

STAT 231: Problem Set 3B

Jett Knight

due by 5 PM on Friday, March 12

This homework assignment is designed to help you further 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>

```
library(shiny)
library(shinythemes)
library(tidyverse)
library(DT)
library(ggrepel)

skateboards <- read_csv("electric_skateboards.txt")

hist_choice_values <- c("price", "range", "top_speed", "weight", "battery")
hist_choice_names <- c("Price", "Range", "Top Speed", "Weight", "Battery")
names(hist_choice_values) <- hist_choice_names

drv_choices <- unique(skateboards$drive)

size_choice_values <- c("range", "top_speed", "battery")
size_choice_names <- c("Range", "Top Speed", "Battery")
names(size_choice_values) <- size_choice_names

name_choices <- unique(skateboards$board)

cmpy_choices <- unique(skateboards$company)

ui <- navbarPage(

  title="Electric Skateboards",

  fluidPage(theme = shinytheme("cerulean")),
```

```

tabPanel(
  title = "Histogram",
  sidebarLayout(
    sidebarPanel(
      selectInput(inputId = "histvar"
        , label = "Choose a variable of interest to plot:"
        , choices = hist_choice_values
        , selected = "price"),
      checkboxGroupInput(inputId = "drv"
        , label = "Include drive types:"
        , choices = drv_choices
        , selected = drv_choices
        , inline = TRUE)
    ),
    mainPanel(
      plotOutput(outputId = "hist")
    )
  )
),

tabPanel(
  title = "Scatterplot",

  sidebarLayout(

    sidebarPanel(
      radioButtons(inputId = "pt_size"
        , label = "Size points by:"
        , choices = size_choice_values
        , selected = "weight"),
      selectizeInput(inputId = "id_name"
        , label = "Identify skateboard(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 companies:"
        , choices = cmpy_choices
        , selected = "DIYElectric"
        , multiple = TRUE)
    ),

```

```

    mainPanel(
      DT::dataTableOutput(outputId = "table")
    )
  ),
  tabPanel(
    title = "Original Graph",

    sidebarLayout(
      sidebarPanel(
        tags$div(
          HTML(paste("Original figure was presented by "
                    , tags$a(href="https://www.electricskateboardhq.com/boards-comparison/", "HQ SkateboardHQ")
                    , sep = ""))
        )
      ),
      mainPanel(
        h3("Information overload!"),
        plotOutput(outputId = "original")
      )
    )
  )
)

server <- function(input,output){

  data_for_hist <- reactive({
    data <- filter(skateboards, drive %in% input$drv)
  })

  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 Skateboards")
  })

  output$scatter <- renderPlot({
    skateboards %>%
      filter(drive != "Direct") %>%
      ggplot(aes_string(x="range", y="top_speed", size = input$pt_size)) +
      geom_point(color = "#2c7fb8") +
      labs(x = "Range (miles)", y = "Top Speed (mph)"
           , title = "Electric Skateboards", subtitle = "August 2018"
           , size = size_choice_names[size_choice_values == input$pt_size]) +
      geom_label_repel(data = filter(skateboards, board %in% input$id_name)
                      , aes(label = board), show.legend = FALSE) +
      facet_grid(~drive)
  })

  data_for_table <- reactive({
    data <- filter(skateboards, company %in% input$cmpy)
  })

```

```

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

output$original <- renderPlot({
  ggplot(data = skateboards, aes(x=range, y=top_speed, color=company
                                , shape = drive, size = weight)) +
    geom_point() +
    geom_text(aes(label = board), hjust = 0, nudge_x = 0.05, size=3) +
    labs(x = "Range (miles)", y = "Top Speed (mph)"
         , title = "Electric Skateboards", subtitle = "August 2018"
         , shape = "Drive type", size = "Weight of board") +
    guides(color = FALSE)
})
}

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.