Applying Time Series Models

On

New Cases of COVID-19

In Bangladesh

Sharmin Hossain

Student ID: 1337949

11 May 2022

Contents

Introduction

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by a virus, the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China, in December 2019. On March 11, 2020, the World Health Organization (WHO) declared the novel coronavirus (COVID-19) outbreak as a global pandemic. On 8 March, Bangladesh has confirmed 3 laboratories tested coronavirus cases for the very first time. To reduce the transmission rate in Bangladesh, the government declared lockdown throughout the nation from March 23, 2020, to various lengths. Bangladesh is one of the most densely populated globally. For this reason, transmission rate of COVID-19 was increasing day by day.

The prediction of the rate of infection of COVID-19 has become vital for decision and policy makers worldwide. A prediction of the number of infections would assist policy makers in a specific region to assess their current healthcare capacity and decide which measures need to be taken to curb and control the spread of COVID-19.

Time series analysis (TSA) is a statistical technique that consists of data points listed in time order. This technique is suitable for research questions such as forecasting future events. The reason why time series analysis exists, is since the outcome variable in our model is dependent on one single explanatory variable only: time.

In this paper, I have tried to create a model to understand the weekly new cases in near future based on the past infected data of Bangladesh. I have tried with three time series models here; They are general Exponential Smoothing method, Holt’s Exponential Smoothing, Autoregressive integrated moving average (ARIMA). I have tried to understand which model worked better based on their accuracy and residuals error.

Datasets

Dataset for my modeling was collected from World Health Organization (WHO) coronavirus page [6]. The data consisted of several variables such as date reported, country, new cases, cumulative cases, new deaths, cumulative deaths etc. For my paper, I have sorted out my country's information in a separate file and worked with that. I have also worked with only "new cases" column as the purpose of the paper is to find the prediction of new infected cases in Bangladesh in a weekly basis. The timeline for the dataset of Bangladesh started from 08 March 2020 and I have taken data till 09 May 2022.

Methods Used

General Exponential Smoothing Method: ETS (Error, Trend, Seasonal) method is an approach method for forecasting time series univariate. This ETS model focuses on trend and seasonal components. The flexibility of the ETS model lies in its ability to trend and seasonal components of different traits. ETS is estimating both the initial states and smoothing parameters by optimizing the likelihood function (which is only equivalent to optimizing the MSE for the linear additive models). ETS searches over a restricted parameter space to ensure the resulting model is forecastable.

Holt's Winters Exponential Smoothing Method: The Holt-Winters method uses exponential smoothing to encode lots of values from the past and use them to predict "typical" values for the present and future. Holt's Winters is using heuristic values for the initial states and then estimating the smoothing parameters by optimizing the MSE.

Autoregressive integrated moving average (ARIMA): Auto-Regressive Integrated Moving Average (ARIMA), is a time-series auto-regressive technique that calculates future short-term predictions from analyzing time-series of historical data. It uses auto-regression and moving average and incorporates a differencing order to remove trend and/or seasonality. The ARIMA model contains 3 parameters (p, d, q). Parameter p in the ARIMA model represents the periods to lag for. Parameter d represents the number of differencing transformations done to remove trend and/or seasonality therefore turning the time-series into a stationary one, i.e., making the mean and variance constant over time. Parameter q represents the lag of the error component of the ARIMA model. The error component is the part of the time-series that cannot be explained by trend or seasonality.

Data Preparation and Data Wrangling

After importing all the necessary libraries and reading the dataset, the first thing I did was to check out the different variable's summary and how the data generally looked like. Turns out, in the dataset, date was considered as a character type data.

```
covid <- read.csv("Bangladesh.csv")
summary(covid)

##  Date_reported      Country_code         Country           WHO_region
##  Length:793         Length:793         Length:793         Length:793
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    New_cases     Cumulative_cases    New_deaths      Cumulative_deaths
##  Min.   :    0   Min.   :      3   Min.   : -6.00   Min.   :    0
##  1st Qu.:  396   1st Qu.: 352178   1st Qu.:  6.00   1st Qu.: 5007
##  Median : 1470   Median : 666132   Median : 22.00   Median : 9521
##  Mean   : 2463   Mean   : 879287   Mean   : 36.73   Mean   :14097
##  3rd Qu.: 2907   3rd Qu.:1567417   3rd Qu.: 39.00   3rd Qu.:27814
##  Max.   :16230   Max.   :1952776   Max.   :264.00   Max.   :29127

glimpse(covid)

## Rows: 793
## Columns: 8
## $ Date_reported     <chr> "08-03-20", "09-03-20", "10-03-20", "11-03-20", "12-~
## $ Country_code      <chr> "BD", "BD", "BD", "BD", "BD", "BD", "BD", "BD", "BD"~
## $ Country           <chr> "Bangladesh", "Bangladesh", "Bangladesh", "Banglades~
## $ WHO_region        <chr> "SEARO", "SEARO", "SEARO", "SEARO", "SEARO", "SEARO"~
## $ New_cases         <int> 3, 4, 0, 0, 0, 0, 0, 0, 1, 1, 1, 7, 0, 7, 0, 9, 6, 0~
## $ Cumulative_cases  <int> 3, 7, 7, 7, 7, 7, 7, 7, 8, 9, 10, 17, 17, 24, 24, 33~
## $ New_deaths        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1~
## $ Cumulative_deaths <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 3, 4, 5~
```

Figure 1: Summary of Dataset

It will not be possible to consider date as character and make a time series with it. So, I changed the data type to be date using a R function.

Next, I checked if there were any missing values in the dataset for any of the columns. Luckily, there were no missing values there.

Then, I created the timeseries object for the dataset and plot it.

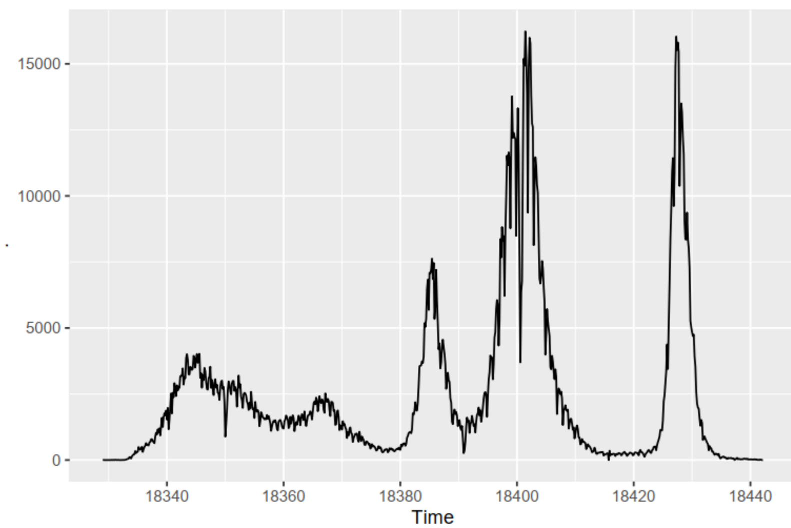


Figure 2: Plot of New Cases

Then I tried to decompose the time series model I just created. The decompose function in R decomposes a time series into seasonal, trend and irregular components using moving averages. Also Deals with additive or multiplicative seasonal component.

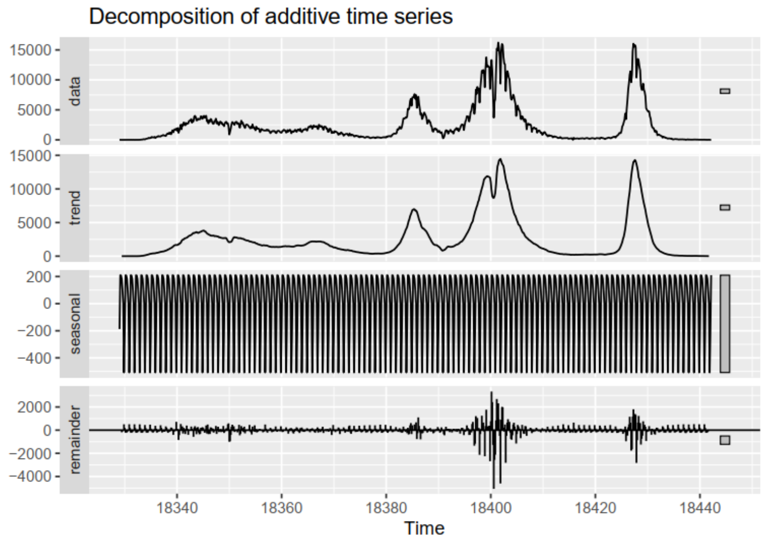


Figure 3: Decomposition of Time Series

Cross Validation

Before doing forecasting, I split my data into training and testing dataset.

Test data is the data of last 7 days (1 week), and the rest is train data.

Model Implementation & Result

From general exponential smoothing, I can see that there is no trend for this dataset. It's an "ANA model" that means error & season type are additive and trend is not present here.

```r
#ETS Model
covid_ets <- ets(y = train, model = "ZZZ")
covid_ets
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = train, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 0.9999
##     gamma = 1e-04
##
##   Initial states:
##     l = 51.5322
##     s = -512.2099 -0.1892 98.2421 182.1805 210.1997 211.5393
##            -189.7625
##
##   sigma:  667.05
##
##      AIC     AICc      BIC
## 15473.68 15473.96 15520.35
```

Figure 4: General ETS Method

Holt Winters method showed that there is trend but no seasonal component.

```r
#Holt Model
covid_holt <- HoltWinters(x = train, gamma = F)
covid_holt
```

8

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = train, gamma = F)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0
##  gamma: FALSE
##
## Coefficients:
##   [,1]
## a   10
## b    1
```

Figure 5: Holt Winters Method

For ARIMA, first I needed to check the stationary. From Augmented Dickey Fuller test, I can see that the null hypothesis can be rejected as p value was less than 0.05. So, the dataset is stationary.

After that, to choose p & q values for ARIMA model, I plotted the ACF (Autocorrelation Function) & PACF (Partial Autocorrelation Function) plot. From the plot, we can clearly decide on the values of q to be 1 based on PACF part. But for p, we can try with different combinations. Here I have tried with 3 values of p (1,2,3) to test which one works better. And for d value, it will be zero because I did not have to differentiate it to make it stationary.

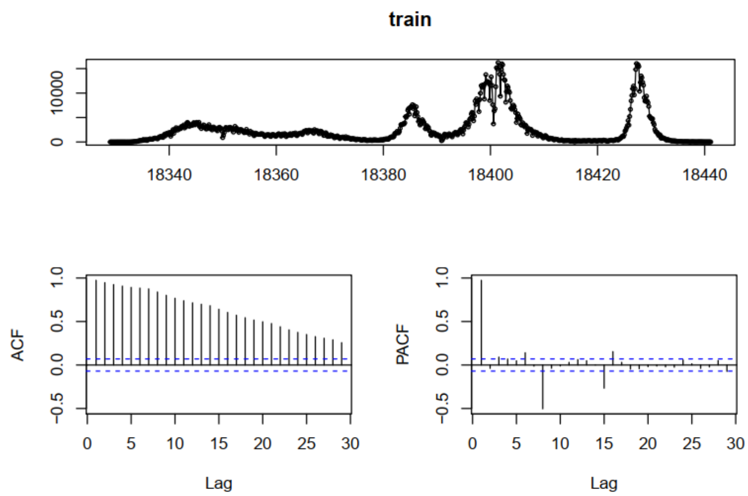


Figure 6: ACF & PACF Plot

And for ARIMA, I have also tried to use the auto Arima function to see if it worked better with my dataset.

```r
#Testing ARIMA modeling
covid_arima1 <- Arima(y = train, order = c(1,0,1))
covid_arima2 <- Arima(y = train, order = c(1,0,2))
covid_arima3 <- Arima(y = train, order = c(1,0,3))

#Testing Auto ARIMA
covid_arima_auto <- auto.arima(y = train)
covid_arima_auto
```

```
## Series: train
## ARIMA(1,1,3)(1,0,1)[7]
##
## Coefficients:
##          ar1     ma1     ma2      ma3   sar1     sma1
##      -0.9578  0.9986  0.0255  -0.0415  0.807  -0.3954
## s.e.  0.0279  0.0452  0.0507   0.0361  0.034   0.0544
##
## sigma^2 estimated as 346509:  log likelihood=-6119.06
## AIC=12252.11   AICc=12252.26   BIC=12284.77
```

Figure 7: Auto ARIMA Model

Based on the AIC value of exponential smoothing and ARIMA models (auto and manual), it seemed that Auto ARIMA model worked best.

Then I tried to forecast the model with my test dataset. For 95% confidence interval, the values did not differ much for those 3 models but point forecast was weird for Holt Winters method. For accuracy, considering Mean error, ARIMA auto method did the best here.

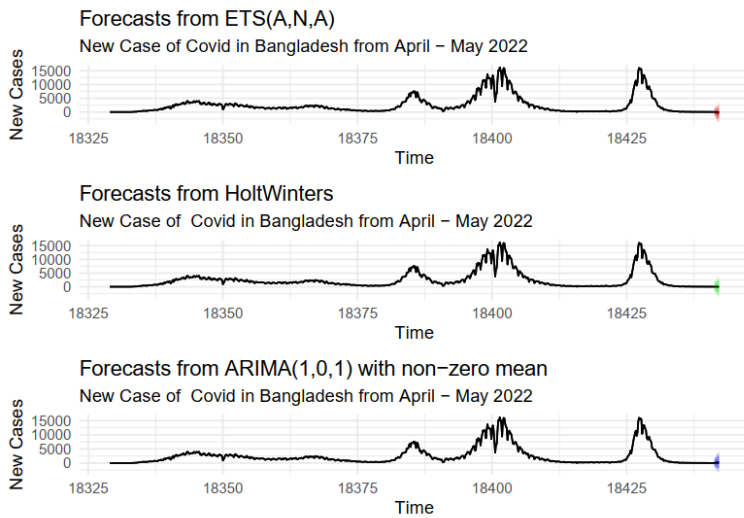


Figure 8: Forecasts Plot of Different Models

```
##          Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 18441.29       8.740591  -846.1184  863.5996 -1298.653 1316.135
## 18441.43     -19.243960 -1228.1319 1189.6439 -1868.078 1829.591
## 18441.57    -103.193359 -1583.7459 1377.3592 -2367.503 2161.116
## 18441.71    -201.651852 -1911.2313 1507.9276 -2816.228 2412.924
## 18441.86    -713.607019 -2624.9646 1197.7506 -3636.776 2209.562
## 18442.00    -391.204816 -2484.9846 1702.5750 -3593.365 2810.955
## 18442.14      10.003313 -2251.5639 2271.5705 -3448.765 3468.772
```

```
covid_holt_f
```

```
##          Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 18441.29             11  -911.2149  933.2149 -1399.406 1421.406
## 18441.43             12 -1292.2088 1316.2088 -1982.615 2006.615
## 18441.57             13 -1584.3231 1610.3231 -2429.895 2455.895
## 18441.71             14 -1830.4298 1858.4298 -2806.812 2834.812
## 18441.86             15 -2047.1352 2077.1352 -3138.764 3168.764
## 18442.00             16 -2242.9560 2274.9560 -3438.775 3470.775
## 18442.14             17 -2422.9513 2456.9513 -3714.584 3748.584
```

```
covid_arima_f
```

```
##          Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 18441.29       72.93587  -842.9466  988.8184 -1327.786 1473.657
## 18441.43      137.94148 -1170.7657 1446.6487 -1863.553 2139.436
## 18441.57      201.11518 -1392.0041 1794.2345 -2235.351 2637.581
## 18441.71      262.50858 -1558.8873 2083.9044 -2523.076 3048.093
## 18441.86      322.17186 -1691.1896 2335.5333 -2756.999 3401.342
## 18442.00      380.15378 -1799.0341 2559.3417 -2952.627 3712.934
## 18442.14      436.50171 -1888.4634 2761.4669 -3119.226 3992.229
```

Figure 9: Forecasts of Different Models

Conclusion

Based on both training and testing accuracy, ARIMA model with auto function did best for my dataset. The future work of this research will focus on improving the performance of my model by using a huge data and applying the proposed model to more countries. Also, the further work on this dataset can be done by predicting death rates using time series model.

Although ARIMA was used in epidemiology predictions for past diseases as well as the current pandemic of COVID-19, it has not been relied on heavily because it is regarded as unsuitable to use in complex and dynamic situations. In the end there is no perfect model. As a famous British statistician George E.P Box once said,

“All models are wrong, but some are useful.”

References

1. On the accuracy of ARIMA based prediction of COVID-19 spread (https://www.sciencedirect.com/science/article/pii/S2211379721006197#b45)
2. Time Series Analysis in R - Decomposing Time Series (https://rpubs.com/davoodastaraky/TSA1)
3. Time Series Analysis: Identifying AR and MA using ACF and PACF Plots (https://towardsdatascience.com/identifying-ar-and-ma-terms-using-acf-and-pacf-plots-in-time-series-forecasting-ccb9fd073db8)
4. Predicting number of Covid19 deaths using Time Series Analysis (ARIMA MODEL) (https://towardsdatascience.com/predicting-number-of-covid19-deaths-using-time-series-analysis-arima-model-4ad92c48b3ae)
5. ARMA Modelling of Covid-19 Cases in Indonesia (https://rpubs.com/dyanafam/841563)
6. WHO Dataset Collection (https://covid19.who.int/data)

Appendix

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.5

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(TTR)
```

```
## Warning: package 'TTR' was built under R version 4.0.5
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```r
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.5
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.0.5

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
covid <- read.csv("Bangladesh.csv")
summary(covid)
```

```
##  Date_reported      Country_code         Country           WHO_region
##  Length:793         Length:793         Length:793         Length:793
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    New_cases     Cumulative_cases    New_deaths     Cumulative_deaths
##  Min.   :    0   Min.   :      3   Min.   : -6.00   Min.   :    0
##  1st Qu.:  396   1st Qu.: 352178   1st Qu.:  6.00   1st Qu.: 5007
##  Median : 1470   Median : 666132   Median : 22.00   Median : 9521
##  Mean   : 2463   Mean   : 879287   Mean   : 36.73   Mean   :14097
##  3rd Qu.: 2907   3rd Qu.:1567417   3rd Qu.: 39.00   3rd Qu.:27814
##  Max.   :16230   Max.   :1952776   Max.   :264.00   Max.   :29127
```

```r
glimpse(covid)
```

```
## Rows: 793
## Columns: 8
## $ Date_reported     <chr> "08-03-20", "09-03-20", "10-03-20", "11-03-20", "12-~
## $ Country_code      <chr> "BD", "BD", "BD", "BD", "BD", "BD", "BD", "BD", "BD"~
## $ Country           <chr> "Bangladesh", "Bangladesh", "Bangladesh", "Banglades~
## $ WHO_region        <chr> "SEARO", "SEARO", "SEARO", "SEARO", "SEARO", "SEARO"~
## $ New_cases         <int> 3, 4, 0, 0, 0, 0, 0, 0, 1, 1, 1, 7, 0, 7, 0, 9, 6, 0~
## $ Cumulative_cases  <int> 3, 7, 7, 7, 7, 7, 7, 7, 8, 9, 10, 17, 17, 24, 24, 33~
## $ New_deaths        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1~
## $ Cumulative_deaths <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 3, 4, 5~
```

```r
#Checking Missing Values
colSums(is.na(covid))
```

```
##     Date_reported      Country_code           Country        WHO_region
##                 0                 0                 0                 0
##         New_cases  Cumulative_cases        New_deaths Cumulative_deaths
##                 0                 0                 0                 0
```

```
#Convert Date From Character
covid$Date_reported <- as.Date(covid$Date_reported,"%d-%m-%y")

#Checking Type and Class
typeof(covid$Date_reported)
```

```
## [1] "double"
```

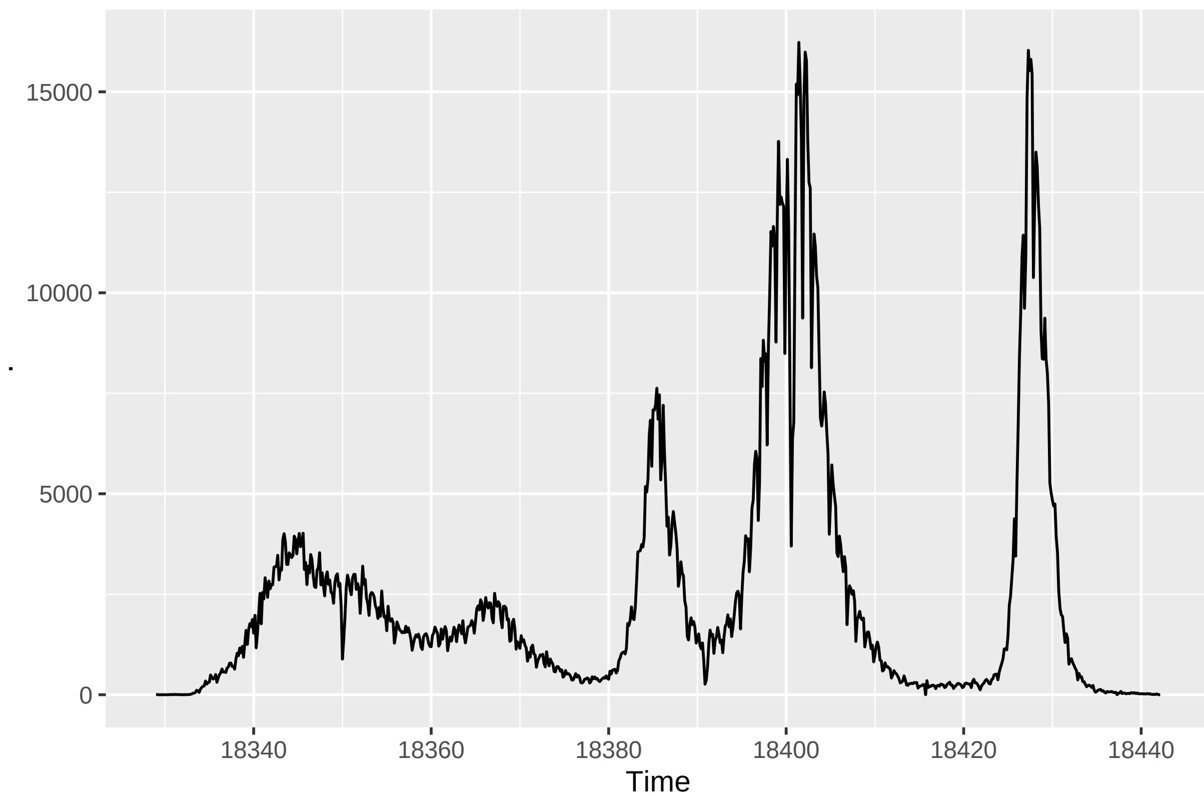
```
class(covid$Date_reported)
```

```
## [1] "Date"
```

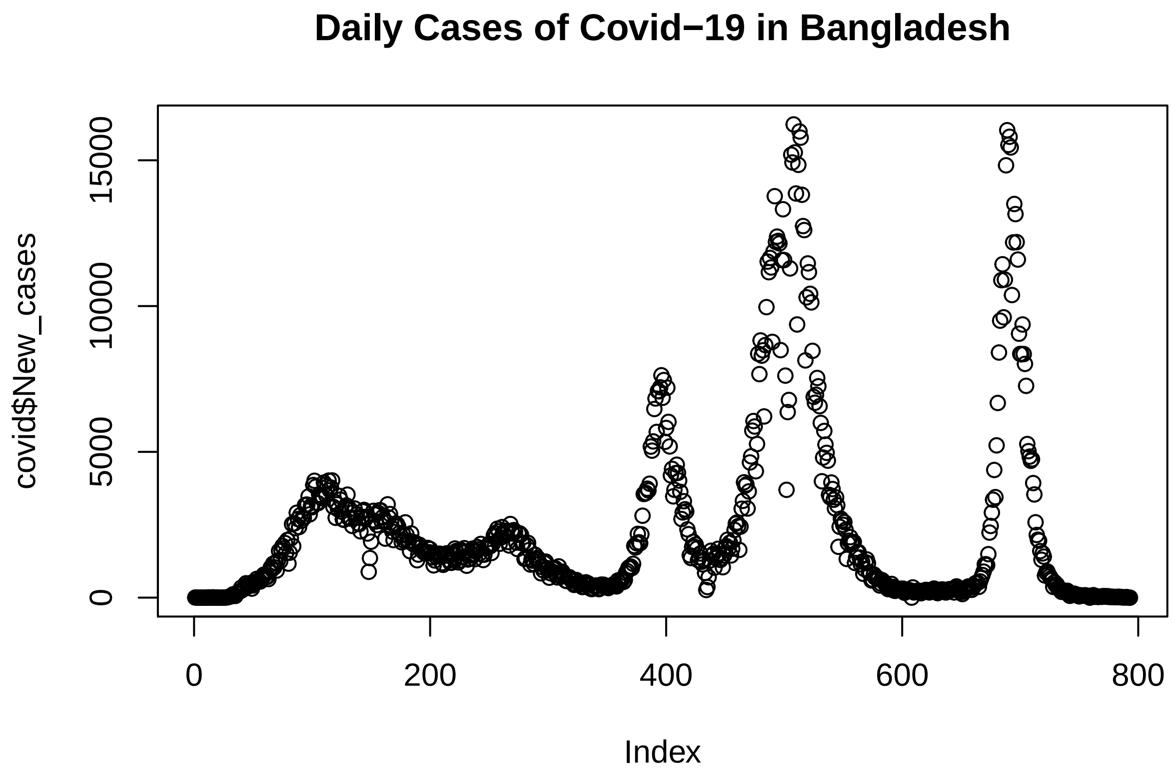
```
#Checking ranges of date variable
range(covid$Date_reported)
```

```
## [1] "2020-03-08" "2022-05-09"
```

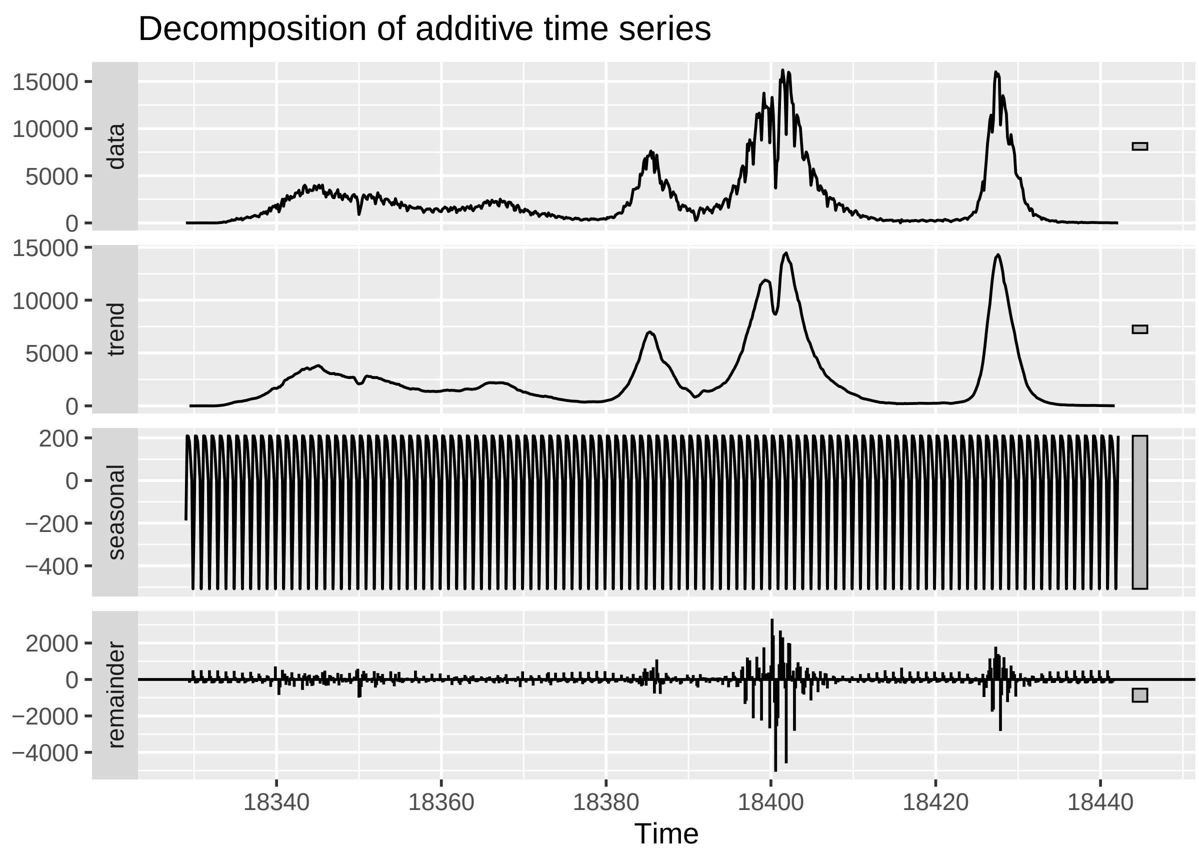
```
#create object ts
covid_ts <- ts(data = covid$New_cases,
               start = min(covid$Date_reported),
               frequency = 7) #weekly seasonality

#visualise object covid_ts
covid_ts %>% autoplot()
```



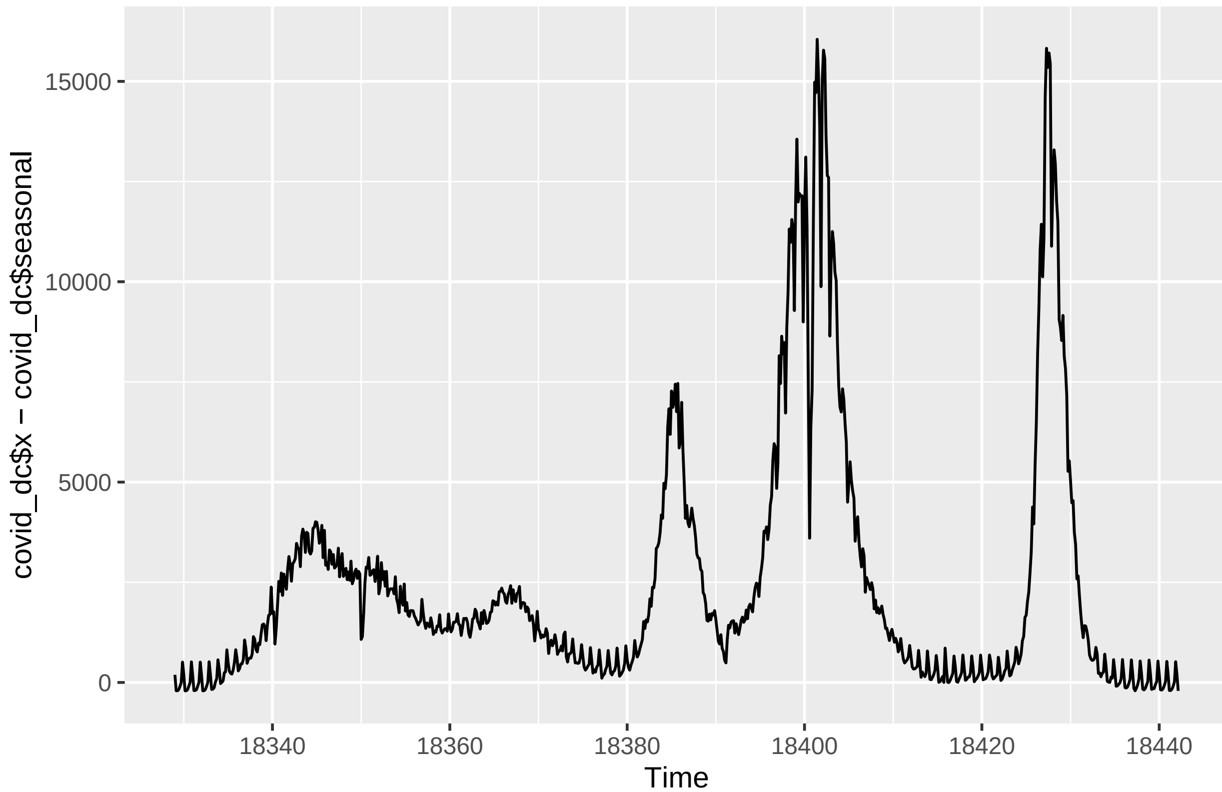

```
#Visualize New Cases
plot(covid$New_cases, main = "Daily Cases of Covid-19 in Bangladesh")
```



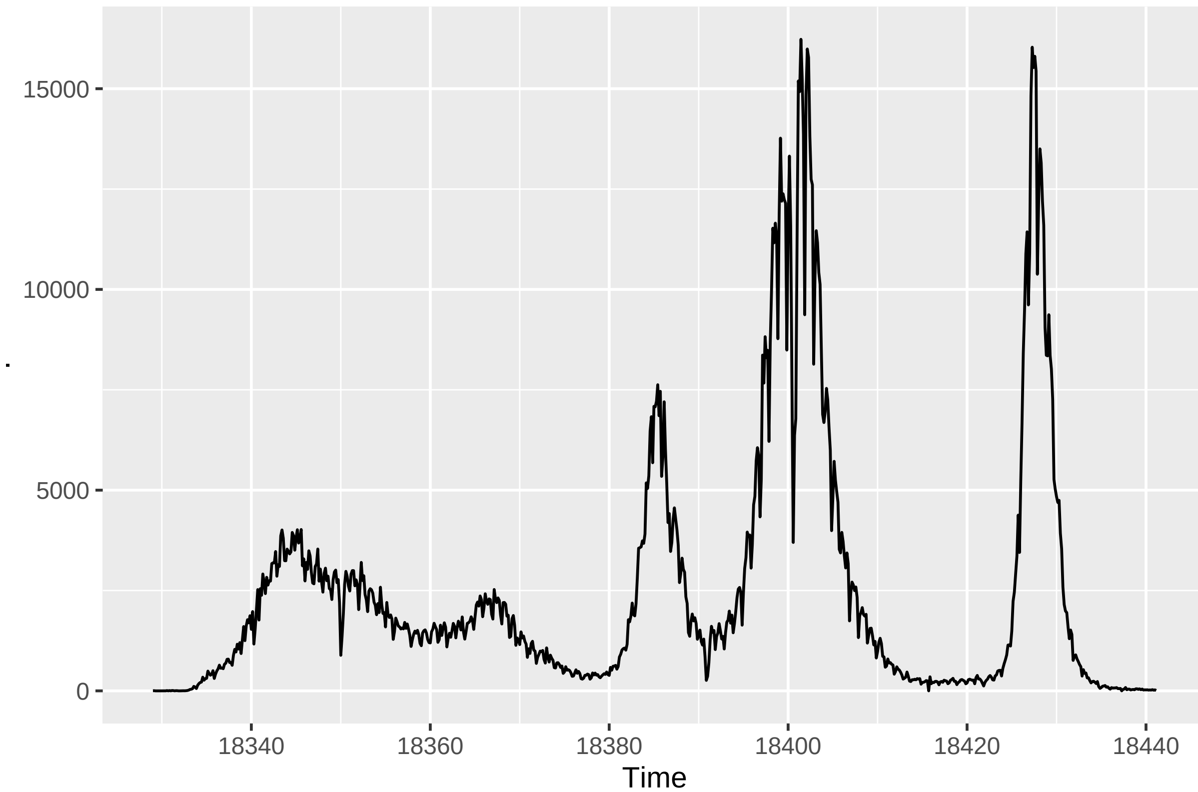

```
#Decompose TS
covid_dc <- decompose(covid_ts)
covid_dc %>% autoplot()
```




```
autoplot(covid_dc$x - covid_dc$seasonal)
```




```
#Setting Testing and Training data
test <- tail(covid_ts, 7) #get 7 last days
train <- head(covid_ts, length(covid_ts) - length(test)) #get the rest data

train %>% autoplot()
```




```r
#ETS Model
covid_ets <- ets(y = train, model = "ZZZ")
covid_ets
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = train, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 0.9999
##     gamma = 1e-04
##
##   Initial states:
##     l = 51.5322
##     s = -512.2099 -0.1892 98.2421 182.1805 210.1997 211.5393
##            -189.7625
##
##   sigma:  667.05
##
##      AIC     AICc      BIC
## 15473.68 15473.96 15520.35
```

```r
#Holt Model
covid_holt <- HoltWinters(x = train, gamma = F)
covid_holt
```

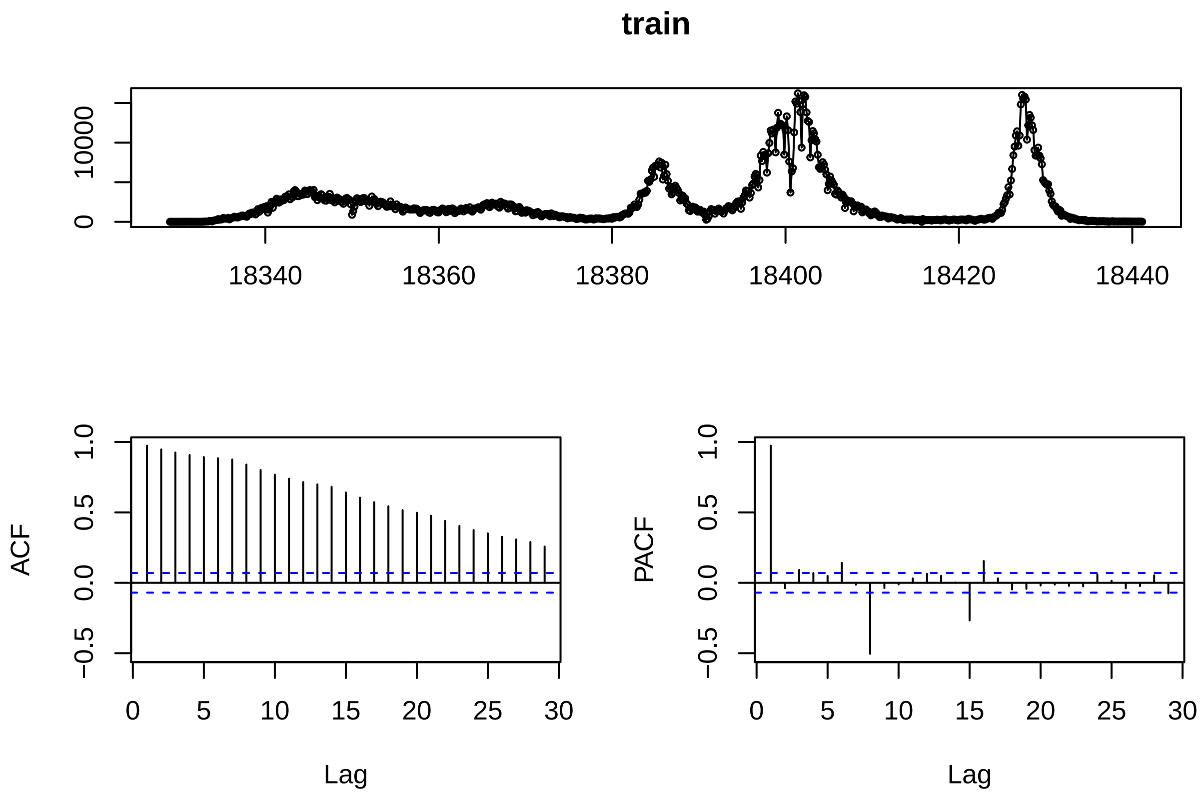
```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = train, gamma = F)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0
##  gamma: FALSE
##
## Coefficients:
##   [,1]
## a   10
## b    1
```

```
#Testing Stationarity
adf.test(train)
```

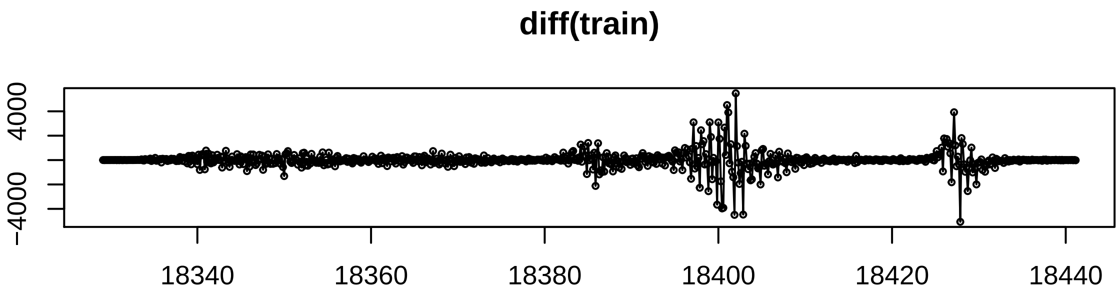
```
## Warning in adf.test(train): p-value smaller than printed p-value
```

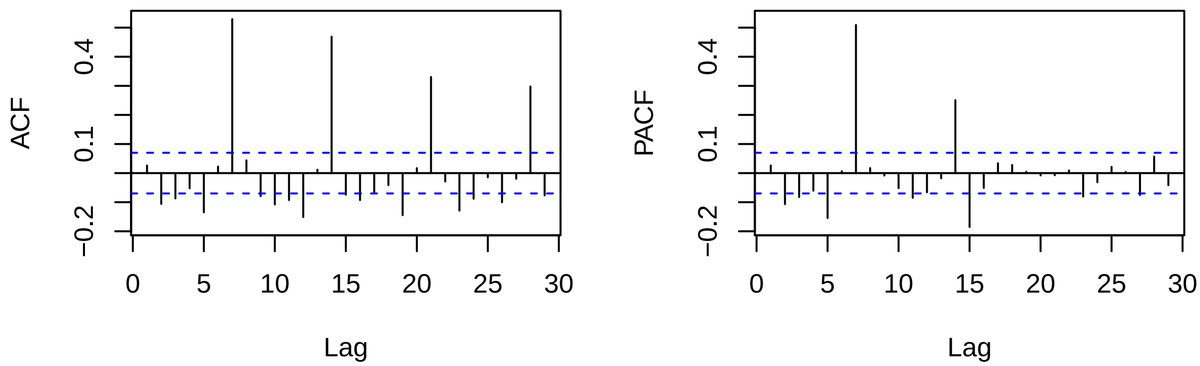
```
##
##  Augmented Dickey-Fuller Test
##
## data:  train
## Dickey-Fuller = -4.0373, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

```
#Plot of ACF & PACF
tsdisplay(train)
```




```
#Plot of ACF & PACF (Diff)
tsdisplay(diff(train))
```






```r
#Testing ARIMA modeling
covid_arima1 <- Arima(y = train, order = c(1,0,1))
covid_arima2 <- Arima(y = train, order = c(1,0,2))
covid_arima3 <- Arima(y = train, order = c(1,0,3))

#Testing Auto ARIMA
covid_arima_auto <- auto.arima(y = train)
covid_arima_auto
```

```
## Series: train
## ARIMA(1,1,3)(1,0,1)[7]
##
## Coefficients:
##           ar1     ma1     ma2      ma3   sar1     sma1
##       -0.9578  0.9986  0.0255  -0.0415  0.807  -0.3954
## s.e.   0.0279  0.0452  0.0507   0.0361  0.034   0.0544
##
## sigma^2 estimated as 346509:  log likelihood=-6119.06
## AIC=12252.11   AICc=12252.26   BIC=12284.77
```

```r
#Accuracy

accuracy(covid_ets)
```

```
##                       ME   RMSE      MAE MPE MAPE      MASE       ACF1
## Training set -0.3218918 663.22 355.8808 NaN  Inf 0.4493051 0.02240211
```

```r
accuracy(covid_arima1)
```

```
##                    ME     RMSE      MAE  MPE MAPE      MASE         ACF1
## Training set 2.225368 713.3017 342.9311 -Inf  Inf 0.4329559 -0.003235571
```

```r
accuracy(covid_arima2)
```

```
##                    ME     RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set 2.390689 710.2794 339.2293 -Inf  Inf 0.4282823 0.009282414
```

```r
accuracy(covid_arima3)
```

```
##                    ME     RMSE      MAE  MPE MAPE      MASE         ACF1
## Training set 1.828161 707.4105 338.4331 -Inf  Inf 0.4272772 -0.004163246
```

```r
accuracy(covid_arima_auto)
```

```
##                       ME     RMSE     MAE MPE MAPE      MASE          ACF1
## Training set -0.01705068 586.0228 285.705 NaN  Inf 0.3607071 -9.795246e-05
```

```r
#AIC
covid_ets$aic
```

```
## [1] 15473.68
```

```r
covid_arima1$aic
```

```
## [1] 12569.45
```

```r
covid_arima2$aic
```

```
## [1] 12564.79
```

```r
covid_arima3$aic
```

```
## [1] 12560.46
```

```r
covid_arima_auto$aic
```

```
## [1] 12252.11
```

```r
#Forecasting
covid_ets_f <- forecast(covid_ets, h = 7)
covid_holt_f <- forecast(covid_holt, h = 7)
covid_arima_f <- forecast(covid_arima1, h = 7)

covid_ets_f
```

```
##          Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 18441.29       8.740591  -846.1184  863.5996 -1298.653 1316.135
## 18441.43     -19.243960 -1228.1319 1189.6439 -1868.078 1829.591
## 18441.57    -103.193359 -1583.7459 1377.3592 -2367.503 2161.116
## 18441.71    -201.651852 -1911.2313 1507.9276 -2816.228 2412.924
## 18441.86    -713.607019 -2624.9646 1197.7506 -3636.776 2209.562
## 18442.00    -391.204816 -2484.9846 1702.5750 -3593.365 2810.955
## 18442.14      10.003313 -2251.5639 2271.5705 -3448.765 3468.772
```

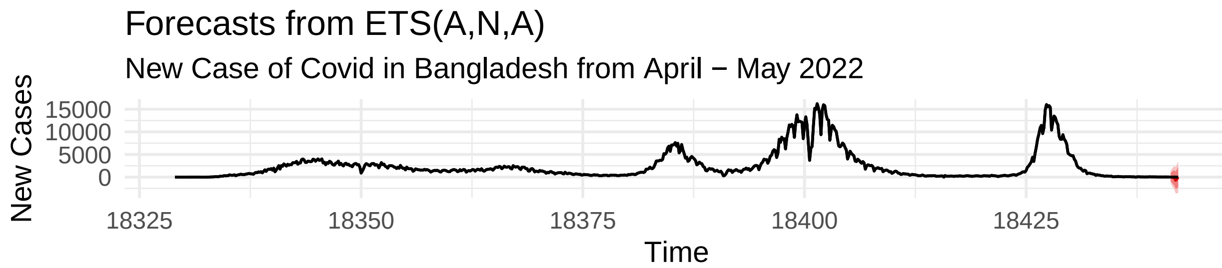
```r
covid_holt_f
```

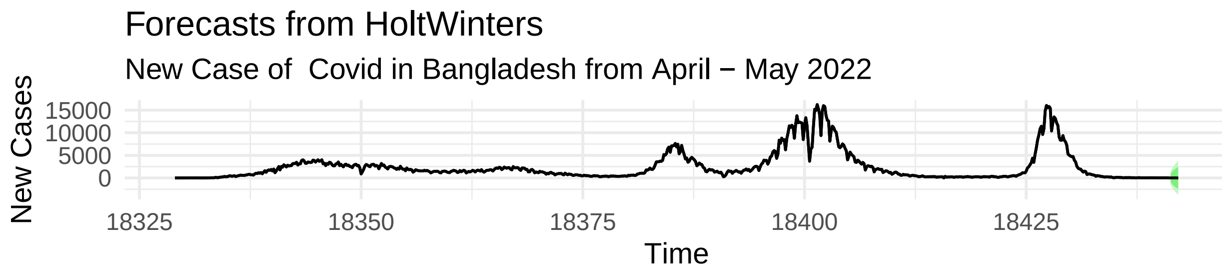
```
##          Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 18441.29             11  -911.2149  933.2149 -1399.406 1421.406
## 18441.43             12 -1292.2088 1316.2088 -1982.615 2006.615
## 18441.57             13 -1584.3231 1610.3231 -2429.895 2455.895
## 18441.71             14 -1830.4298 1858.4298 -2806.812 2834.812
## 18441.86             15 -2047.1352 2077.1352 -3138.764 3168.764
## 18442.00             16 -2242.9560 2274.9560 -3438.775 3470.775
## 18442.14             17 -2422.9513 2456.9513 -3714.584 3748.584
```

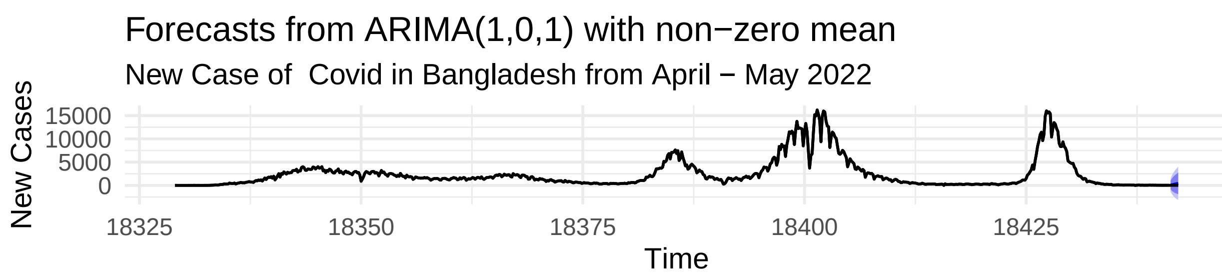
```r
covid_arima_f
```

```
##          Point Forecast      Lo 80     Hi 80     Lo 95    Hi 95
## 18441.29       72.93587  -842.9466  988.8184 -1327.786 1473.657
## 18441.43      137.94148 -1170.7657 1446.6487 -1863.553 2139.436
## 18441.57      201.11518 -1392.0041 1794.2345 -2235.351 2637.581
## 18441.71      262.50858 -1558.8873 2083.9044 -2523.076 3048.093
## 18441.86      322.17186 -1691.1896 2335.5333 -2756.999 3401.342
## 18442.00      380.15378 -1799.0341 2559.3417 -2952.627 3712.934
## 18442.14      436.50171 -1888.4634 2761.4669 -3119.226 3992.229
```

```r
#Plot of forecasting

a <- autoplot(covid_ets_f, series = "ETS", fcol = "red") +
  autolayer(covid_ts, series = "Actual", color = "black") +
  labs(subtitle = "New Case of Covid in Bangladesh from April - May 2022",
       y = "New Cases") +
  theme_minimal()

b <- autoplot(covid_holt_f, series = "HOLT", fcol = "green") +
  autolayer(covid_ts, series = "Actual", color = "black") +
  labs(subtitle = "New Case of  Covid in Bangladesh from April - May 2022",
       y = "New Cases") +
  theme_minimal()

c <- autoplot(covid_arima_f, series = "ARIMA", fcol = "blue") +
  autolayer(covid_ts, series = "Actual", color = "black") +
  labs(subtitle = "New Case of  Covid in Bangladesh from April - May 2022",
       y = "New Cases") +
  theme_minimal()

grid.arrange(a,b,c)
```








```r
#Accuracy of Forecasting
accuracy(covid_ets_f, test)
```

```
##                        ME     RMSE      MAE MPE MAPE      MASE       ACF1
## Training set  -0.3218918 663.2200 355.8808 NaN  Inf 0.4493051 0.02240211
## Test set     208.5938717 324.6502 211.9493 NaN  Inf 0.2675893 0.26552305
##              Theil's U
## Training set        NA
## Test set         19.72
```

```r
accuracy(covid_holt_f, test)
```

```
##                      ME       RMSE        MAE  MPE MAPE       MASE       ACF1
## Training set -0.9923469 719.149723 339.397959 -Inf  Inf 0.42849527 0.02607242
## Test set     -6.8571429   9.971388   8.285714 -Inf  Inf 0.01046084 0.20672007
##              Theil's U
## Training set        NA
## Test set     0.5666677
```

```r
accuracy(covid_arima_f, test)
```

```
##                        ME     RMSE      MAE  MPE MAPE      MASE         ACF1
## Training set    2.225368 713.3017 342.9311 -Inf  Inf 0.4329559 -0.003235571
## Test set     -251.904067 280.6547 251.9041 -Inf  Inf 0.3180329  0.571029803
```

```
##              Theil's U
## Training set        NA
## Test set      19.70011
```

```r
#Residuals
shapiro.test(covid_ets_f$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  covid_ets_f$residuals
## W = 0.72639, p-value < 2.2e-16
```

```r
shapiro.test(covid_holt_f$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  covid_holt_f$residuals
## W = 0.67157, p-value < 2.2e-16
```

```r
shapiro.test(covid_arima_f$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  covid_arima_f$residuals
## W = 0.66739, p-value < 2.2e-16
```

```r
#Box Plot
Box.test(covid_ets_f$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  covid_ets_f$residuals
## X-squared = 0.39596, df = 1, p-value = 0.5292
```

```r
Box.test(covid_holt_f$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  covid_holt_f$residuals
## X-squared = 0.53498, df = 1, p-value = 0.4645
```

```r
Box.test(covid_arima_f$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  covid_arima_f$residuals
## X-squared = 0.00826, df = 1, p-value = 0.9276
```