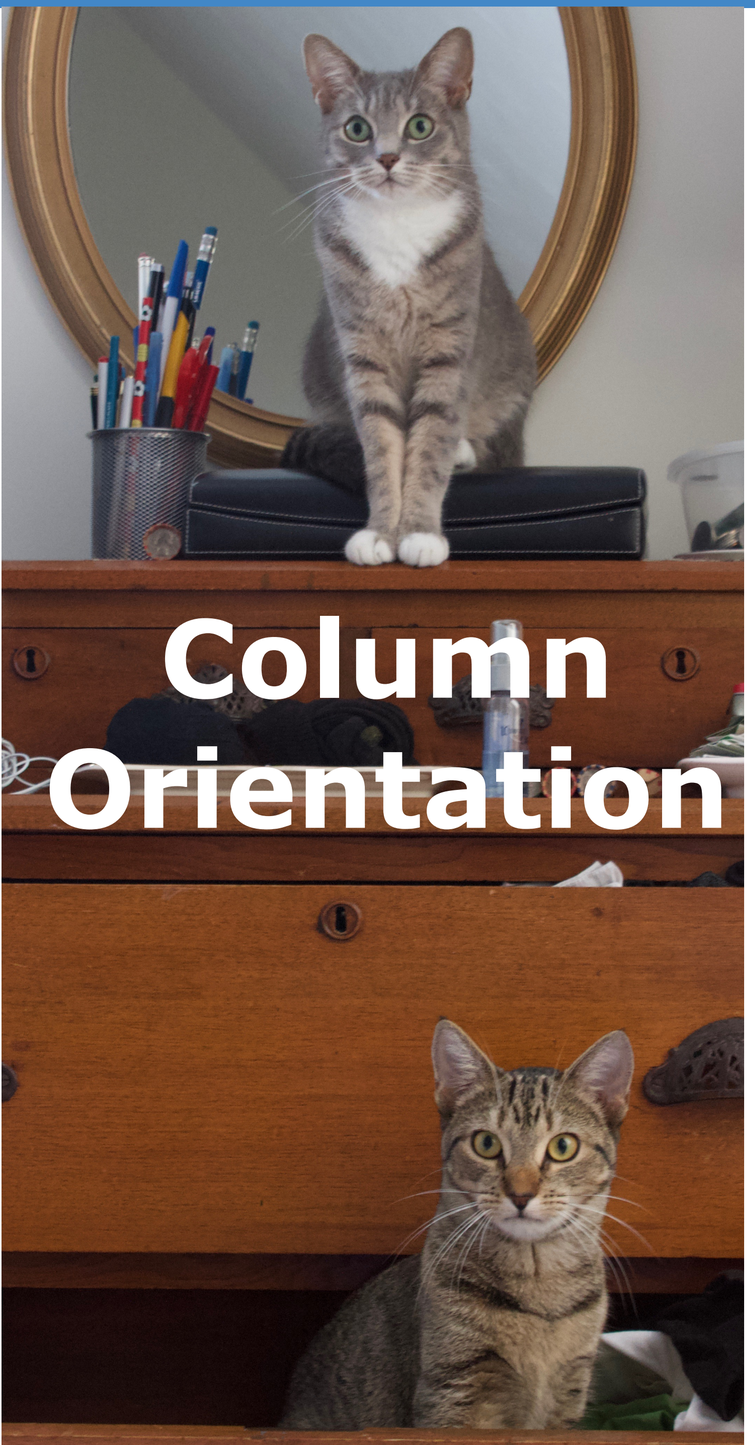
Despite Standardization, Car Dashboard Design Is Essential To The Product

rstd.io/RMAID

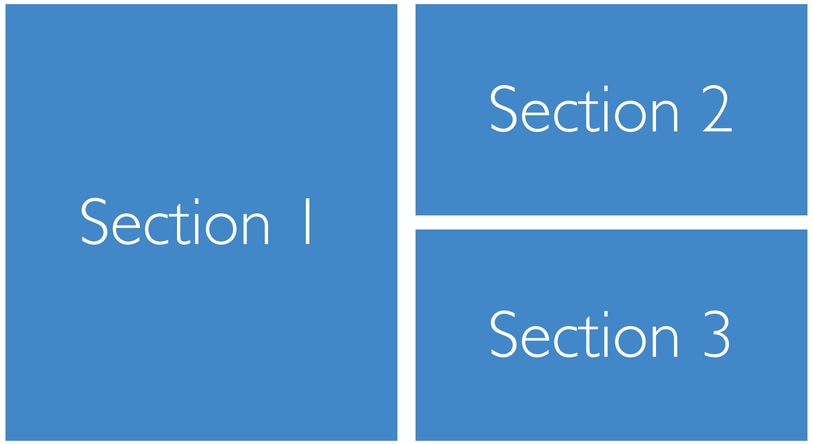
R Markdown layouts use header information to mark rows and columns on pages

Your First Decision: Row Or Column Orientation?

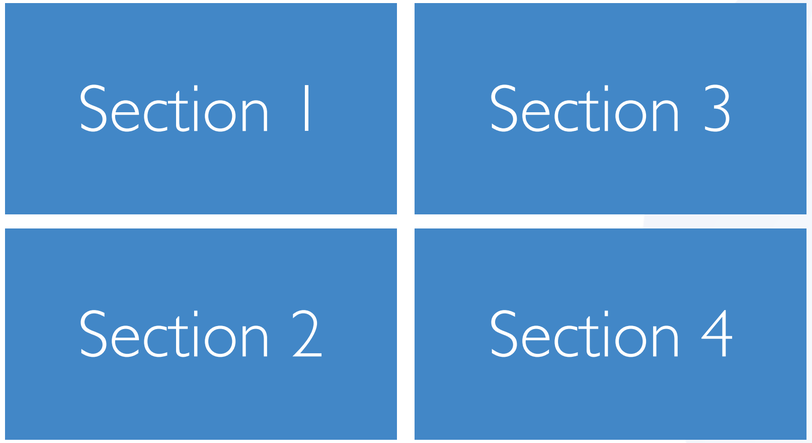
- Our layout maps the one-dimentional R Markdown code onto a grid.
- Your `orientation` YAML parameter determines whether you want your level 2 sections to be laid out on the grid according to columns or rows.



Column Orientation



or



rstd.io/RMAID

Row Orientation

Section 1 | Section 2

Section 3 | Section 4

Section 1

Section 2 | Section 3

rstd.io/RMAID

Layout Interpretations

R Markdown Symbol	Alternate Symbol	`flexdashboard orientation: rows`	`flexdashboard orientation: columns`
#	===	Top level page	Top level page
##	----	New row	New column
###		New column in row	New row in column

RStudio Cloud

EXERCISE 41

Parameterize A Report

1.Open the project **04-Web-Based-Dashboards**
2. Click on **`41-layout-column-orientation.Rmd`** in the Files pane to open that file
3. Knit the file and observe the result
4. Change the YAML line **`orientation: columns`** to **`orientation: rows`** and re-knit.
5. Try adding a **`### Text Description`** row at the bottom of column 1 and putting some text below it. Knit the result. Try improving the esthetics of the result by adding **`{row-height=250}`** to **`### Text Description.`** Play with the row-height value for the best look to your eyes.
6. Add the text **`{.tabset .tabset-fade}`** to the end of the text ## Column 2 (which is actually row 2 now). What has changed?

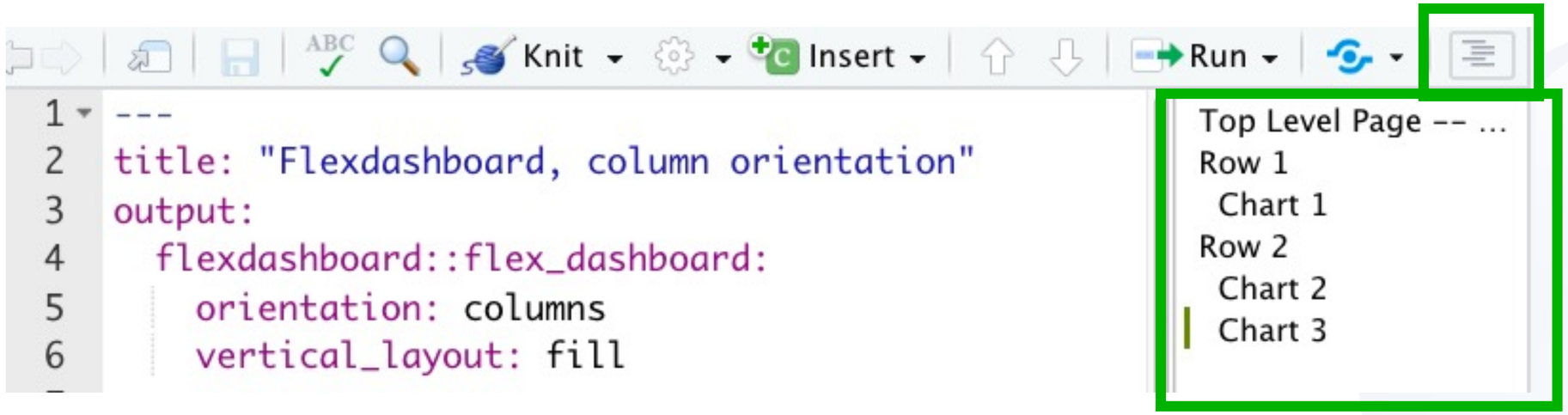
10m 00s

OBSERVATIONS

- Layout documentation is at

https://rmarkdown.rstudio.com/flexdashboard/layouts.html#overview

- You may find the RStudio IDE outline button useful for keeping track of your layout hierarchy.



OBSERVATIONS (CONTINUED)

- You can create multiple pages by adding more level 1 headers (`# header`).
- Column and row orientations are local to each page. You change the orientation by appending `{data-orientation=rows}` or `{data-orientation=columns}` to your level 1 header
- Level 4 headers or more don't do anything in dashboards

Storyboards Allow More Text In your Layout

- This layout allows you to tell a story through a sequence of graphs with text descriptions
- The R Markdown syntax doesn't really make sense. You identify Frames with level 3 headers, and then you add commentary after ***
- You can't use Level 1 or Level 2 headers with Storyboards

RStudio Cloud

Exercise 42

Try a Storybook Dashboard

Knit the `42-storyboard-layout.Rmd`
How is this layout different from the grid layouts?

3m 00s

Dashboards Are Composed Of...

Displays

HTML widgets create display interactivity using the user's browser

HTML Widgets Create Display Interactivity

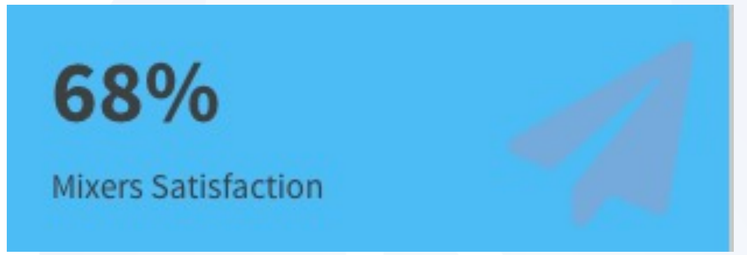
- HTML Widgets create interactivity within a Flexdashboard layout pane.
- See all 106 submitted HTML Widgets at: http://gallery.htmlwidgets.org
- You can develop your own if you know Javascript.
- We'll focus on a few:
  - Valueboxes and Gauges for individual values
  - Plotting with plotly
  - Tables with datatable (DT)
  - Maps with leaflet

VALUEBOXES SHOW SINGLE VALUES

- They dress up single values with an icon and a color

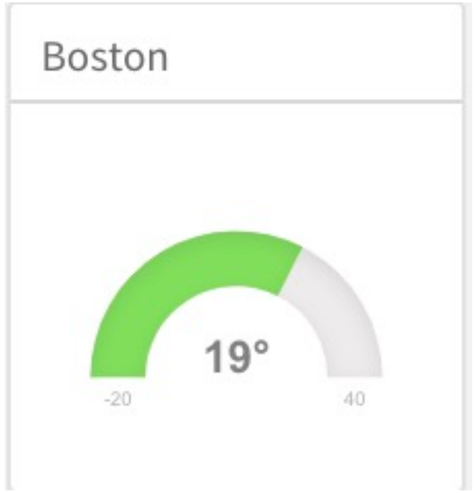
```
valueBox(params$conf_attendees, "Attendees", icon = "fa-users")
```

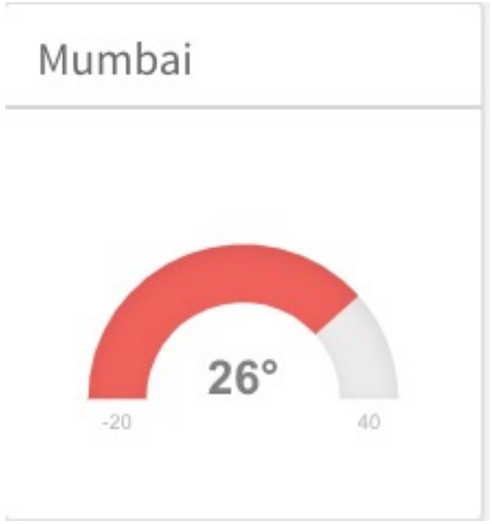

```
valueBox(paste0(mixers_sat_percent, "%"),
         icon = "fa-paper-plane",
         color = ifelse(mixers_sat_percent >= 50,
                        ifelse(mixers_sat_percent >= 75, params$success_color,
                               params$warning_color),
                               params$danger_color))
```



GAUGES SHOW SINGLE VALUES IN CONTEXT

```
gauge(my_weather$temperature, min = gauge_minimum_temp, max =
gauge_maximum_temp, symbol = "°",
      sectors = gaugeSectors(warning = c(gauge_minimum_temp, gauge_cold_temp),
                             success = c(gauge_cold_temp,gauge_hot_temp),
                             danger = c(gauge_hot_temp,gauge_maximum_temp),
      colors = gauge_colors))
```



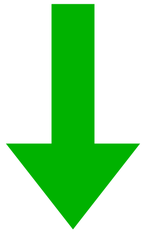


rstd.io/RMAID

Plotting With Plotly

- Plotly allows you to make any ggplot interactive
- Simply assign your ggplot output to a variable and ggplotly that variable

```
ggplot(mtcars, aes(disp, hp, color = cyl)) + geom_point()
```

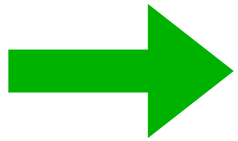


```
g <- ggplot(mtcars, aes(disp, hp, color = cyl)) + geom_point()
ggplotly(g)
```

Tables With Data Table

- Displays tables in a nice paged output
- Allows interactive searching and sorting of a table
- Easy to use: just use DT::dataframe(data_frame) to display your table.
- Gotcha: the data table library is called DT, not datatable.

```
DT::datatable(mtcars)
```



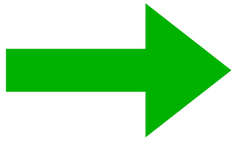
Show 10 entries Search:

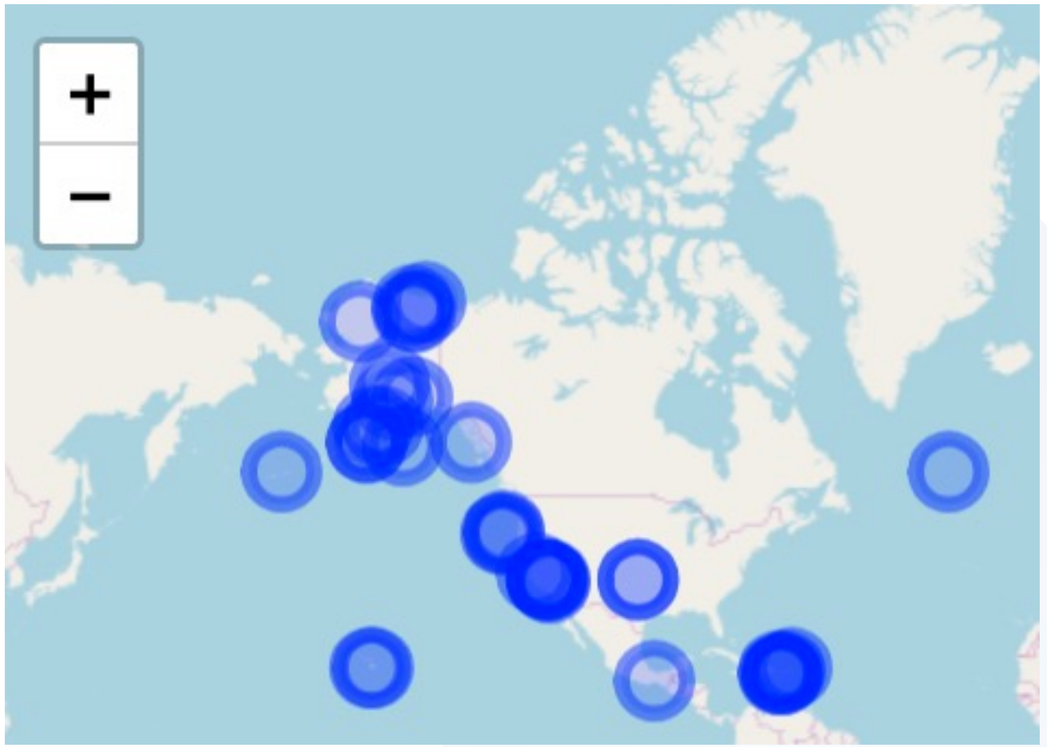
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0

Maps With Leaflet

- An easy way to create maps with overlays
- All you need is a data.table or tibble that has columns lat and lng for latitude and longitude
- leaflet() %>% addTiles() gives you a basic map
- Usually you just pipe in new layers of markers or polygons that show your data

```
leaflet() %>%
  addTiles()  %>%
  addCircleMarkers(data =
     latest_earthquakes)
```





RStudio Cloud

Exercise 43

Build Your Output Dashboard

1. Click on **`43-add-displays.Rmd`** in the Files pane to open that file
2. Convert all the ggplots to **`plotly`** widgets.
3. Change the **`mtcars`** pane to display a datatable
4. Add layers in the Earthquakes Map pane to display **`latest_earthquakes`**.

Hint: make one change at a time and test each one before knitting.

10m 00s

Dashboards Are Composed Of...

User controls

Frankly, We Only Have Two Control Options So Far

- Parameterized reports where you edit the parameters and knit the source
- Parameterized reports where you do a Knit with Parameters

To Build Interactive Controls, We Need

rstd.io/RMAID

We build interactive controls using either kniting with parameters or using Shiny

Summary

- Dashboards consist of 3 components
  - Layouts constructed using R Markdown headers and YAML directives specifying
    - Row orientation
    - Column orientation
    - Storyboard panes
  - Displays built from R code and HTML widgets
    - Valueboxes
    - Plotly
    - Datatable
    - Leaflet
  - Interactive controls which need Shiny or some other mechanism

rstd.io/RMAID