Coronavirus Forecasting Using Recurrent Neural Networks

Soumyakant Sahoo (Sam)

09/03/2020

Load Data

```
# Confirmed COVID-19 cases
corona_confirmed = read.csv("time_series_19_covid_confirmed.csv",header = TRUE)
# COVID-19 death cases
corona_deaths = read.csv("time_series_19_covid_deaths.csv",header = TRUE)
# COVID-19 recovered cases
corona_recovered = read.csv("time_series_19_covid_recovered.csv",header = TRUE)
```

Data Pre-processing

```
# Renaming the date column
colnames(corona_confirmed)[1] = "Date"
colnames(corona_deaths)[1] = "Date"
colnames(corona_recovered)[1] = "Date"
```

Latitude, Longitude and Province columns were not important for analysis hence those were dropped. Since each country had multiple provinces, the counts for all the provinces for each country were aggregated to form the counts for the country.

```
# Longitude, Latitude and Province columns are dropped
corona_confirmed <- corona_confirmed[3:nrow(corona_confirmed),]
corona_deaths <- corona_deaths[3:nrow(corona_deaths),]
corona_recovered <- corona_recovered[3:nrow(corona_recovered),]

cnames <- tolower(colnames(corona_confirmed))

# Aggregate counts of confirmed cases for countries with multiple provinces
new_df<-data.frame(China=rowSums(corona_confirmed[, grep("china", cnames)]),
         US=rowSums(corona_confirmed[, grep("us", cnames)]),
         Australia=rowSums(corona_confirmed[, grep("australia", cnames)]),
         France=rowSums(corona_confirmed[, grep("france", cnames)]),
         UK=rowSums(corona_confirmed[, grep("united", cnames)]),
         Denmark=rowSums(corona_confirmed[, grep("denmark", cnames)]),
         Canada=rowSums(corona_confirmed[, grep("canada", cnames)]))

# Once the aggregated count of country are created. Drop the existing columns
corona_confirmed<- corona_confirmed[,-grep("China",colnames(corona_confirmed))]
corona_confirmed<- corona_confirmed[,-grep("US",colnames(corona_confirmed))]
corona_confirmed<- corona_confirmed[,-grep("Australia",colnames(corona_confirmed))]
```

```r
corona_confirmed<- corona_confirmed[,-grep("France",colnames(corona_confirmed))]
corona_confirmed<- corona_confirmed[,-grep("Canada",colnames(corona_confirmed))]
corona_confirmed<- corona_confirmed[,-grep("United",colnames(corona_confirmed))]
corona_confirmed<- corona_confirmed[,-grep("Denmark",colnames(corona_confirmed))]

corona_confirmed<- cbind(corona_confirmed,new_df)


cnames <- tolower(colnames(corona_deaths))

# Aggregate counts of death cases for countries with multiple provinces
new_df<-data.frame(China=rowSums(corona_deaths[, grep("china", cnames)]),
         US=rowSums(corona_deaths[, grep("us", cnames)]),
         Australia=rowSums(corona_deaths[, grep("australia", cnames)]),
         France=rowSums(corona_deaths[, grep("france", cnames)]),
         UK=rowSums(corona_deaths[, grep("united", cnames)]),
         Denmark=rowSums(corona_deaths[, grep("denmark", cnames)]),
         Canada=rowSums(corona_deaths[, grep("canada", cnames)]))

# Once the aggregated count of country are created. Drop the existing columns
corona_deaths<- corona_deaths[,-grep("China",colnames(corona_deaths))]
corona_deaths<- corona_deaths[,-grep("US",colnames(corona_deaths))]
corona_deaths<- corona_deaths[,-grep("Australia",colnames(corona_deaths))]
corona_deaths<- corona_deaths[,-grep("France",colnames(corona_deaths))]
corona_deaths<- corona_deaths[,-grep("Canada",colnames(corona_deaths))]
corona_deaths<- corona_deaths[,-grep("United",colnames(corona_deaths))]
corona_deaths<- corona_deaths[,-grep("Denmark",colnames(corona_deaths))]

corona_deaths<- cbind(corona_deaths,new_df)


cnames <- tolower(colnames(corona_recovered))

# Aggregate counts of recovered cases for countries with multiple provinces
new_df<-data.frame(China=rowSums(corona_recovered[, grep("china", cnames)]),
         US=rowSums(corona_recovered[, grep("us", cnames)]),
         Australia=rowSums(corona_recovered[, grep("australia", cnames)]),
         France=rowSums(corona_recovered[, grep("france", cnames)]),
         UK=rowSums(corona_recovered[, grep("united", cnames)]),
         Denmark=rowSums(corona_recovered[, grep("denmark", cnames)]),
         Canada=rowSums(corona_recovered[, grep("canada", cnames)]))

# Once the aggregated count of country are created. Drop the existing columns
corona_recovered<- corona_recovered[,-grep("China",colnames(corona_recovered))]
corona_recovered<- corona_recovered[,-grep("US",colnames(corona_recovered))]
corona_recovered<- corona_recovered[,-grep("Australia",colnames(corona_recovered))]
corona_recovered<- corona_recovered[,-grep("France",colnames(corona_recovered))]
corona_recovered<- corona_recovered[,-grep("Canada",colnames(corona_recovered))]
corona_recovered<- corona_recovered[,-grep("United",colnames(corona_recovered))]
corona_recovered<- corona_recovered[,-grep("Denmark",colnames(corona_recovered))]

corona_recovered<- cbind(corona_recovered,new_df)
```

```
# Format the date column and convert to factor
dates <- as.Date(factor(corona_confirmed$Date),format = "%Y-%m-%d")
```

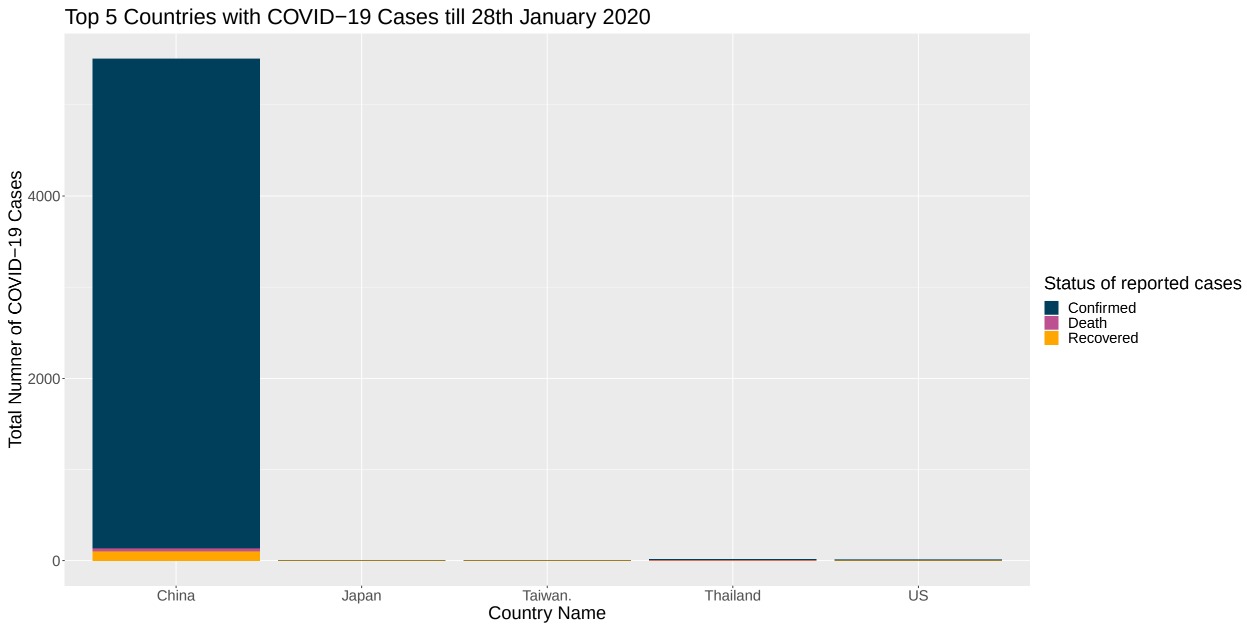
Exploratory Data Analysis

Visualize the Initial Outbreak of Coronavirus

- (Timeline: 31st December 2019 - 28th January 2020)

The epicenter of initial outbreak as visualized was China with 5509 corona virus cases reported in a months time with extremely low recovery rate of 101 patients and 131 deaths. The virus spread to the other countries through people who travelled to China and later reported ill after returning back.

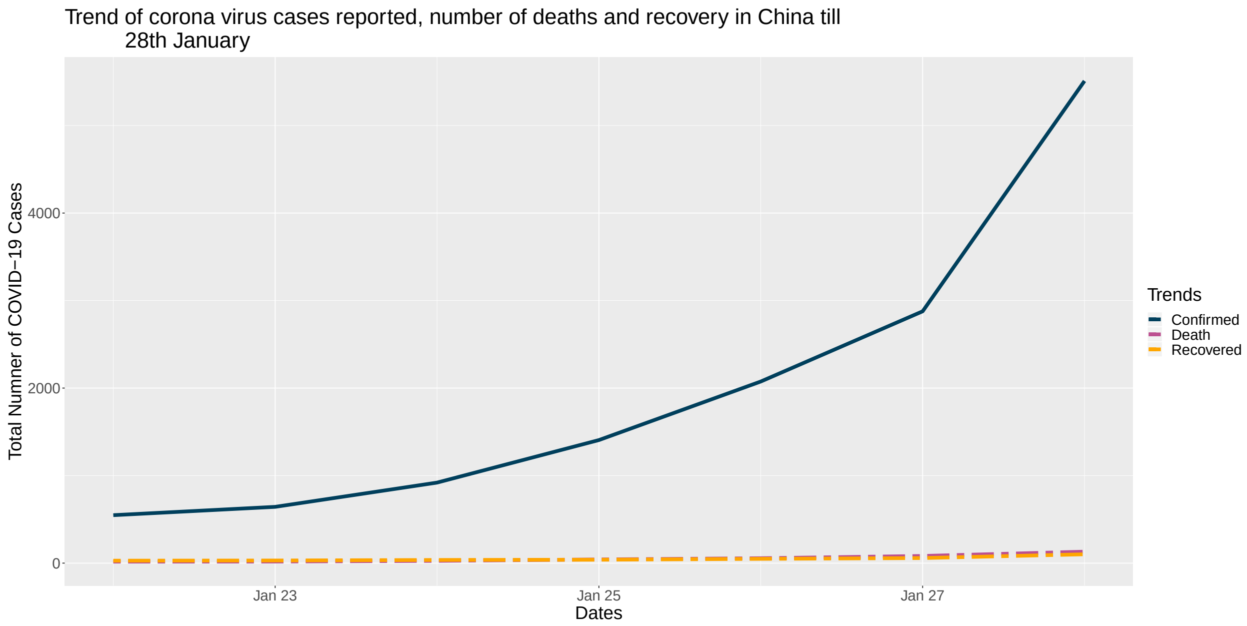
```
country_name <- colnames(corona_confirmed[2:ncol(corona_confirmed)])
countries <- data.frame(country_name)
counts_28th <- transpose(corona_confirmed[7,2:ncol(corona_confirmed)])
deaths_28th <- transpose(corona_deaths[7,2:ncol(corona_deaths)])
colnames(deaths_28th) <- c("Death_count")
recovered_28th <- transpose(corona_recovered[7,2:ncol(corona_recovered)])
colnames(recovered_28th) <- c("Recovered_count")

df <-cbind(countries,counts_28th,deaths_28th,recovered_28th)

df<-arrange(df, desc(V1))
ggplot(df[1:5,], aes(x=country_name))+
  geom_bar(aes(y=V1,fill='Confirmed'),stat = "identity")+
  geom_bar(aes(y=Death_count,fill='Death'),stat = "identity")+
  geom_bar(aes(y=Recovered_count,fill='Recovered'),stat = "identity")+
  scale_fill_manual(name="Status of reported cases",values = c('Confirmed' = '#003f5c', 'Death' = '#bc5
  xlab("Country Name") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Top 5 Countries with COVID-19 Cases till 28th January 2020") + # Set title
  theme(text = element_text(size=21))
```



Visualize the Trend of Confirmed, Death and Recovery cases till 28th January

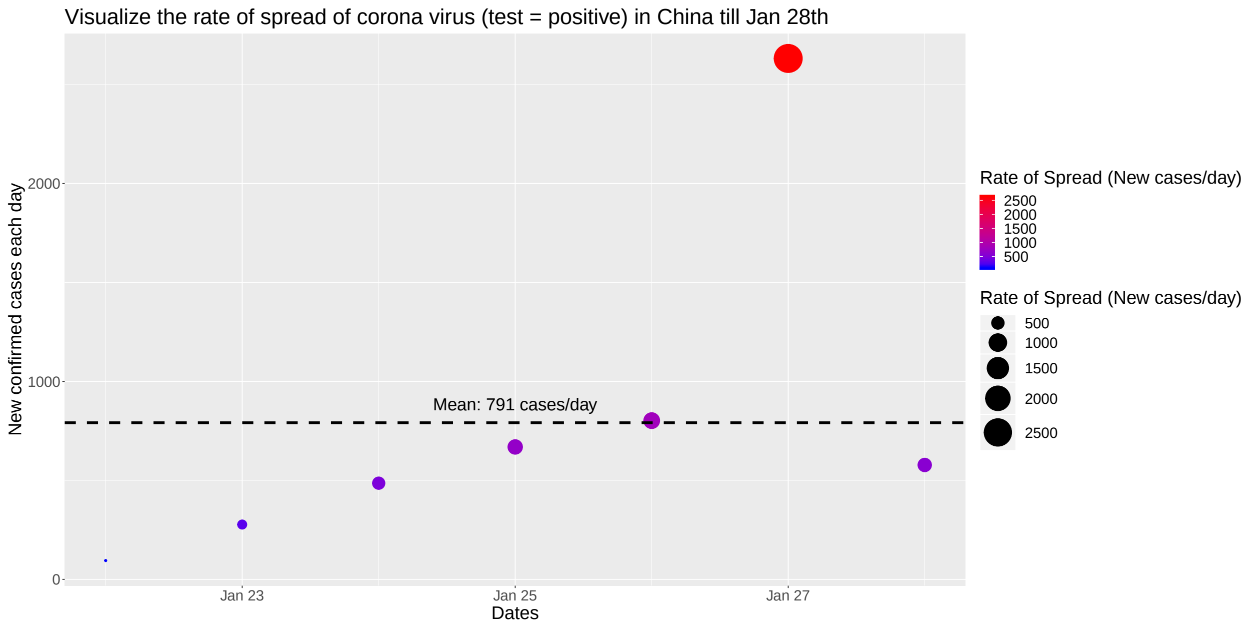
The number of corona virus cases in China continued to peak as the efforts were made by the government to flatten the curve and contain the spread of virus.

```
ggplot(corona_confirmed[1:7,],aes(x = dates[1:7]))+
  geom_line(aes(y = corona_confirmed$China[1:7],colour='Confirmed'),size=2)+
  geom_line(aes(y = corona_deaths$China[1:7],colour='Death'),linetype="twodash",size=2)+
  geom_line(aes(y = corona_recovered$China[1:7],colour='Recovered'),linetype="twodash",size=2)+
  scale_color_manual(name = "Trends", values = c('Confirmed' = '#003f5c', 'Death' = '#bc5090','Recovere
  xlab("Dates") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus cases reported, number of deaths and recovery in China till
          28th January") + # Set title
  theme(text = element_text(size=21))
```



Visualize the Trend of Rate of Spread Virus till 28th January

```r
rate_of_spread = 0

for (i in (1:7)){
  rate_of_spread[i] = corona_confirmed$China[i+1] - corona_confirmed$China[i]
}

ggplot(NULL, aes(x=dates[1:7],y=rate_of_spread))+
  geom_point(aes(size = rate_of_spread,color=rate_of_spread))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in China till Jan 28th")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean(rate_of_spread),color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-01-25"),y=mean(rate_of_spread),label ="Mean: 791 cases/day",
              vjust = -1),size=7)+
  theme(text = element_text(size=21))
```



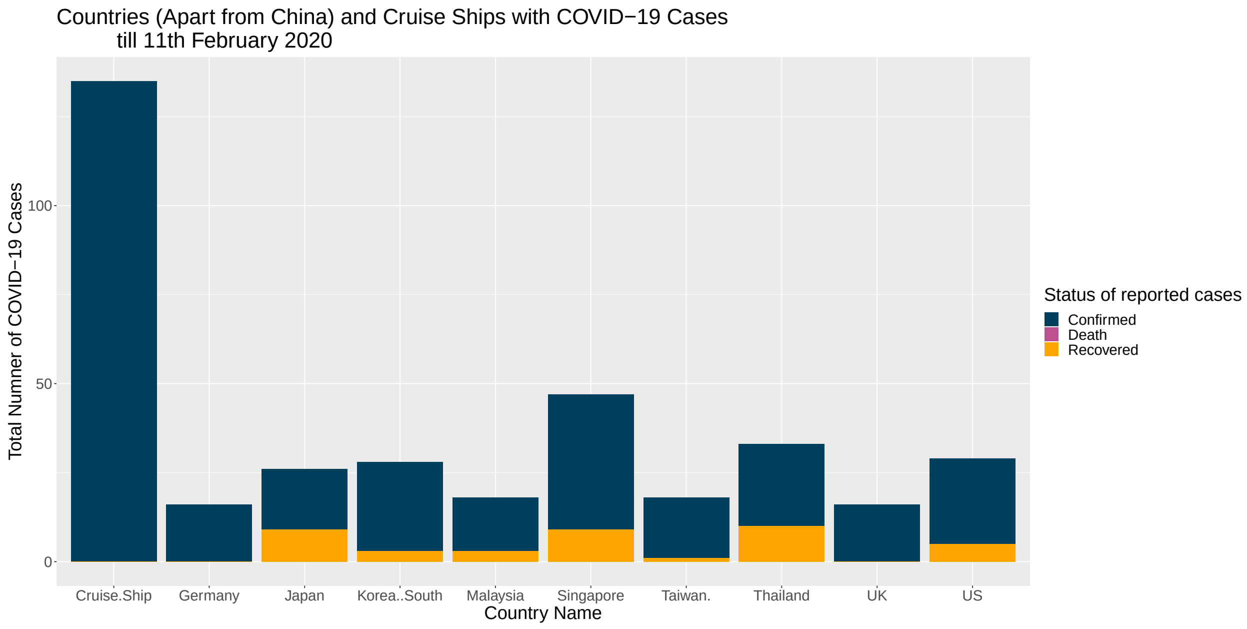
The number of new confirmed cases in China saw a steady increase each day with single highest increase on Jan 27th with 2632 cases confirmed on a single day. The mean number of confirmed new cases was 791.2857143 during the first 30 days of outbreak.

Visualize the Global Spread of Corona Virus till 11th February

- (Timeline: 22nd January 2020 - 11th February 2020)

By 11th Feb 2020, coronavirus spread to more countries around the world and over 100 cases were reported in cruise ships on which people were quarantined and were not allowed to de-board. However, it can be seen due to coordinated efforts and planning among the countries, some countries like Japan, Thailand had good recovery rates for patients.

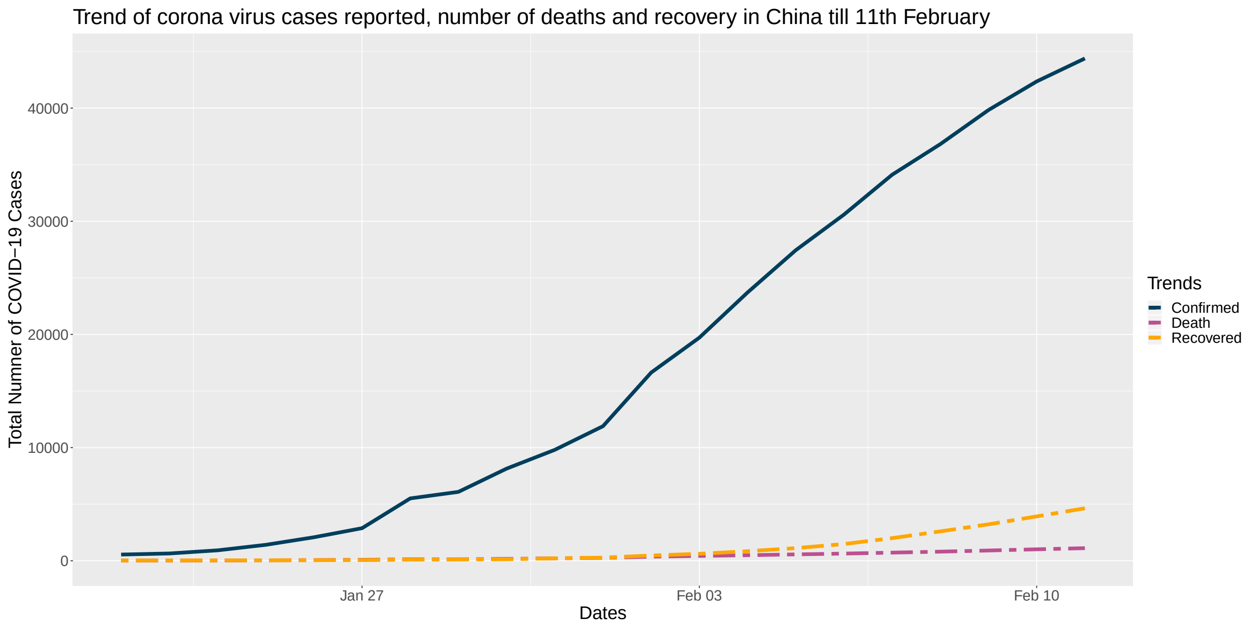
```
counts_11th <- transpose(corona_confirmed[21,2:ncol(corona_confirmed)])
deaths_11th <- transpose(corona_deaths[21,2:ncol(corona_deaths)])
colnames(deaths_11th) <- c("Death_count")
recovered_11th <- transpose(corona_recovered[21,2:ncol(corona_recovered)])
colnames(recovered_11th) <- c("Recovered_count")

df <-cbind(countries,counts_11th,deaths_11th,recovered_11th)

df<-arrange(df, desc(V1))
ggplot(df[2:11,], aes(x=country_name))+
  geom_bar(aes(y=V1,fill='Confirmed'),stat = "identity")+
  geom_bar(aes(y=Death_count,fill='Death'),stat = "identity")+
  geom_bar(aes(y=Recovered_count,fill='Recovered'),stat = "identity")+
  scale_fill_manual(name="Status of reported cases",values = c('Confirmed' = '#003f5c', 'Death' = '#bc50
  xlab("Country Name") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Countries (Apart from China) and Cruise Ships with COVID-19 Cases
          till 11th February 2020") + # Set title
  theme(text = element_text(size=21))
```



Visualize the Trend of Confirmed, Death and Recovery cases till 11th February

The number of confirmed cases increased 10 times in 10 days from January 28th with more than 40000 people now infected by the COVID-19 virus in China.However, governments decision for a complete lockdown seems to have started flattening the confirmed cases curve.

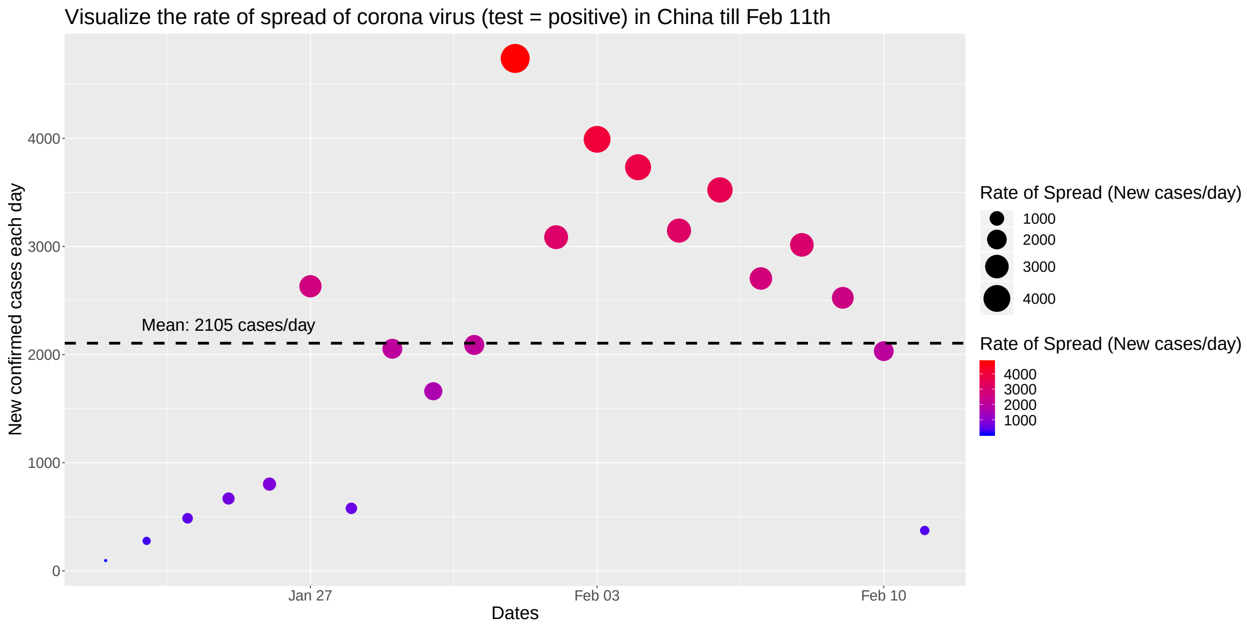
```
ggplot(corona_confirmed[1:21,],aes(x = dates[1:21]))+
  geom_line(aes(y = corona_confirmed$China[1:21],colour='Confirmed'),size=2)+
  geom_line(aes(y = corona_deaths$China[1:21],colour='Death'),linetype="twodash",size=2)+
  geom_line(aes(y = corona_recovered$China[1:21],colour='Recovered'),linetype="twodash",size=2)+
  scale_color_manual(name = "Trends", values = c('Confirmed' = '#003f5c', 'Death' = '#bc5090','Recovere
  xlab("Dates") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus cases reported, number of deaths and recovery in China till 11th Februa
  theme(text = element_text(size=21))
```



Visualize the Trend of Rate of Spread Virus till 11th February

The rate of spread of COVID-19 was rapid as the average number of cases reported in China more than doubled from 791 to 2105 cases per day by Feb 11th. However a declining rate of spread pattern can be seen with lesser number of cases reported after February 3rd.

```r
rate_of_spread = 0

for (i in (1:21)){
  rate_of_spread[i] = corona_confirmed$China[i+1] - corona_confirmed$China[i]
}

ggplot(NULL, aes(x=dates[1:21],y=rate_of_spread))+
  geom_point(aes(size = rate_of_spread,color=rate_of_spread))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in China till Feb 11th")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean(rate_of_spread),color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-01-25"),y=mean(rate_of_spread),label ="Mean: 2105 cases/day",
              vjust = -1),size=7)+
  theme(text = element_text(size=21))
```

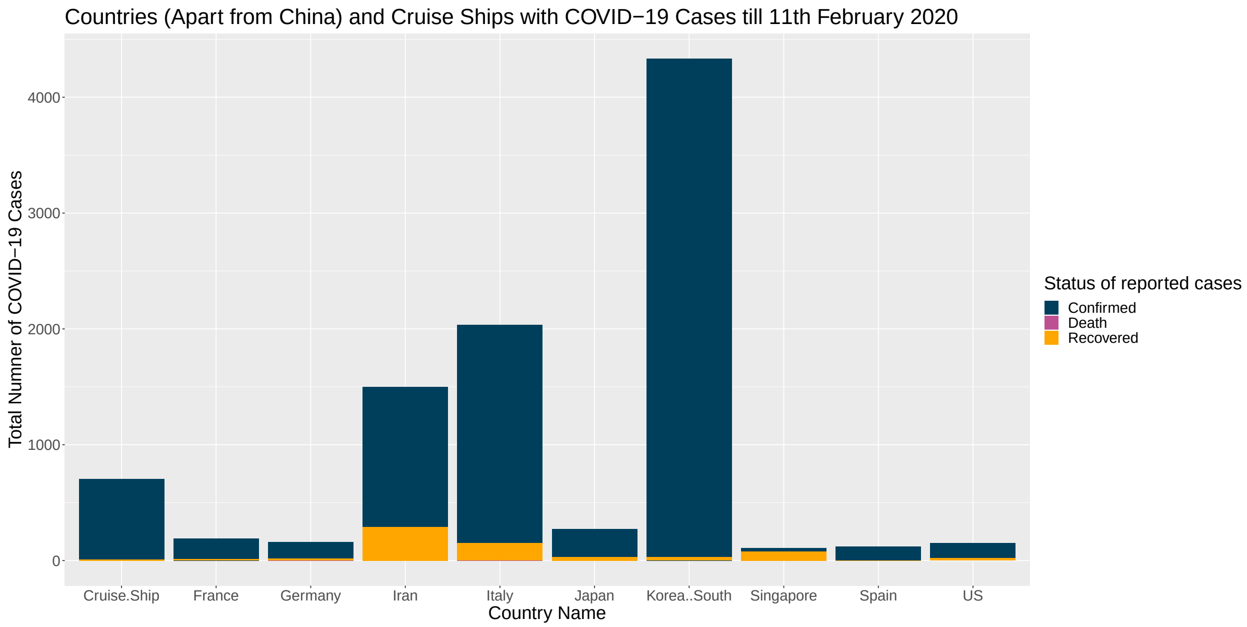


Countries Struggling To Contain COVID-19

- Timeline: 22nd January 2020 - 2nd March 2020

Apart from China, COVID-19 spread to other countries around the world and there was a surge in number of people infected in countries like Iran, Italy and South Korea.

```
counts_2nd <- transpose(corona_confirmed[41,2:ncol(corona_confirmed)])
deaths_2nd <- transpose(corona_deaths[41,2:ncol(corona_deaths)])
colnames(deaths_2nd) <- c("Death_count")
recovered_2nd <- transpose(corona_recovered[41,2:ncol(corona_recovered)])
colnames(recovered_2nd) <- c("Recovered_count")

df <-cbind(countries,counts_2nd,deaths_2nd,recovered_2nd)

df<-arrange(df, desc(V1))
ggplot(df[2:11,], aes(x=country_name))+
  geom_bar(aes(y=V1,fill='Confirmed'),stat = "identity")+
  geom_bar(aes(y=Death_count,fill='Death'),stat = "identity")+
  geom_bar(aes(y=Recovered_count,fill='Recovered'),stat = "identity")+
  scale_fill_manual(name="Status of reported cases",values = c('Confirmed' = '#003f5c', 'Death' = '#bc5
  xlab("Country Name") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Countries (Apart from China) and Cruise Ships with COVID-19 Cases till 11th February 2020")
  theme(text = element_text(size=21))
```



Trend of Confirmed, Death and Recovery Cases in China, South Korea, Italy and Singapore till 2nd March

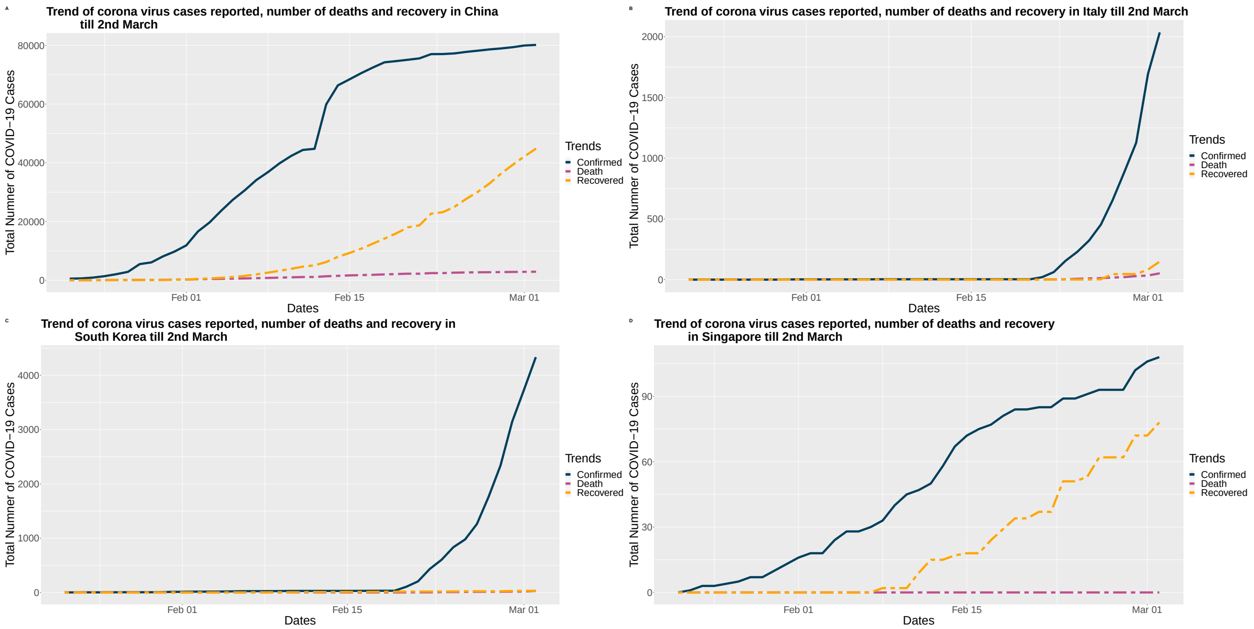
```
China <- ggplot(corona_confirmed[1:41,],aes(x = dates[1:41]))+
  geom_line(aes(y = corona_confirmed$China[1:41],colour='Confirmed'),size=3)+
  geom_line(aes(y = corona_deaths$China[1:41],colour='Death'),linetype="twodash",size=3)+
  geom_line(aes(y = corona_recovered$China[1:41],colour='Recovered'),linetype="twodash",size=3)+
  scale_color_manual(name = "Trends", values = c('Confirmed' = '#003f5c', 'Death' = '#bc5090','Recovere
  xlab("Dates") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus cases reported, number of deaths and recovery in China
          till 2nd March") + # Set title
 theme(text = element_text(size=35),plot.title = element_text(size = 35, face = "bold"),
    legend.title=element_text(size=35))
```

```
Italy<-ggplot(corona_confirmed[1:41,],aes(x = dates[1:41]))+
  geom_line(aes(y = corona_confirmed$Italy[1:41],colour='Confirmed'),size=3)+
  geom_line(aes(y = corona_deaths$Italy[1:41],colour='Death'),linetype="twodash",size=3)+
  geom_line(aes(y = corona_recovered$Italy[1:41],colour='Recovered'),linetype="twodash",size=3)+
  scale_color_manual(name = "Trends", values = c('Confirmed' = '#003f5c', 'Death' = '#bc5090','Recovere
  xlab("Dates") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus cases reported, number of deaths and recovery in Italy till 2nd March"
  theme(text = element_text(size=35),plot.title = element_text(size = 35, face = "bold"),
    legend.title=element_text(size=35))
```

```
South_Korea<-ggplot(corona_confirmed[1:41,],aes(x = dates[1:41]))+
  geom_line(aes(y = corona_confirmed$Korea..South[1:41],colour='Confirmed'),size=3)+
  geom_line(aes(y = corona_deaths$Korea..South[1:41],colour='Death'),linetype="twodash",size=3)+
  geom_line(aes(y = corona_recovered$Korea..South[1:41],colour='Recovered'),linetype="twodash",size=3)+
  scale_color_manual(name = "Trends", values = c('Confirmed' = '#003f5c',
                                                'Death' = '#bc5090','Recovered'='#ffa600'))+
  xlab("Dates") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus cases reported, number of deaths and recovery in
          South Korea till 2nd March") + # Set title
  theme(text = element_text(size=35),plot.title = element_text(size = 35, face = "bold"),
    legend.title=element_text(size=35))
```

```
Singapore<-ggplot(corona_confirmed[1:41,],aes(x = dates[1:41]))+
  geom_line(aes(y = corona_confirmed$Singapore[1:41],colour='Confirmed'),size=3)+
  geom_line(aes(y = corona_deaths$Singapore[1:41],colour='Death'),linetype="twodash",size=3)+
  geom_line(aes(y = corona_recovered$Singapore[1:41],colour='Recovered'),linetype="twodash",size=3)+
  scale_color_manual(name = "Trends", values = c('Confirmed' = '#003f5c', 'Death' = '#bc5090','Recovere
  xlab("Dates") + ylab("Total Numner of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus cases reported, number of deaths and recovery
          in Singapore till 2nd March") + # Set title
  theme(text = element_text(size=35),plot.title = element_text(size = 35, face = "bold"),
    legend.title=element_text(size=35))
```

```
figure <- ggarrange(China, Italy,South_Korea,Singapore,
                    labels = c("A", "B", "C","D"),
                    ncol = 2, nrow = 2)
figure
```



Trend of Rate of Spread in China, South Korea, Italy and Singapore till 2nd March

The rate of spread in China showed a declining trend with daily cases below the average number 1944 cases per day. Countries like Italy and South Korea struggled to limit the spread of virus with daily cases continously increasing upwards. Singapore started with an aggressive strategy to contain the virus and succeeded in keeping the rate of spread under control.

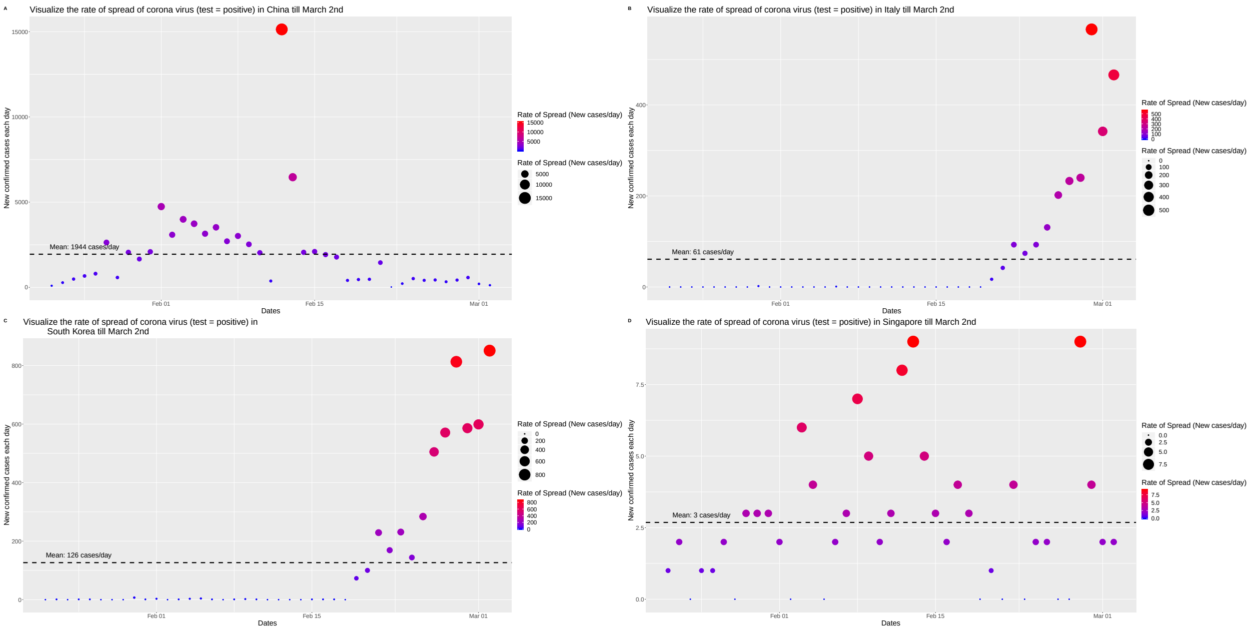
```
rate_of_spread_1 = 0

for (i in (1:41)){
  rate_of_spread_1[i] = corona_confirmed$China[i+1] - corona_confirmed$China[i]
}

mean_china = mean(rate_of_spread_1)

chn<-ggplot(NULL, aes(x=dates[1:41],y=rate_of_spread_1))+
  geom_point(aes(size = rate_of_spread_1,color=rate_of_spread_1))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in China till March 2nd")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean_china,color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-01-25"),y=mean_china,label ="Mean: 1944 cases/day", vjust = -1),size=7)+
  theme(text = element_text(size=21))
```

```
rate_of_spread_2 = 0

for (i in (1:41)){
  rate_of_spread_2[i] = corona_confirmed$Italy[i+1] - corona_confirmed$Italy[i]
}

mean_italy = mean(rate_of_spread_2)

Italy_rate<-ggplot(NULL, aes(x=dates[1:41],y=rate_of_spread_2))+
  geom_point(aes(size = rate_of_spread_2,color=rate_of_spread_2))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in Italy till March 2nd")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean_italy,color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-01-25"),y=mean_italy,label ="Mean: 61 cases/day", vjust = -1),size=7)+
  theme(text = element_text(size=21))
```

```
rate_of_spread_3 = 0

for (i in (1:41)){
  rate_of_spread_3[i] = corona_confirmed$Korea..South[i+1] - corona_confirmed$Korea..South[i]
}

mean_korea = mean(rate_of_spread_3)

Korea_rate<-ggplot(NULL, aes(x=dates[1:41],y=rate_of_spread_3))+
  geom_point(aes(size = rate_of_spread_3,color=rate_of_spread_3))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in
          South Korea till March 2nd")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean_korea,color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-01-25"),y=mean_korea,label ="Mean: 126 cases/day", vjust = -1),size=7)+
  theme(text = element_text(size=21))
```

```
rate_of_spread_4 = 0

for (i in (1:41)){
  rate_of_spread_4[i] = corona_confirmed$Singapore[i+1] - corona_confirmed$Singapore[i]
}

mean_singapore = mean(rate_of_spread_4)

Singapore_rate<-ggplot(NULL, aes(x=dates[1:41],y=rate_of_spread_4))+
  geom_point(aes(size = rate_of_spread_4,color=rate_of_spread_4))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in Singapore till March 2nd")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean_singapore,color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-01-25"),y=mean_singapore,label ="Mean: 3 cases/day", vjust = -1),size=7
  theme(text = element_text(size=21))
```

```
figure1 <- ggarrange(chn, Italy_rate,Korea_rate,Singapore_rate,
                    labels = c("A", "B", "C","D"),
                    ncol = 2, nrow = 2)
figure1
```



COVID-19 the Global Pandemic

- Total Countries Affected : 124
- Total Deaths : 145983
- Total Recovered : 70286
- Total Deaths : 5408

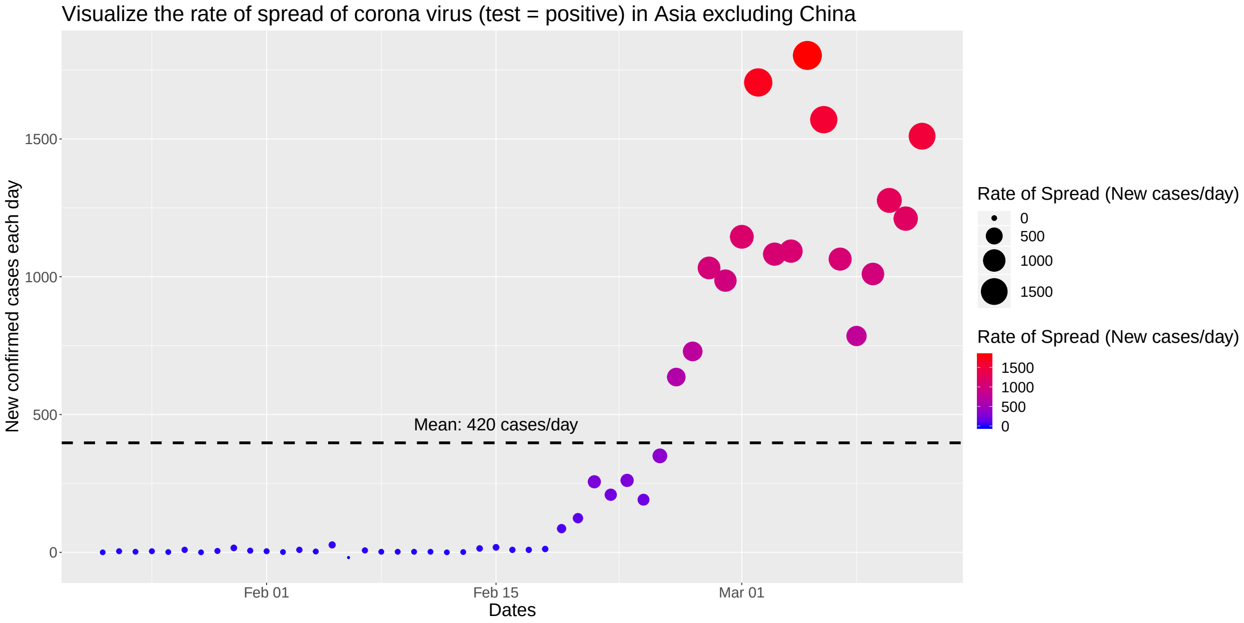
Asia vs COVID-19

```
# Aggregate the counts of confirmed cases for countries in Asia
corona_confirmed$Asia <- corona_confirmed$Korea..South+corona_confirmed$Iran+
corona_confirmed$Japan+corona_confirmed$India+corona_confirmed$Thailand+corona_confirmed$Indonesia
+corona_confirmed$Vietnam+corona_confirmed$Singapore+corona_confirmed$Philippines+
corona_confirmed$Malaysia+corona_confirmed$Pakistan+corona_confirmed$Israel+
corona_confirmed$Saudi.Arabia+corona_confirmed$Taiwan.+corona_confirmed$Iraq+
corona_confirmed$Bangladesh+corona_confirmed$Cambodia+corona_confirmed$Sri.Lanka+
corona_confirmed$Maldives+corona_confirmed$Nepal+corona_confirmed$Afghanistan+
corona_confirmed$Kuwait+corona_confirmed$Oman+corona_confirmed$Lebanon
```

```
##  [1]    1    4    8   12   15   19   24   27   33   37   44   47   49   58
## [15]   64   71   74   85   93  101  104  107  117  126  134  139  143  147
## [29]  151  152  157  157  163  170  183  211  232  237  268  292  317  342
## [43]  365  389  467  511  599  631  736  857  917 1141
```

Visualize the Rate of Spread of virus in Asia

```
rate_of_spread_asia = 0

for (i in (1:51)){
  rate_of_spread_asia[i] = corona_confirmed$Asia[i+1] - corona_confirmed$Asia[i]
}

mean_asia = mean(rate_of_spread_asia)

ggplot(NULL, aes(x=dates[1:51],y=rate_of_spread_asia))+
  geom_point(aes(size = rate_of_spread_asia,color=rate_of_spread_asia))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in Asia excluding China")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean_asia,color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-02-15"),y=mean_asia,label ="Mean: 420 cases/day", vjust = -1),size=7)+
  theme(text = element_text(size=21))
```

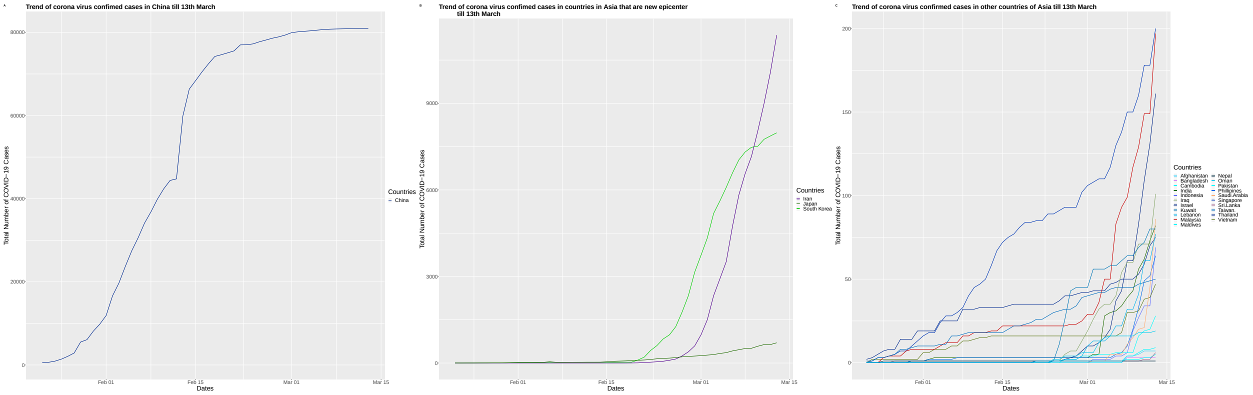


Visualize the Comparative Trend of Confirmed Cases in China and Rest of Asia

```
asia_high <- ggplot(corona_confirmed[1:52,],aes(x = dates[1:52]))+
  geom_line(aes(y = corona_confirmed$Korea..South[1:52],colour='South Korea'),size=1)+
  geom_line(aes(y = corona_confirmed$Iran[1:52],colour='Iran'),size=1)+
  geom_line(aes(y = corona_confirmed$Japan[1:52],colour='Japan'),size=1)+
  scale_color_manual(name = "Countries", values = c('South Korea'='#00ce00',
                                                    'Iran'='#51008f',
                                                    'Japan'='#1f6c00'))+
  xlab("Dates") + ylab("Total Number of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus confimed cases in countries in Asia that are new epicenter
        till 13th March") + # Set title
  theme(text = element_text(size=30),plot.title = element_text(size = 30, face = "bold"),
  legend.title=element_text(size=30))
```

```
asia_low <- ggplot(corona_confirmed[1:52,],aes(x = dates[1:52]))+
  geom_line(aes(y = corona_confirmed$India[1:52],colour='India'),size=1)+
  geom_line(aes(y = corona_confirmed$Thailand[1:52],colour='Thailand'),size=1)+
  geom_line(aes(y = corona_confirmed$Indonesia[1:52],colour='Indonesia'),size=1)+
  geom_line(aes(y = corona_confirmed$Vietnam[1:52],colour='Vietnam'),size=1)+
  geom_line(aes(y = corona_confirmed$Singapore[1:52],colour='Singapore'),size=1)+
  geom_line(aes(y = corona_confirmed$Philippines[1:52],colour='Phillipines'),size=1)+
  geom_line(aes(y = corona_confirmed$Malaysia[1:52],colour='Malaysia'),size=1)+
  geom_line(aes(y = corona_confirmed$Pakistan[1:52],colour='Pakistan'),size=1)+
  geom_line(aes(y = corona_confirmed$Israel[1:52],colour='Israel'),size=1)+
  geom_line(aes(y = corona_confirmed$Saudi.Arabia[1:52],colour='Saudi.Arabia'),size=1)+
  geom_line(aes(y = corona_confirmed$Taiwan.[1:52],colour='Taiwan.'),size=1)+
  geom_line(aes(y = corona_confirmed$Iraq[1:52],colour='Iraq'),size=1)+
  geom_line(aes(y = corona_confirmed$Bangladesh[1:52],colour='Bangladesh'),size=1)+
  geom_line(aes(y = corona_confirmed$Cambodia[1:52],colour='Cambodia'),size=1)+
  geom_line(aes(y = corona_confirmed$Sri.Lanka[1:52],colour='Sri.Lanka'),size=1)+
  geom_line(aes(y = corona_confirmed$Maldives[1:52],colour='Maldives'),size=1)+
  geom_line(aes(y = corona_confirmed$Nepal[1:52],colour='Nepal'),size=1)+
  geom_line(aes(y = corona_confirmed$Afghanistan[1:52],colour='Afghanistan'),size=1)+
  geom_line(aes(y = corona_confirmed$Kuwait[1:52],colour='Kuwait'),size=1)+
  geom_line(aes(y = corona_confirmed$Oman[1:52],colour='Oman'),size=1)+
  geom_line(aes(y = corona_confirmed$Lebanon[1:52],colour='Lebanon'),size=1)+
  scale_color_manual(name = "Countries", values = c('Thailand' = '#002a90',
  'Iran'='#00ce00','Japan'='#51008f','India'='#1f6c00','Indonesia'='#6471ff','Vietnam'='#607b1a',    'S
  'Oman'='#00caef','Lebanon'='#00b3ed'))+
  xlab("Dates") + ylab("Total Number of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus confirmed cases in other countries of Asia till 13th March") +
  #Set title
  theme(text = element_text(size=30),plot.title = element_text(size = 30, face = "bold"),
  legend.title=element_text(size=30))
```

```
figure2 <- ggarrange(china_stat,asia_high, asia_low,
                    labels = c("A", "B", "C"),
                    ncol = 3, nrow = 1)
figure2
```



Europe vs COVID-19

```
# Aggregate the counts of confirmed cases for countries in Europe
corona_confirmed$Europe <- corona_confirmed$Italy+corona_confirmed$France+corona_confirmed$UK+corona_co
+corona_confirmed$Poland+corona_confirmed$Greece+corona_confirmed$Austria+corona_confirmed$Belgium
```

```
##  [1]    0    0    0    0    0    0    0    0    0    0    0    0    0    1
## [15]    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## [29]    1    1    1    1    1    1    3    4    7    8   14   23   33   41
## [43]   62  123  214  299  388  459  560  690  764 1321
```

```
+corona_confirmed$Sweden+corona_confirmed$Ukraine+corona_confirmed$Iceland+corona_confirmed$Norway
```

```
##  [1]    0    0    0    0    0    0    0    0    0    1    1    1    1    1
## [15]    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## [29]    1    1    1    1    1    1    1    3    8   14   28   36   46   65
## [43]  118  216  253  359  430  512  825 1184 1405 1947
```

```
+corona_confirmed$Ireland+corona_confirmed$Czechia+corona_confirmed$Denmark
```

```
##  [1]    0    0    0    0    0    0    0    0    0    0    0    0    0    0
## [15]    0    0    0    0    0    0    0    0    0    0    0    0    0    0
## [29]    0    0    0    0    0    0    0    0    1    1    4    8    8   13
## [43]   25   29   60   61   87  144  339  578  754 1035
```

```
+corona_confirmed$Romania+corona_confirmed$Hungary+corona_confirmed$Croatia
```

```
##  [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [18]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [35]   1   4   4   8   9  10  10  12  16  18  22  25  34  36  48  77  81
## [52] 140
```

```
+corona_confirmed$Finland+corona_confirmed$Malta+corona_confirmed$Bulgaria
```

```
##  [1]   0   0   0   0   0   0   0   1   1   1   1   1   1   1   1   1   1
## [18]   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## [35]   1   2   2   2   3   6   6   6   6  12  15  18  30  37  49  72  72
## [52] 190
```

```
+corona_confirmed$Albania+corona_confirmed$Luxembourg+corona_confirmed$Serbia
```

```
##  [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [18]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [35]   0   0   0   0   1   1   1   1   1   1   3   3   4   6  20  31  61
## [52] 102
```

```
+corona_confirmed$Cyprus+corona_confirmed$Slovakia+corona_confirmed$Slovenia
```

```
##  [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [18]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [35]   0   0   0   0   0   0   0   0   0   2   8   8  19  21  41  73 111
## [52] 187
```

```
+corona_confirmed$Monaco+corona_confirmed$Belarus+corona_confirmed$Estonia
```

```
##  [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [18]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## [35]   0   0   1   2   3   3   3   4   9  10  17  17  17  17  22  26  30
## [52] 108
```

```
+corona_confirmed$Lithuania+corona_confirmed$Latvia+corona_confirmed$Moldova
```

```
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [24]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  2  2  2  2  2  2
## [47]  4  8 12 16 16 29
```

```
+corona_confirmed$North.Macedonia+corona_confirmed$Bosnia.and.Herzegovina+corona_confirmed$Andorra
```

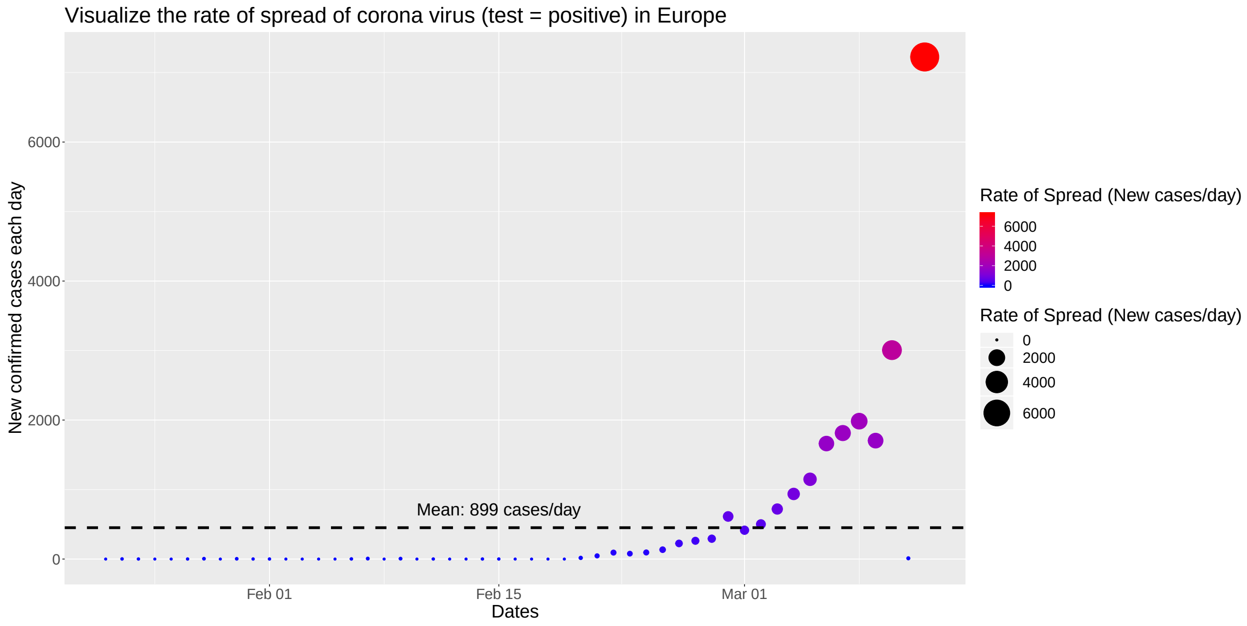
```
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [24]  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  2  2  2  4  6  7
## [47]  7  7 13 15 19 28
```

```
+corona_confirmed$San.Marino
```

```
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [24]  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  8 10 16 21 21 23
## [47] 36 36 51 62 69 80
```

Visualize the Rate of Spread of virus in Europe

```
rate_of_spread_europe = 0

for (i in (1:51)){
  rate_of_spread_europe[i] = corona_confirmed$Europe[i+1] - corona_confirmed$Europe[i]
}

mean_europe = mean(rate_of_spread_europe)

ggplot(NULL, aes(x=dates[1:51],y=rate_of_spread_europe))+
  geom_point(aes(size = rate_of_spread_europe,color=rate_of_spread_europe))+
  ggtitle("Visualize the rate of spread of corona virus (test = positive) in Europe")+
  labs(y="New confirmed cases each day", x = "Dates")+
  scale_size(range = c(1, 15),name="Rate of Spread (New cases/day)")+
  scale_color_gradient(low="blue", high="red",name="Rate of Spread (New cases/day)")+
  geom_hline(yintercept = mean_europe,color="black",size=1.5,linetype="dashed")+
  geom_text(aes(x=as.Date("2020-02-15"),y=mean_europe,label ="Mean: 899 cases/day", vjust = -1),size=7)
  theme(text = element_text(size=21))
```

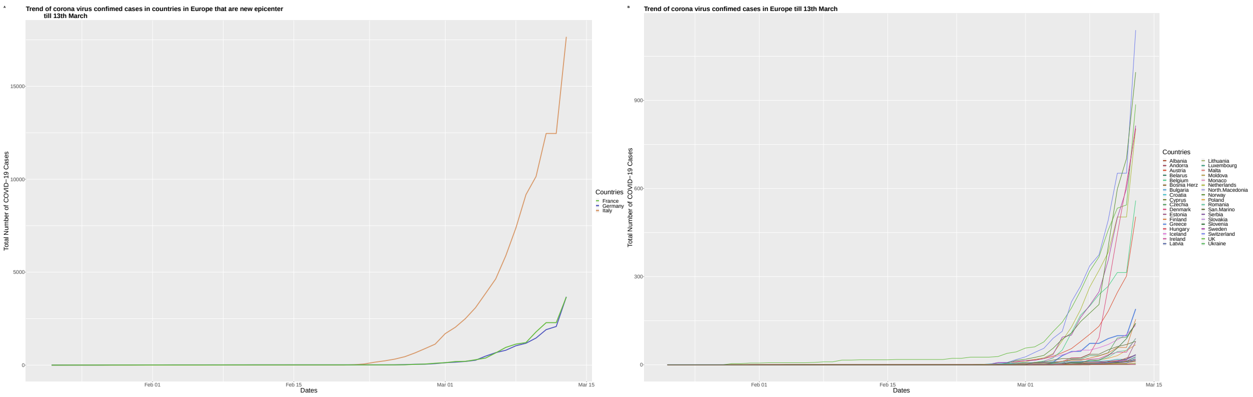


Visualize the Comparative Trend of Confirmed Cases in New Epicenters of Europe and Rest of Europe

```r
euro_high<-ggplot(corona_confirmed[1:52,],aes(x = dates[1:52]))+
  geom_line(aes(y = corona_confirmed$Italy[1:52],colour='Italy'),size=2)+
  geom_line(aes(y = corona_confirmed$Germany[1:52],colour='Germany'),size=2)+
  geom_line(aes(y = corona_confirmed$France[1:52],colour='France'),size=2)+
  scale_color_manual(name = "Countries", values = c('Italy' = '#d89868',
                                                'Germany'='#4e54be',
                                                'France'='#65bc3f'))+
  xlab("Dates") + ylab("Total Number of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus confimed cases in countries in Europe that are new epicenter
          till 13th March") + # Set title
  theme(text = element_text(size=30),plot.title = element_text(size = 30, face = "bold"),
  legend.title=element_text(size=30))
```

```r
euro_low<-ggplot(corona_confirmed[1:52,],aes(x = dates[1:52]))+
  geom_line(aes(y = corona_confirmed$UK[1:52],colour='UK'),size=1)+
  geom_line(aes(y = corona_confirmed$Netherlands[1:52],colour='Netherlands'),size=1)+
  geom_line(aes(y = corona_confirmed$Switzerland[1:52],colour='Switzerland'),size=1)+
  geom_line(aes(y = corona_confirmed$Poland[1:52],colour='Poland'),size=1)+
  geom_line(aes(y = corona_confirmed$Greece[1:52],colour='Greece'),size=2)+
  geom_line(aes(y = corona_confirmed$Austria[1:52],colour='Austria'),size=1)+
  geom_line(aes(y = corona_confirmed$Belgium[1:52],colour='Belgium'),size=1)+
  geom_line(aes(y = corona_confirmed$Sweden[1:52],colour='Sweden'),size=1)+
  geom_line(aes(y = corona_confirmed$Ukraine[1:52],colour='Ukraine'),size=1)+
  geom_line(aes(y = corona_confirmed$Iceland[1:52],colour='Iceland'),size=1)+
  geom_line(aes(y = corona_confirmed$Norway[1:52],colour='Norway'),size=1)+
  geom_line(aes(y = corona_confirmed$Ireland[1:52],colour='Ireland'),size=1)+
  geom_line(aes(y = corona_confirmed$Czechia[1:52],colour='Czechia'),size=1)+
  geom_line(aes(y = corona_confirmed$Denmark[1:52],colour='Denmark'),size=1)+
  geom_line(aes(y = corona_confirmed$Romania[1:52],colour='Romania'),size=1)+
  geom_line(aes(y = corona_confirmed$Hungary[1:52],colour='Hungary'),size=1)+
  geom_line(aes(y = corona_confirmed$Croatia[1:52],colour='Croatia'),size=1)+
  geom_line(aes(y = corona_confirmed$Finland[1:52],colour='Finland'),size=1)+
  geom_line(aes(y = corona_confirmed$Malta[1:52],colour='Malta'),size=1)+
  geom_line(aes(y = corona_confirmed$Bulgaria[1:52],colour='Bulgaria'),size=1)+
  geom_line(aes(y = corona_confirmed$Albania[1:52],colour='Albania'),size=1)+
  geom_line(aes(y = corona_confirmed$Luxembourg[1:52],colour='Luxembourg'),size=1)+
  geom_line(aes(y = corona_confirmed$Serbia[1:52],colour='Serbia'),size=1)+
  geom_line(aes(y = corona_confirmed$Cyprus[1:52],colour='Cyprus'),size=1)+
  geom_line(aes(y = corona_confirmed$Slovakia[1:52],colour='Slovakia'),size=1)+
  geom_line(aes(y = corona_confirmed$Slovenia[1:52],colour='Slovenia'),size=1)+
  geom_line(aes(y = corona_confirmed$Monaco[1:52],colour='Monaco'),size=1)+
  geom_line(aes(y = corona_confirmed$Belarus[1:52],colour='Belarus'),size=1)+
  geom_line(aes(y = corona_confirmed$Estonia[1:52],colour='Estonia'),size=1)+
  geom_line(aes(y = corona_confirmed$Lithuania[1:52],colour='Lithuania'),size=1)+
  geom_line(aes(y = corona_confirmed$Latvia[1:52],colour='Latvia'),size=1)+
  geom_line(aes(y = corona_confirmed$Moldova[1:52],colour='Moldova'),size=1)+
  geom_line(aes(y = corona_confirmed$North.Macedonia[1:52],colour='North.Macedonia'),size=1)+
  geom_line(aes(y = corona_confirmed$Bosnia.and.Herzegovina[1:52],colour='Bosnia Herz'),size=1)+
  geom_line(aes(y = corona_confirmed$Andorra[1:52],colour='Andorra'),size=1)+
  geom_line(aes(y = corona_confirmed$San.Marino[1:52],colour='San.Marino'),size=1)+
  scale_color_manual(name = "Countries", values = c('Italy' = '#d89868','Germany'='#4e54be',       'UK'=
  xlab("Dates") + ylab("Total Number of COVID-19 Cases") +# Set axis labels
  ggtitle("Trend of corona virus confimed cases in Europe till 13th March") + # Set title
  theme(text = element_text(size=30),plot.title = element_text(size = 30, face = "bold"),
  legend.title=element_text(size=30))
```

```
figure3 <- ggarrange(euro_high,euro_low,
                    labels = c("A", "B"),
                    ncol = 2, nrow = 1)
figure3
```



Forecasting Using LSTM Recurrent Neural Network

Extract Time independent Feature from Chronological Sequence

```
# Calculate the difference between consecutive readings. This gives more predictive power to LSTM
time_diff = diff(corona_confirmed$China,differences = 1)
head(time_diff)
```

```
## [1]   95  277  486  669  802 2632
```

Method to Normalize and De-Normalize Data

```
# As for any neural network, it is recommended to normalize the data for better results
normData <- function(inputData,minVal,maxVal) {
  normD <- (inputData - minVal)/(maxVal - minVal)
  return(normD)
}
denormData <- function(inputData,minVal,maxVal) { # This method is used to revert the predictions
  denormVal = (inputData*(maxVal - minVal)+minVal)
  return(denormVal)
}
```

Normalizing Data

```r
min_val = min(time_diff)
max_val = max(time_diff)
time_diff <- normData(time_diff,min_val,max_val)
```

Transform data into supervised learning format (time - k, time)

```r
# Transform data into (x,y) format where x is input and y is the label
supervised_sequence <- function(seq, steps){
  target_k = NULL
  target = NULL

  for (i in sequence(length(seq))){
    tail = i+steps
    if( tail <= length(seq)){
      target_k<-rbind(target_k,c(seq[i:(tail-1)]))
      target<-rbind(target,c(seq[tail]))
    }

    if(tail > length(seq)){
      break
    }
  }
  return (as.data.frame(cbind(target_k,target)))
}

step = 3 #LSTM will look at t-3 days of counts to make prediction for t day
sup_df = supervised_sequence(time_diff,step)
```

Split Data into training and testing set

```r
total_rows = nrow(sup_df)
div = 0.7 * total_rows # 70% Training Data and 30% testing Data
div = round(div,digits = 0)
train = sup_df[1:div,]
test = sup_df[(div+1):total_rows,]
```

Reshape training and testing data into input format for LSTM (samples,timesteps,features)

```r
x_train <- array(unlist(train[,1:step]),dim = c(dim(train[,1:step])[1],length(train[,1:step]),1))
y_train <- train[,(step+1)]

x_test <- array(unlist(test[,1:step]),dim = c(dim(test[,1:step])[1],length(test[,1:step]),1))
y_test <- test[,(step+1)]
```

Prepare LSTM Model

```r
model <- keras_model_sequential() #Using sequential model
model%>% # Using stateful=TRUE to point the model that order is important for data
  layer_lstm(1, batch_input_shape = c(1,step, 1), stateful= TRUE,activation = "relu")%>%
  layer_dense(units=1) # Fully connected Dense Layer

model %>% compile(
  loss = 'mean_squared_error', # loss function
  optimizer = optimizer_adam( lr= 0.001, decay = 1e-6 ), # learning rate and decay can be tuned
  metrics = c('accuracy') # accuracy to assess model performance
)
summary(model) # See the summary of LSTM Model designed above
```

```
## Model: "sequential"
## ___________________________________________________________________________
## Layer (type)                     Output Shape                  Param #
## ===========================================================================
## lstm (LSTM)                      (1, 1)                        12
## ___________________________________________________________________________
## dense (Dense)                    (1, 1)                        2
## ===========================================================================
## Total params: 14
## Trainable params: 14
## Non-trainable params: 0
## ___________________________________________________________________________
```

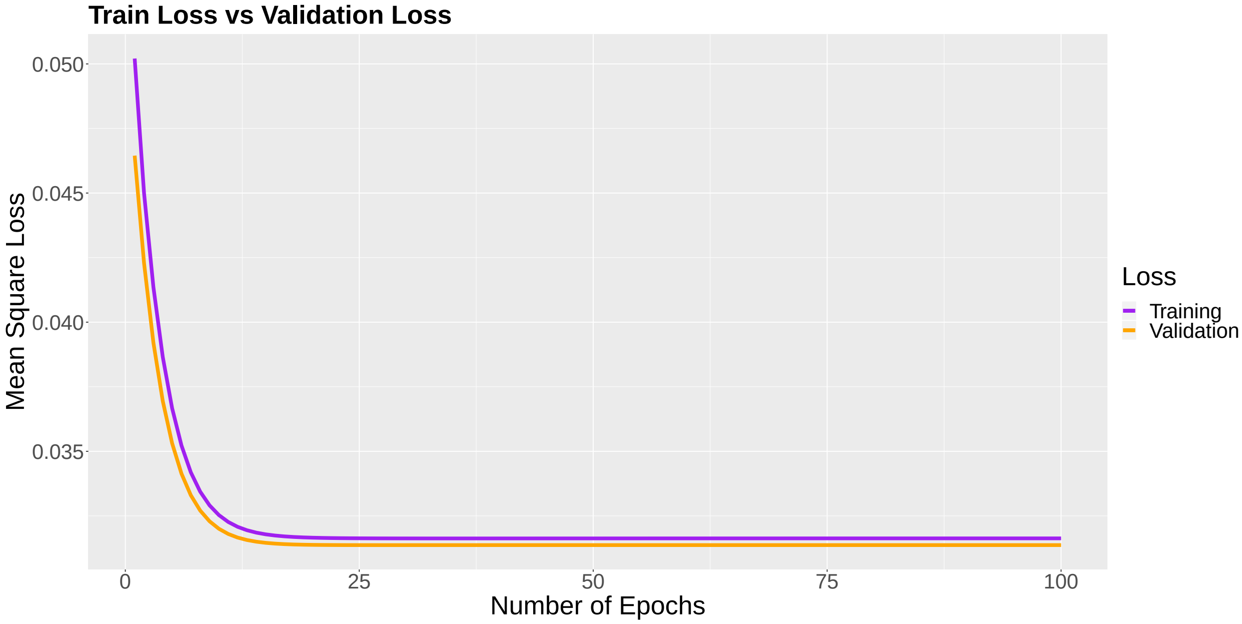
Fit Model on Training Data

```r
Epochs = 100 # Can be fine tuned to run more iterations. Process intensive
training_loss = numeric(Epochs)
validation_loss = numeric(Epochs)
for(i in 1:Epochs ){# shuffle is false to maintain the order of data
  fitted_mod<- model %>% fit(x_train, y_train, epochs=1, batch_size=1, verbose=0,
                             shuffle=FALSE,validation_data=list(x_train,y_train))
  training_loss[i]=fitted_mod$metrics$loss #Calculate training loss
  validation_loss[i]=fitted_mod$metrics$val_loss #Calculate Validation loss
  model %>% reset_states() # LSTM requires resetting the states after each epoch
}
```

Model Performance

Plot Training vs Validation loss for the Model

```r
ggplot(NULL, aes(x=sequence(Epochs)))+
  geom_line(aes(y=validation_loss, group=1,colour='Validation'),size=2)+
  geom_line(aes(y=training_loss, group=1,colour='Training'),size=2)+
  scale_color_manual(name = "Loss", values = c('Validation' = 'orange','Training' = 'purple'))+
  ggtitle("Train Loss vs Validation Loss")+
  xlab("Number of Epochs") + ylab("Mean Square Loss")+
  theme(text = element_text(size=30),plot.title = element_text(size = 30, face = "bold"),
    legend.title=element_text(size=30))
```



Make prediction on testing data

```
predictions = numeric(length(y_test))
raw_preds = numeric(length(y_test))
for(i in 1:length(y_test)){
     X = as.double(x_test[i,1:3,1])
     dim(X) = c(1,3,1)
     yhat = model %>% predict(X, batch_size=1) #Make predictions on test data
     raw_preds[i]<-yhat
     yhat = denormData(yhat,min_val,max_val) #De-normalize the predictions
     yhat  = yhat + corona_confirmed$China[(div+i+step)] #Change country Name
     predictions[i] <- yhat
}
```

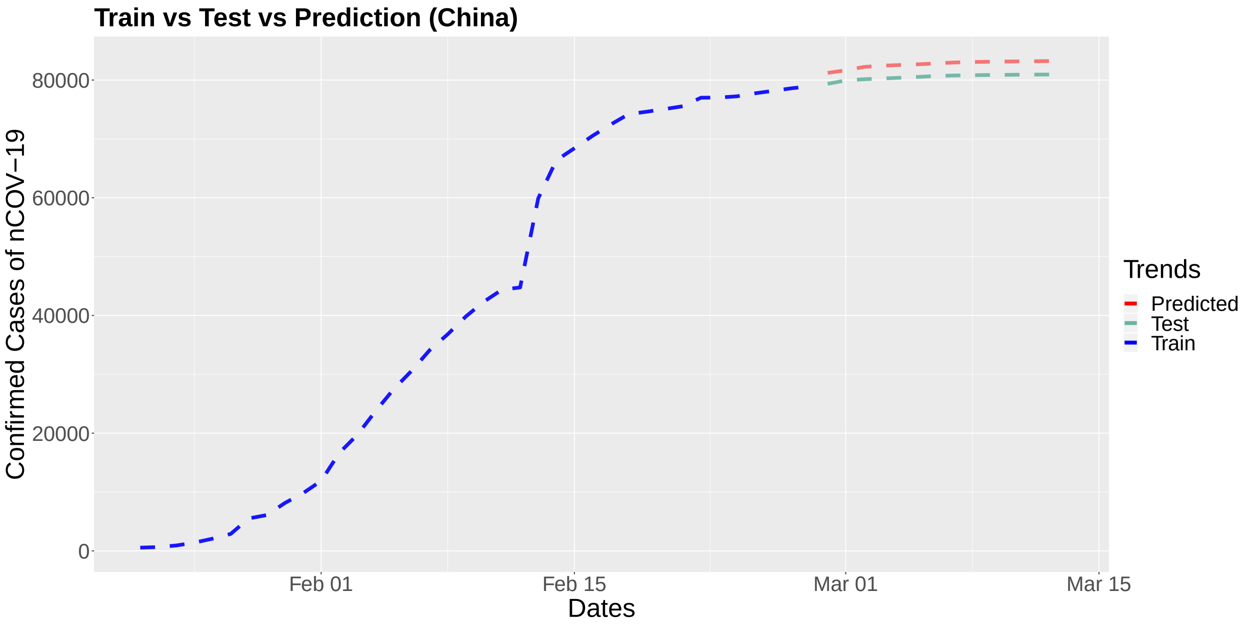
Model Performance

```
test_RMSE = sqrt(sum((raw_preds - y_test)^2)/length(y_test))
test_RMSE
```

```
## [1] 0.1424275
```

Plot Training vs Testing vs Predicted

```
x <- dates[39:52]
# Change country Name for the plot
ggplot(NULL, aes(x=dates)) +
  geom_line( aes(x=dates[1:38],y=corona_confirmed$China[1:38],color='Train')
             , size=2, alpha=0.9, linetype=2)+
  geom_line( aes(x=dates[39:52],y=corona_confirmed$China[39:52],color='Test')
             , size=2, alpha=0.9, linetype=2) +
  geom_line(aes(x=dates[39:52],y=predictions,color='Predicted')
            ,size=2,alpha=0.5,linetype=2)+
  scale_color_manual(name = "Trends", values = c('Train'='blue',
                                                 'Test' = '#69b3a2','Predicted'='red'))+
  xlab("Dates") + ylab("Confirmed Cases of nCOV-19")+
  ggtitle("Train vs Test vs Prediction (China)")+
  theme(text = element_text(size=30),plot.title = element_text(size = 30, face = "bold"),
    legend.title=element_text(size=30))
```


Train vs Test vs Prediction (China)
Confirmed Cases of nCOV−19
Feb 01
Feb 15
Mar 01
Mar 15
Dates
Trends
Predicted
Test
Train