



# Shiny and Flexdashboards

## Creating dashboards and interactive websites in R

Joel Herndon, Center for Data and Visualization Sciences

# Prerequisites

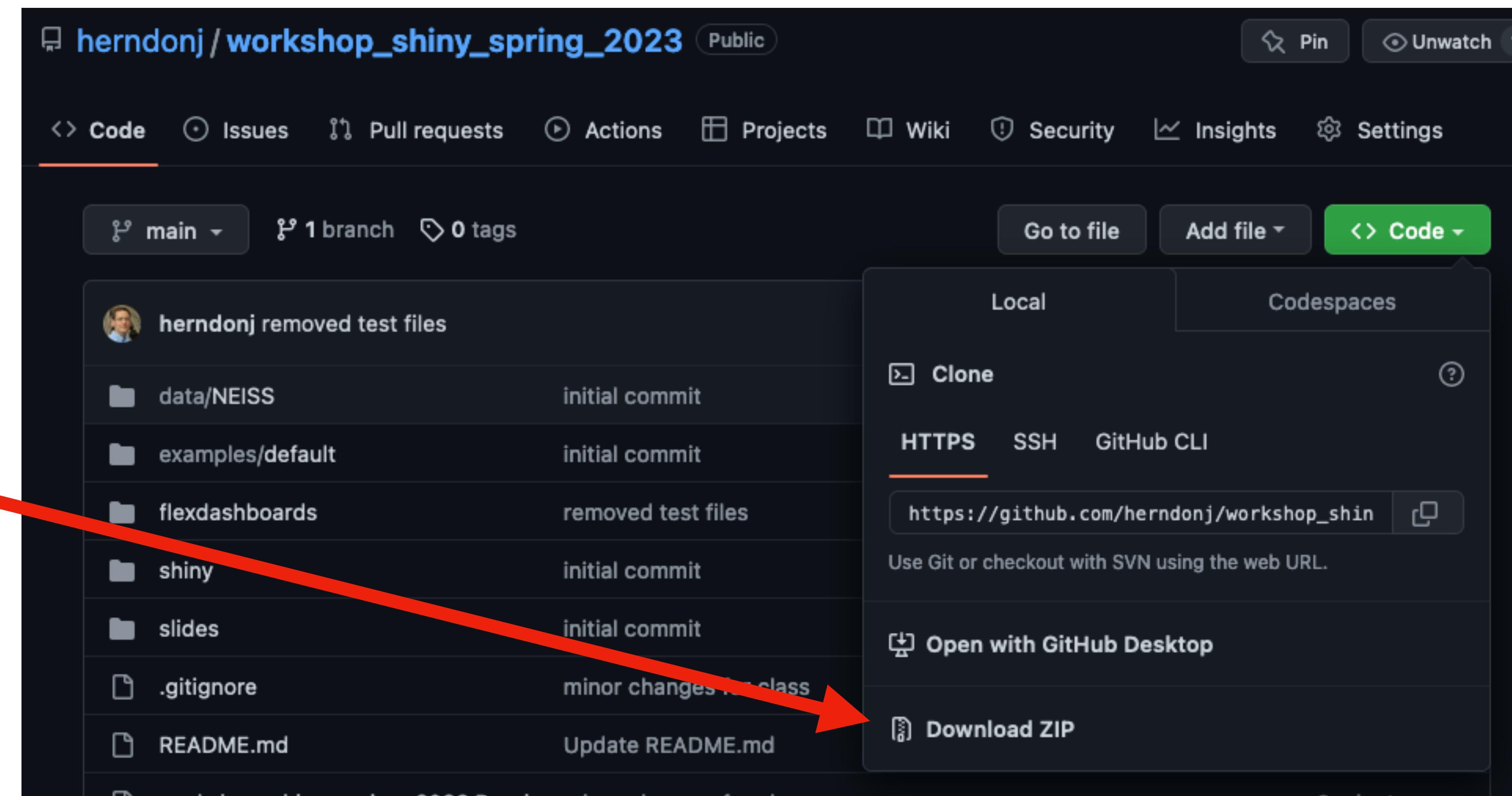
- Rstudio and R
- The following packages installed
  - tidyverse
  - flexdashboard
  - shiny



# Slides and code for the workshop

<https://bit.ly/shiny-spring-23>

Download the zip file



# Goals for Today's Workshop

1. Building flexdashboards
2. Creating shiny apps
3. Understanding reactivity

Plus:

1. Workflow tips/considerations
2. Resources for getting help
3. Hands-on exercises

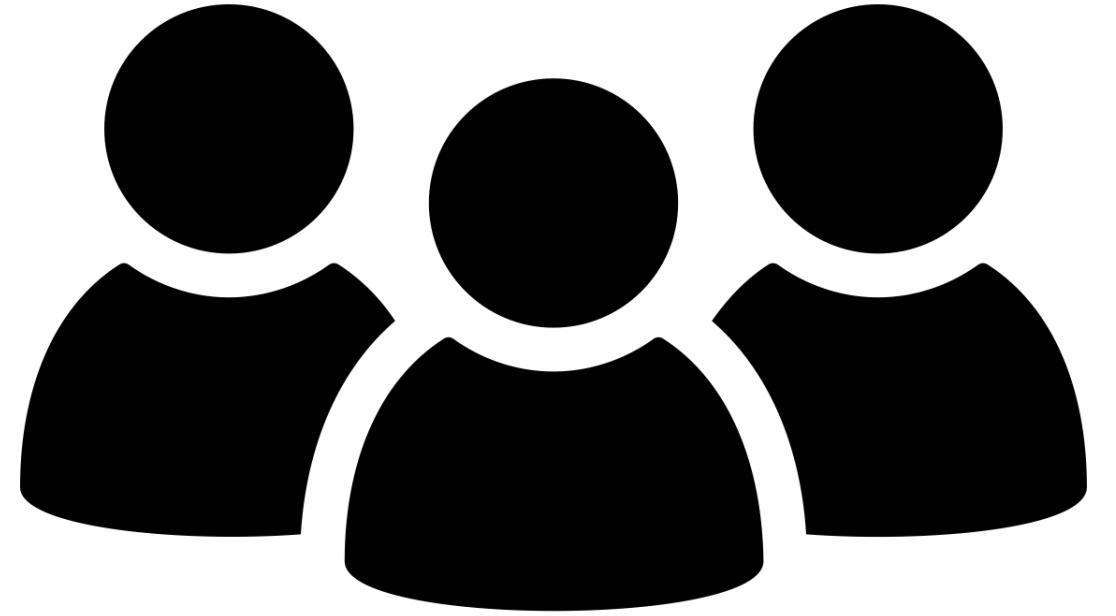


# Working assumptions

1. You have R experience, but are new to Shiny and flexdashboards
2. We're going to focus on a bit more on concepts than code
3. It's hard to see the screen in the back of this room! I'll try to make sure you can follow slides on your computer.

Let me know if you have questions or email [askdata@duke.edu](mailto:askdata@duke.edu) later.





# 2 minute meet and share

# Dashboard Benefits

- Visually compelling
- Expands the audience for your data/argument
- Empowers you, your team, and your client/audience
- Can save time for both the dashboard creator and audience

# Dashboard Types

## Static Dashboards

- Easier to design
- Easy to share by email, web, slides
- Can be refreshed to reflect changes



## Dynamic Dashboards

- Allow user interaction
- Require specialized code
- (Shiny) Requires a server to host



# Data used in today's workshop

## National Electronic Injury Surveillance System (NEISS)

- Focus: consumer product related injuries
- Sample: nationally representative sample of hospitals
- Coverage: United States, 2017
- Source: Hadley Wickham (<https://github.com/hadley/neiss>)



# **Building flexdashboards**



# Static Flexdashboards

“Sometimes simpler is easier!”



# 1. YAML Header

```
---
```

```
title: "Estimated US emergency room visit related to basketball (2017)"
```

```
output:
```

```
  flexdashboard::flex_dashboard:
```

```
    orientation: rows
```

```
    source_code: embed
```

```
---
```

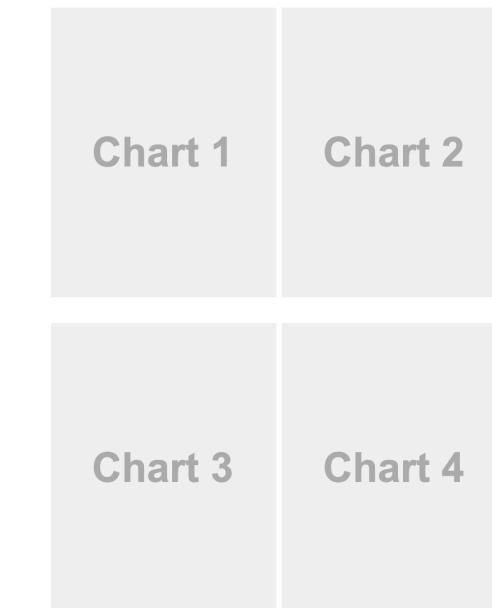
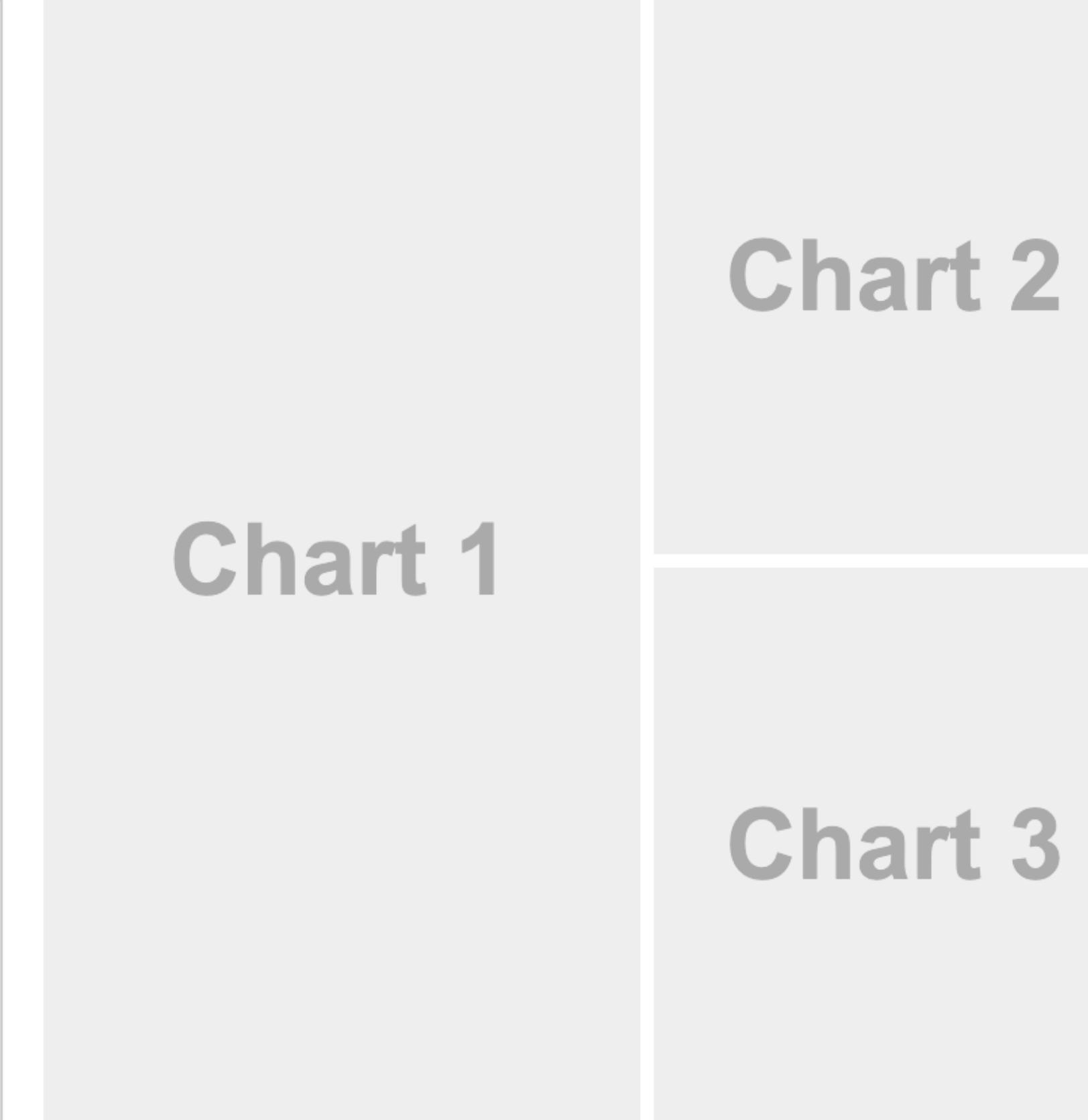
Estimated US emergency room visit related to basketball (2017)

 Source Code

# 2. Pick a layout



```
1 ---  
2 title: "Focal Chart (Left)"  
3 output: flexdashboard::flex_dashboard  
4 ---  
5  
6 Column {data-width=600}  
7 -----  
8  
9 ### Chart 1  
10  
11 `r  
12 ...  
13  
14 Column {data-width=400}  
15 -----  
16  
17 ### Chart 2  
18  
19 `r  
20 ...  
21  
22 ### Chart 3  
23  
24 `r  
25 ...  
26
```



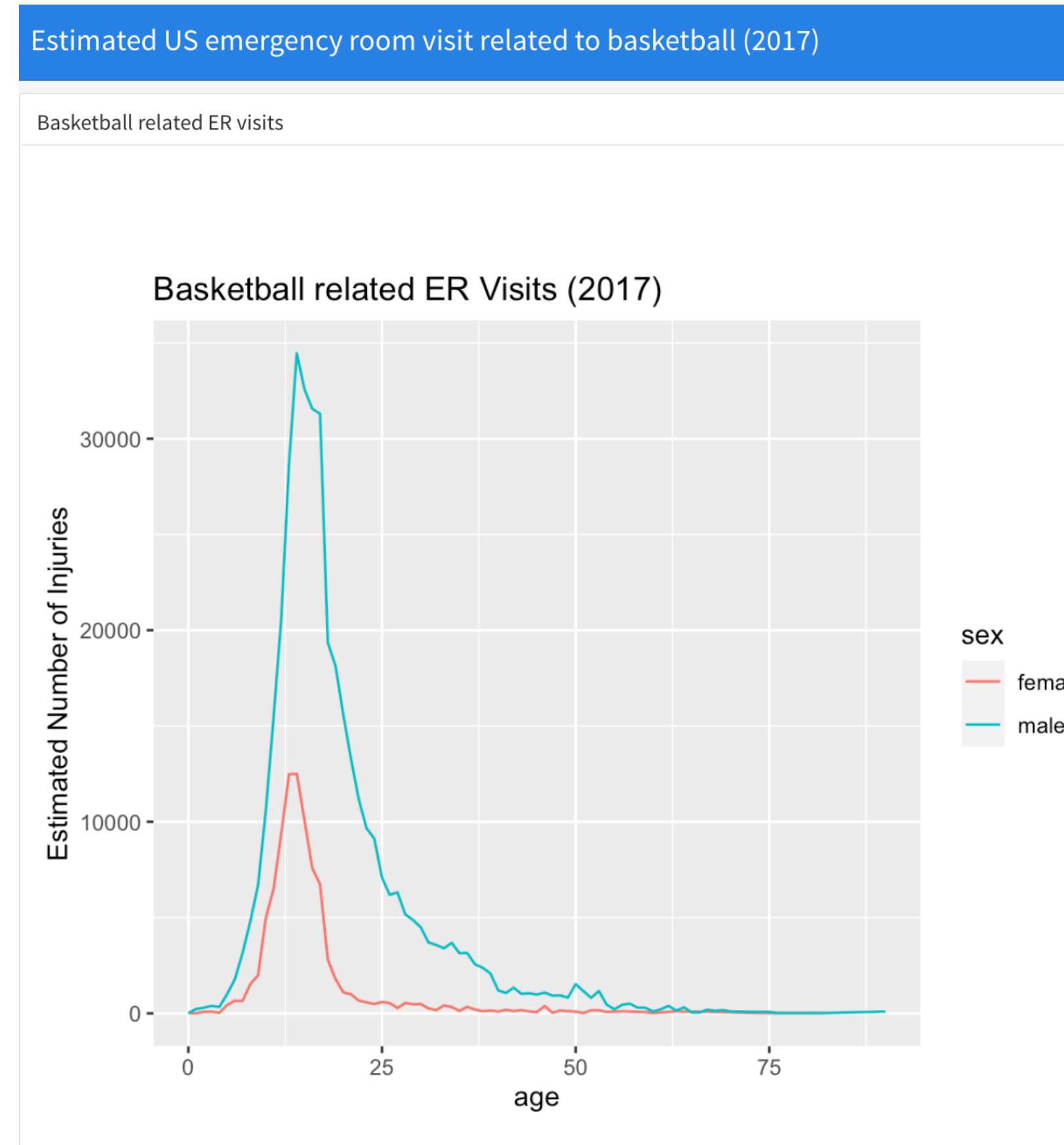
For more layouts and the associated code: <https://pkgs.rstudio.com/flexdashboard/articles/layouts.html>

# 3. Add R (or shiny) code

```
### Basketball related ER visits {data-width=650}

```{r}
basketball_injuries_demographics <- injuries |> filter(prod_code == 1205) |>
  count(age, sex, wt = weight, sort = TRUE)
basketball_injuries_demographics |> ggplot(aes(x = age, y = n, color = sex)) +
  geom_line() +
  labs(title = "Basketball related ER Visits (2017)",
       y = "Estimated Number of Injuries")
```

```

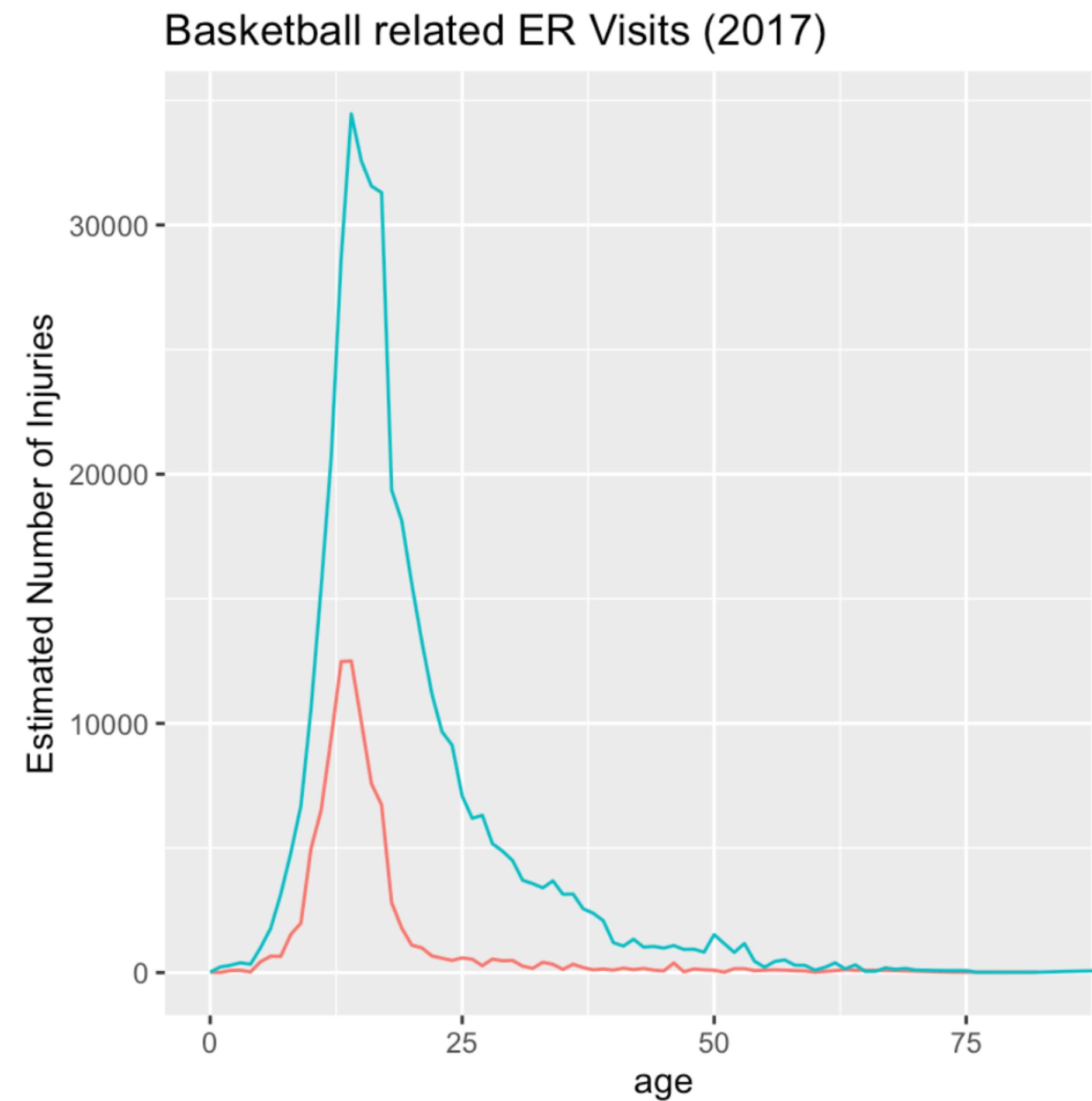


# flexdashboards/01\_static\_flexdashboard.html

Estimated US emergency room visit related to basketball (2017)

[Source Code](#)

Basketball related ER visits

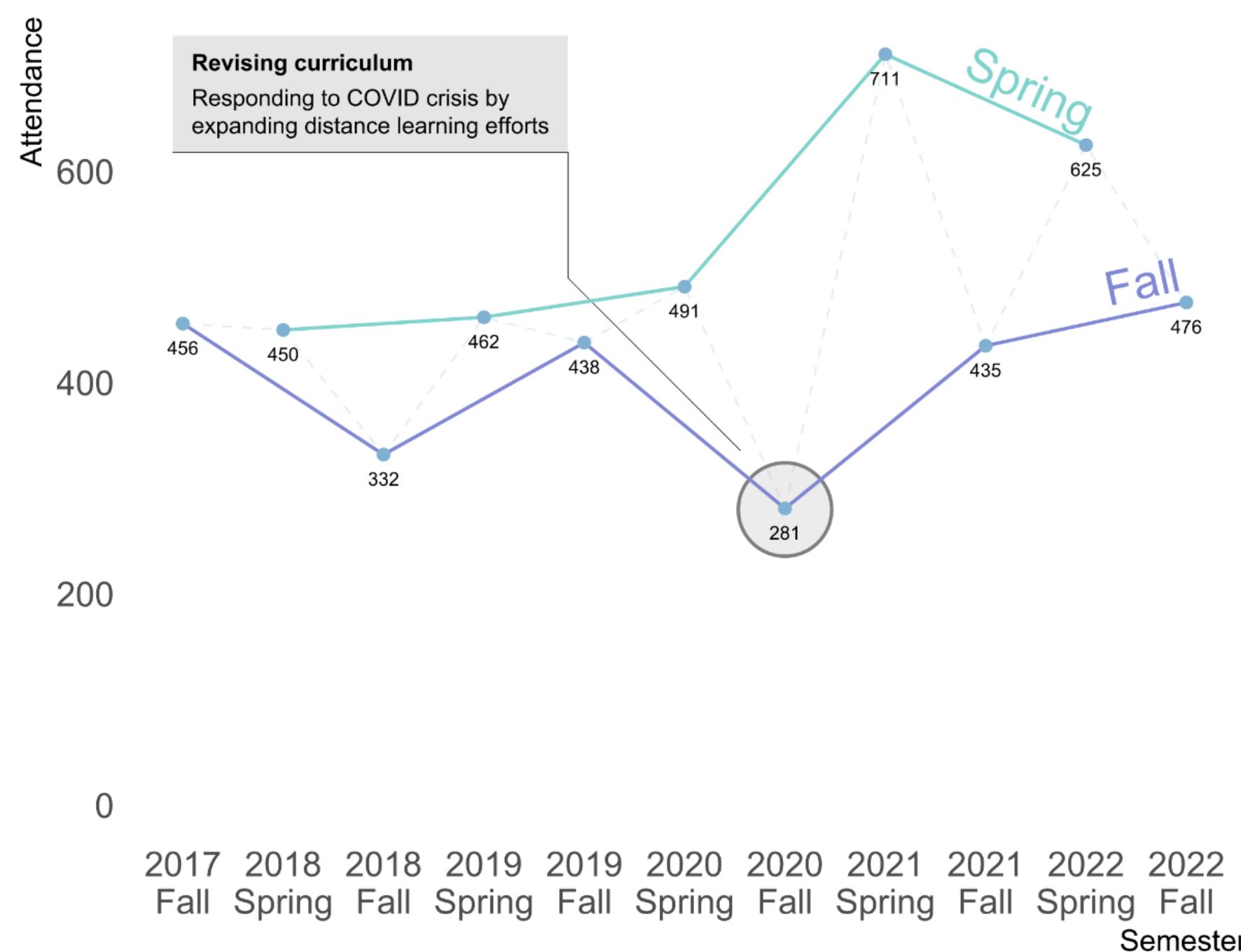


Top 20 demographic groups for basketball related ER visits (2017)

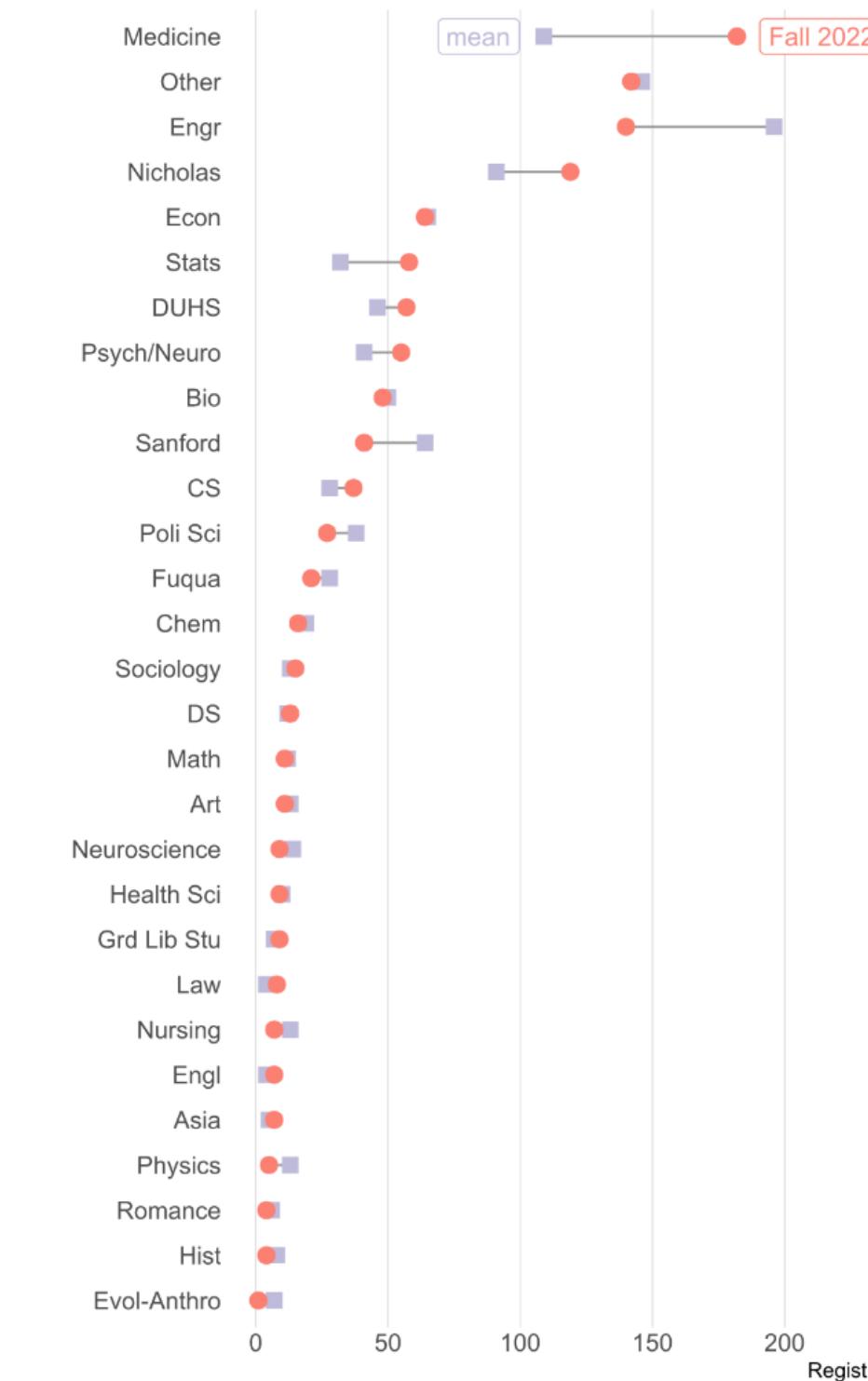
| age | sex    | n         |
|-----|--------|-----------|
| 14  | male   | 34456.708 |
| 15  | male   | 32565.140 |
| 16  | male   | 31558.694 |
| 17  | male   | 31300.324 |
| 13  | male   | 28695.674 |
| 12  | male   | 20628.453 |
| 18  | male   | 19360.890 |
| 19  | male   | 18137.962 |
| 20  | male   | 15600.353 |
| 11  | male   | 15425.811 |
| 21  | male   | 13281.578 |
| 14  | female | 12503.775 |
| 13  | female | 12478.700 |
| 22  | male   | 11176.448 |
| 10  | male   | 10569.878 |
| 15  | female | 10059.873 |
| 23  | male   | 9664.258  |
| 12  | female | 9389.971  |

**476**Current Semester Attendance  
Fall 2022**5,157**Total Attendance  
Fall 2017 to Fall 2022**10,281**Total Registrations  
Fall 2017 to Fall 2022

CDVS attendance over time



Registrants by school/department



# Dynamic Flexdashboards



# 1. YAML Header

```
---
```

```
title: "Estimated US emergency room visit related to basketball (2017)"  
output:  
  flexdashboard::flex_dashboard:  
    orientation: rows  
    source_code: embed  
runtime: shiny
```

Enables interactive shiny code in  
the dashboard

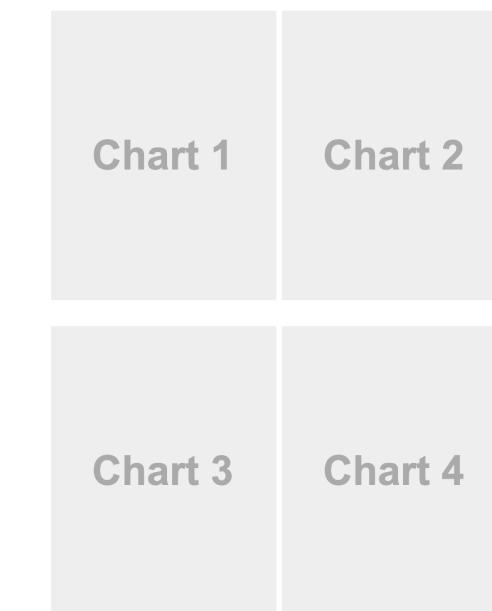
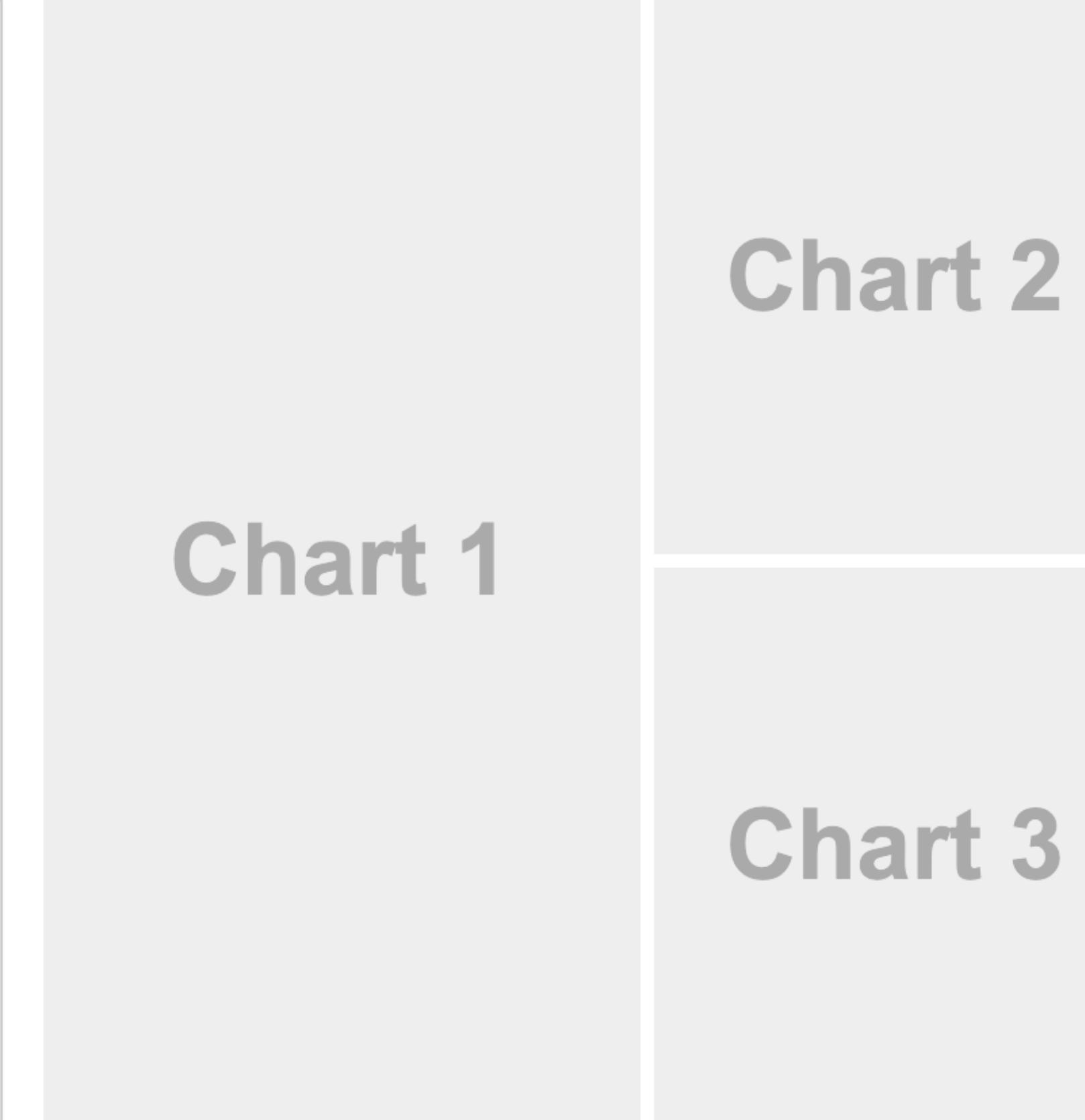
Estimated US emergency room visit related to basketball (2017)

🔗 Source Code

# 2. Pick a layout



```
1 ---  
2 title: "Focal Chart (Left)"  
3 output: flexdashboard::flex_dashboard  
4 ---  
5  
6 Column {data-width=600}  
7 -----  
8  
9 ### Chart 1  
10  
11 `r  
12 ...  
13  
14 Column {data-width=400}  
15 -----  
16  
17 ### Chart 2  
18  
19 `r  
20 ...  
21  
22 ### Chart 3  
23  
24 `r  
25 ...  
26
```



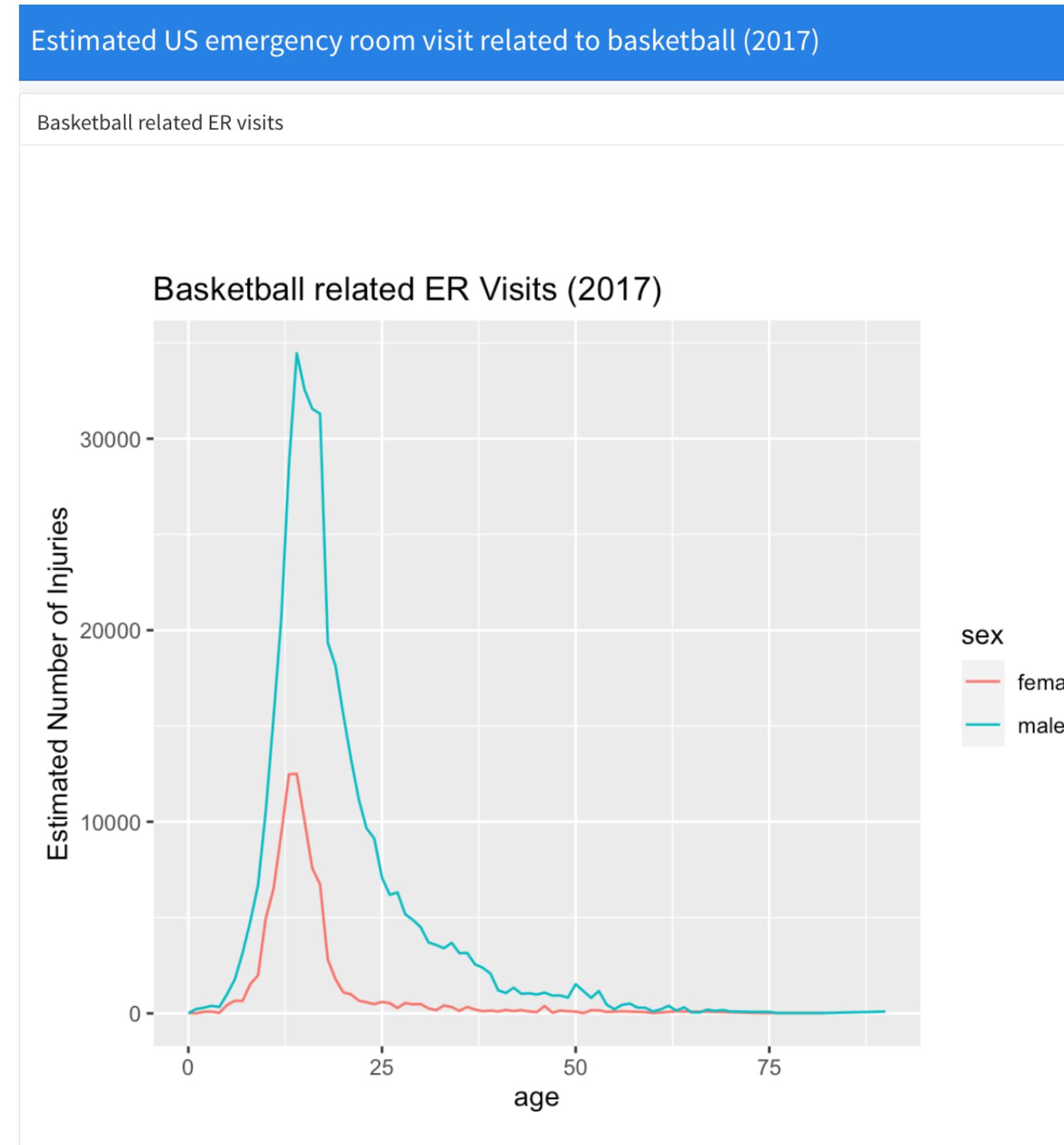
For more layouts and the associated code: <https://pkgs.rstudio.com/flexdashboard/articles/layouts.html>

# 3. Add R (or shiny) code

```
### Basketball related ER visits {data-width=650}

```{r}
basketball_injuries_demographics <- injuries |> filter(prod_code == 1205) |>
  count(age, sex, wt = weight, sort = TRUE)
basketball_injuries_demographics |> ggplot(aes(x = age, y = n, color = sex)) +
  geom_line() +
  labs(title = "Basketball related ER Visits (2017)",
       y = "Estimated Number of Injuries")
```

```



# flexdashboards/02\_interactive\_flexdashboard.Rmd

Estimated US emergency room visits (2017)

Dashboard

Data

Source Code

## NEISS 2017 Explorer

The National Electronic Injury Surveillance System provides a representative sample of US emergency room visits associated with a consumer product. Data in this dashboard cover emergency room visits in 2017.

### Product

basketball (activity, apparel or equipment) ▾

**493,039**

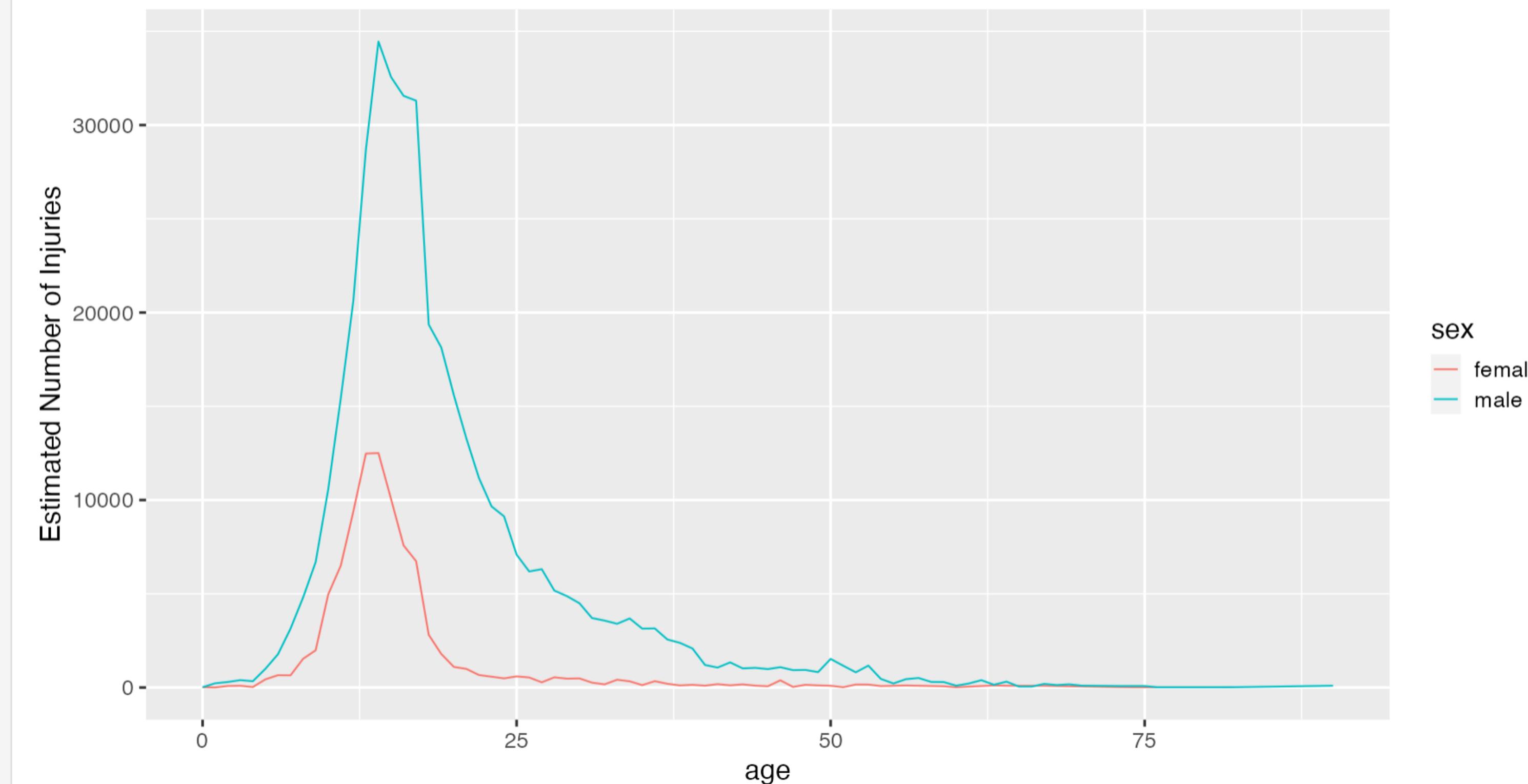
Number of injuries in the US (2017)

**5.13%**

Percentage of injuries with this product



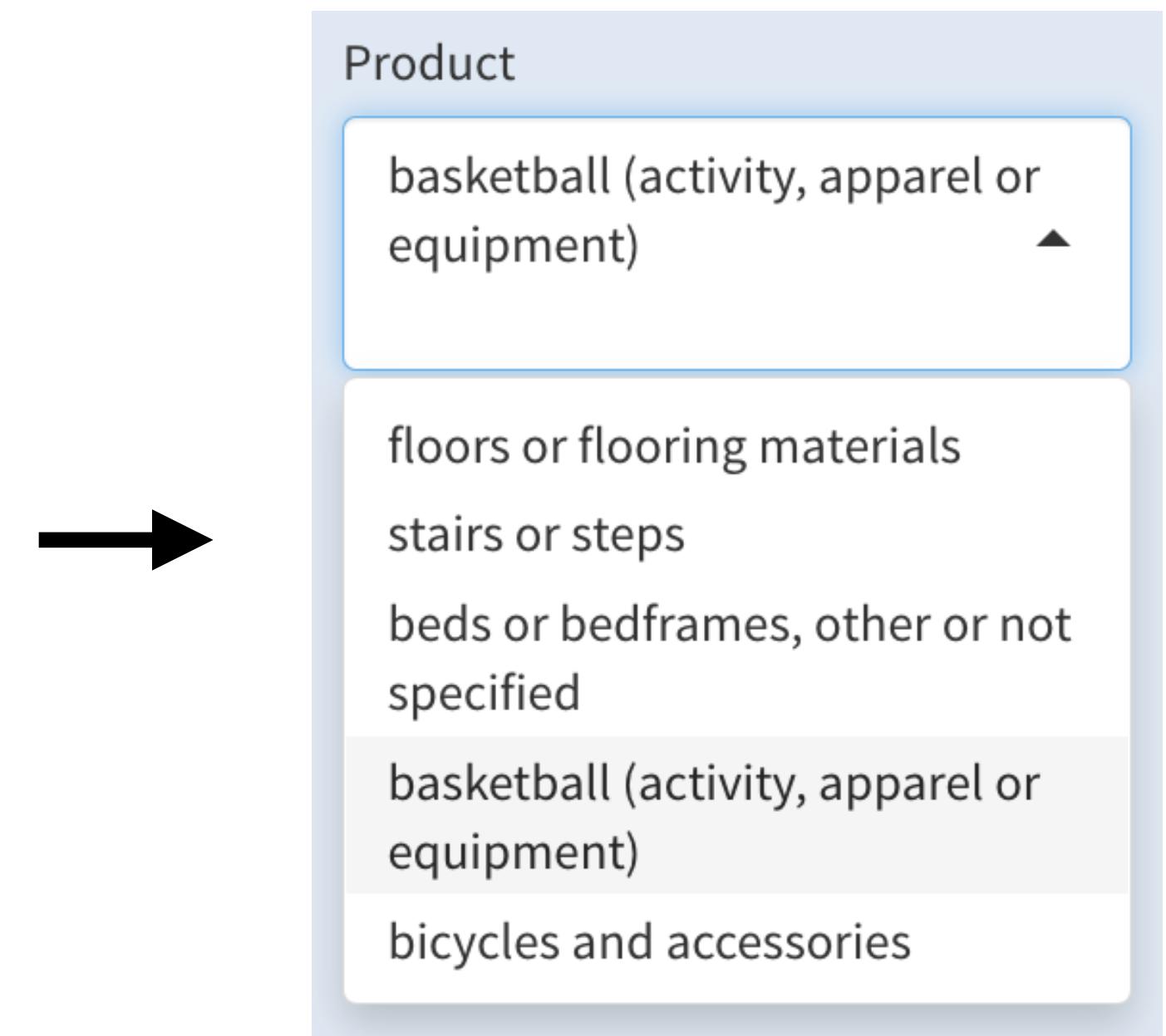
Emergency room visits related to the product



# Let's try it!

## flexdashboards/02\_interactive\_flexdashboard.Rmd

```
selectInput("code", label = "Product",
  choices = c( "floors or flooring materials" = 1807,
  "stairs or steps" = 1842,
  "beds or bedframes, other or not specified" = 4076,
  "basketball (activity, apparel or equipment)" = 1205,
  "bicycles and accessories" = 5040),
  selected = 1205)
```



- Open “02\_interactive\_flexdashboard.Rmd”
- Change the default injury to “bicycles and accessories” (Hint: code = 5040)
- Click “Run Document”
- Change the default view option (next to “Run Document”) to “Preview Window” or “Preview Pane”
- Click “Run Document” again

03:00

# Building apps in



# **Building apps in**



“But why not just use flexdashboards?”

# Google trend index

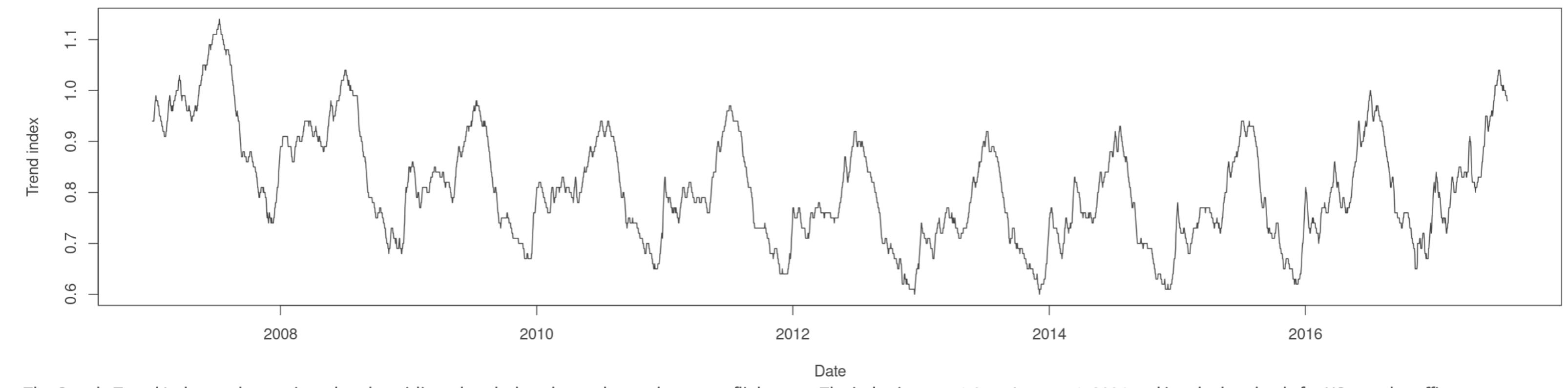
Trend index

Travel

Date range

2007-01-01 to 2017-07-31

Overlay smooth trend line



[bit.ly/g-trend](http://bit.ly/g-trend)

# Let's try it!

- Open a new file, and choose “Shiny Web App”
- Open (it should be open) the app.R file, and click “Run App”
- Select another output option by using the dropdown menu on “Run app”
- Click the stop sign icon in the Console to stop the app
- Change the default view option (next to “Run Document”) to “Preview Window” or “Preview Pane”

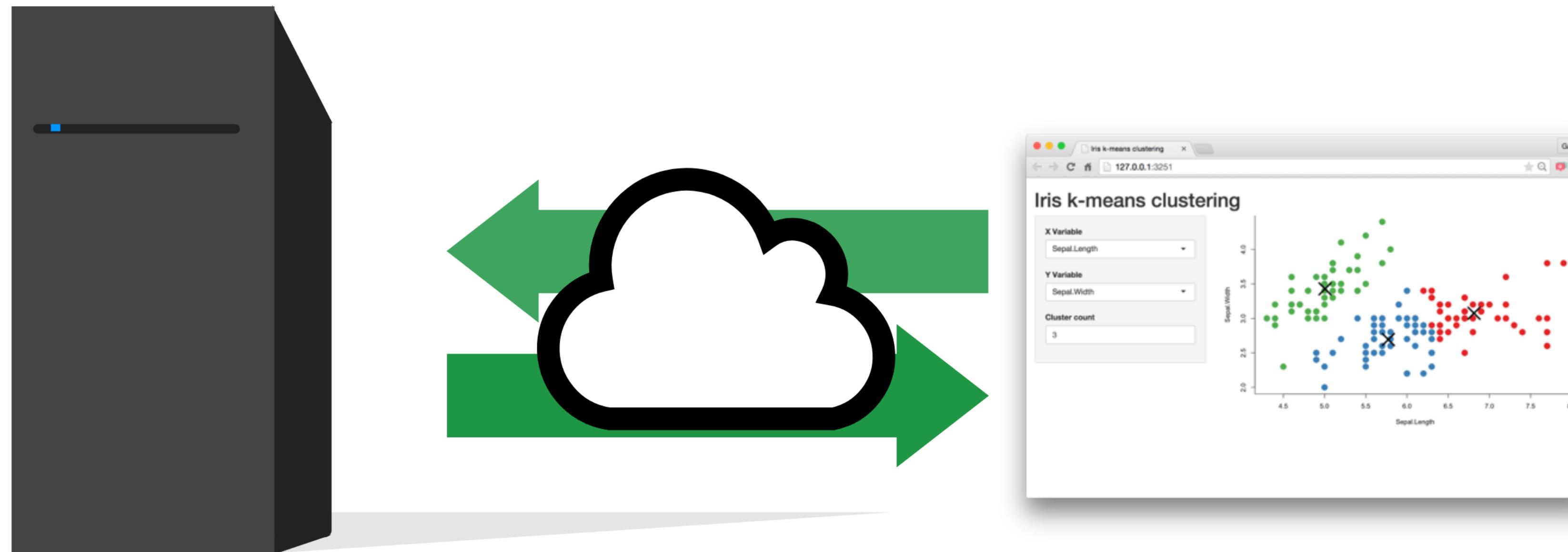
03:00

**So how does shiny work?**

# Every Shiny app is maintained by a computer running R



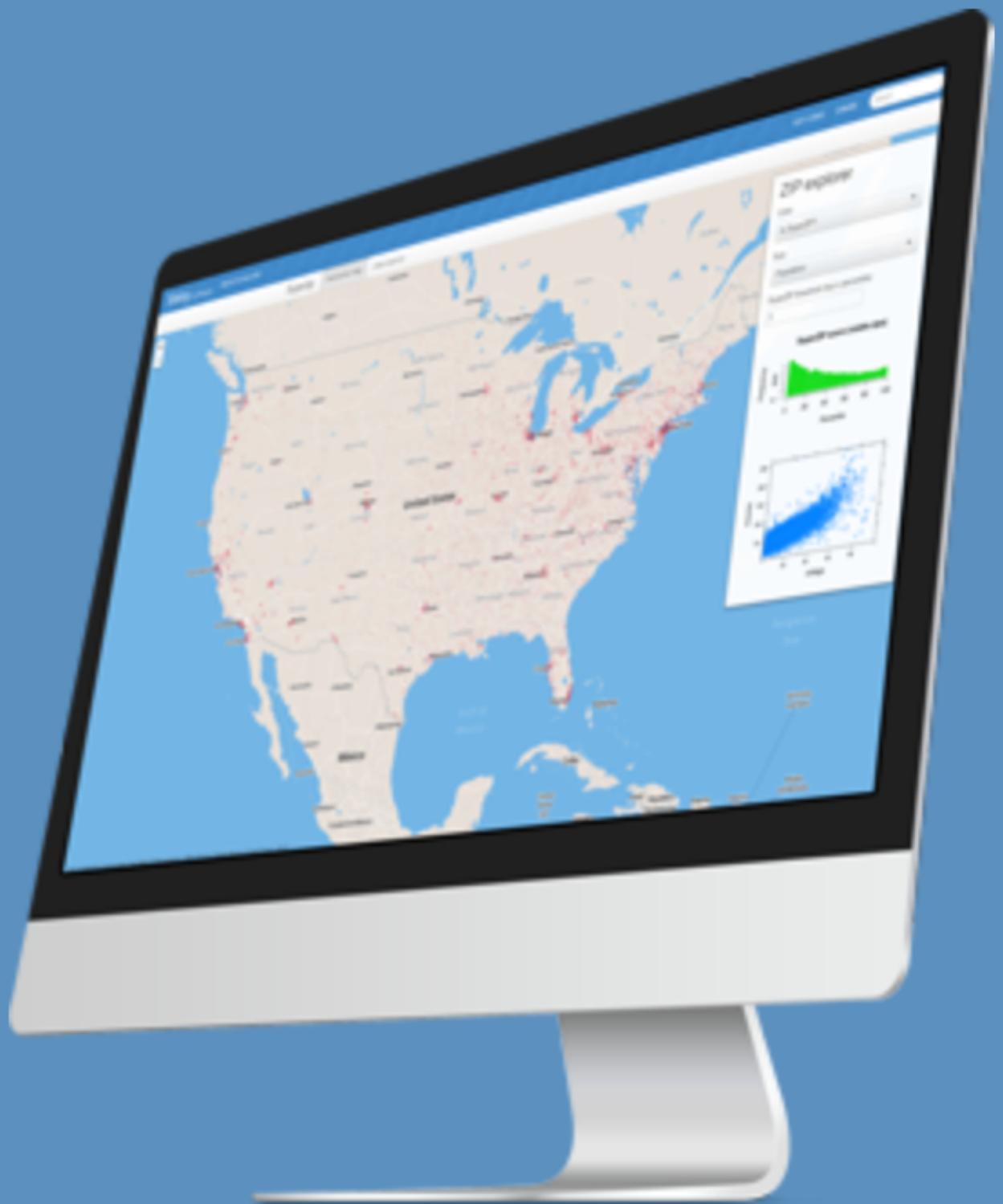
# Every Shiny app is maintained by a computer running R



# Share your Shiny Applications Online

Deploy your Shiny applications on the Web in minutes

Sign Up



<https://shinyapps.io>



Server Instructions

User Interface (UI)



# Shiny apps have three elements

1. A **user interface** or “ui”

```
library(shiny)
```

```
ui <- fluidPage(  
  )
```

2. A **server** component

```
server <- function(input, output, session) {  
  }
```

3. A **shinyApp** function

```
shinyApp(ui, server)
```

# Building a NEISS app using Shiny

## National Electronic Injury Surveillance System (NEISS)

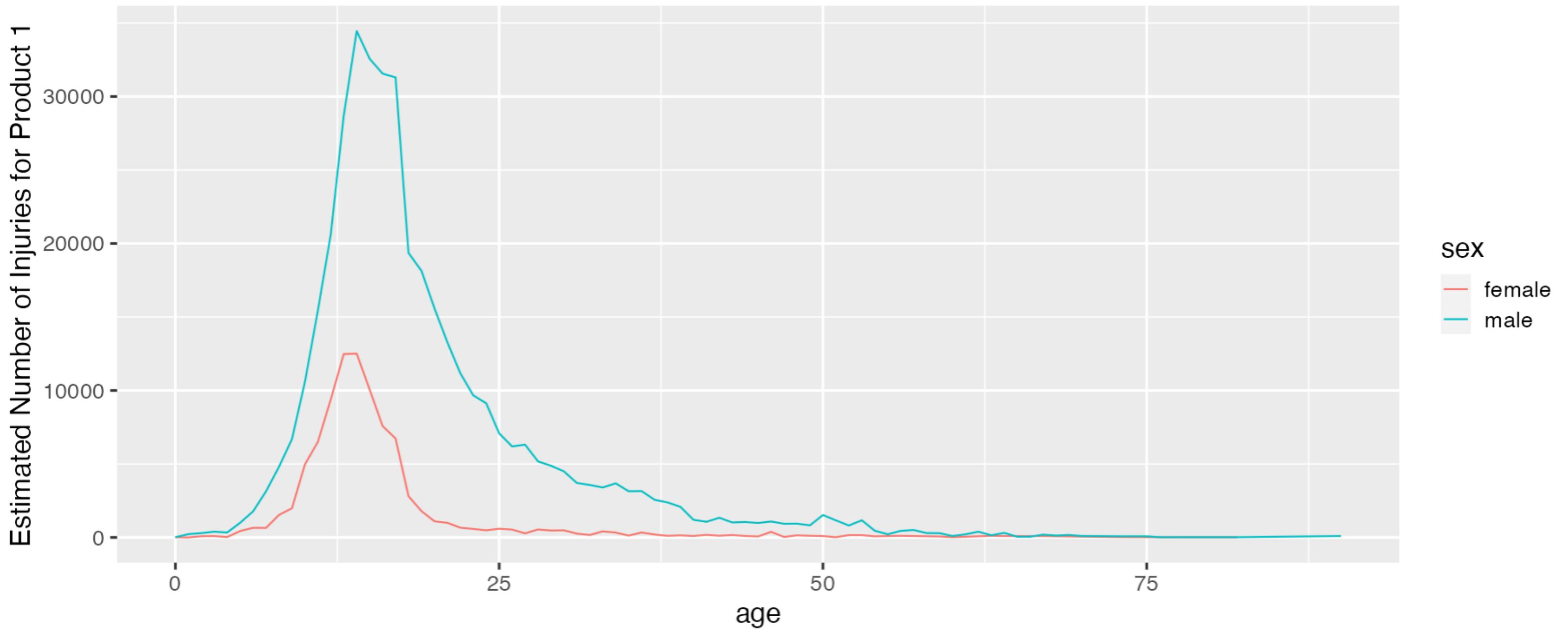
- Focus: consumer product related injuries
- Sample: nationally representative sample of hospitals
- Coverage: United States, 2017
- Source: Hadley Wickham (<https://github.com/hadley/neiss>)



# NEISS Demographic Comparisons by Product

Product 1

- basketball (activity, apparel or equipment)
- floors or flooring materials
- stairs or steps
- beds or bedframes, other or not specified
- basketball (activity, apparel or equipment)
- bicycles and accessories



shiny/neiss\_shiny/app.r

# App outline

```
library(shiny)
library(tidyverse)

injuries <- read_tsv("../..../data/NEISS/injuries.tsv")
```

**Header**

```
ui <- fluidPage(
  )
```

**User Interface (ui)**

```
server <- function(input, output, session) {
}
```

**Server function**

```
shinyApp(ui = ui, server = server)
```

**shinyApp**

# App outline

```
library(shiny)
library(tidyverse)

injuries <- read_tsv("../..../data/NEISS/injuries.tsv")
```

## Header

Run once  
when  
app  
launches

```
ui <- fluidPage(
  )
```

## User Interface (ui)

```
server <- function(input, output, session) {
  }
```

## Server function

```
shinyApp(ui = ui, server = server)
```

## shinyApp

# App outline

```
library(shiny)
library(tidyverse)

injuries <- read_tsv("../..../data/NEISS/injuries.tsv")
```

**Header**

```
ui <- fluidPage(
  )
```

**User Interface (ui)**

```
server <- function(input, output, session) {
  }
```

**Server function**

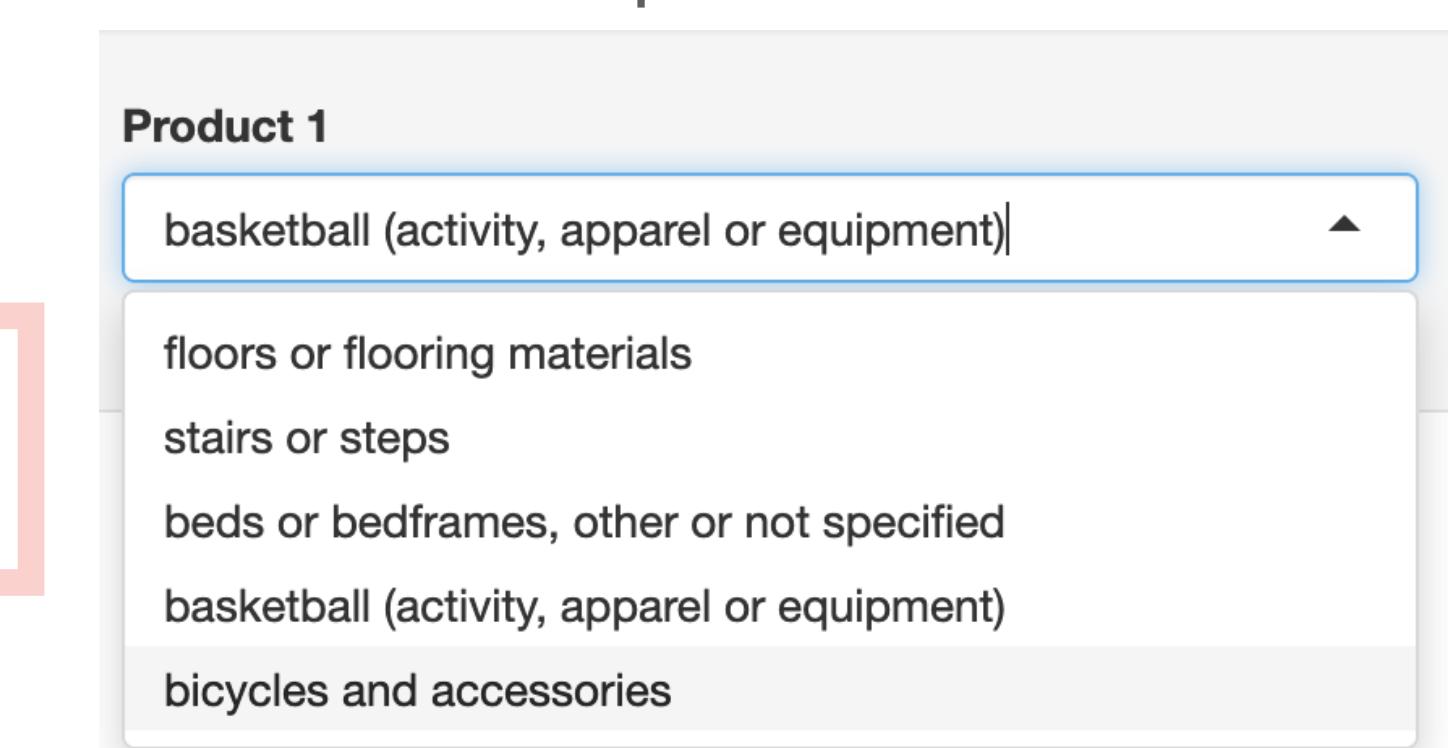
Runs  
once for  
each  
user

```
shinyApp(ui = ui, server = server)
```

**shinyApp**

# user interface

```
ui <- fluidPage(← Create a responsive page
  # Application title ← Add a title
  titlePanel("NEISS Demographic Comparisons by Product"), ←
  # Sidebar with a slider input for number of bins ← We want a layout with a sidebar
  sidebarLayout(←
    sidebarPanel(
      selectInput(inputID = "code", label = "Product 1", ←
        choices = c( "floors or flooring materials" = 1807, ←
          "stairs or steps" = 1842, ←
          "beds or bedframes, other or not specified" = 4076, ←
          "basketball (activity, apparel or equipment)" = 1205, ←
          "bicycles and accessories" = 5040), ←
        selected = 1205), ←
      ), ←
      input$code
    ),
    mainPanel(
      plotOutput("injurePlot1"), ← We create an output for our plot.
    )
)
```



**Let's add an input!**

# Shiny offers a range of inputs!

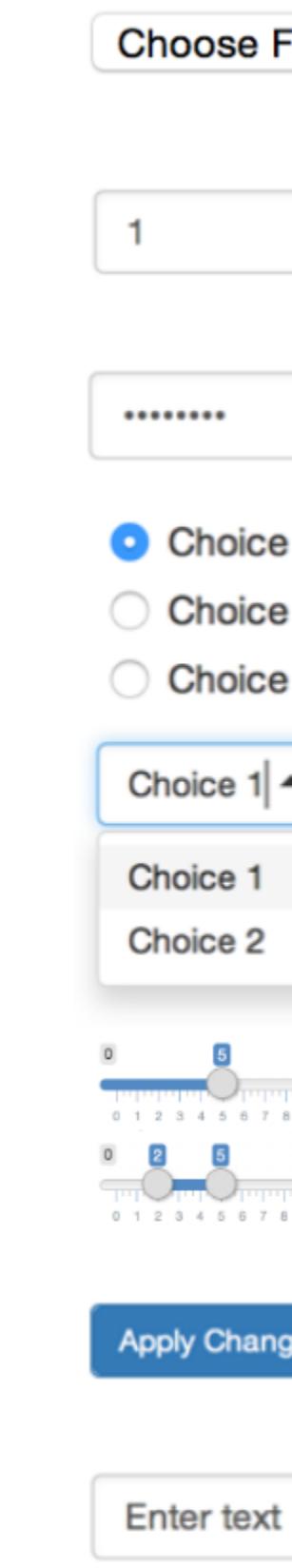


## Inputs

Collect values from the user.

Access the current value of an input object with `input$<inputId>`. Input values are **reactive**.

|                                                                                                                                                                                       |                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Action                                                                                                                                                                                | <code>actionButton(inputId, label, icon, width, ...)</code>                                                                                                                  |
| Link                                                                                                                                                                                  | <code>actionLink(inputId, label, icon, ...)</code>                                                                                                                           |
| <input checked="" type="checkbox"/> Choice 1<br><input checked="" type="checkbox"/> Choice 2<br><input type="checkbox"/> Choice 3<br><br><input checked="" type="checkbox"/> Check me | <code>checkboxGroupInput(inputId, label, choices, selected, inline, width, choiceNames, choiceValues)</code><br><br><code>checkboxInput(inputId, label, value, width)</code> |
|                                                                                                                                                                                       | <code>dateInput(inputId, label, value, min, max, format, startview, weekstart, language, width, autoclose, datesdisabled, daysofweekdisabled)</code>                         |



`fileInput(inputId, label, multiple, accept, width, buttonLabel, placeholder)`

`numericInput(inputId, label, value, min, max, step, width)`

`passwordInput(inputId, label, value, width, placeholder)`

`radioButtons(inputId, label, choices, selected, inline, width, choiceNames, choiceValues)`

`selectInput(inputId, label, choices, selected, multiple, selectize, width, size)`  
Also `selectizeInput()`

`sliderInput(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post, timeFormat, timezone, dragRange)`

`submitButton(text, icon, width)`  
(Prevent reactions for entire app)

`textInput(inputId, label, value, width, placeholder)` Also `textAreaInput()`

<https://shiny.rstudio.com/articles/cheatsheet.html>

# Exercise: Let's add a slider

shiny/neiss\_shiny/app.r

## Instructions

- Open “shiny/neiss\_shiny/app.r”
- Add the code for a **sliderInput** (see pink box)
- Enter the following properties:
  - inputID = “text\_size”
  - label = “Magnify Labels”
  - min = 1
  - max = 25
  - value = 16
  - step = 1

```
# Sidebar with a slider input for number of bins
sidebarLayout(
  sidebarPanel(
    selectInput(inputId = "code",
                label = "Product 1",
                choices = c( "floors " = 1807,
                           selected = 1205),
    hr(),
    sliderInput(inputId = "text_size",
                label = "",
                min = ,
                max = ,
                value = ,
                step =
),
),
)
```

03:00

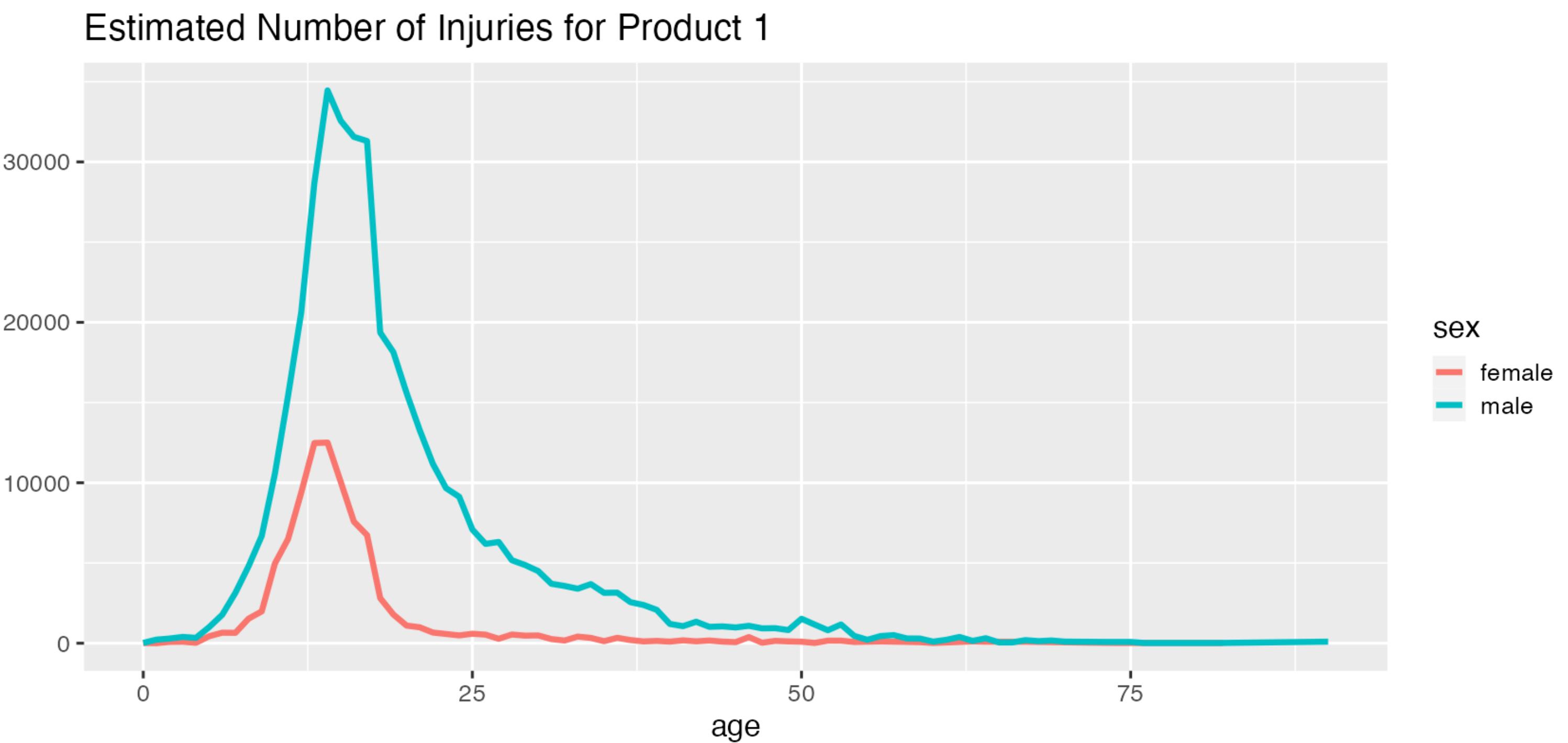
# NEISS Demographic Comparisons by Product

**Product 1**

basketball (activity, apparel or equipment) ▾

**Magnify labels**

1 16 25



shiny/neiss\_shiny\_ui\_change\_themes/app.r

# Server

```
server <- function(input, output) {  
  injuries_demographics <- reactive ({  
    injuries %> filter(prod_code == input$code) %>  
    count(age, sex, wt = weight, sort = TRUE)  
  })  
  
  output$injurePlot1 <- renderPlot({injuries_demographics() %>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = 16)  
  })  
}
```

Subset the data

# Server

```
server <- function(input, output) {  
  
  injuries_demographics <- reactive ({  
    injuries |> filter(prod_code == input$code) |>  
    count(age, sex, wt = weight, sort = TRUE)  
    })  
  
  output$injurePlot1 <- renderPlot({injuries_demographics() |>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = 16)  
  })  
}
```

input\$code  
←————

Product 1

basketball (activity, apparel or equipment) ▾

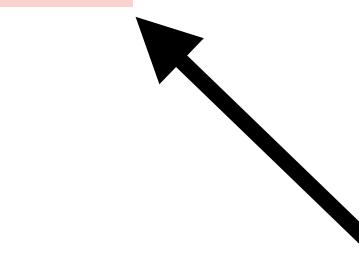
# Server

```
server <- function(input, output) {  
  injuries_demographics <- reactive ({  
    injuries |> filter(prod_code == input$code) |>  
    count(age, sex, wt = weight, sort = TRUE)  
  })  
  
  output$injurePlot1 <- renderPlot({injuries_demographics() |>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = 16)  
  })  
}
```

Create a graph

# Server

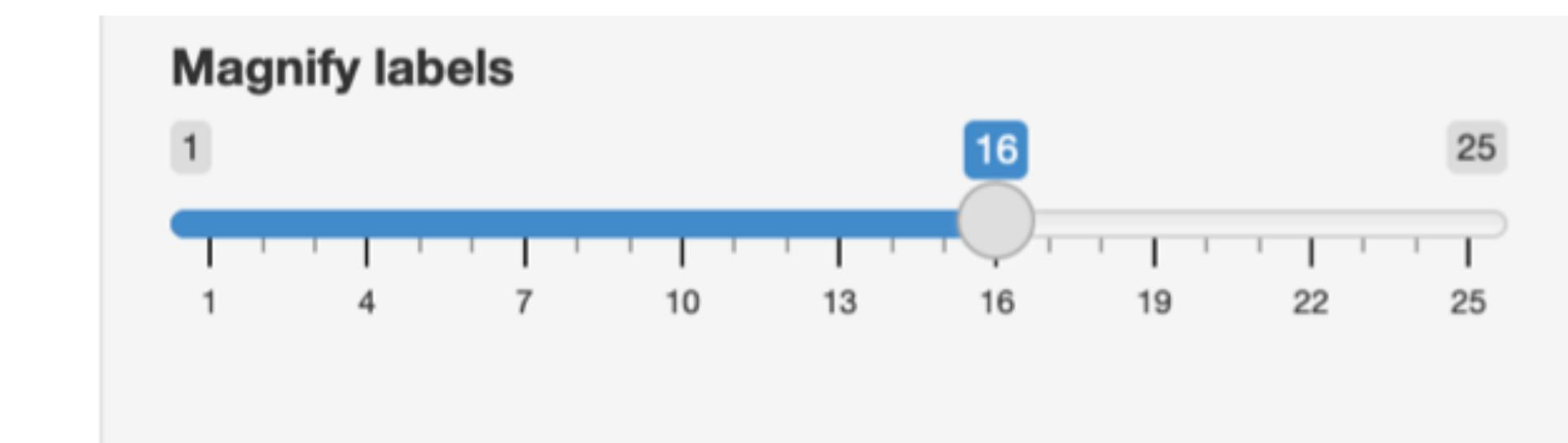
```
server <- function(input, output) {  
  injuries_demographics <- reactive ({  
    injuries %> filter(prod_code == input$code) %>  
    count(age, sex, wt = weight, sort = TRUE)  
  })  
  
  output$injurePlot1 <- renderPlot({injuries_demographics() %>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = 16)  
  })  
}
```



We are setting the size of the axis labels and titles here. We want the slider to control this.

# Server

```
server <- function(input, output) {  
  injuries_demographics <- reactive ({  
    injuries %> filter(prod_code == input$code) %>  
    count(age, sex, wt = weight, sort = TRUE)  
  })  
  
  output$injurePlot1 <- renderPlot({injuries_demographics() %>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = input$text_size)  
})  
}
```



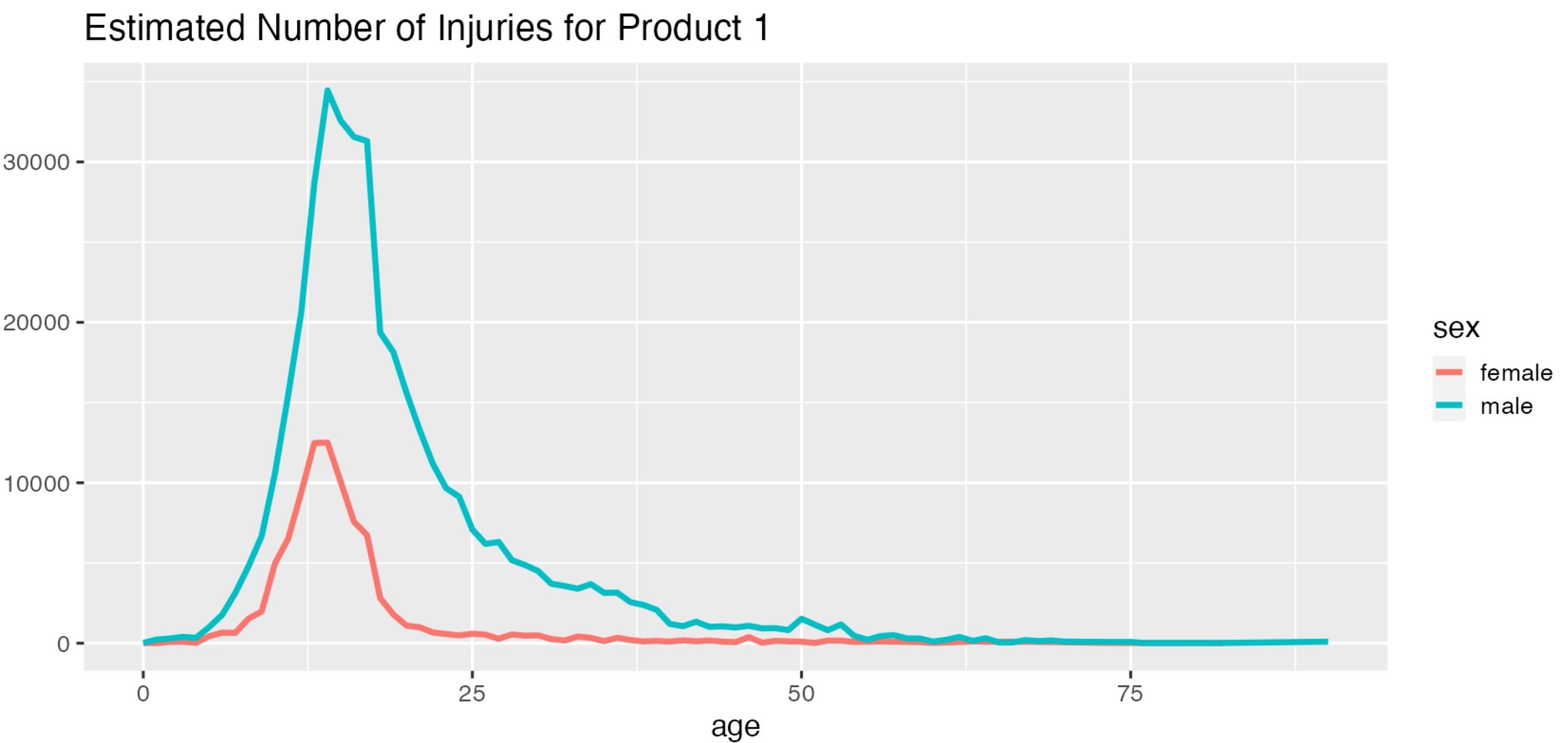
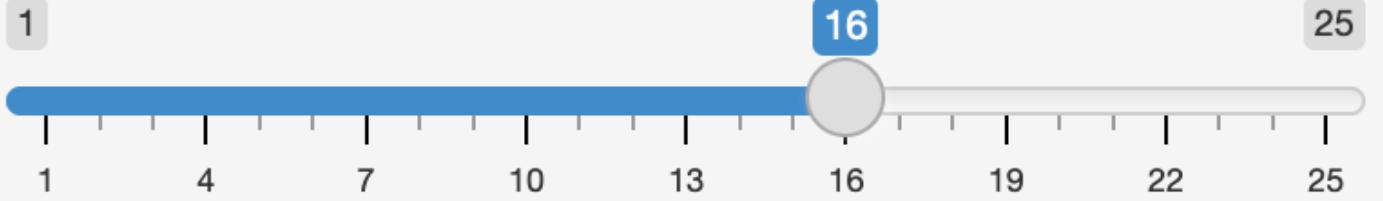
# NEISS Demographic Comparisons by Product

**Product 1**

basketball (activity, apparel or equipment) ▾

**Magnify labels**

1 16 25



shiny/neiss\_shiny\_ui\_change\_themes/app.R

# shinyApp

```
# Run the application  
shinyApp(ui = ui, server = server)
```

# Questions?

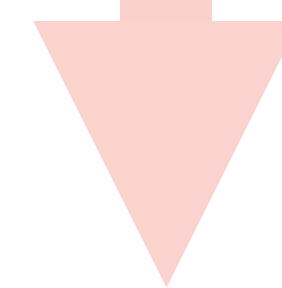
# **Reactivity**

**(Shiny behind the interface)**



# Inputs

```
selectInput(inputID = "code",  
           label = "Product 1",  
           choices = c("floors..." = 1807,  
                      "Stairs..." = 1842,  
                      "Beds..." = 4076,  
                      "basketball" = 1205,  
                      "bicycles" = 5040),  
           selected = 1205),
```



input\$code

The figure shows three separate dropdown menus, each labeled 'Product 1'. The first dropdown has the option 'basketball (activity, apparel or equipment)' selected. The second dropdown has the option 'bicycles and accessories' selected. The third dropdown has the option 'stairs or steps' selected.

input\$code=1205

input\$code=5040

input\$code=1842

We are building a list of inputs as we select items in the user interface.

# Outputs

Reactivity occurs when we use one of our inputs to create (render) an output.

```
server <- function(input, output) {  
  output$injurePlot1 <- renderPlot({injuries_demographics() |>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = input$text_size)  
}
```

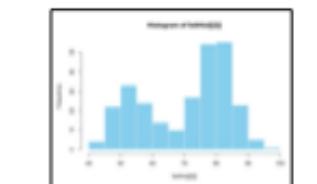
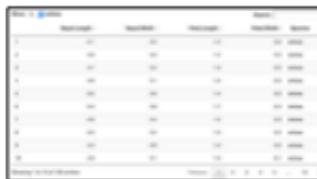


output\$injurePlot1

# Shiny offers a range of outputs!

## Outputs

`render*`() and `*Output()` functions work together to add R output to the UI.



```
Data Frame: 150 rows, 5 variables
```

|     | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species     |
|-----|--------------|-------------|--------------|-------------|-------------|
| 1   | 4.3          | 3.0         | 1.4          | 0.2         | Iris-setosa |
| 2   | 4.4          | 3.4         | 1.3          | 0.2         | Iris-setosa |
| 3   | 4.5          | 3.0         | 1.5          | 0.2         | Iris-setosa |
| 4   | 4.5          | 3.0         | 1.6          | 0.2         | Iris-setosa |
| 5   | 4.6          | 3.6         | 1.4          | 0.2         | Iris-setosa |
| 6   | 4.6          | 3.0         | 1.5          | 0.2         | Iris-setosa |
| 7   | 4.7          | 3.2         | 1.3          | 0.2         | Iris-setosa |
| 8   | 4.7          | 3.7         | 1.5          | 0.2         | Iris-setosa |
| 9   | 4.8          | 3.4         | 1.6          | 0.2         | Iris-setosa |
| 10  | 4.8          | 3.0         | 1.4          | 0.2         | Iris-setosa |
| 11  | 4.9          | 3.3         | 1.4          | 0.2         | Iris-setosa |
| 12  | 4.9          | 3.9         | 1.7          | 0.4         | Iris-setosa |
| 13  | 5.0          | 3.4         | 1.4          | 0.2         | Iris-setosa |
| 14  | 5.0          | 3.4         | 1.5          | 0.2         | Iris-setosa |
| 15  | 5.0          | 3.4         | 1.6          | 0.2         | Iris-setosa |
| 16  | 5.1          | 3.7         | 1.5          | 0.2         | Iris-setosa |
| 17  | 5.1          | 3.0         | 1.9          | 0.2         | Iris-setosa |
| 18  | 5.1          | 3.4         | 1.7          | 0.4         | Iris-setosa |
| 19  | 5.2          | 3.5         | 1.5          | 0.2         | Iris-setosa |
| 20  | 5.2          | 3.4         | 1.6          | 0.2         | Iris-setosa |
| 21  | 5.3          | 3.7         | 1.5          | 0.2         | Iris-setosa |
| 22  | 5.4          | 3.9         | 1.7          | 0.4         | Iris-setosa |
| 23  | 5.4          | 3.7         | 1.5          | 0.2         | Iris-setosa |
| 24  | 5.5          | 4.0         | 1.6          | 0.2         | Iris-setosa |
| 25  | 5.5          | 3.5         | 1.7          | 0.2         | Iris-setosa |
| 26  | 5.6          | 3.5         | 1.9          | 0.2         | Iris-setosa |
| 27  | 5.6          | 3.0         | 1.8          | 0.2         | Iris-setosa |
| 28  | 5.7          | 3.8         | 1.7          | 0.3         | Iris-setosa |
| 29  | 5.7          | 3.0         | 1.8          | 0.2         | Iris-setosa |
| 30  | 5.8          | 4.0         | 1.8          | 0.2         | Iris-setosa |
| 31  | 5.8          | 3.7         | 1.9          | 0.2         | Iris-setosa |
| 32  | 5.9          | 3.3         | 1.8          | 0.2         | Iris-setosa |
| 33  | 5.9          | 3.0         | 2.0          | 0.2         | Iris-setosa |
| 34  | 6.0          | 3.4         | 1.8          | 0.2         | Iris-setosa |
| 35  | 6.0          | 3.0         | 2.0          | 0.2         | Iris-setosa |
| 36  | 6.1          | 3.6         | 1.9          | 0.2         | Iris-setosa |
| 37  | 6.1          | 3.0         | 2.1          | 0.2         | Iris-setosa |
| 38  | 6.2          | 3.4         | 2.0          | 0.2         | Iris-setosa |
| 39  | 6.2          | 3.0         | 2.2          | 0.2         | Iris-setosa |
| 40  | 6.3          | 3.3         | 2.0          | 0.2         | Iris-setosa |
| 41  | 6.3          | 3.0         | 2.3          | 0.2         | Iris-setosa |
| 42  | 6.4          | 3.5         | 2.0          | 0.2         | Iris-setosa |
| 43  | 6.4          | 3.0         | 2.4          | 0.2         | Iris-setosa |
| 44  | 6.5          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 45  | 6.5          | 3.0         | 2.5          | 0.2         | Iris-setosa |
| 46  | 6.6          | 3.4         | 2.0          | 0.2         | Iris-setosa |
| 47  | 6.6          | 3.0         | 2.6          | 0.2         | Iris-setosa |
| 48  | 6.7          | 3.5         | 2.0          | 0.2         | Iris-setosa |
| 49  | 6.7          | 3.0         | 2.7          | 0.2         | Iris-setosa |
| 50  | 6.8          | 3.7         | 2.0          | 0.2         | Iris-setosa |
| 51  | 6.8          | 3.0         | 2.8          | 0.2         | Iris-setosa |
| 52  | 6.9          | 3.5         | 2.0          | 0.2         | Iris-setosa |
| 53  | 6.9          | 3.0         | 2.9          | 0.2         | Iris-setosa |
| 54  | 7.0          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 55  | 7.0          | 3.0         | 3.0          | 0.2         | Iris-setosa |
| 56  | 7.1          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 57  | 7.1          | 3.0         | 3.1          | 0.2         | Iris-setosa |
| 58  | 7.2          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 59  | 7.2          | 3.0         | 3.2          | 0.2         | Iris-setosa |
| 60  | 7.3          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 61  | 7.3          | 3.0         | 3.3          | 0.2         | Iris-setosa |
| 62  | 7.4          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 63  | 7.4          | 3.0         | 3.4          | 0.2         | Iris-setosa |
| 64  | 7.5          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 65  | 7.5          | 3.0         | 3.5          | 0.2         | Iris-setosa |
| 66  | 7.6          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 67  | 7.6          | 3.0         | 3.6          | 0.2         | Iris-setosa |
| 68  | 7.7          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 69  | 7.7          | 3.0         | 3.7          | 0.2         | Iris-setosa |
| 70  | 7.8          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 71  | 7.8          | 3.0         | 3.8          | 0.2         | Iris-setosa |
| 72  | 7.9          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 73  | 7.9          | 3.0         | 3.9          | 0.2         | Iris-setosa |
| 74  | 8.0          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 75  | 8.0          | 3.0         | 4.0          | 0.2         | Iris-setosa |
| 76  | 8.1          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 77  | 8.1          | 3.0         | 4.1          | 0.2         | Iris-setosa |
| 78  | 8.2          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 79  | 8.2          | 3.0         | 4.2          | 0.2         | Iris-setosa |
| 80  | 8.3          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 81  | 8.3          | 3.0         | 4.3          | 0.2         | Iris-setosa |
| 82  | 8.4          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 83  | 8.4          | 3.0         | 4.4          | 0.2         | Iris-setosa |
| 84  | 8.5          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 85  | 8.5          | 3.0         | 4.5          | 0.2         | Iris-setosa |
| 86  | 8.6          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 87  | 8.6          | 3.0         | 4.6          | 0.2         | Iris-setosa |
| 88  | 8.7          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 89  | 8.7          | 3.0         | 4.7          | 0.2         | Iris-setosa |
| 90  | 8.8          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 91  | 8.8          | 3.0         | 4.8          | 0.2         | Iris-setosa |
| 92  | 8.9          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 93  | 8.9          | 3.0         | 4.9          | 0.2         | Iris-setosa |
| 94  | 9.0          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 95  | 9.0          | 3.0         | 5.0          | 0.2         | Iris-setosa |
| 96  | 9.1          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 97  | 9.1          | 3.0         | 5.1          | 0.2         | Iris-setosa |
| 98  | 9.2          | 3.6         | 2.0          | 0.2         | Iris-setosa |
| 99  | 9.2          | 3.0         | 5.2          | 0.2         | Iris-setosa |
| 100 | 9.3          | 3.6         | 2.0          | 0.2         | Iris-setosa |

foo



`renderTable(expr, striped, hover, bordered, spacing, width, align, rownames, colnames, digits, na, ..., env, quoted, outputArgs)`

`renderText(expr, env, quoted, outputArgs, sep)`

`renderUI(expr, env, quoted, outputArgs)`

`DT::renderDataTable(expr, options, searchDelay, callback, escape, env, quoted, outputArgs)`

`renderImage(expr, env, quoted, deleteFile, outputArgs)`

`renderPlot(expr, width, height, res, ..., alt, env, quoted, execOnResize, outputArgs)`

`renderPrint(expr, env, quoted, width, outputArgs)`

`renderText(expr, env, quoted, outputArgs, sep)`

`renderUI(expr, env, quoted, outputArgs)`

`dataTableOutput(outputId)`

`imageOutput(outputId, width, height, click, dblclick, hover, brush, inline)`

`plotOutput(outputId, width, height, click, dblclick, hover, brush, inline)`

`verbatimTextOutput(outputId, placeholder)`

`tableOutput(outputId)`

`textOutput(outputId, container, inline)`

`uiOutput(outputId, inline, container, ...)`

`htmlOutput(outputId, inline, container, ...)`

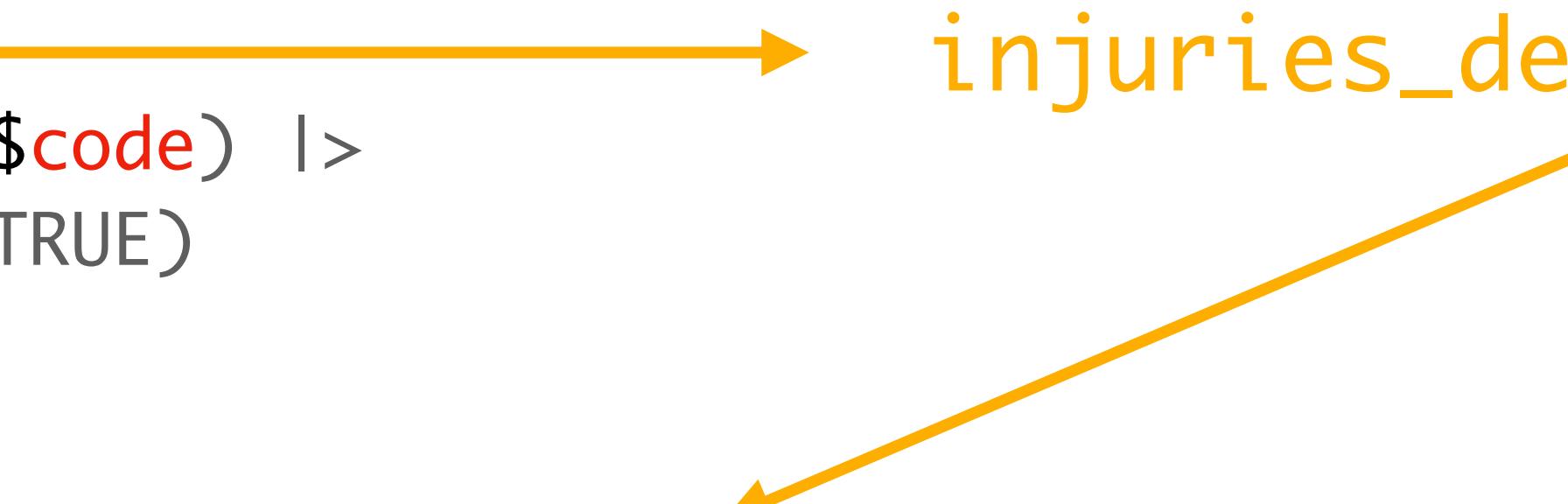
These are the core output types. See [htmlwidgets.org](https://htmlwidgets.org) for many more options.

<https://shiny.rstudio.com/articles/cheatsheet.html>

# Reactive Outputs

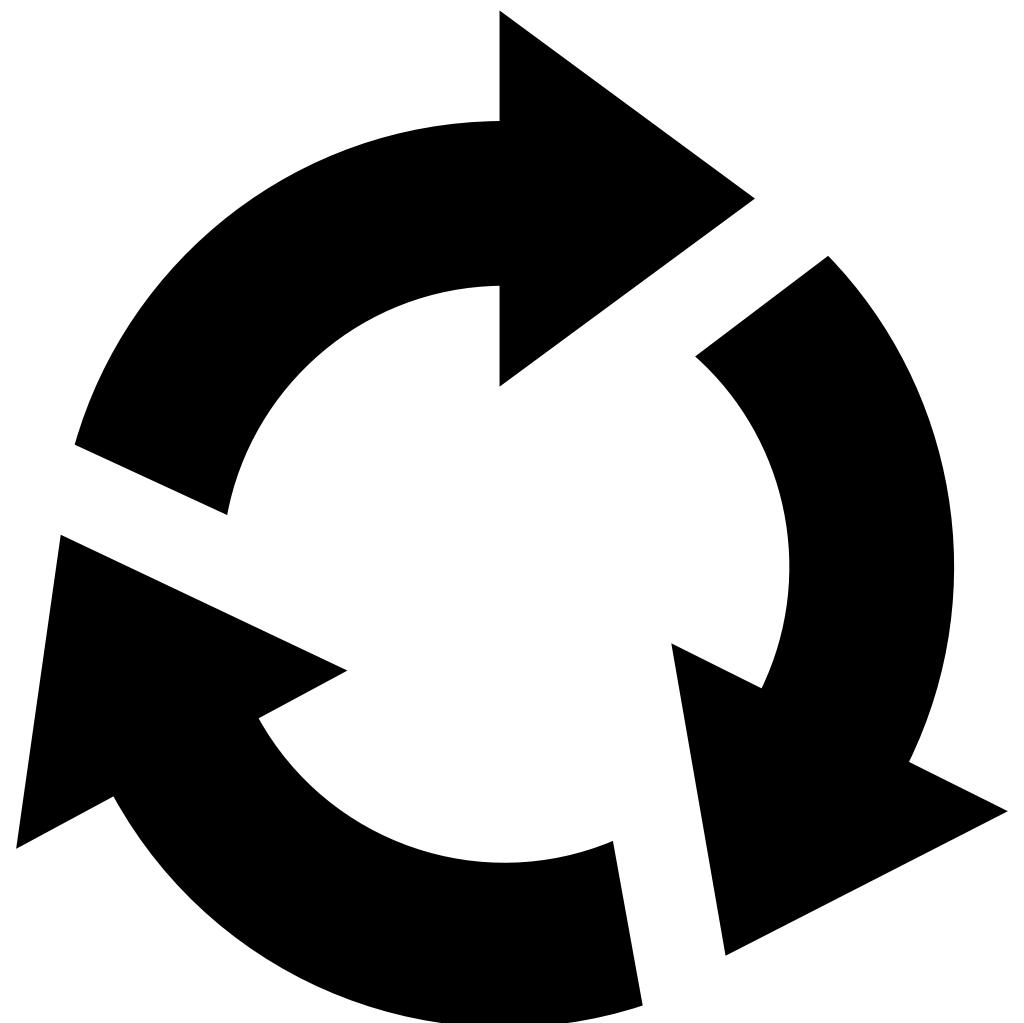
Reactive expressions that return a value based on a reactive value or expression are also common.

```
server <- function(input, output) {  
  injuries_demographics <- reactive ({  
    injuries |> filter(prod_code == input$code) |>  
    count(age, sex, wt = weight, sort = TRUE)  
  })  
  
  output$injurePlot1 <- renderPlot({injuries_demographics() |>  
    ggplot(aes(x = age, y = n, color = sex)) +  
    geom_line() +  
    labs(title = "", y = "Estimated Number of Injuries for Product 1") +  
    theme_grey(base_size = input$text_size)  
  })
```



The diagram illustrates the reactive dependency between the reactive assignment and its use in the plot rendering. A yellow arrow points from the reactive assignment `injuries_demographics <- reactive ({ ... })` to the call to `injuries_demographics()` inside the `renderPlot` function.

# Reactivity Considerations





# Questions and next steps...



# Working with Dashboards in R

# Closing thoughts on dashboards in R

1. Develop your content for your dashboard independently of the design
2. Embrace iterative design (start simple and add features)
3. Consider your end user (interface design, hosting of dashboard, concepts shared)



# A special thanks to:

- Mine Çetinkaya Rundell for pedagogy and structure suggestions for this workshop
- Garrett Grolemund and RStudio for inspiration on the slides in the “shiny basics” section
- Hadley Wickham for his subset on the NEISS data and Mastering shiny



# Online resources for learning more...

## Online Reference and Tutorials

- **Flexdashboard:** [pkgs.rstudio.com/flexdashboard](http://pkgs.rstudio.com/flexdashboard)
- **Shiny:** [shiny.rstudio.com](http://shiny.rstudio.com)  
[mastering-shiny.org](http://mastering-shiny.org)

## Training videos

- **Shiny Workshop:** [rfun.library.duke.edu/portfolio/shiny\\_workshop](http://rfun.library.duke.edu/portfolio/shiny_workshop)



# Reactivity Exercise

## 1. Add a second select menu to create a second injury plot

- inputId = “code2”
- label = “Product 2”
- choices = (you can use the same choices as the first select)
- selected = 5040

## 2. Create a second plot by

- Copy the code inside the server function and paste a copy below the existing code
- In the copied code:
  - Change injuries\_demographics to injuries\_demographics2
  - Change input\$code to input\$code2
  - Change output\$injurePlot1 to output\$injurePlot2
  - Change injuries\_demographics() to injuries\_demographics2()

## 3. Run the new app



05:00

# NEISS Demographic Comparisons by Product

**Product 1**

basketball (activity, apparel or equipment)

**Product 2**

bicycles and accessories

