

STAT 231: Problem Set 3B

Lorraine Oloo

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>

```
## 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)

Data <- fivethirtyeight::mad_men %>%
  mutate(status_correct = case_when(str_detect(status, "End") ~ "END"
                                     , TRUE ~ status))

#remove names of performers that are not string for my scatterplot to work
Data <- Data[-c(16,91,131,188,218), ]

#Bargraph tab1
bar_choice_values <- c("num_lead","num_support",
                      "num_shows")
bar_choice_names <- c("The number of leading roles in films the performer has appeared in",
                      "The number of supporting roles in films the performer has appeared in",
                      "Number of seasons actors appeared in atleast half of the episodes")
names(bar_choice_values) <- bar_choice_names
```

```

# for checkboxGroupInput for bargraph
Status_in_show <- unique(Data$status_correct)

#Table tab2
show_choices <- unique(Data$show)

#scatterplot tab3
yaxis_values <- c("score", "num_shows", "num_support")
yaxis_names <- c("Score", "Number of shows", "Number of support")
names(yaxis_values) <- yaxis_names

# for selectizeInput choices for actor name, pull directly from data
name_choices <- unique(Data$performer)

#ui
ui <- navbarPage(

  title="Series and their actors",

  #bargraph
  tabPanel(
    title = "Bar Graph",
    sidebarLayout(
      sidebarPanel(
        selectInput(inputId = "barvar"
          , label = "Choose a variable of interest to plot:"
          , choices = bar_choice_values
          , selected = "years_since"),
        checkboxGroupInput(inputId = "show"
          , label = "Include the type of show:"
          , choices = Status_in_show
          , selected = Status_in_show
          , inline = TRUE)
      ),
      mainPanel(
        plotOutput(outputId = "bar")
      )
    )
  ),

  #table
  tabPanel(
    title = "Table",

    sidebarLayout(
      sidebarPanel(
        selectizeInput(inputId = "Show"
          , label = "Select show of interest:"
          , choices = show_choices
          , selected = "Damages"
          , multiple = FALSE)
      ),

```

```

    mainPanel(
      DT::dataTableOutput(outputId = "table")
    )
  )
),

#scatterplot
tabPanel(
  title = "Scatterplot",

  sidebarLayout(
    sidebarPanel(
      radioButtons(inputId = "yaxis"
        , label = "Relationship to the number of leading roles by:"
        , choices = yaxis_values
        , selected = "score"),
      selectizeInput(inputId = "id_name"
        , label = "Identify actors in the scatterplot:"
        , choices = name_choices
        , selected = NULL
        , multiple = TRUE)
    ),
    mainPanel(
      plotOutput(outputId = "scatter")
    )
  )
)

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

  # TAB 1: BARGRAPH
  data_for_bar <- reactive({
    data <- filter(Data, status_correct %in% input$show)
  })

  output$bar <- renderPlot({
    ggplot(data = data_for_bar(), aes_string(x = input$barvar)) +
      geom_bar()
  })

  # TAB 2: TABLE
  data_for_table <- reactive({
    data <- filter(Data, show %in% input$Show)
  })

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

```

```

# TAB 3: SCATTERPLOT
output$scatter <- renderPlot({
  Data %>%

  ggplot(aes_string(x="num_lead", y=input$yaxis)) +
  geom_point(color = "#2c7fb8") +
  labs(x = "Number of leading roles", y = input$yaxis
    , title = "Scatterplot") +
  geom_label_repel(data = filter(Data, performer %in% input$id_name)
    , aes(label = performer)) +
  facet_grid(~status_correct)
})
}

# 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.