

### First Shiny App

For this assignment, you will be creating your first Shiny app. Be sure that you have installed the Shiny package (to do this, use your console to run the `install.packages("shiny")` command).

You will be creating the ER Injuries Shiny App. The instructions for this assignment can be found at:

<https://mastering-shiny.org/basic-case-study.html>

Follow the instructions in this case study to complete the assignment. To help you get started, I have provided some instructions below:

1. The instructions in the book assume you have already created the new Shiny project. Here is a recap for creating a new Shiny project:
  - a. Click on File -> New Project
  - b. Click on New Directory and select the third option down which is Shiny Web Application
  - c. Set the Directory name to **Mod6Assignment3** and save it to your Module6 folder for the course.
2. This will create a new project with 2 files: the project file (.Rproj) and an **app.R** file
3. From Canvas, download the following 3 files into the Mod6Assignment3 folder you just created (Note: these must be in the same folder as the app.R file and must have the same filenames):
  - a. injuries.tsv.gz
  - b. products.tsv
  - c. population.tsv
4. The last step you need to do is install a new package we will be using for this application. The name of the package is vroom and can be installed by copying/running the following code in the Console:

```
install.packages("vroom")
```

### Creating a new Shiny app

5. Open the **app.R** file and you will notice there is already a pre-built Shiny app in the file.
6. Delete all the code in the file as we are going to create a new Shiny app from scratch.
7. The first lines of code you will need to run to create the app are listed below (this comes from 4.1 of the Case study: ER injuries):

```
library(shiny)
library(vroom)
library(tidyverse)
```

These will be used to run the Shiny app, manipulated the data (i.e., vroom) and create plots (i.e., tidyverse).

8. Next, you will also need to include the following code to create the data files to use in your app. The code uses vroom which helps with loading larger sets of data (for more information on vroom see <https://vroom.r-lib.org/>)

```
injuries <- vroom::vroom("injuries.tsv.gz")
products <- vroom::vroom("products.tsv")
population <- vroom::vroom("population.tsv")
```

9. At this point, we have gone through 4.1 and 4.2 from of the ER Injuries Case Study. Skip to section 4.4 and begin adding the code to create the Shiny app. Both the code for the ui and server needs to be added after the code you just added to app.R that loads the data.

The ui code looks like this:

```
prod_codes <- setNames(products$prod_code, products$title)

ui <- fluidPage(
  fluidRow(
    column(6,
      selectInput("code", "Product", choices = prod_codes)
    )
  ),
  fluidRow(
    column(4, tableOutput("diag")),
    column(4, tableOutput("body_part")),
    column(4, tableOutput("location"))
  ),
  fluidRow(
    column(12, plotOutput("age_sex"))
  )
)
```

The servers code looks like this:

```
server <- function(input, output, session) {
  selected <- reactive(injuries %>% filter(prod_code == input$code))

  output$diag <- renderTable(
    selected() %>% count(diag, wt = weight, sort = TRUE)
  )
  output$body_part <- renderTable(
    selected() %>% count(body_part, wt = weight, sort = TRUE)
  )
  output$location <- renderTable(
    selected() %>% count(location, wt = weight, sort = TRUE)
  )
}
```

```
summary <- reactive({
  selected() %>%
    count(age, sex, wt = weight) %>%
    left_join(population, by = c("age", "sex")) %>%
    mutate(rate = n / population * 1e4)
})

output$age_sex <- renderPlot({
  summary() %>%
    ggplot(aes(age, n, colour = sex)) +
    geom_line() +
    labs(y = "Estimated number of injuries")
}, res = 96)
}
```

Also, you will need to add the following to the end of the **app.R** file:

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

This will let R know that all the code before it is part of a Shiny app.

10. Run the app and notice how the output is not exactly great. Continue through the ER Injuries case study through 4.7 (you don't need to complete 4.8 in the assignment). This will require you to go back to both the ui code and server code to make modifications. If you get stuck, the author of the book has provided his source code via links in the book to his github.
11. Once you have completed the assignment, upload your **app.R** file to Canvas. This will allow me to run the code to create the app locally for grading.