

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

Part 1 – Downloading the data files using the console

3. Before we start creating the app, we need to get the files downloaded. Copy the following code into the Console which will create a new directory in your project:

```
dir.create("neiss")
```

4. Notice how there is now a folder called neiss that has been created in your project folder. The next thing to do is copy and paste the following code in the Console and run:

```
download <- function(name) {  
  url <- "https://github.com/hadley/mastering-shiny/raw/master/neiss/"  
  download.file(paste0(url, name), paste0("neiss/", name), quiet = TRUE)  
}
```

5. This code will create a new function called download (see your Environment pane in the upper corner of RStudio). This function will be used to download the files we need to use in the Shiny application. The next step would be to run the following lines of code in the Console.

```
download("injuries.tsv.gz")  
download("population.tsv")  
download("products.tsv")
```

6. If you click on the neiss folder, you should now see the files that will be used for the application have been downloaded to this folder.

7. The last step you need to do in the Console 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")
```

Part 2 – Creating a new Shiny app

8. Open the **app.R** file and you will notice there is already a pre-built Shiny app in the file.
9. Delete all the code in the file as we are going to create a new Shiny app from scratch.
10. 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, manipulate the data (i.e., vroom) and create plots (i.e., tidyverse).

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

12. At this point, we have gone through 4.1 and 4.2 from 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.

13. 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

Module 6: Assignment #3

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.

14. 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.