STAT 231: Problem Set 3B

Evan Daisy

due by 5 PM on Friday, March 12

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 `electric_skateboards` 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:

- `candy_rankings` (candy characteristics and popularity)
- `hate_crimes` (hate crimes in US states, 2010-2015)
- `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)

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

If using the `electric_skateboards` 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(shiny)
library(shinythemes)
library(tidyverse)
library(DT)
library(ggrepel)
library(fivethirtyeight)

# import data #
data(candy_rankings)

#create hist options
hist_choice_values <- c("sugarpercent","pricepercent","winpercent")
hist_choice_names <- c("Sugar Percentile","Price Percentile","Win Percentile")
names(hist_choice_values) <- hist_choice_names

#For the histogram faceting
facet_choice_values <- c("pluribus", "fruity", "peanutyalmondy", "chocolate")
facet_choice_names <- c("Many in a package?", "Fruit?", "Nuts?", "Chocolate?")
names(facet_choice_values) <- facet_choice_names

#For the histogram facet labels and the legend on the scatterplot
```

```
candy_hist <- candy_rankings %>%
  mutate(chocolate = case_when(chocolate == TRUE ~"Chocolate",
                               chocolate == FALSE ~"No chocolate")) %>%
  mutate(pluribus = case_when(pluribus == TRUE ~"One of Many",
                              pluribus == FALSE ~"One of Few")) %>%
  mutate(peanutyalmondy = case_when(peanutyalmondy == TRUE ~"Nuts",
                                    peanutyalmondy == FALSE ~"No Nuts")) %>%
  mutate(fruity = case_when(fruity == TRUE ~"Fruit",
                                    fruity == FALSE ~"No Fruit")) %>%
  mutate(nougat = case_when(nougat == TRUE ~"Nougat",
                                    nougat == FALSE ~"No Nougat")) %>%
  mutate(caramel = case_when(caramel == TRUE ~"Caramel",
                             caramel == FALSE ~"No Caramel"))

#For the scatterplot color choices
color_choice_values <- c("nougat", "fruity", "caramel", "peanutyalmondy", "chocolate")
color_choice_names <- c("Nougat?", "Fruit?", "Caramel?", "Nuts?", "Chocolate?")
names(color_choice_values) <- color_choice_names

#allow people to choose the scatterplot independent variable
scat_choice_values <- c("sugarpercent","pricepercent")
scat_choice_names <- c("Sugar Content (Percentile)","Price (Percentile)")
names(scat_choice_values) <- scat_choice_names

#Let people identify a specific brand in the scatterplot
name_choices <- unique(candy_rankings$competitorname)

#Select only the numerical columns for the table
#(assuming people know what's in each candy)
candy_rankings2 <- candy_rankings %>%
  select(competitorname, sugarpercent, pricepercent, winpercent) %>%
  #Standardize the percents (making them easier to compare)
  mutate(sugarpercent = sugarpercent*100) %>%
  mutate(pricepercent = pricepercent*100)

ui <- navbarPage(

  title="Candy Stats",

  tabPanel(
    title = "Histogram",
    sidebarLayout(
      sidebarPanel(
        selectInput(inputId = "histvar"
                    , label = "Choose a variable of interest to plot:"
                    , choices = hist_choice_values
                    , selected = "Price Percentile"),
        radioButtons(inputId = "facet"
                     , label = "Facet by:"
                     , choices = facet_choice_values
                     , selected = NULL)
      ),
      mainPanel(
```

```
        plotOutput(outputId = "hist")
      )
    )
  ),

  tabPanel(
    title = "Scatterplot",

    sidebarLayout(
      sidebarPanel(
        selectInput(inputId = "scatvar"
                    , label = "Choose a variable of interest
                    to plot against popularity:"
                    , choices = scat_choice_values
                    , selected = "Sugar Content"),
        radioButtons(inputId = "pt_color"
                     , label = "Color points by:"
                     , choices = color_choice_values
                     , selected = "Chocolate?"),
        selectizeInput(inputId = "id_name"
                       , label = "Identify candy(s) in the scatterplot:"
                       , choices = name_choices
                       , selected = NULL
                       , multiple = TRUE)
      ),
      mainPanel(
        plotOutput(outputId = "scatter")
      )
    )
  ),

  tabPanel(
    title = "Table",

    sidebarLayout(
      sidebarPanel(
        selectizeInput(inputId = "cmpy"
                       , label = "Choose one or more brands:"
                       , choices = name_choices
                       , selected = "3 Musketeers"
                       , multiple = TRUE)
      ),
      mainPanel(
        DT::dataTableOutput(outputId = "table")
      )
    )
  )
)

server <- function(input,output){

  data_for_hist <- reactive({
    data <- candy_hist
```

```
    #filter(candy_rankings, input$type == TRUE)
  })

  output$hist <- renderPlot({
    ggplot(data = data_for_hist(), aes_string(x = input$histvar)) +
      geom_histogram(color = "#2c7fb8", fill = "#7fcdbb", alpha = 0.7) +
      labs(x = hist_choice_names[hist_choice_values == input$histvar]
           , y = "Number of Candies")  +
      facet_grid(rows = vars(get(input$facet)))
  })

  # TAB 2: INTERACTIVE SCATTERPLOT
  output$scatter <- renderPlot({
    candy_hist %>%
      ggplot(aes_string(x= input$scatvar, y="winpercent", color = input$pt_color)) +
      geom_point(size = 3) +
      labs(x = scat_choice_names[scat_choice_values == input$scatvar], y = "Popularity (Win Percentile)
           , title = "Candy", subtitle = "2015"
           , color = color_choice_names[color_choice_values == input$pt_color]) +
      geom_label_repel(data = filter(candy_rankings, competitorname %in% input$id_name)
                       , aes(label = competitorname), show.legend = FALSE) +
      geom_smooth(method = "lm", se = FALSE)
    #facet_grid(~chocolate, labeller = labeller(chocolate = choc.labs))
  })

  # TAB 3: TABLE
  data_for_table <- reactive({
    data <- filter(candy_rankings2, competitorname %in% input$cmpy)
  })

  output$table <- DT::renderDataTable({
    data_for_table()
  })
}

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

2. Publish your app. Then, go to the Google group conversation "PS3B: 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 "PS3B: Shiny Apps" Google conversation thread.