STAT 231: Problem Set 3B

Sean Wei

due by 5 PM on Friday, September 18

This homework assignment is designed to help you futher ingest, practice, and expand upon the material covered in class over the past week(s). You are encouraged to work with other students, but all code and text must be written by you, and you must indicate below who you discussed the assignment with (if anyone).

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps3B.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps3B.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

If you discussed this assignment with any of your peers, please list who here:

ANSWER:

Shiny app

1. Finish your app from Lab04b and add your app code to the R code chunk below:

(1) update the Lab04b app to still explore the `hate_crimes` dataset, but with different app functionality (e.g. different widgets, variables, layout, theme...); OR
(2) use it as a template to create a Shiny app for a different dataset, choosing from:

- `mad_men` (tv performers and their post-show career)
- `ncaa_w_bball_tourney` (women's NCAA div 1 basketball tournament, 1982-2018)
- `nfl_suspensions` (NFL suspensions, 1946-2014)
- `candy_rankings` (candy characteristics and popularity)

These four datasets are also part of the `fivethirtyeight` package and their variable definitions are included in a pdf posted to the Moodle course page.

If using the `hate_crimes` dataset, be sure to update:

- at least 2 different widgets; and
- the layout (e.g. not in tabs or different page layout) or the theme
  - check out: https://rstudio.github.io/shinythemes/
- like a challenge? incorporate one of the click, hover or brush features
  - check out: https://shiny.rstudio.com/articles/plot-interaction.html

```
## keep eval = FALSE in this code chunk option so your app doesn't
## try to run when knitting the document

## add your app code here (including any packages and datasets loaded,
## the ui call, the server call, and the shinyApp call)

library(fivethirtyeight)
library(shinythemes)
library(shinybusy)
library(tidyverse)
library(ggrepel)
library(kableExtra)
library(dplyr)
library(tidyr)
library(stringr)

#Convert data to tidy for later
candy_rankings_tidy <- candy_rankings %>%
  pivot_longer(-c(competitorname, sugarpercent, pricepercent, winpercent),
               names_to = "characteristics", values_to = "present") %>%
  mutate(present = as.logical(present)) %>%
  arrange(competitorname)

data(candy_rankings)

# Define Predictor Variable Choices
predictors <- as.list(names(candy_rankings)[11:12])
predictor_names <- c("Sugar Percent"
```

```
                    , "Price Percent")
names(predictors) <- predictor_names

# ui
ui <- fluidPage(

  # CSS
  tags$head(
    tags$style(HTML("
      h1, h3 {
        color: orange;
        text-align: center;
      }
    "))
  ),

  HTML("<h1>Halloween Candy Power Rankings</h1>
        <h3>by Sean Wei</h3>"),

  sidebarLayout(
    sidebarPanel(

      selectInput(inputId = "predictor",
                  label = "Choose a predictor variable of interest for the scatterplot:",
                  choices = predictors,
                  selected = "Sugar Percent")
    ),

    mainPanel(
      #let users know things might take a while to load
      add_busy_spinner(spin = "double-bounce"),
      tabsetPanel(type = "tabs",
          tabPanel("Scatterplot",
            HTML("<p>The scatterplot below is interactive - try hovering over the
                 points to see the candy brand!</p>"),
            plotOutput(outputId = "scatter", hover = hoverOpts(id = "plot_hover", delay = 0))),

          tabPanel("Rankings",
            HTML("<p>This table explains how often a candy of a given type won its
                 matchups against other brands."),
            tableOutput(outputId = "table")),

          tabPanel("Candy Characteristics",
            HTML("<p>Curious to see the different possible characteristics of your
                 favorite candy? Use this table to explore and search for your favorites.</p>"),
            fluidRow(
              selectInput(
                inputId = "characteristic",
                label = "Characteristic",
                choices = c(
                  "ALL",
                  unique(as.character(candy_rankings_tidy$characteristics))
                )
```

```
              ),
              fluidRow(
                DT::dataTableOutput("table2")
              )
            )
          )
        )
    )
  ),
  uiOutput("dynamic")

)

# server
server <- function(input, output){

  output$scatter <- renderPlot({
    ggplot(data = candy_rankings, aes_string(x = input$predictor, y = "winpercent")) +
      geom_point() +
      labs(x = names(predictors)[predictors == input$predictor]
           , y = "Win Percentage") +
      ggtitle("Win Percentage Based on Sugar % or Price %")
  })

  output$dynamic <- renderUI({
    req(input$plot_hover)
    textOutput("vals")
  })

  output$vals <- renderPrint({
    hover <- input$plot_hover
    y <- nearPoints(candy_rankings, input$plot_hover)["competitorname"]
    req(nrow(y) != 0)
    print(y$competitorname)
  })

  output$table <- function() {
    candy_rankings %>%
      select(competitorname, winpercent) %>%
      arrange(desc(winpercent)) %>%
      rename(Candy = competitorname, `Win Percentage` = winpercent) %>%
      knitr::kable("html") %>%
      kable_styling(bootstrap_options = "striped")
  }

  datasetcandy <- reactive({  # Filter data based on selections
    data <- candy_rankings_tidy %>%
      filter(present == TRUE) %>%
      select(-c(winpercent, sugarpercent, pricepercent, present)) %>%
      rename(Candy = competitorname, Characteristic = characteristics)
    req(input$characteristic) # wait until there's a selection
    #If there is a filter
    if (input$characteristic != "ALL") {
```

```
      data <- data %>%
        filter(Characteristic == input$characteristic)
    }
    #If selection is ALL
    else {
      data <- data
    }
  })

  output$table2 <- DT::renderDataTable(DT::datatable(datasetcandy()))

}

# call to shinyApp
shinyApp(ui = ui, server = server)
```

2. Publish your app. Then, go to the GitHub discussion "Shiny Apps" and reply to the message with (1) the URL to your published Shiny app; and (2) a paragraph explaining what story your Shiny app is telling, and how the interactivity you created enhances the telling of that story.

   ANSWER: Do not include anything here. The link to your app and the paragraph should be posted to the "Shiny Apps" discussion thread on GitHub.