The Analysis of Coronavirus (COVID-19) New Confirmed Cases Comparison Between Top 10 Countries and Taiwan

Technical Detail

Micky Lee (Wen Chi Lee)
UCLA Extension
Introduction to Data Science
Instructor: Daniel D. Gutierrez
March 22, 2020

Contents

1. Introduction

1.1 Project Description

Since the novel coronavirus (COVID-19) had spread all over the whole world, I want to find out how the new confirmed cases changed in the top 10 confirmed cases countries and my country, Taiwan. I use multiple linear regression to predict and fit between 11 countries. Then I compare the trends of these countries and try to explain the differences and why the result comes out.

1.2 The Dataset

The dataset source is from https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset. I use the **covid_19_data.csv** as my dataset, which contains 6722 observations and eight variables.

Variable	**Description**
SNo	The sequence number or data
ObservationDate	The date when confirmed observed
Province/State	The province or state where found confirmed
Country/Region	The country the confirmed case belongs to
Last Update	Last update time
Confirmed	The accumulated confirmed cases until the observation date
Deaths	The accumulated deaths until the observation date
Recovered	The accumulated recovered cases until the observation date

The dataset collected confirmed, deaths, and recovered data from 179 countries from January 22, 2020, to March 18, 2020.

2. Data Munging

2.1 Basic understanding

The dataset is loaded from the CSV file and assigns to a variable **virus**; here is a summary.

```
summary(virus)
```

```
      SNo        ObservationDate   Province.State        Country.Region
 Min.   :   1   03/18/2020: 284            :2766   Mainland China:1765
 1st Qu.:1681   03/17/2020: 276   Gansu    :  59   US            :1388
 Median :3362   03/16/2020: 272   Hebei    :  59   Australia     : 287
 Mean   :3362   03/09/2020: 266   Anhui    :  57   Canada        : 208
 3rd Qu.:5042   03/15/2020: 258   Beijing  :  57   France        :  91
 Max.   :6722   03/08/2020: 255   Chongqing:  57   UK            :  68
                (Other)   :5111   (Other)  :3667   (Other)       :2915
              Last.Update     Confirmed           Deaths          Recovered
 2020-03-11T20:00:00: 197   Min.   :    0.0   Min.   :   0.00   Min.   :    0.0
 2020-03-16T14:38:45: 105   1st Qu.:    2.0   1st Qu.:   0.00   1st Qu.:    0.0
 2020-03-15T18:20:18:  87   Median :   13.0   Median :   0.00   Median :    0.0
 2020-02-01T19:43:03:  63   Mean   :  601.2   Mean   :  19.86   Mean   :  226.3
 2020-02-01T19:53:03:  63   3rd Qu.:  108.0   3rd Qu.:   1.00   3rd Qu.:   11.0
 2020-02-24T23:33:02:  63   Max.   :67800.0   Max.   :3122.00   Max.   :56927.0
 (Other)            :6144
```

There is no NA that exist in the data. Let's preview the data.

```
head(virus)
```

```
  SNo ObservationDate Province.State Country.Region     Last.Update Confirmed Deaths
1   1      01/22/2020          Anhui Mainland China 1/22/2020 17:00         1      0
2   2      01/22/2020        Beijing Mainland China 1/22/2020 17:00        14      0
3   3      01/22/2020      Chongqing Mainland China 1/22/2020 17:00         6      0
4   4      01/22/2020         Fujian Mainland China 1/22/2020 17:00         1      0
5   5      01/22/2020          Gansu Mainland China 1/22/2020 17:00         0      0
6   6      01/22/2020      Guangdong Mainland China 1/22/2020 17:00        26      0
  Recovered
1         0
2         0
3         0
4         0
5         0
6         0
```

2.2 Transformatting the data

We do not need the **Last.Update** and **SNo** columns, so we remove them from the data.

We choose the top 10 confirmed cases countries, which are China, Italy, Spain, Germany, Iran, the United States, France, South Korea, Switzerland, United Kindom, and my country, Taiwan.

Some countries have provinces or states information, but others do not have this information. We need to separate them and use several customed functions to calculate the daily new confirmed cases, new deaths, and new recovered cases.

The **groupByProvinceStateDate** function selects the data by the specific country and groups output by the **ProvinceState** and **ObservationDate** columns.

```
groupByProvinceStateDate <- function(data, country) {
  # convert date to string before using sqldf to avoid sqldf change the date type
  data$ObservationDate <- as.character(data$ObservationDate)

  newData <- sqldf(sprintf("select ObservationDate, ProvinceState,
                  sum(Confirmed) as Confirmed,
                  sum(Deaths) as Deaths, sum(Recovered) as Recovered
                  from data
                  where CountryRegion = '%s'
                  group by ProvinceState, ObservationDate", country))

  # convert string to date again
  newData$ObservationDate <- as.Date(newData$ObservationDate)
  newData
}
```

The **groupByDate** function produces a new data.frame only group by **ObservationDate**.

```
groupByDate <- function(countryData) {
  sqldf("select ObservationDate, sum(Confirmed) as Confirmed,
        sum(Deaths) as Deaths,
        sum(Recovered) as Recovered
        from countryData
        group by ObservationDate")
}
```

We use the above functions to sum the accumulated count of each country, which has states or provinces. Therefore we will have only one record each day for each country.

```
chinaByProvince <- groupByProvinceStateDate(virus, "Mainland China")
head(chinaByProvince)
```

```
  ObservationDate ProvinceState Confirmed Deaths Recovered
1      2020-01-22         Anhui         1      0         0
2      2020-01-23         Anhui         9      0         0
3      2020-01-24         Anhui        15      0         0
4      2020-01-25         Anhui        39      0         0
5      2020-01-26         Anhui        60      0         0
6      2020-01-27         Anhui        70      0         0
```

```
chinaByDate <- groupByDate(chinaByProvince)
head(chinaByDate)
```

```
  ObservationDate Confirmed Deaths Recovered
1      2020-01-22       547     17        28
2      2020-01-23       639     18        30
3      2020-01-24       916     26        36
4      2020-01-25      1399     42        39
5      2020-01-26      2062     56        49
6      2020-01-27      2863     82        58
```

Then we use the **calculateNew** function to calculate the new confirmed cases, new deaths, and new recovered cases according to the previous day.

```
calculateNew <- function(countryData) {
  countryData %>%
    mutate(ConfirmedNew = order_by(ObservationDate, Confirmed-lag(Confirmed))) %>%
    mutate(ConfirmedNew = ifelse(is.na(ConfirmedNew), 0, ConfirmedNew)) %>%
    mutate(DeathsNew = order_by(ObservationDate, Deaths - lag(Deaths))) %>%
    mutate(DeathsNew = ifelse(is.na(DeathsNew), 0, DeathsNew)) %>%
    mutate(RecoveredNew = order_by(ObservationDate, Recovered-lag(Recovered))) %>%
    mutate(RecoveredNew = ifelse(is.na(RecoveredNew), 0, RecoveredNew))
}

chinaNew <- calculateNew(chinaByDate)
head(chinaNew)
```

```
  ObservationDate Confirmed Deaths Recovered ConfirmedNew DeathsNew RecoveredNew
1      2020-01-22       547     17        28            0         0            0
2      2020-01-23       639     18        30           92         1            2
3      2020-01-24       916     26        36          277         8            6
4      2020-01-25      1399     42        39          483        16            3
5      2020-01-26      2062     56        49          663        14           10
6      2020-01-27      2863     82        58          801        26            9
```

Now, we have three new columns called **ConfirmedNew**, **DeathsNew**, and **RecoveredNew**.

Some countries, like Italy and Iran, don't have provinces or states; we use another way to produce the same data frame.

```
italy <- subset(virus, CountryRegion == "Italy")
italyNew <- calculateNew(italy)
italyNew <- italyNew[, c(-2,-3)]    # remove column ProvinceState, CountryRegion
italyNew$ObservationDate <- as.Date(italyNew$ObservationDate) #convert to date
head(italyNew)
```

```
  ObservationDate Confirmed Deaths Recovered ConfirmedNew DeathsNew RecoveredNew
1      2020-01-31         2      0         0            0         0            0
2      2020-02-01         2      0         0            0         0            0
3      2020-02-02         2      0         0            0         0            0
4      2020-02-03         2      0         0            0         0            0
5      2020-02-04         2      0         0            0         0            0
6      2020-02-05         2      0         0            0         0            0
```

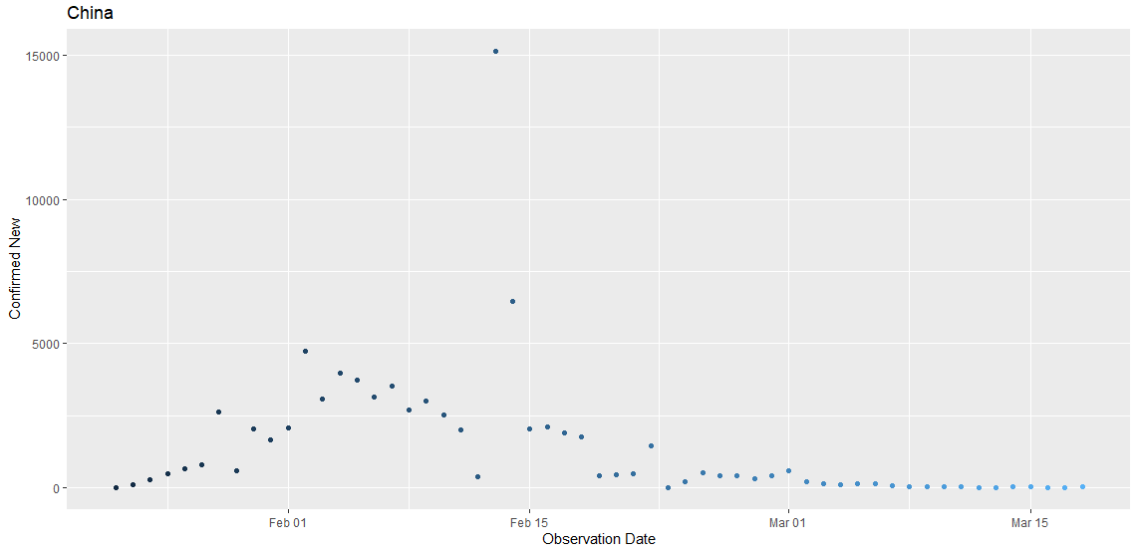
We process all the countries by these functions to get 11 new subsets .

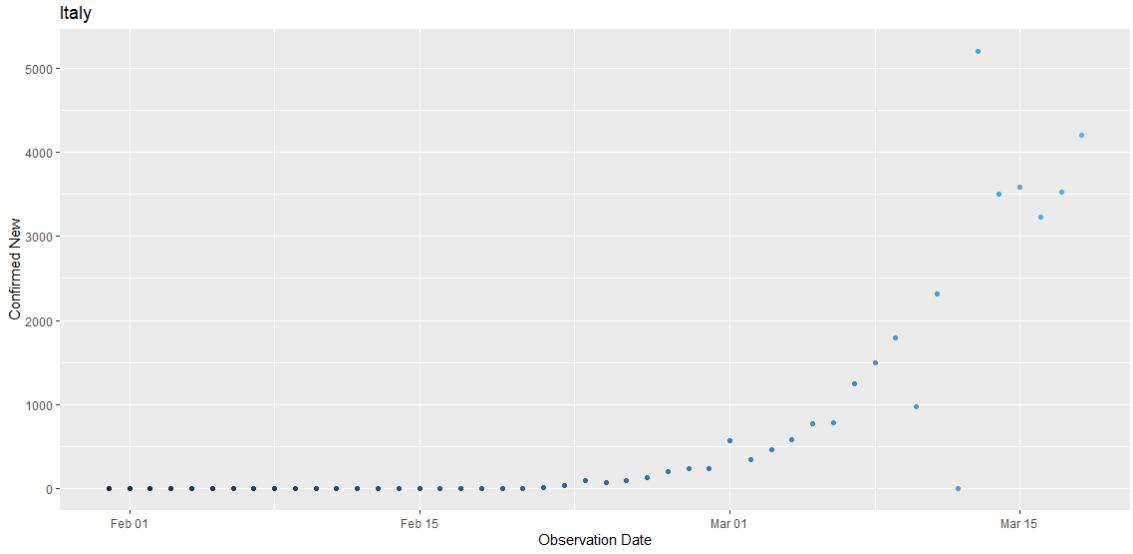
3. Exploratory Data Analysis (EDA)

3.1 Scatterplots

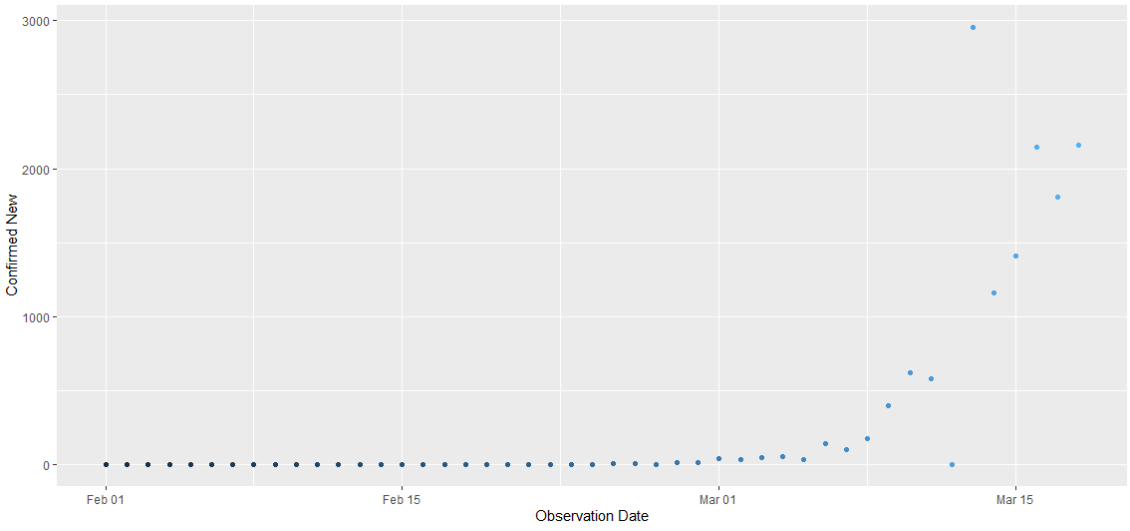
We use **ggplot** to observe the relationship between **ObservationDate** and **ConfirmedNew** variables in each country.

```
ggplot(data=chinaNew, mapping=aes(x=ObservationDate, y=ConfirmedNew,
 color=ObservationDate)) +
 geom_point() +
 ggtitle("China") + xlab("Observation Date") + ylab("Confirmed New") +
 theme(legend.position = "none")
```

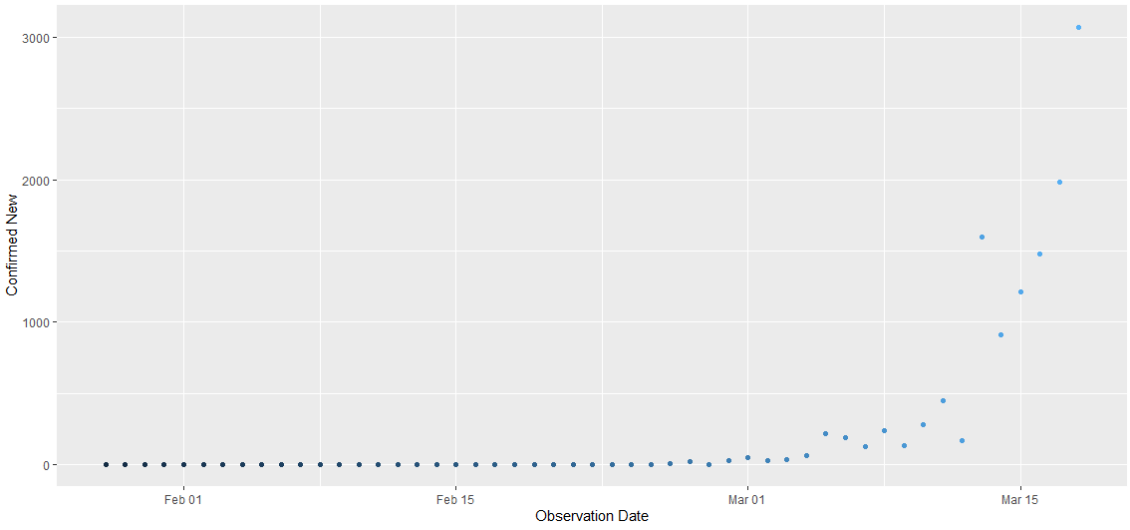




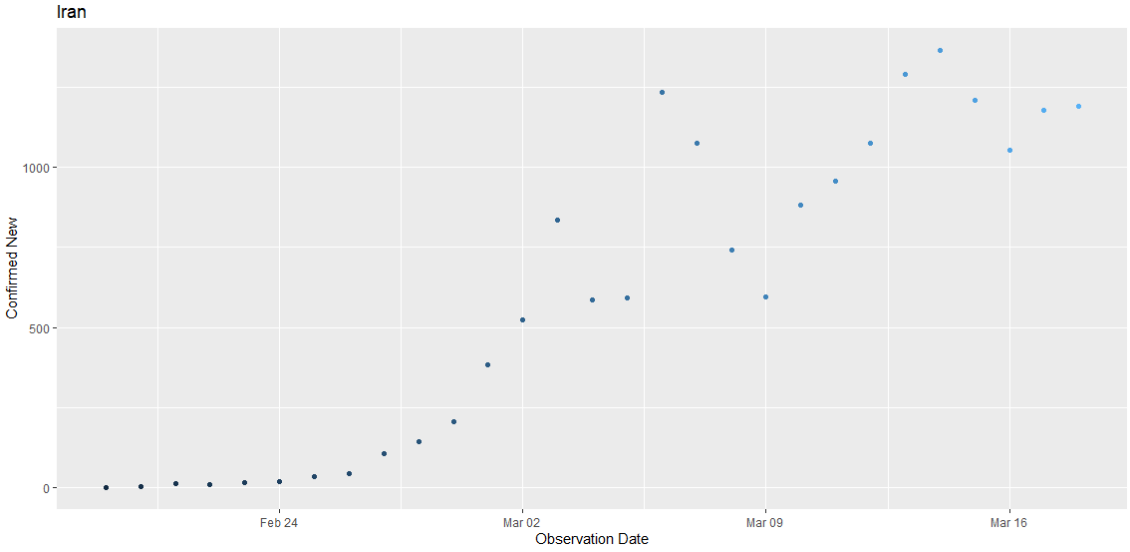
Spain

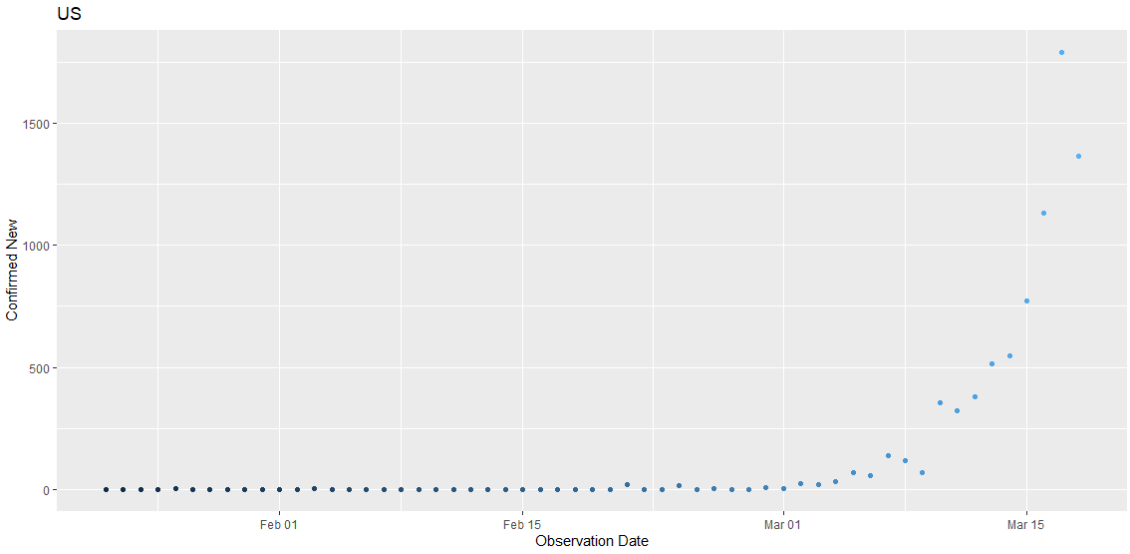


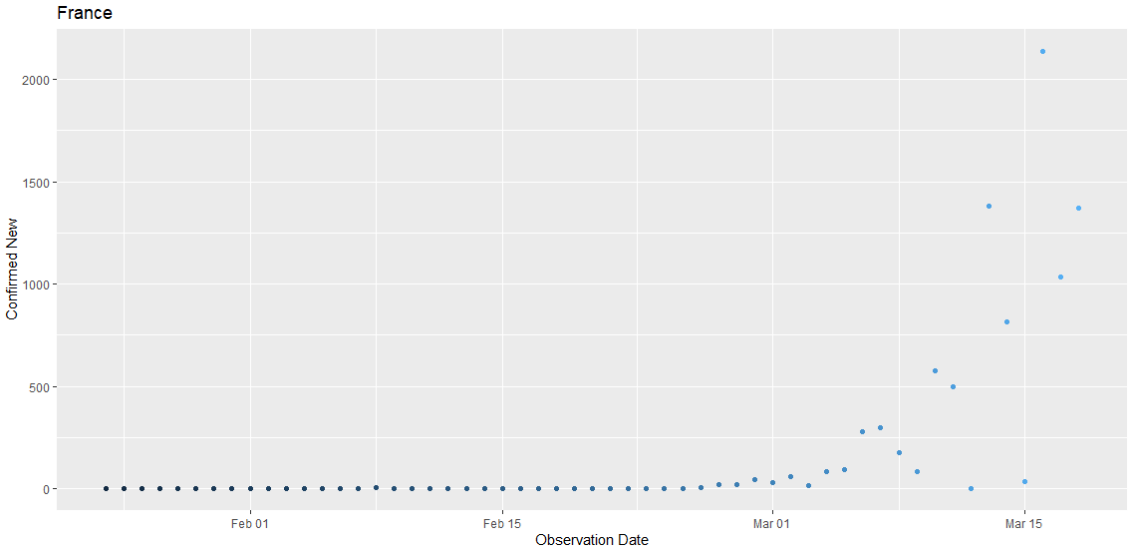
Germany

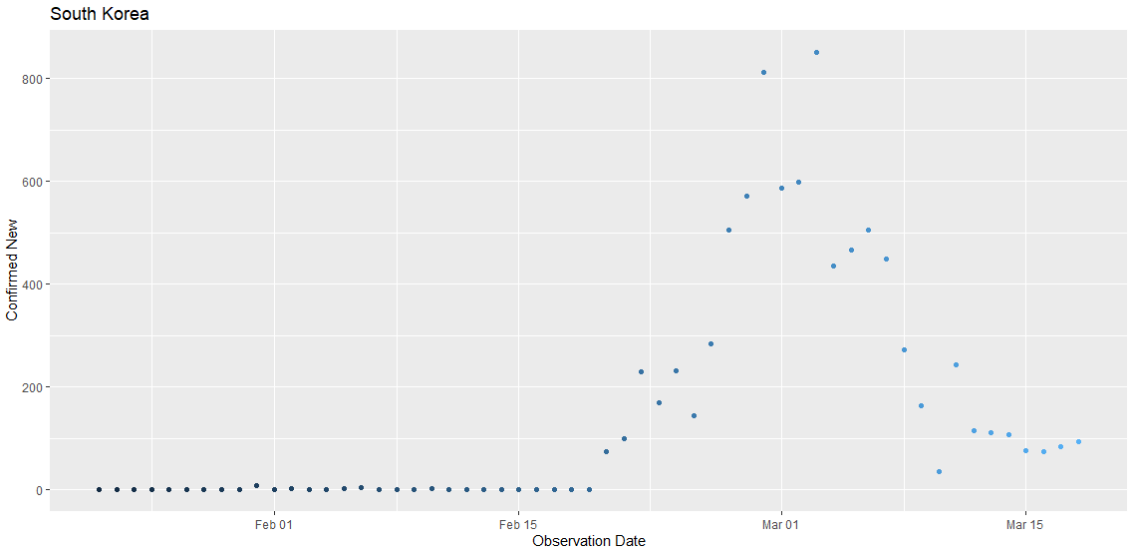


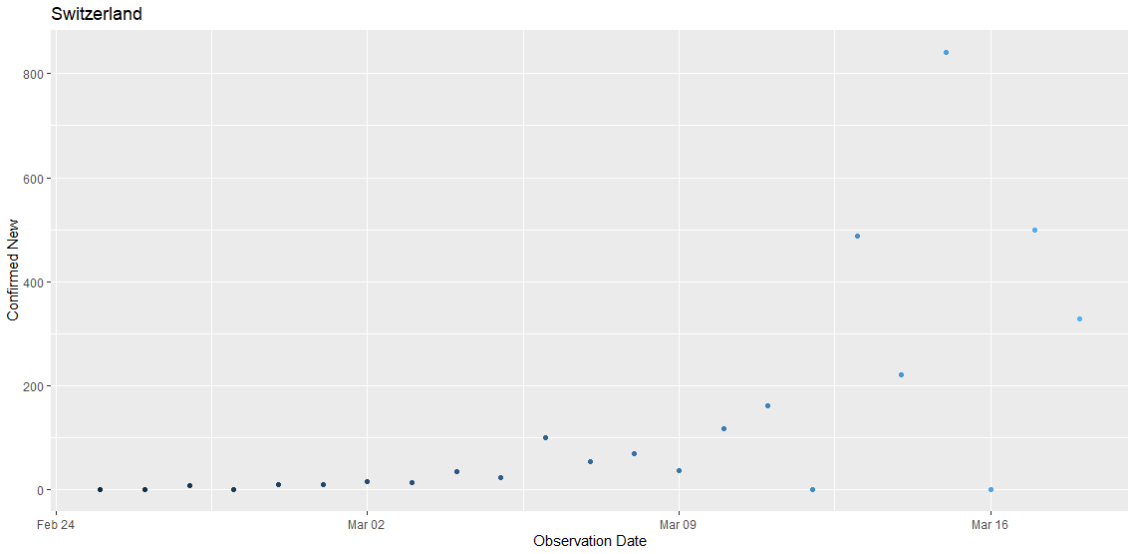
Iran

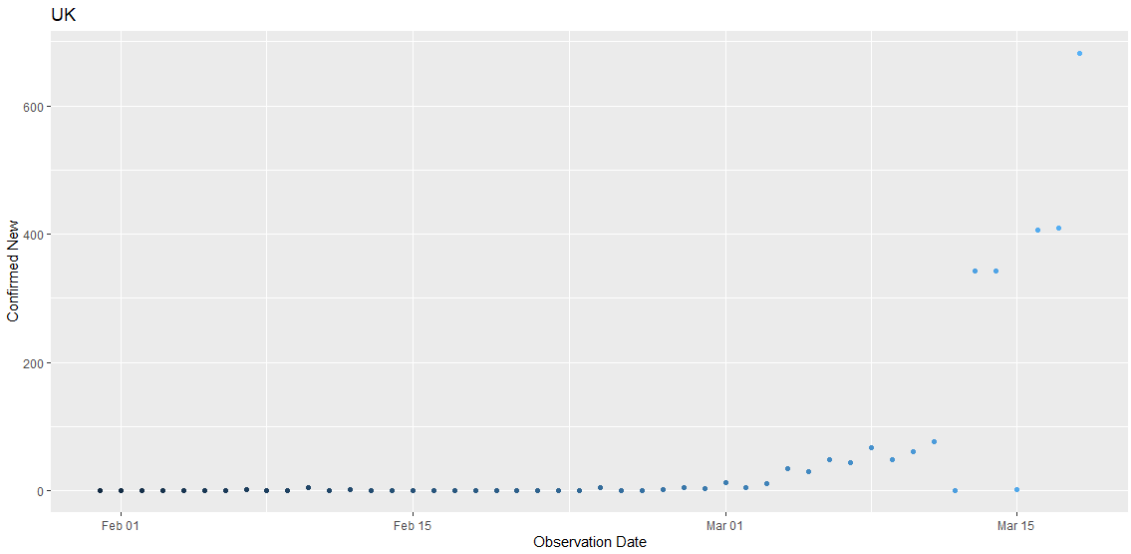


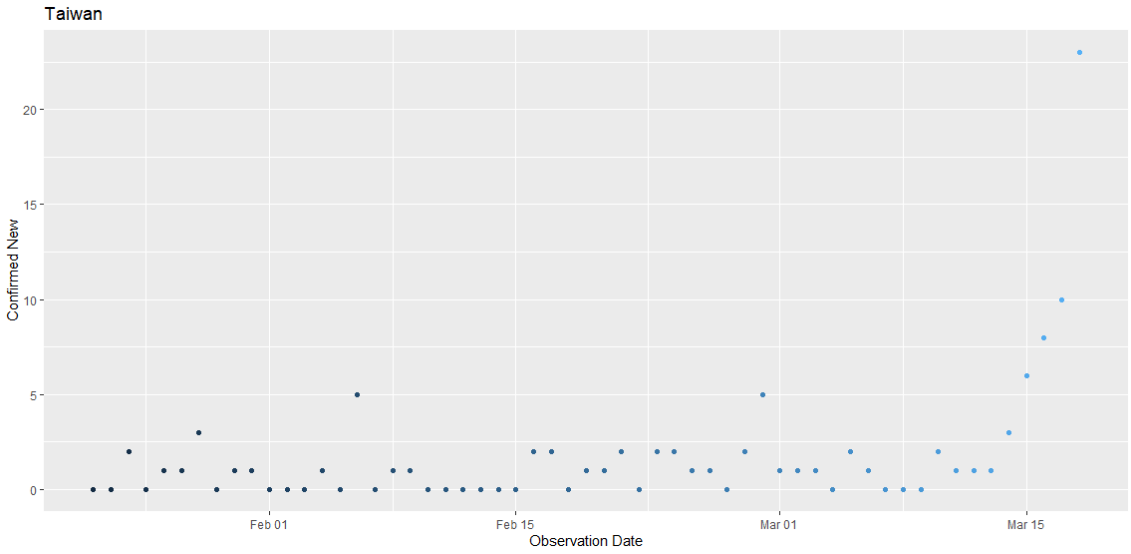

US
1500
1000
500
0
Confirmed New
Feb 01
Feb 15
Mar 01
Mar 15
Observation Date


France
2000
1500
1000
500
0
Confirmed New
Feb 01
Feb 15
Mar 01
Mar 15
Observation Date


South Korea
800
600
400
200
0
Confirmed New
Feb 01
Feb 15
Mar 01
Mar 15
Observation Date


Switzerland
800
600
400
200
0
Confirmed New
Feb 24
Mar 02
Mar 09
Mar 16
Observation Date


UK
600
400
200
0
Confirmed New
Feb 01
Feb 15
Mar 01
Mar 15
Observation Date

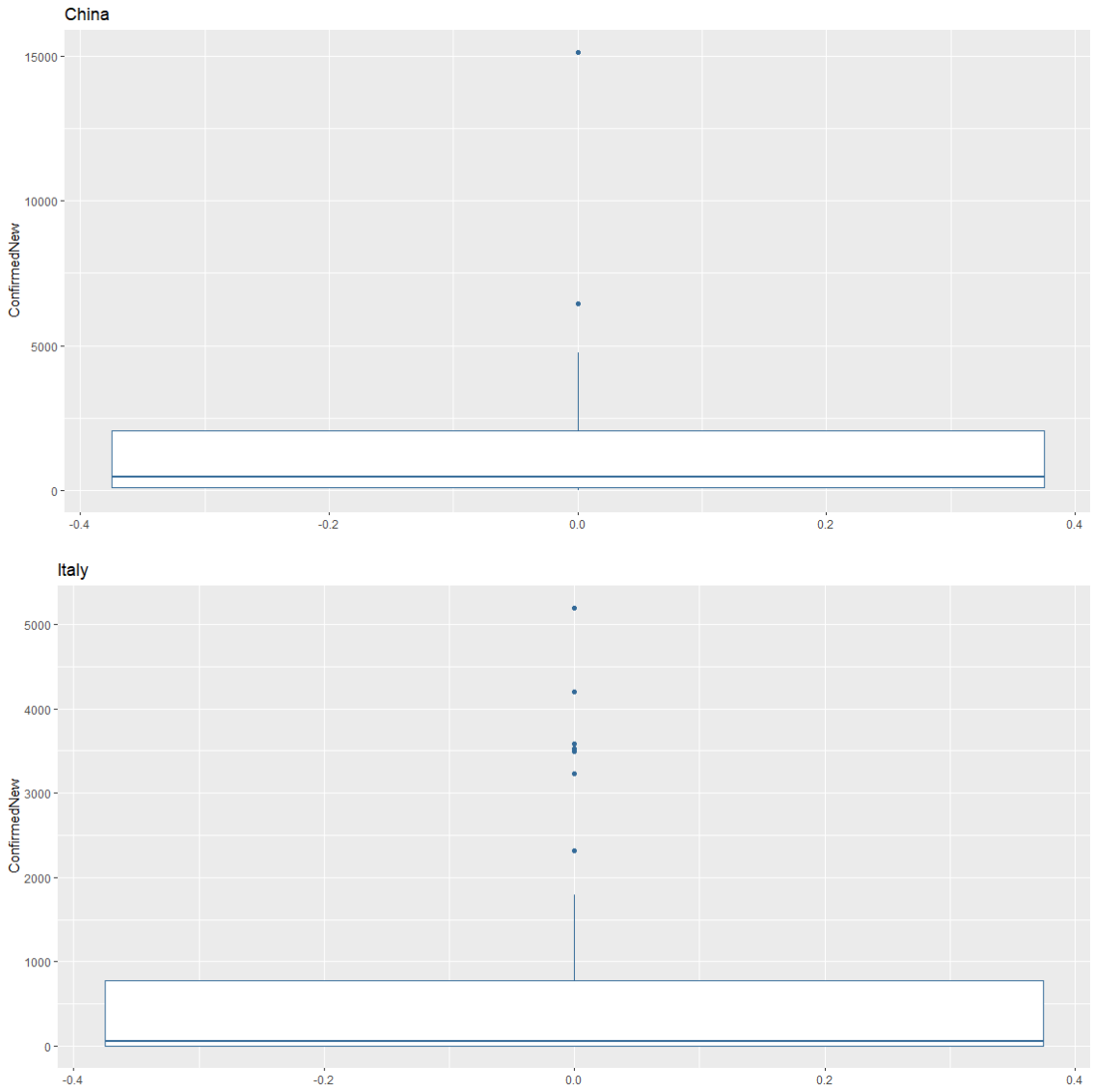

Taiwan
20
15
10
5
0
Confirmed New
Feb 01
Feb 15
Mar 01
Mar 15
Observation Date

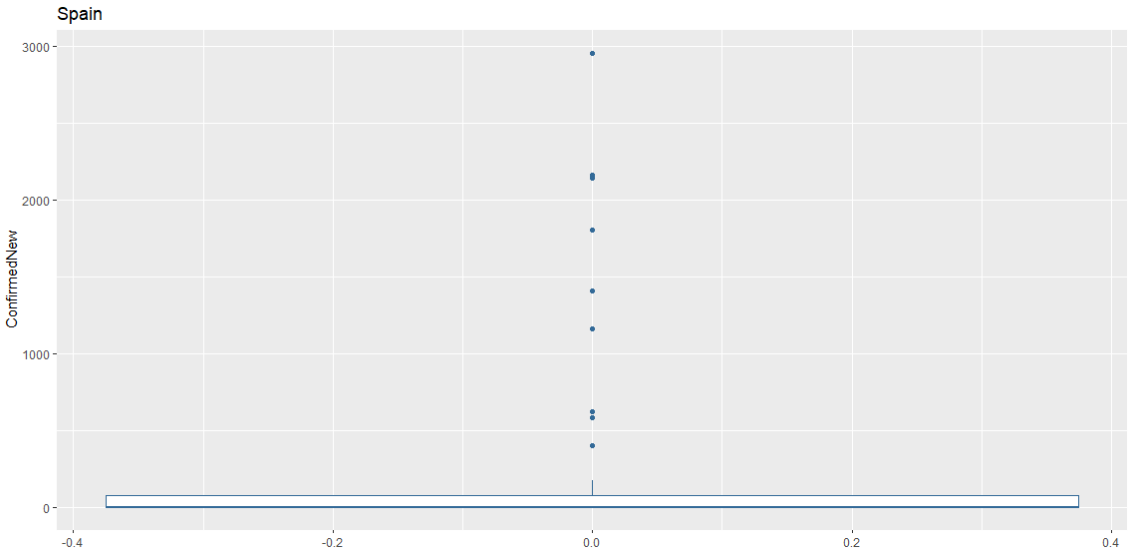
According to above scatterplots, almost all country has an increasing or decreasing trend of new confirmed cases daily. The plot of South Korea shows that it has controlled the newly confirmed cases in a stable decreasing trend from the peak. However, China's plot seems to be strange. It goes above 2500 new confirmed daily to above 15000 new confirmed cases daily, and then the new confirmed cases drastically down to only double digits daily, which is impossible and hard to believe.

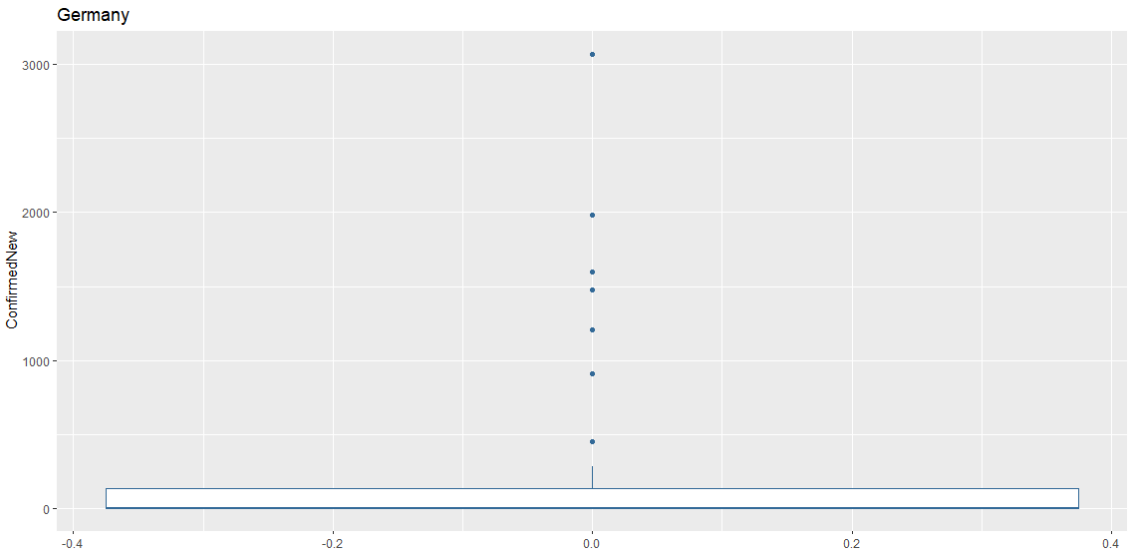
3.2 Boxplots

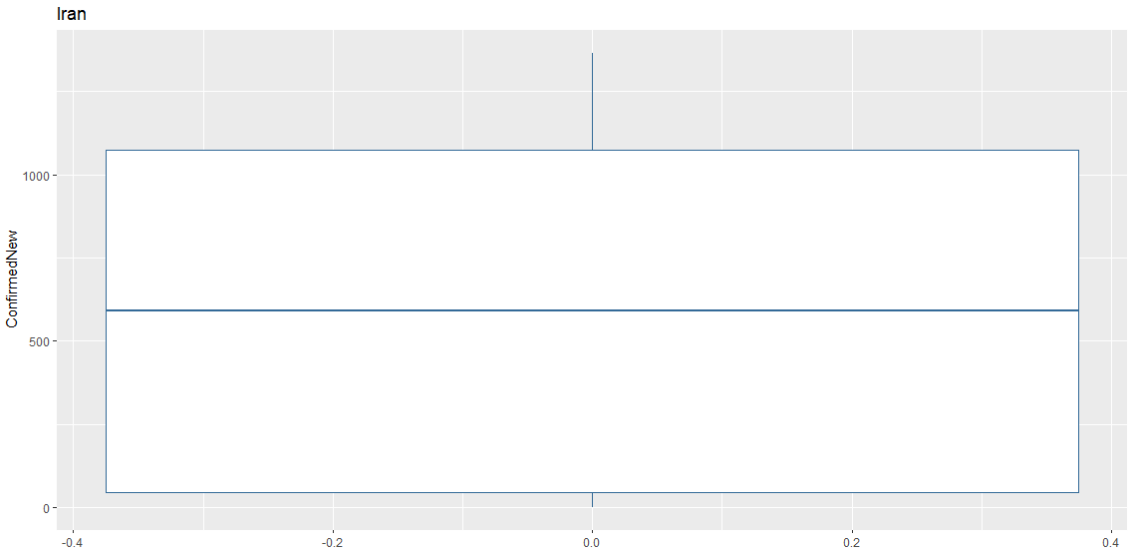
We use boxplots to view the quantitative distribution of each country's data.

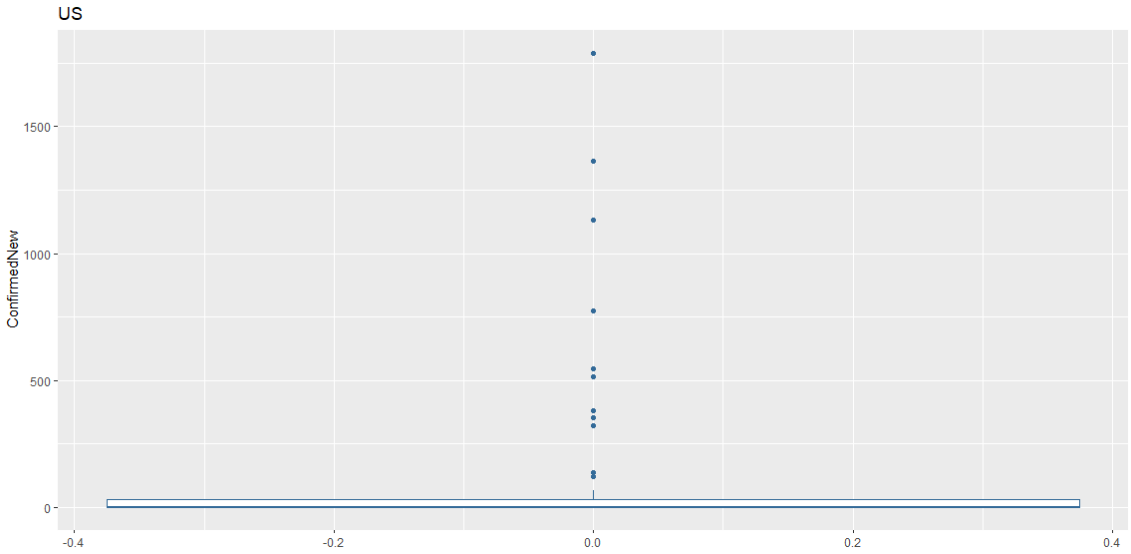
```
ggplot(chinaNew)+geom_boxplot(aes(y=ConfirmedNew,color=1)) +
  ggtitle("China") + theme(legend.position = "none")
```

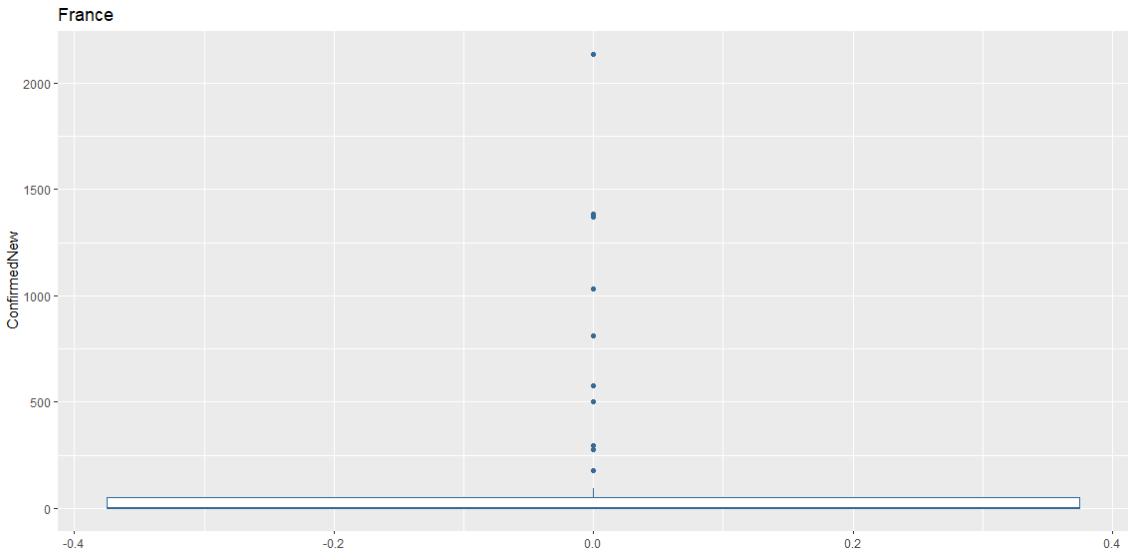


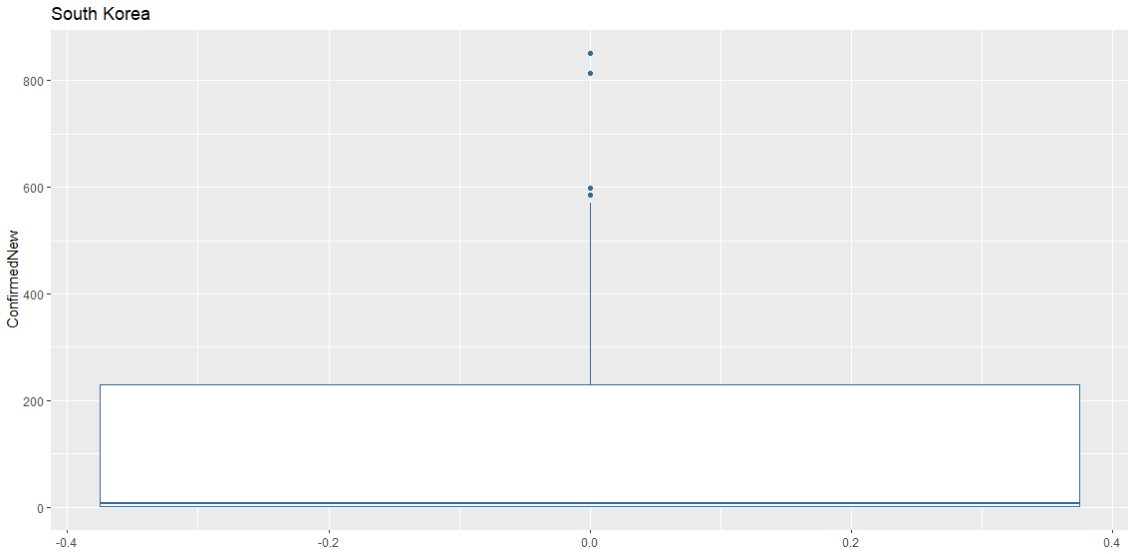

Spain
ConfirmedNew
3000
2000
1000
0
-0.4
-0.2
0.0
0.2
0.4

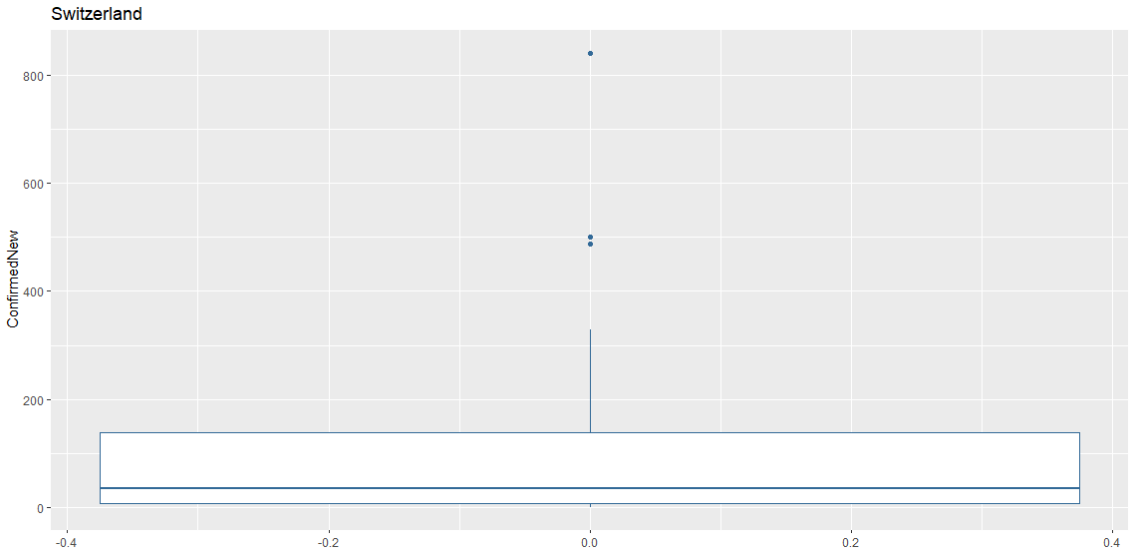

Germany
ConfirmedNew
3000
2000
1000
0
-0.4
-0.2
0.0
0.2
0.4

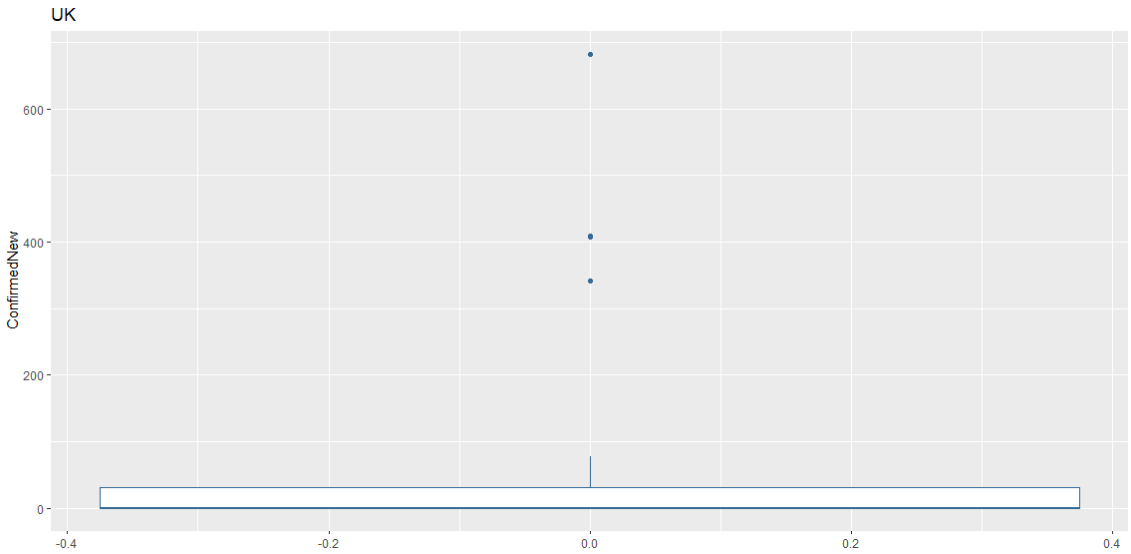

Iran
ConfirmedNew
1000
500
0
-0.4
-0.2
0.0
0.2
0.4

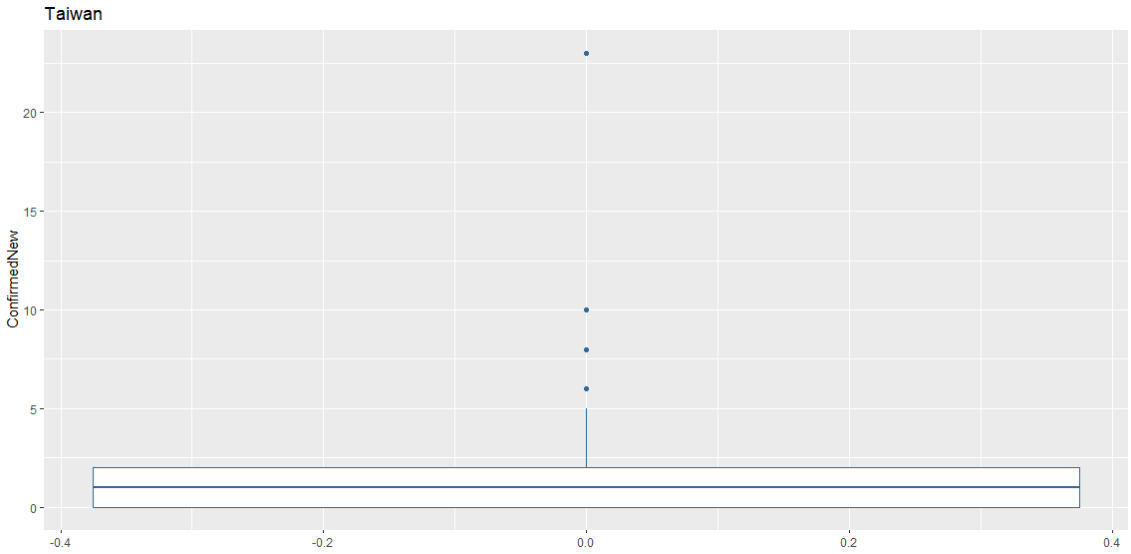

US
ConfirmedNew
1500
1000
500
0
-0.4
-0.2
0.0
0.2
0.4


France
ConfirmedNew
2000
1500
1000
500
0
-0.4
-0.2
0.0
0.2
0.4


South Korea
ConfirmedNew
800
600
400
200
0
-0.4
-0.2
0.0
0.2
0.4


Switzerland
ConfirmedNew
800
600
400
200
0
-0.4
-0.2
0.0
0.2
0.4


UK
ConfirmedNew
600
400
200
0
-0.4
-0.2
0.0
0.2
0.4


Taiwan
ConfirmedNew
20
15
10
5
0
-0.4
-0.2
0.0
0.2
0.4

Those boxplots seem not to be so useful except Iran's boxplot. The boxplot of Iran has a normal distribution of the data and quantitative variables. We can see the third quartile of China is very close to 2500 new confirmed cases a day.

3.3 Correlation matrix pairs

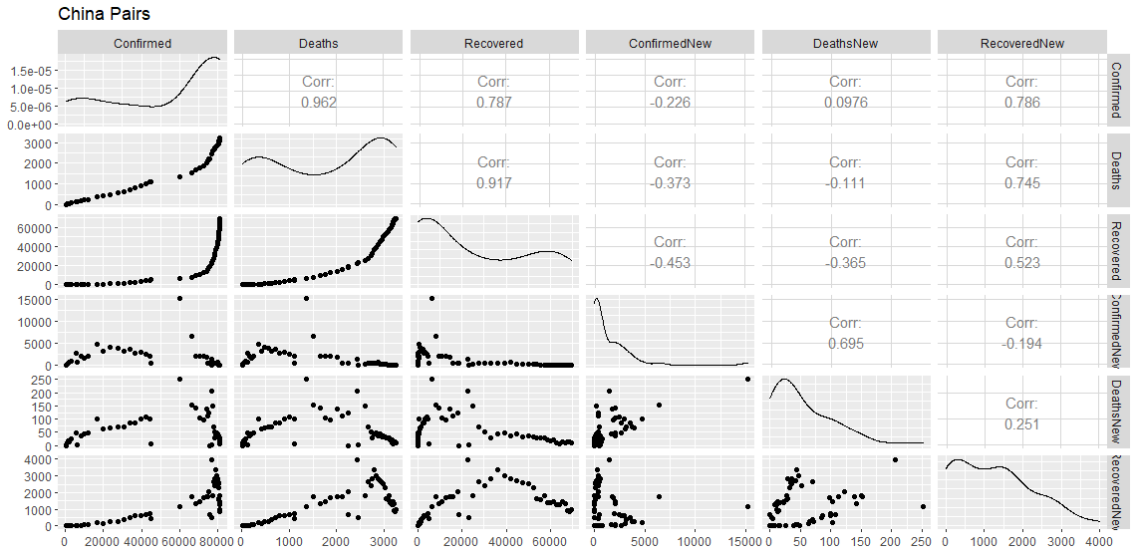
Next, we compute the correlation matrix to view the relationship between variables.

```
cor(chinaNew[,c(2:7)])
```

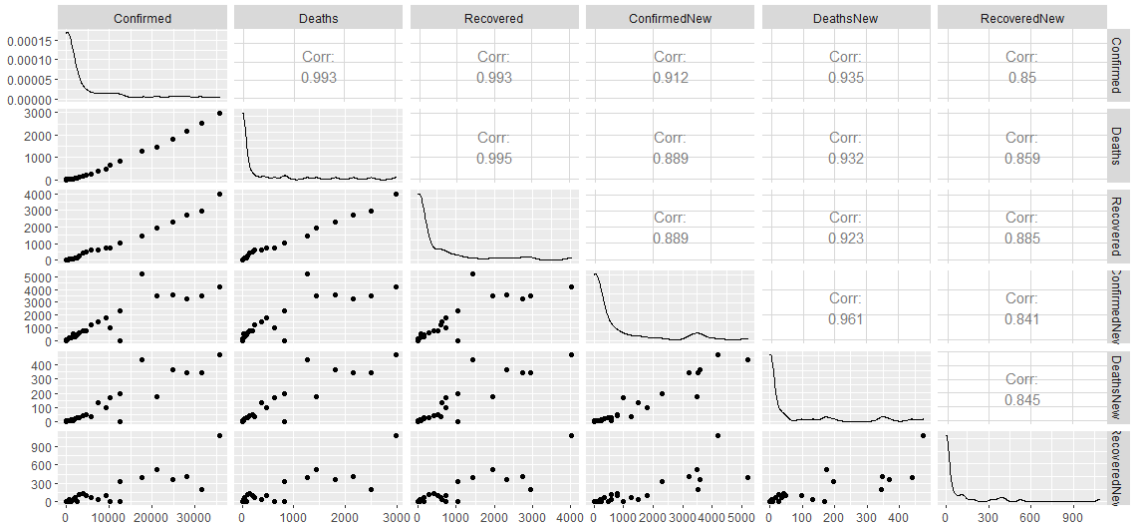
```
                Confirmed     Deaths  Recovered ConfirmedNew   DeathsNew RecoveredNew
Confirmed     1.00000000  0.9620572  0.7870693   -0.2264827  0.09757346    0.7861816
Deaths        0.96205718  1.0000000  0.9170125   -0.3726086 -0.11122551    0.7454746
Recovered     0.78706933  0.9170125  1.0000000   -0.4527927 -0.36514645    0.5230210
ConfirmedNew -0.22648265 -0.3726086 -0.4527927    1.0000000  0.69482854   -0.1936115
DeathsNew     0.09757346 -0.1112255 -0.3651465    0.6948285  1.00000000    0.2509042
RecoveredNew  0.78618158  0.7454746  0.5230210   -0.1936115  0.25090417    1.0000000
```

Since it is easy to understand images than words, we draw plot pairs of each country by using **ggpairs**.

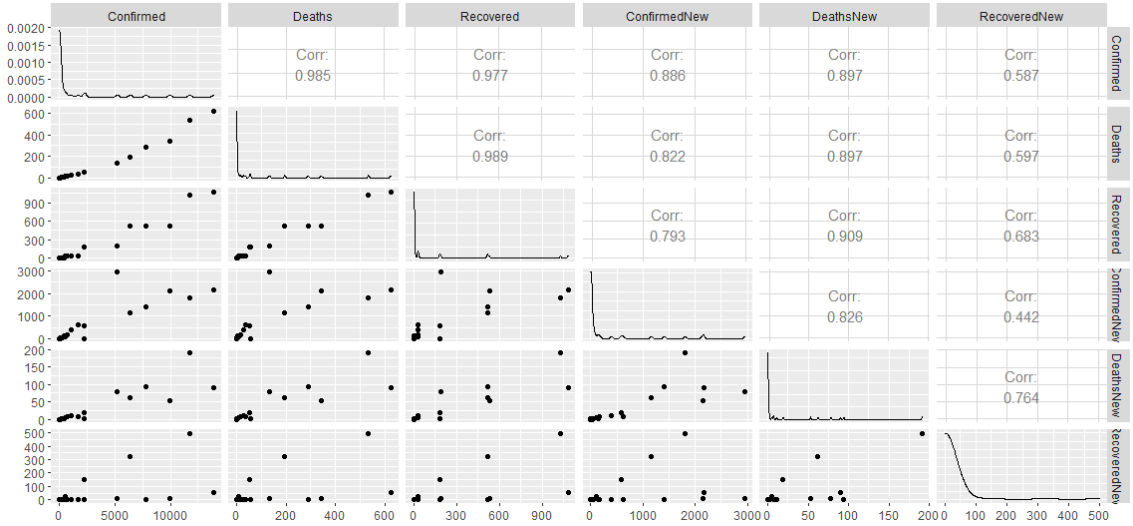
```
ggpairs(chinaNew[,c(2:7)]) + ggtitle("China Pairs")
```



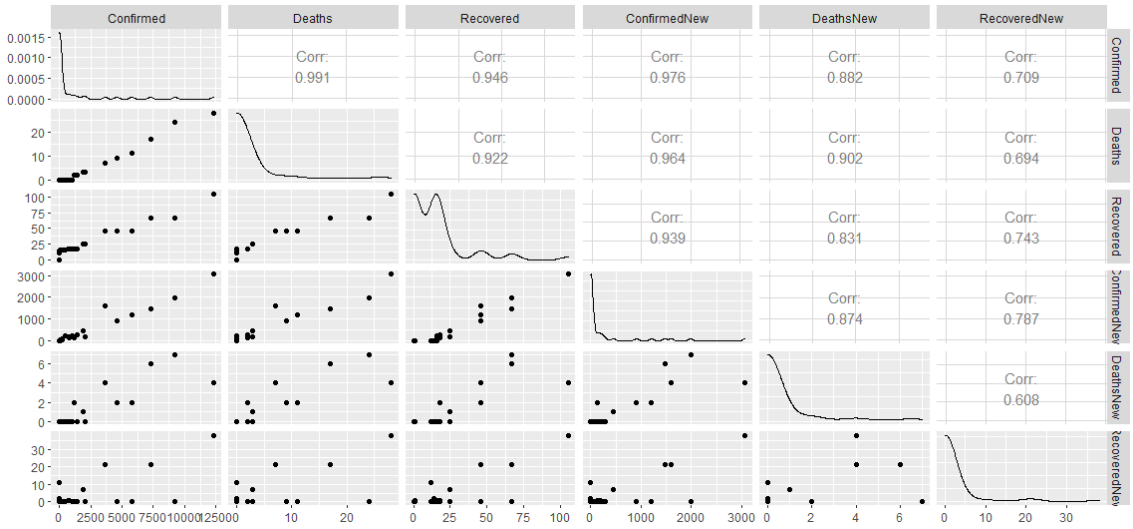
Italy Pairs



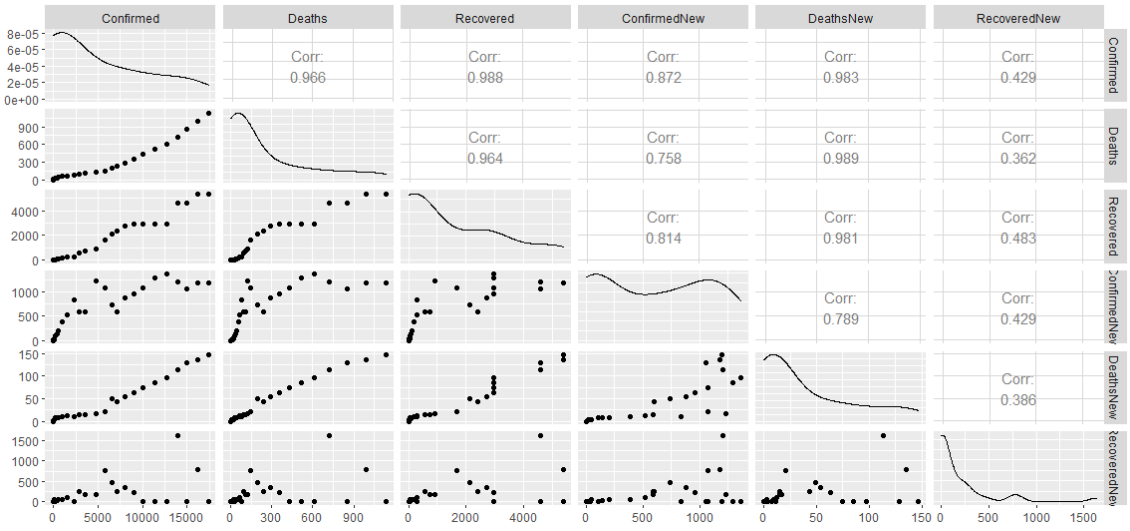
Spain Pairs



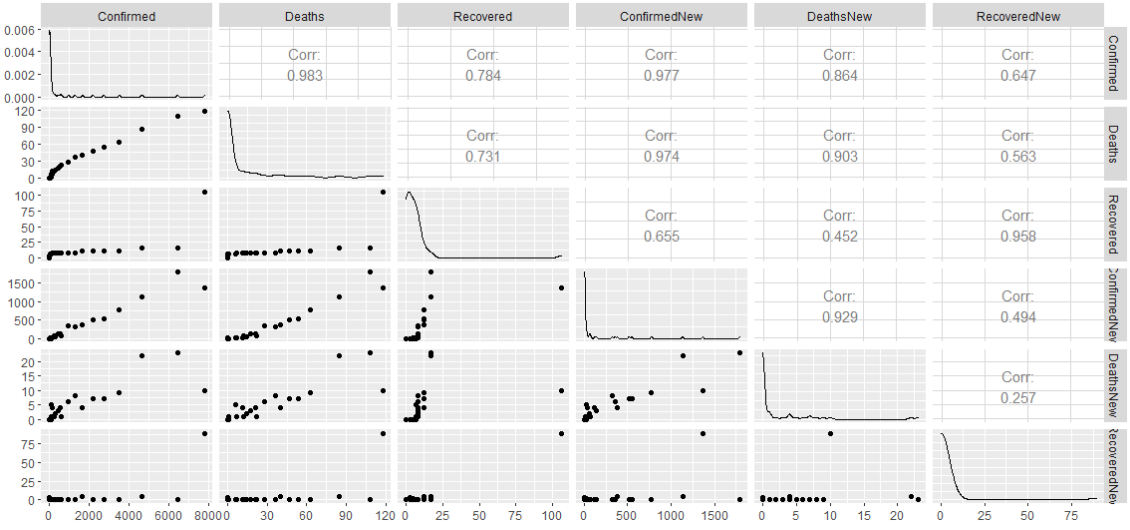
Germany Pairs



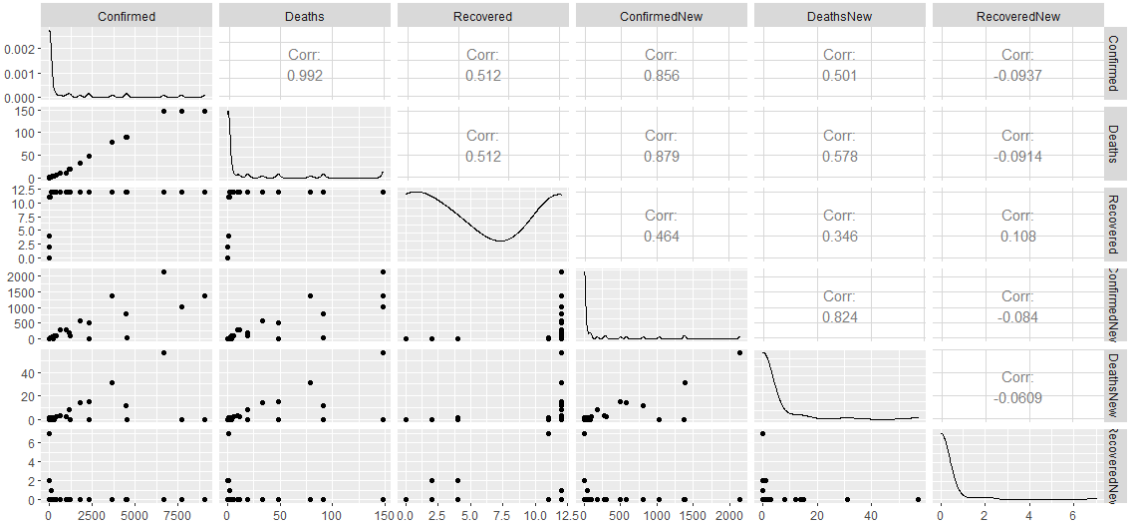
Iran Pairs



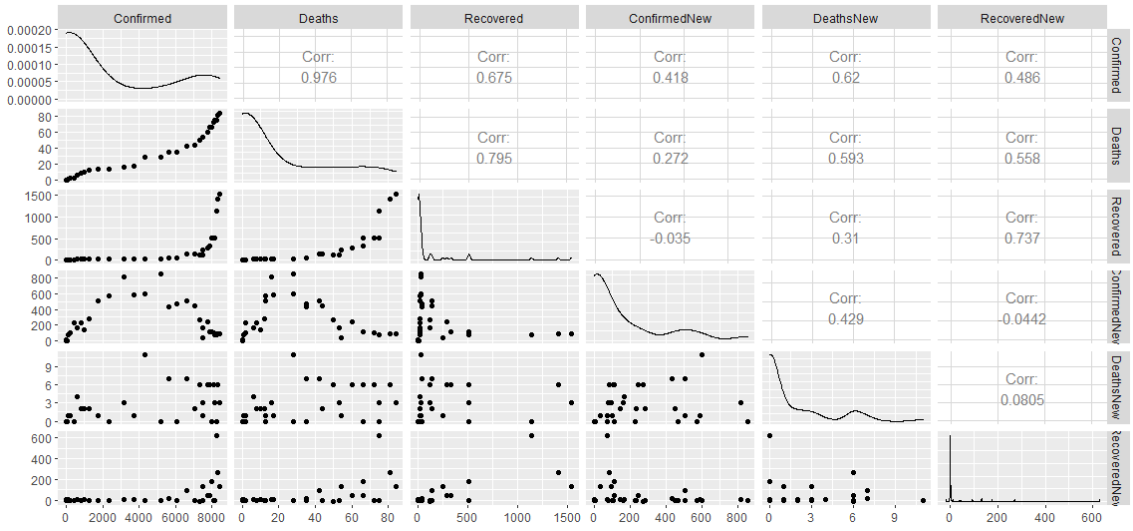
US Pairs



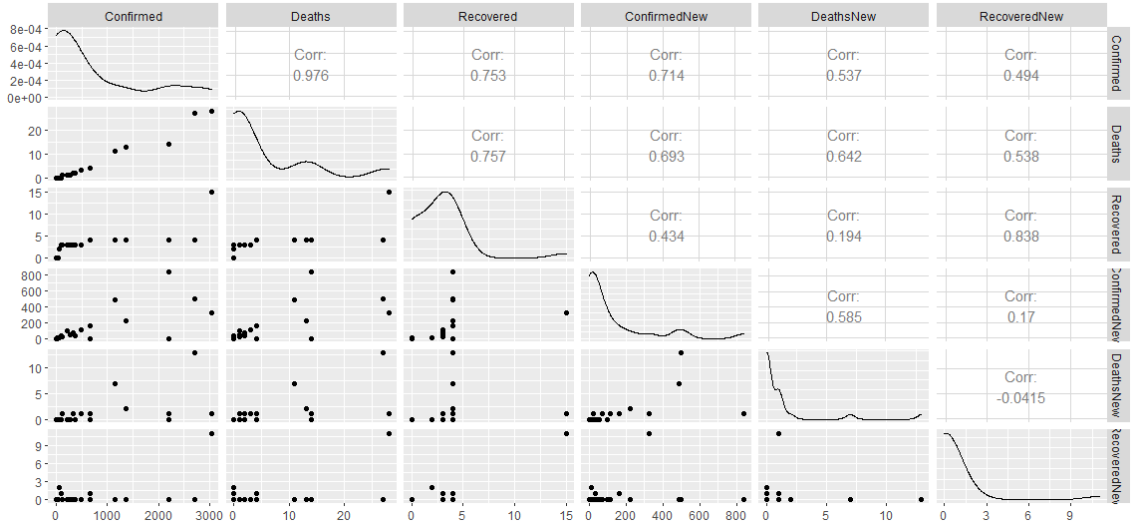
France Pairs



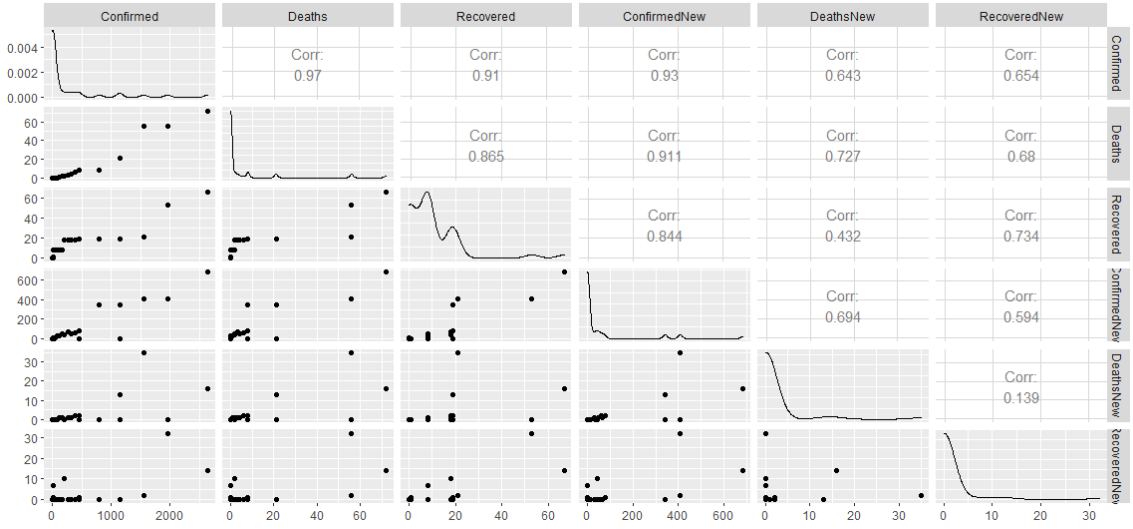
South Korea Pairs

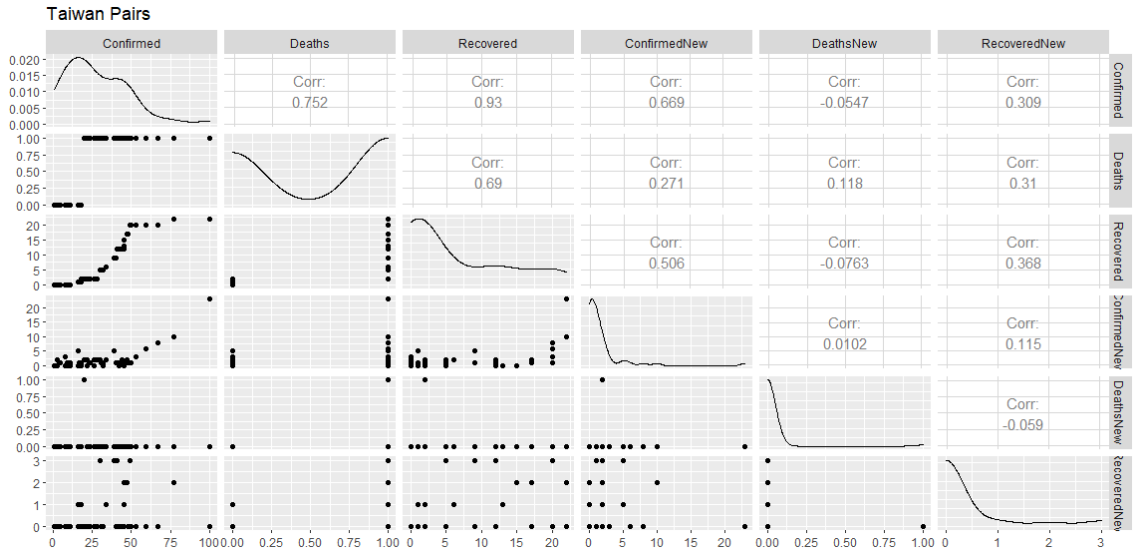


Switzerland Pairs



UK Pairs





According to these pairs, almost all countries have the same shapes except Iran, China, South Korea, and Taiwan's pairs. But the correlation of ConfirmedNew is still good. The worst correlation of ConfirmedNew case is China's pairs. As we can see, the ConfirmedNew has no relationship with Confirmed, Deaths, and Recovered, which is very strange again.

4. Machine Learning

4.1 Multiple linear regression

We try to compare the different trends between the time sequence and the exact date. So we add a sequence to each data.

```
addSeq <- function(data) {
  n <- nrow(data)
  data$seq <- c(1:n)
  data
}

# add sequence
chinaNew <- addSeq(chinaNew)
italyNew <- addSeq(italyNew)
spainNew <- addSeq(spainNew)
germanyNew <- addSeq(germanyNew)
iranNew <- addSeq(iranNew)
usNew <- addSeq(usNew)
franceNew <- addSeq(franceNew)
koreaNew <- addSeq(koreaNew)
switzerlandNew <- addSeq(switzerlandNew)
ukNew <- addSeq(ukNew)
taiwanNew <- addSeq(taiwanNew)
```

We need to combine all the countries' data to draw related plots later, so we add a column **Country** to each dataset.

```
chinaNew$Country <- c("China")
italyNew$Country <- c("Italy")
spainNew$Country <- c("Spain")
germanyNew$Country <- c("Germany")
iranNew$Country <- c("Iran")
usNew$Country <- c("US")
franceNew$Country <- c("France")
koreaNew$Country <- c("South Korea")
switzerlandNew$Country <- c("Switzerland")
ukNew$Country <- c("UK")
taiwanNew$Country <- c("Taiwan")
```

After considering the geographical location, confirmed cases, and better plots, I choose **Iran** as the training data to run the linear regression.

```
summary(iranNew)
```

```
ObservationDate       Confirmed         Deaths          Recovered     ConfirmedNew
Min.   :2020-02-19   Min.   :    2   Min.   :   2.0   Min.   :   0   Min.   :   0.0
1st Qu.:2020-02-26   1st Qu.:  139   1st Qu.:  19.0   1st Qu.:  49   1st Qu.:  44.0
Median :2020-03-04   Median : 2922   Median :  92.0   Median : 552   Median : 591.0
Mean   :2020-03-04   Mean   : 5201   Mean   : 247.1   Mean   :1516   Mean   : 598.6
3rd Qu.:2020-03-11   3rd Qu.: 9000   3rd Qu.: 354.0   3rd Qu.:2959   3rd Qu.:1075.0
Max.   :2020-03-18   Max.   :17361   Max.   :1135.0   Max.   :5389   Max.   :1365.0
   DeathsNew       RecoveredNew      Country             seq
Min.   :  0.00   Min.   :   0.0   Length:29          Min.   : 1
1st Qu.:  4.00   1st Qu.:   0.0   Class :character   1st Qu.: 8
Median : 15.00   Median :  24.0   Mode  :character   Median :15
Mean   : 39.07   Mean   : 185.8                      Mean   :15
3rd Qu.: 63.00   3rd Qu.: 228.0                      3rd Qu.:22
Max.   :147.00   Max.   :1631.0                      Max.   :29
```

```
head(iranNew)
```

```
  ObservationDate Confirmed Deaths Recovered ConfirmedNew DeathsNew RecoveredNew
1      2020-02-19         2      2         0            0         0            0
2      2020-02-20         5      2         0            3         0            0
3      2020-02-21        18      4         0           13         2            0
4      2020-02-22        28      5         0           10         1            0
5      2020-02-23        43      8         0           15         3            0
6      2020-02-24        61     12         0           18         4            0
  Country seq
1    Iran   1
2    Iran   2
3    Iran   3
4    Iran   4
5    Iran   5
6    Iran   6
```

Starting the training, we choose 60% of Iran data as the training set; another 40% is the test set.

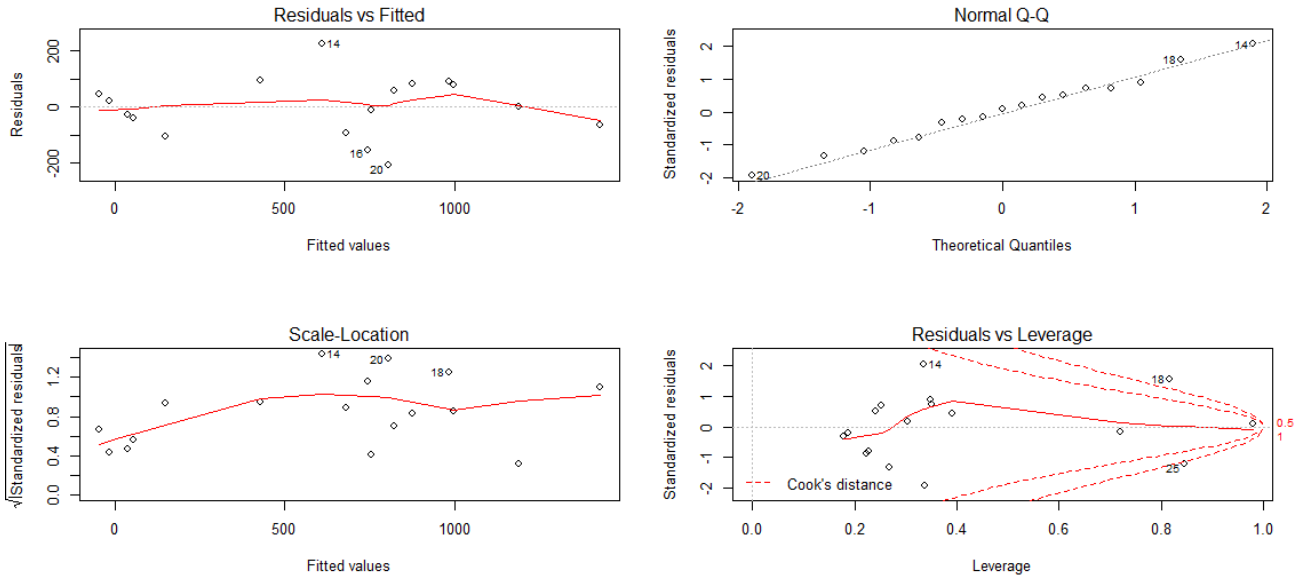
```
n <- nrow(iranNew)
ntrain <- round(n * 0.6)
set.seed(206) # set seed for reproductible results
tindex <- sample(n, ntrain)
trainIran <- iranNew[tindex,] # training set
testIran <- iranNew[-tindex,] # test set

formula <- ConfirmedNew ~ ObservationDate + Confirmed + Deaths +
           Recovered + DeathsNew + RecoveredNew

lm1 <- lm(formula, data=trainIran)
summary(lm1)
```

```
Call:
lm(formula = formula, data = trainIran)

Residuals:
     Min       1Q   Median       3Q      Max
-209.496  -63.841    1.933   78.064  225.178

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -5.199e+05  2.548e+05  -2.041   0.0686 .
ObservationDate  2.839e+01  1.391e+01   2.041   0.0685 .
Confirmed        2.048e-01  7.116e-02   2.878   0.0164 *
Deaths          -1.497e-01  6.507e-01  -0.230   0.8227
Recovered       -3.711e-01  1.349e-01  -2.750   0.0205 *
DeathsNew       -6.418e+00  9.032e+00  -0.711   0.4936
RecoveredNew     1.761e-01  2.852e-01   0.618   0.5507
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 134.3 on 10 degrees of freedom
Multiple R-squared:  0.9467,     Adjusted R-squared:  0.9148
F-statistic: 29.62 on 6 and 10 DF,  p-value: 8.229e-06
```

Only Confirmed and Recovered p-values are close to 0 with one significance star, and the R-squared is 94.67%, which means there are relationships between the predictor and the variables.

We use four plots to view the linear model.

```
par(mfrow=c(2,2))
plot(lm1)
```



The Residuals-vs-Fitted plot shows there is a somewhat linear pattern.

The Normal Q-Q plot shows residuals normally distributed.

The Scale-Location plot shows the residuals are spread almost equally along with the ranges of predictors.

The Residuals-vs-Leverage shows there are some outliers and influential observations.

4.2 Predicting with the dataset

Now we have the training model **lm1**. We can make predictions by using the predict() function for the linear model **lm1**.

```
predictIran <- predict(lm1, newdata = testIran)
cor(predictIran, testIran$ConfirmedNew)
```

```
[1] 0.9846674
```

It shows 98.46%, which means the predicted and the model are positively linearly related. We can say the training model is good. Then we try to predict other countries by Iran's linear regression model.

```
predictChina <- predict(lm1, newdata = chinaNew)
cor(predictChina, chinaNew$ConfirmedNew)
```

```
[1] 0.4457583
# the model is not suited for China
```

```
predictItaly <- predict(lm1, newdata = italyNew)
cor(predictItaly, italyNew$ConfirmedNew)
```

```
[1] 0.7914265
# the model is suited for Italy
```

```
predictSpain <- predict(lm1, newdata = spainNew)
cor(predictSpain, spainNew$ConfirmedNew)
```

```
[1] 0.8147282
# the model is suited for Spain
```

```
predictGermany <- predict(lm1, newdata = germanyNew)
cor(predictGermany, germanyNew$ConfirmedNew)
```

```
[1] 0.8927174
# the model is suited for Germany
```

```
predictUs <- predict(lm1, newdata = usNew)
cor(predictUs, usNew$ConfirmedNew)
```

```
[1] 0.8180316
# the model is suited for US
```

```
predictFrance <- predict(lm1, newdata = franceNew)
cor(predictFrance, franceNew$ConfirmedNew)
```

```
[1] 0.7410682
# the model is suited for France
```

```
predictKorea <- predict(lm1, newdata = koreaNew)
cor(predictKorea, koreaNew$ConfirmedNew)
```

```
[1] 0.5055457
# the model is not suited for South Korea
```

```
predictSwitzerland <- predict(lm1, newdata = switzerlandNew)
cor(predictSwitzerland, switzerlandNew$ConfirmedNew)
```

```
[1] 0.6885858
# the model is suited for Switzerland
```

```
predictUK <- predict(lm1, newdata = ukNew)
cor(predictUK, ukNew$ConfirmedNew)
```

```
[1] 0.6621043
# the model is suited for UK
```

```
predictTaiwan <- predict(lm1, newdata = taiwanNew)
cor(predictTaiwan, taiwanNew$ConfirmedNew)
```

```
[1] 0.4098046
# the model is not suited for Taiwan
```

Almost all countries fit the model except Taiwan, China, and South Korea. I notice that they are all in Asia. Taiwan and South Korea both have the first confirmed case on January 22, 2020, which is earlier than other countries. Moreover, both Taiwan and South Korea are very close to China geographically. I will try to explain these situations later.

4.3 Trend comparison

I compare the trend of new confirmed cases of each country in three different categories.

First, we remove the outlier of China's dataset, which is 15133 new confirmed cases on February 13, 2020, to avoid too wide range in the later plots.

```
chinaNew <- subset(chinaNew, ConfirmedNew < 10000)
```

Then, we combine all countries and transfer the Country column to factor.

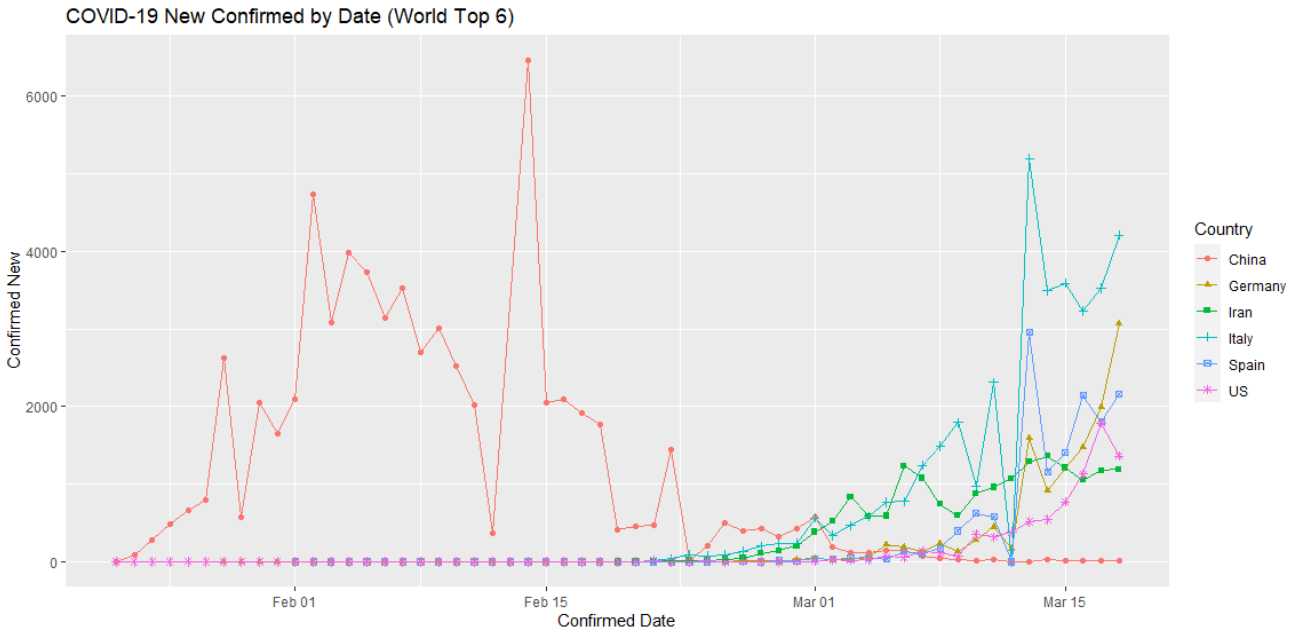
```
allNew <- rbind(chinaNew, italyNew, spainNew, germanyNew, iranNew,
                usNew, franceNew, koreaNew, switzerlandNew, ukNew, taiwanNew)
allNew$Country <- as.factor(allNew$Country)
levels(allNew$Country)
```

```
[1] "China"       "France"      "Germany"     "Iran"    "Italy"   "South Korea"   "Spain"
[8] "Switzerland" "Taiwan"      "UK"          "US"
```

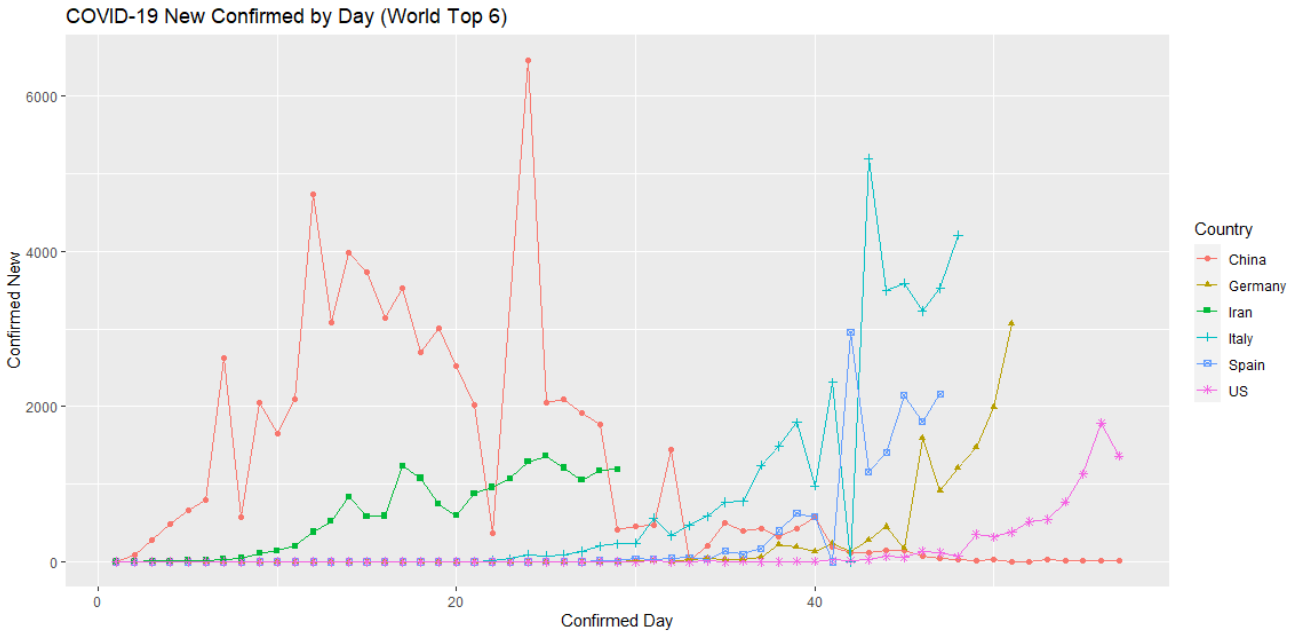
Now the data is ready. We can compare the trends of new confirmed cases between these countries. Because ggplot only accepts six shapes at maximum, so I separate the comparison into three categories with two different variables. The first variable is ObservationDate, which is mapping the exact date when new confirmed cases happened. The second variable is seq, which indicates how many days since the first confirmed cases. Through comparing these two variables, we may found something useful.

Category 1: World Top 6

```
top6 <- allNew[allNew$Country %in% c("China", "France", "Germany", "Iran",
                                     "Italy", "South Korea"),]
ggplot(data=top6,
       mapping=aes(x=ObservationDate, y=ConfirmedNew, shape=Country,
       color=Country)) + geom_point() + geom_line() +
        ggtitle("COVID-19 New Confirmed by Date (World Top 6)") +
        xlab("Confirmed Date") + ylab("Confirmed New")
```




```
ggplot(data=top6,
       mapping=aes(x=seq, y=ConfirmedNew, shape=Country, color=Country)) +
       geom_point() + geom_line() +
       ggtitle("COVID-19 New Confirmed by Day (World Top 6)") +
       xlab("Confirmed Day") + ylab("Confirmed New")
```



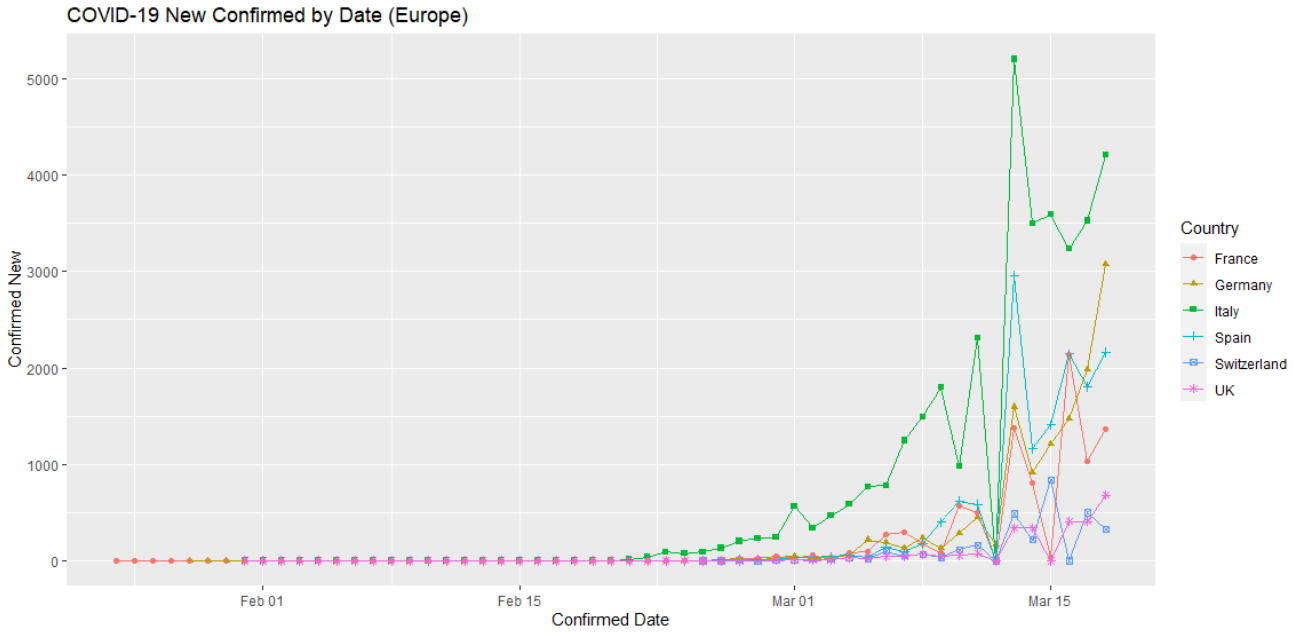
The first plot shows the new confirmed cases by **date**, and the second plot shows the trend by **day**.

In the first plot, we can see that when China's new confirmed cases are going down, the new confirmed cases in other countries are just starting the outbreak of COVID-19. According to the second plot,

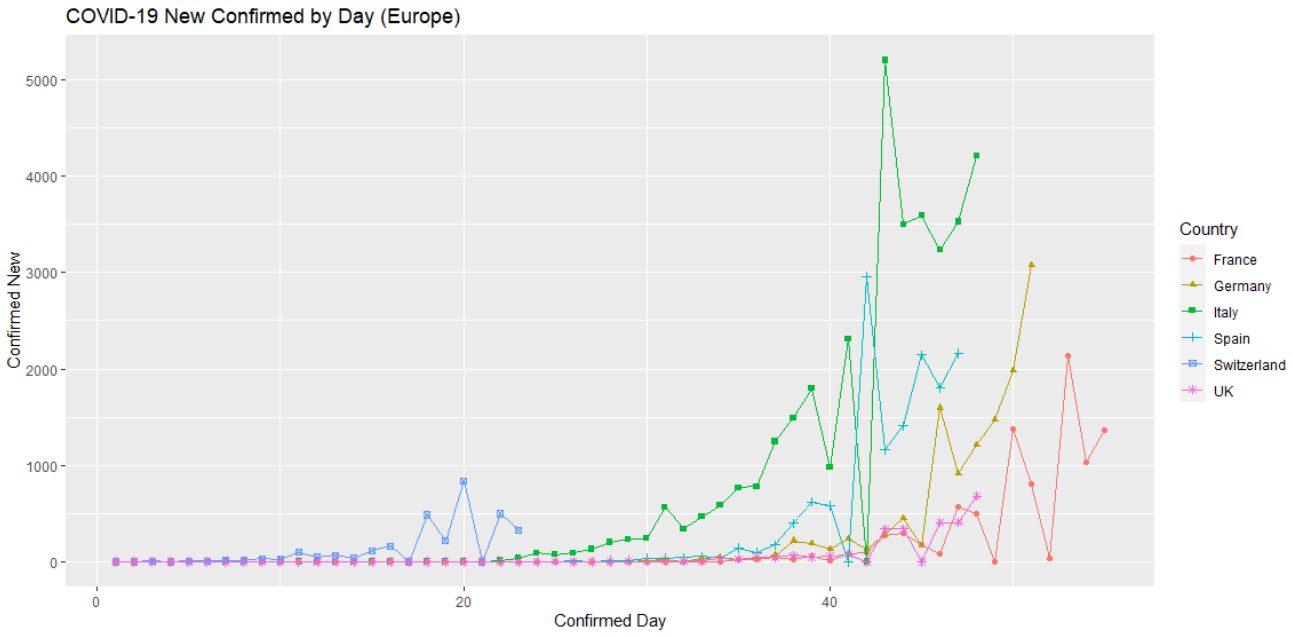
we can see that after 30 days, Italy, Spain, Germany, and the US, they are all increasing the new confirmed cases daily. But for Iran, it is in the increasing trend after just ten days passed. Let's talk about China's data, it kept average above 2000 confirmed cases daily for at least 16 days, but the number drastically down to only double digits in 10 days, which is believed that the report may be fake. Further, there is a super peak 15133 new confirmed cases eliminated from the data to avoid the range is too wide. According to the media and news during the period after the highest peak, Xi Jinping, the president of China, said the disease is under control. After then, the report of new confirmed cases is drastically down.

Category 2: Europe

```
europe <- allNew[allNew$Country %in% c("Spain", "France", "Germany",
                                       "Switzerland", "Italy", "UK"),]
ggplot(data=europe,
       mapping=aes(x=ObservationDate, y=ConfirmedNew, shape=Country,
       color=Country)) + geom_point() + geom_line() +
       ggtitle("COVID-19 New Confirmed by Date (Europe)") +
       xlab("Confirmed Date") + ylab("Confirmed New")
```




```
ggplot(data=europe,
       mapping=aes(x=seq, y=ConfirmedNew, shape=Country, color=Country)) +
       geom_point() + geom_line() +
       ggtitle("COVID-19 New Confirmed by Day (Europe)") +
       xlab("Confirmed Day") + ylab("Confirmed New")
```

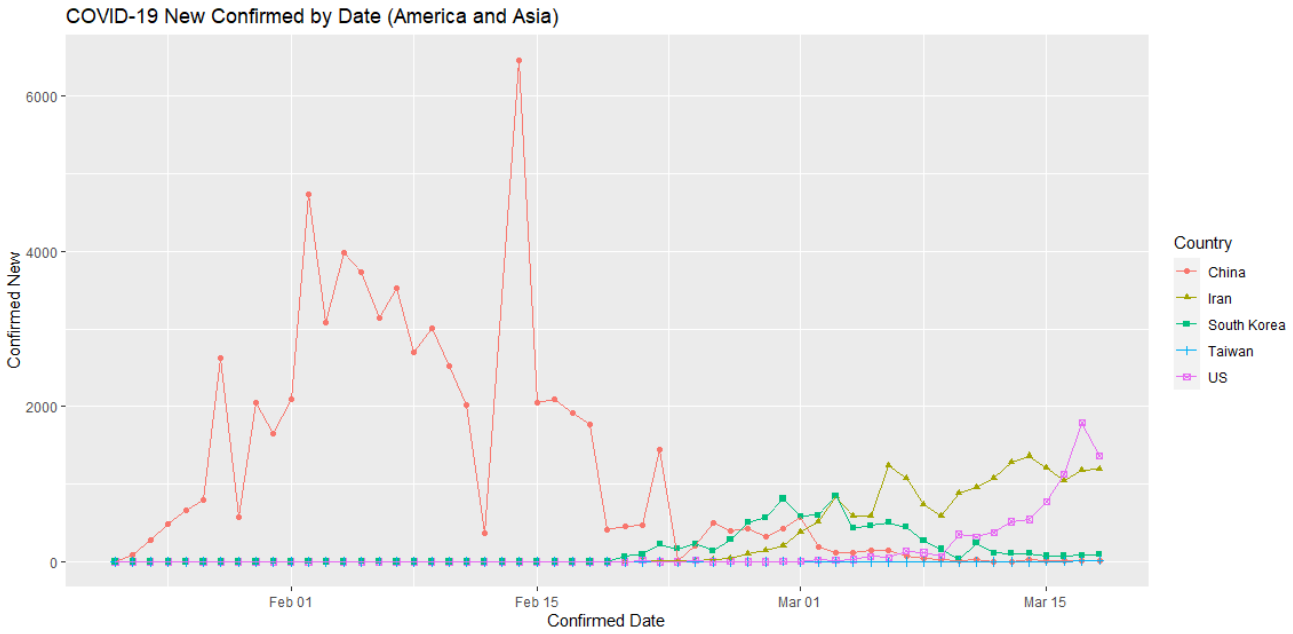


The first plot shows the new confirmed cases by **date,** and the second plot shows the trend by **day**.

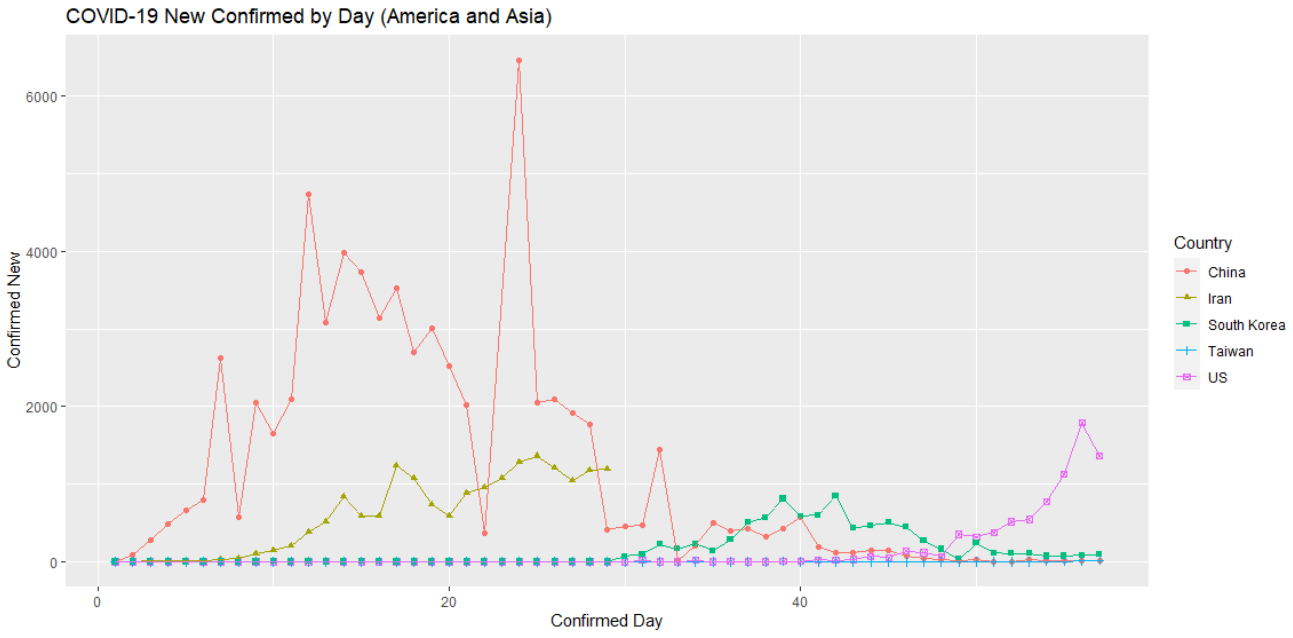
In the first plot, we can see that only Italy has the increasing new confirmed cases in the earlier time, and then almost the same day, all European countries increased new confirmed cases drastically on March 13, 2020. According to the second plot, we realize that the coronavirus just spread in Switzerland for about 23 days, which is fewer than in other countries. The other countries increased a large amount of new confirmed cases after 35~40 days since the first confirmed case.

Category 3: America and Asia

```
americaAsia <- allNew[allNew$Country %in% c("China", "US", "Iran",
                                            "South Korea", "Taiwan"),]
ggplot(data=americaAsia,
       mapping=aes(x=ObservationDate, y=ConfirmedNew, shape=Country,
       color=Country)) + geom_point() + geom_line() +
       ggtitle("COVID-19 New Confirmed by Date (America and Asia)") +
       xlab("Confirmed Date") + ylab("Confirmed New")
```




```
      ggplot(data=americaAsia,
             mapping=aes(x=seq, y=ConfirmedNew, shape=Country, color=Country)) +
        geom_point() + geom_line() + ggtitle("COVID-19 New Confirmed by Day (America and
Asia)") +
        xlab("Confirmed Day") + ylab("Confirmed New")
```



The first plot shows the new confirmed cases by **date,** and the second plot shows the trend by **day**.

In the first plot, we can see that, when the new confirmed cases are going down in China, other countries started to increase the new confirmed cases except Taiwan. According to the second plot, when

other countries had COVID-19 for near 60 days, Iran just happened the disease for less than 30 days. We also notice that China, South Korea, and Taiwan seem to control the outbreak. The trends show that only Iran and the US are still in the increasing trend; other Asian countries are already in the decreasing trend.

Actually, China had the first coronavirus confirmed case in November 2019. Therefore we know the COVID-19 has already existed more than 100 days since then. The peak of the new confirmed case in China is about 70~80 days after the outbreak.

5. Extra thought

5.1 Multiple linear regression

After the above comparison, I wonder what will happen if I choose South Korea as a linear regression model. South Korea is a good sample because it is very close to China and has more cases than Taiwan and fewer cases than China.

```
summary(koreaNew)
```

```
 ObservationDate      Confirmed        Deaths        Recovered       ConfirmedNew
Min.   :2020-01-22   Min.   :   1   Min.   : 0.00   Min.   :   0.0   Min.   :  0.0
1st Qu.:2020-02-05   1st Qu.:  19   1st Qu.: 0.00   1st Qu.:   0.0   1st Qu.:  1.0
Median :2020-02-19   Median :  31   Median : 0.00   Median :  12.0   Median :  7.0
Mean   :2020-02-19   Mean   :2419   Mean   :18.39   Mean   : 130.6   Mean   :147.6
3rd Qu.:2020-03-04   3rd Qu.:5621   3rd Qu.:35.00   3rd Qu.:  41.0   3rd Qu.:229.0
Max.   :2020-03-18   Max.   :8413   Max.   :84.00   Max.   :1540.0   Max.   :851.0
  DeathsNew       RecoveredNew      Country              seq
Min.   : 0.000   Min.   :-17.00   Length:57          Min.   : 1
1st Qu.: 0.000   1st Qu.:  0.00   Class :character   1st Qu.:15
Median : 0.000   Median :  0.00   Mode  :character   Median :29
Mean   : 1.474   Mean   : 27.02                      Mean   :29
3rd Qu.: 2.000   3rd Qu.:  2.00                      3rd Qu.:43
Max.   :11.000   Max.   :627.00                      Max.   :57
```

```
head(koreaNew)
```

```
  ObservationDate Confirmed Deaths Recovered ConfirmedNew DeathsNew RecoveredNew
1      2020-01-22         1      0         0            0         0            0
2      2020-01-23         1      0         0            0         0            0
3      2020-01-24         2      0         0            1         0            0
4      2020-01-25         2      0         0            0         0            0
5      2020-01-26         3      0         0            1         0            0
6      2020-01-27         4      0         0            1         0            0
      Country seq
1 South Korea   1
2 South Korea   2
3 South Korea   3
4 South Korea   4
5 South Korea   5
6 South Korea   6
```

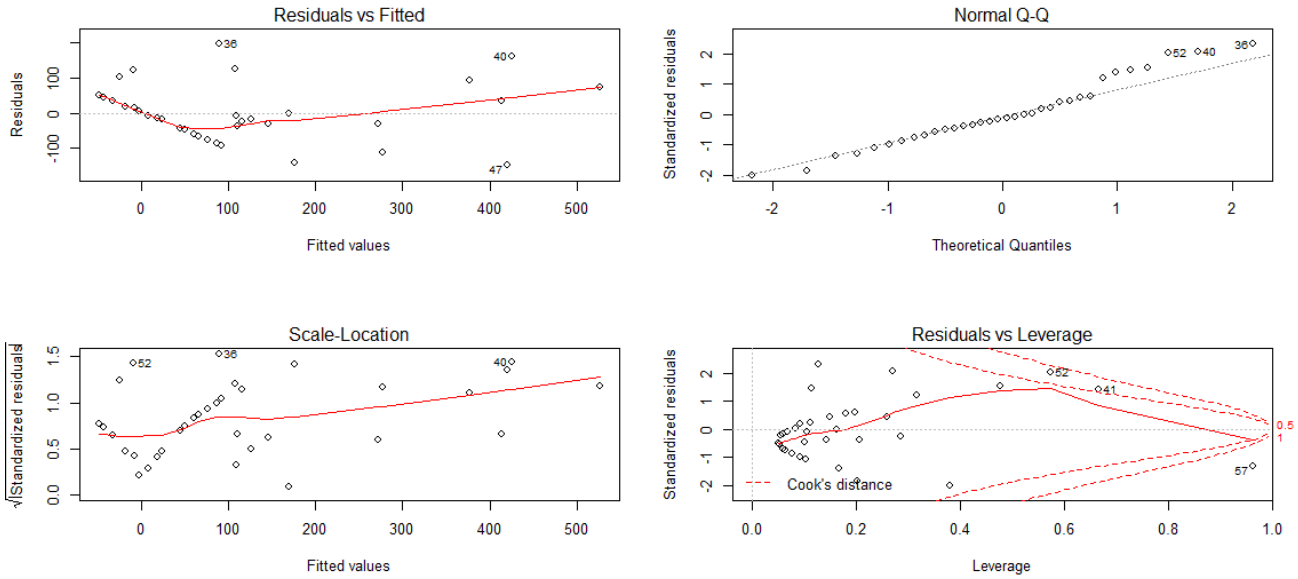
Perform linear regression and training.

```
n <- nrow(koreaNew)
ntrain <- round(n * 0.6)
set.seed(206) # set seed for reproductible results
tindex <- sample(n, ntrain)
trainKorea <- koreaNew[tindex,] # training set
testKorea <- koreaNew[-tindex,] # test set

formula <- ConfirmedNew ~ ObservationDate + Confirmed + Deaths +
           Recovered + DeathsNew + RecoveredNew
lm2 <- lm(formula, data=trainKorea)
summary(lm2)
```

```
Call:
lm(formula = formula, data = trainKorea)

Residuals:
    Min      1Q  Median      3Q     Max
-147.84  -46.84  -11.93   42.02  194.53

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -8.793e+04  3.738e+04  -2.352  0.02621 *
ObservationDate  4.806e+00  2.043e+00   2.353  0.02616 *
Confirmed        1.964e-01  3.283e-02   5.982 2.22e-06 ***
Deaths          -2.825e+01  4.704e+00  -6.006 2.08e-06 ***
Recovered        4.000e-01  1.343e-01   2.978  0.00607 **
DeathsNew        2.865e+01  8.467e+00   3.384  0.00220 **
RecoveredNew    -6.087e-01  5.342e-01  -1.139  0.26457
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 89.6 on 27 degrees of freedom
Multiple R-squared:  0.782,      Adjusted R-squared:  0.7335
F-statistic: 16.14 on 6 and 27 DF,  p-value: 8.396e-08
```

The R-squared is 78.2%, which is good. The p-values are good too, and there are many significant stars in response variables. Let’s see some diagnostic plots provided by the linear model.

```
par(mfrow=c(2,2))
plot(lm2)
```



The Residuals-vs-Fitted plot shows there is a somewhat linear pattern, with some labeled outliers.

The Normal Q-Q plot shows residuals almost normally distributed

The Scale-Location plot shows some residuals are spread equally along with the ranges of predictors, but some outliners exist.

The Residuals-vs-Leverage shows that there are some outliers and influential observations. Points with a large Cook's distance may distort the outcome and accuracy of a regression.

5.2 Predicting with the dataset

Now we have the training model **lm2**. We can make predictions by using the predict() function for the linear model **lm2**.

```
predictKorea <- predict(lm2, newdata = testKorea)
cor(predictKorea, testKorea$ConfirmedNew)
```

```
[1] 0.8181736
# the model is quite ok, strong
```

It shows 81.82%, which means the predicted and the model are positively linearly related. We can say the training model is well. Then we try to predict other countries by South Korea's linear regression model.

```
predictKorea <- predict(lm2, newdata = testKorea)
cor(predictKorea, testKorea$ConfirmedNew)
```

```
[1] 0.8181736
# the model is quite ok, strong
```

```
predictChina <- predict(lm2, newdata = chinaNew)
cor(predictChina, chinaNew$ConfirmedNew)
```

```
[1] 0.555154
# the model is not suited for China, moderate
```

```
predictItaly <- predict(lm2, newdata = italyNew)
cor(predictItaly, italyNew$ConfirmedNew)
```

```
[1] -0.8485371
# the model is suited for Italy, very strong
```

```
predictIran <- predict(lm2, newdata = iranNew)
cor(predictIran, iranNew$ConfirmedNew)
```

```
[1] -0.7147556
# the model is suited for Iran, strong
```

```
predictSpain <- predict(lm2, newdata = spainNew)
cor(predictSpain, spainNew$ConfirmedNew)
```

```
[1] -0.7331091
# the model is suited for Spain, strong
```

```
predictGermany <- predict(lm2, newdata = germanyNew)
cor(predictGermany, germanyNew$ConfirmedNew)
```

```
[1] 0.9622448
# the model is suited for Germany, very strong
```

```
predictUs <- predict(lm2, newdata = usNew)
cor(predictUs, usNew$ConfirmedNew)
```

```
[1] -0.9147847
# the model is suited for USy, very strong
```

```
predictFrance <- predict(lm2, newdata = franceNew)
cor(predictFrance, franceNew$ConfirmedNew)
```

```
[1] -0.6731894
# the model is suited for France, moderate
```

```
predictSwitzerland <- predict(lm2, newdata = switzerlandNew)
cor(predictSwitzerland, switzerlandNew$ConfirmedNew)
```

```
[1] 0.5597923
# the model is not suited for Switzerland, moderate
```

```
predictUK <- predict(lm2, newdata = ukNew)
cor(predictUK, ukNew$ConfirmedNew)
```

```
[1] -0.6594296
# the model is suited for UK, moderate
```

```
predictTaiwan <- predict(lm2, newdata = taiwanNew)
cor(predictTaiwan, taiwanNew$ConfirmedNew)
# [1] 0.4427289
# the model is not suited for Taiwan, poor
```

The result is very shocking. We can see that China and Switzerland have about 55% correlation with the linear model. However, Taiwan has the lowest 44% correlation with the linear model.

5.3 Linear regression result comparison

The linear models trained by Iran and South Korea all suited for European and American countries in our samples. The only two exceptions are China and Taiwan. As we all know, China government ignored and hid the disease at the very beginning time, and then it blocked the information from the world. In the end, the outbreak is boomed in Wuhan city. The confirmed cases and deaths suddenly increased a lot. After that, President Xi Jinping controlled the media, news, and new confirmed cases. It is believed that the government may report the fake data to WHO when after one of Xi's talk to the citizens in China.

Taiwan is in the other situation. Taiwan suffered SARS in 2003. Since then, Taiwan has always prepared for another disease outbreak. So when Taiwan first heard there might be new coronavirus appeared in Wuhan in the early of January 2020, the government soon decided to build the Central Epidemic Command Center (CECC) to manage all the information about COVID-19. CECC arranged many policies to prevent the disease from spreading and monitor people who may be in danger. It turns out to make the incredible few confirmed cases in Taiwan, although Taiwan is the closest country to China in the world. Do not forget that there are two to three million people fly to and back between China and Taiwan every year.

According to the above reason, other countries are not so aware of COVID-19 or China like Taiwan. Even the WHO also declared that this disease is not dangerous. So other countries will not prepare the outbreak because they believe WHO. But people in Taiwan never believe China and know that China actually controls the WHO. Hence, Taiwan prepared for it in advance. The population of Taiwan is 23 million people; meanwhile, the people of South Korea are 51 million. Compare to the only 100 confirmed cases in Taiwan, and there are 8413 confirmed cases in South Korea, it significantly showed why I said Taiwan truly prevents the COVID-19 from spreading.

These explanations tell us why the linear models trained by Iran and South Korea do not suit for China and Taiwan but suit other countries. The reasons are China's data is not precise, and Taiwan's situation is under control without any outbreak. We know that the first confirmed case in Taiwan is the same day as the

first confirmed case in South Korea and Japan. But it turns out that South Korea has 8413 confirmed cases, and Japan has 889 confirmed cases, which both are the severe COVID-19 affected area.

6. Conclusion

According to the trends and plots showed above, the top 10 confirmed cases countries have a similar situation since the disease appeared in those countries. The only exception is China in the top 10. By these comparisons, they showed the peak of the new confirmed case would occur after 30 ~ 40 days since the first confirmed case. The linear models produced by Iran and South Korea are almost perfectly suited for other countries. This means the COVID-19 outbreak has the same pattern in almost any country except China and Taiwan. For Taiwan, because it prepares and controls in advance before WHO's announcement. For China, its strange patterns about the disease are obviously controlled by the communist government. Until now, the new confirmed cases are still not transparent in China.