# Interactivity

## Session 11

PMAP 8921: Data Visualization with R
Andrew Young School of Policy Studies
Fall 2024

# Plan for today

**Making interactive graphics**

**Sharing content**

# Making interactive graphics

# Three general methods

## Single plots with {plotly}

**Easy!**

## Dashboards with {flexdashboard}

**Slightly more complicated**

## Complete interactive apps with Shiny

**Super complicated!**

# Single plots with plotly

**Plotly** is special software for creating interactive plots with JavaScript

No knowledge of JavaScript needed!

`ggplotly()` in the {plotly} R package translates between R and Javascript for you!

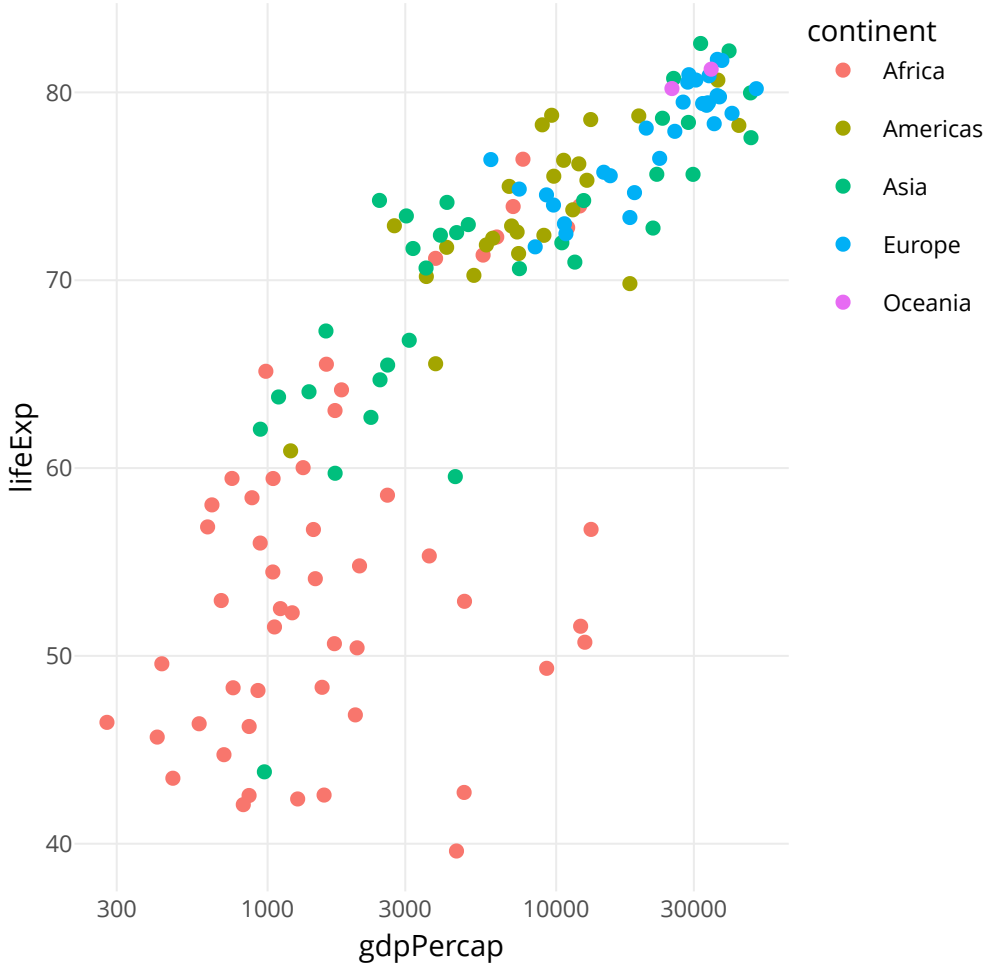
# Plotly

```
library(gapminder)
library(plotly)

gapminder_2007 <- filter(gapminder,
                         year == 2007)

my_plot <- ggplot(
  data = gapminder_2007,
  mapping = aes(x = gdpPercap, y = lifeExp,
                color = continent)) +
  geom_point() +
  scale_x_log10() +
  theme_minimal()
```

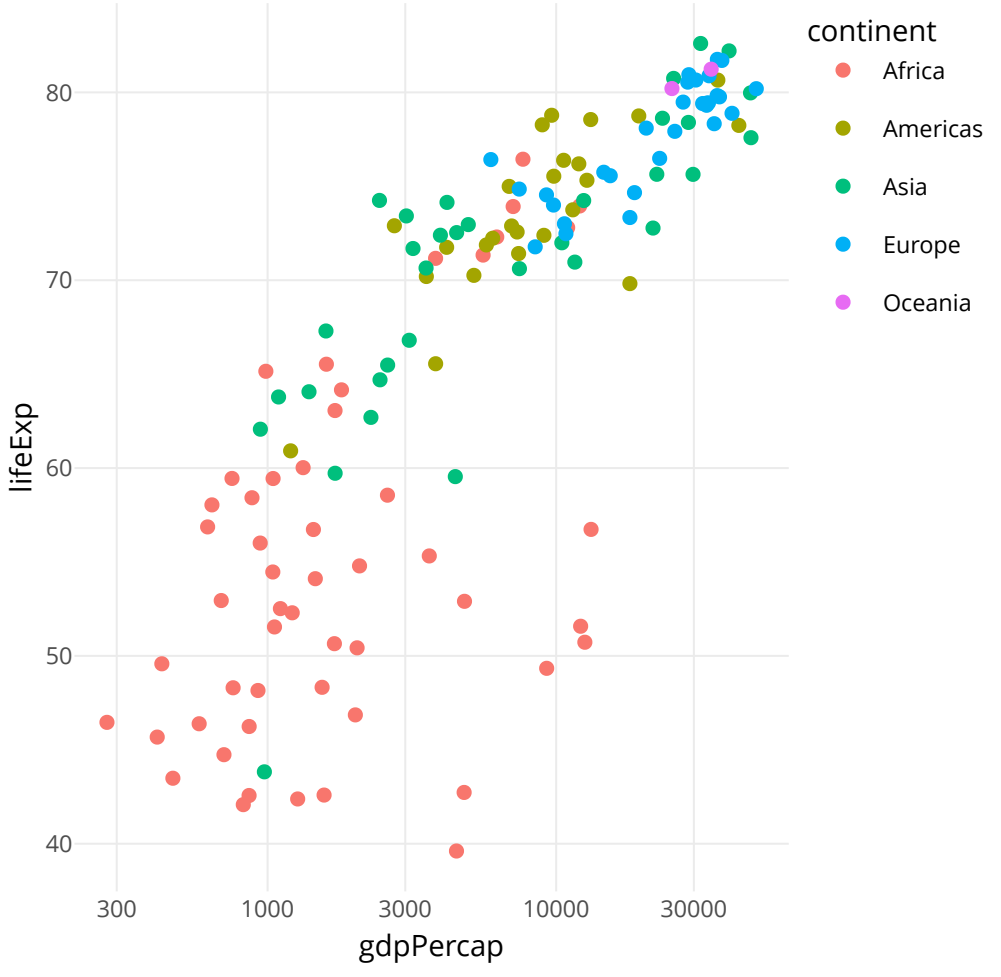
```
ggplotly(my_plot)
```



# Plotly tooltips

```
my_plot <- ggplot(
  data = gapminder_2007,
  mapping = aes(x = gdpPercap, y = lifeExp,
                color = continent)) +
  geom_point(aes(text = country)) +
  scale_x_log10() +
  theme_minimal()
```

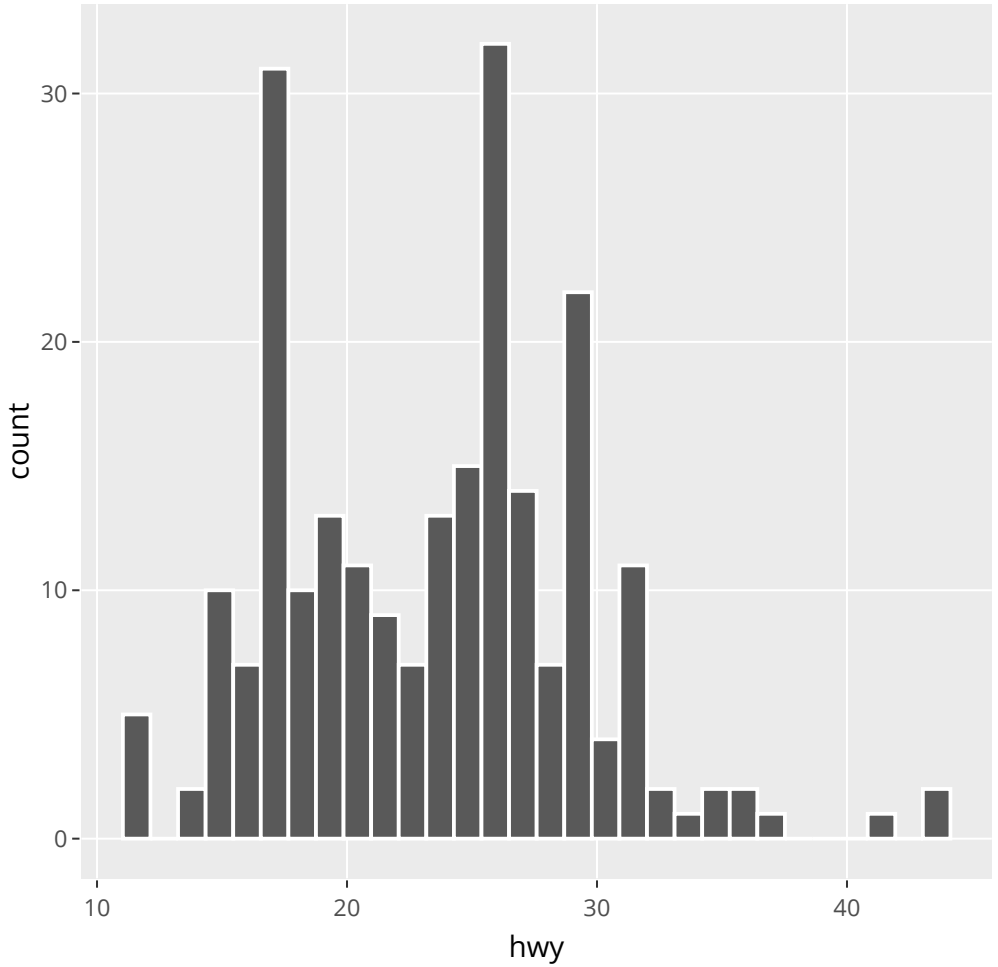
```
interactive_plot <- ggplotly(
  my_plot, tooltip = "text"
)
interactive_plot
```



# Works with most geoms!

```
car_hist <- ggplot(mpg,
                   aes(x = hwy)) +
  geom_histogram(binwdith = 2,
                 boundary = 0,
                 color = "white")
```

```
ggplotly(car_hist)
```



# Save as HTML

**Save a self-contained HTML version of it with `saveWidget()` in the {htmlwidgets} R package**

```
# This is like ggsave, but for interactive HTML plots
htmlwidgets::saveWidget(interactive_plot, "fancy_plot.html")
```

# Fully documented

The **documentation** for ggplot2 + plotly is full of examples of how to customize everything

Rely on that ↑ + Google to make really fancy (and easy!) interactive plots

# Three general methods

## Single plots with {plotly}

**Easy!**

## Dashboards with {flexdashboard}

**Slightly more complicated**

# Dashboards with {flexdashboard}

## Use basic R Markdown to build a dashboard!

```
---
title: "Single Column (Fill)"
output:
  flexdashboard::flex_dashboard:
    vertical_layout: fill
---

### Chart 1

```{r}

```

### Chart 2

```{r}

```
```

# Dashboards with {flexdashboard}

## Make any kind of block arrangement

```
---
title: "Multiple Columns"
output: flexdashboard::flex_dashboard
---

Column {data-width=600}
-------------------------------------

### Chart 1

```{r}

```

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

### Chart 2

```{r}

```

### Chart 3

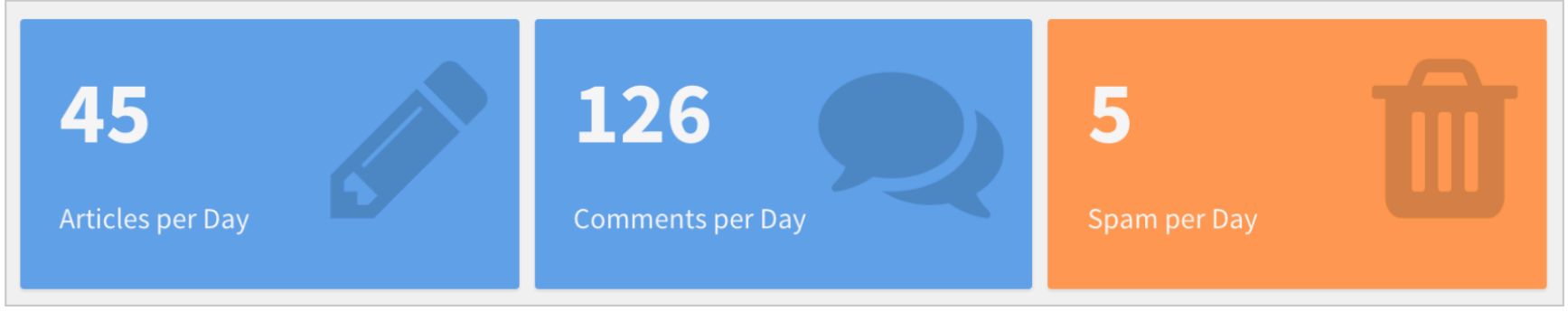
```{r}

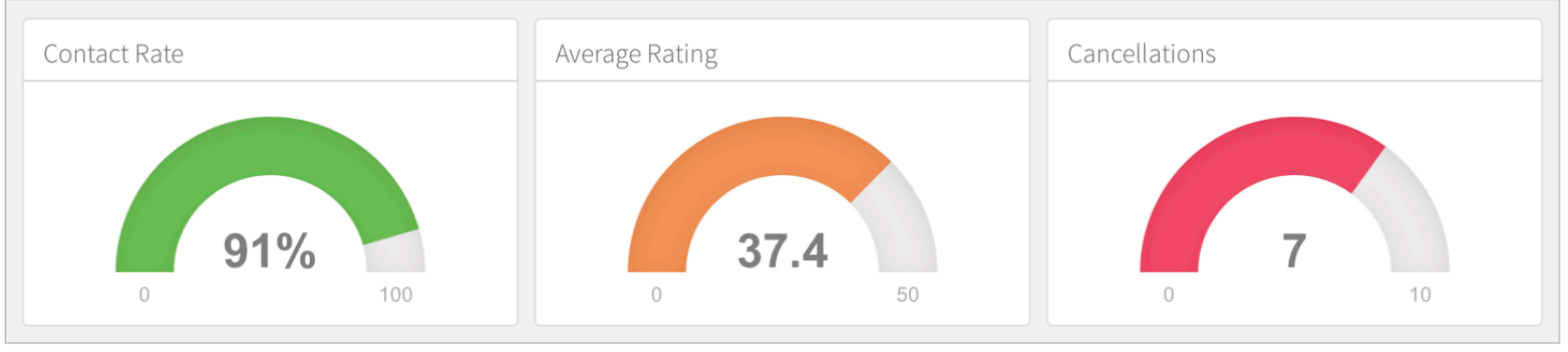
```

```

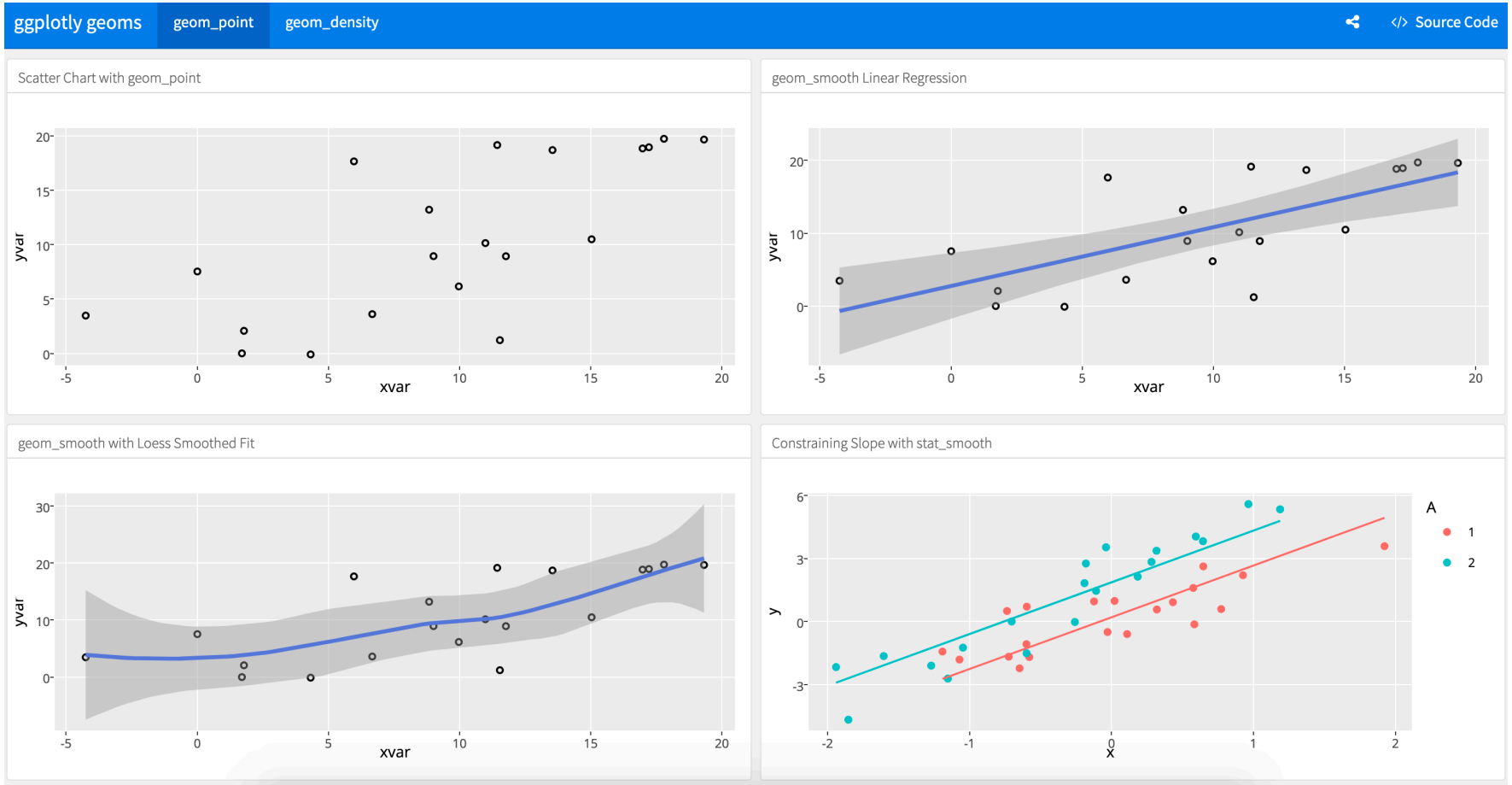
# Dashboards with {flexdashboard}

## Add other elements like text and gauges



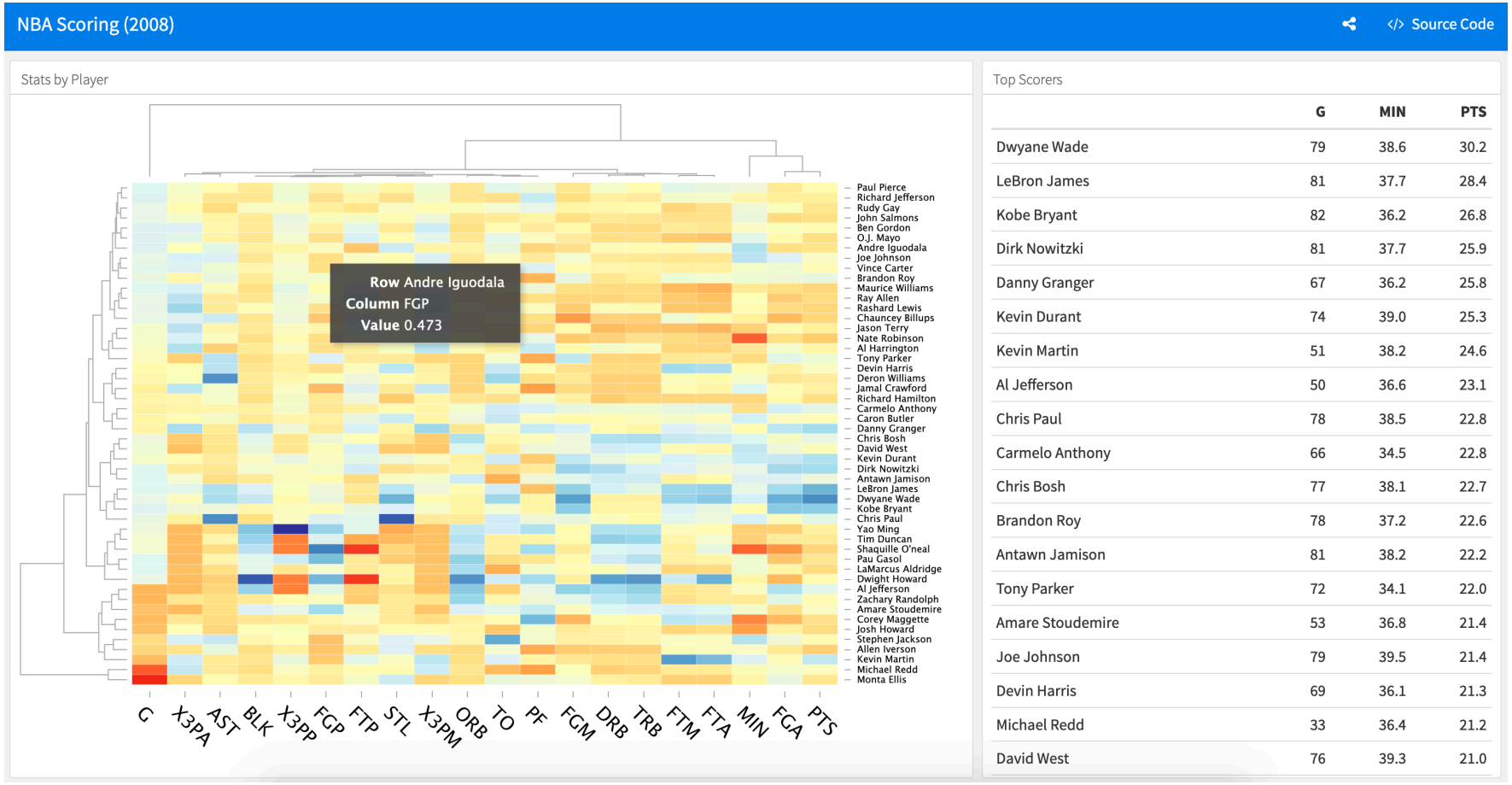



# Example dashboards




ggplot2 geoms

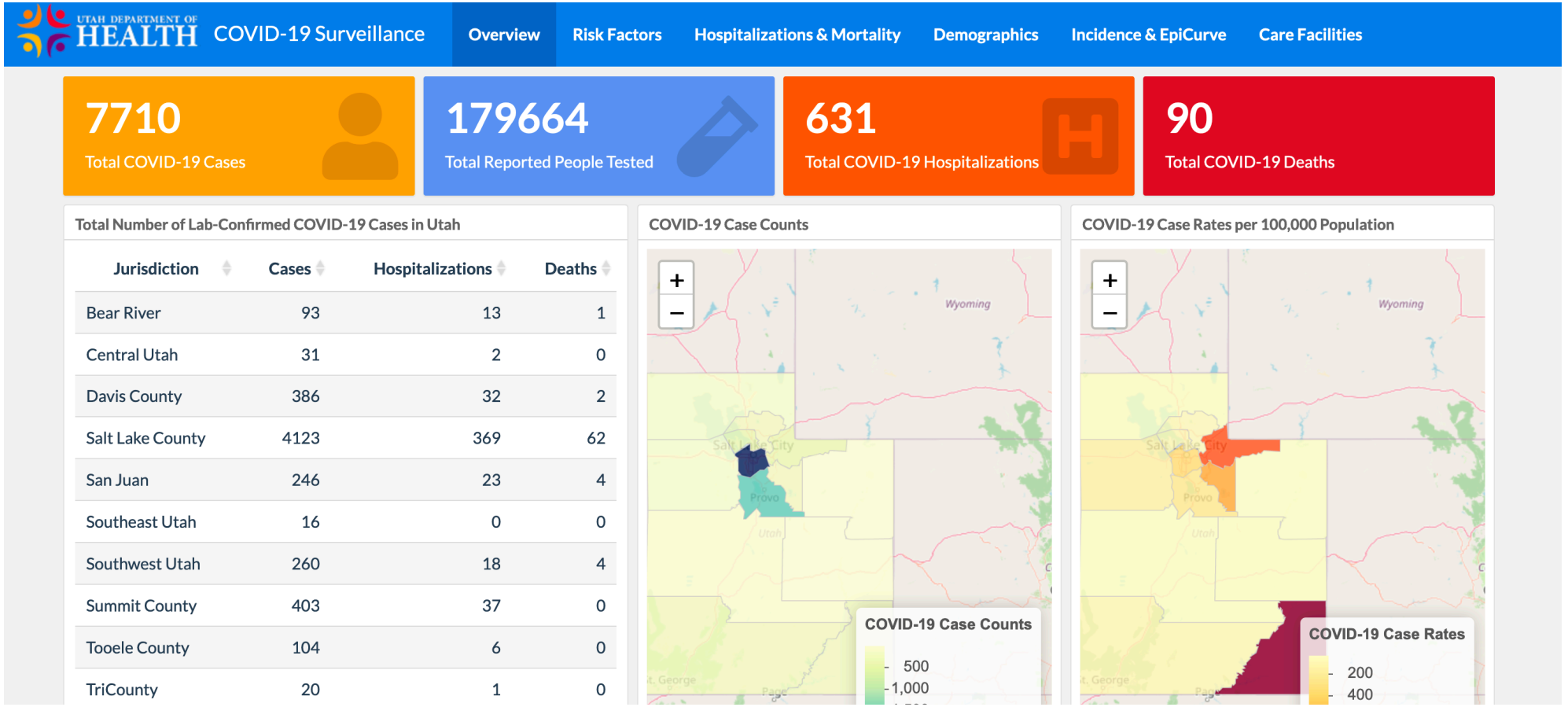
# Example dashboards




| | G | MIN | PTS |
|---|---|---|---|
| Dwyane Wade | 79 | 38.6 | 30.2 |
| LeBron James | 81 | 37.7 | 28.4 |
| Kobe Bryant | 82 | 36.2 | 26.8 |
| Dirk Nowitzki | 81 | 37.7 | 25.9 |
| Danny Granger | 67 | 36.2 | 25.8 |
| Kevin Durant | 74 | 39.0 | 25.3 |
| Kevin Martin | 51 | 38.2 | 24.6 |
| Al Jefferson | 50 | 36.6 | 23.1 |
| Chris Paul | 78 | 38.5 | 22.8 |
| Carmelo Anthony | 66 | 34.5 | 22.8 |
| Chris Bosh | 77 | 38.1 | 22.7 |
| Brandon Roy | 78 | 37.2 | 22.6 |
| Antawn Jamison | 81 | 38.2 | 22.2 |
| Tony Parker | 72 | 34.1 | 22.0 |
| Amare Stoudemire | 53 | 36.8 | 21.4 |
| Joe Johnson | 79 | 39.5 | 21.4 |
| Devin Harris | 69 | 36.1 | 21.3 |
| Michael Redd | 33 | 36.4 | 21.2 |
| David West | 76 | 39.3 | 21.0 |

NBA scoring

# Example dashboards




Utah's COVID-19 dashboard

# Outstanding documentation

The **documentation** for {flexdashboard} is full of examples and details of everything you can do

Rely on that ↑ + Google to make really fancy (and easy!) dashboards!

# Three general methods

## Single plots with {plotly}

**Easy!**

## Dashboards with {flexdashboard}

**Slightly more complicated**

## Complete interactive apps with Shiny

**Super complicated!**

# Shiny

Shiny is a complete web application framework for interactive statistics

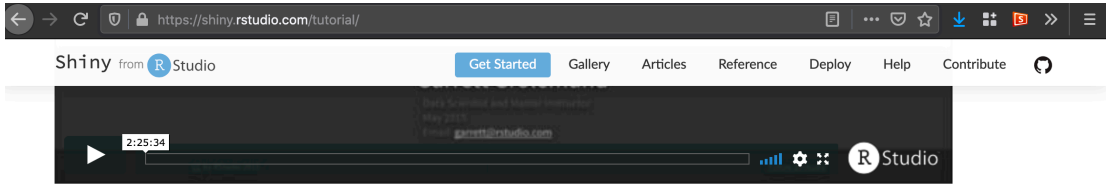
It's super complex and hard for beginners

**I've never made a standalone Shiny app!**

(And I don't plan on trying anytime soon)

# Lots of resources to help start

## RStudio has a whole website for helping you get started




Part 1 - How to build a Shiny app

1. Introduction
2. R
3. App architecture
4. App template
5. Inputs and outputs
6. The server function
7. Sharing apps
8. Shinyapps.io
9. Shiny servers
10. Recap - Part 1

Part 2 - How to customize reactions

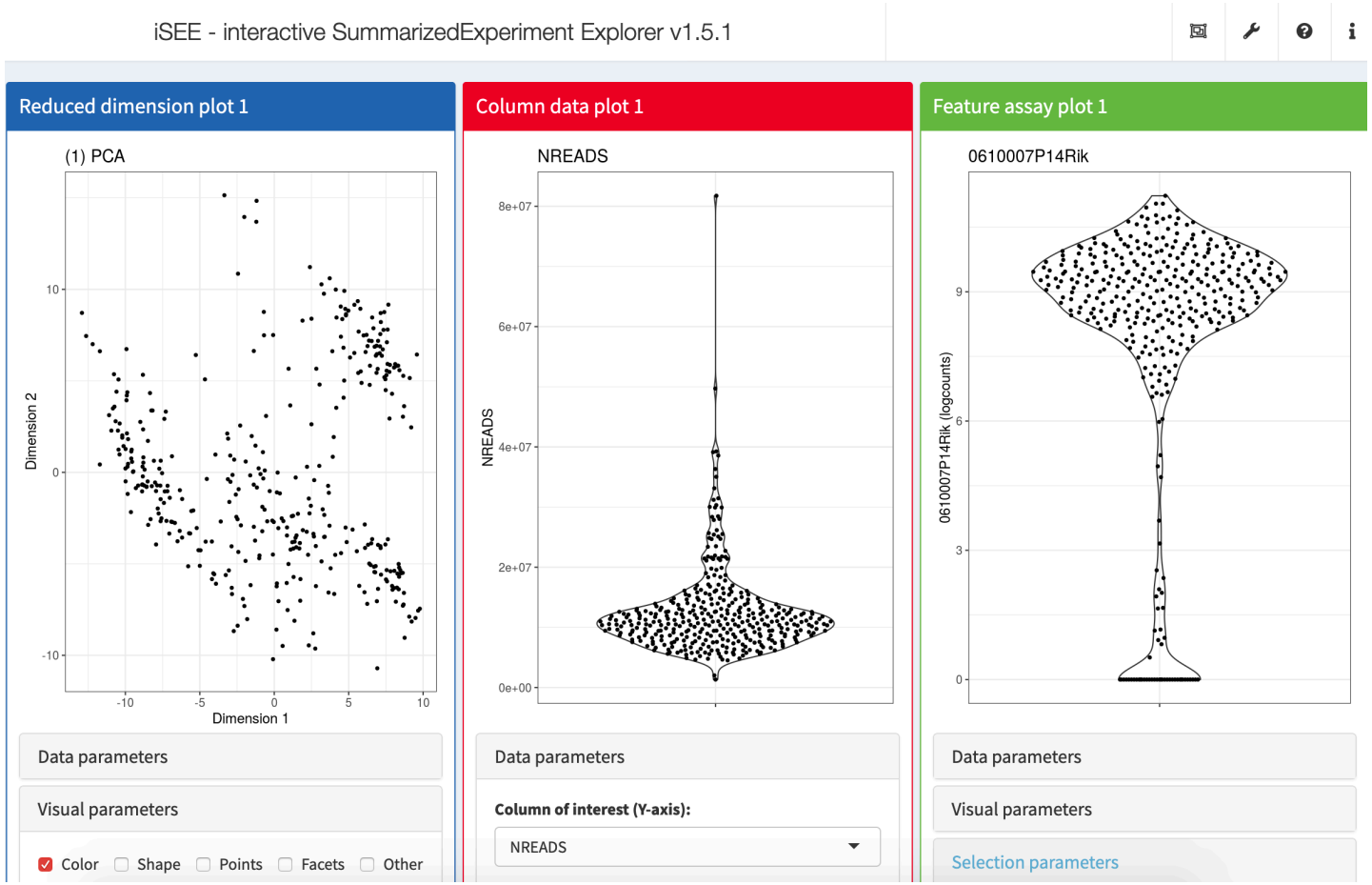
11. Introduction
12. Review of Part 1
13. Reactivity
14. Reactive values
15. Reactive functions
16. render*()
17. reactive()
18. isolate()
19. observeEvent()
20. eventReactive()
21. reactiveValues()
22. Recap - Part 2
23. Parting tips

Part 3 - How to customize appearance

24. Introduction
25. Review of Parts 1 and 2
26. HTML UI
27. Adding static content
28. Building layouts
29. Panels and tabsets
30. Prepackaged layouts
31. CSS
32. Recap - Part 3

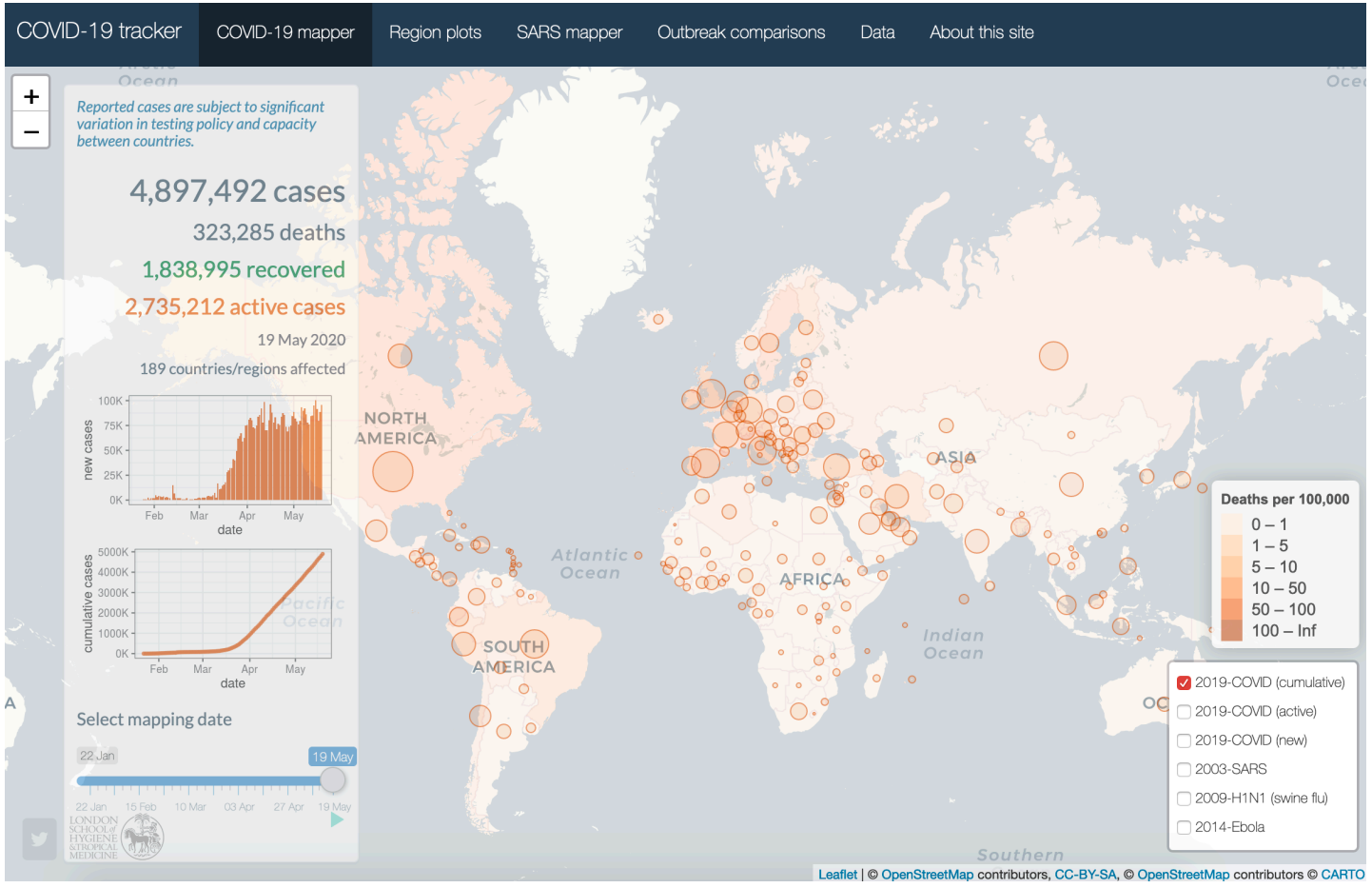
Getting started with Shiny

# Really neat examples!



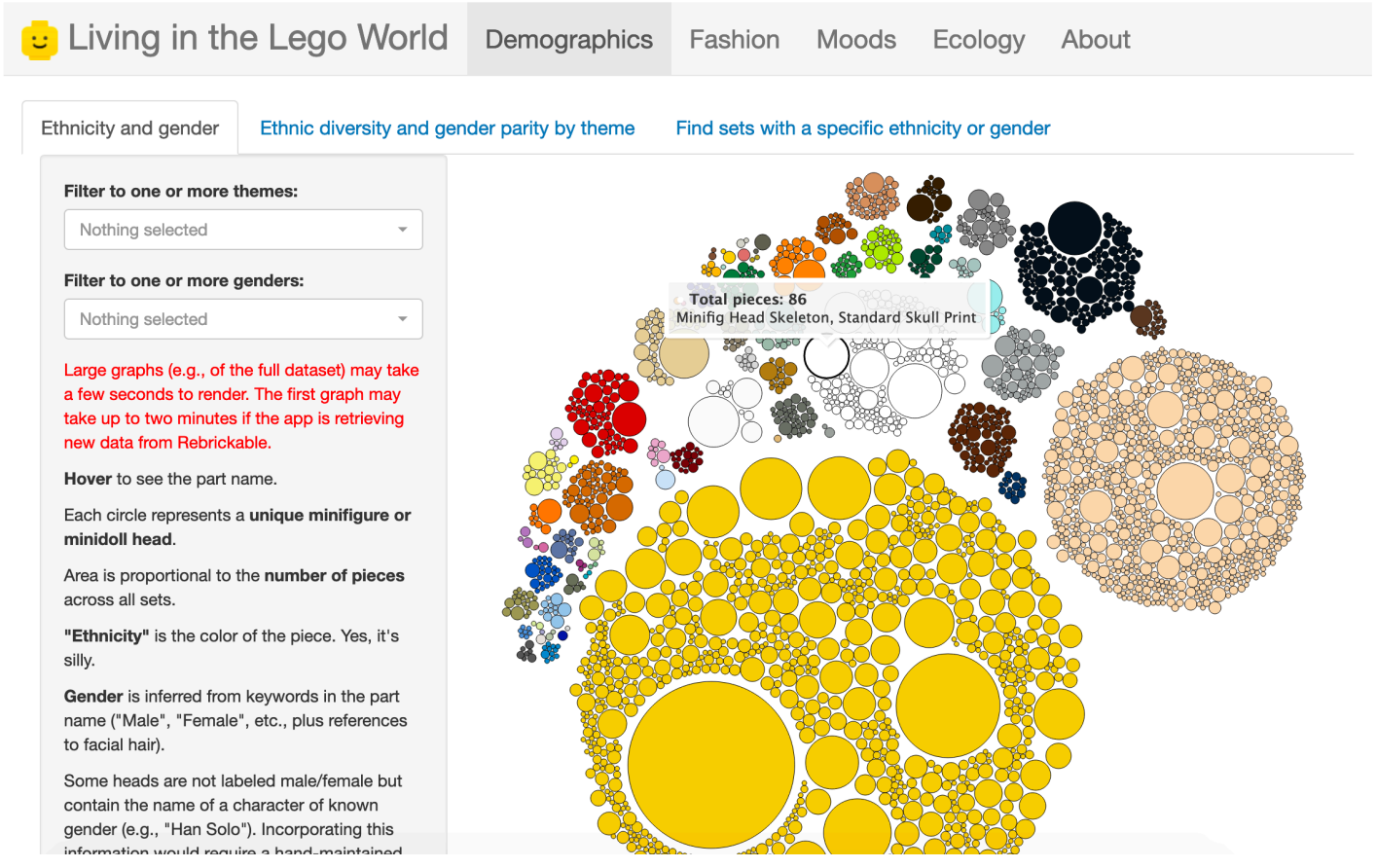

iSEE (interactive SummarizedExperiment Explorer)

# Really neat examples!




COVID-19 tracker

# Really neat examples!

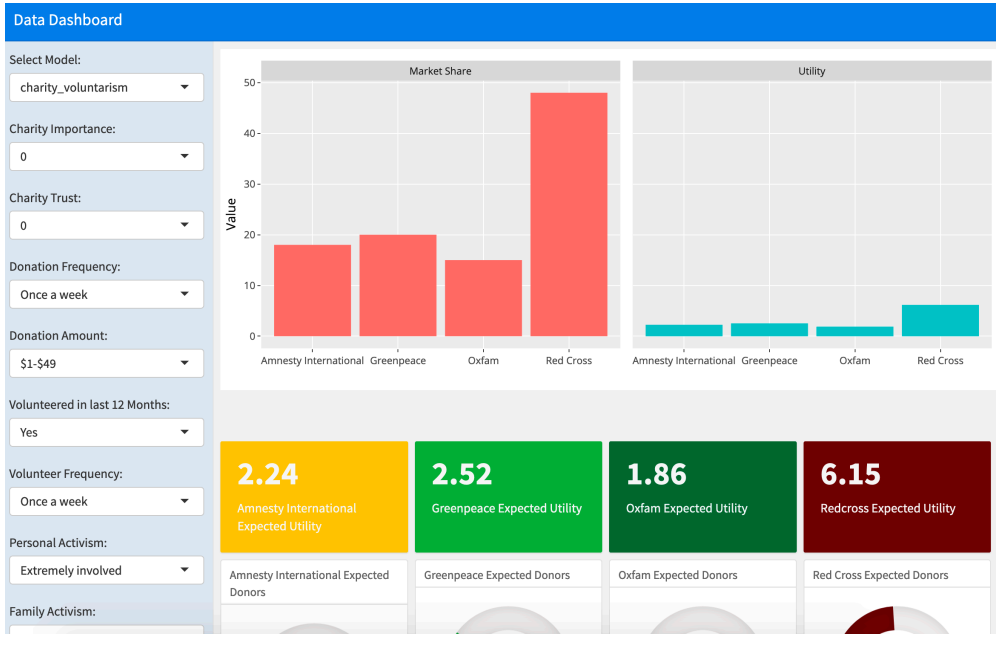



Living in the LEGO world

# flexdashboard + Shiny

**You can use reactive Shiny things in flexdashboards without building a complete Shiny app!**

**I *have* done this**



# Sharing content

# What do you do after you knit?

When knitting to PDF or Word, you make a standalone file

E-mail it, message it, Slack it, whatever

When knitting to HTML, you make a website

By default it's a standalone `.html` file with graphics embedded,
so you can still e-mail it, etc., but it can get huge if there are lots of images

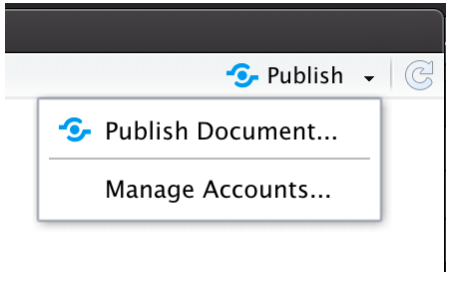
Standalone files won't work well if there's anything interactive

You can also post it online!

# Places to put HTML documents

RPubs for knitted HTML documents

Built in to RStudio; works with ggplotly!




RPubs or shinyapps.io for flexdashboards

Your own web server for anything, if you have one