Assignment 3

Name: Milind Shaileshkumar Parvatia

Sid: s3806853

URL to code submit: <https://milindparvatia.shinyapps.io/Assingment3_readmd/>

URL to visualization: <https://milindparvatia.shinyapps.io/Assignment3Storytelling/>

Code:

library(readr)

library(shiny)

library(ggplot2)

library(dplyr)

library(lubridate)

library(scales)

library(ggthemes)

library("plotly")

library(rsconnect)

daily\_tests <- read\_csv("tests-per-confirmed-case-daily-smoothed.csv")

total\_deaths <- read\_csv("total-deaths-covid-19.csv")

tests <- read\_csv("full-list-total-tests-for-covid-19.csv")

confirm\_case <- read\_csv("total-cases-covid-19.csv")

daily\_tests <- mutate(daily\_tests, Date= as.Date(Date, format= "%b %d,%Y"))

total\_deaths <- mutate(total\_deaths, Date= as.Date(Date, format= "%b %d,%Y"))

tests <- mutate(tests, Date= as.Date(Date, format= "%b %d,%Y"))

confirm\_case <- mutate(confirm\_case, Date= as.Date(Date, format= "%b %d,%Y"))

# Define UI for application that draws a histogram

ui <- fluidPage(

# Application title

titlePanel("Can testing late can be blamed for grime milestone of COVID-19 deaths in the United States ?"),

# Sidebar with a slider input for number of bins

sidebarLayout(

sidebarPanel(

tags$div(

tags$h4("What do we know about Covid-19?"),

tags$p("So far, we don't have any vaccine for the epidemic of COVID-19 and in the early stage of this new virus to

reduce it's spreading countries took different approaches."),

tags$p("Here, I would like to say hypothetically that countries with an early focus on testing have a better chance of reducing

the spread of COVID-19 than countries with delaying in testing. "),

),

sliderInput("slider", "Play",

min = as.Date("2019-12-31"),

max =as.Date("2020-06-15"),

value=as.Date("2019-12-31"),

timeFormat="%b %Y",

animate = animationOptions(interval = 300, loop = FALSE)

),

tags$div(

tags$h4("Which way we can compare these countries?"),

tags$p("One important way to understand if countries are testing sufficiently is to ask: How many tests does a country do to find one COVID-19 case? Which way we can compare these countries?"),

tags$li("Some countries, like Australia, South Korea did hundreds or even thousands of very early on tests for each case they find."),

tags$li("While countries like the US, UK, and Italy started their testing late or in very few numbers comparatively"),

),

tags$div(

tags$h4("What do we learn from this?"),

tags$p("We can see that the USA has the highest test performed but their testing started very late of mid-may and their tests per

confirmed cases rates are extremely low, with the total death toll of 110k."),

tags$p("Compare to that we can see that, South Korea has stared testing very early on from last week of Jan and able to put control over the spread of COVID-19,

Due to that, the death toll is also very low."),

tags$p("So even though both of these countries have the same date of first Positive case and with the same testing technology of COVID-19, but their different approach has made a huge difference

between how much both countries can respond to COVID-19."),

),

),

# Show a plot of the generated distribution

mainPanel(

tabsetPanel(type = "tabs",

tabPanel("Total Confirmed cases",plotlyOutput("casesPlot")),

tabPanel("Tests per Confirmed cases",plotlyOutput("testPlot")),

tabPanel("Total Tests Performed",plotlyOutput("alltestPlot")),

tabPanel("Total Deaths", plotlyOutput("deathPlot"))

),

tabsetPanel(type = "tabs",

tabPanel("Details",

tags$h3("What do these chats say?"),

tags$p("Total Confirmed cases: How many positive cases of COVID-19 each countries have."),

tags$p("Tests per Confirmed case: Tests ver conducted per each positive case of COVID-19."),

tags$p("Total Tests Performed: how many tests overall each country performed."),

tags$p("Total Deaths: how many deaths overall each country have due to COVID-19."),

)

),

)

)

)

cbp1 <- c("#999999", "#E69F00", "#56B4E9", "#009E73",

"#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# Define server logic required to draw a histogram

server <- function(input, output) {

output$casesPlot <- renderPlotly({

# generate bins based on input$bins from ui.R

ggplot() +

geom\_line(data = filter(confirm\_case, Entity == "Italy"), aes(sort(Date), `Total confirmed cases of COVID-19 (cases)`, color = "Italy"))+

geom\_line(data = filter(confirm\_case, Entity == "Australia"), aes(sort(Date), `Total confirmed cases of COVID-19 (cases)`, color = "Australia"))+

geom\_line(data = filter(confirm\_case, Entity == "South Korea"), aes(sort(Date), `Total confirmed cases of COVID-19 (cases)`, color = "South Korea"))+

geom\_line(data = filter(confirm\_case, Entity == "United States"), aes(sort(Date), `Total confirmed cases of COVID-19 (cases)`, color = "United States"))+

geom\_line(data = filter(confirm\_case, Entity == "United Kingdom"), aes(sort(Date), `Total confirmed cases of COVID-19 (cases)`, color = "United Kingdom"))+

xlab("Month")+

scale\_y\_continuous(trans='log2')+

scale\_color\_manual(name = "Country Name",

breaks = c("Italy", "Australia","South Korea", "United States", "United Kingdom"),

values = c("Italy" = "#E69F00", "Australia" = "yellow", "South Korea" = "#56B4E9", "United States" = "blue", "United Kingdom" = "#009E73"))+

geom\_vline(xintercept = as.numeric(input$slider), linetype="dotted", color = "#999999", size=1)

})

output$alltestPlot <- renderPlotly({

# generate bins based on input$bins from ui.R

ggplot() +

geom\_line(data = filter(tests, Entity == "Italy"), aes(sort(Date), `Total tests`, color = "Italy"))+

geom\_line(data = filter(tests, Entity == "Australia"), aes(sort(Date), `Total tests`, color = "Australia"))+

geom\_line(data = filter(tests, Entity == "South Korea"), aes(sort(Date), `Total tests`, color = "South Korea"))+

geom\_line(data = filter(tests, Entity == "United States"), aes(sort(Date), `Total tests`, color = "United States"))+

geom\_line(data = filter(tests, Entity == "United Kingdom"), aes(sort(Date), `Total tests`, color = "United Kingdom"))+

xlab("Month")+

scale\_y\_continuous(trans='log2')+

scale\_color\_manual(name = "Country Name",

breaks = c("Italy", "Australia","South Korea", "United States", "United Kingdom"),

values = c("Italy" = "#E69F00", "Australia" = "yellow", "South Korea" = "#56B4E9", "United States" = "blue", "United Kingdom" = "#009E73"))+

geom\_vline(xintercept = as.numeric(input$slider), linetype="dotted", color = "#999999", size=1)

})

output$testPlot <- renderPlotly({

# generate bins based on input$bins from ui.R

ggplot() +

geom\_line(data = filter(daily\_tests, Entity == "Italy"), aes(sort(Date), `Tests per confirmed case – daily (tests per confirmed case)`, color = "Italy"))+

geom\_line(data = filter(daily\_tests, Entity == "Australia"), aes(sort(Date), `Tests per confirmed case – daily (tests per confirmed case)`, color = "Australia"))+

geom\_line(data = filter(daily\_tests, Entity == "South Korea"), aes(sort(Date), `Tests per confirmed case – daily (tests per confirmed case)`, color = "South Korea"))+

geom\_line(data = filter(daily\_tests, Entity == "United States"), aes(sort(Date), `Tests per confirmed case – daily (tests per confirmed case)`, color = "United States"))+

geom\_line(data = filter(daily\_tests, Entity == "United Kingdom"), aes(sort(Date), `Tests per confirmed case – daily (tests per confirmed case)`, color = "United Kingdom"))+

scale\_y\_continuous(trans='log10')+

xlab("Month")+

ylab("Tests per confirmed case – daily")+

scale\_color\_manual(name = "Country Name",

breaks = c("Italy", "Australia","South Korea", "United States", "United Kingdom"),

values = c("Italy" = "#E69F00", "Australia" = "yellow", "South Korea" = "#56B4E9", "United States" = "blue", "United Kingdom" = "#009E73"))+

geom\_vline(xintercept = as.numeric(input$slider), linetype="dotted", color = "#999999", size=1)

})

output$deathPlot <- renderPlotly({

# generate bins based on input$bins from ui.R

p <- ggplot() +

geom\_line(data = filter(total\_deaths, Entity == "Australia"), aes(sort(Date), `Total confirmed deaths due to COVID-19 (deaths)`, color = "Australia"))+

geom\_line(data = filter(total\_deaths, Entity == "Italy"), aes(sort(Date), `Total confirmed deaths due to COVID-19 (deaths)`, color = "Italy"))+

geom\_line(data = filter(total\_deaths, Entity == "South Korea"), aes(sort(Date), `Total confirmed deaths due to COVID-19 (deaths)`, color = "South Korea"))+

geom\_line(data = filter(total\_deaths, Entity == "United States"), aes(sort(Date), `Total confirmed deaths due to COVID-19 (deaths)`, color = "United States"))+

geom\_line(data = filter(total\_deaths, Entity == "United Kingdom"), aes(sort(Date), `Total confirmed deaths due to COVID-19 (deaths)`, color = "United Kingdom"))+

xlab("Month")+

scale\_y\_continuous(trans='log2')+

scale\_color\_manual(name = "Country Name",

breaks = c("Italy", "Australia","South Korea", "United States", "United Kingdom"),

values = c("Italy" = "#E69F00", "Australia" = "yellow", "South Korea" = "#56B4E9", "United States" = "blue", "United Kingdom" = "#009E73"))+

geom\_vline(xintercept = as.numeric(input$slider), linetype="dotted", color = "#999999", size=1)

ggplotly(p)

})

}

# Run the application

shinyApp(ui = ui, server = server)