

{flexdashboards}

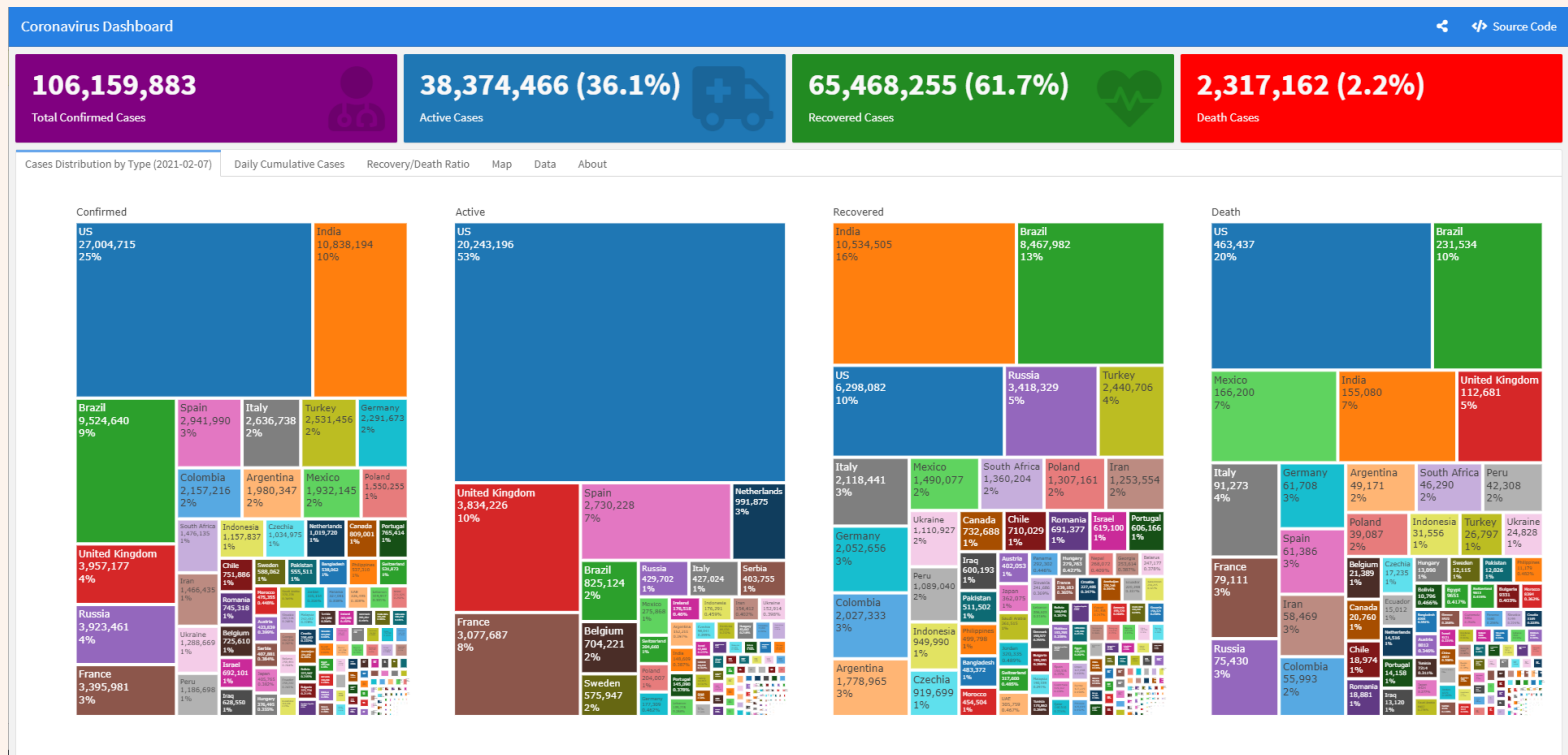
Data visuals. On a board.

Akhila Nekkanti

Winter 2021

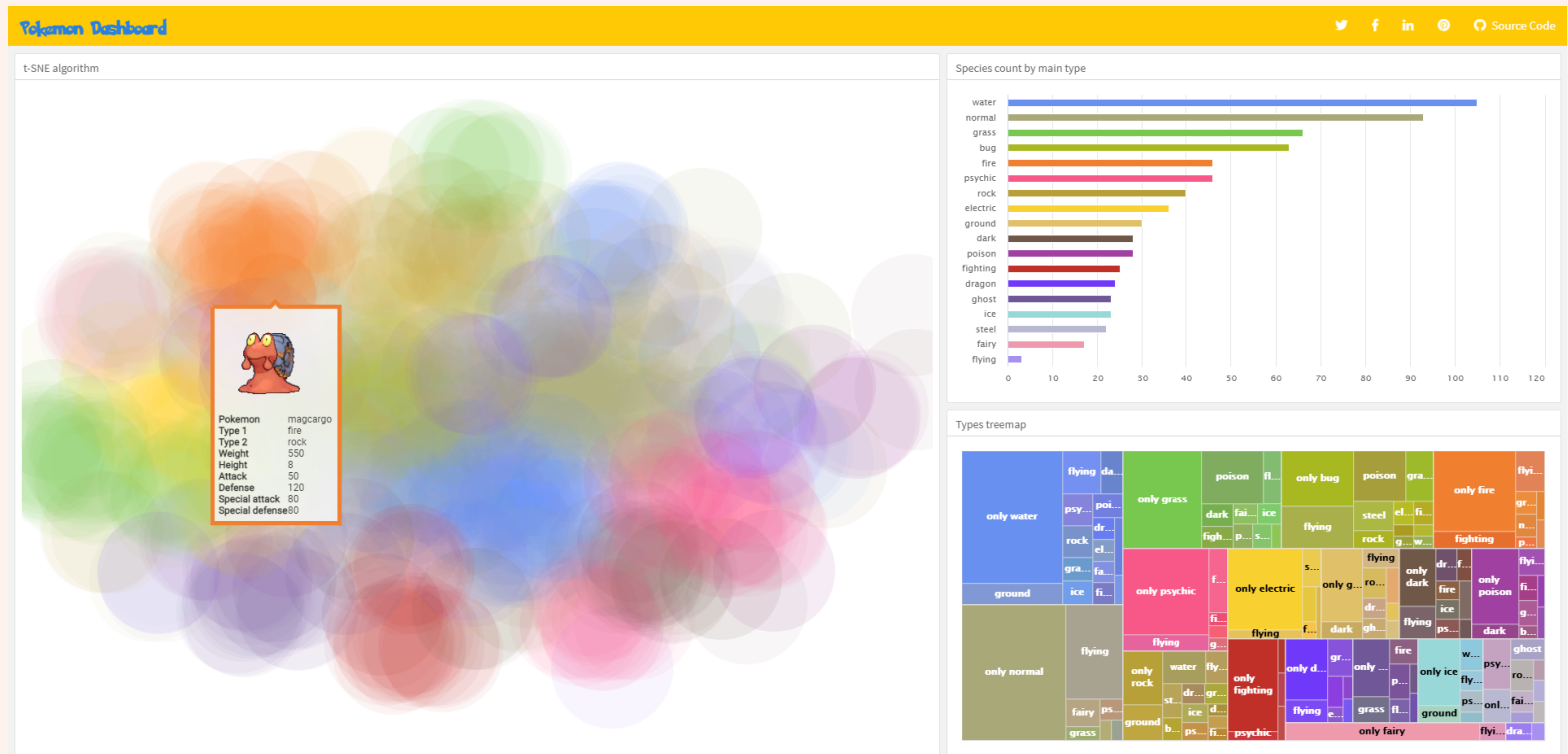
Why use a dashboard?

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



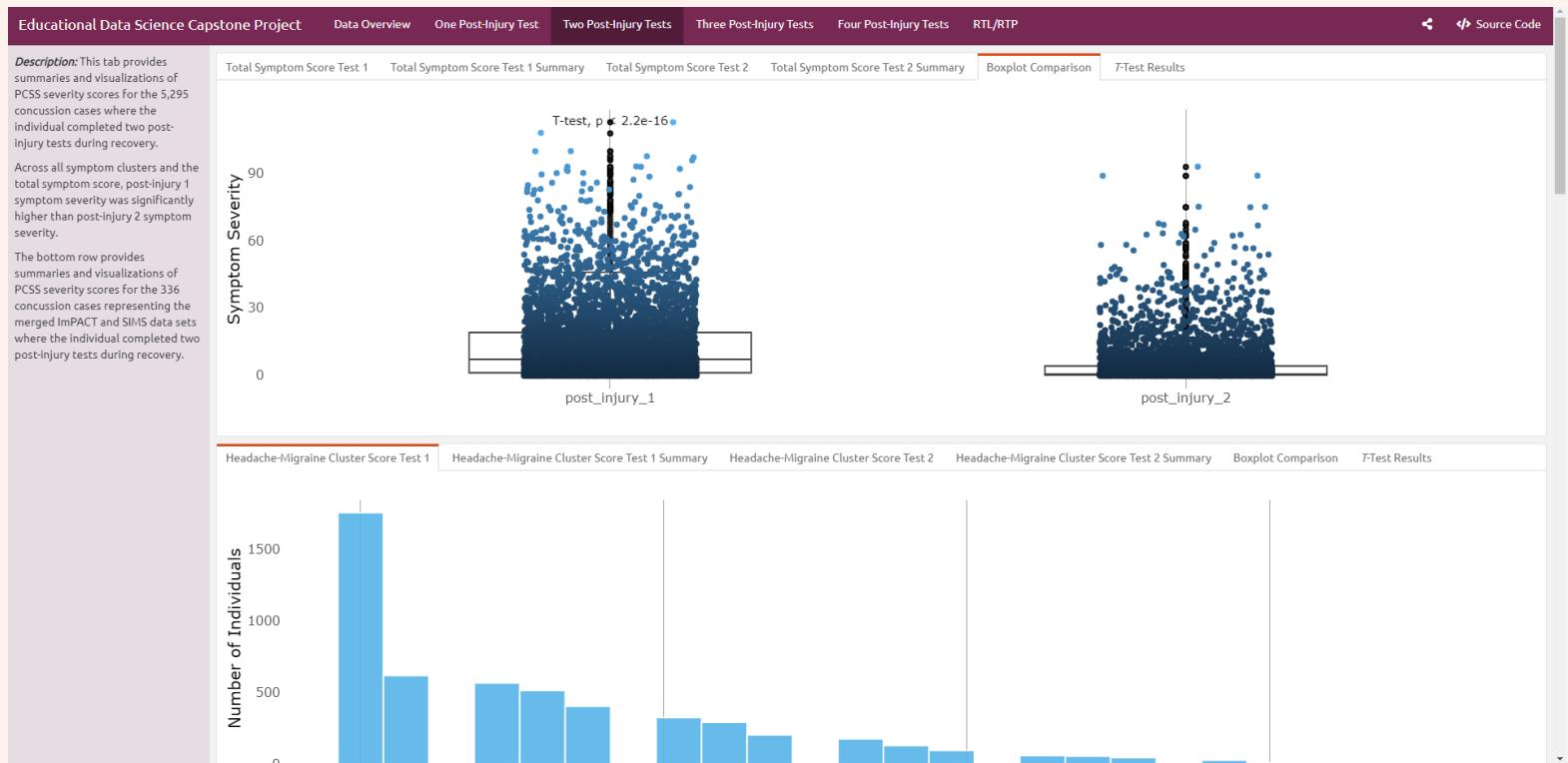
Why use a dashboard?

<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/akhilaneekkanti1/FlexDashboards>

Oregon Child Abuse Prevalence Study

Background Abuse Rates Community Response Foster Care Placement Effects

[Source Code](#)


Data Source

The plots displayed on this dashboard are generated from a synthetic dataset intended to represent original data from the Oregon Child Abuse Prevalence Study (OCAPS) Pilot.




The OCAPS pilot surveys high school students across 5 school districts in Lane County to assess students' current experiences of abuse, harassment, and social support.

Daniel Anderson produced this synthetic data using the `synthpop` package.

Akhila Nekkanti created the plots shown using the `ggplot` package.



Policymakers and funders rely heavily on state-wide child maltreatment data to prioritize public concerns, enact policy, and establish budgets for evidence-based programming (PEW Data Report, 2018). As such, the accurate translation of analyses with such data into visually compelling, easily digestible means is a critical pre-requisite to bridging the gap between advocates, researchers, and legislative bodies. In Oregon these data are dependent on either retrospective surveys with adults (Oregon BRFFS), youth convenience sample surveys with fewer than 10 abuse and neglect items (Oregon Healthy Teens Survey, Oregon Student Wellness Survey), and reports to child protective services (Children's Bureau, 2019). This critical need for accurate data is met by a glaring gap as researchers and advocates agree that child maltreatment rates in the United States are significantly underreported (Swahn et al., 2006; Flaherty et al., 2008). University of Oregon's Center for the Prevention of Abuse and Neglect (CPAN) implemented a pilot study to determine safer, more accurate methods of data collection in the school setting. Partnered with the Oregon Department of Education and multiple statewide agencies and advocacy groups, CPAN is currently implementing the first state-wide representative survey of abuse experiences with 1500-1800 youth in Oregon, to be completed in June 2021.



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.

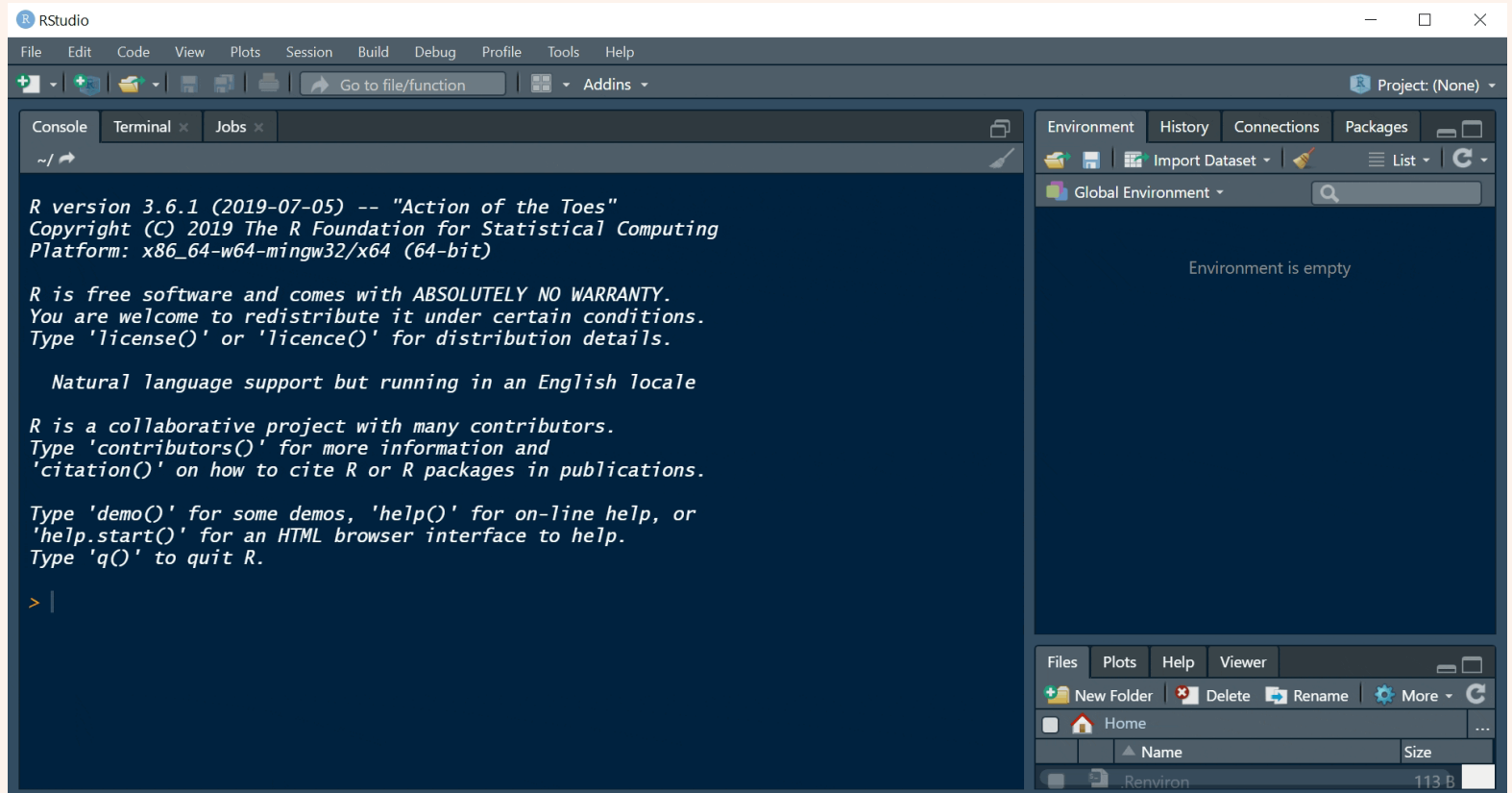
1. Create a new Project
2. Create a new Rmarkdown from template {flexdashboards}
3. Knit right away
4. Create a github Repo
5. Link your local and remote repos with GitKraken

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

The screenshot shows a web browser window displaying the GitHub profile of user 'akhilanekkanti1'. The browser's address bar shows 'github.com/akhilanekkanti1'. The page features a circular profile picture of a person sitting by a waterfall, the username 'akhilanekkanti1', and a bio '1 follower · 7 following · ☆ 3'. A 'ProTip!' banner encourages updating the profile. The 'Overview' tab is active, showing 'Popular repositories' with entries like 'brainhack-eugene.github.io', 'selbosh.github.io', 'project', and 'final_project_EDLD610'. A Windows taskbar is visible at the bottom with various application icons and a system clock showing 12:34 PM on 2/8/2021.

akhilanekkanti1

Search or jump to... Pull requests Issues Marketplace Explore

Overview Repositories 21 Projects 1 Packages

ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you. [Edit profile](#)

Popular repositories Customize your pins

- [brainhack-eugene.github.io](#)
Forked from brainhack-eugene/brainhack-eugene.github.io
Eugene Brainhack 2018
HTML
- [selbosh.github.io](#)
Forked from Selbosh/selbosh.github.io
Personal web site, "Tea & Stats", built with blogdown and Hugo
HTML
- [project](#) Archived
- [final_project_EDLD610](#) Archived
Forked from sdimakis/final_project_EDLD610
Grah Dimakis' Group: Final project for EDLD Introduction to Data Science.
We report personality and political belief changes across the lifespan.

gifcap.dev is sharing your screen. [Stop sharing](#) [Hide](#)

Type here to search

12:34 PM
2/8/2021

Ready to PUBLISH!

The screenshot shows a web browser window displaying the GitHub repository page for 'akhilanekkanti1/dashboard_demo'. The browser's address bar shows the URL 'github.com/akhilanekkanti1/dashboard_demo'. The repository page has a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the repository name 'akhilanekkanti1 index' is shown, along with the commit hash '4aa5835', the time '31 seconds ago', and '3 commits'. A table lists the files in the repository:

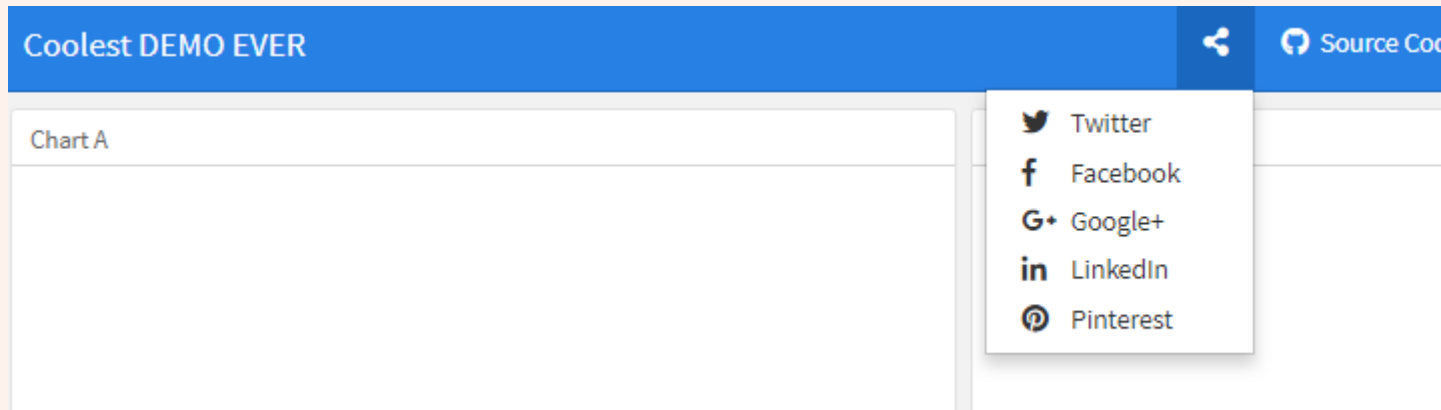
File	Commit	Time
.gitignore	init	2 hours ago
dashboard_demo.Rproj	init	2 hours ago
demo.Rmd	init	38 minutes ago
demo.html	init	38 minutes ago
index.Rmd	index	31 seconds ago
index.html	index	31 seconds ago

Below the file list, there is a blue box with the text 'Help people interested in this repository understand your project by adding a README.' and a green button labeled 'Add a README'. On the right side of the page, there are sections for 'About' (with the text 'No description, website, or topics provided.'), 'Releases' (with the text 'No releases published' and a link 'Create a new release'), 'Packages' (with the text 'No packages published' and a link 'Publish your first package'), and 'Environments' (with a link 'github-pages' and a status 'Active').

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 >

Cooltest DEMO EVER

Weird Flex

But Okay



Source Code

Chart A

Chart B



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/akhilaneekkanti1/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/akhilaneekkanti1/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}
```

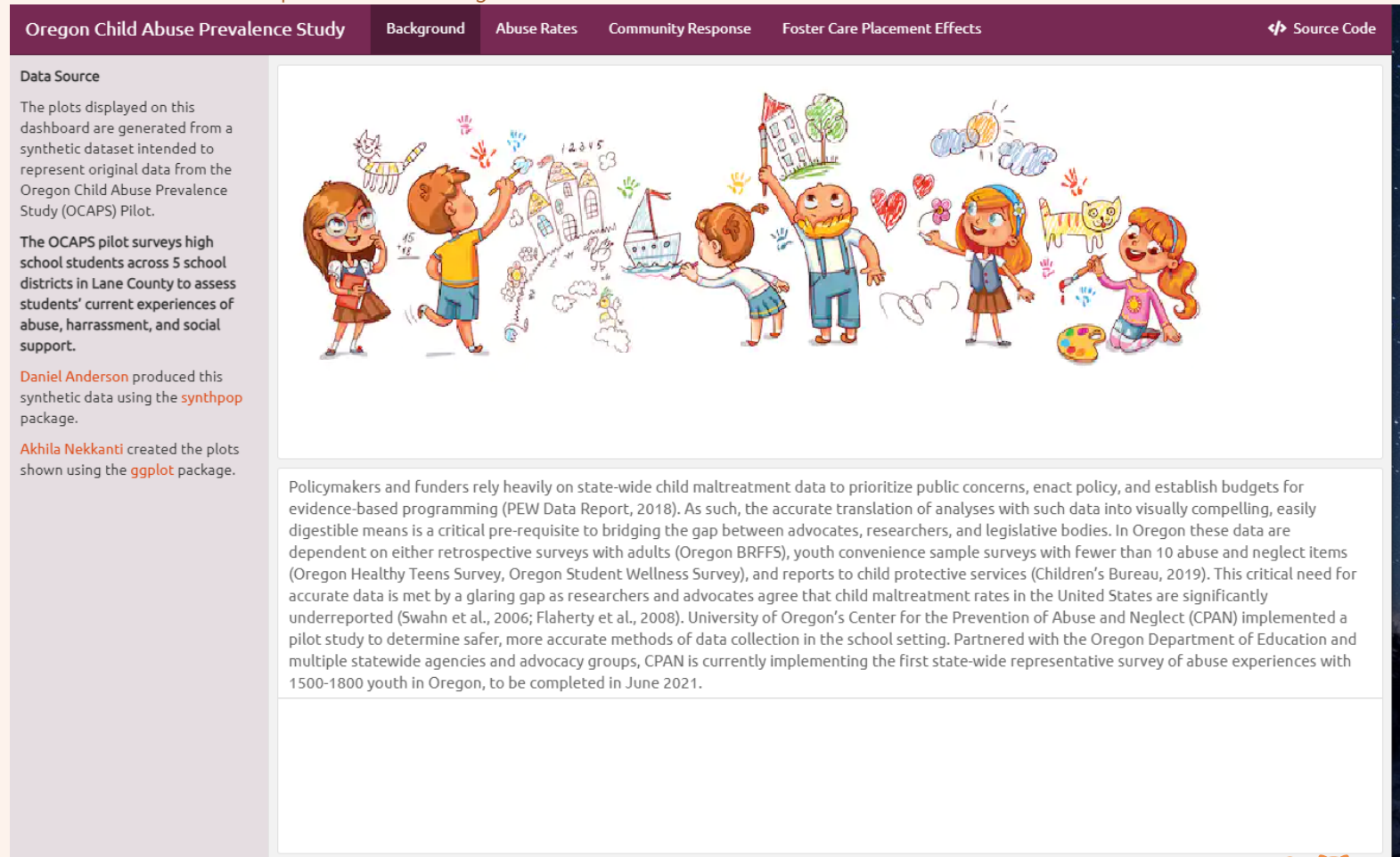
- Add a new square within each row

```
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!

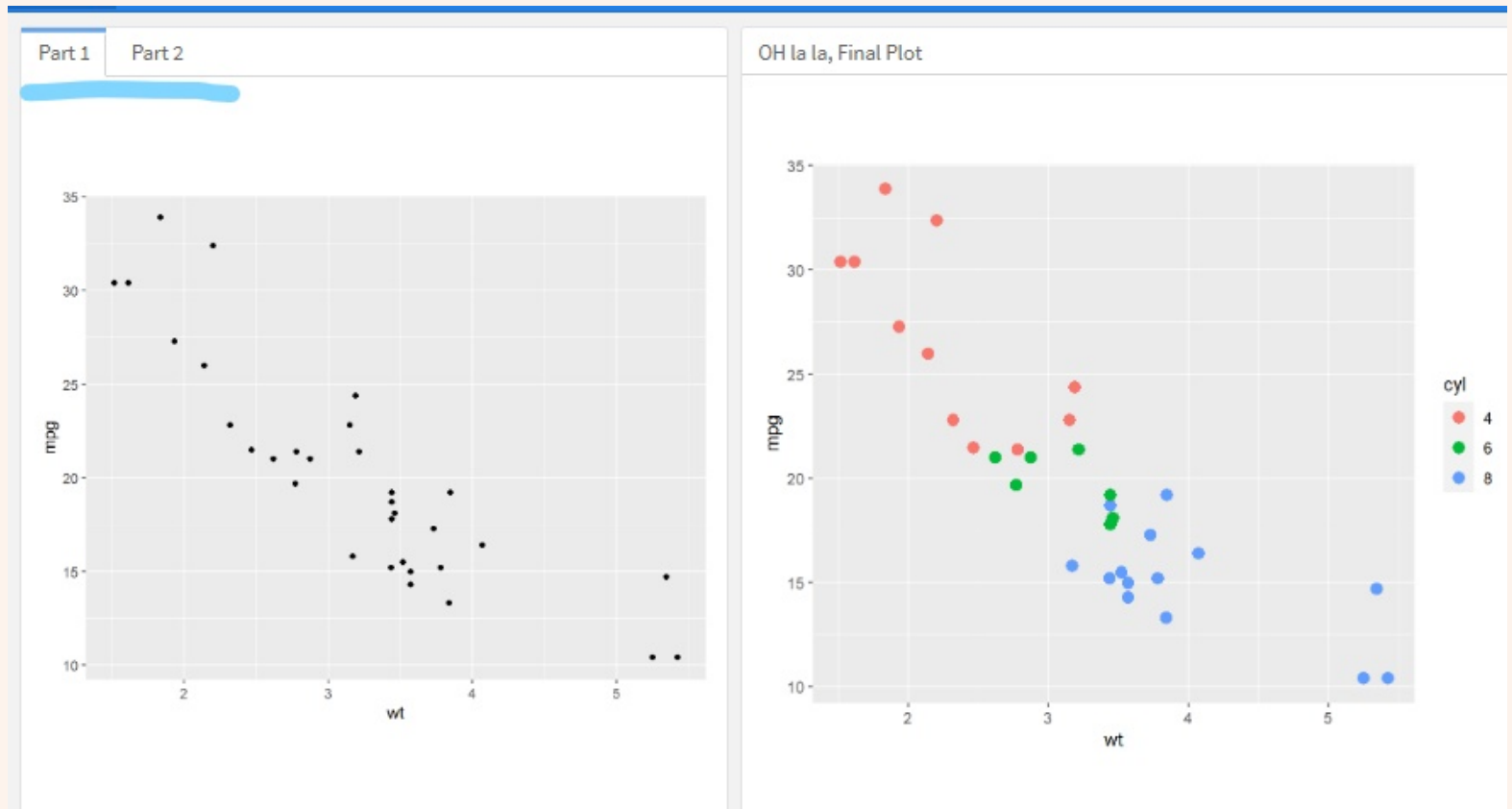
[Click here](https://youtu.be/dQw4w9WgXcQ).
```

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.

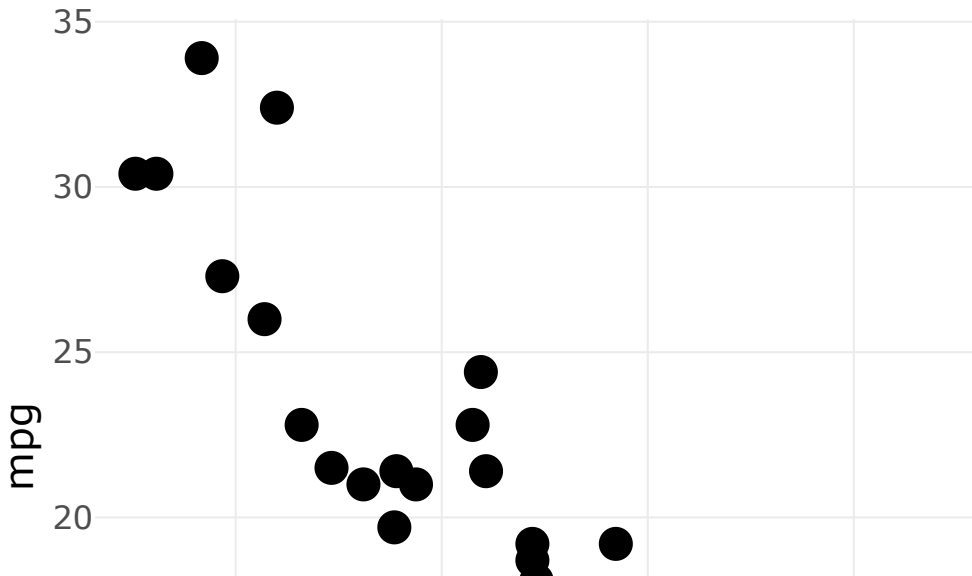
```
Column {.tabset data-width=350}
```

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"}  
=====
```

Coollest DEMO EVER

 Weird Flex

But Okay



Source Code

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/akhilaneekkanti1/dashboard_demo
---
```

Storyboards

HTML Widgets Showcase

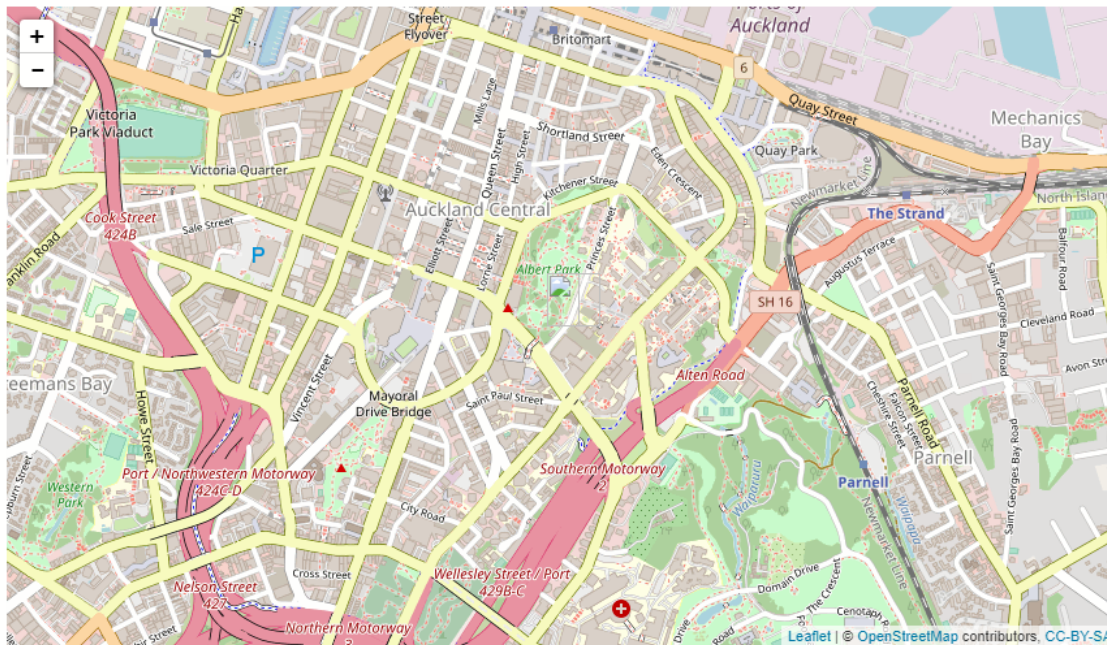
Source Code

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/akhilaneekkanti1/dashboard_demo  
---
```

Thanks!

Slides created via the R package **xaringan**.

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