STAT 443 project

Pival Dhingra

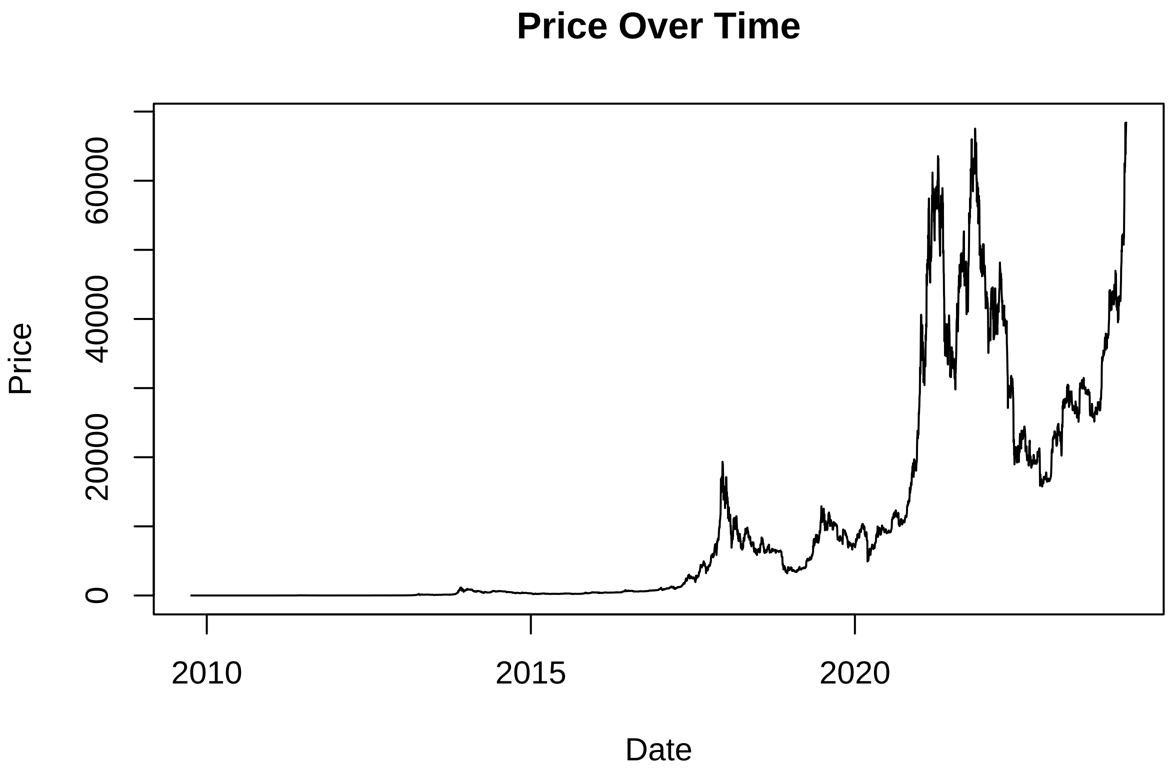
2024-04-06

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

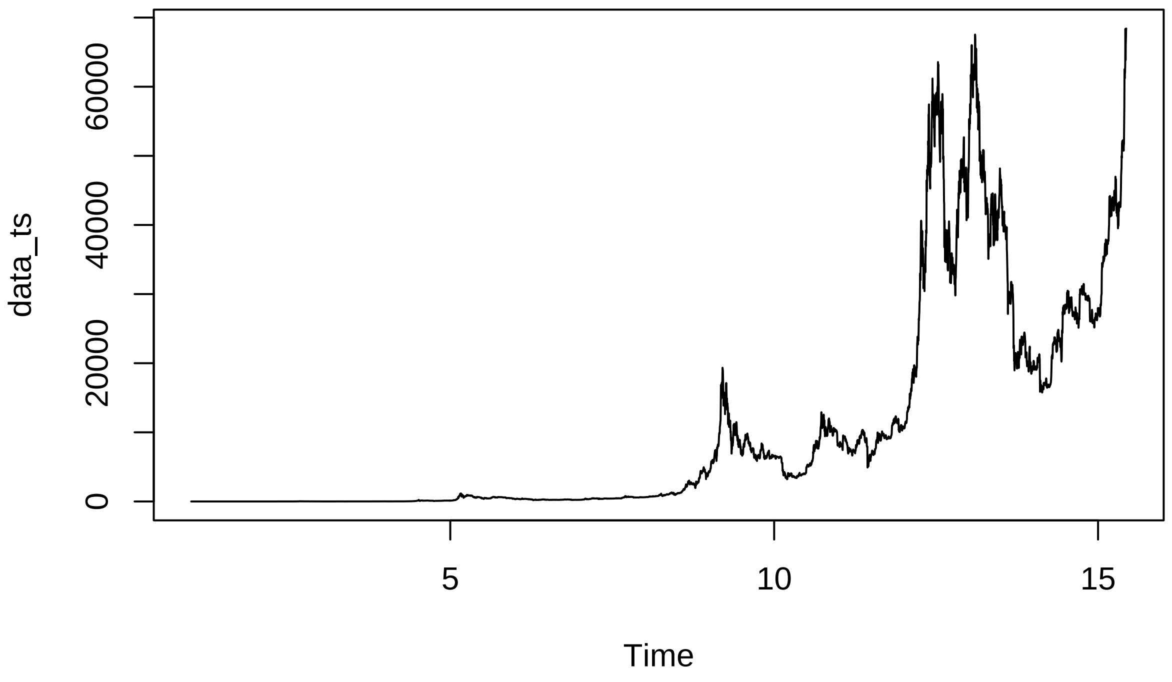
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
data <- read.csv("/Users/pivaldhingra/Desktop/University courses/STAT 443 project /Data_Group24.csv")
data$date <- as.Date(data$date)
plot(data$date, data$price, type = "l",
     xlab = "Date", ylab = "Price",
     main = "Price Over Time")
```

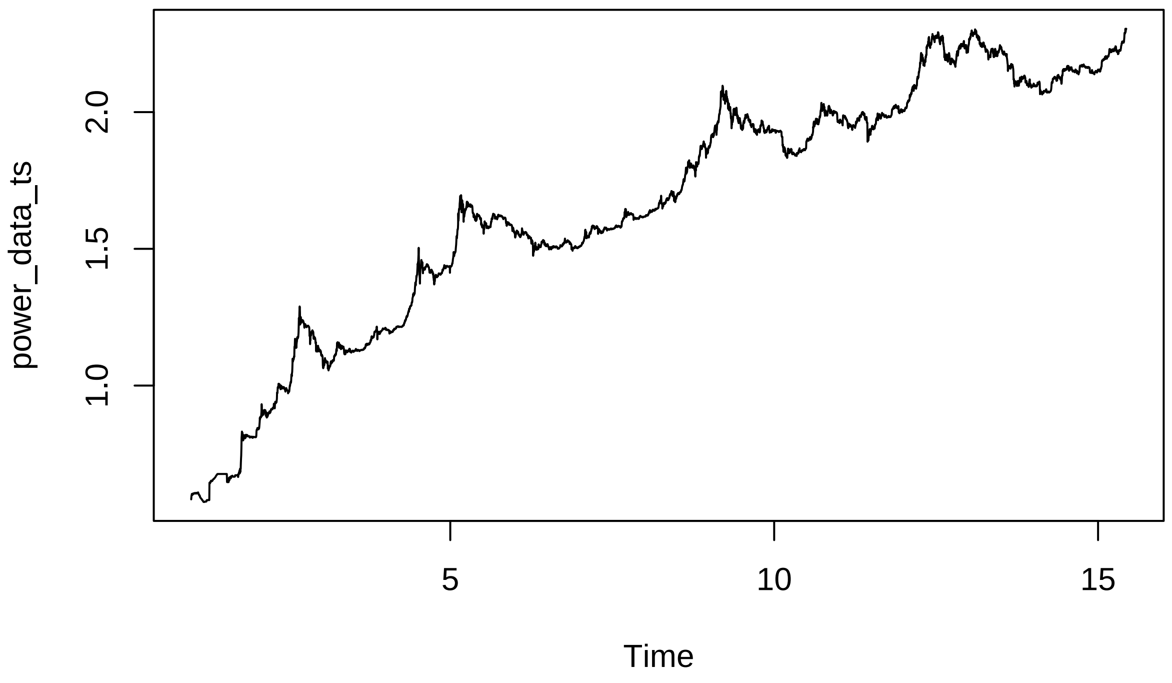


From the plot, we observe a change point in 2017 because of 2017 market manipulation. Also, after 2019, there seems to be an increase trend. Moreover, in 2021, there was more disposable income from government credits which led to pandemic fuel trading boom as seen in the plot there is a lot increase in the price.

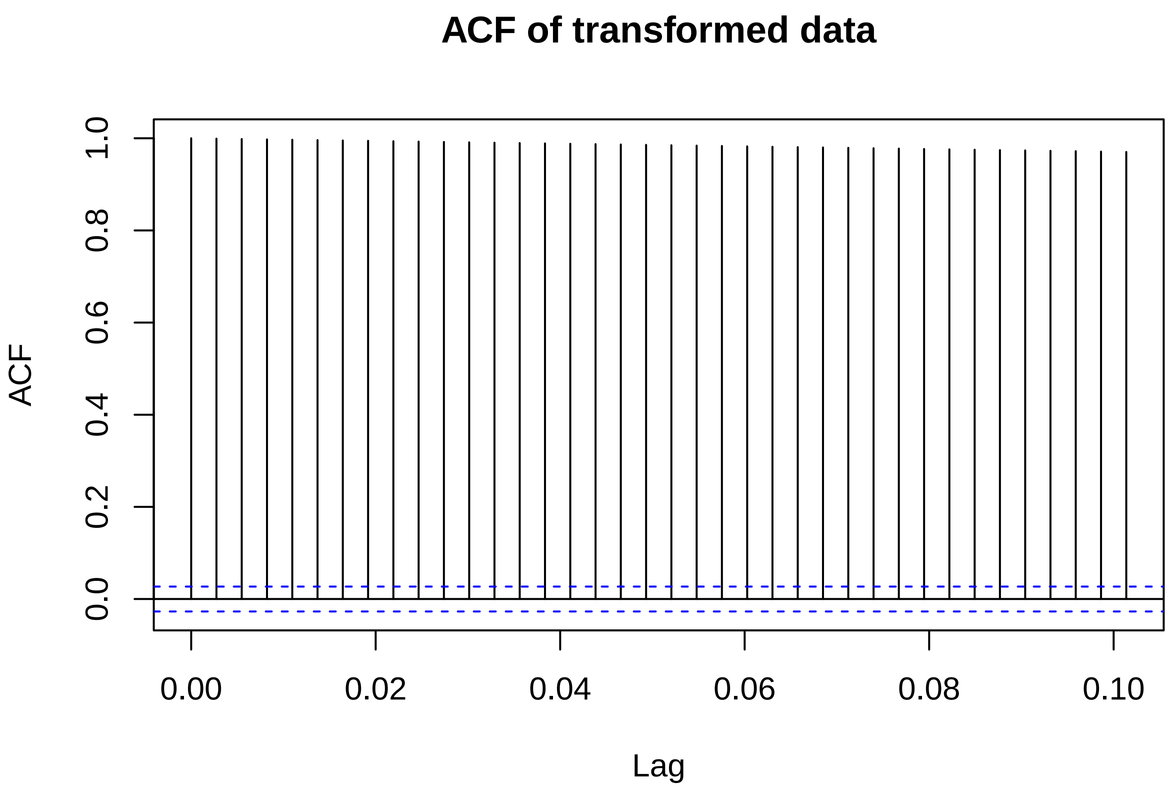
```r
data_ts = ts(data$price, frequency = 365)
power_data_ts = data_ts^0.075
plot(data_ts)
```



```r
plot(power_data_ts)
```



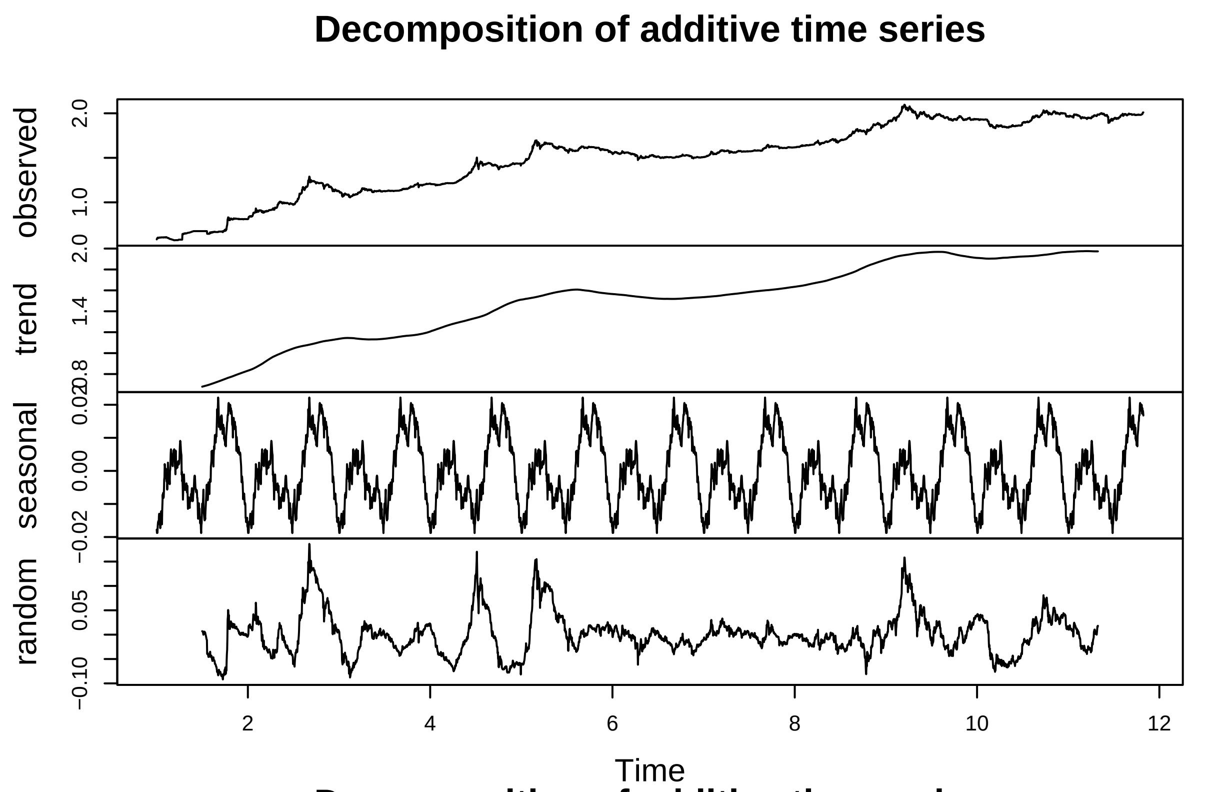
```r
acf(power_data_ts, main="ACF of transformed data")
```



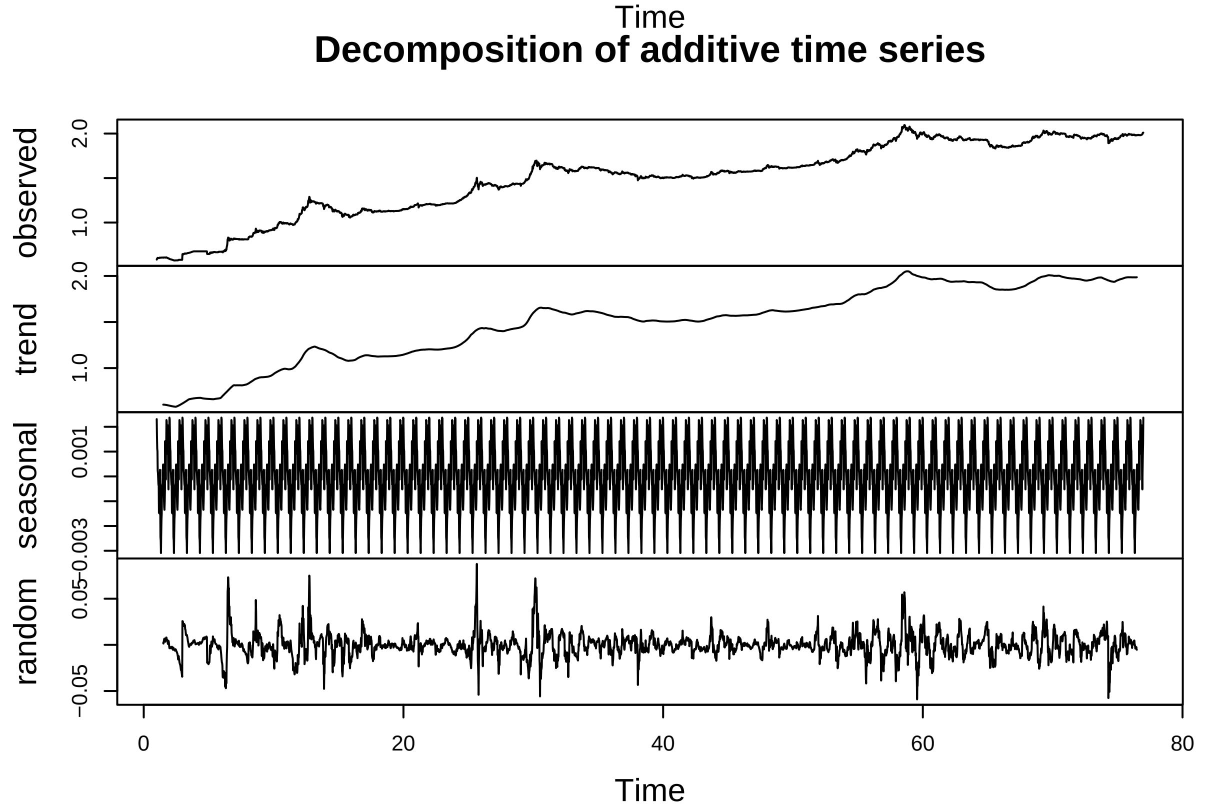
From the plot we observe that there is an increasing trend. Also, from the acf plot there is an linear decay showcasing that it is not stationary and there is trend.

```
## training set 3952 days
## test set 1318 days
train <- read.csv("train.csv")
test <- read.csv("test.csv")
train_ts <- ts(train$price, frequency=365)
power_train_ts <- train_ts^0.075
test_ts <- ts(test$price, frequency=365)
power_test_ts <- test_ts^0.075
```

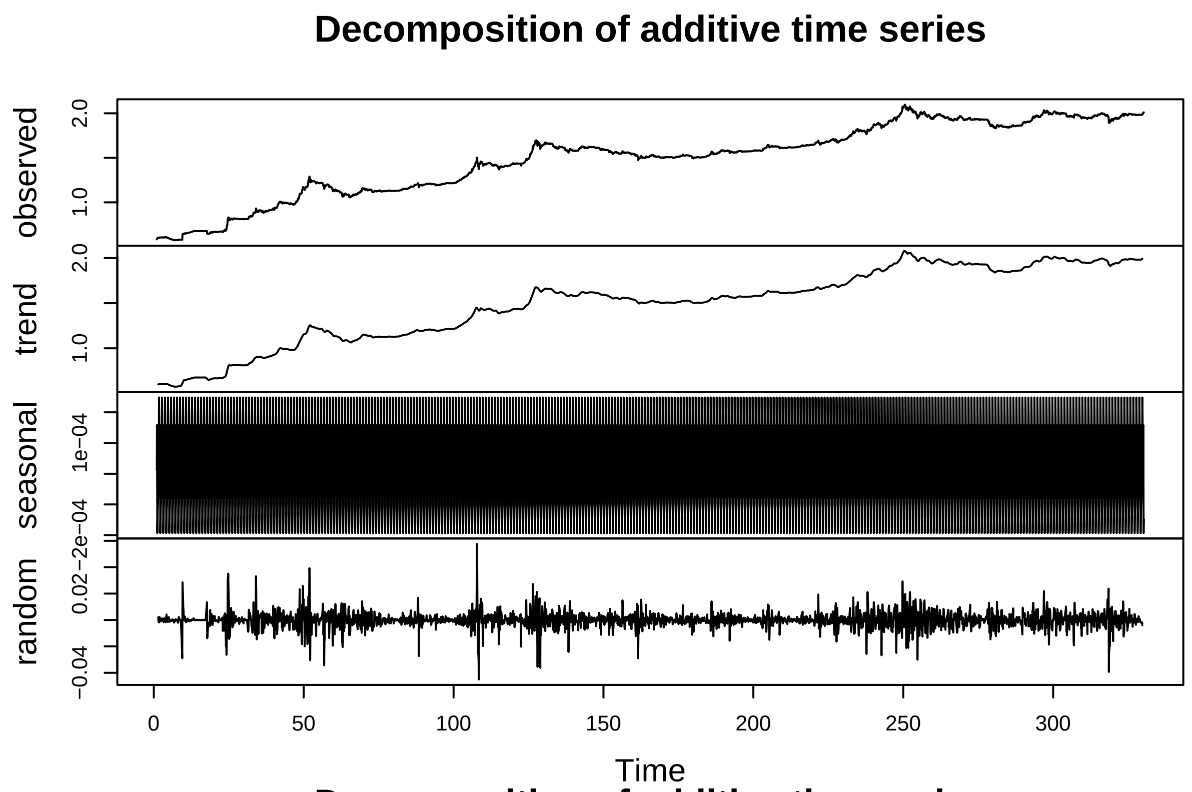
```
## Applying smoothing methods
## Applying decompose method for different seasonal periods
for (i in c(365, 52, 12, 4)) {
  power_train_ts_season <- ts(power_train_ts, frequency = i)
  add_decompose_ts <- decompose(power_train_ts_season, type="additive")
  par(mfrow=c(4,4))
  plot(add_decompose_ts)
}
```

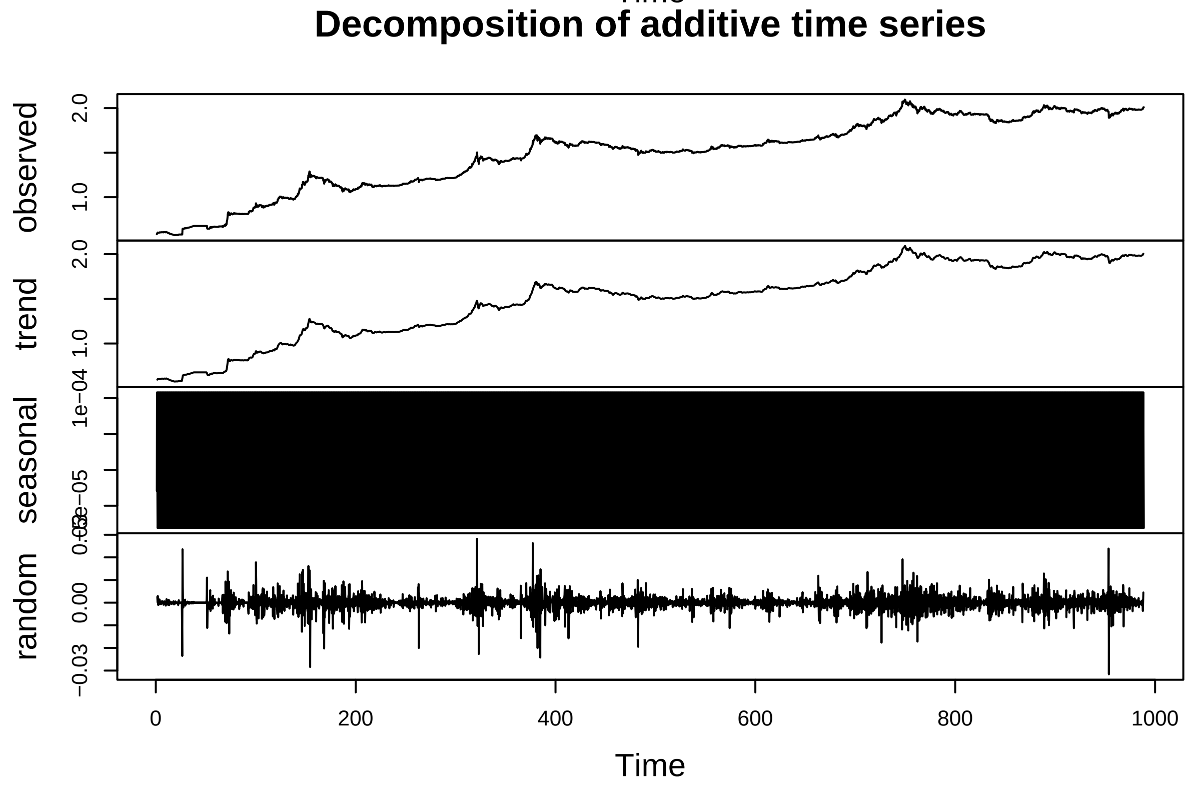


Decomposition of additive time series
observed
trend
seasonal
random
Time
2.0
1.0
1.4
0.8
0.02
0.00
−0.02
0.05
−0.10
2
4
6
8
10
12



Decomposition of additive time series
observed
trend
seasonal
random
Time
2.0
1.0
0.001
−0.003
0.05
−0.05
0
20
40
60
80





From these 4 plots, we observe that for seasonal period 365 and 4, there seems to a trend in the residual, so it is not stationary and we don't choose this model. Only models with seasonal period 52 and 12 seems to be a good fit. We will explore more using other smoothing methods to choose the best model.

```r
## Applying Exponential smoothing and Holts Winter method for different seasonal periods
## Initializing variables
best_period_es <- NULL
best_period_hw <- NULL
best_period_hw_additive <- NULL
best_period_hw_multiplicative <- NULL
best_model_es <- NULL
best_model_hw <- NULL
best_model_hw_additive <- NULL
best_model_hw_multiplicative <- NULL
min_APSE_es <- Inf
min_APSE_hw <- Inf
min_APSE_hw_additive <- Inf
min_APSE_hw_multiplicative <- Inf

for (i in c(7, 14, 30, 60, 90)) {
  power_train_ts_season <- ts(power_train_ts, frequency = i)
  # Fitting Holt-Winters models with different seasonal periods
  es <- HoltWinters(power_train_ts_season, gamma=FALSE, beta=FALSE)
  hw <- HoltWinters(power_train_ts_season, gamma=FALSE)
  hw_additive <- HoltWinters(power_train_ts_season, seasonal=c("additive"))
  hw_multiplicative <- HoltWinters(power_train_ts_season, seasonal=c("multiplicative"))
  # Predictions
  es_pred <- predict(es, n.ahead = length(power_test_ts))
  hw_pred <- predict(hw, n.ahead = length(power_test_ts))
  hw_additive_pred <- predict(hw_additive, n.ahead = length(power_test_ts))
  hw_multiplicative_pred <- predict(hw_multiplicative, n.ahead = length(power_test_ts))
  ## For calculating APSE, removing the power transformation
  es_pred_data <- es_pred^(1/0.075)
  hw_pred_data <- hw_pred^(1/0.075)
  hw_additive_pred_data <- hw_additive_pred^(1/0.075)
  hw_multiplicative_pred_data <- hw_multiplicative_pred^(1/0.075)
  # APSE for each model
  APSE_es <- mean((es_pred_data - as.vector(power_test_ts))^2)
  APSE_hw <- mean((hw_pred_data - as.vector(power_test_ts))^2)
  APSE_hw_additive <- mean((hw_additive_pred_data - as.vector(power_test_ts))^2)
  APSE_hw_multiplicative <- mean((hw_multiplicative_pred_data - as.vector(power_test_ts))^2)

  # Check if this model has the lowest APSE so far for each method
  if (APSE_es < min_APSE_es) {
    best_period_es <- i
    best_model_es <- es
    min_APSE_es <- APSE_es
  }

  if (APSE_hw < min_APSE_hw) {
    best_period_hw <- i
    best_model_hw <- hw
    min_APSE_hw <- APSE_hw
  }

  if (APSE_hw_additive < min_APSE_hw_additive) {
    best_period_hw_additive <- i
    best_model_hw_additive <- hw_additive
    min_APSE_hw_additive <- APSE_hw_additive
  }

  if (APSE_hw_multiplicative < min_APSE_hw_multiplicative) {
    best_period_hw_multiplicative <- i
    best_model_hw_multiplicative <- hw_multiplicative
    min_APSE_hw_multiplicative <- APSE_hw_multiplicative
  }
}
cat("Best seasonal period for ES:", best_period_es, "\n")
```

```
## Best seasonal period for ES: 7
```

```r
cat("APSE for ES :", min_APSE_es, "\n")
```

```
## APSE for ES : 122131856
```

```r
cat("Best seasonal period for HW:", best_period_hw, "\n")
```

```
## Best seasonal period for HW: 7
```

```r
cat("APSE for HW :", min_APSE_hw, "\n")
```

```
## APSE for HW : 189537201772
```

```r
cat("Best seasonal period for HW Additive:", best_period_hw_additive, "\n")
```

```
## Best seasonal period for HW Additive: 60
```

```r
cat("APSE for HW Additive:", min_APSE_hw_additive, "\n")
```

```
## APSE for HW Additive: 765090221
```

```r
cat("Best seasonal period for HW Multiplicative:", best_period_hw_multiplicative, "\n")
```

```
## Best seasonal period for HW Multiplicative: 30
```

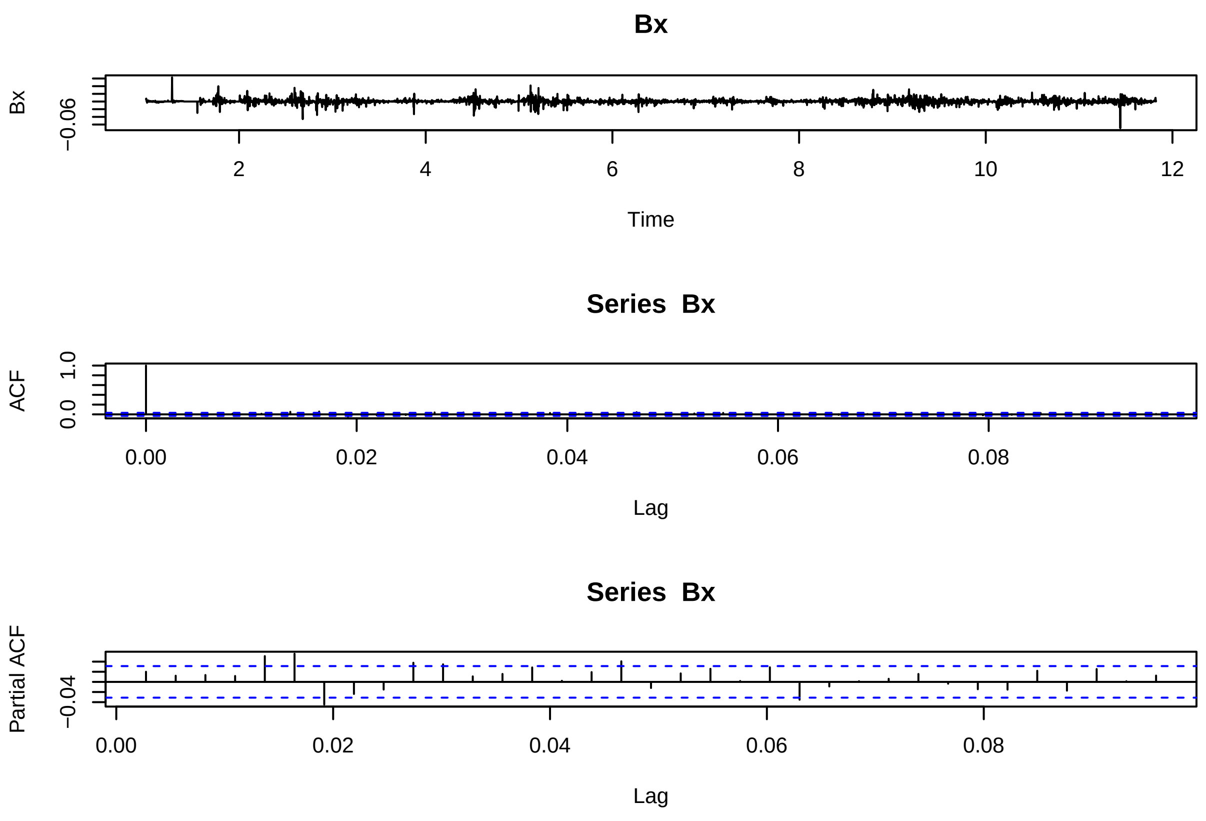
```r
cat("APSE for HW Multiplicative:", min_APSE_hw_multiplicative, "\n")
```

```
## APSE for HW Multiplicative: 929775867
```

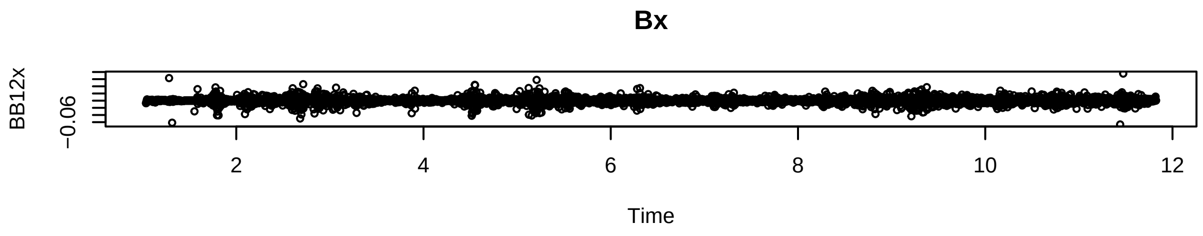
```r
## Now selecting the min APSE among the 4 models
min(min_APSE_es, min_APSE_hw, min_APSE_hw_additive, min_APSE_hw_multiplicative)
```

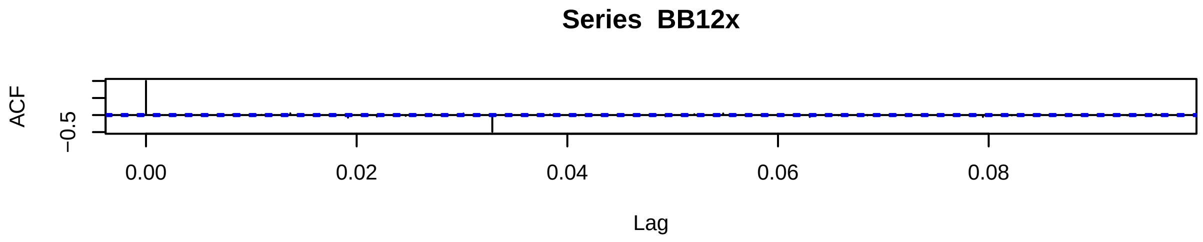
```
## [1] 122131856
```

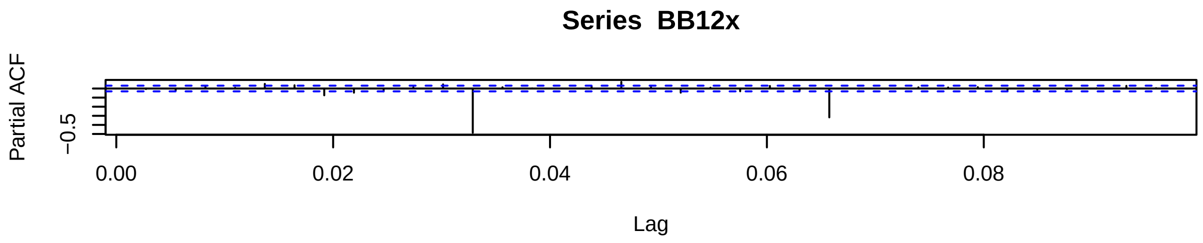
```r
## Now applying differencing method
## regular differencing
Bx <- diff(power_train_ts)
par(mfrow=c(3,1))
plot(Bx, type="l", main="Bx")
acf(Bx)
pacf(Bx)
```



```r
## one time differencing in lag 12 of the already differenced data in lag 1
BB12x <- diff(Bx, lag=12)
par(mfrow=c(3,1))
plot(BB12x, type="p", main="Bx")
acf(BB12x)
pacf(BB12x)
```





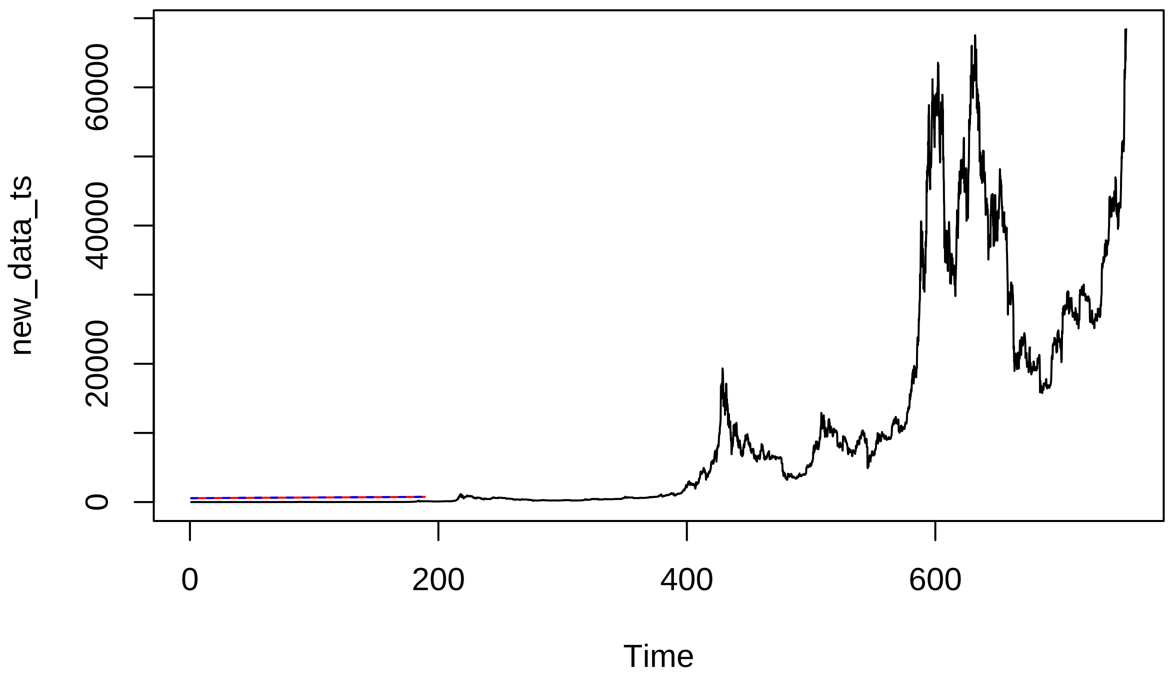


From these calculations, we observe that Holts Winter multiplicative model with seasonal frequency 7 i.e weakly seasonality fits the best.

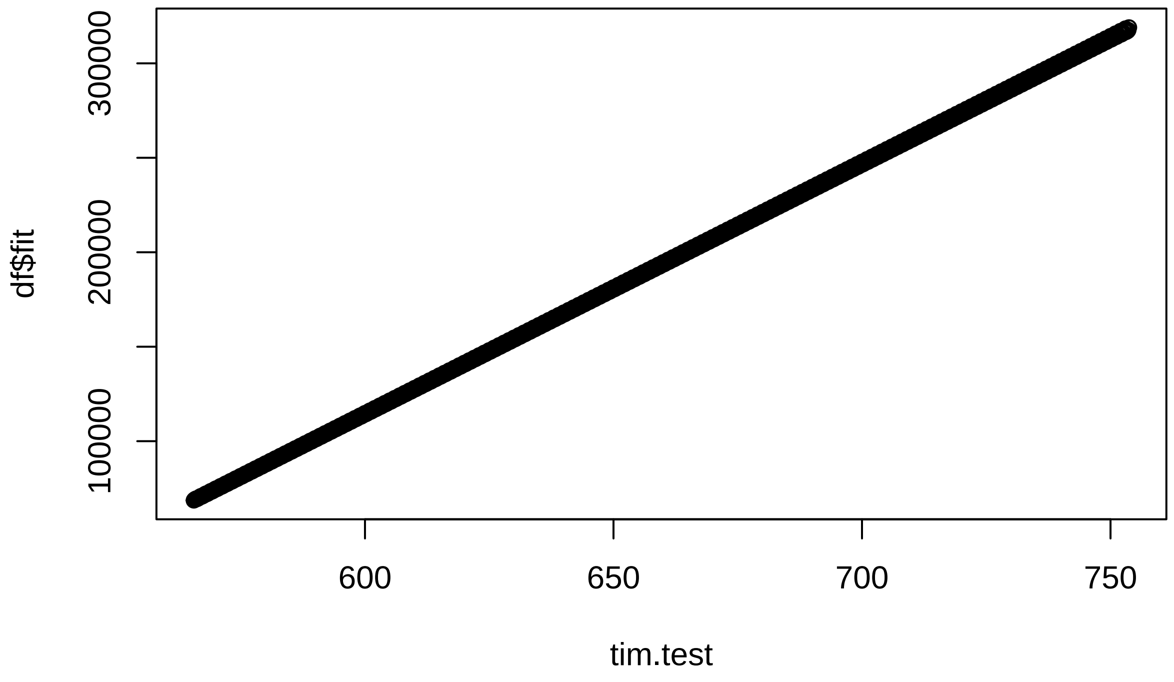
```
new_data_ts = ts(data$price, frequency = 7)
new_power_data_ts = new_data_ts^0.075
new_train_ts <- ts(train$price, frequency=7)
new_test_ts <- ts(test$price, frequency=7)

best_model <- HoltWinters(new_data_ts, seasonal ="multiplicative")
best_model_pred <- predict(best_model, n.ahead=length(new_test_ts),
                           prediction.interval = TRUE, level = 0.95)
best_model_pwr <- HoltWinters(new_power_data_ts, seasonal ="multiplicative")
```

```
## Warning in HoltWinters(new_power_data_ts, seasonal = "multiplicative"):
## optimization difficulties: ERROR: ABNORMAL_TERMINATION_IN_LNSRCH
```

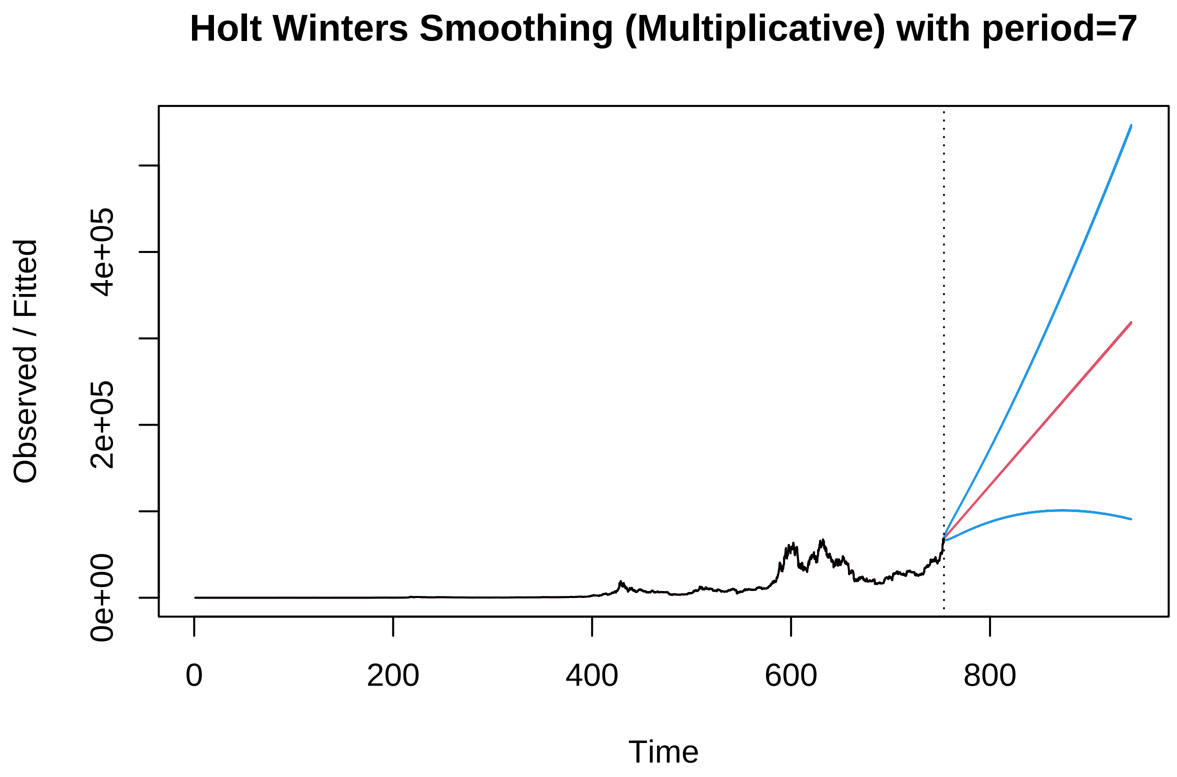
```
best_model_pred_pwr <- predict(best_model_pwr, n.ahead=length(new_test_ts),
                           prediction.interval = TRUE, level = 0.95)
df <- as.data.frame(best_model_pred)
tim.test = time(new_test_ts, offset=length(new_train_ts))
plot(new_data_ts)
lines(tim.test, df$fit, col = "red", ty='l')
lines(tim.test,df$upr, col = "blue", ty='l', lty = 2)
lines(tim.test,df$lwr, col = "blue", ty='l', lty = 2)
```



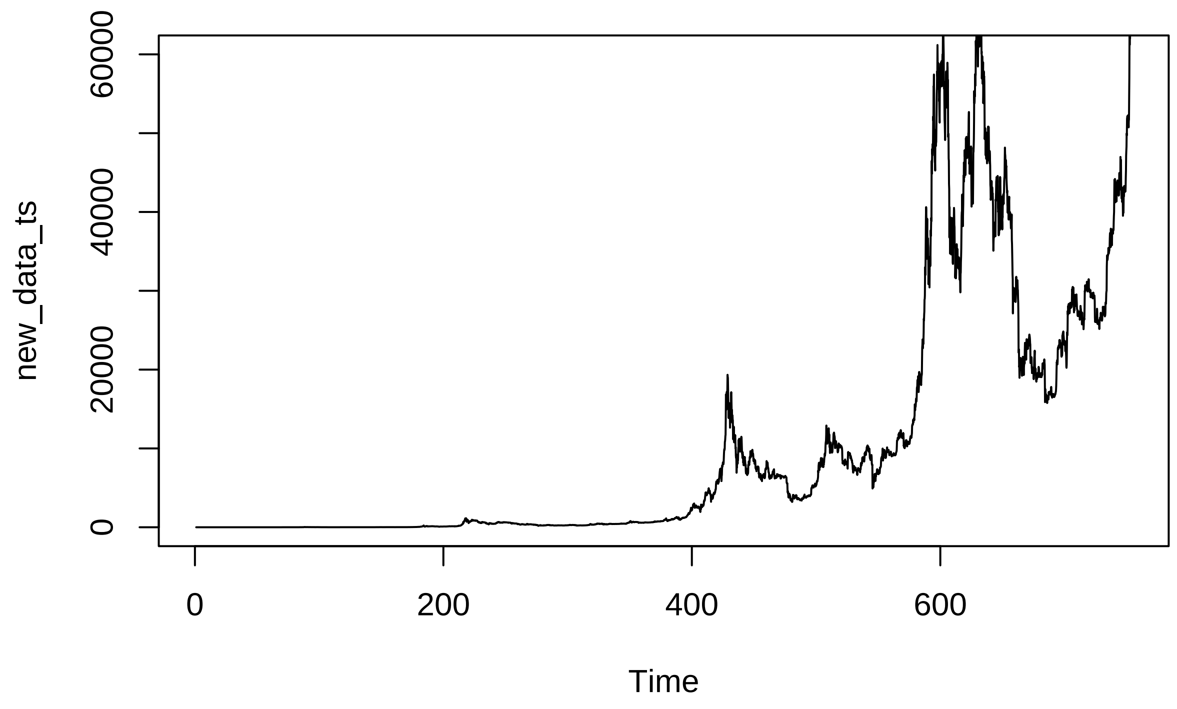
```r
## Forecasting the best model based on the APSE.
plot(tim.test, df$fit)
lines(tim.test,df$upr, col = "blue", ty='l', lty = 2)
lines(tim.test,df$lwr, col = "blue", ty='l', lty = 2)
```



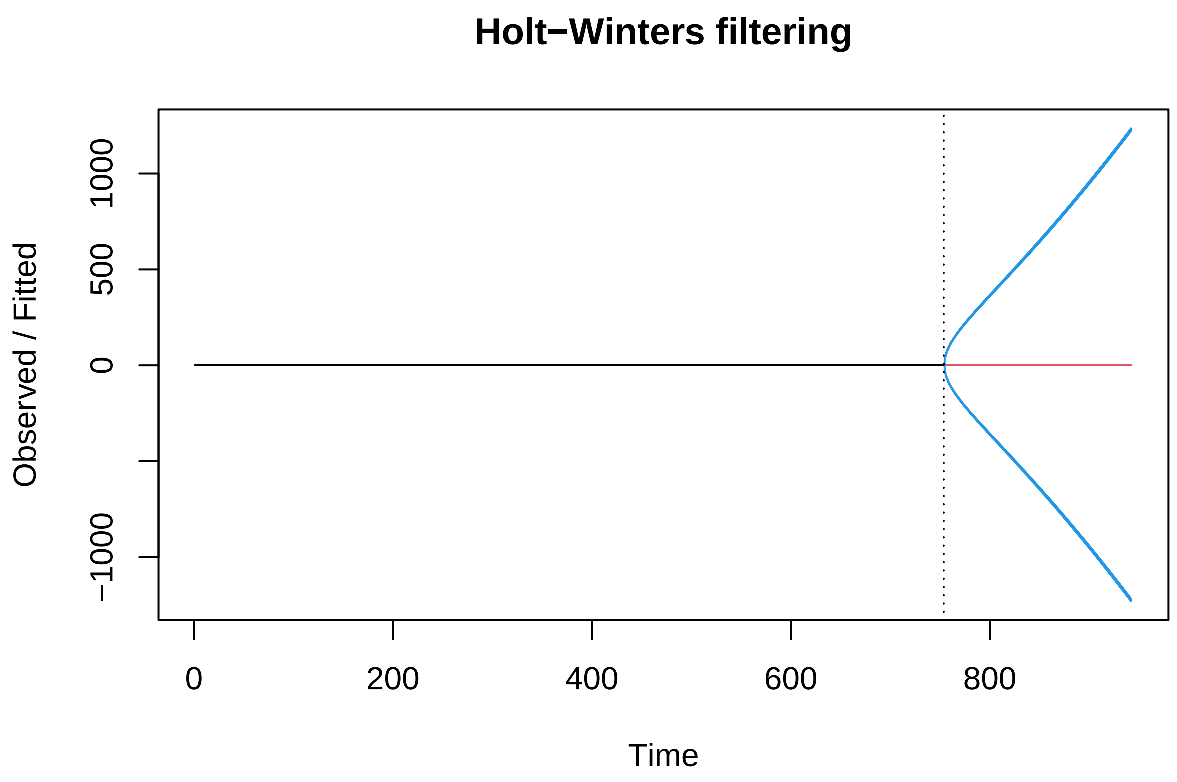
```r
plot(best_model, best_model_pred, main="Holt Winters Smoothing (Multiplicative) with period=7")
lines(length(new_train_ts) + 1:length(new_test_ts), new_test_ts, col="yellow")
```



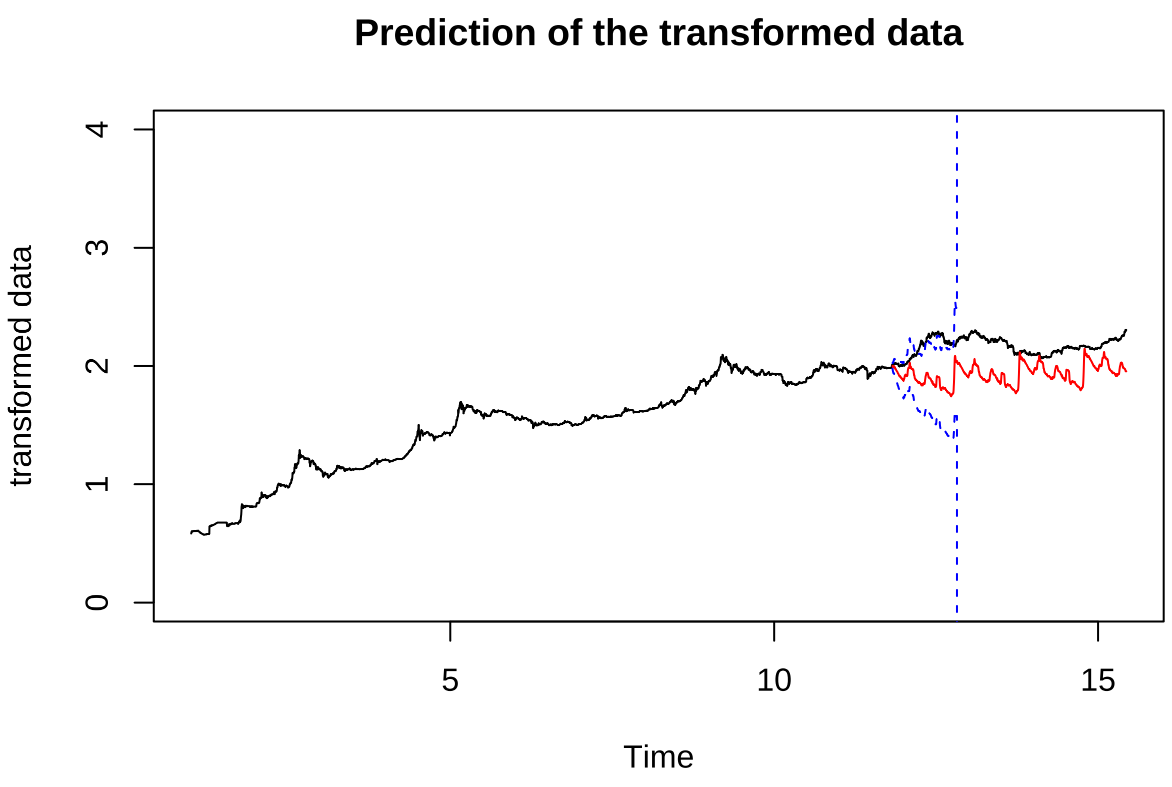
```r
plot(new_data_ts, ylim=c(0, 60000))
```



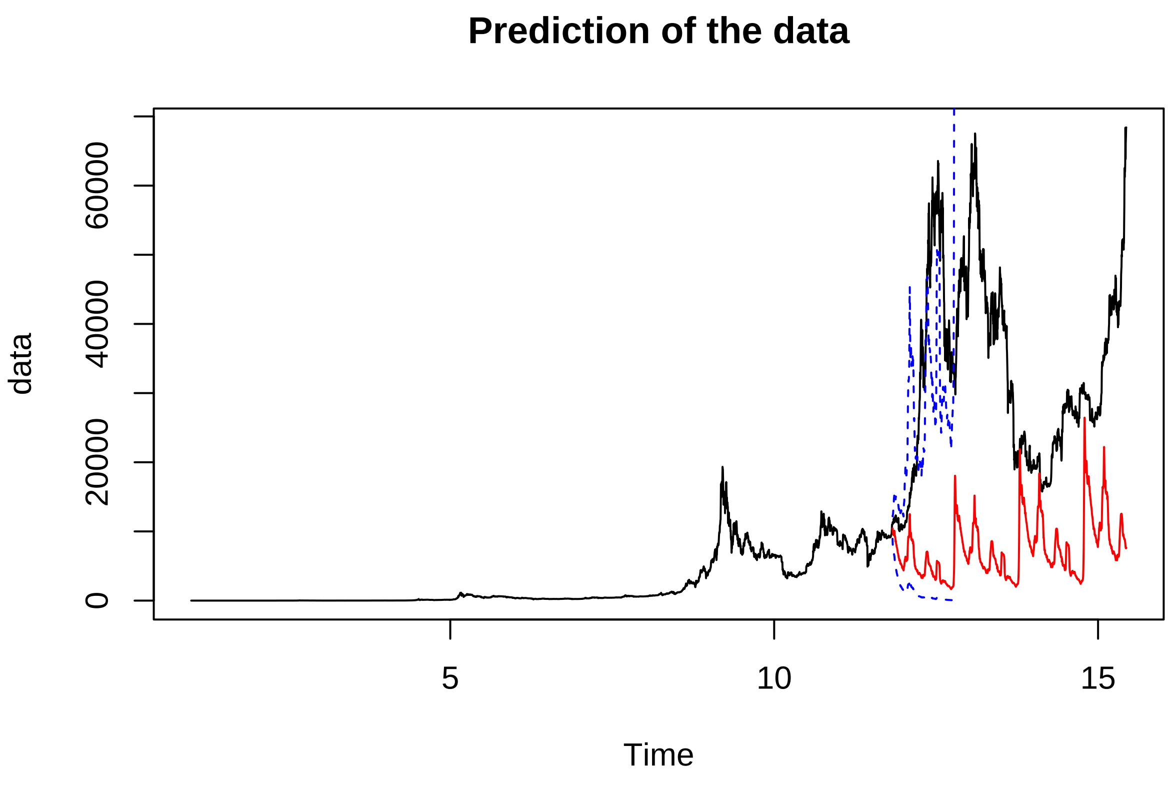
```
plot(best_model_pwr, best_model_pred_pwr )
```



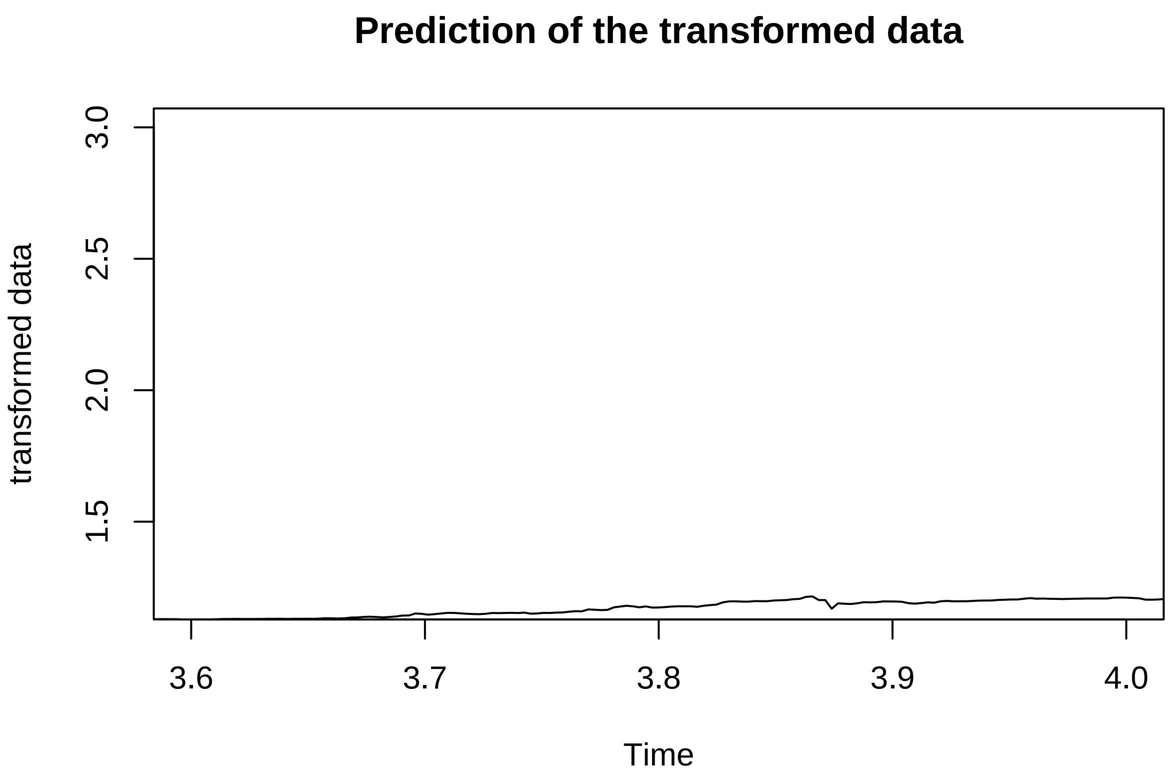
```
period = 365
new_data_ts = ts(data$price, frequency=period)
new_power_data_ts = new_data_ts^0.075
new_train_ts <- ts(train$price, frequency=period)
new_power_train_ts = new_train_ts^0.075
new_test_ts <- ts(test$price, frequency=period)
best_model_pwr <- HoltWinters(new_power_train_ts, seasonal="multiplicative")
best_model_pred_pwr <- predict(best_model_pwr, n.ahead=length(new_test_ts),
                           prediction.interval = TRUE, level = 0.95)
df <- as.data.frame(best_model_pred_pwr)
tim.test = as.vector(time(new_test_ts, offset=length(new_train_ts)))

plot(new_power_data_ts, ylim=c(0, 4), main="Prediction of the transformed data", ylab="transformed data
lines(tim.test, df$fit, col = "red")
lines(tim.test,df$upr, col = "blue", lty = 2)
lines(tim.test,df$lwr, col = "blue", lty = 2)
```



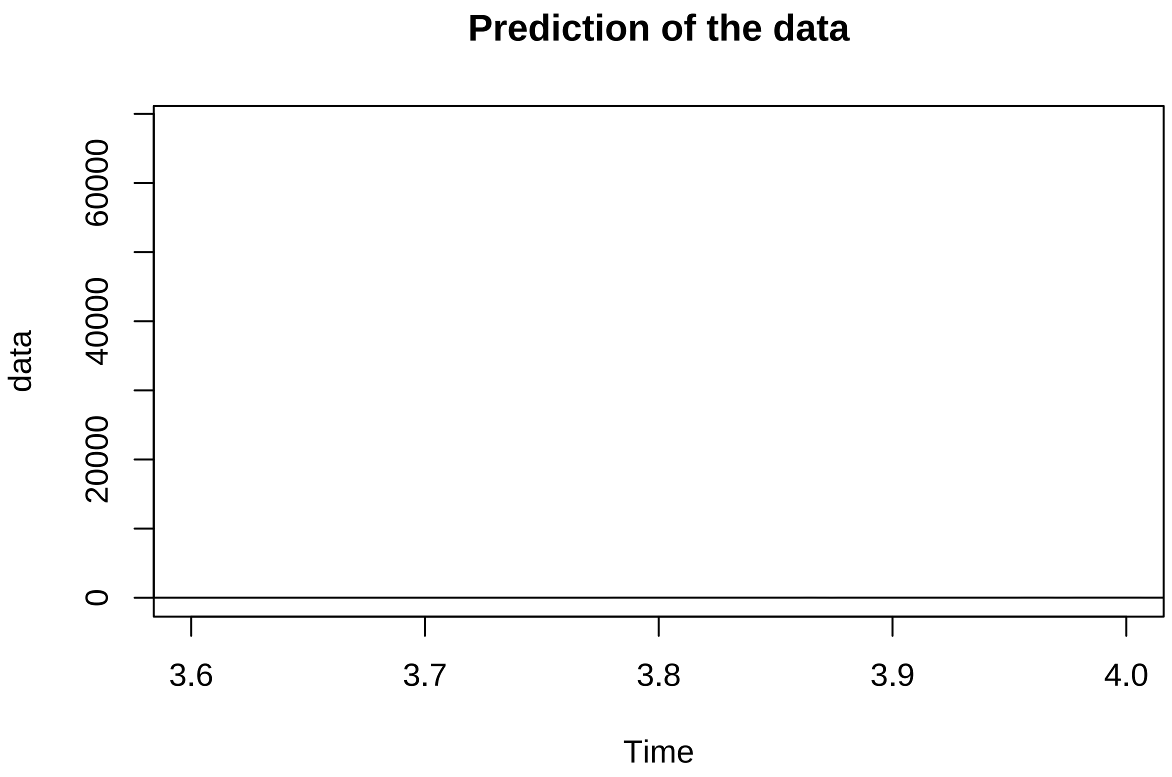
```r
plot(new_data_ts, main="Prediction of the data", ylab="data")
lines(tim.test, (df$fit)^(1/0.075), col = "red")
lines(tim.test,(df$upr)^(1/0.075), col = "blue", lty = 2)
lines(tim.test,(df$lwr)^(1/0.075), col = "blue", lty = 2)
```



```r
plot(new_power_data_ts, xlim=c(3.6, 4.0),  ylim=c(1.2, 3), main="Prediction of the transformed data", yl
lines(tim.test, df$fit, col = "red")
lines(tim.test,df$upr, col = "blue", lty = 2)
lines(tim.test,df$lwr, col = "blue", lty = 2)
```



```r
plot(new_data_ts, xlim=c(3.6, 4.0), main="Prediction of the data", ylab="data")
lines(tim.test, (df$fit)^(1/0.075), col = "red")
lines(tim.test,(df$upr)^(1/0.075), col = "blue", lty = 2)
lines(tim.test,(df$lwr)^(1/0.075), col = "blue", lty = 2)
```


Prediction of the data
data
0
20000
40000
60000
3.6
3.7
3.8
3.9
4.0
Time