**Shonil Dabreo, s3835204**

**Link:** <https://rpubs.com/NightxWalker/779378>

**Code:**

defaultWarn <- getOption("warn")

options(warn = -1)

#----------------------Packages---------------------------------

library(readr) # to read data

library(dplyr) # to prepare the data

library(ggplot2) # to plot static graphs

library(tidyr) # to manipulate and shape data

library(plotly) # to make graphs interactive

library(leaflet) # to plot dynamic map

library(flexdashboard) # to display dashboard

#----------------------Data-------------------------------------

# Reading the data

covid <- read.csv("https://raw.githubusercontent.com/RamiKrispin/coronavirus/master/csv/coronavirus.csv", stringsAsFactors = FALSE) %>%

mutate(date = as.Date(date, "%Y-%m-%d"),

country = factor(country, levels = unique(country)),

lat = as.numeric(lat),

long = as.numeric(long),

type = factor(type, levels = unique(type)))

# dropping province column and setting data types

covid <- select(covid, -matches("province"))

# New variable as month and year

covid$month <- format(covid$date, "%m")

covid$year <- format(covid$date, "%y")

#---- Wider format

covidW <- covid %>%

select(country, lat, long, type, cases) %>%

group\_by(country, lat, long, type) %>%

summarise(cases = sum(cases, na.rm = TRUE)) %>%

pivot\_wider(names\_from = type,

values\_from = cases) %>%

mutate(active = confirmed - death - recovered) %>%

arrange(-active)

#---- Longer format

covidL <-

covidW %>%

pivot\_longer(cols = confirmed:active,

names\_to = "type",

values\_to = "cases")

# Global type cases

coW <- sum(as.numeric(covidW$confirmed), na.rm = TRUE)

reW <- sum(as.numeric(covidW$death), na.rm = TRUE)

deW <- sum(as.numeric(covidW$recovered), na.rm = TRUE)

acW <- sum(as.numeric(covidW$active), na.rm = TRUE)

# France cases.

covidFr <- covid %>% filter(country == "France")

# Daily new cases type for last 14 days in France. Also cumulative cases (sum of cases for each day)

frVis <- covidFr %>%

filter(date %in% tail(unique(as.Date(covidFr$date)), 14) ) %>%

select(date, type, cases) %>%

group\_by(date, type) %>%

summarise(cases = sum(cases, na.rm = TRUE)) %>%

group\_by(type) %>%

mutate(cumulative\_cases = cumsum(cases))

# Daily new cases for Visualization

frDaily <- frVis %>%

select(date, type, cases) %>%

pivot\_wider(names\_from = type,

values\_from = cases)

# Cumulative cases for Visualization

frCum <- frVis %>%

select(date, type, cumulative\_cases) %>%

pivot\_wider(names\_from = type,

values\_from = cumulative\_cases) %>%

mutate(active = confirmed - death - recovered) %>%

mutate(active = cumsum(active))

# Growth rate for past 6 months

frGrowth <- covidFr %>% filter(year == "21") %>%

select(month, type, cases) %>%

group\_by(month, type) %>% summarise(cases = sum(cases)) %>%

ungroup() %>%

group\_by(type) %>% mutate(growth = ((cases - lag(cases))/lag(cases)) \* 100) %>%

select(month, type, growth) %>%

pivot\_wider(names\_from = type,

values\_from = growth)

# Recovery rate for past 6 months.

frRecovery <- covidFr %>%

filter(year == "21") %>%

select(month, type, cases) %>%

group\_by(month, type) %>% summarise(cases = sum(cases)) %>%

ungroup() %>%

group\_by(type) %>%

pivot\_wider(names\_from = type,

values\_from = cases) %>%

mutate(recovery\_rate = (recovered / confirmed) \* 100) %>%

mutate(recovery\_rate = if\_else(is.na(recovery\_rate), 0, recovery\_rate))

# preparing for recovery rate visualization

frRecovery$rec\_cri <- ifelse(frRecovery$recovery\_rate <= 30 , frRecovery$recovery\_rate, NA)

frRecovery$rec\_avg <- ifelse(frRecovery$recovery\_rate >= 31 && frRecovery$recovery\_rate <= 60 , frRecovery$recovery\_rate, NA)

frRecovery$rec\_bet <- ifelse(frRecovery$recovery\_rate >= 61 , frRecovery$recovery\_rate, NA)

#-------------------------------Map-------------------------------------

# adding log cases for radius purpose

covidL <- covidL %>%

mutate(log\_cases = 2 \* log(cases))

# Split act as 3 data frames for 3 types

covidL.split <- covidL %>%

split(covidL$type)

# Map with different overlay

groupI <- names(covidL.split)

#-----------------------------Top 5 Countries---------------------------

# top 5 countries with most active cases

covCountry <- covidW %>%

ungroup() %>%

top\_n(5, active)

# Extract those top 5 country names

countnames <- covCountry$country

# Contains top 5 countries with their daily cases

coworld <- covid %>%

select(date, country, type, cases) %>%

group\_by(date, country, type) %>%

summarise(cases = sum(cases)) %>%

pivot\_wider(names\_from = type,

values\_from = cases) %>%

mutate(active = confirmed - death - recovered) %>%

filter(country %in% countnames) %>%

arrange(date)

# Now considering top 5 countries with their daily cases for past 14 days

tailWorld <- coworld %>%

tail(70, date) %>%

select(date, country, confirmed) %>%

pivot\_wider(names\_from = country,

values\_from = confirmed)

### Map

# Map with different overlay circles

leaflet() %>% addTiles() %>% addProviderTiles(providers$CartoDB.Positron) %>%

addCircleMarkers(data = covidL.split[[1]], lat = ~lat, lng = ~long,

color = "blue",

stroke = FALSE,

fillOpacity = 0.5,

radius = ~log\_cases,

popup = paste0("Country: ", covidL.split[[1]]$country,

"Type: ", covidL.split[[1]]$type,

"Cases: ", covidL.split[[1]]$cases),

group = groupI[1],

labelOptions = labelOptions(noHide = F,

direction = 'auto')) %>%

addCircleMarkers(data = covidL.split[[2]], lat = ~lat, lng = ~long,

color = "orange",

stroke = FALSE,

fillOpacity = 0.5,

radius = ~log\_cases,

popup = paste0("Country: ", covidL.split[[2]]$country,

"Type: ", covidL.split[[2]]$type,

"Cases: ", covidL.split[[2]]$cases),

group = groupI[2],

labelOptions = labelOptions(noHide = F,

direction = 'auto')) %>%

addCircleMarkers(data = covidL.split[[3]], lat = ~lat, lng = ~long,

color = "red",

stroke = FALSE,

fillOpacity = 0.5,

radius = ~log\_cases,

popup = paste0("Country: ", covidL.split[[3]]$country,

"Type: ", covidL.split[[3]]$type,

"Cases: ", covidL.split[[3]]$cases),

group = groupI[3],

labelOptions = labelOptions(noHide = F,

direction = 'auto')) %>%

addCircleMarkers(data = covidL.split[[4]], lat = ~lat, lng = ~long,

color = "green",

stroke = FALSE,

fillOpacity = 0.5,

radius = ~log\_cases,

popup = paste0("Country: ", covidL.split[[4]]$country,

"Type: ", covidL.split[[4]]$type,

"Cases: ", covidL.split[[4]]$cases),

group = groupI[4],

labelOptions = labelOptions(noHide = F,

direction = 'auto')) %>%

addLayersControl(overlayGroups = names(covidL.split),

options = layersControlOptions(collapsed = FALSE))

### Top 5 countries with cases

# Daily Confirmed Cases in Top 5 Countries for past 14 days (Log scale)

plot\_ly(data = tailWorld,

x = ~ date,

y = ~ US,

name = 'US',

mode = 'lines+markers',

type = 'scatter',

stackgroup = 'one') %>%

add\_trace(y = ~ France,

name = "France",

mode = 'lines+markers') %>%

add\_trace(y = ~ `United Kingdom`,

name = "United Kingdom",

mode = 'lines+markers') %>%

add\_trace(y = ~ Spain,

name = "Spain",

mode = 'lines+markers') %>%

add\_trace(y = ~ `Netherlands`,

name = "Netherlands",

mode = 'lines+markers') %>%

layout(title = "Daily Confirmed Cases in Top 5 Countries for past 14 days (Log scale)",

yaxis = list(title = "Daily Confirmed Cases",

type = "log"),

xaxis = list(title = "Date",

type = "date"),

legend = list(x = 0.1, y = 0.9),

hovermode = "compare")

### Daily New Cases

plot\_ly(frDaily, x = ~ date, y = ~ confirmed, name = 'Confirmed', color = 'rgba(243, 156, 18, 1)',

type = 'scatter', mode = 'lines') %>%

add\_trace(y = ~ recovered, name = "Recovered", color = 'rgba(46, 204, 113, 1)') %>%

add\_trace(y = ~ death, name = "Death", color = 'rgba(240, 52, 52, 1)') %>%

layout(title = "Daily New Cases in France for past 14 days", yaxis = list(title = "Daily New Cases"),

xaxis = list(title = "Date", type = "date"), autosize = FALSE, legend = list(x = 0.1, y = 0.9), hovermode = "compare")

### Cumulative Cases

plot\_ly(frCum, x = ~ date, y = ~ confirmed, name = 'Confirmed', color = 'rgba(243, 156, 18, 1)',

type = 'scatter', mode = 'lines') %>%

add\_trace(y = ~ recovered, name = "Recovered", color = 'rgba(46, 204, 113, 1)') %>%

add\_trace(y = ~ death, name = "Death", color = 'rgba(240, 52, 52, 1)') %>%

layout(title = "Cumulative Cases in France for past 14 days", yaxis = list(title = "Cumulative Cases"),

xaxis = list(title = "Date", type = "date"), autosize = FALSE, legend = list(x = 0.1, y = 0.9), hovermode = "compare")

### Growth Rate

plot\_ly(frGrowth, x = ~month, y = ~ confirmed, name = 'Confirmed', color = 'rgba(46, 204, 113, 1)',

type = 'scatter', mode = 'lines+markers') %>%

add\_trace(y = ~ recovered, name = "Recovered", color = 'rgba(240, 52, 52, 1)', type = 'scatter', mode = 'lines+markers') %>%

add\_trace(y = ~ death, name = "Death", color = 'rgba(243, 156, 18, 1)', type = 'scatter', mode = 'lines+markers') %>%

layout(title = "Growth rate in France for past 5 months", yaxis = list(title = "Growth rate"),

xaxis = list(

ticktext = list("Feb", "March", "April", "May", "June"),

tickvals = list(2, 3, 4, 5, 6),

tickmode = "array",

title = "Months"

), autosize = FALSE, legend = list(x = 0.1, y = 0.9), hovermode = "compare")

### Recovery

plot\_ly(frRecovery, x = ~month, y = ~ round(rec\_bet, 2), name = 'Better recovery rate', marker = list(color = 'green', size = 10),

type = 'scatter', mode = 'markers') %>%

add\_trace(y = ~ round(rec\_cri, 2), name = "Critical recovery rate", marker = list(color = 'red', size = 10),

type = 'scatter', mode = 'markers') %>%

add\_trace(y = ~ round(rec\_avg, 2), name = "Average recovery rate", marker = list(color = 'blue', size = 10),

type = 'scatter', mode = 'markers') %>%

layout(title = "Recovery rate in France for past 6 months", yaxis = list(title = "Recovery rate"),

xaxis = list(

ticktext = list("Jan", "Feb", "March", "April", "May", "June"),

tickvals = list(1, 2, 3, 4, 5, 6),

tickmode = "array",

title = "Months"

), autosize = FALSE)

options(warn = defaultWarn)

**References**

\* GitHub. 2021. COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. [online] Available at: [Accessed 9 June 2021].

\* Stack Overflow. 2021. [online] Available at: [Accessed 9 June 2021].

\* Plotly. 2021. [online] Available at: [Accessed 9 June 2021].

\* Shiny R Studio. 2021. [online] Available at: [Accessed 9 June 2021].

\* GitHub. 2021. [online] Available at: [Accessed 9 June 2021].