Gourav Singh Bajeli
AM.EN.U4AIE19031

Assignment-1
FTSE

●Select a unique stock (make sure it has not been selected by anyone else in the shared
sheet (first come basis)
● Download data from Jan 1, 2015 onwards to Dec 31 2021.
● In this Assignment you will apply all the forecasting techniques discussed thus far -Moving Averages, SES, Holt, Winter, etc.
● Analyze the data as a block from 2015-2021.
● Analyze data separately; 2015-2019 (pre-Covid) and 2019-2021 (post-Covid).
● Compare and contrast the performance across all scenarios using relevant metrics and also plot the error charts for each case.
● Summarize your findings in a report. You will need to submit R code (documented) + Dataset (CSV) + Report; as a ZIP file.

Gourav Singh Bajeli
AM.EN.U4AIE19031

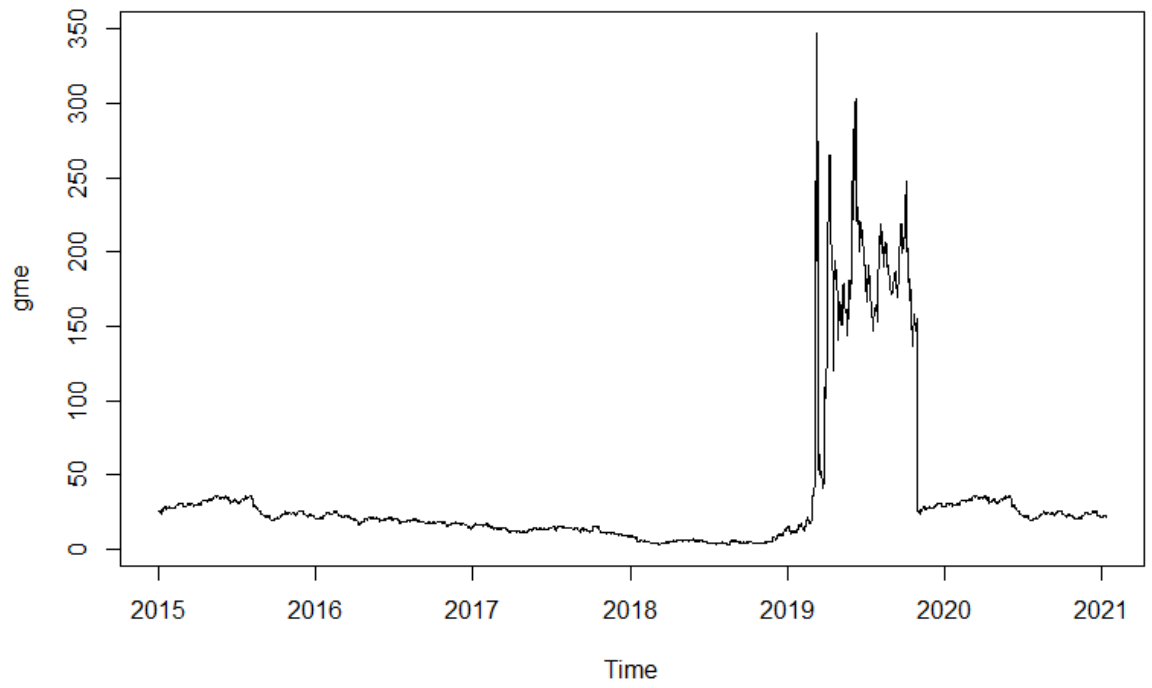
Libraries:

```
library(forecast)
library(dplyr)
library(quantmod)
library(ggplot2)
library(xts)
```

Importing the Data:

```
g <- getSymbols(c("GME"), src = "yahoo", from =
"2015-01-01", to = "2021-12-31",auto.assign=FALSE)
gme <- ts(g$GME.Adjusted, start=c(2015, 1), end=c(2021,
12),frequency=365)
```

Plot:



Gourav Singh Bajeli
AM.EN.U4AIE19031

Separated the time-series data into train and test data before forecasting it. The goal is to see how good the model is at predicting data that hasn't been used to train it.

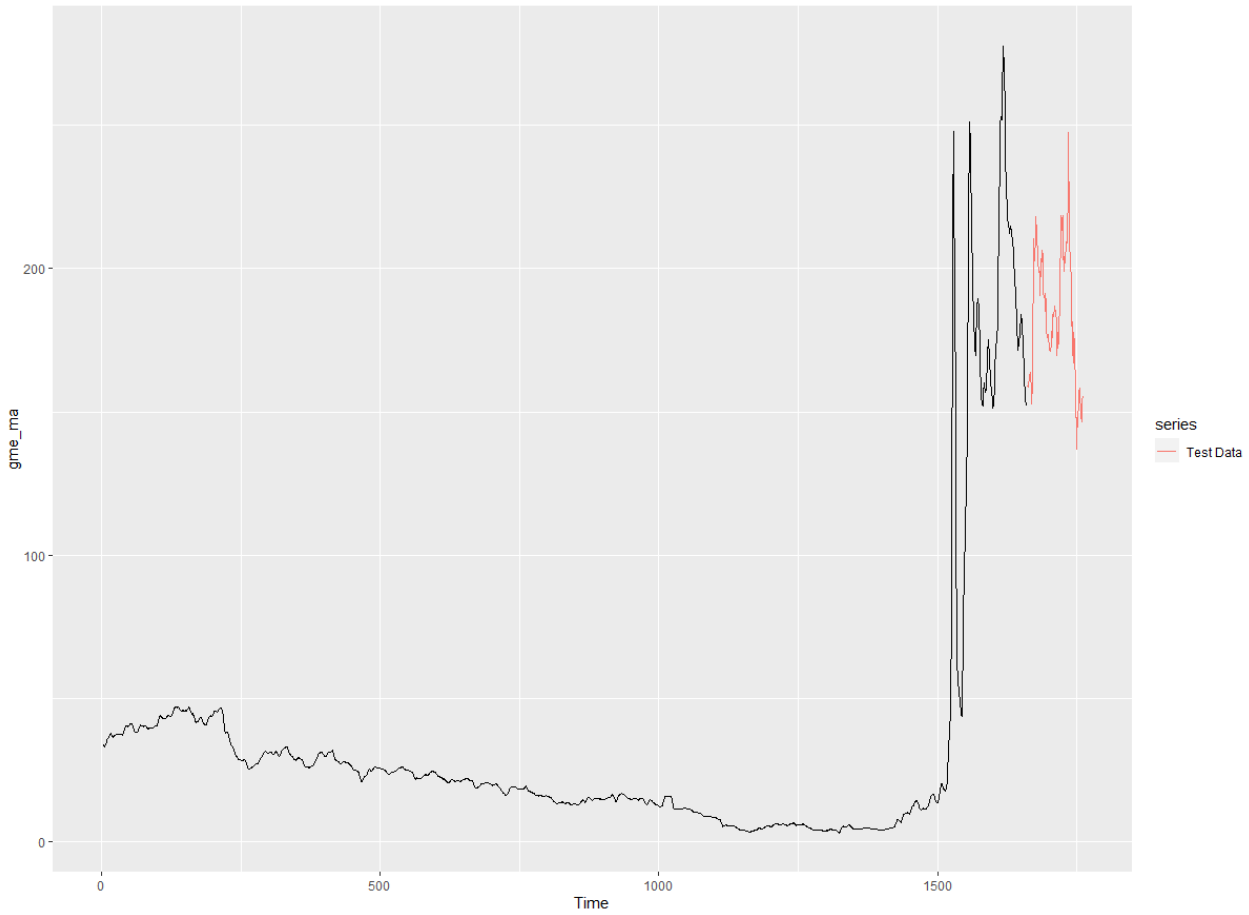
- Data used to fit the model is known as train data.
- Data used to evaluate the model is referred to as test data.

```
train <- head(Cl(g), length(Cl(g))-100)
test <- tail(Cl(g), 100)
```

Forecasting methods:

1. **Moving Average**

```
gme_ma <- ma(train,5)
autoplot(gme_ma)+autolayer(ts(test,start=length(train)),s
eries="Test Data")
summary(gme_ma)
```

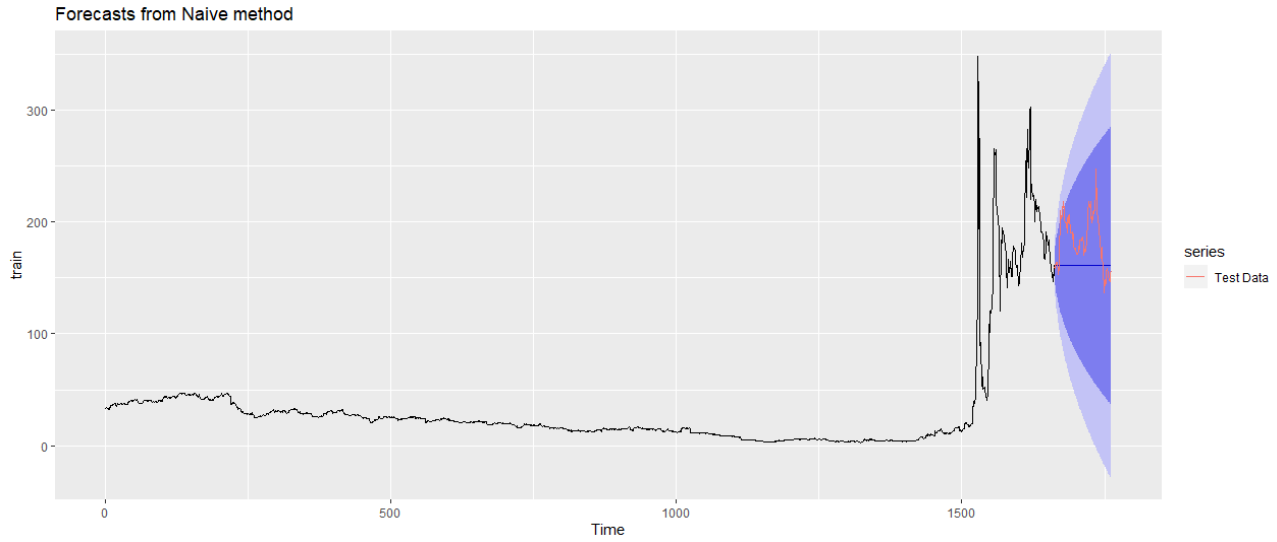



```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  3.052  11.390  20.071  32.499  31.067 277.822       4
```

2. Naive

```
gme_na <- naive(train,h=100)
autoplot(gme_na)+autolayer(ts(test,start=length(train)),s
eries="Test Data")
summary(gme_na)
```

The Naive Approach is a forecasting method that uses our data's most recent observation as the prediction result. In forecasting, it serves as the foundation model.



The dark and light darker regions reflect the 80 %and 95 % confidence intervals, respectively. The blue line is the mean of our projection. We can see disparities between the outcome and the real test results if we compare them.

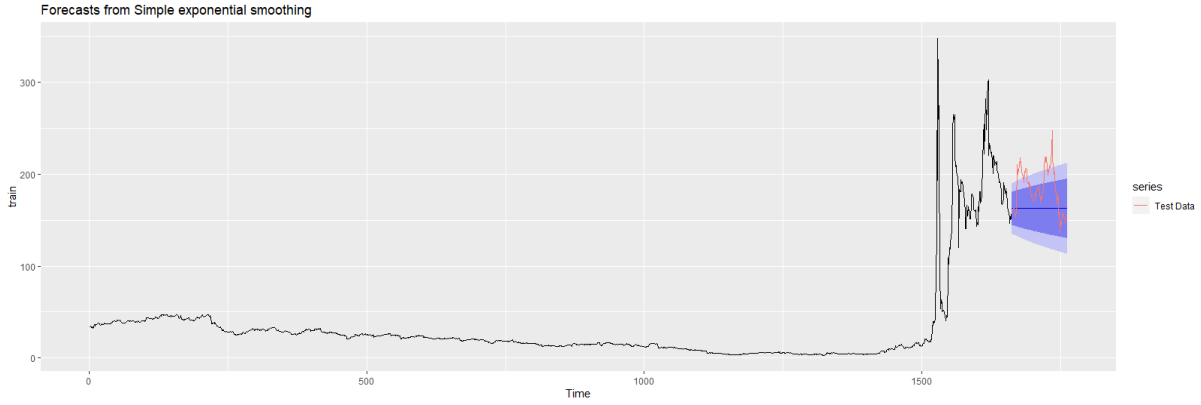
```
Forecast method: Naive method

Model Information:
Call: naive(y = train, h = 100)

Residual sd: 9.7061

Error measures:
                     ME     RMSE      MAE         MPE     MAPE MASE        ACF1
Training set 0.07665864 9.706133 1.700897 -0.1411091 3.126698    1 -0.2211963

Forecasts:
     Point Forecast     Lo 80    Hi 80       Lo 95    Hi 95
1663         161.13 148.69109 173.5689 142.1063332 180.1537
1664         161.13 143.53873 178.7213 134.2264703 188.0335
1665         161.13 139.58518 182.6748 128.1800388 194.0800
1666         161.13 136.25218 186.0078 123.0826613 199.1773
1667         161.13 133.31576 188.9443 118.5917816 203.6682
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

3. Simple Exponential Smoothing

```
gme_ses <- ses(train,h=100,alpha=0.15)
autoplot(gme_ses)+autolayer(ts(test,start=length(train)),
series="Test Data")
summary(gme_ses)
```

For data with no trend or seasonal structure, the Simple Exponential Smoothing technique is utilised. Among all the exponential smoothing approaches, the SES is the simplest. We know that in any sort of exponential smoothing, we give greater weight to recent values or observations than to older values or observations. A smoothing parameter, or alpha, determines the weight of each and every parameter. Alpha is a number that ranges from 0 to 1. In practise, SES will perform well if alpha is between 0.1 and 0.2. Because the method gives more weight to past data when alpha is close to 0, it is termed slow learning.



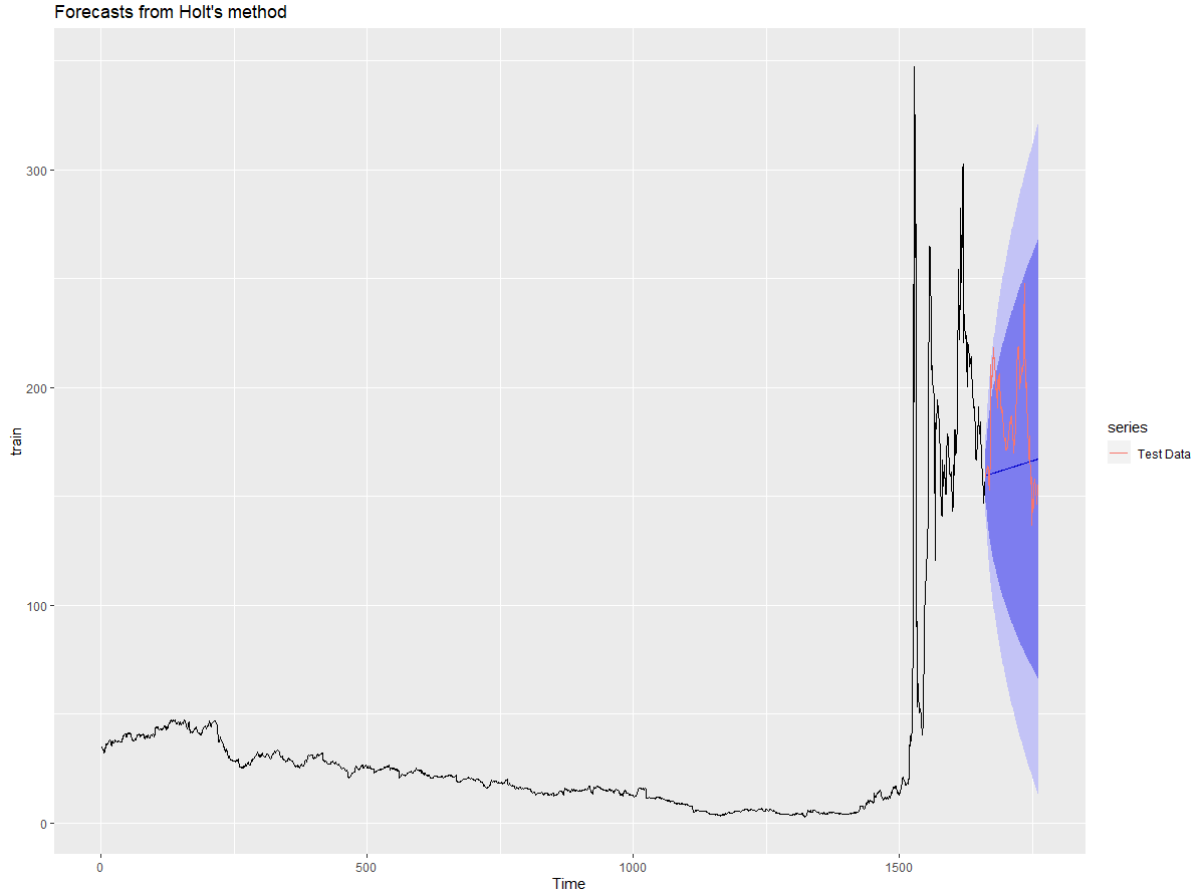

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = train, h = 100, alpha = 0.15)

  Smoothing parameters:
    alpha = 0.15

  Initial states:
    l = 34.4501

  sigma:  14.0044

     AIC     AICc      BIC
21100.29 21100.30 21111.12

Error measures:
                    ME     RMSE    MAE        MPE     MAPE    MASE      ACF1
Training set 0.5136717 13.99597 3.2671 -0.8853452 6.533057 1.92081 0.7301305

Forecasts:
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
1663       162.5085 144.5611 180.4558 135.0603 189.9566
1664       162.5085 144.3603 180.6566 134.7533 190.2637
1665       162.5085 144.1617 180.8552 134.4495 190.5674
1666       162.5085 143.9653 181.0517 134.1491 190.8678
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

4. Holt's Trend Method

```
gme_h <- holt(train,h=100)
autoplot(gme_h)+autolayer(ts(test,start=length(train)),series="Test Data")
summary(gme_h)
```

This is a variation of the simple exponential smoothing method that takes the trend component into account when creating forecasts. Two smoothing equations are used in this method, one for the level and one for the trend component.



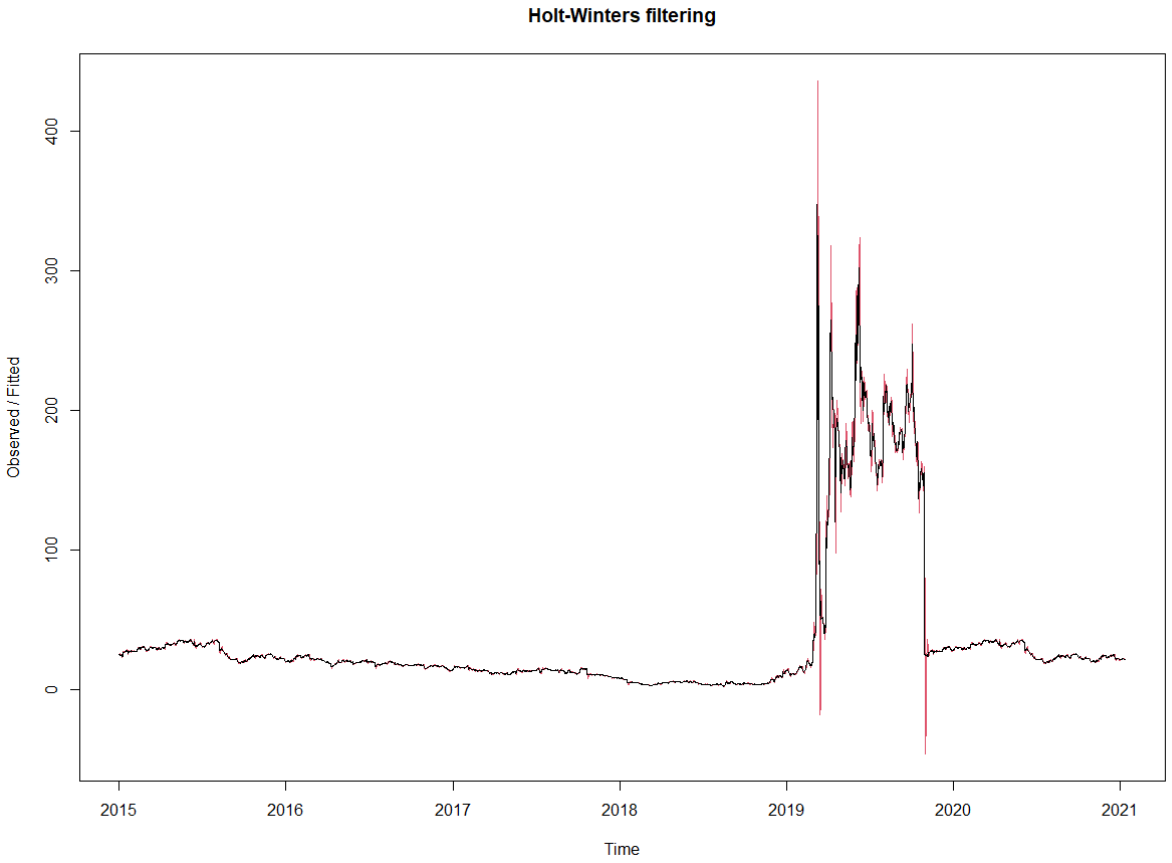
Gourav Singh Bajeli
AM.EN.U4AIE19031

```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = train, h = 100)

  Smoothing parameters:
    alpha = 0.8206
    beta  = 1e-04

  Initial states:
    l = 34.3459
    b = 0.0755

  sigma:  9.5197

     AIC     AICc      BIC
19821.19 19821.23 19848.27

Error measures:
                      ME     RMSE      MAE        MPE    MAPE     MASE         ACF1
Training set 0.008613976 9.508219 1.727507 -0.7928107 3.29128 1.015644 -0.02598143

Forecasts:
     Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
1663       159.5669 147.36699 171.7669 140.90872 178.2252
1664       159.6439 143.86136 175.4264 135.50659 183.7812
1665       159.7208 141.02960 178.4120 131.13507 188.3066
1666       159.7977 138.59258 181.0029 127.36724 192.2283
```

5. Holt-Winters Method

```
gme_hw <- HoltWinters(gme,beta=TRUE,gamma=FALSE)
plot(gme_hw)
summary(gme_hw)
```

Time series can be made smooth and utilise the data to forecast areas of interest using the Holt-Winters forecasting technique. Exponential smoothing reduces the weight of earlier data by assigning exponentially decreasing weights and values to it.




```
            Length Class  Mode
fitted      6600   mts    numeric
x           2202   ts     numeric
alpha          1   -none- numeric
beta           1   -none- logical
gamma          1   -none- logical
coefficients   2   -none- numeric
seasonal       1   -none- character
SSE            1   -none- numeric
call           4   -none- call
> 
```

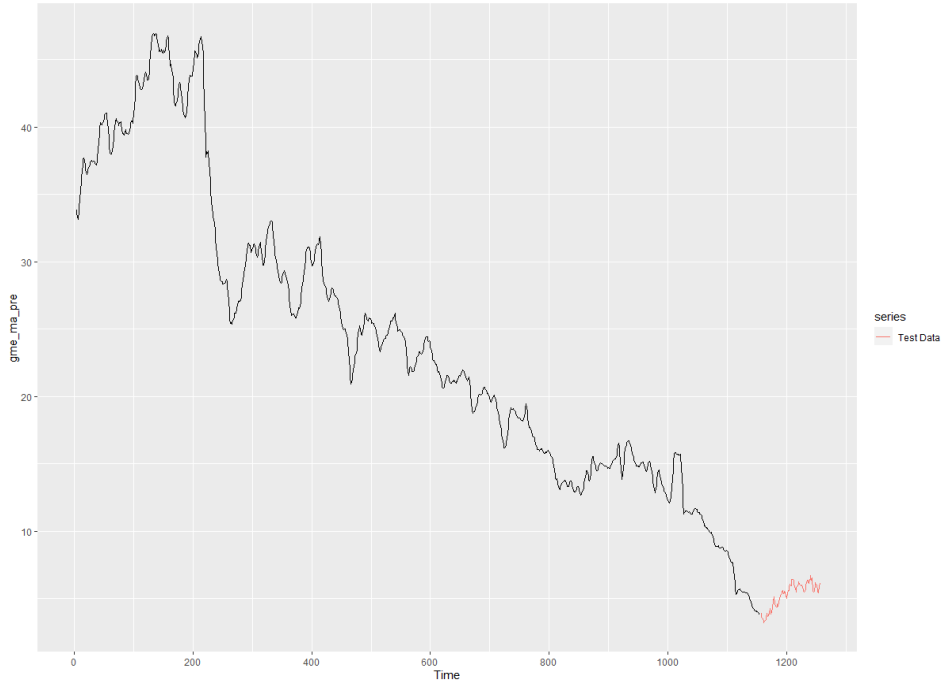
Gourav Singh Bajeli
AM.EN.U4AIE19031

Analyzing Pre & Post Covid Data

```
g_pre <- getSymbols(c("GME"), src = "yahoo", from =
"2015-01-01", to = "2019-12-31",auto.assign=FALSE)
g_post <- getSymbols(c("GME"), src = "yahoo", from =
"2020-01-01", to = "2021-12-31",auto.assign=FALSE)
gme_pre <- ts(g_pre$GME.Adjusted, start=c(2015, 1),
end=c(2019, 12),frequency=365)
gme_post <- ts(g$GME.Adjusted, start=c(2020, 1),
end=c(2021, 12),frequency=365)

train_pre <- head(Cl(g_pre), length(Cl(g_pre))-100)
test_pre <- tail(Cl(g_pre), 100)

train_post <- head(Cl(g_post), length(Cl(g_post))-100)
test_post <- tail(Cl(g_post), 100)
```

Moving Averages:

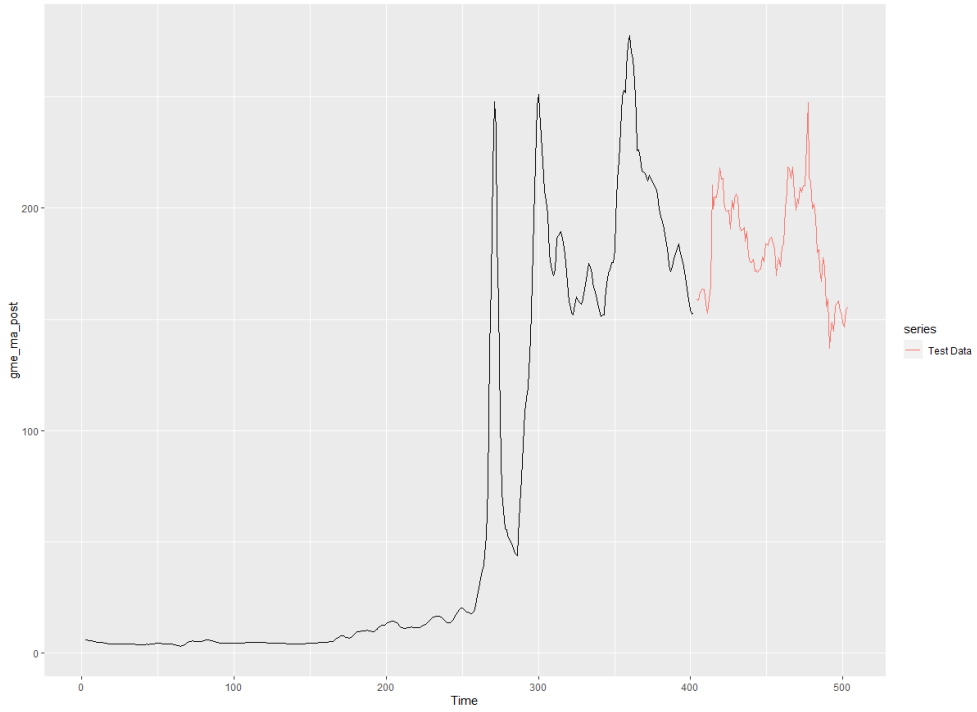
Pre-covid:




```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
3.79   15.06   22.67   24.01   30.73   46.97       4
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

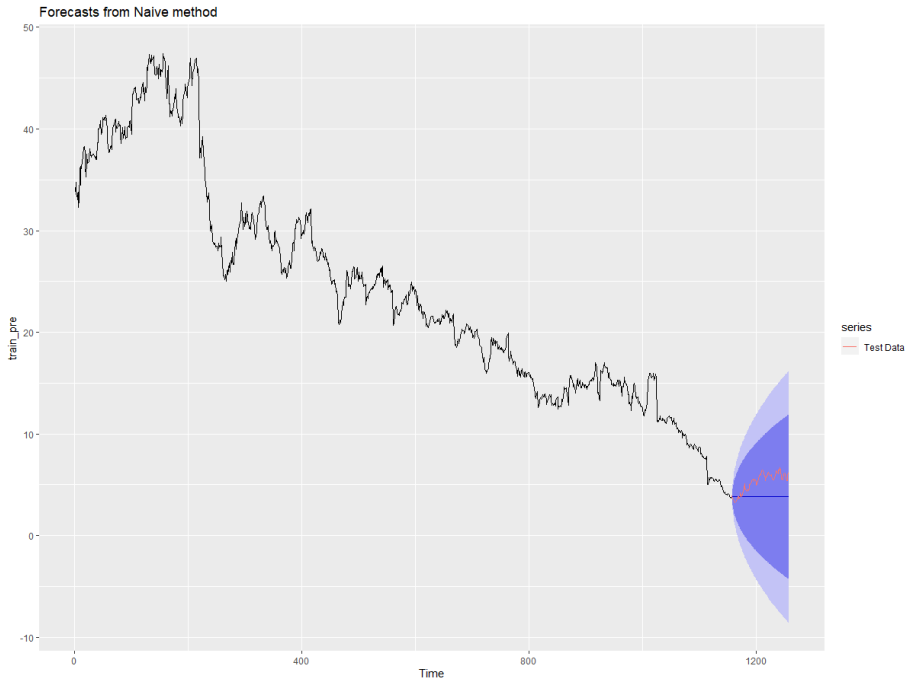
Post-covid:




```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
3.052   4.429  11.433  64.137 156.841 277.822       4
```

Naive method:

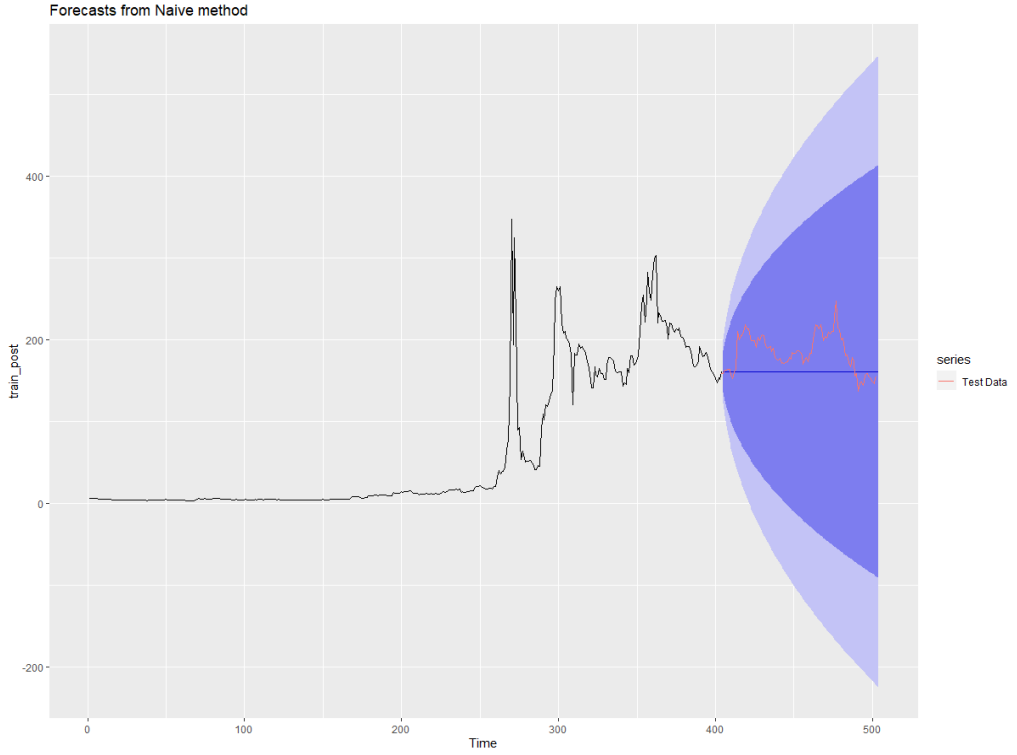
Pre-covid:



Gourav Singh Bajeli
AM.EN.U4AIE19031

```
Forecast method: Naive method

Model Information:
Call: naive(y = train_pre, h = 100)

Residual sd: 0.6308

Error measures:
                      ME      RMSE       MAE         MPE     MAPE MASE         ACF1
Training set -0.02594291 0.6307885 0.3989705 -0.2354652 1.823351    1 -0.05038629

Forecasts:
     Point Forecast       Lo 80     Hi 80       Lo 95     Hi 95
1158           3.81  3.00161204  4.618388  2.57367730  5.046323
1159           3.81  2.66676678  4.953233  2.06157567  5.558424
1160           3.81  2.40983098  5.210169  1.66862627  5.951374
1161           3.81  2.19322407  5.426776  1.33735460  6.282645
1162           3.81  2.00238956  5.617610  1.04549840  6.574502
1163           3.81  1.82986198  5.790138  0.78164022  6.838360
1164           3.81  1.67120649  5.948794  0.53899759  7.081002
1165           3.81  1.52353356  6.096466  0.31315134  7.306849
1166           3.81  1.38483611  6.235164  0.10103190  7.518968
```

Post-covid:

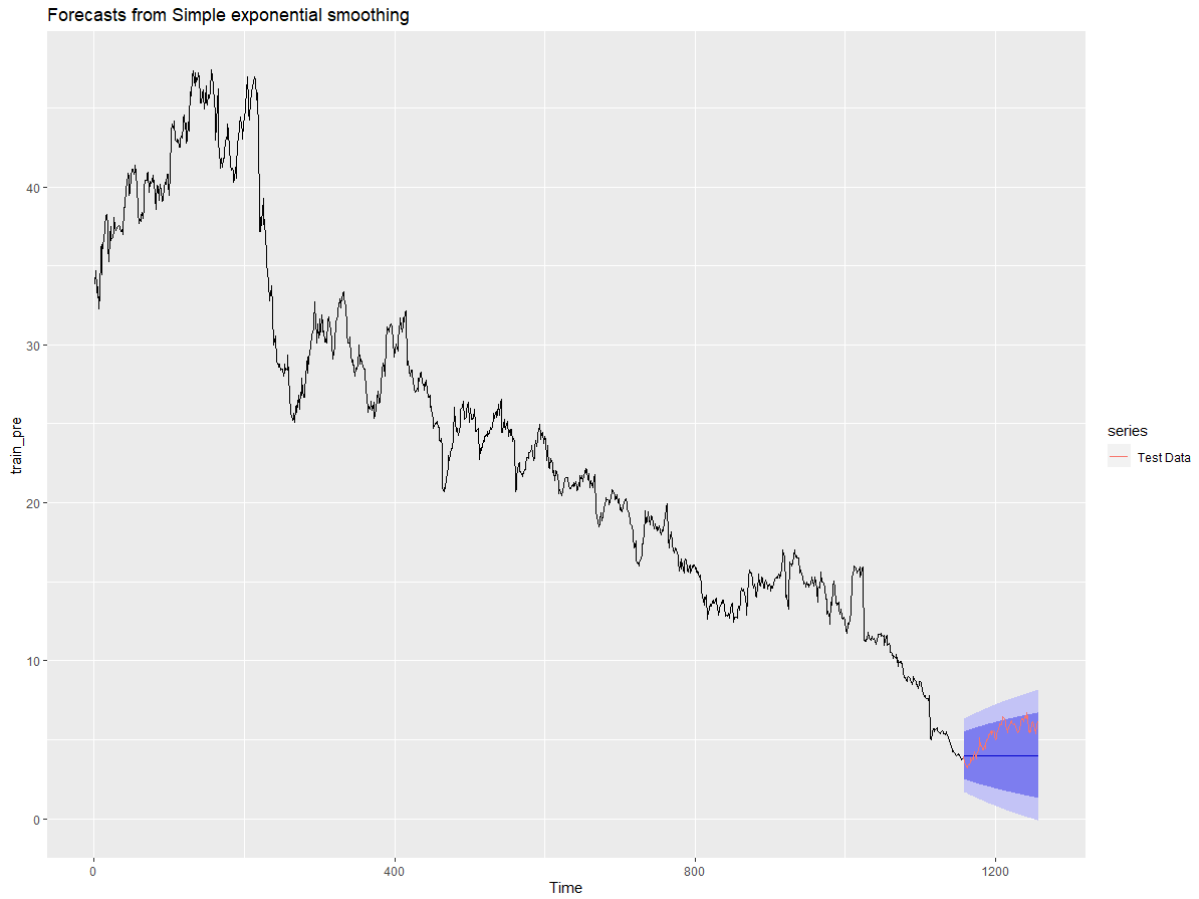



```
Forecast method: Naive method

Model Information:
Call: naive(y = train_post, h = 100)

Residual sd: 19.6758

Error measures:
                    ME     RMSE      MAE          MPE     MAPE MASE       ACF1
Training set 0.3841687 19.67577 5.821886 -0.006476911 6.795788    1 -0.2220873

Forecasts:
    Point Forecast      Lo 80    Hi 80       Lo 95    Hi 95
405         161.13 135.914488 186.3455  122.566199 199.6938
406         161.13 125.469879 196.7901  106.592548 215.6675
407         161.13 117.455448 204.8046   94.335534 227.9245
408         161.13 110.698971 211.5610   84.002394 238.2576
409         161.13 104.746395 217.5136   74.898714 247.3613
410         161.13  99.364854 222.8952   66.668359 255.5917
411         161.13  94.416017 227.8440   59.099766 263.1602
412         161.13  89.809752 232.4503   52.055091 270.2049
413         161.13  85.483454 236.7766   45.438588 276.8214
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

Simple Exponential Smoothing

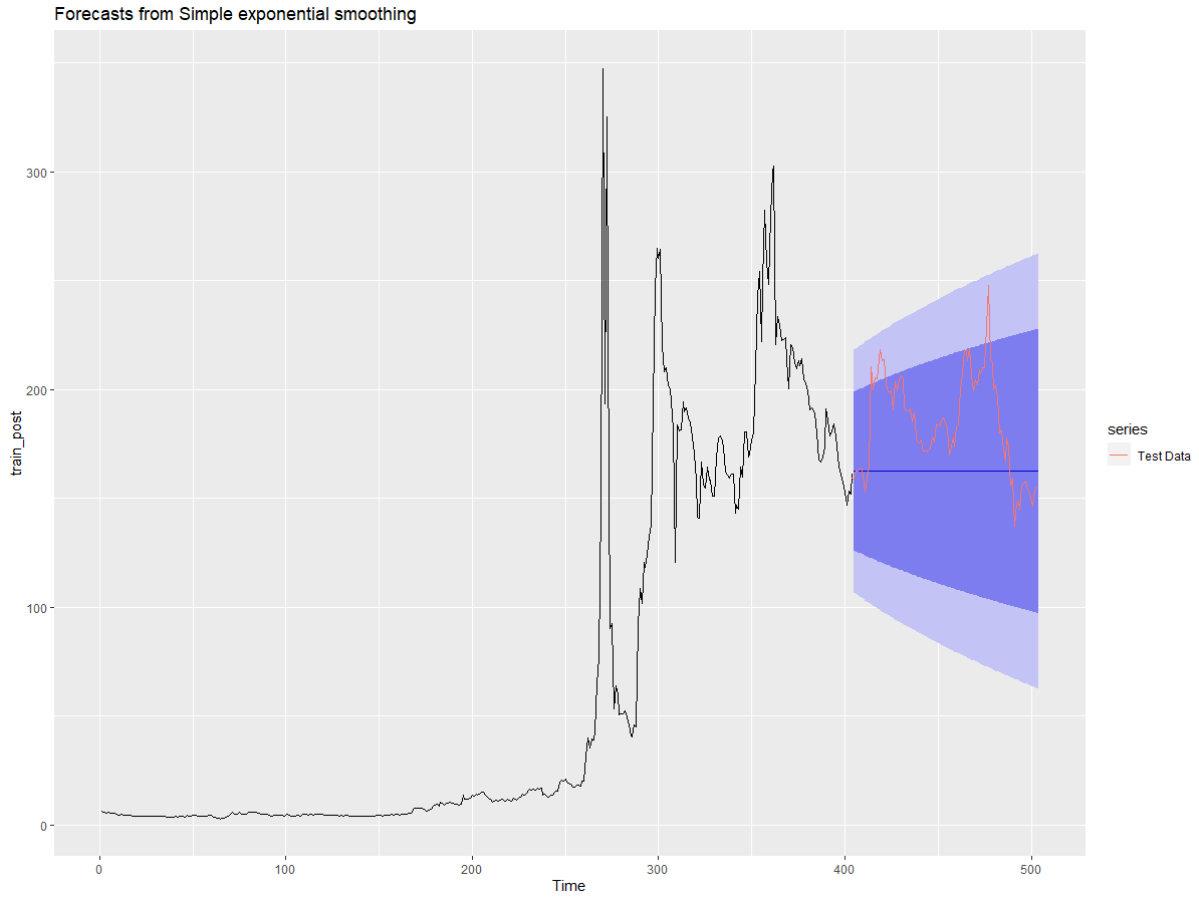
Pre-covid:




```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = train_pre, h = 100, alpha = 0.15)

  Smoothing parameters:
    alpha = 0.15

  Initial states:
    l = 34.4402

  sigma:  1.172

     AIC     AICc      BIC
8530.189 8530.199 8540.296

Error measures:
                     ME     RMSE       MAE       MPE     MAPE     MASE      ACF1
Training set -0.1754215 1.170953 0.8395801 -1.528263 3.997076 2.104366 0.8389772

Forecasts:
     Point Forecast    Lo 80    Hi 80        Lo 95    Hi 95
1158       3.995789 2.493854 5.497724  1.6987779620 6.292800
1159       3.995789 2.477051 5.514527  1.6730803309 6.318498
1160       3.995789 2.460433 5.531146  1.6476639151 6.343914
1161       3.995789 2.443992 5.547587  1.6225196796 6.369059
1162       3.995789 2.427723 5.563855  1.5976390630 6.393939
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

Post-covid:

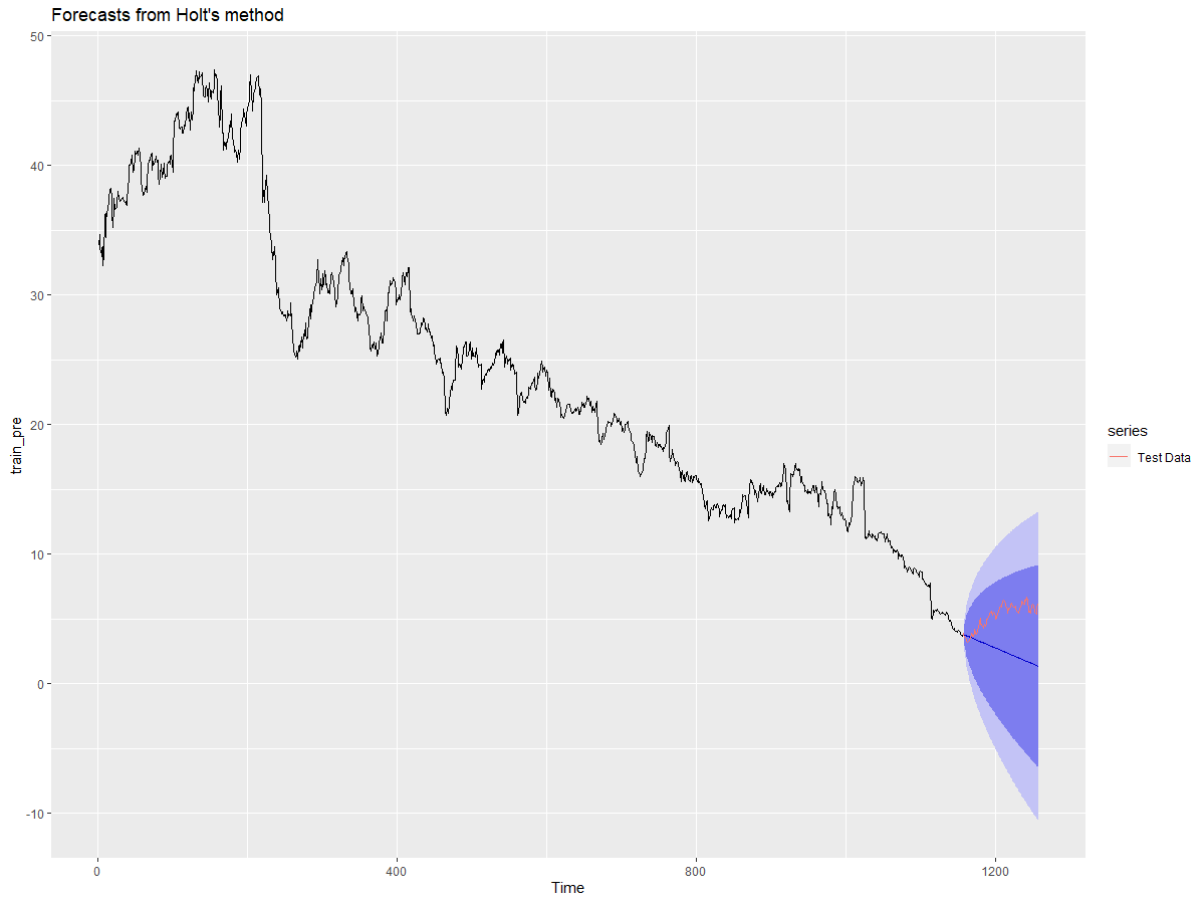



```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = train_post, h = 100, alpha = 0.15)

  Smoothing parameters:
    alpha = 0.15

  Initial states:
    l = 5.4322

  sigma:  28.388

     AIC     AICc      BIC
5130.107 5130.137 5138.110

Error measures:
                    ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
Training set 2.592017 28.31762 10.95393 0.3724507 13.82399 1.881509 0.7276359

Forecasts:
    Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
405       162.5085 126.12781 198.8891 106.86905 218.1479
406       162.5085 125.72080 199.2961 106.24659 218.7703
407       162.5085 125.31825 199.6987 105.63094 219.3860
408       162.5085 124.92001 200.0969 105.02188 219.9950
409       162.5085 124.52594 200.4910 104.41921 220.5977
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

Holt's Trend Method

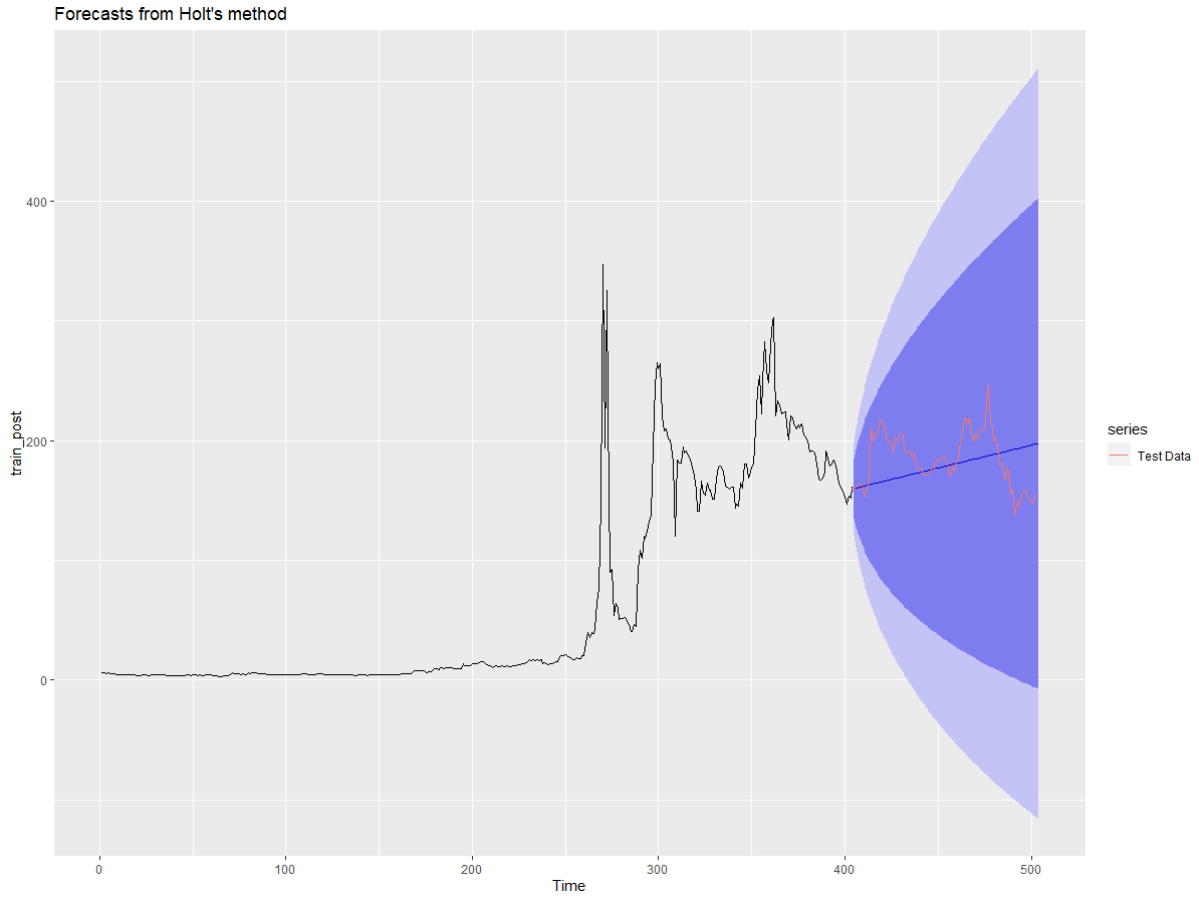
Pre-covid:




```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = train_pre, h = 100)

  Smoothing parameters:
    alpha = 0.9578
    beta  = 1e-04

  Initial states:
    l = 32.8189
    b = -0.0244

  sigma:  0.6311

     AIC     AICc      BIC
7102.055 7102.107 7127.323

Error measures:
                     ME      RMSE       MAE         MPE    MAPE     MASE          ACF1
Training set -0.00124664 0.6300563 0.4003736 -0.1041332 1.81849 1.003517 -0.003472005

Forecasts:
     Point Forecast       Lo 80    Hi 80        Lo 95     Hi 95
1158       3.784208  2.97535942 4.593057   2.54718062  5.021236
1159       3.759671  2.63961759 4.879725   2.04669691  5.472646
1160       3.735134  2.37317631 5.097092   1.65219926  5.818070
1161       3.710597  2.14360520 5.277590   1.31408965  6.107105
1162       3.686060  1.93788286 5.434238   1.01245360  6.359667
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

Post-covid:

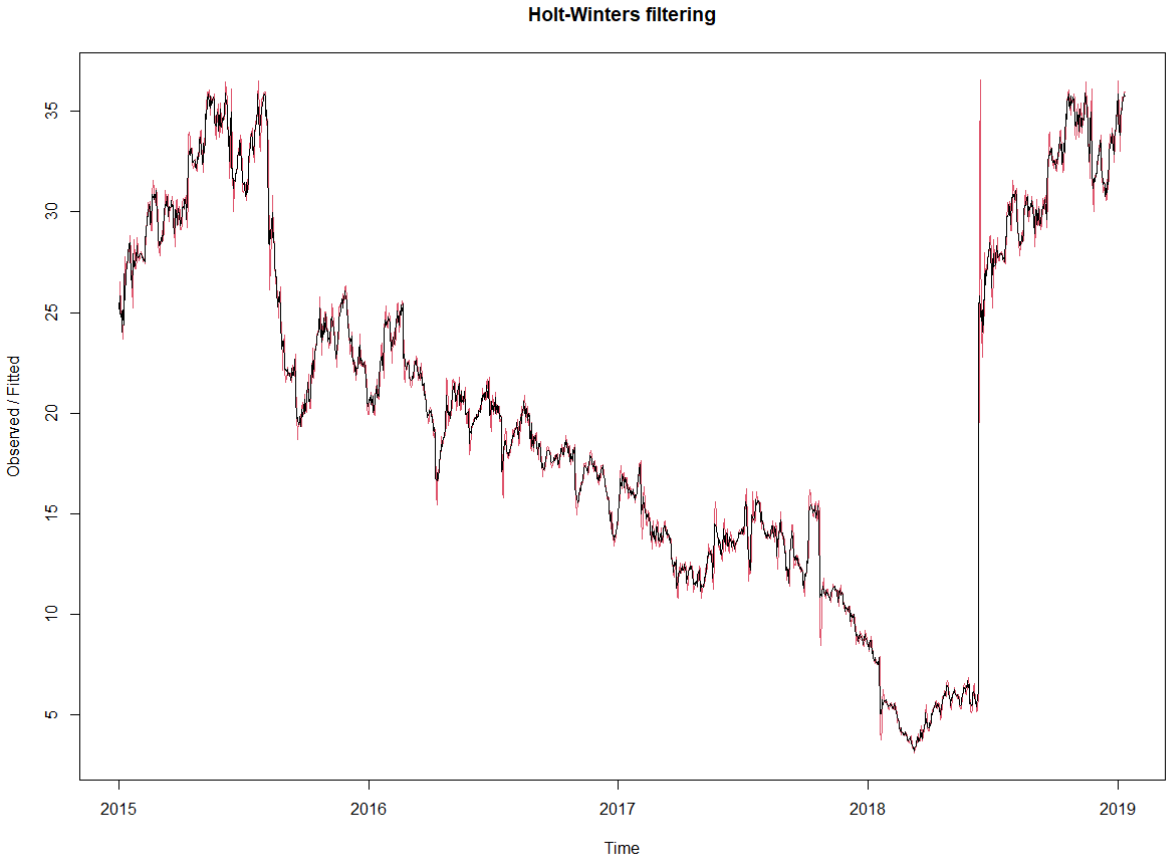



```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = train_post, h = 100)

  Smoothing parameters:
    alpha = 0.8197
    beta  = 1e-04

  Initial states:
    l = 5.8941
    b = 0.3798

  sigma:  19.3461

     AIC     AICc      BIC
4824.243 4824.394 4844.250

Error measures:
                      ME     RMSE      MAE       MPE    MAPE     MASE         ACF1
Training set 0.003229001 19.25005 6.02108 -5.374787 9.91556 1.034215 -0.02599433

Forecasts:
    Point Forecast        Lo 80    Hi 80        Lo 95    Hi 95
405       159.9287 135.13575317 184.7217  122.0111426 197.8463
406       160.3087 128.24883940 192.3686  111.2773680 209.3401
407       160.6887 122.72713400 198.6502  102.6314995 218.7459
408       161.0687 118.00563482 204.1317   95.2094408 226.9279
409       161.4486 113.82647967 209.0708   88.6168256 234.2805
```

Gourav Singh Bajeli
AM.EN.U4AIE19031

Holt-Winters Method:

Pre-covid:



	Length	Class	Mode
fitted	4410	mts	numeric
x	1472	ts	numeric
alpha	1	-none-	numeric
beta	1	-none-	logical
gamma	1	-none-	logical
coefficients	2	-none-	numeric
seasonal	1	-none-	character
SSE	1	-none-	numeric
call	4	-none-	call

Post-covid:

	Length	Class	Mode
fitted	1125	mts	numeric
x	377	ts	numeric
alpha	1	-none-	numeric
beta	1	-none-	logical
gamma	1	-none-	logical
coefficients	2	-none-	numeric
seasonal	1	-none-	character
SSE	1	-none-	numeric
call	4	-none-	call

Gourav Singh Bajeli
AM.EN.U4AIE19031