Corona Virus Analysis and Forecast using R

Eden Kim

3/4/2020

1. Load packages

```
library(dplyr)
library(ggplot2)
library(maps)
library(lubridate)
```

2. Read the data

```
covid <- read.csv("covid_19_clean_complete.csv")
dim(covid)
```

```
## [1] 5822    8
```

```
head(covid)
```

```
##   Province.State Country.Region     Lat     Long    Date Confirmed Deaths
## 1          Anhui Mainland China 31.8257 117.2264 1/22/20         1      0
## 2        Beijing Mainland China 40.1824 116.4142 1/22/20        14      0
## 3      Chongqing Mainland China 30.0572 107.8740 1/22/20         6      0
## 4         Fujian Mainland China 26.0789 117.9874 1/22/20         1      0
## 5          Gansu Mainland China 36.0611 103.8343 1/22/20         0      0
## 6      Guangdong Mainland China 23.3417 113.4244 1/22/20        26      0
##   Recovered
## 1         0
## 2         0
## 3         0
## 4         0
## 5         0
## 6         0
```

- The data set is divided into the provinces and states for China, US. The most recent entry gives the latest case numbers. I would like to know how many cases are confirmed positive per country. First, I will extract data from Country.Region, Province.State and Confirmed. Then get the latest value for the province/state and then sum up the cases from the province/state to get the total confirmed cases for the country.

3. Total number of Confirmed cases per contry

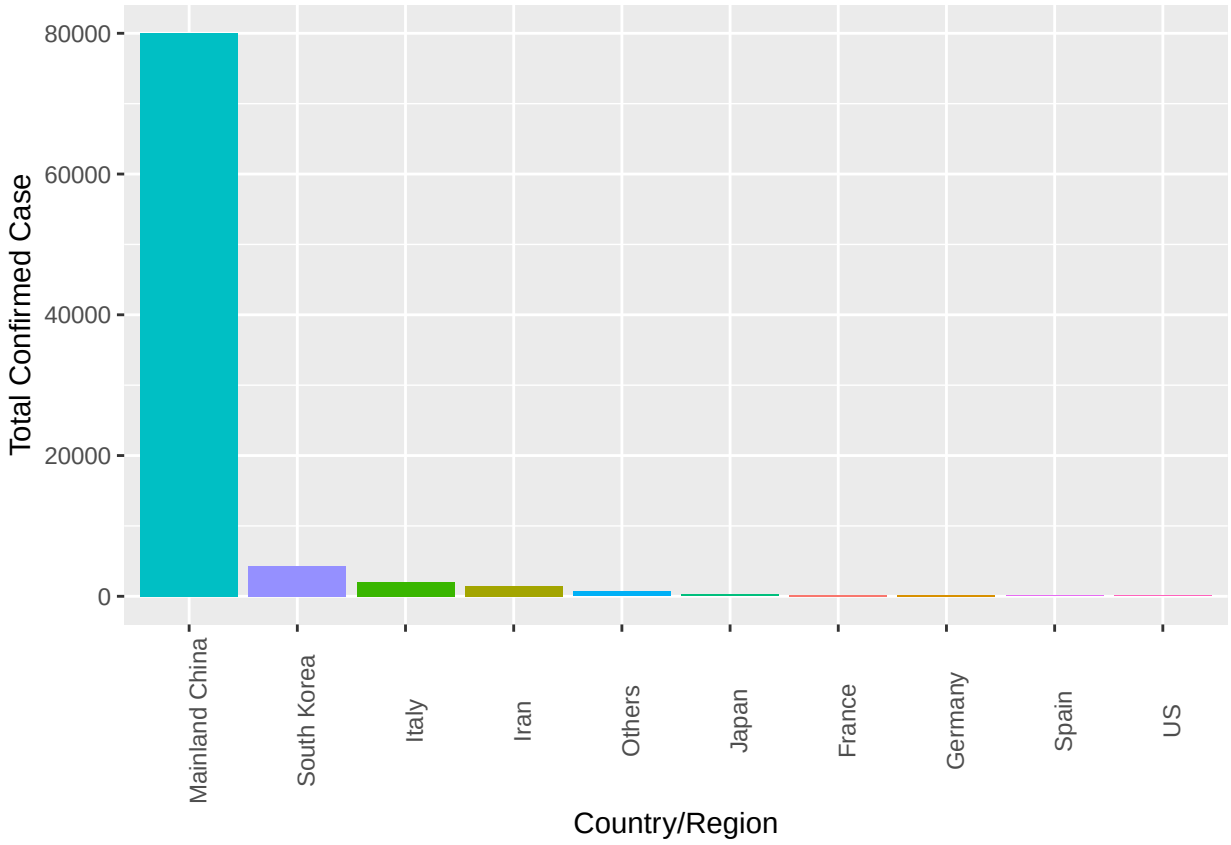
```
# get total number of cases per country
confirmed_per_country <- covid %>%
  select(Country.Region, Province.State, Confirmed) %>%
  group_by(Country.Region, Province.State) %>%
  summarise(Total.Confirmed = max(Confirmed)) %>%
  group_by(Country.Region) %>%
  summarise(Total.Confirmed = sum(Total.Confirmed)) %>%
  arrange(desc(Total.Confirmed))
confirmed_per_country
```

```
## # A tibble: 75 x 2
##    Country.Region Total.Confirmed
##    <fct>                    <dbl>
```

```
##  1 Mainland China           80026
##  2 South Korea               4335
##  3 Italy                     2036
##  4 Iran                      1501
##  5 Others                     705
##  6 Japan                      274
##  7 France                     191
##  8 Germany                    159
##  9 Spain                      120
## 10 US                         119
## # ... with 65 more rows
```

```
# bar graph comparing the top 10 countries
confirmed_per_country %>%
  top_n(10) %>%    # select top 10
  ggplot(aes(x = reorder(Country.Region,-Total.Confirmed), y = Total.Confirmed, fill = Country.Region))
  geom_bar(stat = "identity") +
  labs(y = "Total Confirmed Case", x = "Country/Region") +
  theme(legend.position="none", axis.text.x = element_text(angle=90))
```

```
## Selecting by Total.Confirmed
```



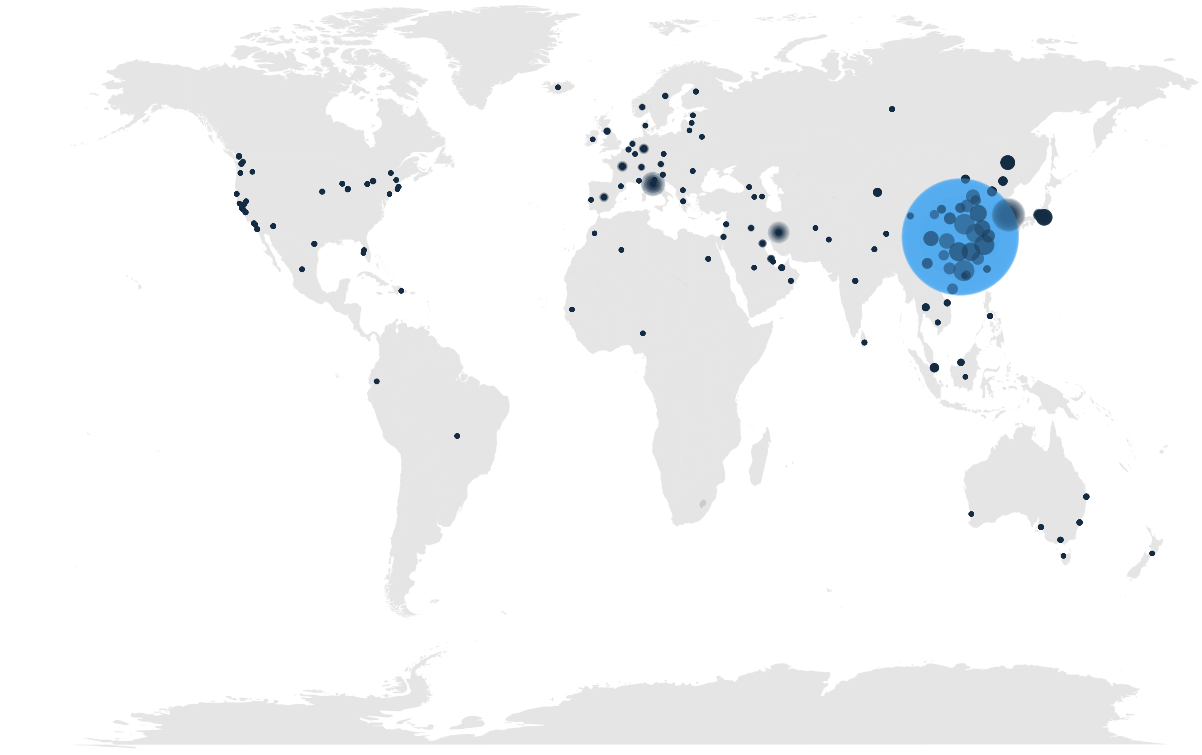

```
# world map of the spread
countries <- covid %>%
  select(Country.Region, Lat, Long, Confirmed) %>%
  group_by(Country.Region)
countries
```

```
## # A tibble: 5,822 x 4
```

```
## # Groups:   Country.Region [75]
##    Country.Region   Lat  Long Confirmed
##    <fct>          <dbl> <dbl>     <int>
##  1 Mainland China  31.8  117.         1
##  2 Mainland China  40.2  116.        14
##  3 Mainland China  30.1  108.         6
##  4 Mainland China  26.1  118.         1
##  5 Mainland China  36.1  104.         0
##  6 Mainland China  23.3  113.        26
##  7 Mainland China  23.8  109.         2
##  8 Mainland China  26.8  107.         1
##  9 Mainland China  19.2  110.         4
## 10 Mainland China  38.0  115.         1
## # ... with 5,812 more rows
```

```
worldmap = map_data("world")

ggplot() +
  geom_polygon(data = worldmap, aes(x = long, y = lat, group = group), fill="grey", alpha=0.4) +
  geom_point(data = countries, aes(x = Long, y = Lat, color = Confirmed, size=Confirmed), alpha=0.2) +
  scale_size_continuous(range=c(.2,20)) +
  ggtitle("Corona Virus Spread Map") +
  theme_void() +
  theme(legend.position="none")
```



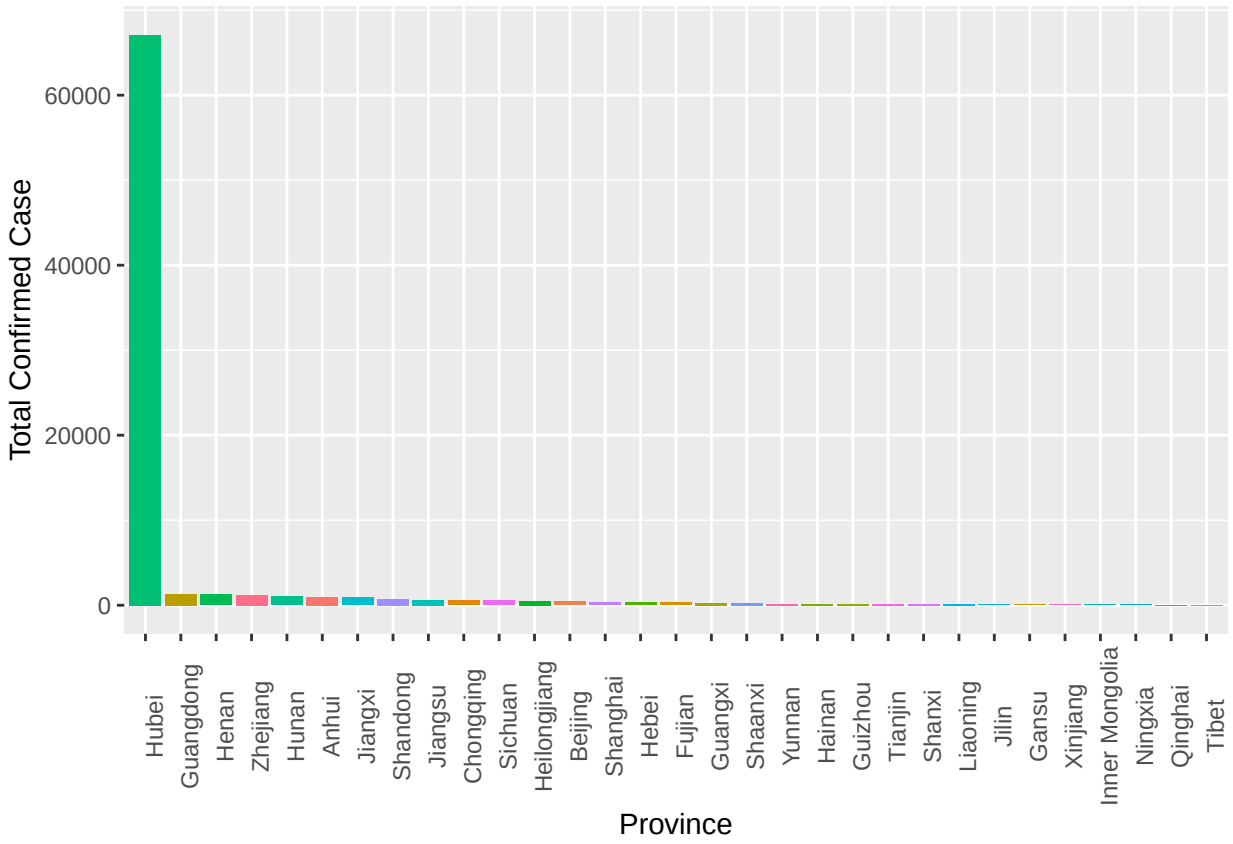
4. The spread throuout the Provinces in China

```
# get confirmed cases per procince
per_ch_province <- covid %>%
```

```
  filter(Country.Region == "Mainland China") %>%
  select(Province.State, Confirmed) %>%
  group_by(Province.State) %>%
  summarise(Total.Confirmed = max(Confirmed)) %>%
  arrange(desc(Total.Confirmed))
per_ch_province
```

```
## # A tibble: 31 x 2
##    Province.State Total.Confirmed
##    <fct>                    <dbl>
##  1 Hubei                    67103
##  2 Guangdong                 1350
##  3 Henan                     1272
##  4 Zhejiang                  1206
##  5 Hunan                     1018
##  6 Anhui                      990
##  7 Jiangxi                    935
##  8 Shandong                   758
##  9 Jiangsu                    631
## 10 Chongqing                  576
## # ... with 21 more rows
```

```
# bar graph spread per province
per_ch_province %>%
  #top_n(10) %>%    # select top 10
  ggplot(aes(x = reorder(Province.State,-Total.Confirmed), y = Total.Confirmed, fill = Province.State)) +
  geom_bar(stat = "identity") +
  labs(y = "Total Confirmed Case", x = "Province") +
  theme(legend.position="none", axis.text.x = element_text(angle=90))
```



5. Confirmed, Deaths, and Recovered

```
# latest data by country
data_by_country <- covid %>%
  select(Country.Region, Confirmed, Deaths, Recovered) %>%
  group_by(Country.Region) %>%
  summarise(Confirmed = max(Confirmed), Deaths = max(Deaths), Recovered = max(Recovered)) %>%
  arrange(desc(Confirmed))
data_by_country
```

```
## # A tibble: 75 x 4
##    Country.Region Confirmed Deaths Recovered
##    <fct>              <dbl>  <dbl>     <dbl>
##  1 Mainland China     67103   2803     33934
##  2 South Korea         4335     28        30
##  3 Italy               2036     52       149
##  4 Iran                1501     66       291
##  5 Others               705      6        10
##  6 Japan                274      6        32
##  7 France               191      3        12
##  8 Germany              159      0        16
##  9 Spain                120      0         2
## 10 Singapore            108      0        78
## # ... with 65 more rows
```

```
# case fatality rate by country (death/confirmed)
fatality_rate <- data_by_country %>%
  mutate(Fatality.Rate = Deaths/Confirmed) %>%
```

```
  arrange(desc(Fatality.Rate))
fatality_rate
```

```
## # A tibble: 75 x 5
##    Country.Region Confirmed Deaths Recovered Fatality.Rate
##    <fct>              <dbl>  <dbl>     <dbl>         <dbl>
##  1 Philippines            3      1         1         0.333
##  2 US                    45      5         2         0.111
##  3 Australia              9      1         4         0.111
##  4 Iran                1501     66       291         0.0440
##  5 Mainland China     67103   2803     33934         0.0418
##  6 Italy               2036     52       149         0.0255
##  7 Taiwan                41      1        12         0.0244
##  8 Thailand              43      1        31         0.0233
##  9 Japan                274      6        32         0.0219
## 10 Hong Kong            100      2        36         0.02
## # ... with 65 more rows
```

- Philippines had the highest case fatality rate with 33.3% which the death number was higher compared to the number of confirmed cases. Both US and Australia had case fatality rate of 11.1%.

```
# recovery rate by country (recovered/confirmed)
recovery_rate <- data_by_country %>%
  mutate(Recovery.Rate = Recovered/Confirmed) %>%
  arrange(desc(Recovery.Rate))
recovery_rate
```

```
## # A tibble: 75 x 5
##    Country.Region Confirmed Deaths Recovered Recovery.Rate
##    <fct>              <dbl>  <dbl>     <dbl>         <dbl>
##  1 Vietnam               16      0        16         1
##  2 Cambodia               1      0         1         1
##  3 Nepal                  1      0         1         1
##  4 Sri Lanka              1      0         1         1
##  5 Macau                 10      0         8         0.8
##  6 Singapore            108      0        78         0.722
##  7 Thailand              43      1        31         0.721
##  8 Russia                 3      0         2         0.667
##  9 Malaysia              29      0        18         0.621
## 10 India                  5      0         3         0.6
## # ... with 65 more rows
```

- Recovery rate for Vietnam, Cambodia, Nepal, and Sri Lanka were 100%. Everyone who were diagnosed with Corona virus were recovered in the 4 countries.

6. Total confirmed, deaths, and recovery nation wide

```
# latest corona virus spread nation wide
world_data <- data_by_country %>%
  select(Confirmed, Deaths, Recovered) %>%
  summarise(Confirmed = sum(Confirmed), Deaths = max(Deaths), Recovered = max(Recovered))
world_data
```

```
## # A tibble: 1 x 3
##   Confirmed Deaths Recovered
##       <dbl>  <dbl>     <dbl>
## 1     77299   2803     33934
```

```r
# case fatality rate world wide
w_fatality_rate <- world_data %>%
  mutate(Fatality.Rate = Deaths/Confirmed)
w_fatality_rate
```

```
## # A tibble: 1 x 4
##   Confirmed Deaths Recovered Fatality.Rate
##       <dbl>  <dbl>     <dbl>         <dbl>
## 1     77299   2803     33934        0.0363
```

```r
# recovery rate world wide
w_recovery_rate <- world_data %>%
  mutate(Recovery.Rate = Recovered/Confirmed)
w_recovery_rate
```

```
## # A tibble: 1 x 4
##   Confirmed Deaths Recovered Recovery.Rate
##       <dbl>  <dbl>     <dbl>         <dbl>
## 1     77299   2803     33934         0.439
```

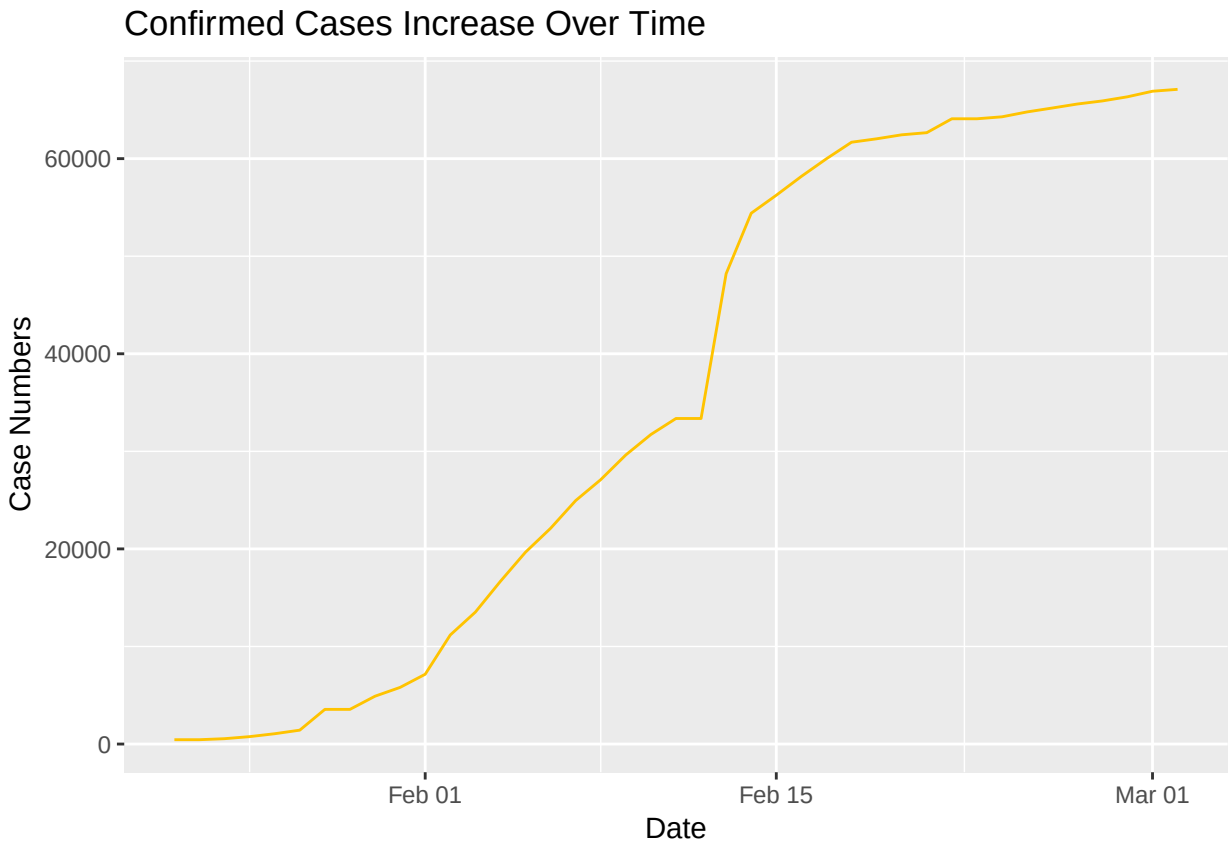
- The fatality rate was 3.63% and the recovery rate was 43.9%. The recovery rate is much more higher than the fatality rate.

7. Corona virus spread over time

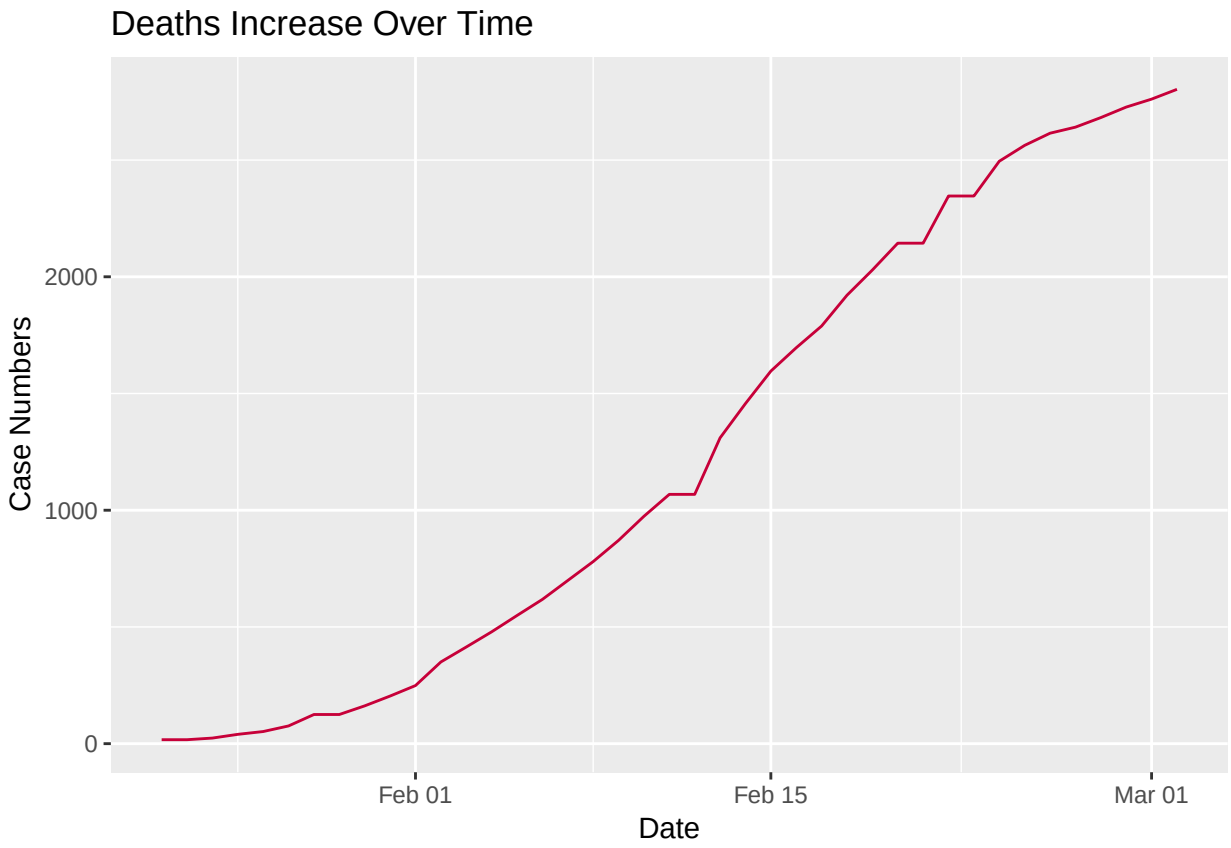
```r
# get the data over time
over_time_data <- covid %>%
  select(Date, Confirmed, Deaths, Recovered) %>%
  mutate(Date = as_date(mdy(Date))) %>%
  group_by(Date) %>%
  summarise(Confirmed = max(Confirmed), Deaths = max(Deaths), Recovered = max(Recovered))
over_time_data
```

```
## # A tibble: 41 x 4
##    Date       Confirmed Deaths Recovered
##    <date>         <dbl>  <dbl>     <dbl>
##  1 2020-01-22       444     17        28
##  2 2020-01-23       444     17        28
##  3 2020-01-24       549     24        31
##  4 2020-01-25       761     40        32
##  5 2020-01-26      1058     52        42
##  6 2020-01-27      1423     76        45
##  7 2020-01-28      3554    125        80
##  8 2020-01-29      3554    125        88
##  9 2020-01-30      4903    162        90
## 10 2020-01-31      5806    204       141
## # ... with 31 more rows
```

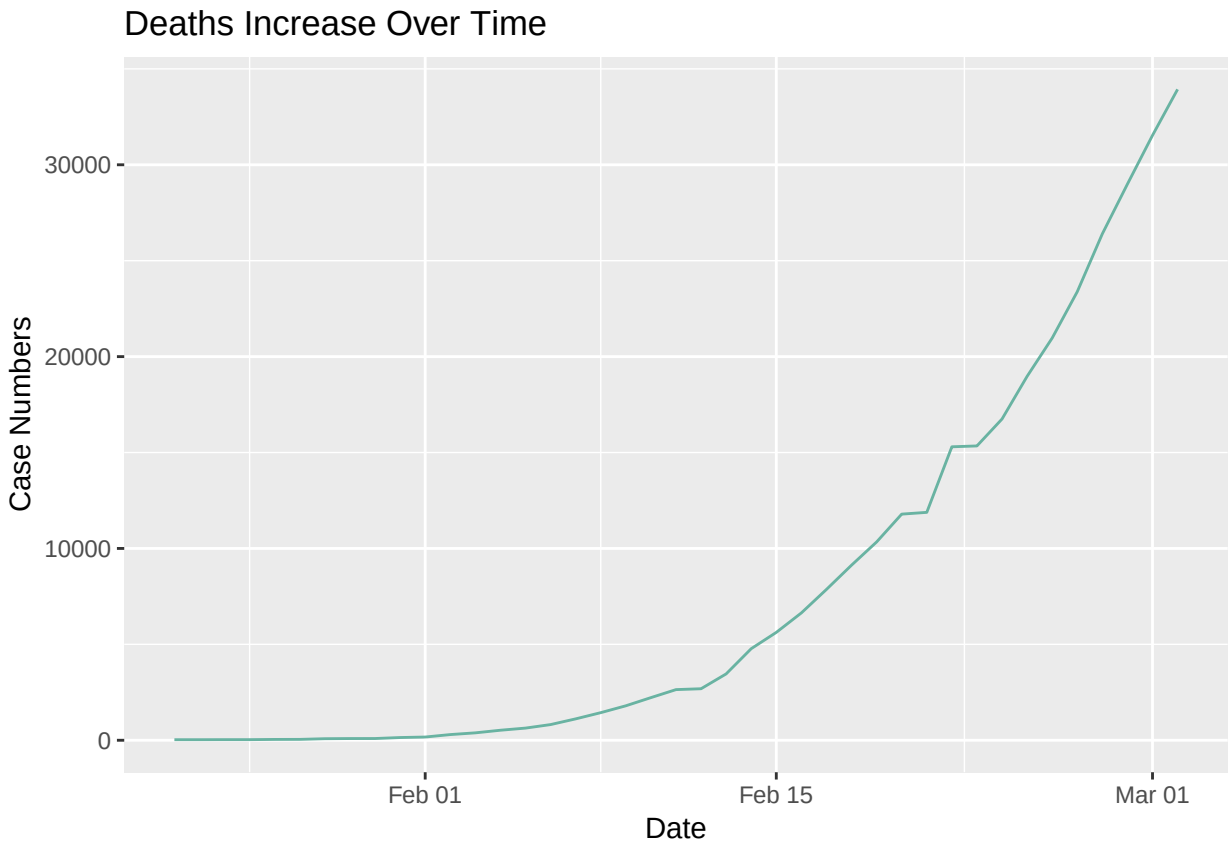
```r
# graph of confirmed cases over time
ggplot(over_time_data, aes(x=Date, y=Confirmed)) +
  geom_line(color="#FFC300") +
  ylab("Case Numbers") + ggtitle("Confirmed Cases Increase Over Time")
```




```
# graph of deaths over time
ggplot(over_time_data, aes(x=Date, y=Deaths)) +
  geom_line(color="#C70039") +
  ylab("Case Numbers") + ggtitle("Deaths Increase Over Time")
```




```
# graph of recovery over time
ggplot(over_time_data, aes(x=Date,y=Recovered)) +
  geom_line(color="#69b3a2") +
  ylab("Case Numbers") + ggtitle("Deaths Increase Over Time")
```



8. Forecasting number of confirmed cases world wide by March 10th

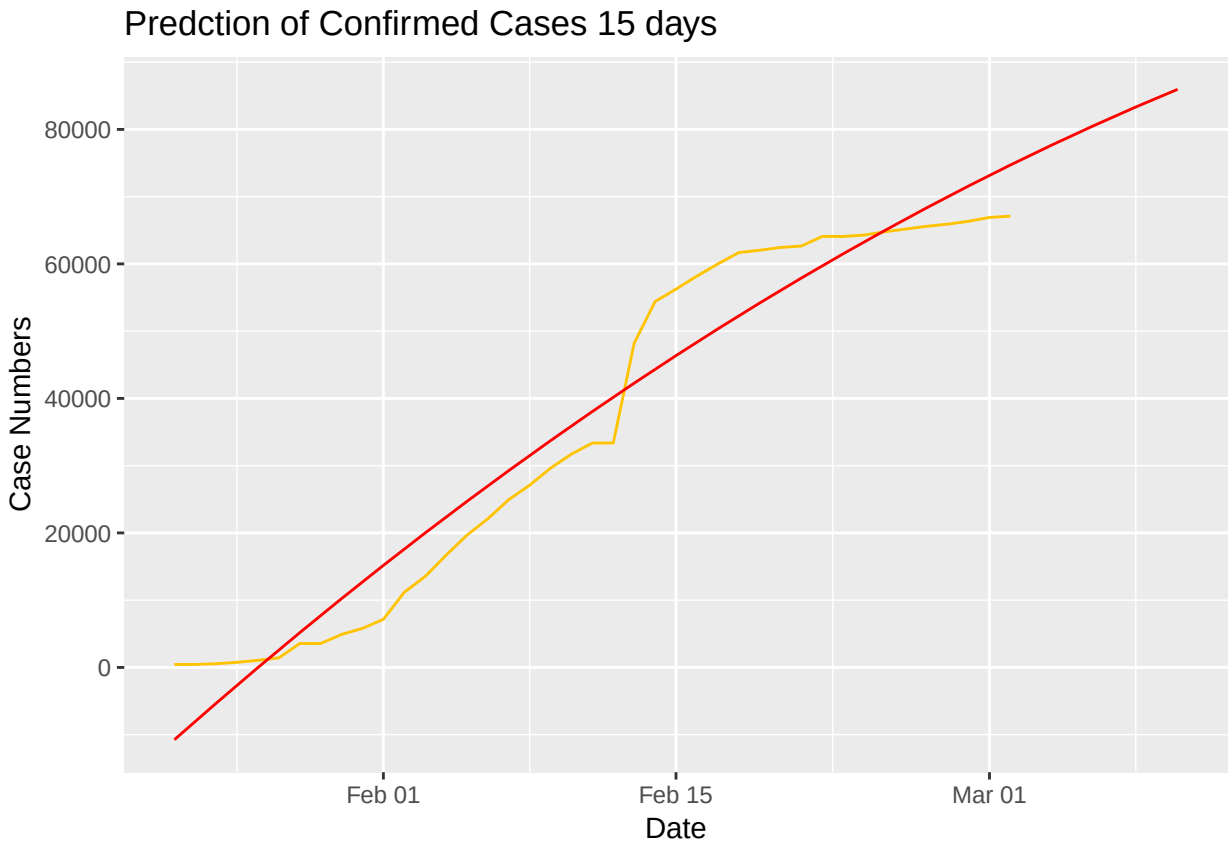
```
# logistic regression model prediction
lm <-lm(Confirmed~poly(Date,2),data = over_time_data)
pred <- data.frame(Date=over_time_data$Date+0:15)
```

```
## Warning in unclass(e1) + unclass(e2): longer object length is not a
## multiple of shorter object length
```

```
pred$Confirmed <- predict(lm(Confirmed~poly(Date,2),data = over_time_data), newdata=pred)
tail(pred)
```

```
##          Date Confirmed
## 36 2020-02-29  71586.34
## 37 2020-03-02  74703.90
## 38 2020-03-04  77699.87
## 39 2020-03-06  80574.25
## 40 2020-03-08  83327.04
## 41 2020-03-10  85958.24
```

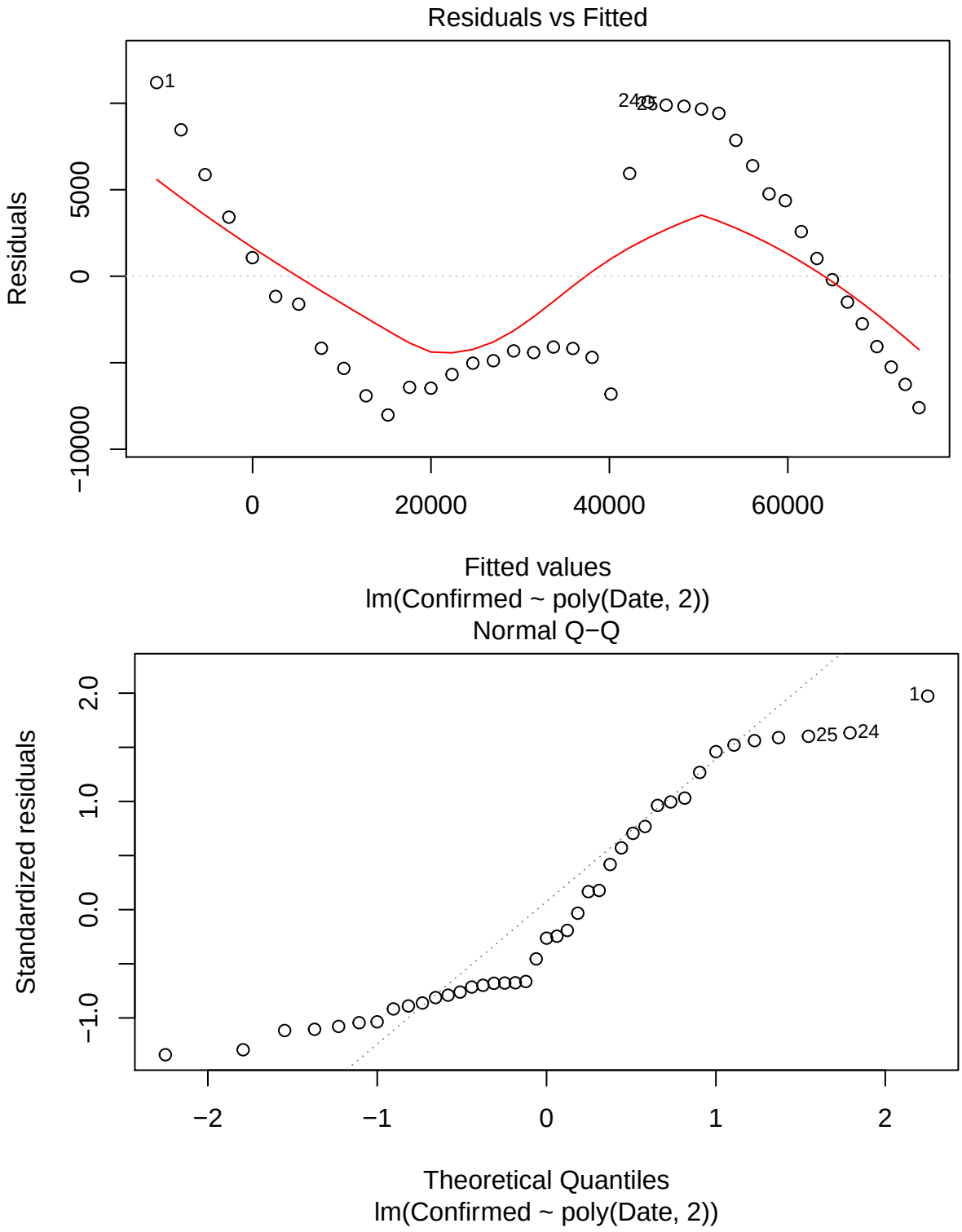
```
# plot prediction
ggplot(over_time_data, aes(x=Date, y=Confirmed)) +
  geom_line(color="#FFC300") +
  ylab("Case Numbers") + ggtitle("Predction of Confirmed Cases 15 days") +
  geom_line(data=pred, color="red")
```

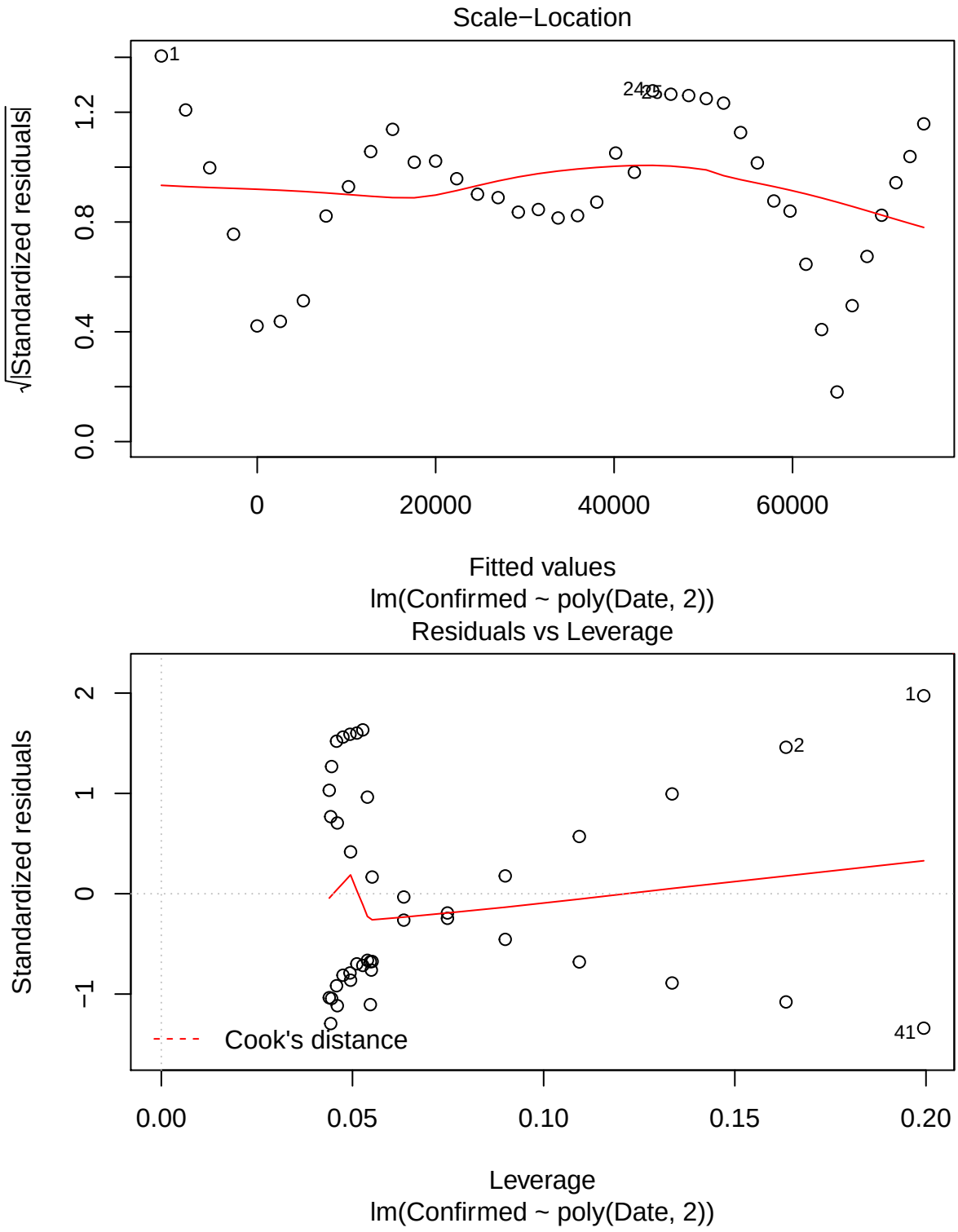


- Using linear regression model, the confirmed cases will be 85959 world wide by March 10th.

9. Check for model adequacy

```
# model adequacy
plot(lm)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Confirmed ~ poly(Date, 2))
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Confirmed ~ poly(Date, 2))



The residuals are normally not distributed and the residuals are not random and the variance of the residuals is not constant. The model did not pass the adquacy test and needs to explore other models.