{flexdashboards}

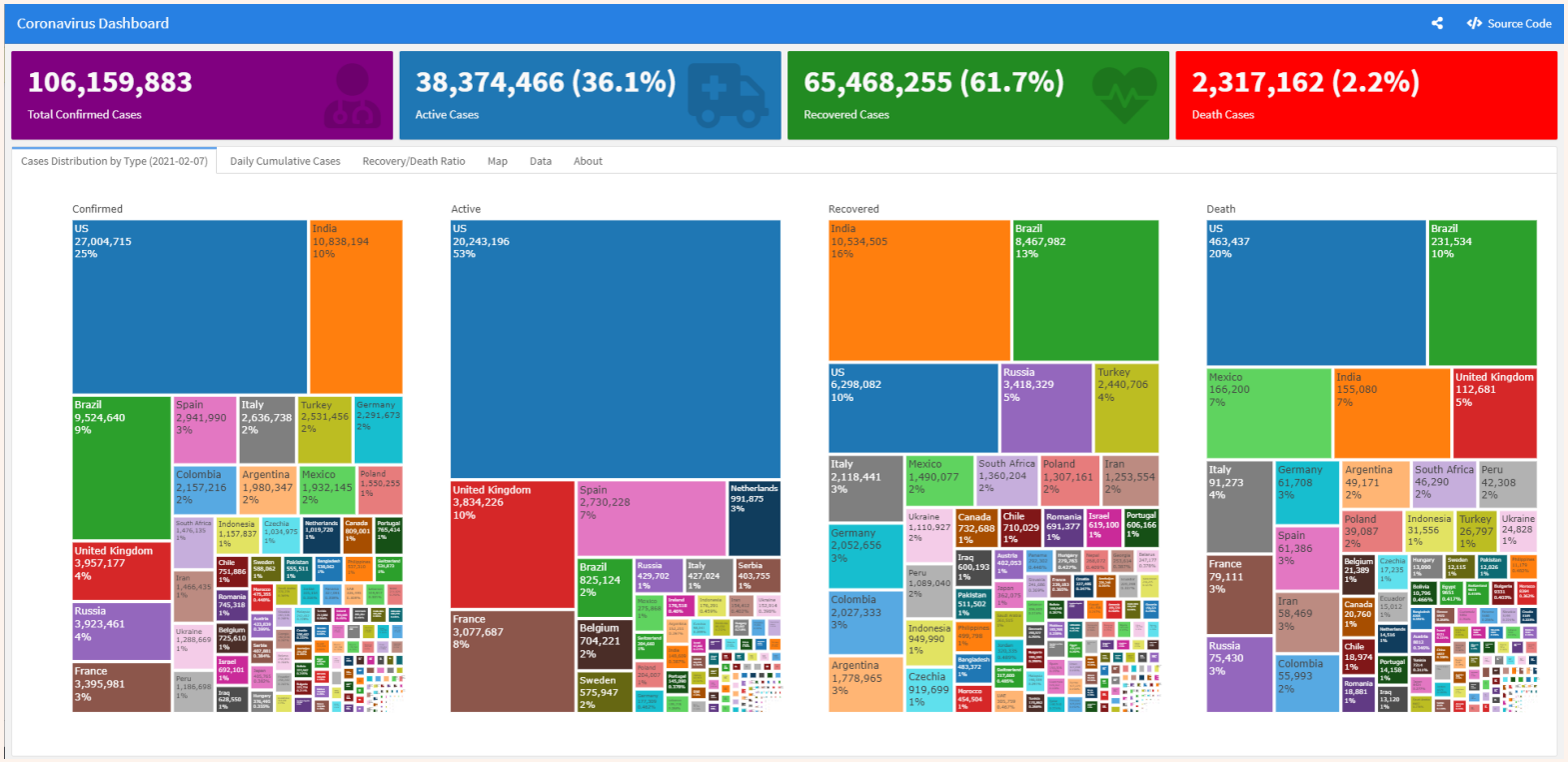
Data visuals. On a board.

Akhila Nekkanti

Winter 2021

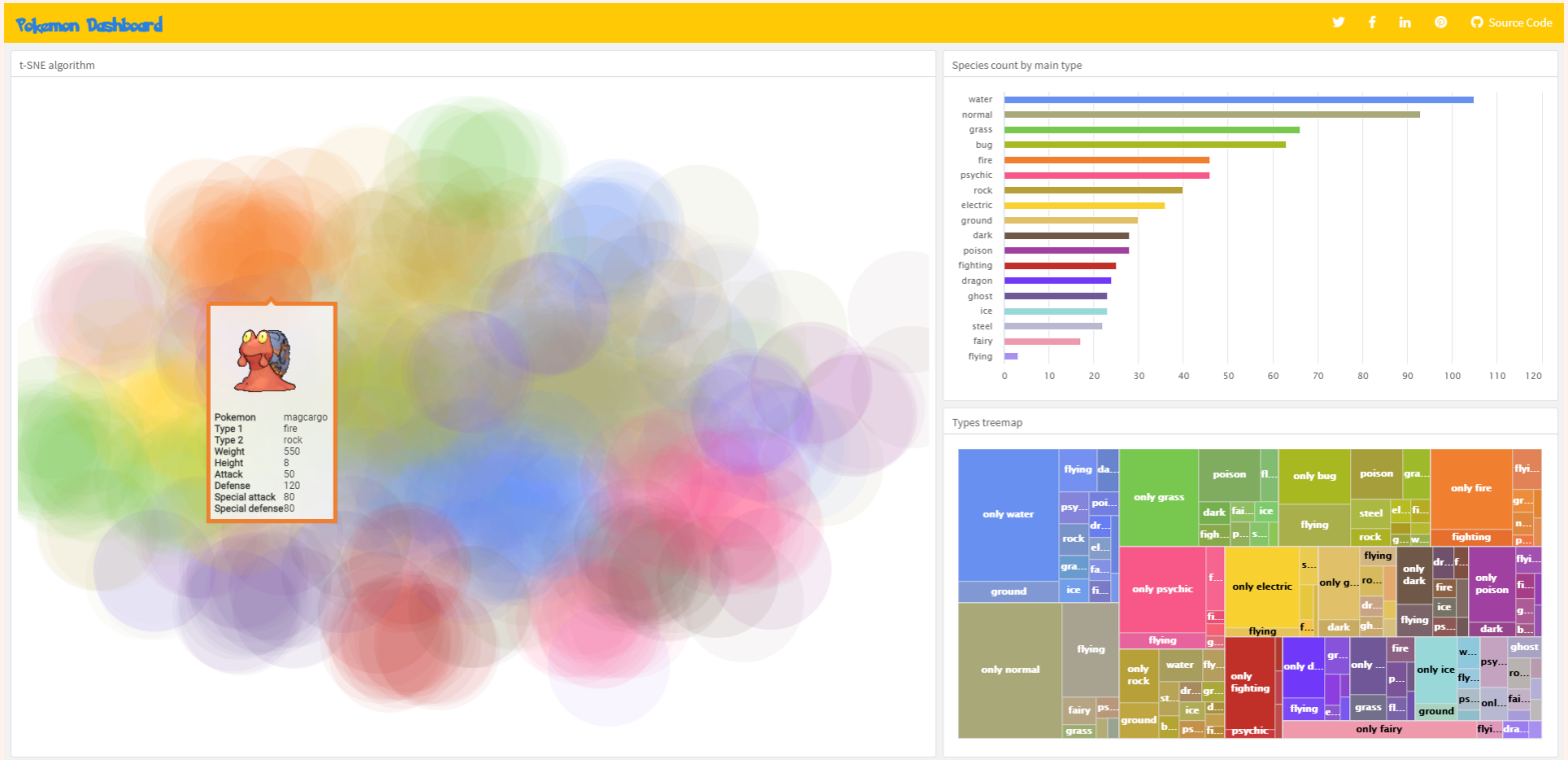
Why use a dashboard?

https://ramikrispin.github.io/coronavirus_dashboard/



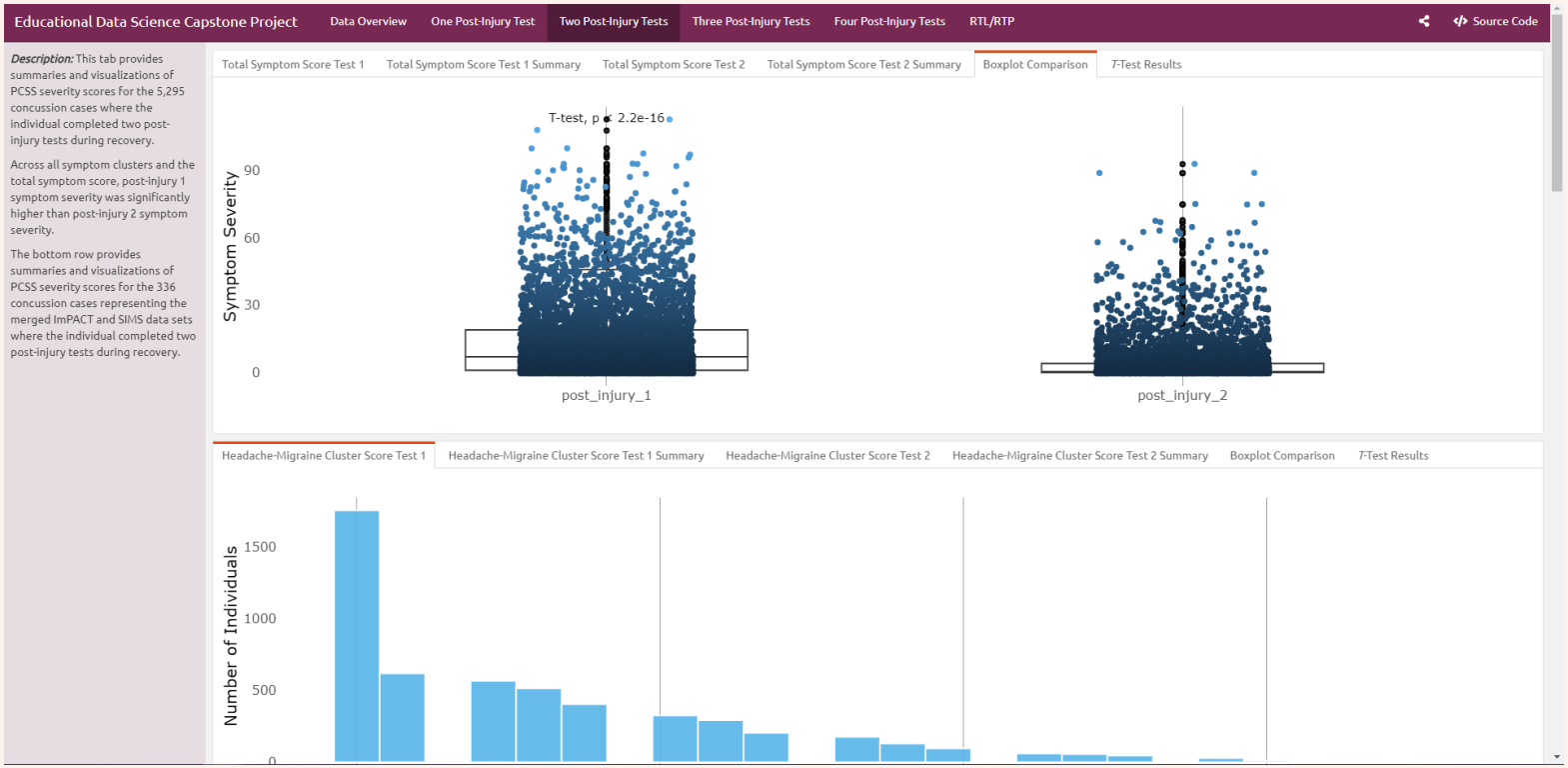
Why use a dashboard?

http://jkunst.com/flexdashboard-highcharter-examples/pokemon/



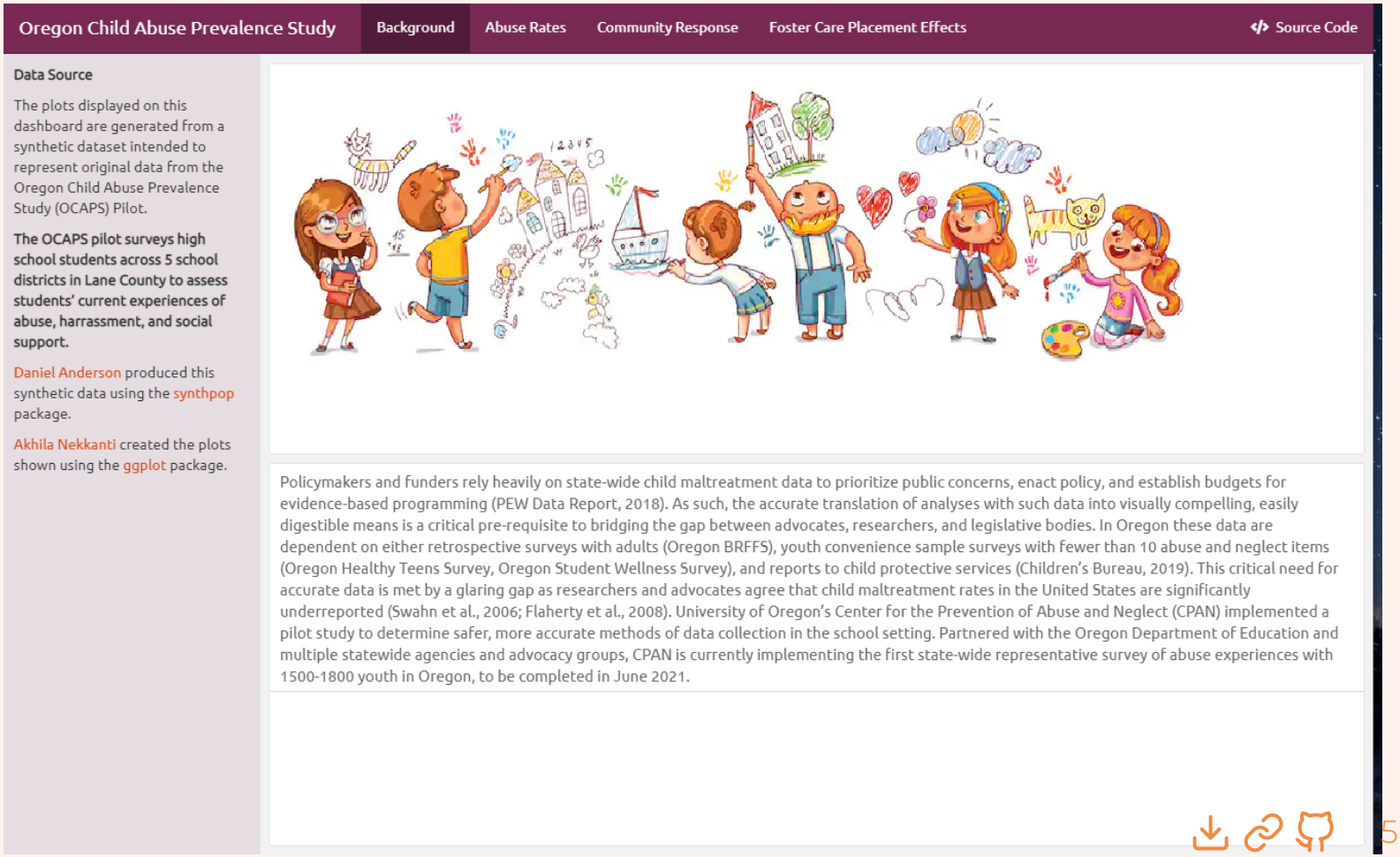
Why use a dashboard?

https://jim-wright90.github.io/data_sci_capstone_project/



Why use a dashboard?

https://github.com/akhilanekkanti1/FlexDashboards



The ultimate source

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

Where to start?

Workflow!

We want to make sure we have the *option* to make our dashboard public.

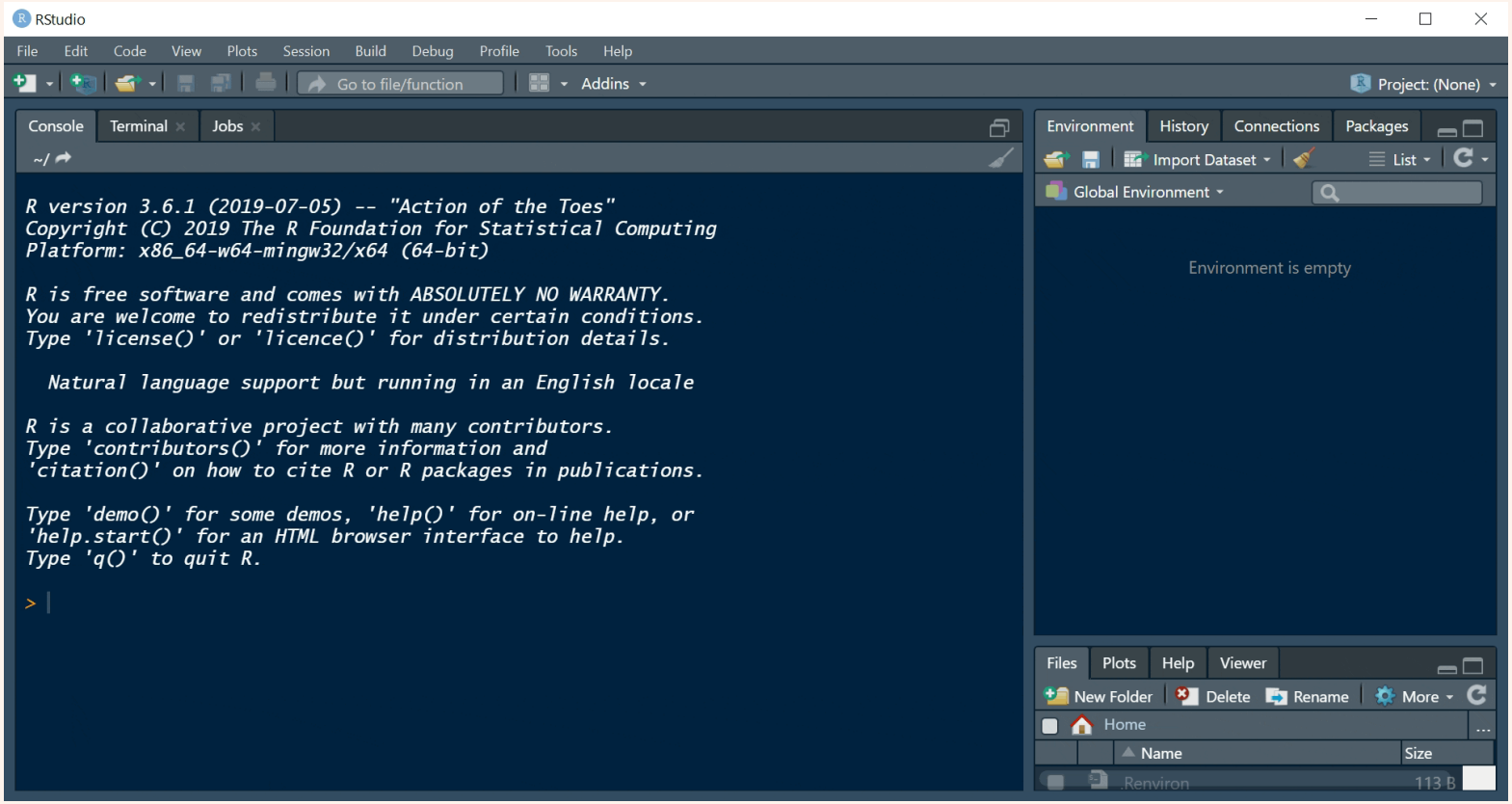
1. Create a new Project
2. Create a new Rmarkdown from template {flexdashboards}
3. Knit right away
4. Create a github Repo
5. Link your local and remote repos with GitKraken

Mise en place: Set yourself up.

- R
- GitKraken
- Github

1. Create a new Project

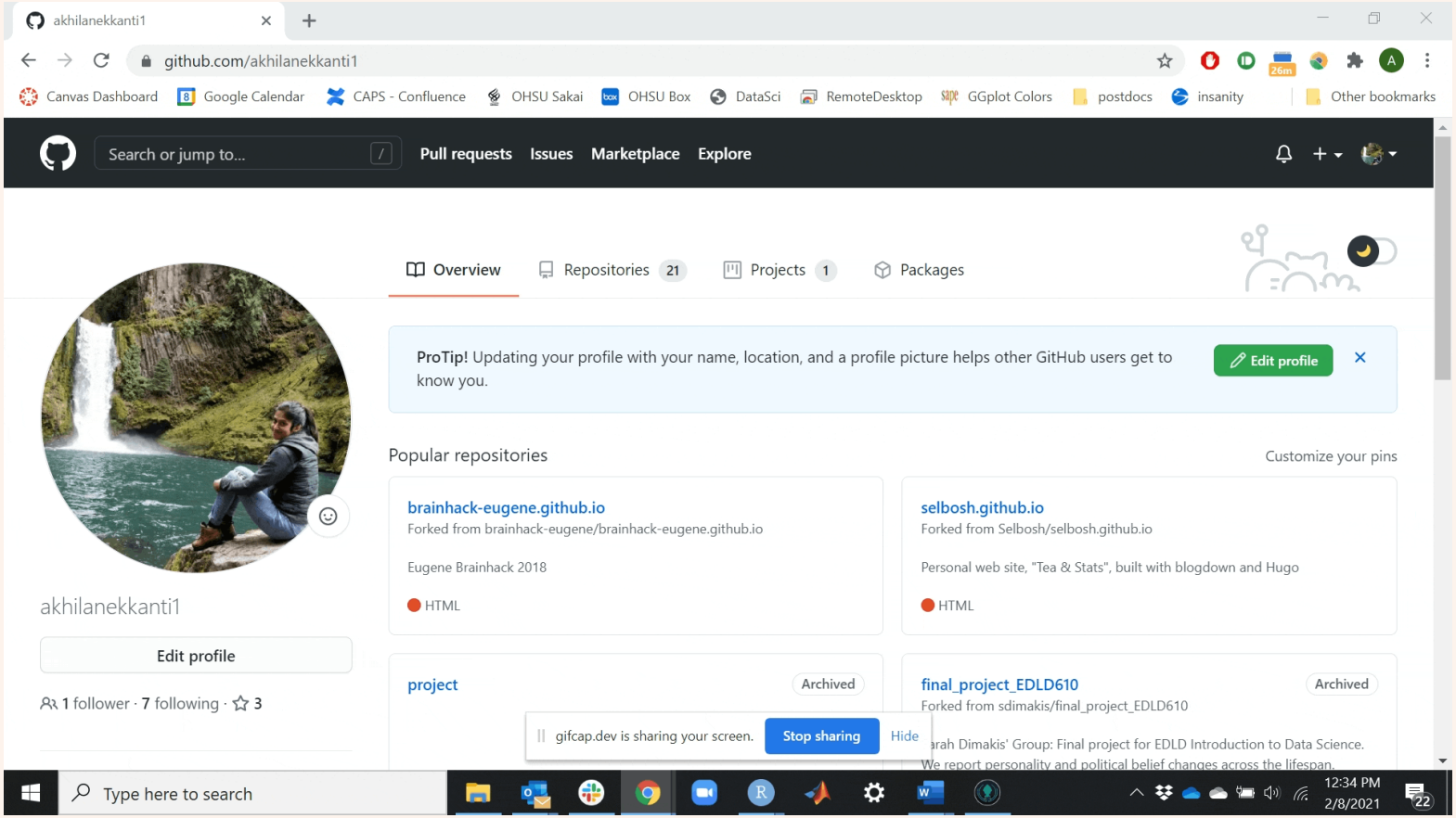
File > New Project > New Directory



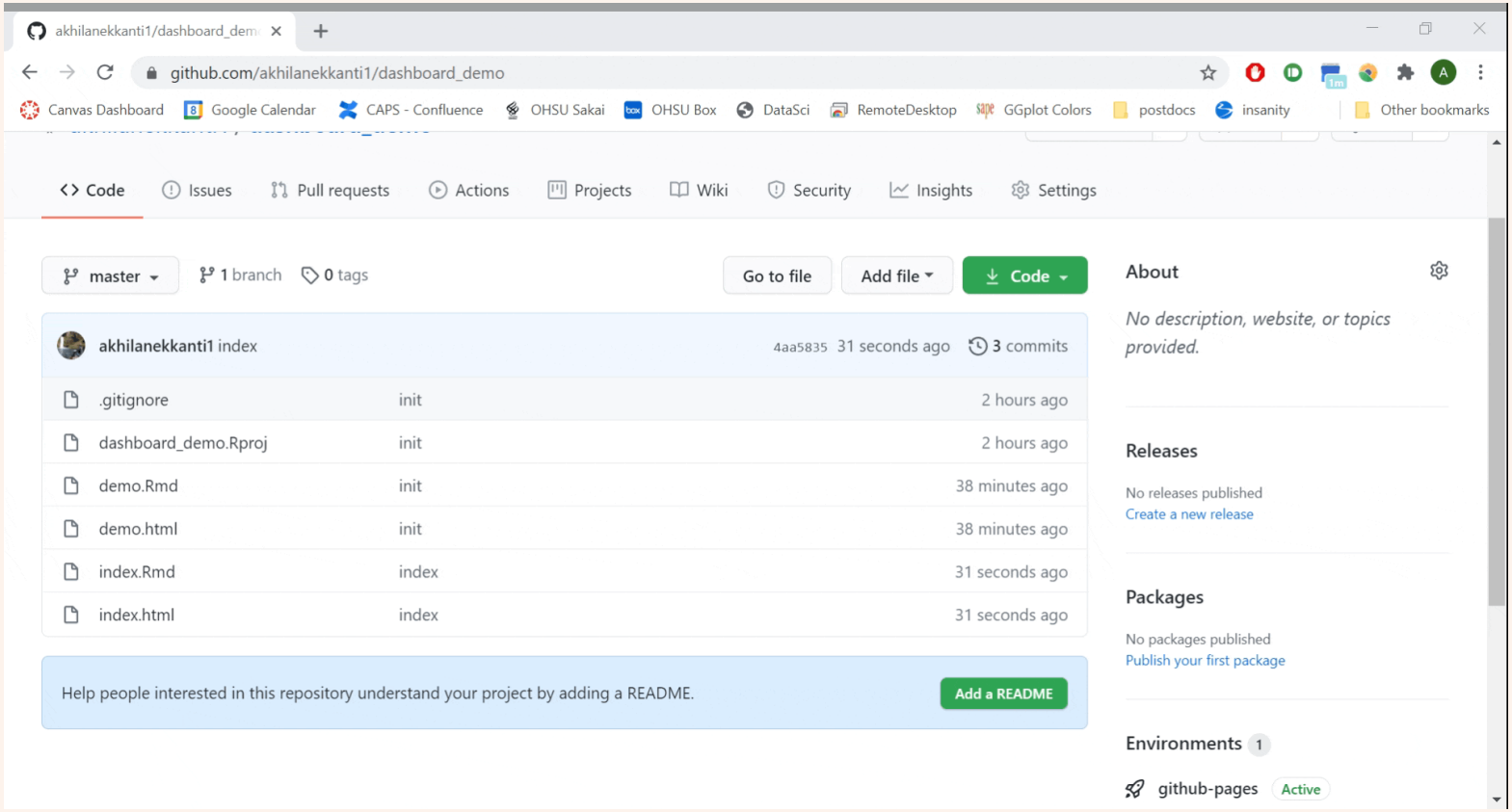
2. Create a new Rmarkdown from template and (#3) Knit

```
#One time only
install.packages("flexdashboard")
```

4. Create a GitHub repo and (#5) Link it with your local



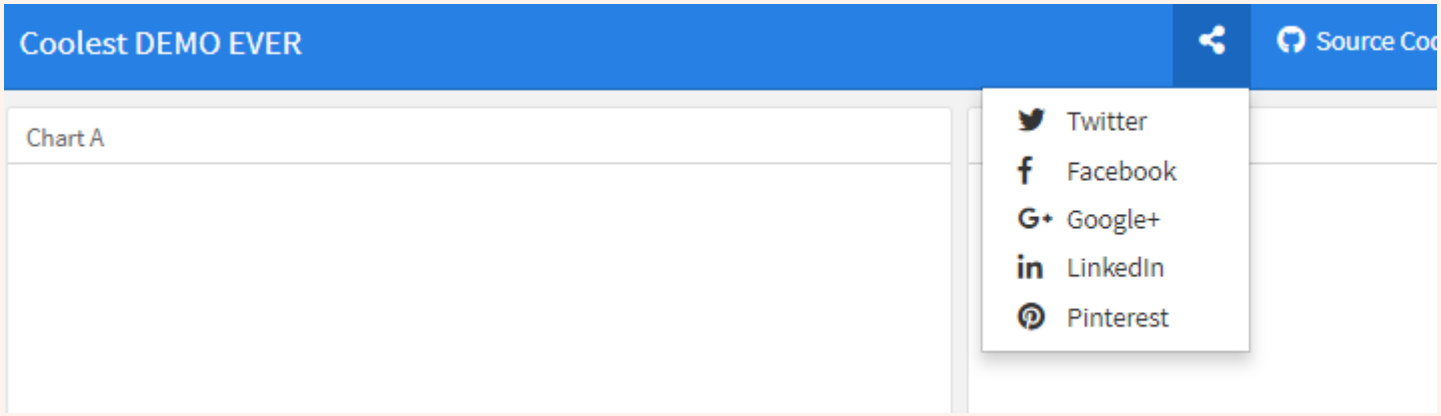
Ready to PUBLISH!



Now let's play around with our dashboard!

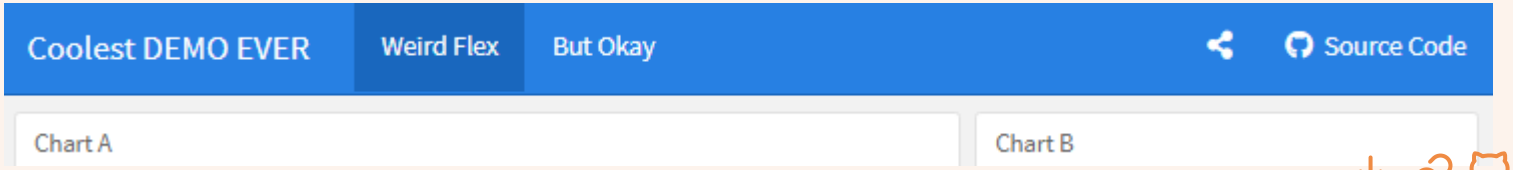
Let's start with the YAML.

```
---
title: "Coolest DEMO EVER"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
    social: menu
    source_code: https://github.com/akhilanekkanti1/dashboard_demo
---
```



Let's add a new page.

```
Weird Flex
==========================

Column {data-width=650}
-----------------------------------------------------------------------

### Chart A
< r code chunk >

Column {data-width=350}
-----------------------------------------------------------------------

### Chart B
< r code chunk >

But Okay
========================
### Chart C
< r code chunk >
```



Play around!

- Change the name of each square
- Try adding a different plot in each square
- Can you change the number of squares in each page?
- Can you change the size of each square?

Plot example:

```
library(tidyverse)

mtcars %>%
  ggplot(aes(wt, mpg, col=as_factor(cyl))) +
  geom_point() +
  facet_grid(~cyl)
```

5 minutes :)

Want to add some text?

Just add it in like you would for a typical Rmarkdown doc.

```
# Page 1

Check out this plot I made.

### Plot A

<R code for your plot>
```

Columns

- Create a new column with

```
Column
--------------------------------------
```

- Change the width of each column
  - Notice there are NO spaces around the '='

```
Column {data-width=400}
--------------------------------------
```

- Add a new square within each column

```
Column {data-width=400}
--------------------------------------
### New Square
```

Columns vs. Rows

- If you want all rows:

```
---
title: "Coolest DEMO EVER"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
    vertical_layout: fill
    social: menu
    source_code: https://github.com/akhilanekkanti1/dashboard_demo
---
```

Columns vs. Rows

- If you want rows for some pages and columns for other pages:
  - don't specify the orientation
  - do add 'data-orientation' when you name your pages.

In this example, Page 1 will have rows and Page 2 will have columns.

```
#change your yaml
---
title: "Coolest DEMO EVER"
output:
  flexdashboard::flex_dashboard:
    vertical_layout: fill
    horizontal_layout: fill
    social: menu
    source_code: https://github.com/akhilanekkanti1/dashboard_demo
---


Page 1 {data-orientation=rows}
==============================


Page 2
==============================
```

Rows

- Create a new row with

```
Row
-------------------------------------
```

- Change the width of each row
  - Notice there are NO spaces around the '='

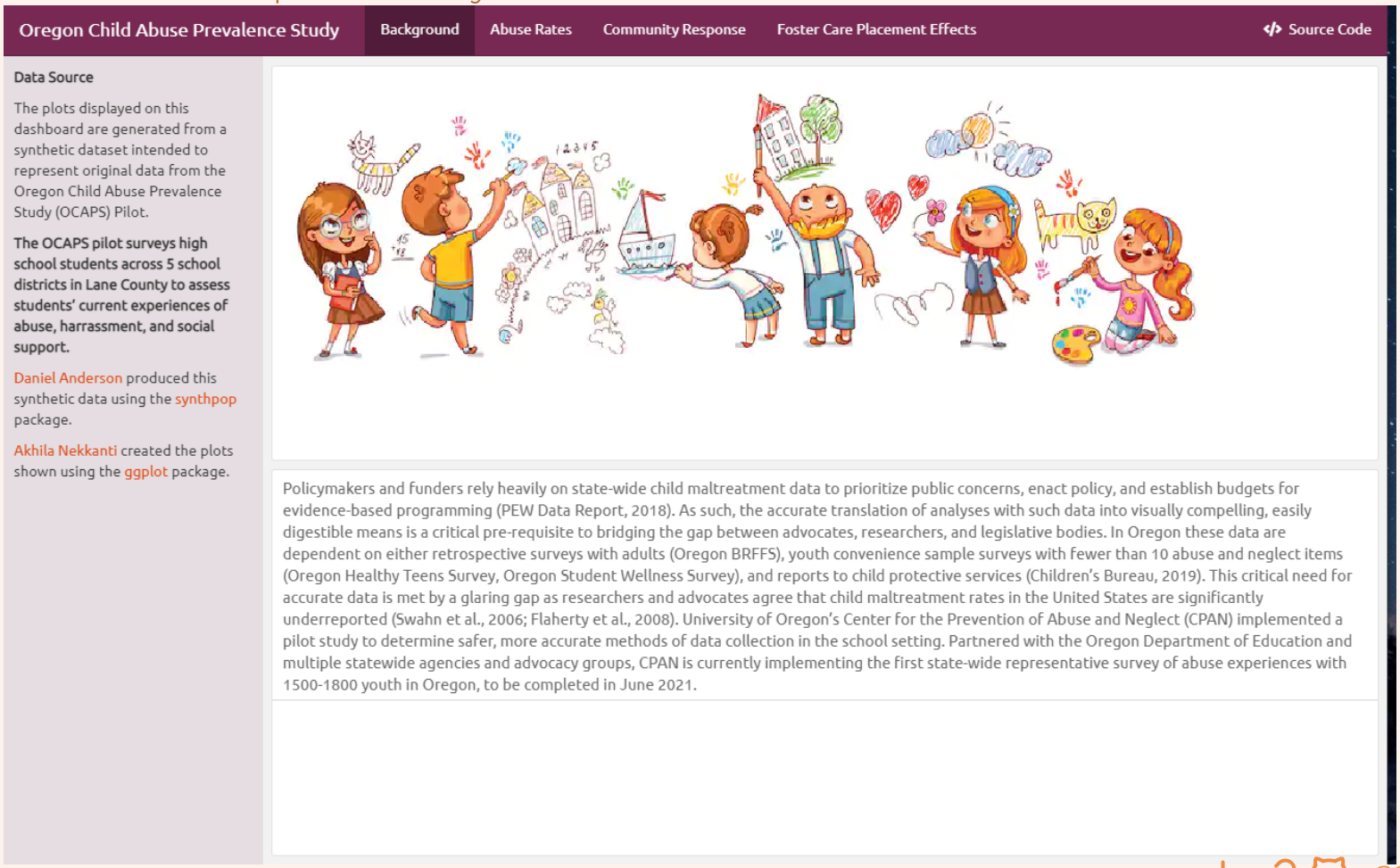
```
Row {data-height=400}
-------------------------------------
```

- Add a new square within each row

```
Row {data-height=400}
-------------------------------------
### New Square
```

You can also add a sidebar!

Here's an example from my dashboard.



You can also add a sidebar!

If you want the sidebar on all pages:

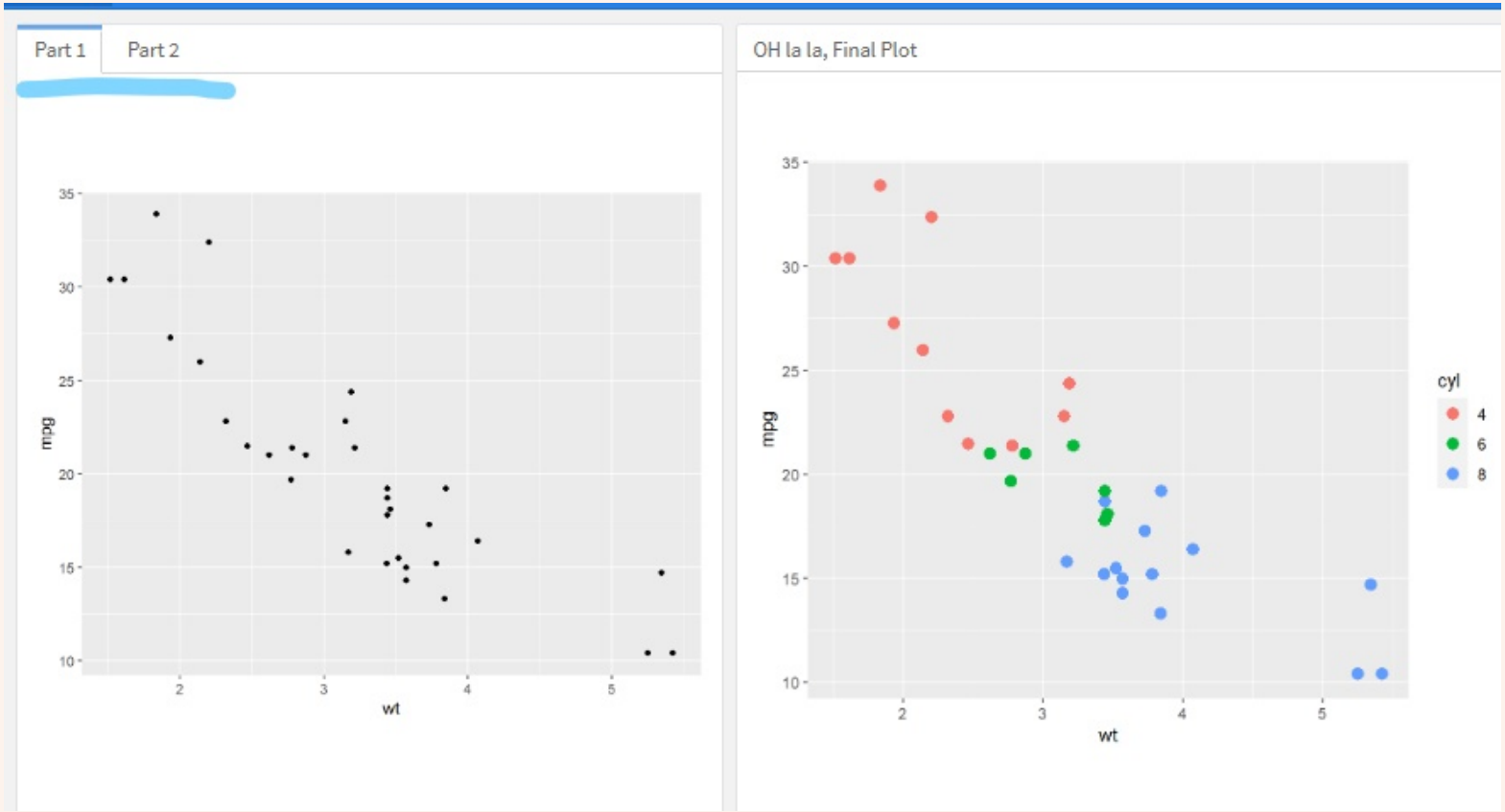
```
Sidebar Title {.sidebar}
===================================

This Dashboard is Important.

RMarkdown syntax works here!

[Click here](https://youtu.be/dQw4w9WgXcQ).
```

If you want the sidebar on just one page:

```
Sidebar Title {.sidebar}
-----------------------------
```

What if you want to organize your columns even more?

You can use tabs:



Use tabset to split your columns or rows into tabs.

```
Column {.tabset}
-------------------------------------

### Part 1

<r code chunk>

### Part 2

<r code chunk>


Column {data-width=650}
-----------------------------------------------------------------------

### OH la la, Final Plot
```

For multiple arguments, there's no comma in between.

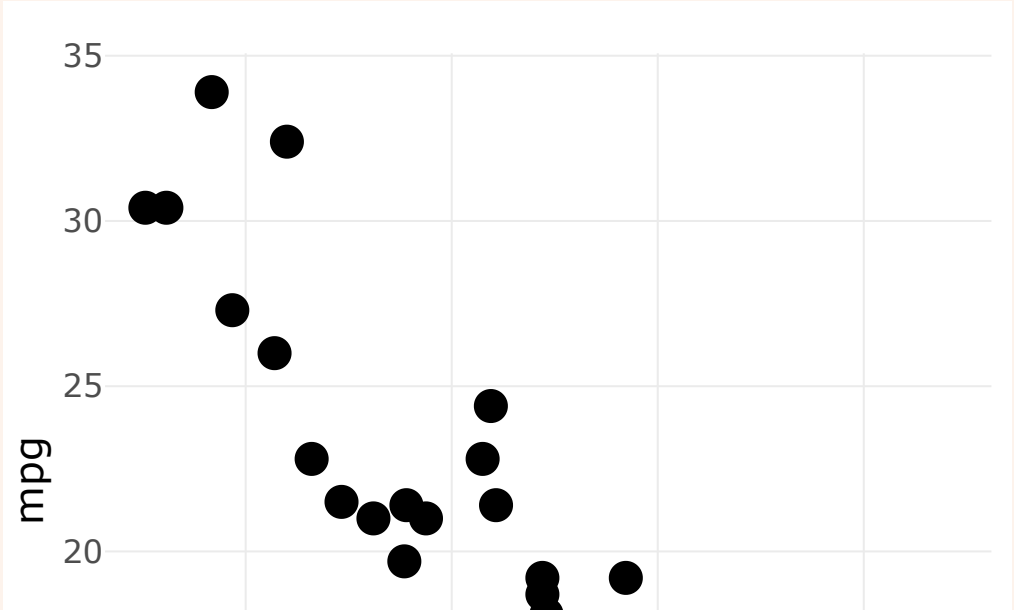
```
Column {.tabset data-width=350}
-------------------------------------
```

Interactivity!

You can use {reactable} or {plotly} to make your tables or plots interactive!

Here's a simple example:

```
p <- mtcars %>%
  ggplot(aes(wt, mpg)) +
  geom_point()

plotly::ggplotly(p)
```



Interactivity!

If you have more than one page, make it *shiny*!

{shiny} allows you to make interactive web applications with R.

First, change your YAML.

```
---
title: "Coolest DEMO EVER"
runtime: shiny
output:
  flexdashboard::flex_dashboard:
```

Next, assign your interactive plot or table to an object, and render.

```
renderPlotly(plot1)

renderReactable(table1)
```

I know what you're thinking.

Interactivity isn't enough. How can I make my dashboard even cooler?

I have the answer. Icons.

I recommend using Fontawesome icons.But you can also use ionicons.

```
#Use the prefix "fa" for fontawesome icons or "ion" for ionicons.

Weird Flex {data-icon="fa-code-branch"}
==================================
```

Want a logo?

So simple, my friends. Download a fontawesome icon to your project folder, and change your YAML.

Check the size of your png file!

```
---
title: "Coolest DEMO EVER"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
    logo: coolpic.png
    favicon: coolicon.png
    social: menu
    source_code: https://github.com/akhilanekkanti1/dashboard_demo

---
```

Storyboards

HTML Widgets Showcase

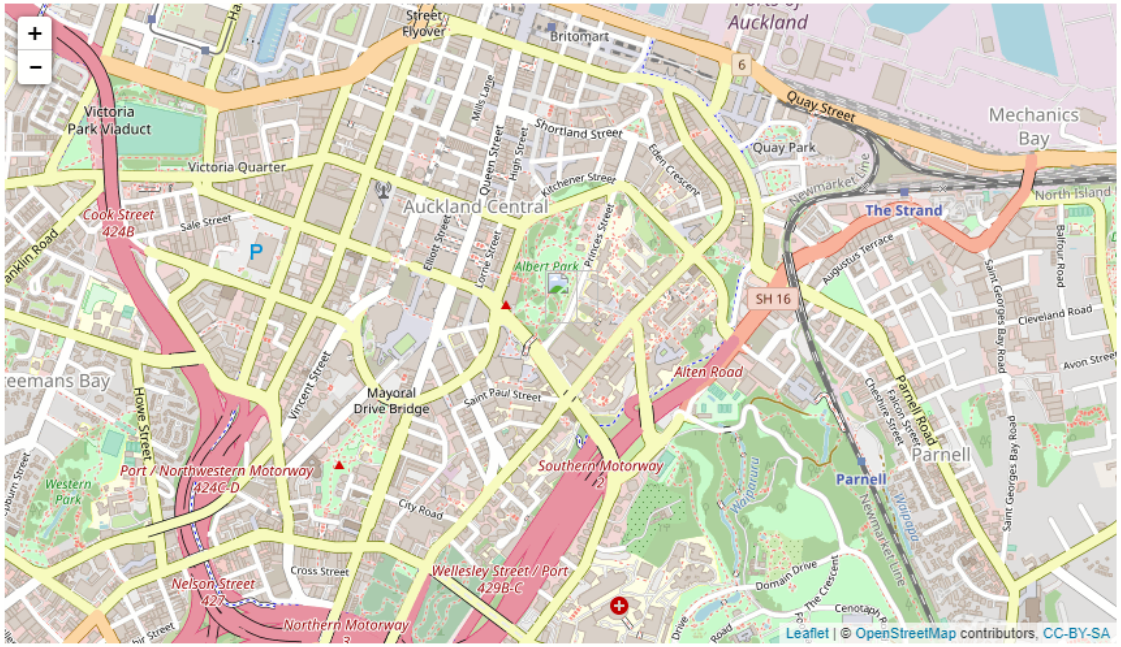
Source Code

Leaflet is a JavaScript library for creating dynamic maps that support panning and zooming along with various annotations.

d3heatmap creates interactive D3 heatmaps including support for row/column highlighting and zooming.

Dygraphs provides rich facilities for charting time-series data in R and includes support for many interactive features.

Plotly provides bindings to the plotly.js library and allows you to easily translate your ggplot2 graphics into an interactive web-based version.



https://rstudio.github.io/leaflet/

- Interactive panning/zooming
- Compose maps using arbitrary combinations of map tiles, markers, polygons, lines, popups, and GeoJSON.
- Create maps right from the R console or RStudio
- Embed maps in knitr/R Markdown documents and Shiny apps
- Easily render Spatial objects from the sp package, or data frames with latitude/longitude columns
- Use map bounds and mouse events to drive Shiny logic

Storyboard all the way.

1. First, change your YAML.
2. Use level 3 headers (###) for each frame in your storyboard.

```
---
title: "Storyboard"
output:
  flexdashboard::flex_dashboard:
    storyboard: true
---

### Frame 1

### Frame 2

### Frame 3
```

Storyboard all the way.

Storyboards on just one page.

1. Just add {.storyboard} to your page title.

```
Cool Story {.storyboard}
=========================
### Frame 1

### Frame 2
```

Last but not least, themes!

https://rmarkdown.rstudio.com/flexdashboard/using.html#themes

```
---
title: "Coolest DEMO EVER"
output:
  flexdashboard::flex_dashboard:
    orientation: columns
    vertical_layout: fill
    theme: united
    social: menu
    source_code: https://github.com/akhilanekkanti1/dashboard_demo

---
```

Thanks!

Slides created via the R package xaringan.

The chakra comes from remark.js, knitr, and R Markdown.