Predict the Spreading of Coronavirus

Task Details

The outbreak of Covid-19 is developing into a major international crisis, and it's starting to influence important aspects of daily life. For example:

- Travel: Bans have been placed on hotspot countries, corporate travel has been reduced, and flight fares have dropped.
- Supply chains: International manufacturing operations have often had to throttle back production and many goods solely produced in China have been halted altogether.
- Grocery stores: In highly affected areas, people are starting to stock up on essential goods.

A strong model that predicts how the virus could spread across different countries and regions may be able to help mitigation efforts. The goal of this task is to build a model that predicts the progression of the virus throughout March 2020.

Data file link: https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset/download

#---#

Import Data file into R environment:

CDH <- read.csv(file = "covid_19_data.csv", header = TRUE,na.strings=c("","NA"))

Data1 <- CDH

summary(CDH)

```
> summary(CDH)
      SNo          ObservationDate                         Province.State
 Min.   :    1    04-10-2020:  321    Diamond Princess cruise ship: 127
 1st Qu.: 4662    04-11-2020:  321    Gansu                       :  97
 Median : 9324    04-06-2020:  320    Hebei                       :  97
 Mean   : 9324    04-07-2020:  320    Anhui                       :  95
 3rd Qu.:13985    04-08-2020:  320    Beijing                     :  95
 Max.   :18646    04-09-2020:  320    (Other)                     :8677
                  (Other)   :16724    NA's                        :9458
        Country.Region       Last.Update        Confirmed            Deaths
 US            :3598   03-08-2020: 1232   Min.   :     0   Min.   :    0.0
 Mainland China:2943   10-04-2020:  321   1st Qu.:    10   1st Qu.:    0.0
 Canada        : 741   11-04-2020:  321   Median :   103   Median :    1.0
 Australia     : 596   04-06-2020:  320   Mean   :  3134   Mean   :  188.5
 France        : 488   07-04-2020:  320   3rd Qu.:   700   3rd Qu.:    8.0
 UK            : 442   08-04-2020:  320   Max.   :282143   Max.   :26384.0
 (Other)       :9838   (Other)   :15812
   Recovered
 Min.   :     0.0
 1st Qu.:     0.0
 Median :     2.0
 Mean   :   795.2
 3rd Qu.:    73.0
 Max.   :109800.0
```

Data file:

CDH × | COVID19.R* × | LD × | Result2 × | Actual × | Data1 ×

Filter

	SNo	ObservationDate	Province.State	Country.Region	Last.Update	Confirmed	Deaths	Recovered
1	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1	0	0
2	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14	0	0
3	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6	0	0
4	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1	0	0
5	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0	0	0
6	6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26	0	0
7	7	01/22/2020	Guangxi	Mainland China	1/22/2020 17:00	2	0	0
8	8	01/22/2020	Guizhou	Mainland China	1/22/2020 17:00	1	0	0
9	9	01/22/2020	Hainan	Mainland China	1/22/2020 17:00	4	0	0

Showing 1 to 10 of 18,646 entries, 8 total columns

Summary of Data set:

```
      SNo           ObservationDate                          Province.State
 Min.   :    1    04-10-2020:   321   Diamond Princess cruise ship: 127
 1st Qu.: 4662    04-11-2020:   321   Gansu                        :  97
 Median : 9324    04-06-2020:   320   Hebei                        :  97
 Mean   : 9324    04-07-2020:   320   Anhui                        :  95
 3rd Qu.:13985    04-08-2020:   320   Beijing                      :  95
 Max.   :18646    04-09-2020:   320   (Other)                      :8677
                  (Other)    :16724   NA's                         :9458
         Country.Region       Last.Update        Confirmed            Deaths
 US                 :3598   03-08-2020: 1232   Min.   :     0   Min.   :    0.0
 Mainland China:2943   10-04-2020:   321   1st Qu.:    10   1st Qu.:    0.0
 Canada             : 741   11-04-2020:   321   Median :   103   Median :    1.0
 Australia          : 596   04-06-2020:   320   Mean   :  3134   Mean   :  188.5
 France             : 488   07-04-2020:   320   3rd Qu.:   700   3rd Qu.:    8.0
 UK                 : 442   08-04-2020:   320   Max.   :282143   Max.   :26384.0
 (Other)            :9838   (Other)    :15812
   Recovered
 Min.   :      0.0
 1st Qu.:      0.0
 Median :      2.0
 Mean   :    795.2
 3rd Qu.:     73.0
 Max.   :109800.0
```

Analysis of Data type in Data file:

```
> str(Data1)
'data.frame':    18646 obs. of  8 variables:
 $ SNo            : int  1 2 3 4 5 6 7 8 9 10 ...
 $ ObservationDate: Factor w/ 95 levels "01/22/2020","01/23/2020",..: 1 1 1 1 1 1 1 1 1 1
 ...
 $ Province.State : chr  "Anhui" "Beijing" "Chongqing" "Fujian" ...
 $ Country.Region : Factor w/ 220 levels " Azerbaijan",..: 122 122 122 122 122 122 122 122
 122 122 ...
 $ Last.Update    : Factor w/ 1812 levels "01-04-2020","02-01-2020",..: 12 12 12 12 12 12 1
2 12 12 12 ...
 $ Confirmed      : int  1 14 6 1 0 26 2 1 4 1 ...
 $ Deaths         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Recovered      : int  0 0 0 0 0 0 0 0 0 0 ...
>
```

Reset the Date format in Data set:

```
Data1$ObservationDate<- as.factor(Data1$ObservationDate)
Data1$Last.Update<- as.factor(Data1$Last.Update)
summary(Data1)
View(Data1)
str(Data1)
```

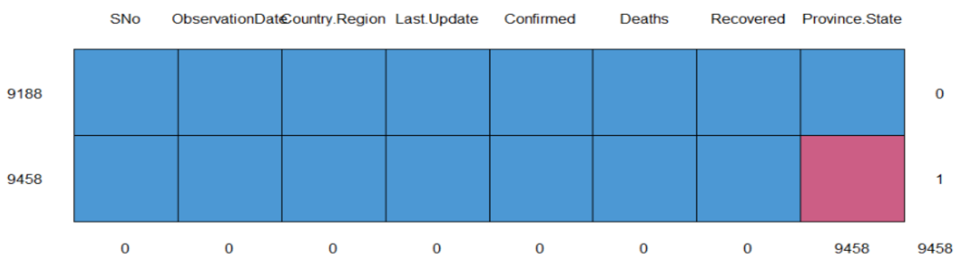
Check NA values in Data set:

```
#----------------------------------------------------------------------------#
P<- function(X)
{ sum(is.na(X))/ length(X)*100}

apply(CDH, 2,P)

library(mice)
md.pattern(CDH)
md.pairs(CDH)
```

Result: *50.72% NA values present under Province State column.*

```
> P<- function(X)
+ { sum(is.na(X))/ length(X)*100}
> apply(CDH, 2,P)
            SNo ObservationDate  Province.State  Country.Region     Last.Update
        0.00000         0.00000        50.72402         0.00000         0.00000
      Confirmed          Deaths       Recovered
        0.00000         0.00000         0.00000
> library(mice)
> md.pattern(CDH)
     SNo ObservationDate Country.Region Last.Update Confirmed Deaths Recovered
9188   1               1              1           1         1      1         1
9458   1               1              1           1         1      1         1
       0               0              0           0         0      0         0
     Province.State
9188              1    0
9458              0    1
               9458 9458
> |
```

Graphical representation of NA values:



Replace NA values with"other_region" of respective state name.

```
#replace NA data with country of respective Province state.

Data1[is.na(Data1$Province.State)]
Data1$Province.State<- as.character(Data1$Province.State)
Data1$Province.State[(Data1$Province.State == " ")] <- NA

Data1$Province.State[which(is.na(Data1$Province.State))]<-'other_region'


View(Data1)
summary(Data1)
```

Result: *NA value replaced with "Other_region"*

```
> Data1[is.na(Data1$Province.State)]
data frame with 0 columns and 18646 rows
> Data1$Province.State<- as.character(Data1$Province.State)
> Data1$Province.State[(Data1$Province.State == " ")] <- NA
> Data1$Province.State[which(is.na(Data1$Province.State))]<-'other_region'
>
```

	SNo	ObservationDate	Province.State	Country.Region	Last.Update	Confirmed	Deaths	Recovered
68	68	01/23/2020	Tianjin	Mainland China	1/23/20 17:00	4	0	0
69	69	01/23/2020	Tibet	Mainland China	1/23/20 17:00	0	0	0
70	70	01/23/2020	Washington	US	1/23/20 17:00	1	0	0
71	71	01/23/2020	Xinjiang	Mainland China	1/23/20 17:00	2	0	0
72	72	01/23/2020	Yunnan	Mainland China	1/23/20 17:00	2	0	0
73	73	01/23/2020	Zhejiang	Mainland China	1/23/20 17:00	27	0	0
74	74	01/23/2020	other_region	Japan	1/23/20 17:00	1	0	0
75	75	01/23/2020	other_region	Thailand	1/23/20 17:00	3	0	0
76	76	01/23/2020	other_region	South Korea	1/23/20 17:00	1	0	0
77	77	01/23/2020	other_region	Singapore	1/23/20 17:00	1	0	0
78	78	01/23/2020	other_region	Philippines	1/23/20 17:00	0	0	0
79	79	01/23/2020	other_region	Malaysia	1/23/20 17:00	0	0	0
80	80	01/23/2020	other_region	Vietnam	1/23/20 17:00	2	0	0
81	81	01/23/2020	other_region	Australia	1/23/20 17:00	0	0	0
82	82	01/23/2020	other_region	Mexico	1/23/20 17:00	0	0	0
83	83	01/23/2020	other_region	Brazil	1/23/20 17:00	0	0	0

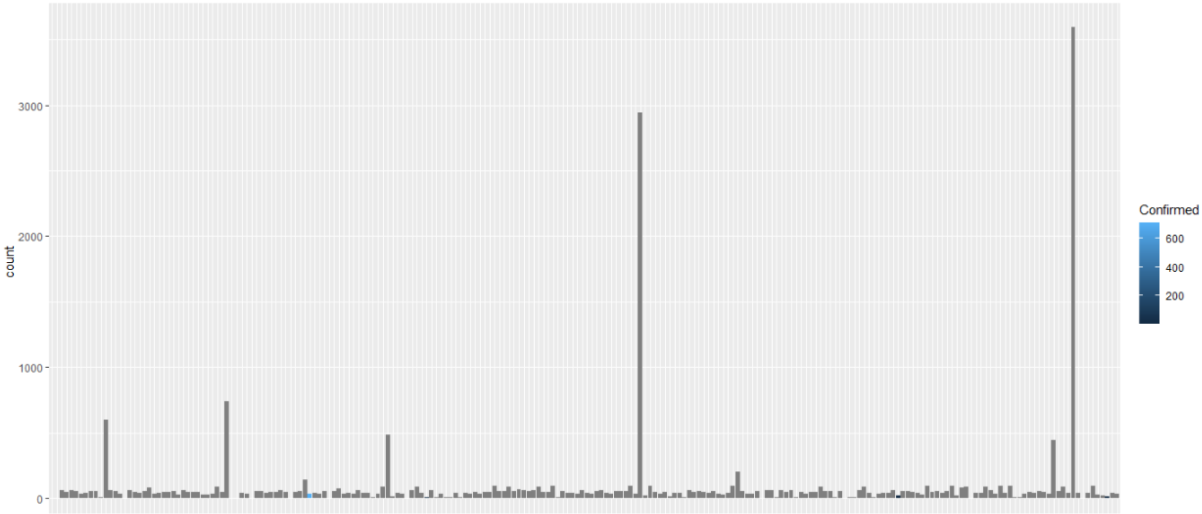
ng 72 to 89 of 18,646 entries, 8 total columns

Graphical representation of confirm cases country wise (128+ countries in Data set)

```
#graphical representtion country vise ..
class(Data1)
Data1<- as.data.frame(Data1)
#confirmed cases country vise:

library(ggplot2)

ggplot(Data1, aes(x= Country.Region, fill = Confirmed))+ geom_bar()

summary(Data1)

ggplot(Data1, aes(x = Deaths, y =Confirmed ))+ geom_point(colour =Data1$Country.Region)

ggplot(Data1, aes(x= Country.Region, fill = Confirmed))+ geom_boxplot()

head (ggplot(Data1, aes(x = Deaths, y = Last.Update)) + geom_point())
```

Results : *total 128 countries listed in below bar plot*

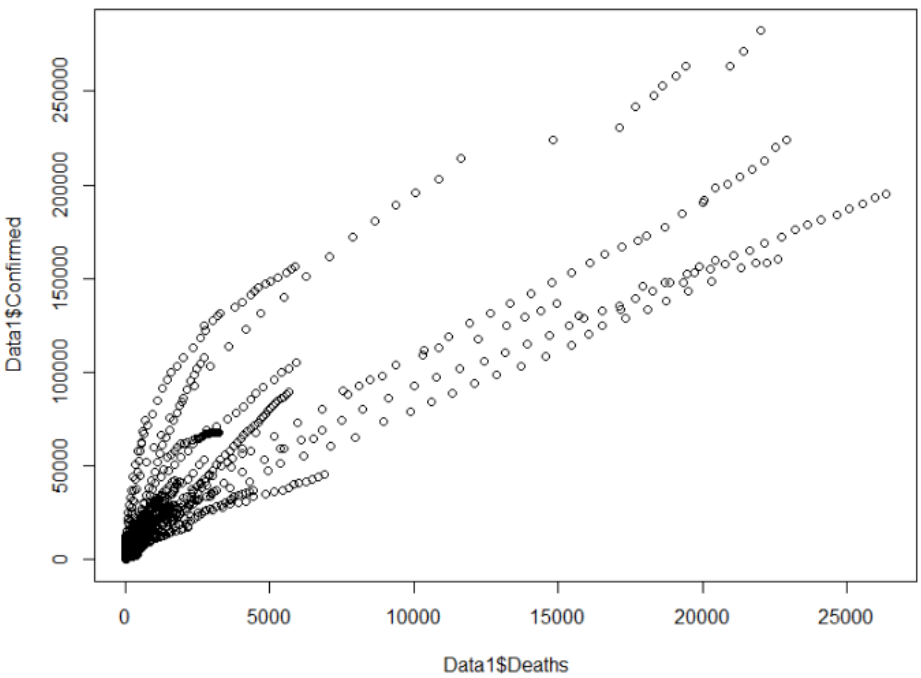
Highest one is USA



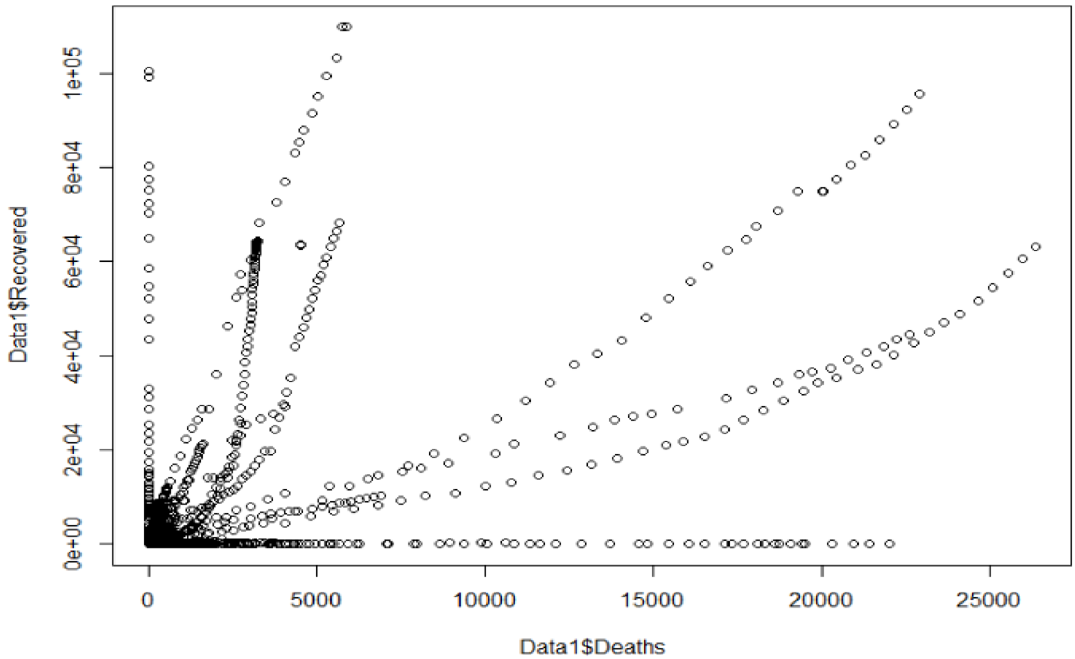

```
plot(Data1$Deaths,Data1$Confirmed )
plot(Data1$Deaths, Data1$Recovered)
plot(Data1$Confirmed, Data1$Recovered)
```

.

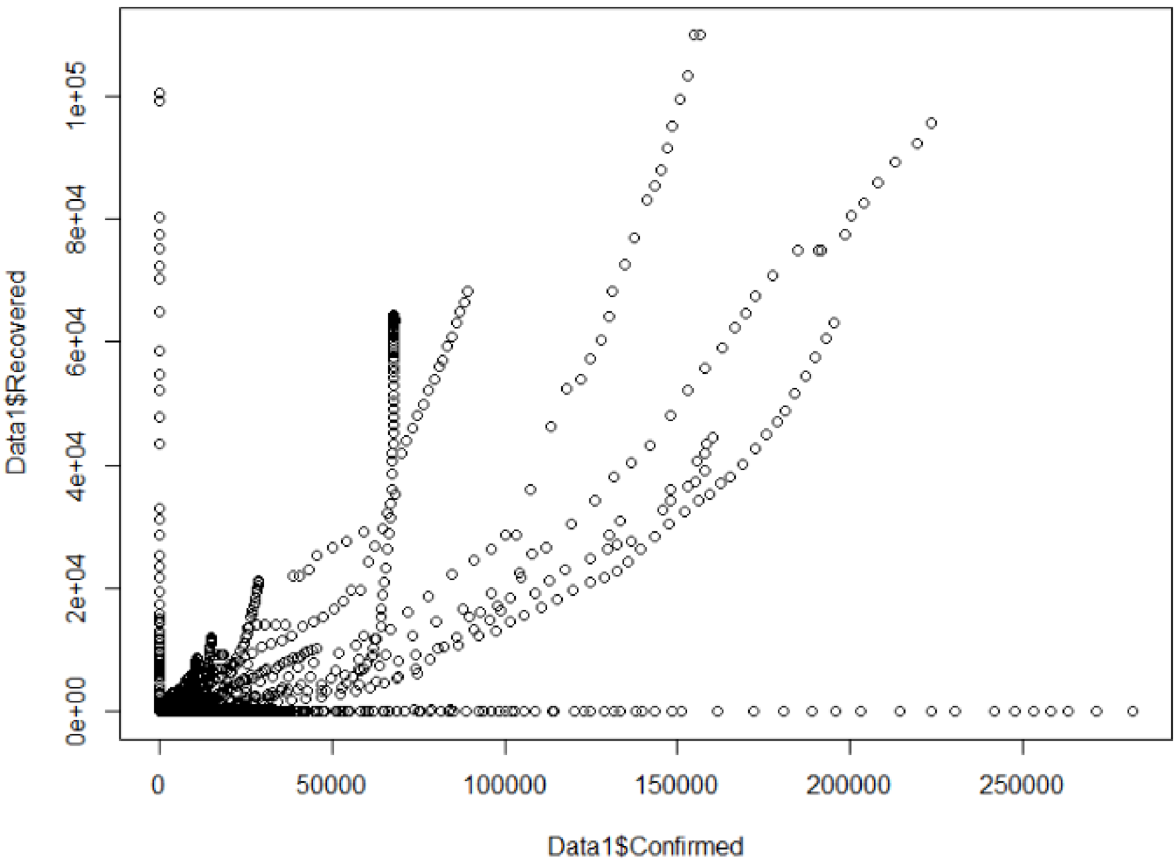
Plot: confirmed cases VS Deaths



Plot: Recovered Vs. Deaths cases

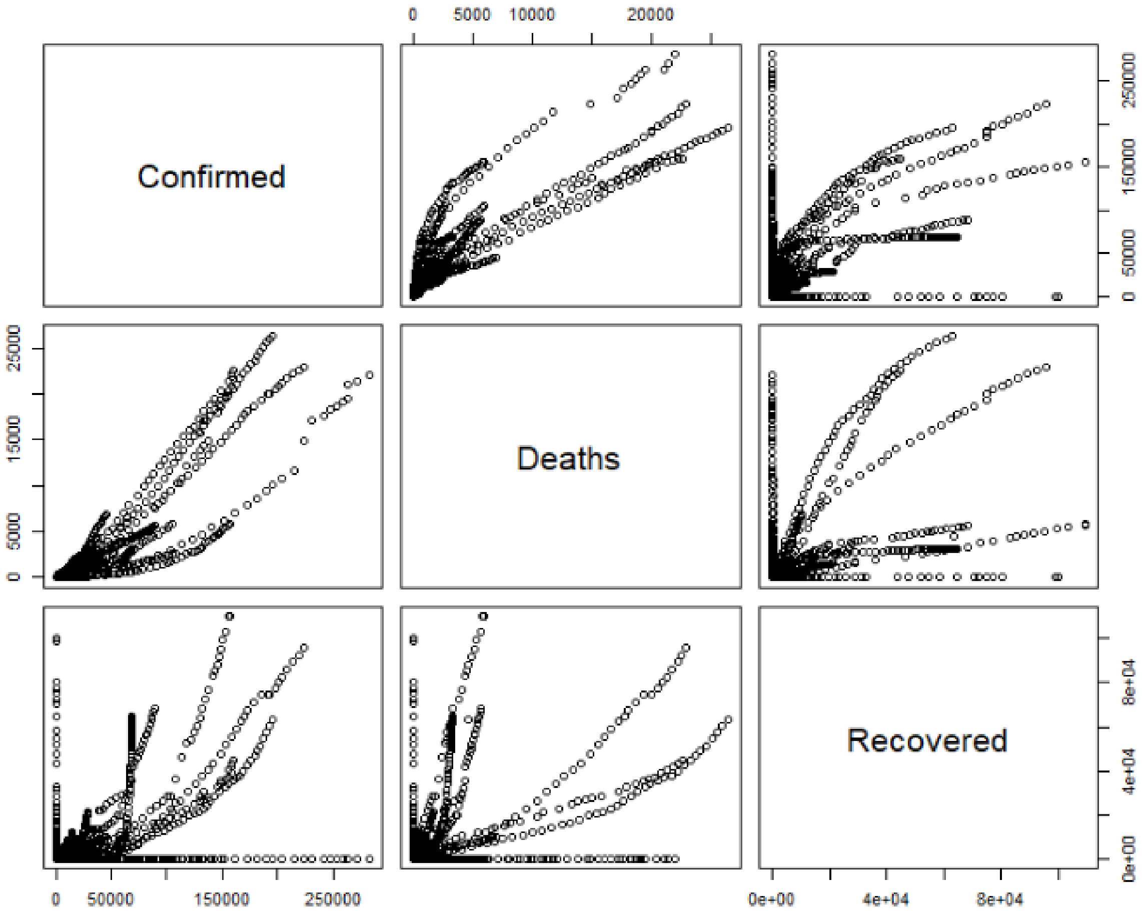


Plot: Confirmed Vs. Recovered cases



```
View(Data1)

pairs(Data1[6:8])
```

Plot: Confirmed VS Deaths VS Recovered Cases



 # Building linear model for the Confirmed cases vs. Recovered and Deaths

```
# building linear model for the Confirmed cases vs Recovered and Deaths
library(caTools)

St <- sample.split(Data1$Confirmed, SplitRatio = 0.60)
Train<- subset(Data1, St == T)
Test <- subset(Data1 , St == F)
nrow(Train)
nrow(Test)

#Model Confirm vs Recovered

Model_conf_rec <- lm(Confirmed~Recovered, data = Train)
summary(Model_conf_rec)
#p-value: < 2.2e-16,   Multiple R-squared:  0.454,
1-2.2e-16
```

We sampled the Data in to Test and Train with 60% sampling ratio.

Summary of linear model [confirmed vs. Recovered]

```
Call:
lm(formula = Confirmed ~ Recovered, data = Train)

Residuals:
    Min      1Q  Median      3Q     Max
-190736   -2534   -2502   -1912  279608

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.535e+03  1.299e+02   19.51   <2e-16 ***
Recovered   1.875e+00  1.917e-02   97.83   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14120 on 12106 degrees of freedom
Multiple R-squared:  0.4415,	Adjusted R-squared:  0.4415
F-statistic:  9571 on 1 and 12106 DF,  p-value: < 2.2e-16
```

Predict the results with Test Data:

```
Result1 <- predict(Model_conf_rec, newdata = Test)

Result1
```

Model has predicted WRT Test Data set (inputs)

```
> Result1 <- predict(Model_conf_rec, newdata = Test)
> Result1
       1        3        4        7        9       11       13       14
2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2587.482
      16       17       19       20       21       22       23       24
2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981
      25       28       31       32       33       35       36       37
2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981
      39       41       43       44       47       49       51       53
2534.981 2534.981 2534.981 2538.731 2534.981 2534.981 2534.981 2534.981
      54       56       57       59       60       61       62       63
2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981 2534.981
```

 # find our Error values and RMS :

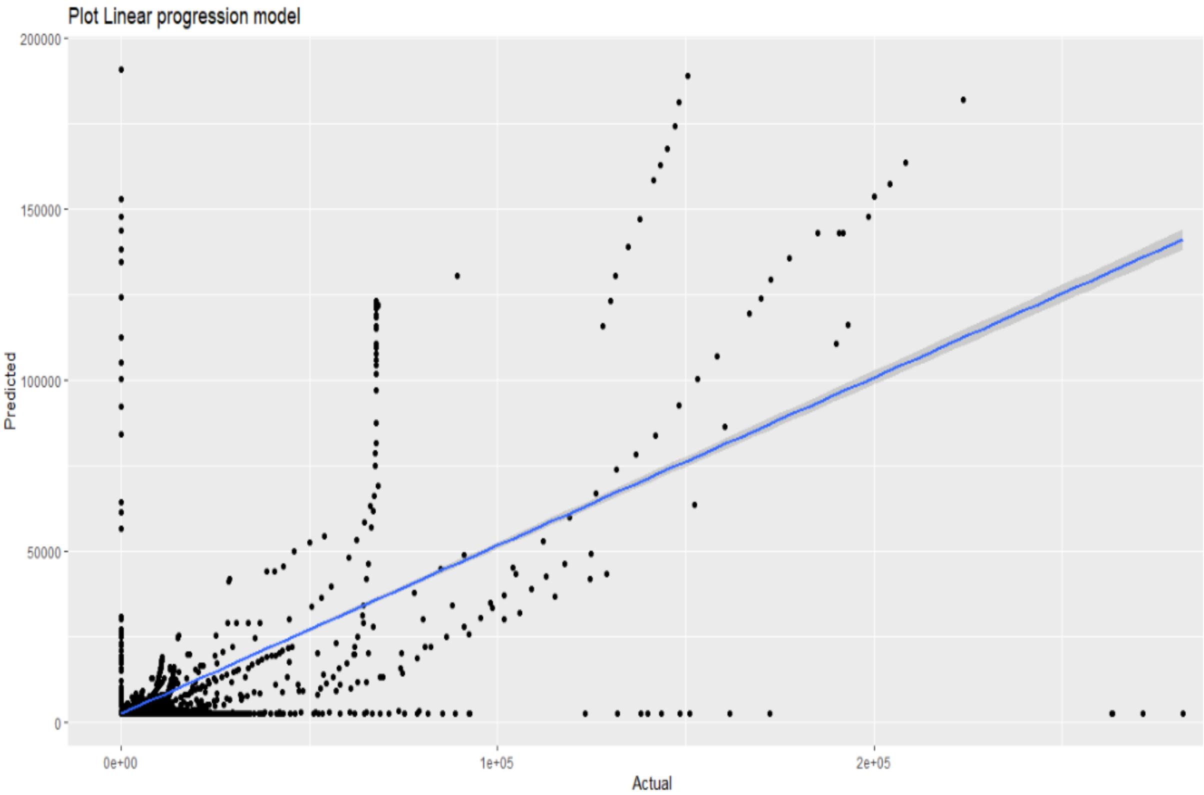
```
FD1 <- table(Actual = Test$Confirmed, Predicted = Result1)
FD<- as.data.frame(FD1)
Error <- FD$Actual-FD$Predicted

Actual <- Test$Confirmed
Predicted <- Result1
View(Predicted)
View(Actual)

Error <- Actual - Predicted

View(Error)
Final_Data <- cbind(Actual,Predicted, Error)

Final_Data

class(Final_Data)
FD<- as.data.frame(Final_Data)
View(FD)

RMS <- sqrt(mean((FD$Error)^2))
RMS
11719.71
```

Result :

Final Data: RMS = 11719.71

	Actual	Predicted	Error
1	1	2534.981	-2533.981
3	6	2534.981	-2528.981
4	1	2534.981	-2533.981
7	2	2534.981	-2532.981
9	4	2534.981	-2530.981
11	0	2534.981	-2534.981
13	0	2534.981	-2534.981
14	444	2587.482	-2143.482
16	0	2534.981	-2534.981
17	1	2534.981	-2533.981
19	0	2534.981	-2534.981

Linear Reg. Model for Predicted results:



#Multiple linear progression Model (M2)

```
M2 <- lm(Confirmed~Deaths+Recovered, data = Train)
M2

summary(M2)
# p-value: < 2.2e-16

#Analysis of variance
anova(Model_conf_rec,M2)

Result2 <- predict(M2, newdata = Test)

View(Result2)

Act2 <- Test$Confirmed
Pred2<- Result2
Error2 <- Act2-Pred2

cbind(Act2,Pred2,Error2)->FD2
FD2
FD2 <- as.data.frame(FD2)

#root mean square
RMS2 <- sqrt(mean((FD2$Error2)^2))
RMS2
5955.463
```

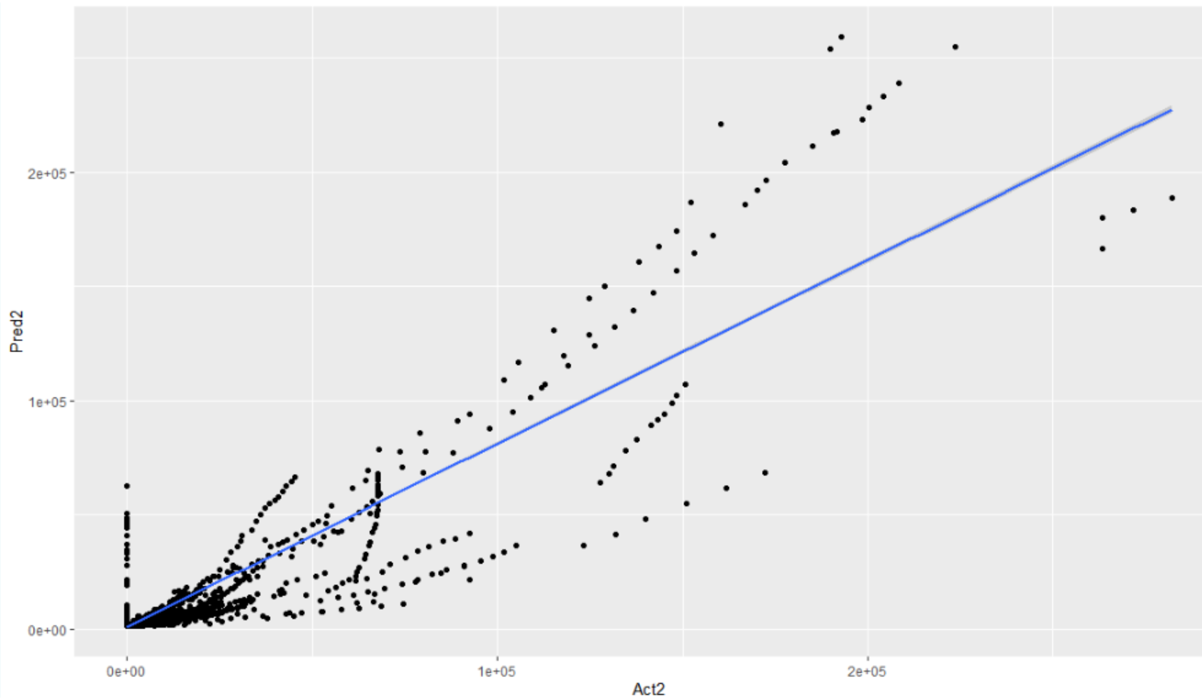
Model 2:

```
Call:
lm(formula = Confirmed ~ Deaths + Recovered, data = Train)

Coefficients:
(Intercept)       Deaths    Recovered
  1501.0015       8.5046       0.6118
```

Summary of Model2:

```
> summary(M2)

Call:
lm(formula = Confirmed ~ Deaths + Recovered, data = Train)

Residuals:
   Min     1Q Median     3Q    Max 
-69155  -1494  -1425   -852 114155 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.501e+03  6.911e+01   21.72   <2e-16 ***
Deaths      8.505e+00  4.831e-02  176.06   <2e-16 ***
Recovered   6.118e-01  1.244e-02   49.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7483 on 12105 degrees of freedom
Multiple R-squared:  0.8432,	Adjusted R-squared:  0.8431 
F-statistic: 3.254e+04 on 2 and 12105 DF,  p-value: < 2.2e-16
```

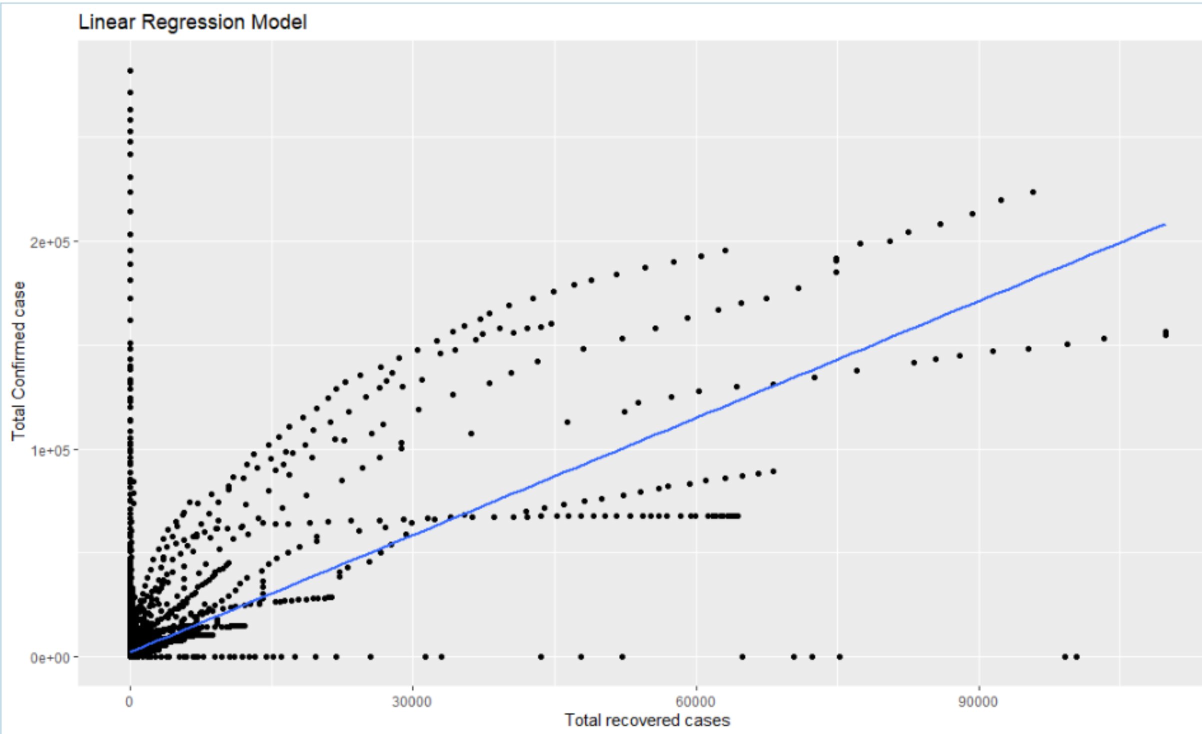
Linear Model for Predicted results:



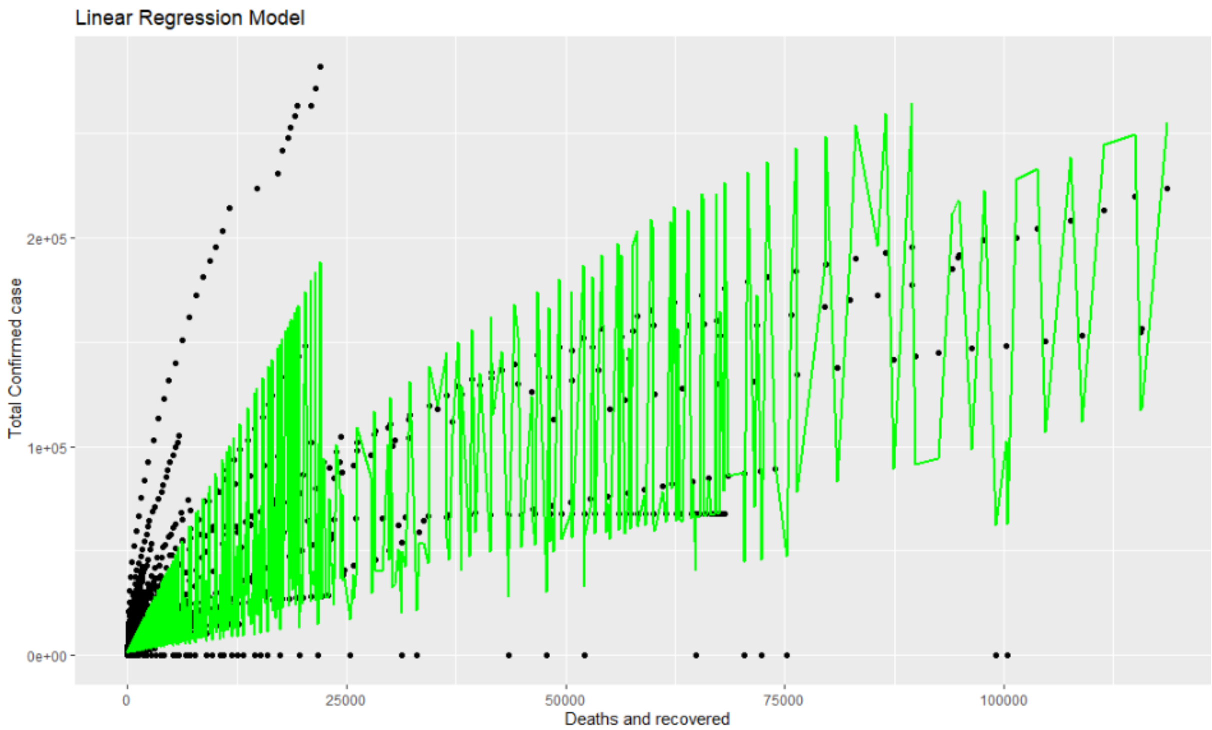
Model visualisation Via GGPLOT2

```
library(dplyr)
library(broom)

#Model 1 = Model_conf_rec
#Moodel 2 = M2

ggplot(augment(Model_conf_rec), aes(y =Confirmed, x = Recovered)) +
  geom_point()+geom_line(aes(y=.fitted), size = 1, col = "blue")+
    labs(x = "Total recovered cases", y= " Total Confirmed case", title = "Linear Regression Model")


#---plot via geom smooth
ggplot(augment(Model_conf_rec), aes(y =Confirmed, x = Recovered)) +geom_point()+
  geom_smooth(method = "lm", se= FALSE)+
  labs(x = "Total recovered cases", y= " Total Confirmed case", title = "Linear Regression Model")

#Model 2

ggplot(augment(M2), aes(y =Confirmed, x = Recovered+Deaths)) +
  geom_point()+geom_line(aes(y=.fitted), size = 1, col = "green")+
  labs(x = "Deaths and recovered", y= " Total Confirmed case", title = "Linear Regression Model")
```

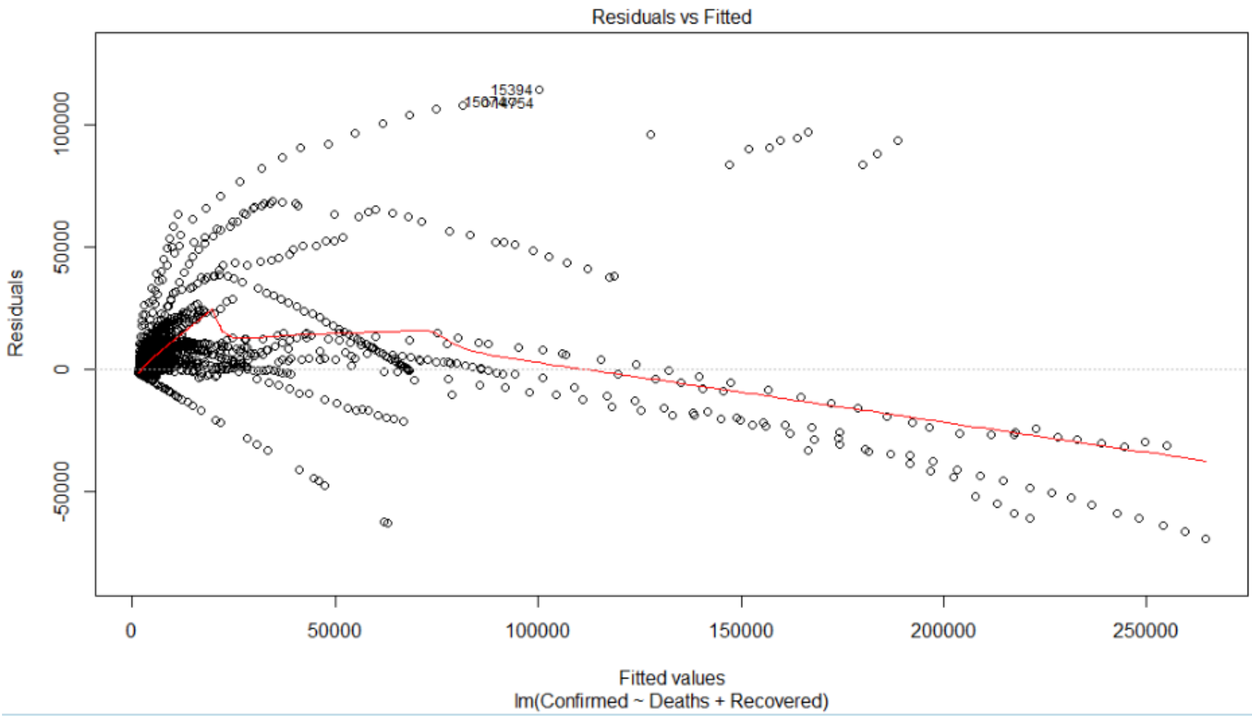
Plot:Final LM Model

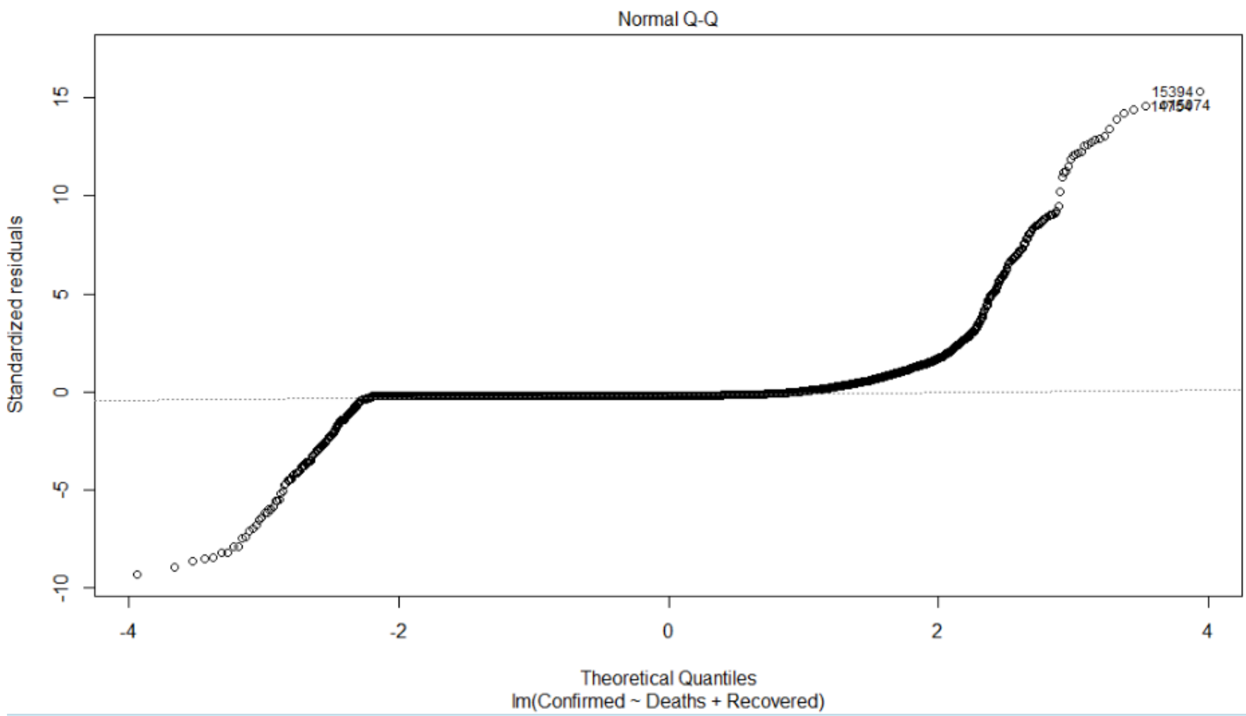


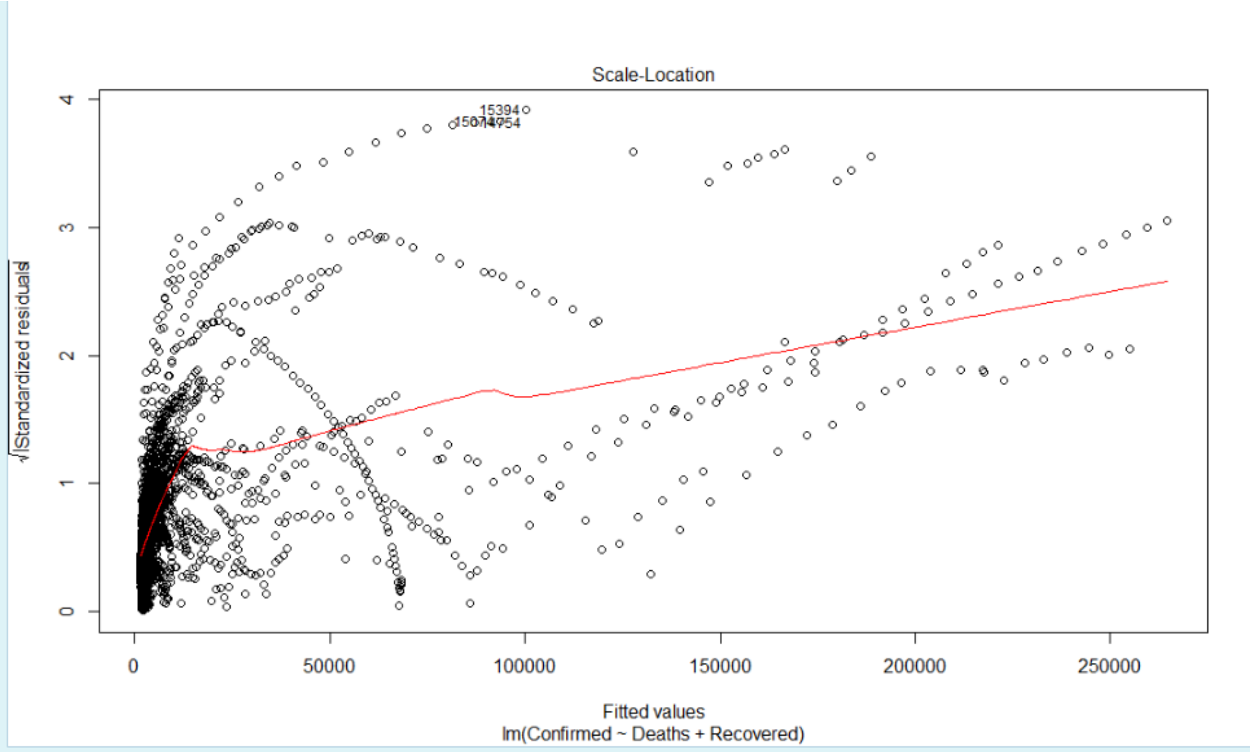
Plot: LM Model with Multiple Independent variable



Model Plots:


Residuals vs Fitted
15394
Residuals
100000
50000
0
-50000
0
50000
100000
150000
200000
250000
Fitted values
lm(Confirmed ~ Deaths + Recovered)


Normal Q-Q
15394
Standardized residuals
15
10
5
0
-5
-10
-4
-2
0
2
4
Theoretical Quantiles
lm(Confirmed ~ Deaths + Recovered)



Growth Factor

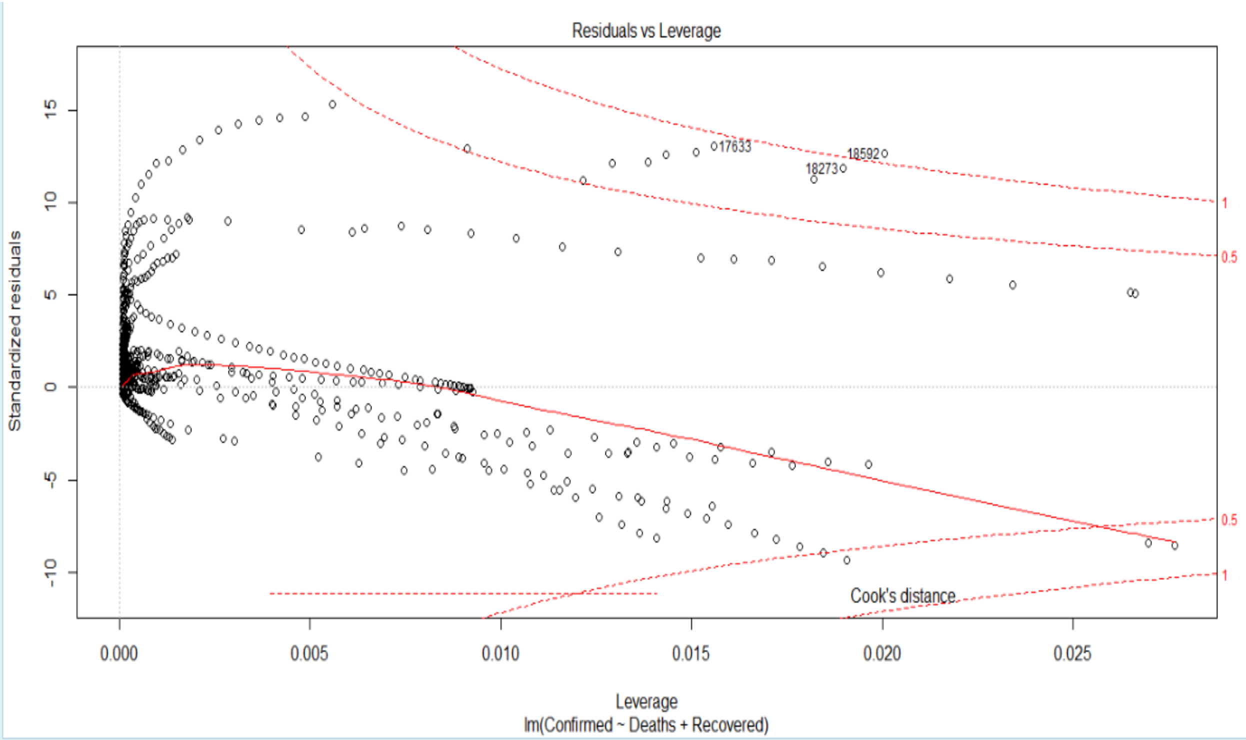
Growth factor is the factor by which a quantity multiplies itself over time. The formula used is:

Formula: Every day's new (Confirmed,Recovered,Deaths) / new (Confirmed,Recovered,Deaths) on the previous day.

A growth factor **above 1 indicates an increase correspoding cases**.

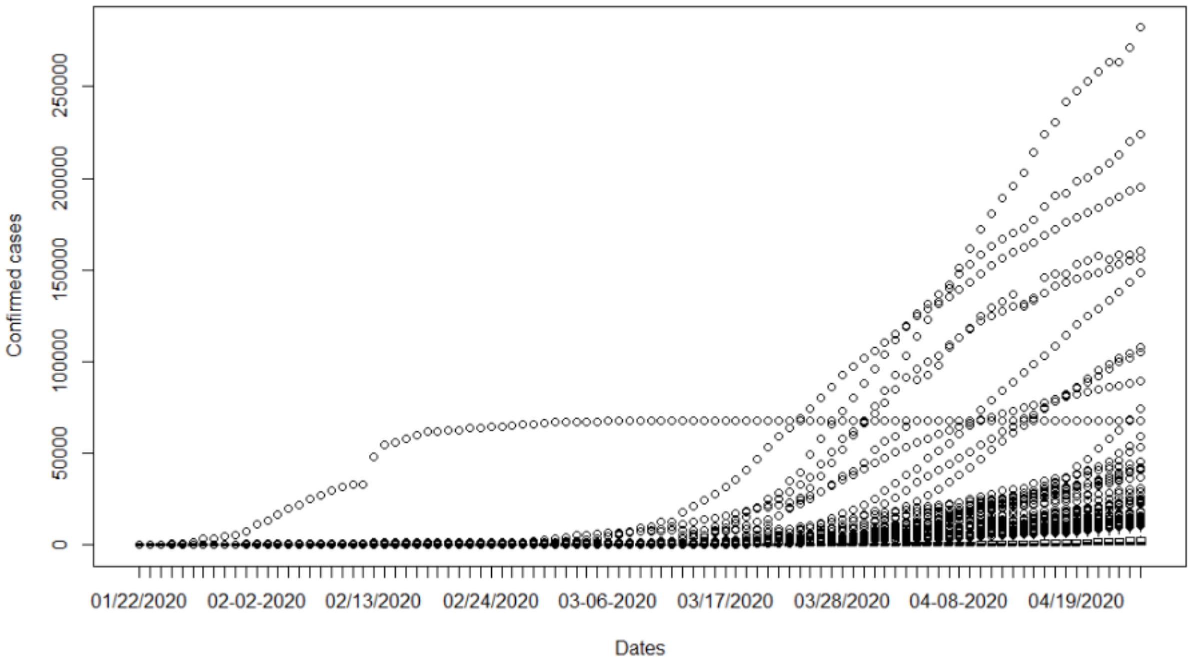
A growth factor **above 1 but trending downward** is a positive sign, whereas a **growth factor constantly above 1 is the sign of exponential growth**.

A growth factor **constant at 1 indicates there is no change in any kind of cases**.

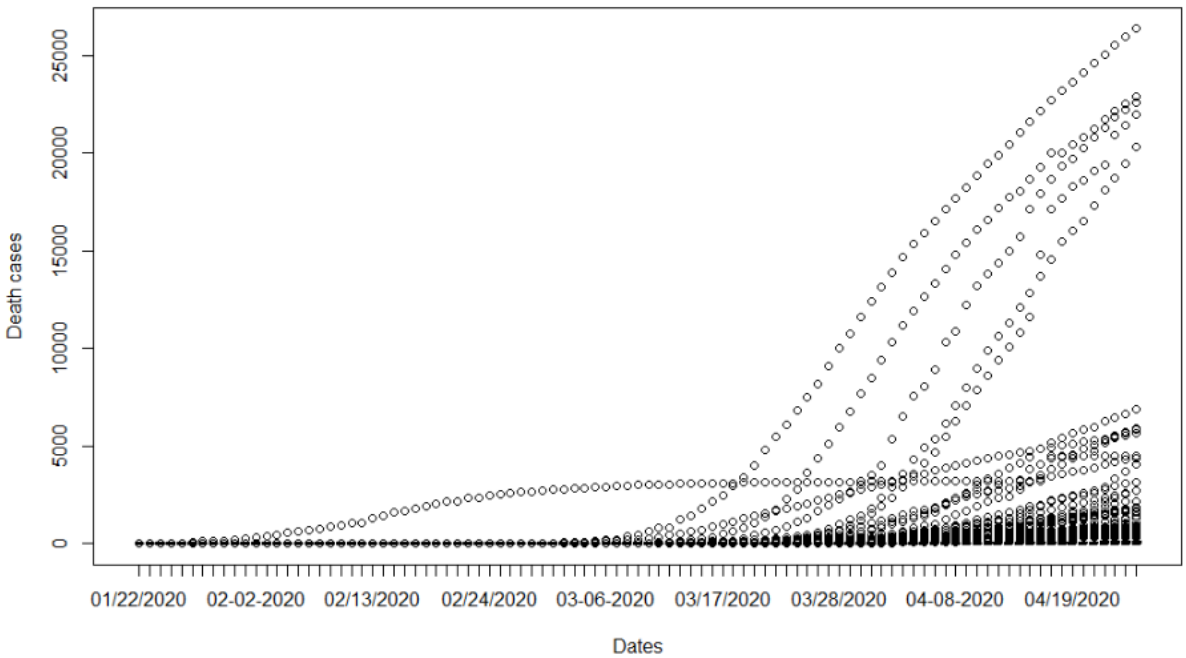


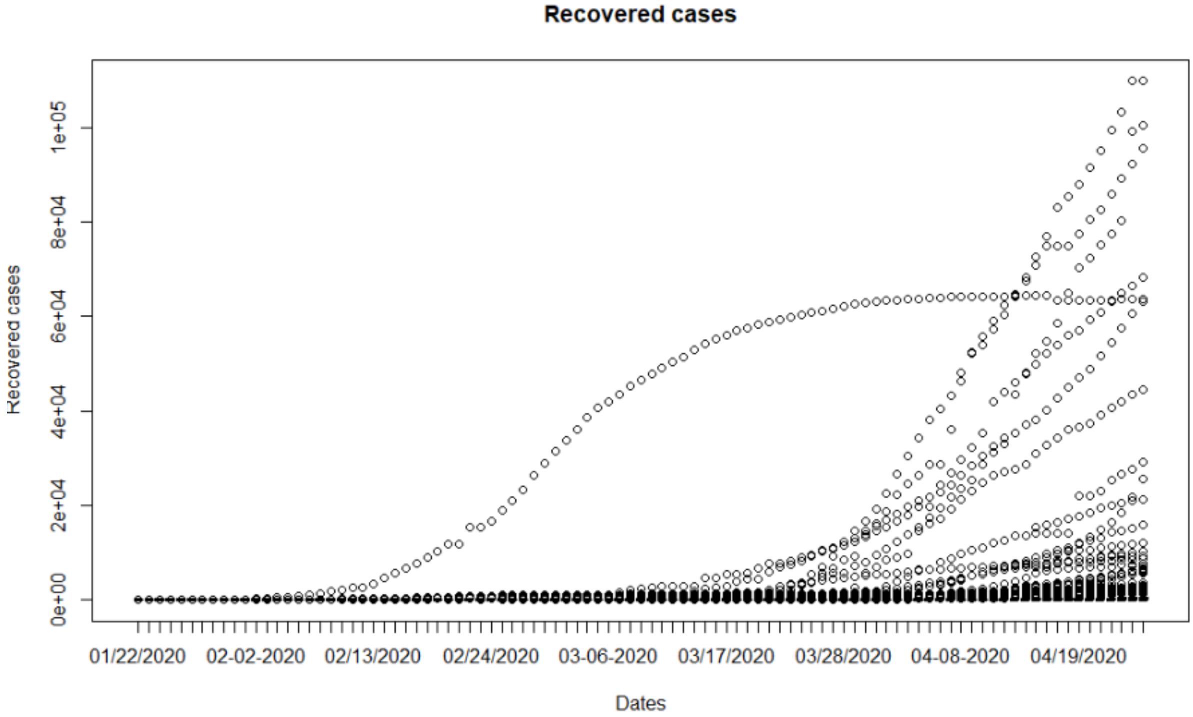
Case Analysis plots till date (24th May 2020)

COnfirmed cases



Death cases





Increase in number of Active Cases is probably an indication of Recovered case or Death case number is dropping in comparison to number of Confirmed Cases drastically.

#--

#Date : 24th April 2020

Akshay Bayas