Final Project

Xueying Liu, Wei Liu, Jonathan Glass

The purpose of this project is to create a way to present and summarize Covid-19 data. As this virus grows the general public want two things from this information. They want to know immediate data as it emerges. They also want data that they can understand about their immediate surroundings. The target we attempted to convey in project was to attempt to satisfy these too things and create a dashboard that can relate the two things together. When selecting our data we thought about the recommended websites given to us. What we really liked about the data was how detailed it was and that it updated daily with new information. However, during October. The website stopped updating. We werent able to draw emerging data from the the website no longer. We then looked elsewhere for data that what satisfy our result. We then found a data source provide by John Hopkins university. Briefly viewing the data, we see a portion of the data and how it counts confirmed cases per day per region.

```
## Import data: Source from Johns Hopkins Github data
confirmed_data <- read.csv(
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti
deaths_data <- read.csv(
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti
recovered_data <- read.csv(
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti


# Data cleaning: To create country level and global level data
library(tidyr)
library(dplyr)
## Country level
confirmed <- confirmed_data %>% gather(key="date", value="confirmed", -c(Country.Region, Province.State

deaths <- deaths_data %>% gather(key="date", value="deaths", -c(Country.Region, Province.State, Lat, Lo

recovered <- recovered_data %>% gather(key="date", value="recovered", -c(Country.Region, Province.State


# Create final country level dataset: combine all three variables
country_data <- full_join(confirmed, deaths) %>% full_join(recovered)
country_data$date <- country_data$date %>% sub("X", "", .) %>% as.Date("%m.%d.%y")

# Create new variable: number of days
country_data <- country_data %>% group_by(Country.Region) %>% mutate(cumconfirmed=cumsum(confirmed), da

## World level
world_data <- country_data %>% group_by(date) %>% summarize(confirmed=sum(confirmed), cumconfirmed=sum(

# Specific countries levels
Albania <- country_data %>% filter(Country.Region=="Albania")
Australia <- country_data %>% filter(Country.Region=="Australia")
```

```
Brazil <- country_data %>% filter(Country.Region=="Brazil")
China <- country_data %>% filter(Country.Region=="China")
Cuba <- country_data %>% filter(Country.Region=="Cuba")
Egypt <- country_data %>% filter(Country.Region=="Egypt")
France <- country_data %>% filter(Country.Region=="France")
Iceland <- country_data %>% filter(Country.Region=="Iceland")
Indonesia <- country_data %>% filter(Country.Region=="Indonesia")
Italy <- country_data %>% filter(Country.Region=="Italy")
Japan <- country_data %>% filter(Country.Region=="Japan")
US <- country_data %>% filter(Country.Region=="US")

head(confirmed_data[,c(1:4,323:326)])
```

```
##   Province.State      Country.Region       Lat      Long X12.5.20 X12.6.20
## 1                        Afghanistan  33.93911  67.70995    47072    47306
## 2                            Albania  41.15330  20.16830    42148    42988
## 3                            Algeria  28.03390   1.65960    87502    88252
## 4                            Andorra  42.50630   1.52180     7005     7050
## 5                             Angola -11.20270  17.87390    15536    15591
## 6                 Antigua and Barbuda  17.06080 -61.79640      144      144
##   X12.7.20 X12.8.20
## 1    47516    47716
## 2    43683    44436
## 3    88825    89416
## 4     7084     7127
## 5    15648    15729
## 6      146      146
```

By using an algorithm that pulls directly from the website, we are at an advantage as each time the program is ran, we have the most updated data. As can be seen from the data above. The data confirms the most recent data of information provided from as early as yesterday. The data source tells us a few amounts of things. Including: the region in which we are counting cases, the location given by latitude and longitude. the counts of recovered, confirmed, and deaths for each day since January 22, 2020.

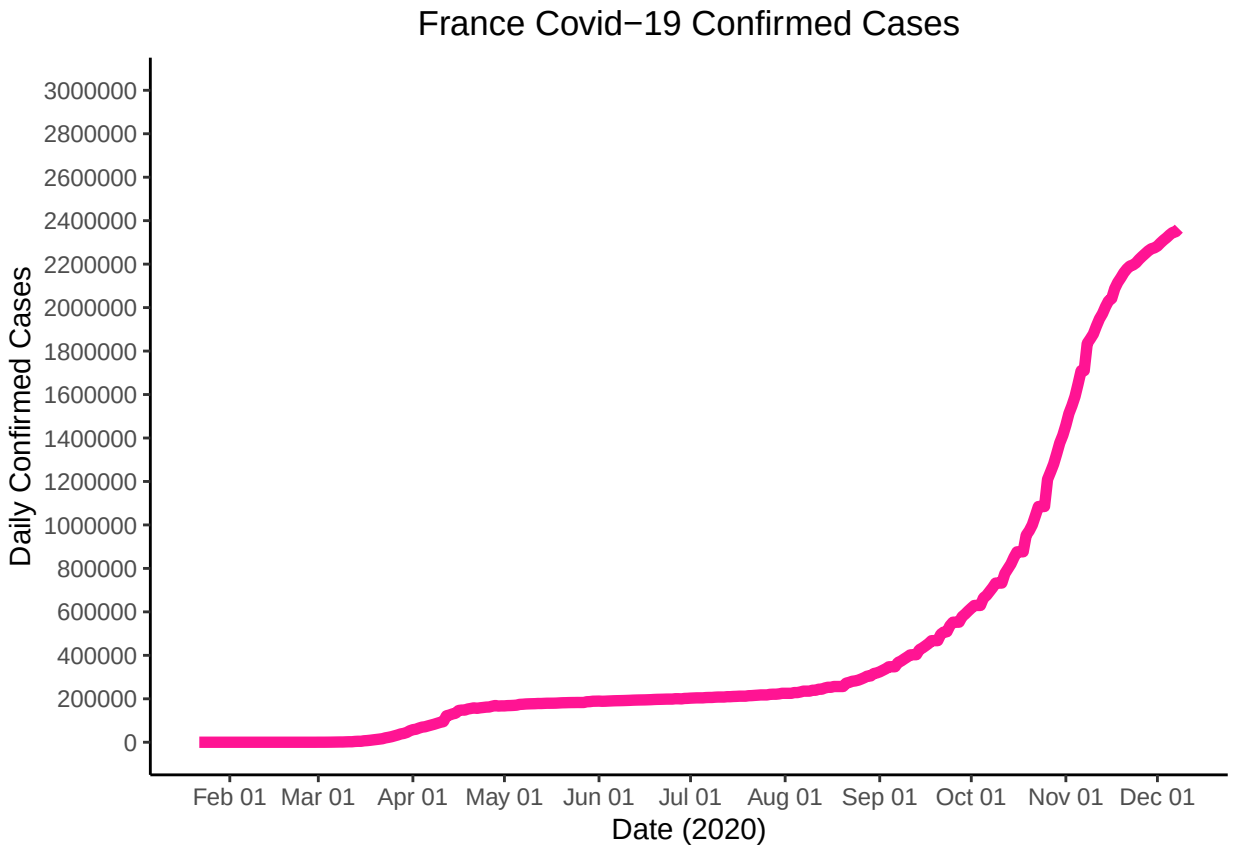
We adapted a shiny app to discuss the data via region to displaying trending data. below we have included the code at the end of the document. The app has two panels in which dives into the emerging data. The first part is a trend plot that shows a graphic over time for any given country we have data for, as well as the type of data you wish to view. confirmed cases, recovered cases, and death from cases. The second panel includes a spread of the world map that shows the magnitude of the confimed cases in the given region.

Regional data trend tracker

This is the time element to our project. Over the course from 1-22 to as recent as yesterday. This interactive applet can let pick the specific regional data you wish to view and then show you the data throughout time you are looking for. You may choose a setting for confirmed covid-19 cases, recovered cases, and deaths caused by confirmed cases. From there you may review any data that you would like. An example of what a graphic may look like is given below:

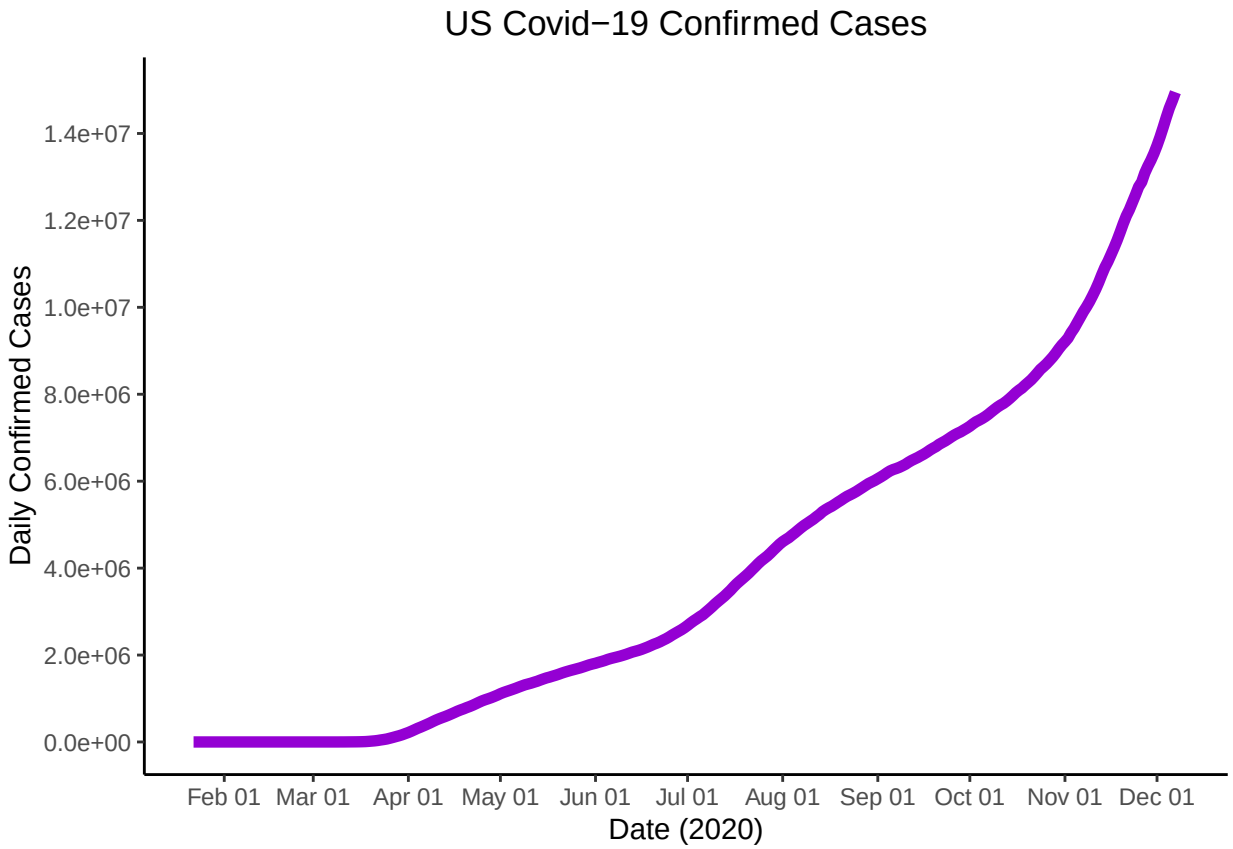
```
library(ggplot2)
Franceplot.confirmed <-ggplot(France, aes(x=date, y=confirmed)) +
  geom_line( color="deeppink", size=2, alpha=1.0) +
  labs(title = "France Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 3e+06), breaks = seq(0, 3e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
```

```
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))
Franceplot.confirmed
```



We are able to see specifically in this example that France, is currently on a very exponential rise of confirmed cases throughout the last few months. However, entering December it seems that the trend is that it will start to plateau. We note that through the summer months that there seemed to be a good system in place as the widespread number of cases were coming in at a dramatically lesser rate. As an observer, I could use this information to decide which time frame provide the results we desired and review public policies in place there to help 'flatten the curve'. Lets view the U.S.

```
USplot.confirmed <-ggplot(US, aes(x=date, y=confirmed)) +
  geom_line( color="darkviolet", size=2, alpha=1.0) +
  labs(title = "US Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1.5e+07), breaks = seq(0, 1.5e+07, by = 0.2e+07))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))
USplot.confirmed
```



As it can be seen from the United States. the data does not seem to 'plateau' anywhere. Suggesting that cases are continuously on the rise since the time of April 2020. The problem we see with this specific dataset though is that it does not include specific data for each state for the united states. Therefore, geographically speaking, the data covers a wide range of different populations over the different states. And with each state holding separate policies, judging the data here may be more of a challenge than that of the French coronavirus data. If we were able to adapt state data, this algorithm could be implemented as such to retain the same information but on a state level as well.

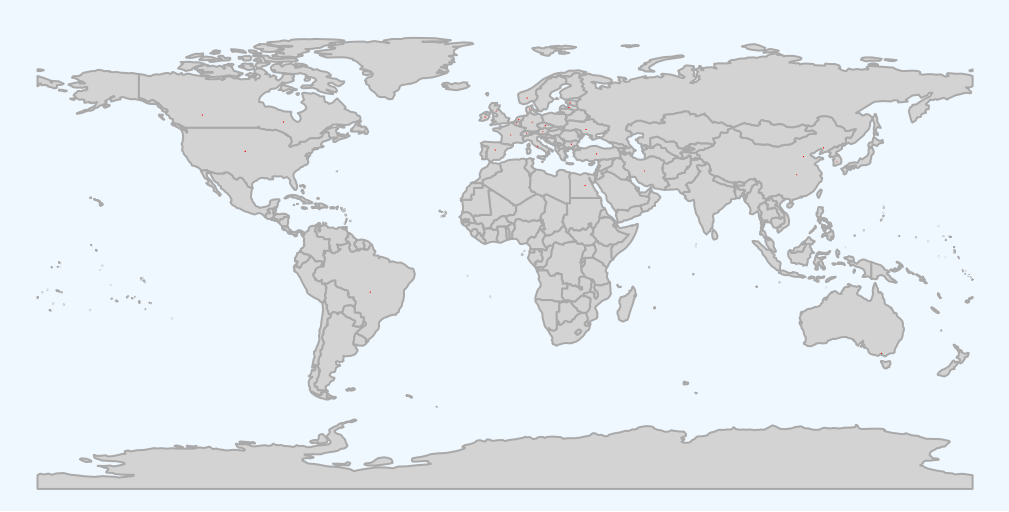
Map of Confirmed Cases Through the World

This panel will show throughout the entire world map the amount of confirmed cases. The graphic is created by taking the latitude and longitude data from the data set and plots it across an accurately placed World map. Over laying the points we are able to pinpoint each of the region in which we collect data from. The map is someone sparce once again for bigger countries such as the United states and India and Brazil as they dont seperate into different regions. The data points of each pinpointed region will contain a red point and the point will be bigger or smaller based on the magnitude of the confirmed case count. The bigger the point, the more of a 'hotspot' is compared to the rest of the world in terms of growing confirmed cases. Lets us view the data from a data back in April 1st, 2020.

```
#data coordinates
latitude<-confirmed_data$Lat
longitude<-confirmed_data$Long
```

```
#install.packages("rworldmap")
library(rworldmap)
library(ggplot2)
# get map
```

```
worldmap <- getMap(resolution = "coarse")
# plot world map
plot(worldmap, col = "lightgrey",
     fill = T, border = "darkgray",
     xlim = c(-180, 180), ylim = c(-90, 90),
     bg = "aliceblue",
     asp = 1, wrap=c(-180,180))
lines(longitude,latitude,type = "p",col="red",pch=21, bg=24, cex=confirmed_data$X4.1.20 /9990000, lwd=.4
```



```
#changing cex as a vector of 12/8s data divided by 9,990,000 (for scale)
#shows well in rmarkdown file but not as well in pdf
```

Note: the Knit to PDF doesn't render the map as well as Rstudios visualizer, please consider viewing through there or through the applet.

As we can see by the map. The spread amount of cases seem to be somewhat equal. No regional point seems to be tremendously bigger than any other region. Let us fast forward to December 8th, 2020.

```
#install.packages("rworldmap")
library(rworldmap)
library(ggplot2)
# get map
worldmap <- getMap(resolution = "coarse")
# plot world map
plot(worldmap, col = "lightgrey",
     fill = T, border = "darkgray",
     xlim = c(-180, 180), ylim = c(-90, 90),
     bg = "aliceblue",
     asp = 1, wrap=c(-180,180))
lines(longitude,latitude,type = "p",col="red",pch=21, bg=24, cex=confirmed_data$X12.8.20/9990000, lwd=.4
```

```
#changing cex as a vector of 12/8s data divided by 9,990,000 (for scale)
#shows well in rmarkdown file but not as well in pdf
```

From looking closely at the map we see some areas are now resulting in way bigger magnitude of confirmed cases. We can view this graph and get a since of scale how the world is progressing with the pandemic. The applet will allow the user to select any given data and they can view the spread of confirmed cases throughout time and study how it grew visually as we are ongoing this pandemic.

Appendix

Here we have included the code used for the applet. It will not be interactive through the pdf but you may click here to view the applet.

{r} [linked phrase] (https://xueying-liu.shinyapps.io/stat_5014_finalproject/)

```
#install.packages("shiny")
#install.packages("dplyr")
#install.packages("tidyr")
#install.packages("ggplot2")
#install.packages("lubridate")
library(ggplot2)
library(shiny)
library(lubridate)

ui <- fluidPage(
  navbarPage(collapsible = TRUE,
             "COVID-19 tracker", id="nav",inverse = TRUE,

             tabPanel("Region plots",
                      headerPanel('Covid-19 Dashboard'),
                      sidebarPanel(
                        selectInput("level_select", "Level:",
                                    choices = c("Global", "Country"),
                                    selected = c("Country"),
                                    multiple = FALSE),

                        selectInput("region_select", "Country/Region:",
                                    choices = c("Albania","Australia","Brazil","China","Cuba","Egypt","
```

```
                                    selected = c("US"),
                                    multiple = FALSE),

                      selectInput("outcome_select", "Outcome:",
                                    choices = c("Confirmed", "Deaths", "Recovered"),
                                    selected = c("Confirmed"),
                                    multiple = FALSE),
                      "Select outcome and regions to update region plots."
                    ),
                    mainPanel(
                      plotOutput('plot1')
                    )),
            tabPanel("World Map",

                    sliderInput("plot_date",
                                label = h5("Select mapping date"),
                                min = as.Date("2020-01-22","%Y-%m-%d"),
                                max = as.Date(current_date,"%Y-%m-%d"),
                                value = as.Date(current_date),width='100%'),
                    timeFormat="%Y-%m-%d",

                    mainPanel(plotOutput('map', width = "100%")))
  )

)

server <- function(input,output){
  plottest <- reactive({
    if (("Global" %in% input$level_select)&&("Confirmed" %in% input$outcome_select)) return(globalplot.
    if (("Country" %in% input$level_select) &&("Albania" %in% input$region_select)&&("Confirmed" %in% i
    if (("Country" %in% input$level_select) &&("Australia" %in% input$region_select)&&("Confirmed" %in%
    if (("Country" %in% input$level_select) &&("Brazil" %in% input$region_select)&&("Confirmed" %in% inp
    if (("Country" %in% input$level_select) &&("China" %in% input$region_select)&&("Confirmed" %in% inpu
    if (("Country" %in% input$level_select) &&("Cuba" %in% input$region_select)&&("Confirmed" %in% input
    if (("Country" %in% input$level_select) &&("Egypt" %in% input$region_select)&&("Confirmed" %in% inpu
    if (("Country" %in% input$level_select) &&("France" %in% input$region_select)&&("Confirmed" %in% inp
    if (("Country" %in% input$level_select) &&("Iceland" %in% input$region_select)&&("Confirmed" %in% i
    if (("Country" %in% input$level_select) &&("Indonesia" %in% input$region_select)&&("Confirmed" %in%
    if (("Country" %in% input$level_select) &&("Italy" %in% input$region_select)&&("Confirmed" %in% inpu
    if (("Country" %in% input$level_select) &&("Japan" %in% input$region_select)&&("Confirmed" %in% inpu
    if (("Country" %in% input$level_select) &&("US" %in% input$region_select)&&("Confirmed" %in% input$

    if (("Global" %in% input$level_select)&&("Recovered" %in% input$outcome_select)) return(globalplot.
    if (("Country" %in% input$level_select) &&("Albania" %in% input$region_select)&&("Recovered" %in% i
    if (("Country" %in% input$level_select) &&("Australia" %in% input$region_select)&&("Recovered" %in%
    if (("Country" %in% input$level_select) &&("Brazil" %in% input$region_select)&&("Recovered" %in% inp
    if (("Country" %in% input$level_select) &&("China" %in% input$region_select)&&("Recovered" %in% inpu
    if (("Country" %in% input$level_select) &&("Cuba" %in% input$region_select)&&("Recovered" %in% input
    if (("Country" %in% input$level_select) &&("Egypt" %in% input$region_select)&&("Recovered" %in% inpu
    if (("Country" %in% input$level_select) &&("France" %in% input$region_select)&&("Recovered" %in% inp
    if (("Country" %in% input$level_select) &&("Iceland" %in% input$region_select)&&("Recovered" %in% i
    if (("Country" %in% input$level_select) &&("Indonesia" %in% input$region_select)&&("Recovered" %in%
    if (("Country" %in% input$level_select) &&("Italy" %in% input$region_select)&&("Recovered" %in% inpu
```

```
    if (("Country" %in% input$level_select) &&("Japan" %in% input$region_select)&&("Recovered" %in% inpu
    if (("Country" %in% input$level_select) &&("US" %in% input$region_select)&&("Recovered" %in% input$

    if (("Global" %in% input$level_select)&&("Deaths" %in% input$outcome_select)) return(globalplot.deat
    if (("Country" %in% input$level_select) &&("Albania" %in% input$region_select)&&("Deaths" %in% input
    if (("Country" %in% input$level_select) &&("Australia" %in% input$region_select)&&("Deaths" %in% inp
    if (("Country" %in% input$level_select) &&("Brazil" %in% input$region_select)&&("Deaths" %in% input$
    if (("Country" %in% input$level_select) &&("China" %in% input$region_select)&&("Deaths" %in% input$c
    if (("Country" %in% input$level_select) &&("Cuba" %in% input$region_select)&&("Deaths" %in% input$ou
    if (("Country" %in% input$level_select) &&("Egypt" %in% input$region_select)&&("Deaths" %in% input$c
    if (("Country" %in% input$level_select) &&("France" %in% input$region_select)&&("Deaths" %in% input$
    if (("Country" %in% input$level_select) &&("Iceland" %in% input$region_select)&&("Deaths" %in% input
    if (("Country" %in% input$level_select) &&("Indonesia" %in% input$region_select)&&("Deaths" %in% inp
    if (("Country" %in% input$level_select) &&("Italy" %in% input$region_select)&&("Deaths" %in% input$c
    if (("Country" %in% input$level_select) &&("Japan" %in% input$region_select)&&("Deaths" %in% input$c
    if (("Country" %in% input$level_select) &&("US" %in% input$region_select)&&("Deaths" %in% input$outc
  })

  output$plot1 <- renderPlot({
    par(mar=c(5.1,4.1,0,1))
    dataplots <- plottest()
    print(dataplots)

  })

  output$map <- renderPlot(map(date = paste('X',month(input$plot_date),".",day(input$plot_date),".","20
}




## Import data: Source from Johns Hopkins Github data
confirmed_data <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
deaths_data <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_
recovered_data <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_


# Data cleaning: To create country level and global level data
library(tidyr)
library(dplyr)
## Country level
confirmed <- confirmed_data %>% gather(key="date", value="confirmed", -c(Country.Region, Province.State

deaths <- deaths_data %>% gather(key="date", value="deaths", -c(Country.Region, Province.State, Lat, Lon

recovered <- recovered_data %>% gather(key="date", value="recovered", -c(Country.Region, Province.State


# Create final country level dataset: combine all three variables
country_data <- full_join(confirmed, deaths) %>% full_join(recovered)
country_data$date <- country_data$date %>% sub("X", "", .) %>% as.Date("%m.%d.%y")
```

```r
# Create new variable: number of days
country_data <- country_data %>% group_by(Country.Region) %>% mutate(cumconfirmed=cumsum(confirmed), da

## World level
world_data <- country_data %>% group_by(date) %>% summarize(confirmed=sum(confirmed), cumconfirmed=sum(

# Specific countries levels
Albania <- country_data %>% filter(Country.Region=="Albania")
Australia <- country_data %>% filter(Country.Region=="Australia")
Brazil <- country_data %>% filter(Country.Region=="Brazil")
China <- country_data %>% filter(Country.Region=="China")
Cuba <- country_data %>% filter(Country.Region=="Cuba")
Egypt <- country_data %>% filter(Country.Region=="Egypt")
France <- country_data %>% filter(Country.Region=="France")
Iceland <- country_data %>% filter(Country.Region=="Iceland")
Indonesia <- country_data %>% filter(Country.Region=="Indonesia")
Italy <- country_data %>% filter(Country.Region=="Italy")
Japan <- country_data %>% filter(Country.Region=="Japan")
US <- country_data %>% filter(Country.Region=="US")

## confirmed
globalplot.confirmed <-ggplot(world_data, aes(x=date, y=confirmed)) +
  geom_line( color="#69b3a2", size=2, alpha=1.0) +
  labs(title = "Covid-19 Global Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 7e+07), breaks = seq(0, 7e+07, by = 1e+07))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Albaniaplot.confirmed <-ggplot(Albania, aes(x=date, y=confirmed)) +
  geom_line( color="coral1", size=2, alpha=1.0) +
  labs(title = "Albania Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Australiaplot.confirmed <- ggplot(Australia, aes(x=date, y=confirmed)) +
  geom_line( color="brown", size=2, alpha=1.0) +
  labs(title = "Australia Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Brazilplot.confirmed <-ggplot(Brazil, aes(x=date, y=confirmed)) +
  geom_line( color="darkgreen", size=2, alpha=1.0) +
  labs(title = "Brazil Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1.5e+07), breaks = seq(0, 1.5e+07, by = 0.2e+07))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))
```

```
Chinaplot.confirmed <-ggplot(China, aes(x=date, y=confirmed)) +
  geom_line( color="darkorange", size=2, alpha=1.0) +
  labs(title = "China Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1e+06), breaks = seq(0, 1e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Cubaplot.confirmed <-ggplot(Cuba, aes(x=date, y=confirmed)) +
  geom_line( color="darkorchid", size=2, alpha=1.0) +
  labs(title = "Cuba Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1e+06), breaks = seq(0, 1e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Egyptplot.confirmed <-ggplot(Egypt, aes(x=date, y=confirmed)) +
  geom_line( color="cadetblue", size=2, alpha=1.0) +
  labs(title = "Egypt Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Franceplot.confirmed <-ggplot(France, aes(x=date, y=confirmed)) +
  geom_line( color="deeppink", size=2, alpha=1.0) +
  labs(title = "France Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 3e+06), breaks = seq(0, 3e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Icelandplot.confirmed <-ggplot(Iceland, aes(x=date, y=confirmed)) +
  geom_line( color="darkseagreen", size=2, alpha=1.0) +
  labs(title = "Iceland Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1e+05), breaks = seq(0, 1e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Indonesiaplot.confirmed <-ggplot(Indonesia, aes(x=date, y=confirmed)) +
  geom_line( color="deepskyblue4", size=2, alpha=1.0) +
  labs(title = "Indonesia Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1e+06), breaks = seq(0, 1e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Italyplot.confirmed <-ggplot(Italy, aes(x=date, y=confirmed)) +
  geom_line( color="coral1", size=2, alpha=1.0) +
  labs(title = "Italy Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 2e+06), breaks = seq(0, 2e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
```

```
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Japanplot.confirmed <-ggplot(Japan, aes(x=date, y=confirmed)) +
  geom_line( color="Black", size=2, alpha=1.0) +
  labs(title = "Japan Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1e+06), breaks = seq(0, 1e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

USplot.confirmed <-ggplot(US, aes(x=date, y=confirmed)) +
  geom_line( color="darkviolet", size=2, alpha=1.0) +
  labs(title = "US Covid-19 Confirmed Cases", x="Date (2020)", y="Daily Confirmed Cases")+
  scale_y_continuous(limits = c(0, 1.5e+07), breaks = seq(0, 1.5e+07, by = 0.2e+07))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

## recovered
globalplot.recovered <-ggplot(world_data, aes(x=date, y=recovered)) +
  geom_line( color="#69b3a2", size=2, alpha=1.0) +
  labs(title = "Covid-19 Global Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 7e+07), breaks = seq(0, 7e+07, by = 1e+07))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Albaniaplot.recovered <-ggplot(Albania, aes(x=date, y=recovered)) +
  geom_line( color="coral1", size=2, alpha=1.0) +
  labs(title = "Albania Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Australiaplot.recovered <- ggplot(Australia, aes(x=date, y=recovered)) +
  geom_line( color="brown", size=2, alpha=1.0) +
  labs(title = "Australia Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Brazilplot.recovered <-ggplot(Brazil, aes(x=date, y=recovered)) +
  geom_line( color="darkgreen", size=2, alpha=1.0) +
  labs(title = "Brazil Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 1.5e+07), breaks = seq(0, 1.5e+07, by = 0.2e+07))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Chinaplot.recovered <-ggplot(China, aes(x=date, y=recovered)) +
```

```
  geom_line( color="darkorange", size=2, alpha=1.0) +
  labs(title = "China Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Cubaplot.recovered <-ggplot(Cuba, aes(x=date, y=recovered)) +
  geom_line( color="darkorchid", size=2, alpha=1.0) +
  labs(title = "Cuba Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Egyptplot.recovered <-ggplot(Egypt, aes(x=date, y=recovered)) +
  geom_line( color="cadetblue", size=2, alpha=1.0) +
  labs(title = "Egypt Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Franceplot.recovered <-ggplot(France, aes(x=date, y=recovered)) +
  geom_line( color="deeppink", size=2, alpha=1.0) +
  labs(title = "France Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 3e+05), breaks = seq(0, 3e+05, by = 0.3e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Icelandplot.recovered <-ggplot(Iceland, aes(x=date, y=recovered)) +
  geom_line( color="darkseagreen", size=2, alpha=1.0) +
  labs(title = "Iceland Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 1e+05), breaks = seq(0, 1e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Indonesiaplot.recovered <-ggplot(Indonesia, aes(x=date, y=recovered)) +
  geom_line( color="deepskyblue4", size=2, alpha=1.0) +
  labs(title = "Indonesia Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 1e+06), breaks = seq(0, 1e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Italyplot.recovered <-ggplot(Italy, aes(x=date, y=recovered)) +
  geom_line( color="coral1", size=2, alpha=1.0) +
  labs(title = "Italy Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 1e+06), breaks = seq(0, 1e+06, by = 0.2e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
```

```
  theme(plot.title = element_text(hjust = 0.5))

Japanplot.recovered <-ggplot(Japan, aes(x=date, y=recovered)) +
  geom_line( color="Black", size=2, alpha=1.0) +
  labs(title = "Japan Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 2e+05), breaks = seq(0, 2e+05, by = 0.2e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

USplot.recovered <-ggplot(US, aes(x=date, y=recovered)) +
  geom_line( color="darkviolet", size=2, alpha=1.0) +
  labs(title = "US Covid-19 Recovered Cases", x="Date (2020)", y="Daily Recovered Cases")+
  scale_y_continuous(limits = c(0, 8e+06), breaks = seq(0, 8e+06, by = 0.8e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

## death
globalplot.deaths <-ggplot(world_data, aes(x=date, y=deaths)) +
  geom_line( color="#69b3a2", size=2, alpha=1.0) +
  labs(title = "Covid-19 Global Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 2e+06), breaks = seq(0, 2e+06, by = 0.3e+06))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Albaniaplot.deaths <-ggplot(Albania, aes(x=date, y=deaths)) +
  geom_line( color="coral1", size=2, alpha=1.0) +
  labs(title = "Albania Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Australiaplot.deaths <- ggplot(Australia, aes(x=date, y=deaths)) +
  geom_line( color="brown", size=2, alpha=1.0) +
  labs(title = "Australia Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Brazilplot.deaths <-ggplot(Brazil, aes(x=date, y=deaths)) +
  geom_line( color="darkgreen", size=2, alpha=1.0) +
  labs(title = "Brazil Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 3e+05), breaks = seq(0, 3e+05, by = 0.5e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Chinaplot.deaths <-ggplot(China, aes(x=date, y=deaths)) +
  geom_line( color="darkorange", size=2, alpha=1.0) +
```

```
  labs(title = "China Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Cubaplot.deaths <-ggplot(Cuba, aes(x=date, y=deaths)) +
  geom_line( color="darkorchid", size=2, alpha=1.0) +
  labs(title = "Cuba Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Egyptplot.deaths <-ggplot(Egypt, aes(x=date, y=deaths)) +
  geom_line( color="cadetblue", size=2, alpha=1.0) +
  labs(title = "Egypt Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Franceplot.deaths <-ggplot(France, aes(x=date, y=deaths)) +
  geom_line( color="deeppink", size=2, alpha=1.0) +
  labs(title = "France Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Icelandplot.deaths <-ggplot(Iceland, aes(x=date, y=deaths)) +
  geom_line( color="darkseagreen", size=2, alpha=1.0) +
  labs(title = "Iceland Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Indonesiaplot.deaths <-ggplot(Indonesia, aes(x=date, y=deaths)) +
  geom_line( color="deepskyblue4", size=2, alpha=1.0) +
  labs(title = "Indonesia Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

Italyplot.deaths <-ggplot(Italy, aes(x=date, y=deaths)) +
  geom_line( color="coral1", size=2, alpha=1.0) +
  labs(title = "Italy Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))
```

```
Japanplot.deaths <-ggplot(Japan, aes(x=date, y=deaths)) +
  geom_line( color="Black", size=2, alpha=1.0) +
  labs(title = "Japan Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 5e+04), breaks = seq(0, 5e+04, by = 1e+04))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))

USplot.deaths <-ggplot(US, aes(x=date, y=deaths)) +
  geom_line( color="darkviolet", size=2, alpha=1.0) +
  labs(title = "US Covid-19 Deaths Cases", x="Date (2020)", y="Daily Deaths Cases")+
  scale_y_continuous(limits = c(0, 3e+05), breaks = seq(0, 3e+05, by = 0.5e+05))+
  scale_x_date(date_breaks = "1 month", date_labels = "%b %d")+
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))


### Map
#data cordinates
latitude<-confirmed_data$Lat
longitude<-confirmed_data$Long

library(rworldmap)
library(ggplot2)
current_date_raw <- tail(colnames(confirmed_data),1)
current_date <- paste("20",strsplit(sub(".","",current_date_raw),"\\.")[[1]][3],"-",strsplit(sub(".",""

map<-function(date){
  worldmap <- getMap(resolution = "coarse")
  plot(worldmap, col = "lightgrey", border = "darkgray",
       xlim = c(-180, 180), ylim = c(-90, 90),
       bg = "aliceblue",
       asp = 1)
  points(longitude,latitude,type = "p",col="red",pch=21, bg=24, cex=confirmed_data[,which(colnames(conf
}



shinyApp(ui=ui,server = server)
```