Indian Institute of Information Technology-Dharwad

Name: Modem Praveen **Roll No:**17BCS035 **Mail:** modemiiit@gmail.com

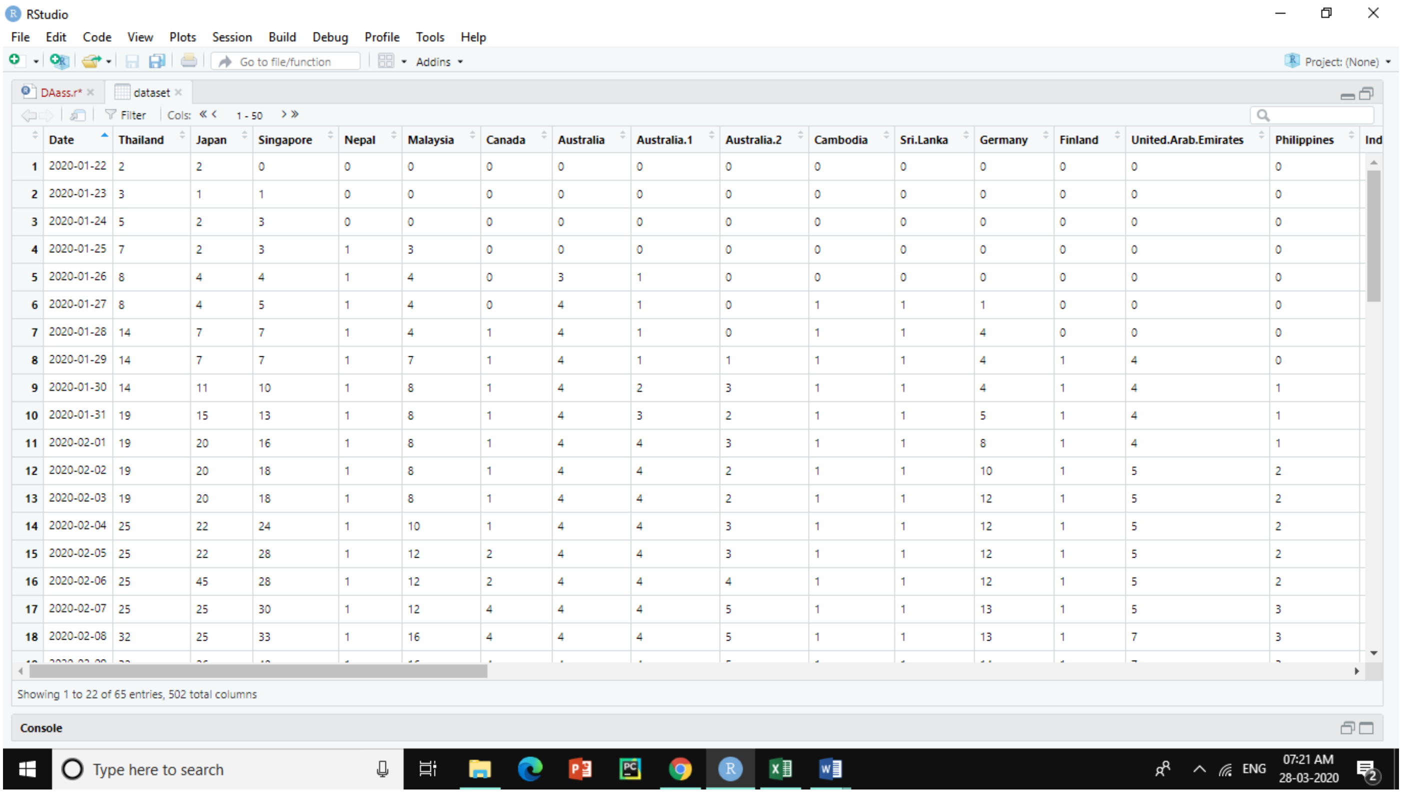
Data Analytics

Time series analysis on Corona virus infected cases

Files Url : https://github.com/modem0011/Covid-19-analysis

This dataset consists of 50 columns,1st column has Dates (22 Jan,2020 – 26 Jan ,2020) and other columns has country wise Corona infected cases.

Data Cleaning and Manipulating: Data cleaning and Data manipulation is done in a convenient way.



	Date	Thailand	Japan	Singapore	Nepal	Malaysia	Canada	Australia	Australia.1	Australia.2	Cambodia	Sri.Lanka	Germany	Finland	United.Arab.Emirates	Philippines	Ind
1	2020-01-22	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	2020-01-23	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
3	2020-01-24	5	2	3	0	0	0	0	0	0	0	0	0	0	0	0	
4	2020-01-25	7	2	3	1	3	0	0	0	0	0	0	0	0	0	0	
5	2020-01-26	8	4	4	1	4	0	3	1	0	0	0	0	0	0	0	
6	2020-01-27	8	4	5	1	4	0	4	1	0	1	1	1	0	0	0	
7	2020-01-28	14	7	7	1	4	1	4	1	0	1	1	4	0	0	0	
8	2020-01-29	14	7	7	1	7	1	4	1	1	1	1	4	1	4	0	
9	2020-01-30	14	11	10	1	8	1	4	2	3	1	1	4	1	4	1	
10	2020-01-31	19	15	13	1	8	1	4	3	2	1	1	5	1	4	1	
11	2020-02-01	19	20	16	1	8	1	4	4	3	1	1	8	1	4	1	
12	2020-02-02	19	20	18	1	8	1	4	4	2	1	1	10	1	5	2	
13	2020-02-03	19	20	18	1	8	1	4	4	2	1	1	12	1	5	2	
14	2020-02-04	25	22	24	1	10	1	4	4	3	1	1	12	1	5	2	
15	2020-02-05	25	22	28	1	12	2	4	4	3	1	1	12	1	5	2	
16	2020-02-06	25	45	28	1	12	2	4	4	4	1	1	12	1	5	2	
17	2020-02-07	25	25	30	1	12	4	4	4	5	1	1	13	1	5	3	
18	2020-02-08	32	25	33	1	16	4	4	4	5	1	1	13	1	7	3	

Showing 1 to 22 of 65 entries, 502 total columns

Selecting Data Set

```
dataset<-read.csv(file.choose())
```

Download the dataset from above URL (given in page 1) and enter above code in R Studio.It will ask which file to select . Select "cleaned_file.csv " from downloaded folder.

Describing data

```
install.packages("Hmisc")
library("Hmisc")
describe(dataset)
```

To know about data, enter above code and check output. It describes whole dataset.

Note: Try "summary(dataset)" , " str(dataset) " for clear understanding about dataset.("str " shows structure of dataset.)

There are many countries are there in dataset so we can focus on some countries which we want to predict and analyse.

Here I am focusing on India, Italy, Norway, Israel, Iraq, Spain, Brazil, Belgium, Sweden.

Select columns using $ symbol

Example:

```
dataset$Date
dataset$India
dataset$Italy
```

Note: There are several methods for selecting columns. Above is one of the method used in this project

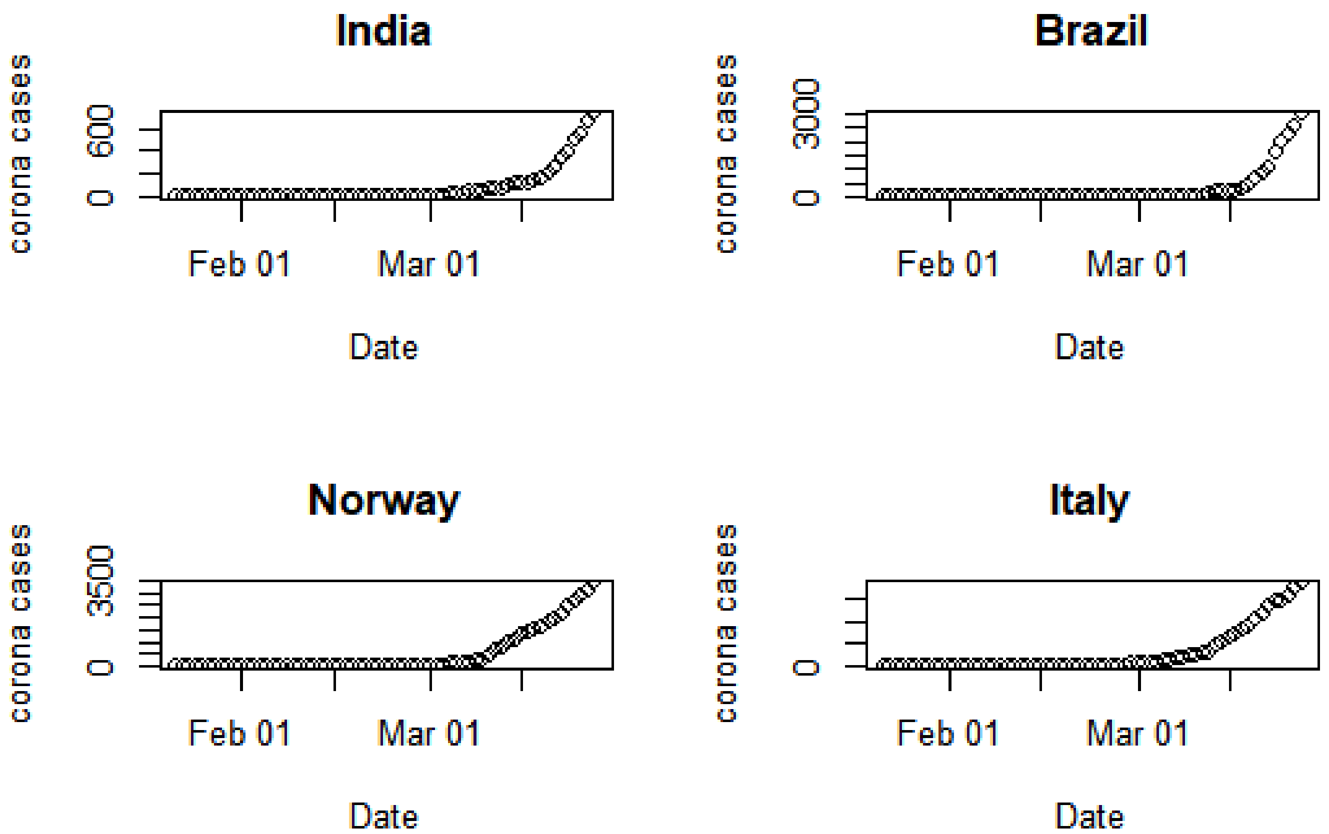
Define format of date is format by using below code.

```
install.packages("lubridate")
library(lubridate)
dataset2$Date<-dmy(dataset$Date)
```

Visualising

```
par(mfrow=c(2,2))
plot(dataset$Date,dataset$India,xlab = "Date",ylab = "corona cases",main = "India")
plot(dataset$Date,dataset$Brazil,xlab = "Date",ylab = "corona cases",main = "Brazil")
plot(dataset$Date,dataset$Norway,xlab = "Date",ylab = "corona cases",main="Norway")
plot(dataset$Date,dataset$Italy,xlab = "Date",ylab = "corona cases",main = "Italy")
```

Note: with par function we plot n plots in a single screen . Here we plotted as (2,2) matrix



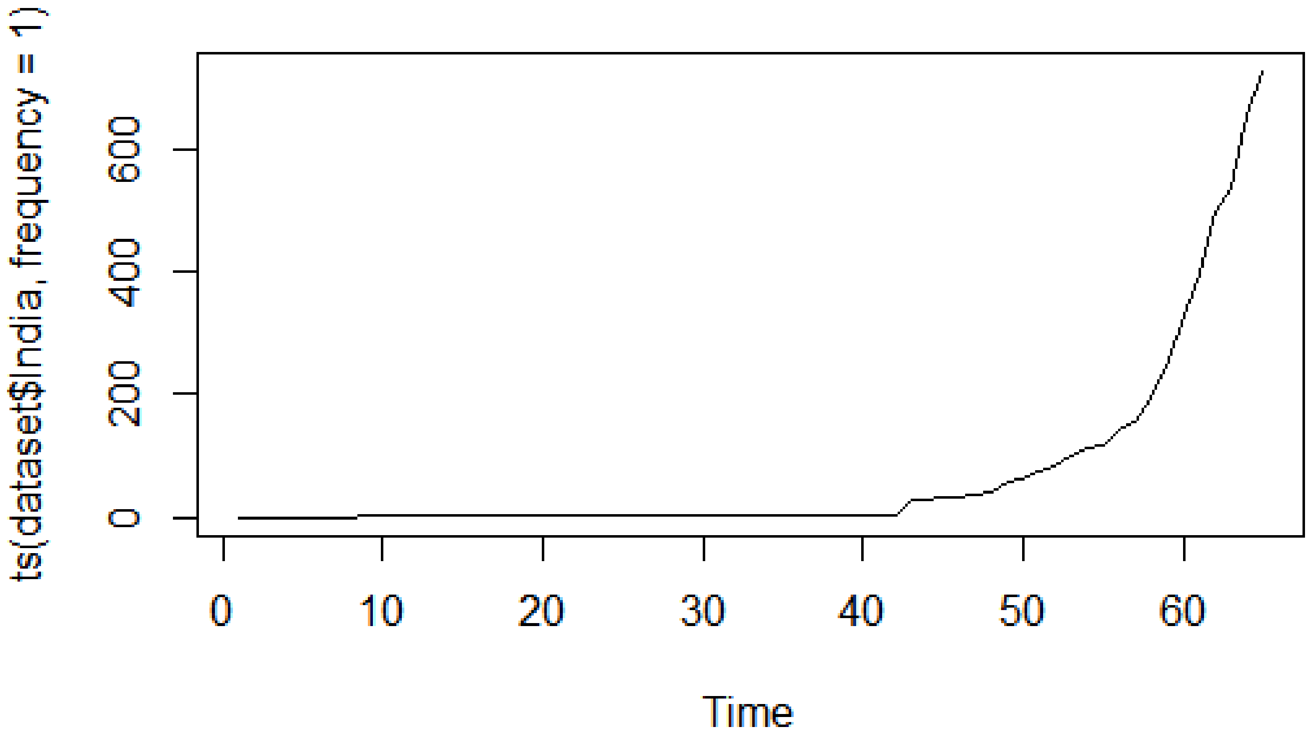
All above plots are looking little similar but some variations are there .If we see Y-axis ranges we can notice changes.

Time series object with the data

Time series is a **series** of data points in which each data point is associated with a timestamp. ... The data for the **time series** is stored in an **R object** called **time-series object**. It is also a **R** data **object** like a vector or data frame. The **time series object** is created by using the ts() function.

Code:

```
ts(dataset$India)
ts(dataset)
par(mfrow=c(1,1))
plot(ts(dataset$India))
```



Finding Mean

This dataset is in the form of cumulative sums of corona cases in different countries. So finding monthly or yearly mean values is not possible.

But we can find mean value of corona cases in selected countries.

Code:

A<-data.frame (dataset$Date, dataset$India, dataset$Italy, dataset$Norway, dataset$Israel, dataset$Iraq, dataset$Spain, dataset$Brazil, dataset$Belgium, dataset$Sweden)

B<-sapply(A, max)

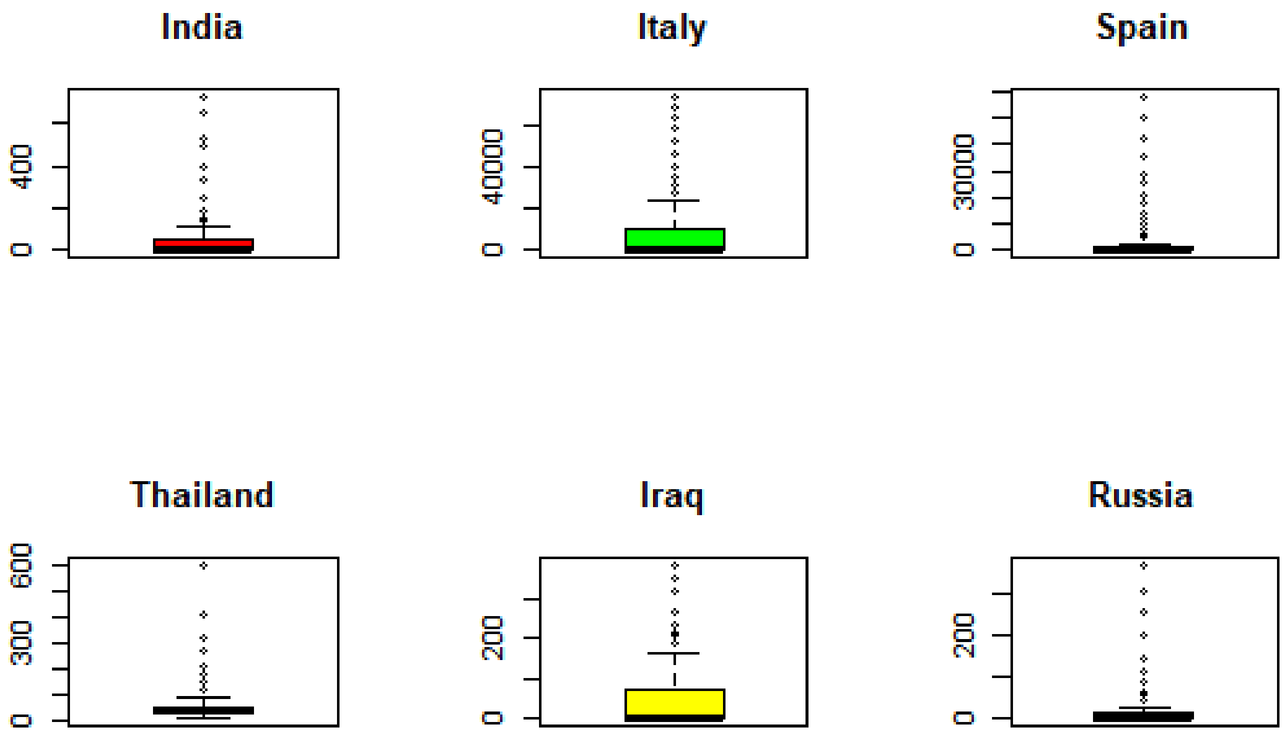
mean(B)

```
> A<-data.frame(dataset$Date,dataset$India,dataset$Italy,dataset$Norway,dataset$Israel,dataset
$Iraq,dataset$Spain,dataset$Brazil,dataset$Belgium,dataset$Sweden)
> sapply(A, max)
   dataset.Date   dataset.India   dataset.Italy  dataset.Norway  dataset.Israel
          18347             727           74386            3372            2693
   dataset.Iraq   dataset.Spain  dataset.Brazil dataset.Belgium  dataset.Sweden
            382           57786            2985            6235            2840
> M<-sapply(A, max)
> mean(M)
[1] 16975.3
```

Above mean value says in an average in every country 16975.3 people got infected by Corona Virus.

Boxplots of a corona cases in a span of 2 months (approx)

```
par(mfrow=c(2,3))
boxplot(dataset$India,main="India",col="red")
boxplot(dataset$Italy,main="Italy",col="green")
boxplot(dataset$Spain,main="Spain",col="pink")
boxplot(dataset$Thailand,main="Thailand",col="blue")
boxplot(dataset$Iraq,main="Iraq",col="yellow")
boxplot(dataset$Russia,main="Russia",col="skyblue")
```



Dots in plots represents outliers

Note:

```
> stl(ts(dataset$India))
Error in stl(ts(dataset$India)) :
  series is not periodic or has less than two periods
```

Stl function required atleast 2 periods . But here we have only 1 period (dates).To overcome this problem we have to collect yearly data . As we know corona started spreading recently .So there is no possibility of collecting more data.
Only 1 seasonality we have .That too not periodic.It's additive.

Residuals:

```
> x1<-ts(dataset$India[20:65])
> model1<-auto.arima(x1)
> model1$residuals
Time Series:
Start = 1
End = 46
Frequency = 1
 [1]  1.341640e-03 -4.024906e-03 -9.125540e-06  5.773160e-15  0.000000e+00 -4.440892e-16
 [7] -4.440892e-16  1.332268e-15  4.440892e-16 -1.332268e-15 -4.440892e-16  1.332268e-15
[13]  4.440892e-16 -1.332268e-15 -4.440892e-16  1.332268e-15  4.440892e-16 -1.332268e-15
[19] -4.440892e-16  1.332268e-15  4.440892e-16  2.000000e+00 -3.764460e-01  2.137645e+01
[25] -2.329128e+00 -1.804732e+01  1.188223e+00  3.623554e+00  6.235540e-01  8.188223e+00
[31]  3.059932e-01 -6.824392e-01  2.058885e+00  9.376446e+00 -7.045274e-02 -1.230599e+01
[37]  1.294111e+01  4.800209e+00  1.669401e+01  3.148265e+01  4.574132e+01  9.223973e+00
[43]  2.076446e+01 -3.596425e+01  3.042272e+01  1.718927e+01
```

Model for the data using the HoltWinters method

x<-ts(dataset$India)

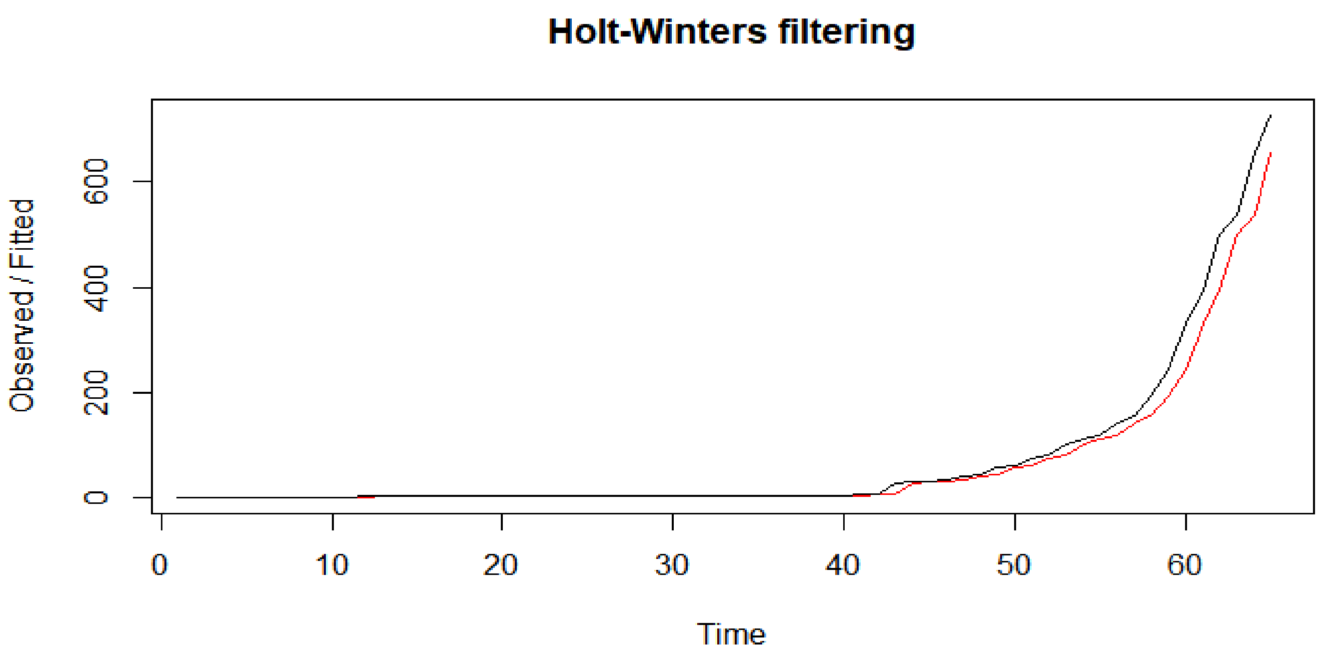
model<-HoltWinters(x,beta = F,gamma = F)

model

plot(model)

Here I mentioned beta and gamma values as False because those are trend and seasonality coefficients .As we know our data don't have proper seasonality .Which I mentioned in page:7. so I assigned them as False and by default it took alpha as 0.9999288 for above data.

```
> model<-HoltWinters(x,beta = F,gamma = F)
> model
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = x, beta = F, gamma = F)

Smoothing parameters:
 alpha: 0.9999288
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 726.995
```



Visualisation of HoltWinters model for different countries

x1<-ts(dataset$India)

x2<-ts(dataset$Italy)

x3<-ts(dataset$Norway)

x4<-ts(dataset$Sweden)

model1<-HoltWinters(x1,beta = F,gamma = F)

model2<-HoltWinters(x2,beta = F,gamma = F)

model3<-HoltWinters(x3,beta = F,gamma = F)

model4<-HoltWinters(x4,beta = F,gamma = F)

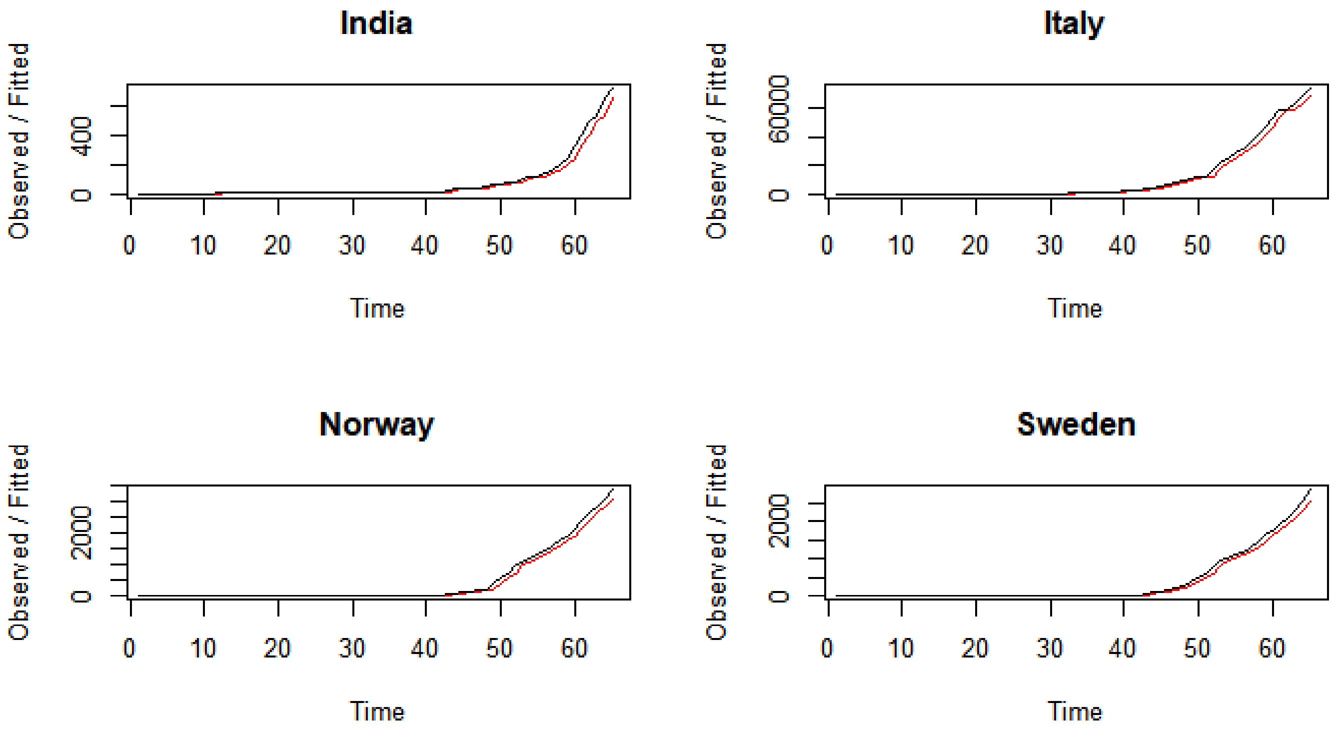
par(mfrow=c(2,2))

plot(model1,main = "India")

plot(model2,main="Italy")

plot(model3,main = "Norway")

plot(model4,main="Sweden")



Observed and Fitted lines are plotted above

Forecasting country wise next 4 days' corona cases.

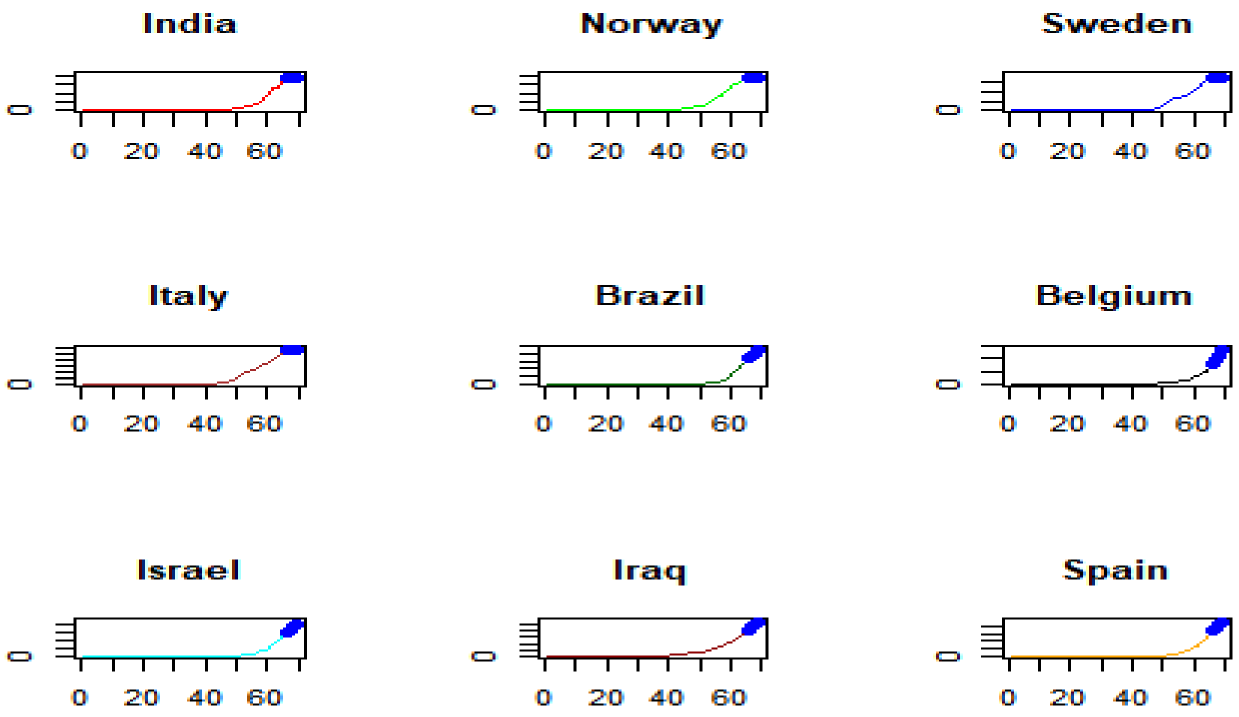
```
x1<-ts(dataset$India)
x2<-ts(dataset$Norway)
x3<-ts(dataset$Sweden)
x4<-ts(dataset$Italy)
x5<-ts(dataset$Brazil)
x6<-ts(dataset$Belgium)
x7<-ts(dataset$Israel)
x8<-ts(dataset$Iraq)
x9<-ts(dataset$Spain)
x10<-ts(dataset$Malaysia)
model1<-HoltWinters(x1,beta = F,gamma = F)
model2<-HoltWinters(x2,beta = F,gamma = F)
model3<-HoltWinters(x3,beta = F,gamma = F)
model4<-HoltWinters(x4,beta = F,gamma = F)
model5<-HoltWinters(x5,beta = F,gamma = F)
model6<-HoltWinters(x6,beta = F,gamma = F)
model7<-HoltWinters(x7,beta = F,gamma = F)
model8<-HoltWinters(x8,beta = F,gamma = F)
model9<-HoltWinters(x9,beta = F,gamma = F)
model10<-HoltWinters(x10,beta = F,gamma = F)
library(forecast)
forecast(model1,4)
forecast(model2,4)
forecast(model3,4)
forecast(model4,4)
forecast(model5,4)
forecast(model6,4)
forecast(model7,4)
forecast(model8,4)
forecast(model9,4)
forecast(model10,4)
```

```
> forecast(model1,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66        726.995 694.2059 759.7842 676.8483 777.1417
67        726.995 680.6258 773.3642 656.0794 797.9106
68        726.995 670.2052 783.7848 640.1425 813.8475
69        726.995 661.4202 792.5698 626.7070 827.2830
> forecast(model2,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       3371.988 3258.868 3485.108 3198.986 3544.989
67       3371.988 3212.016 3531.960 3127.332 3616.644
68       3371.988 3176.064 3567.911 3072.349 3671.627
69       3371.988 3145.756 3598.220 3025.996 3717.980
> forecast(model3,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       2839.975 2743.367 2936.582 2692.227 2987.723
67       2839.975 2703.357 2976.592 2631.036 3048.913
68       2839.975 2672.655 3007.294 2584.081 3095.868
69       2839.975 2646.772 3033.177 2544.496 3135.453
> forecast(model4,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       74385.76 71896.56 76874.97 70578.85 78192.68
67       74385.76 70865.57 77905.95 69002.10 79769.43
68       74385.76 70074.46 78697.06 67792.20 80979.33
69       74385.76 69407.52 79364.01 66772.20 81999.33
> forecast(model5,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66        2984.97 2835.699 3134.240 2756.681 3213.259
67        2984.97 2773.877 3196.062 2662.131 3307.808
68        2984.97 2726.438 3243.501 2589.580 3380.359
69        2984.97 2686.445 3283.494 2528.416 3441.524

> forecast(model6,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       6234.906 5949.105 6520.708 5797.811 6672.002
67       6234.906 5830.736 6639.076 5616.782 6853.031
68       6234.906 5739.907 6729.906 5477.871 6991.942
69       6234.906 5663.334 6806.479 5360.762 7109.051
> forecast(model7,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       2692.983 2556.821 2829.144 2484.742 2901.223
67       2692.983 2500.427 2885.539 2398.494 2987.472
68       2692.983 2457.153 2928.812 2332.312 3053.653
69       2692.983 2420.671 2965.294 2276.518 3109.447
> forecast(model8,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       381.9972 367.9319 396.0626 360.4862 403.5083
67       381.9972 362.1067 401.8878 351.5772 412.4173
68       381.9972 357.6367 406.3578 344.7409 419.2536
69       381.9972 353.8682 410.1263 338.9776 425.0168
> forecast(model9,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       57785.42 55261.00 60309.85 53924.65 61646.20
67       57785.42 54215.47 61355.38 52325.65 63245.20
68       57785.42 53413.19 62157.66 51098.67 64472.18
69       57785.42 52736.83 62834.01 50064.27 65506.58
> forecast(model10,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       2030.981 1953.415 2108.547 1912.354 2149.608
67       2030.981 1921.290 2140.672 1863.224 2198.738
68       2030.981 1896.640 2165.322 1825.524 2236.438
69       2030.981 1875.858 2186.104 1793.741 2268.221
```

Time series started from 1. It considered 22nd Jan 2020 as 1 and so on…According to that we trained model from 1 to 65 (22nd Jan2020 to 26th March 2020) and we Predicted Corona cases for 66,67,68,69 (27th March 2020 – 30th March 2020)

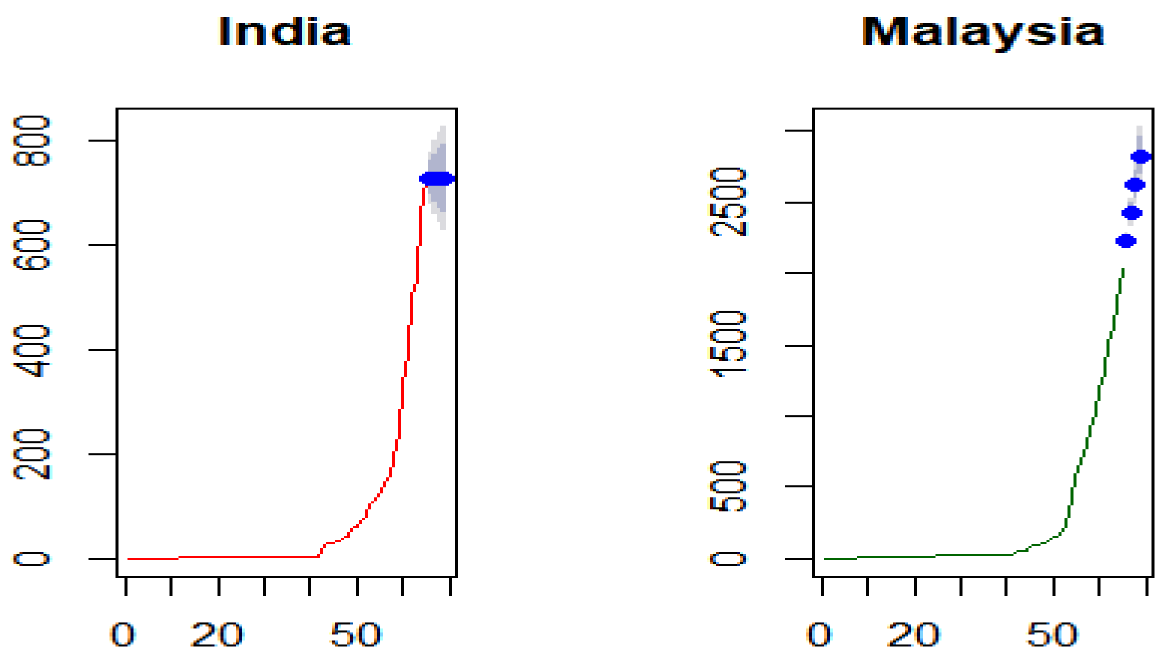
Visualising Predictions

```
par(mfrow=c(3,3))

plot(forecast(model1,4),col = "red",main = "India")

plot(forecast(model2,4),col="green",main = "Norway")

plot(forecast(model3,4),col="blue",main="Sweden")

plot(forecast(model4,4),col="brown",main = "Italy")

plot(forecast(model5,4),col="dark green",main="Brazil")

plot(forecast(model6,4),col = "black",main = "Belgium")

plot(forecast(model7,4),col="cyan",main="Israel")

plot(forecast(model8,4),col = "dark red",main="Iraq")

plot(forecast(model9,4),col="orange",main = "Spain")
```



Thick Blue area in plot is predicted area.

Clear view of plot for India and Malaysia.



Above 4 blue dots are predicted values using HoltWinters model

Actual Corona Cases

Above what we saw is Predicted values and below values are Actual values.

Today is 29 march ,2020 . But I predicted till March 30 so here I am comparing with 27th ,28th actual and predicted values.

Source: CoronaWorldmeter

India	Norway	Sweden	Italy	Brazil	Belgium	Israel	Iraq	Spain	Malaysia
887	3771	3069	86498	3417	7284	3035	458	65719	2161
987	4015	3447	92472	3904	9134	3619	506	73235	2320

We can compare 66,67(27 march 2020,28 march 2020) predicted values(page: 11) with these above values.

Visualizing Actual VS Predicted

par(mfrow=c(2,2))

plot(c(66,67),c(887,987),xlim = c(66,67),ylim = c(100,1500),"b",col="green", ylab="Actual vs Predict",main = "India")

lines(c(66,67),c(777,797),"b",col="red")

plot(c(66,67),c(3771,4015),xlim = c(66,67),ylim = c(1000,5000),"b",col="green", ylab="Actual vs Predict",main = "Norway")

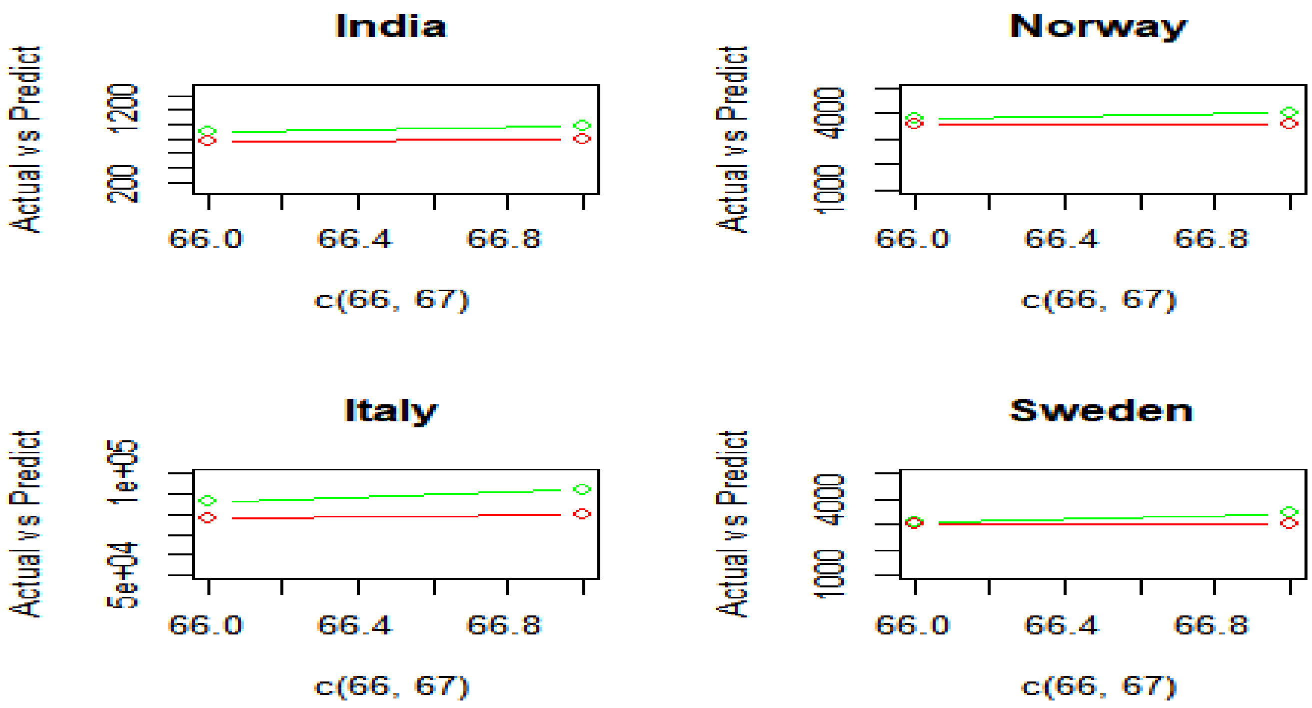
lines(c(66,67),c(3544,3616),"b",col="red")

plot(c(66,67),c(86498,92472),xlim = c(66,67),ylim = c(50000,100000),"b",col="green", ylab="Actual vs Predict",main = "Italy")

lines(c(66,67),c(78192,79769),"b",col="red")

plot(c(66,67),c(3069,3447),xlim = c(66,67),ylim = c(1000,5000),"b",col="green", ylab="Actual vs Predict",main = "Sweden")

lines(c(66,67),c(2987,3048),"b",col="red")



Green is actual and Red is predicted line.

- For this particular dataset model prediction is very poor because of not giving beta and gamma coefficients(seasonality and Trend coefficients).
- I already mentioned reason in page 7.

RMSE

library(Metrics)

rmse(c(887,987),c(777,797))   # india

rmse(c(3771,4015),c(3544,3616))   #Norway

rmse(c(3069,3447),c(2987,3048))   #Sweden

```
> rmse(c(887,987),c(777,797))
[1] 155.2417
> rmse(c(3771,4015),c(3544,3616))
[1] 324.5998
> rmse(c(3069,3447),c(2987,3048))
[1] 288.0321
```

Changing alpha Value and Comparing OutPuts

```
model1<-HoltWinters(x1,beta = F,gamma = F)
model1
a=forecast(model1,2)
a
model1<-HoltWinters(x1,beta = F,gamma = F,alpha=0.7)
a=forecast(model1,2)
a
model1<-HoltWinters(x1,beta = F,gamma = F,alpha=1)
a=forecast(model1,2)
a
```

```
> model1<-HoltWinters(x1,beta = F,gamma = F)
> model1
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = x1, beta = F, gamma = F)

Smoothing parameters:
 alpha: 0.9999288
 beta : FALSE
 gamma: FALSE

Coefficients:
     [,1]
a 726.995
> a=forecast(model1,2)
> a
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66        726.995 694.2059 759.7842 676.8483 777.1417
67        726.995 680.6258 773.3642 656.0794 797.9106
> model1<-HoltWinters(x1,beta = F,gamma = F,alpha=0.7)
> a=forecast(model1,2)
> a
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       693.0398 649.7784 736.3011 626.8773 759.2023
67       693.0398 640.2326 745.8470 612.2781 773.8014
> model1<-HoltWinters(x1,beta = F,gamma = F,alpha=1)
> a=forecast(model1,2)
> a
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66            727 694.2125 759.7875 676.8559 777.1441
67            727 680.6315 773.3685 656.0855 797.9145
> 
```

In above code we gave 3 different alpha values one is default (0.9999288) and other two are 0.7 and 1

Before we found rmse values using default alpha predictions .But here if we give alpha value 0.7 . It is predicting far way value than actual value . so rmse will be higher

If we change alpha value >0.9999288 then predicted value is going closer to actual Value. So rmse will be lower than before alpha values

Model for the data using the ARIMA method

x1<-ts(dataset$India)

x2<-ts(dataset$Norway)

x3<-ts(dataset$Sweden)

x4<-ts(dataset$Italy)

x5<-ts(dataset$Brazil)

x6<-ts(dataset$Belgium)

x7<-ts(dataset$Israel)

x8<-ts(dataset$Iraq)

x9<-ts(dataset$Spain)

x10<-ts(dataset$Malaysia)

model1<-auto.arima(x1)

model2<-auto.arima(x2)

model3<-auto.arima(x3)

model4<-auto.arima(x4)

model5<-auto.arima(x5)

model6<-auto.arima(x6)

model7<-auto.arima(x7)

model8<-auto.arima(x8)

model9<-auto.arima(x9)

model10<-auto.arima(x10)

model1<-auto.arima(x1)   # India

model

```
> model1<-auto.arima(x1)
> model1
Series: x1
ARIMA(1,2,0)

Coefficients:
          ar1
      -0.8185
s.e.   0.0807

sigma^2 estimated as 124.2:  log likelihood=-241.34
AIC=486.68   AICc=486.88   BIC=490.96
```

Here we are using ARIMA model.We didn't mention order so by default it is taking 1,2,0 values.

Predicting Corona cases using ARIMA Model

forecast(model1,4)

forecast(model2,4)

forecast(model3,4)

forecast(model4,4)

forecast(model5,4)

forecast(model6,4)

forecast(model7,4)

forecast(model8,4)

forecast(model9,4)

forecast(model10,4)

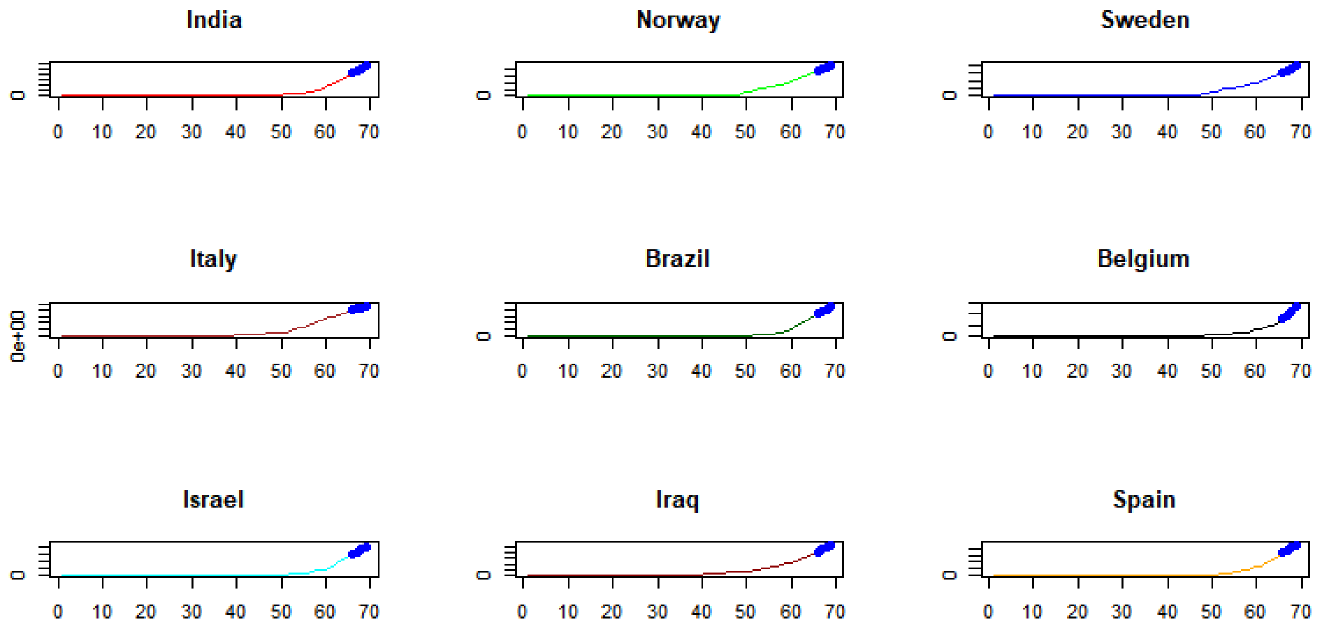
```
   Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
66       838.7443  824.4607  853.0279  816.8994  860.5892
67       916.3202  894.2110  938.4294  882.5071  950.1334
68      1021.8635  985.3669 1058.3602  966.0467 1077.6803
69      1104.5151 1055.0647 1153.9655 1028.8873 1180.1429
> forecast(model2,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       3589.282 3542.812 3635.752 3518.213 3660.351
67       3845.931 3768.204 3923.658 3727.058 3964.804
68       4118.724 3988.111 4249.337 3918.969 4318.480
69       4369.231 4169.421 4569.040 4063.649 4674.813
> forecast(model3,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       3115.017 3075.027 3155.007 3053.857 3176.176
67       3382.007 3307.789 3456.225 3268.500 3495.513
68       3633.834 3506.847 3760.820 3439.625 3828.042
69       3882.162 3695.153 4069.171 3596.156 4168.168
> forecast(model4,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       79110.32 77765.19 80455.45 77053.13 81167.51
67       83834.64 81554.61 86114.67 80347.63 87321.65
68       88558.96 85294.11 91823.82 83565.80 93552.13
69       93283.28 88960.87 97605.70 86672.72 99893.85
> forecast(model5,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       3367.764 3292.001 3443.527 3251.895 3483.634
67       3750.528 3610.201 3890.855 3535.917 3965.140
68       4133.293 3920.119 4346.466 3807.272 4459.313
69       4516.057 4221.810 4810.304 4066.045 4966.068
> forecast(model6,4)
   Point Forecast     Lo 80    Hi 80     Lo 95     Hi 95
66       7793.259  7683.748  7902.77  7625.777  7960.742
67       9598.289  9306.978  9889.60  9152.767 10043.811
68      11403.319 10816.617 11990.02 10506.035 12300.603
69      13208.349 12257.233 14159.46 11753.743 14662.954
```

```
> forecast(model7,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       3042.452 2980.399 3104.506 2947.550 3137.355
67       3370.211 3242.708 3497.715 3175.212 3565.211
68       3716.460 3502.709 3930.211 3389.556 4043.363
69       4046.949 3738.418 4355.481 3575.091 4518.808
> forecast(model8,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       414.8808 405.7850 423.9766 400.9700 428.7917
67       447.7616 432.4421 463.0812 424.3324 471.1909
68       480.6425 458.8079 502.4770 447.2494 514.0355
69       513.5233 484.7172 542.3293 469.4682 557.5784
> forecast(model9,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       65626.51 64773.80 66479.23 64322.40 66930.63
67       73694.69 72177.85 75211.53 71374.88 76014.50
68       81642.46 79215.38 84069.55 77930.56 85354.37
69       89653.91 86234.38 93073.45 84424.19 94883.64
> forecast(model10,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       2229.254 2193.449 2265.060 2174.494 2284.014
67       2427.509 2361.919 2493.099 2327.198 2527.820
68       2625.763 2526.802 2724.724 2474.415 2777.111
69       2824.017 2688.042 2959.993 2616.061 3031.974
```

Time series started from 1 it considered 22nd Jan 2020 as 1 and so on…According to that we trained model from 1 to 65 (22nd Jan2020 to 26th March 2020) and we Predicted Corona cases for 66,67,68,69 (27th March 2020 – 30th March 2020)

Example: For India (model 1) 27th -30th march it predicted 860,950,1077,1180 corona cases in India.

Visualization



Actual Corona Cases

Above what we saw is Predicted values and below values are Actual values .

Today is 29 march ,2020   but I predicted till March 30 so here I am comparing with 27th ,28th actual and predicted values.

India	Norway	Sweden	Italy	Brazil	Belgium	Israel	Iraq	Spain	Malaysia
887	3771	3069	86498	3417	7284	3035	458	65719	2161
987	4015	3447	92472	3904	9134	3619	506	73235	2320

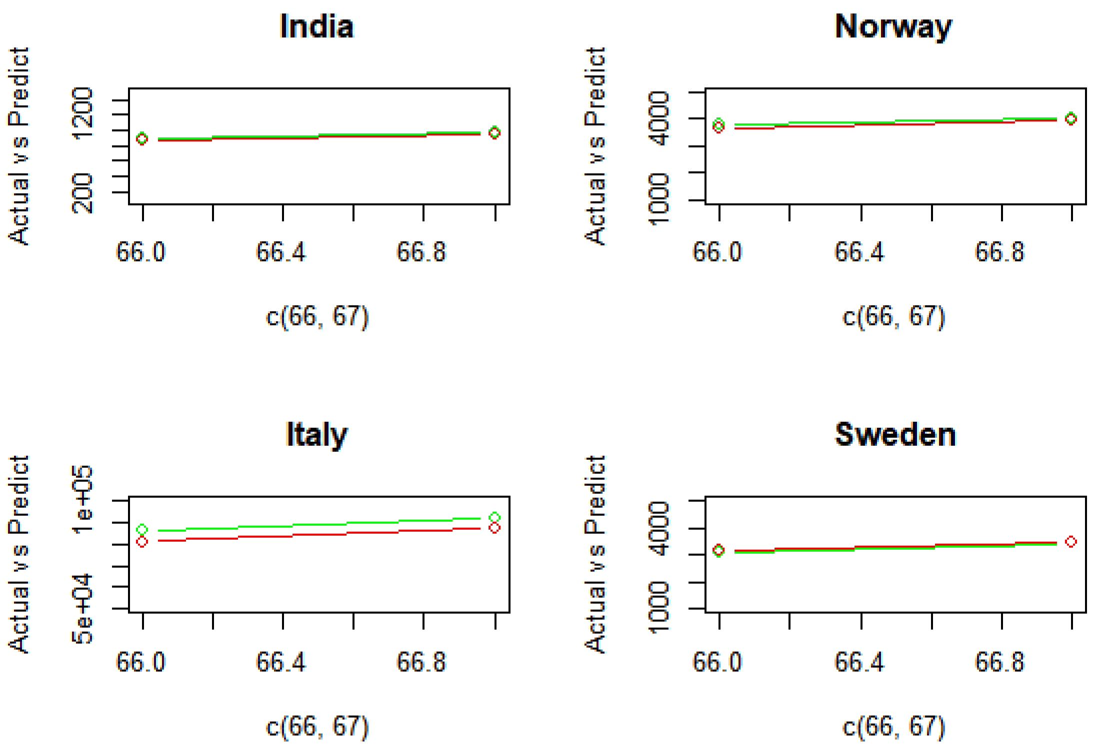
Predicted Corona Cases

India	Norway	Sweden	Italy	Brazil	Belgium	Israel	Iraq	Spain	Malaysia
860	3660	3176	81167	3483	7960	3137	428	66930	2284
950	3964	3495	87321	3965	10043	3565	471	76014	2527

From ARIMA Model we are getting closer predictions.

Actual vs Predicted

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

plot(c(66,67),c(887,987),xlim = c(66,67),ylim = c(100,1500),"b",col="green", ylab="Actual vs Predict",main = "India" )

lines(c(66,67),c(860,950),"b",col="red")

plot(c(66,67),c(3771,4015),xlim = c(66,67),ylim = c(1000,5000),"b",col="green", ylab="Actual vs Predict",main = "Norway" )

lines(c(66,67),c(3660,3964),"b",col="red")

plot(c(66,67),c(86498,92472),xlim = c(66,67),ylim = c(50000,100000),"b",col="green", ylab="Actual vs Predict",main = "Italy" )

lines(c(66,67),c(81167,87321),"b",col="red")

plot(c(66,67),c(3069,3447),xlim = c(66,67),ylim = c(1000,5000),"b",col="green", ylab="Actual vs Predict",main = "Sweden" )

lines(c(66,67),c(3176,3495),"b",col="red" )
```



From plots we can see our model is almost perfectly predicting

RMSE

library(Metrics)

rmse(c(887,987),c(860,950)) # India

rmse(c(3771,4015),c(3660,3964)) #Norway

rmse(c(3069,3447),c(3176,3495)) # Sweden

```
> library(Metrics)
> rmse(c(887,987),c(860,950))
[1] 32.38827
> rmse(c(3771,4015),c(3660,3964))
[1] 86.37708
> rmse(c(3069,3447),c(3176,3495))
[1] 82.92466
```

Cleaning little more

a<-dataset$India[35:65]

x1<-ts(a)

model1<-auto.arima(x1)

forecast(model1,4)

b<-dataset$Italy[35:65]

x2<-ts(b)

model2<-auto.arima(x2)

forecast(model2,4)

```
> a<-dataset$India[35:65]
> x1<-ts(a)
> model1<-auto.arima(x1)
> forecast(model1,4)
   Point Forecast     Lo 80     Hi 80    Lo 95     Hi 95
32       837.8633  816.5941  859.1324 805.3349  870.3916
33       915.9852  882.7820  949.1884 865.2053  966.7651
34      1020.3409  965.7135 1074.9683 936.7955 1103.8862
35      1103.6770 1029.4056 1177.9484 990.0887 1217.2654
> b<-dataset$Italy[35:65]
> x2<-ts(b)
> model2<-auto.arima(x2)
> forecast(model2,4)
   Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
32       79118.24 77117.10 81119.38 76057.76  82178.72
33       83850.49 80446.85 87254.13 78645.07  89055.90
34       88582.73 83696.85 93468.61 81110.43  96055.04
35       93314.98 86834.47 99795.48 83403.90 103226.06
>
```

If we see our dataset in initial dates, we have very less corona infected people (0 or in single digit). Actually those columns are affecting our accuracy.So I took past month corona cases and I predicted values. It gave closer predictions than before .

Changing Order values

a<-dataset$Italy[35:65]

x1<-ts(a)

model1<-auto.arima(x1)

forecast(model1,4)

model1<-arima(x1,order=c(1,1.3,0))

forecast(model1,4)

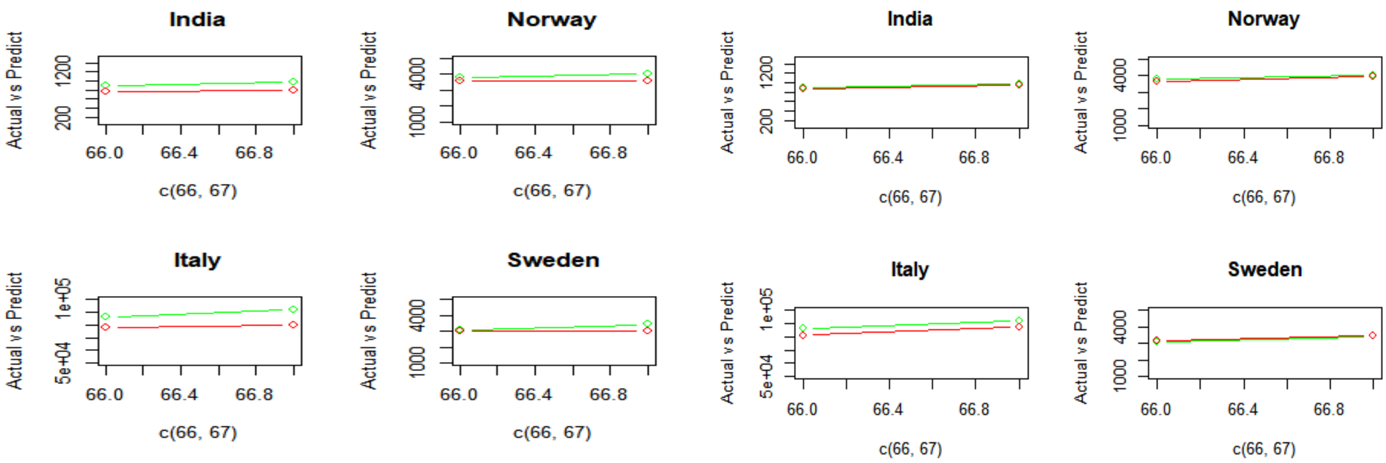
model1<-arima(x1,order=c(1,2,0.3))

forecast(model1,4)

```
> a<-dataset$Italy[35:65]
> x1<-ts(a)
> model1<-auto.arima(x1)
> forecast(model1,4)
   Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
32       79118.24 77117.10 81119.38 76057.76  82178.72
33       83850.49 80446.85 87254.13 78645.07  89055.90
34       88582.73 83696.85 93468.61 81110.43  96055.04
35       93314.98 86834.47 99795.48 83403.90 103226.06
> model1<-arima(x1,order=c(1,1.3,0))
> forecast(model1,4)
   Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
32       78823.79 76574.93 81072.66 75384.45  82263.13
33       82603.83 77871.00 87336.66 75365.59  89842.07
34       85823.61 78340.71 93306.50 74379.50  97267.71
35       88566.16 78191.67 98940.65 72699.75 104432.57
> model1<-arima(x1,order=c(1,2,0.3))
> forecast(model1,4)
   Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
32       79612.83 77524.89  81700.76 76419.60  82806.05
33       84832.39 80948.34  88716.45 78892.24  90772.54
34       90055.09 83840.83  96269.35 80551.20  99558.98
35       95276.44 86456.43 104096.44 81787.41 108765.47
>
```

If we change order values(p,q,d) we are getting good predictions. But above output screenshot we can see that auto.arima is taking order automatically.Internally it will try with all possible permutations of order values and it will select best order values which gives less AIC and BIC values.As we know here we are predicting only 4 upcoming corona cases values so concluding auto.arima predictions are giving less accuracy than other orders is wrong way according to my observations. So I think taking auto.arima is good for this dataset.

Holtwinters VS Arima



Above plots are clearly saying that ARIMA model is giving closer predictions when compared with Holtwinters model. For my dataset Holtwinters model is not suitable.Because as we saw in page:7 we don't have proper seasonality and because of that we can't give beta and gamma values .But prediction will depend on those 2 coefficients too.so we are getting low accuracy for it. Here I'm concluding ARIMA is best method for this dataset than Holtwinters method.

Detrending

x1<-detrend(dataset$India,"constant")

a<-ts(x1)

model1<-auto.arima(a)

forecast(model1,4)

x1<-detrend(dataset$India,"linear")

a<-ts(x1)

model1<-auto.arima(a)

forecast(model1,4)

x1<-dataset$India

a<-ts(x1)

model1<-auto.arima(a)

forecast(model1,4)

```
> x1<-detrend(dataset$India,"constant")
> a<-ts(x1)
> model1<-auto.arima(a)
> forecast(model1,4)
   Point Forecast     Lo 80     Hi 80    Lo 95     Hi 95
66       765.0212 750.7376  779.3049 743.1763  786.8662
67       842.5972 820.4879  864.7064 808.7840  876.4103
68       948.1404 911.6438  984.6371 892.3236 1003.9573
69      1030.7920 981.3416 1080.2424 955.1641 1106.4199
> x1<-detrend(dataset$India,"linear")
> a<-ts(x1)
> model1<-auto.arima(a)
> forecast(model1,4)
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
66       582.8866 568.6030 597.1703 561.0417 604.7316
67       654.9433 632.8341 677.0526 621.1301 688.7565
68       754.9674 718.4707 791.4641 699.1505 810.7843
69       832.0997 782.6492 881.5502 756.4717 907.7277
> x1<-dataset$India
> a<-ts(x1)
> model1<-auto.arima(a)
> forecast(model1,4)
   Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
66       838.7443  824.4607  853.0279  816.8994  860.5892
67       916.3202  894.2110  938.4294  882.5071  950.1334
68      1021.8635  985.3669 1058.3602  966.0467 1077.6803
69      1104.5151 1055.0647 1153.9655 1028.8873 1180.1429
>
```

Trending is removing fluctuations in data and in case if we detrend our model then fluctuations will increase and we will get bad predictions as shown in above output.

Future predictions

```
x1<-ts(dataset$India[20:65])
x2<-ts(dataset$Italy[20:65])
x3<-ts(dataset$Sweden[20:65])
x4<-ts(dataset$Norway[20:65])
x5<-ts(dataset$Spain[20:65])

model1<-auto.arima(x1)
model2<-auto.arima(x2)
model3<-auto.arima(x3)
model4<-auto.arima(x4)
model5<-auto.arima(x5)

forecast(model1,7)
forecast(model2,7)
forecast(model3,7)
forecast(model4,7)
forecast(model5,7)
```

Country	27 Mar,2020	28 Mar,2020	29 Mar,2020	30 Mar,2020	31 Mar,2020	1 April,2020	2 April,2020
India	864	956	1088	1195	1327	1443	1577
Italy	81584	88035	94581	101263	108086	115046	122138
sweden	3197	3560	3937	4327	4728	5140	5560
Norway	3676	3992	4363	4742	5128	5535	5953
spain	67195	76487	86111	95951	105996	116226	126626

Source : [CoronaWorldmeter]

Predictive Analysis

- According to our predictions spreading of virus is very fast .
- So if it continuous then the world will fall in danger.

Prescriptive analysis

- Wash your hands regularly for 20 seconds, with soap and water or alcohol-based hand rub
- Cover your nose and mouth with a disposable tissue or flexed elbow when you cough or sneeze
- Avoid close contact (1 meter or 3 feet) with people who are unwell
- Stay home and self-isolate from others in the household if you feel unwell
- Don't Touch your eyes, nose, or mouth if your hands are not clean