Econ 144 Project 1: Analysis of Tesla's Stock Price

Yucong Chen, Yijia He, Anqi Li

Due 1/27/2023

Contents

I. Introduction

Founded in 2003, Tesla is the biggest multinational automotive and clean energy company in America, the best-selling electric car company, and the publicly traded automaker with the highest market capitalization. It is therefore crucial to analyze how Tesla has developed into its state now and also intriguing to see if it has the potential to keep evolving.

In this project, we will construct various time series plots on Tesla's stock price and build models to forecast the future stock prices. We will also perform trend and seasonal adjustments to the data. The dataset we use contains the daily data from January 1, 2011, to January 9, 2023 with variables Date, Open stock price, High stock price, Low stock price, Close price, Adjusted Close price, and Volume. We will use the Adjusted Close price for our model and focus on data from the latest five years.

II. Results

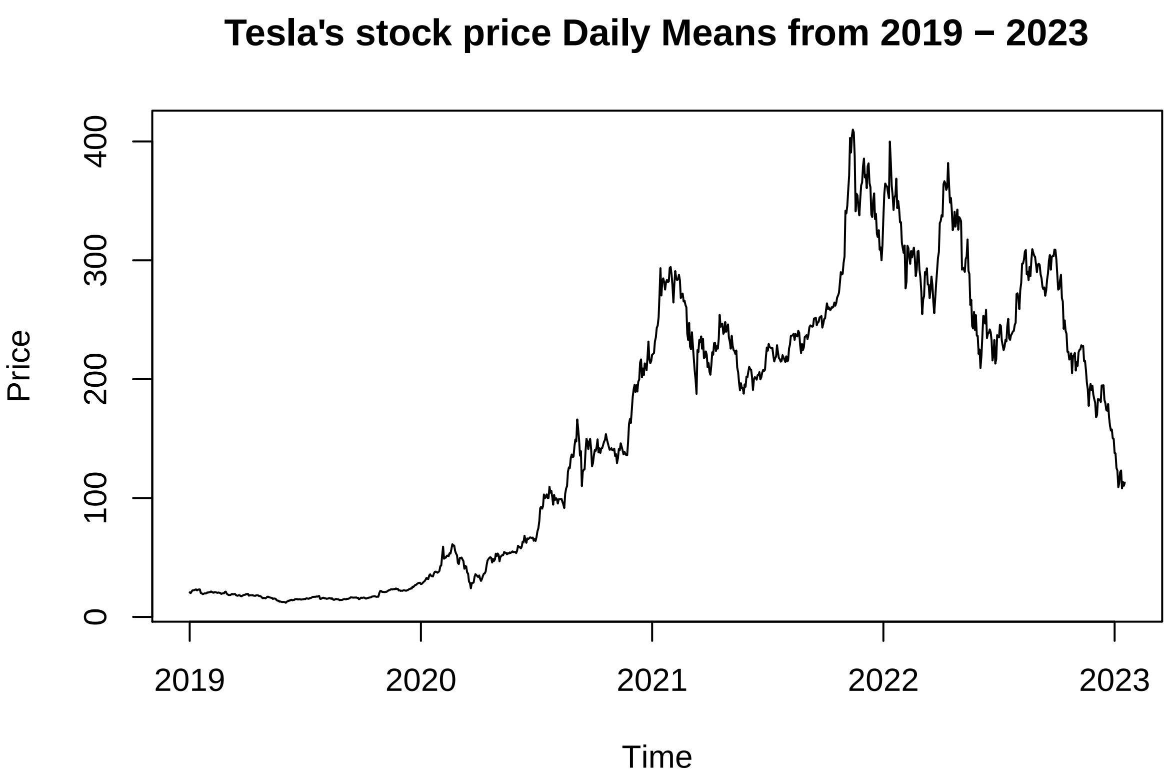
1. Modeling and Forecasting Trend

(a) Show a time-series plot of your data.

```
TSLA <- read_csv("TSLA.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )
```

```
TSLA <- TSLA[TSLA$Date >= "2019-01-01", ]
TSLA_ts = ts(TSLA$`Adj Close`, start = 2019, freq = 250)
t <- seq(2019, 2023, length = length(TSLA_ts))
plot(TSLA_ts, main = "Tesla's stock price Daily Means from 2019 - 2023", ylab = "Price")
```

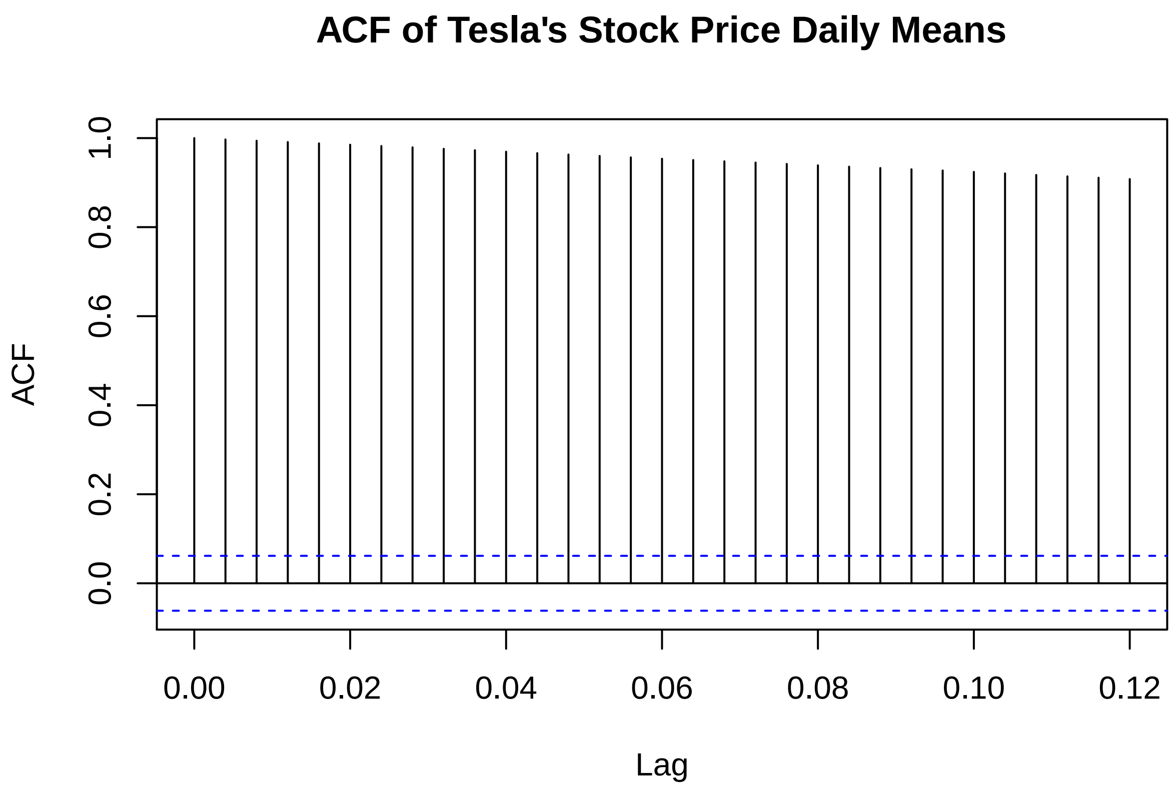


(b) Does your plot in (a) suggest that the data are covariance stationary? Explain your answer.

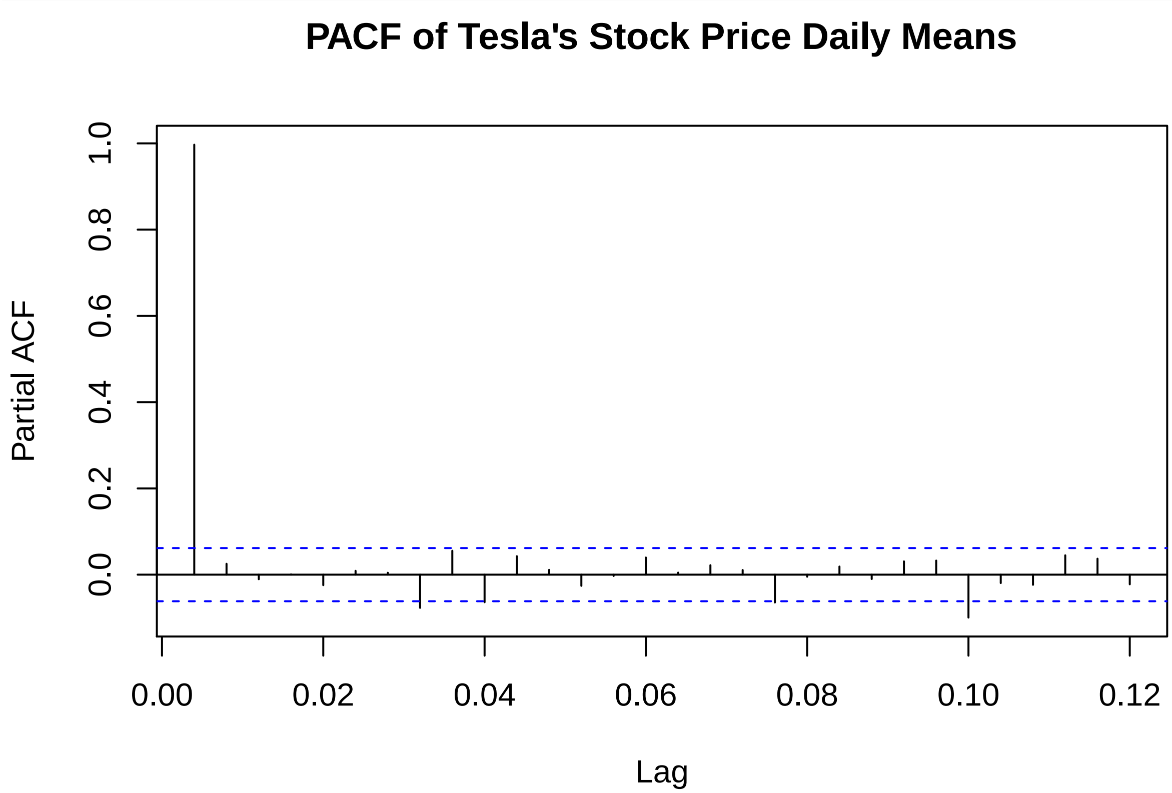
Our plot shows that there's an overall upward trend in Tesla's stock price. There is no first or second order stationarity. For the data to be covariance stationary, it should be second order weakly stationary. But we can see from the plot that the means and the variances are all not the same. So the data is not covariance stationary.

(c) Plot and discuss the ACF and PACF of your data.

```r
acf(TSLA_ts, main = "ACF of Tesla's Stock Price Daily Means")
```



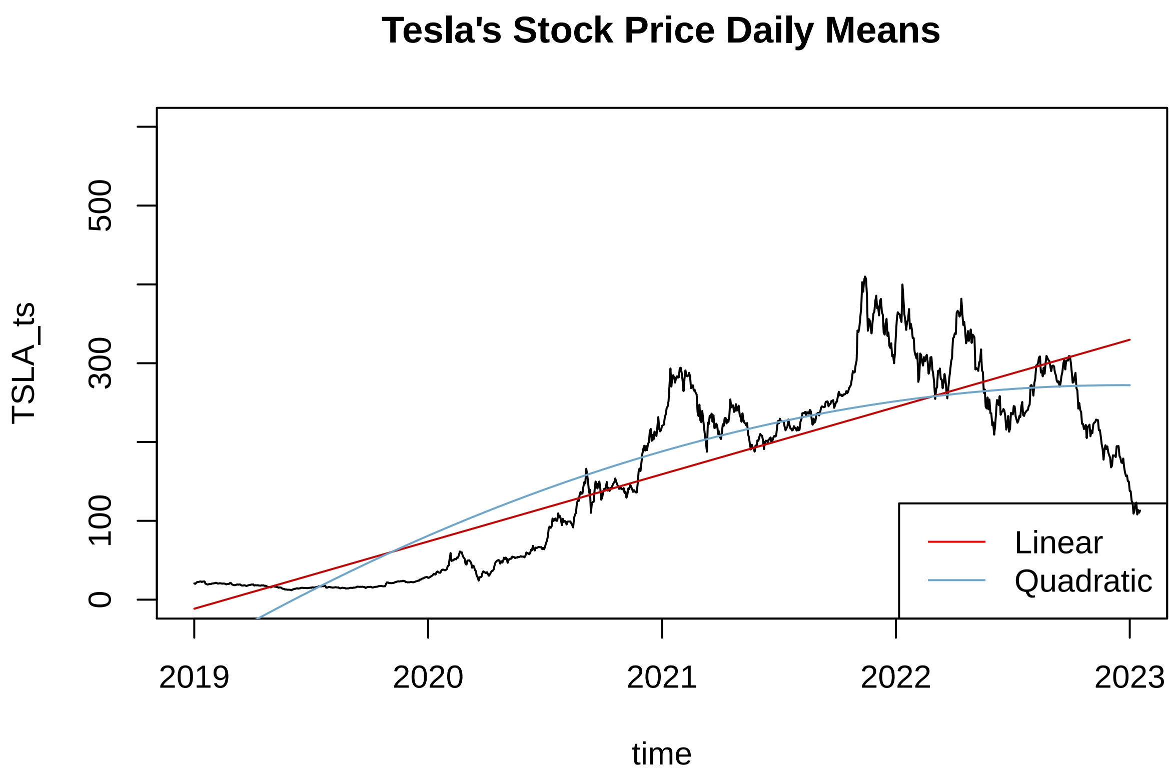

```r
pacf(TSLA_ts, main = "PACF of Tesla's Stock Price Daily Means")
```



The ACF plot of Tesla's stock price daily means shows that all times are significantly non-zero (hence time series nonrandom). There is a slow decay in the autocorrelation over a wide range of lags. While the PACF of Tesla's daily means has a high value at first and no evidence of time dependence past this. There are two bars outside of the 95% confidence interval at the lag of 0.03 and 0.10.

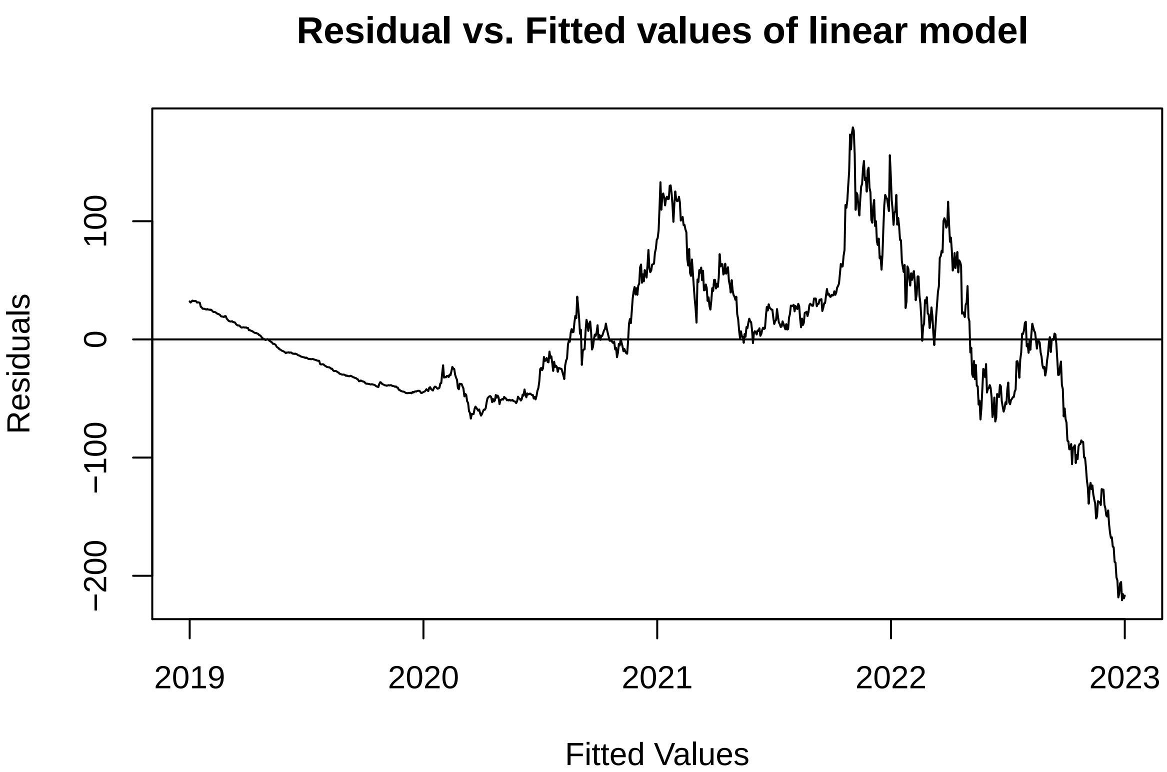
(d) Fit a linear and nonlinear (e.g., polynomial, exponential, quadratic + periodic, etc.) model to your series. In one window, show both figures of the original times series plot with the respective fit.

```
t <- seq(2019, 2023, length = length(TSLA_ts))

# Linear model
plot(TSLA_ts, xlab = "time", lwd = 1, col = 'black', ylim = c(0,600), xlim = c(2019,2023),
     main = "Tesla's Stock Price Daily Means")
linear <- lm(TSLA_ts~t)
lines(t, linear$fit, col="red3", lwd=1)

# Non-linear model: quadratic
nonlinear <-lm(TSLA_ts~t+I(t^2))
lines(t, nonlinear$fit, col="skyblue3", lwd=1)
legend("bottomright", legend = c("Linear", "Quadratic"), lwd = 1,col = c("red","skyblue3"))
```



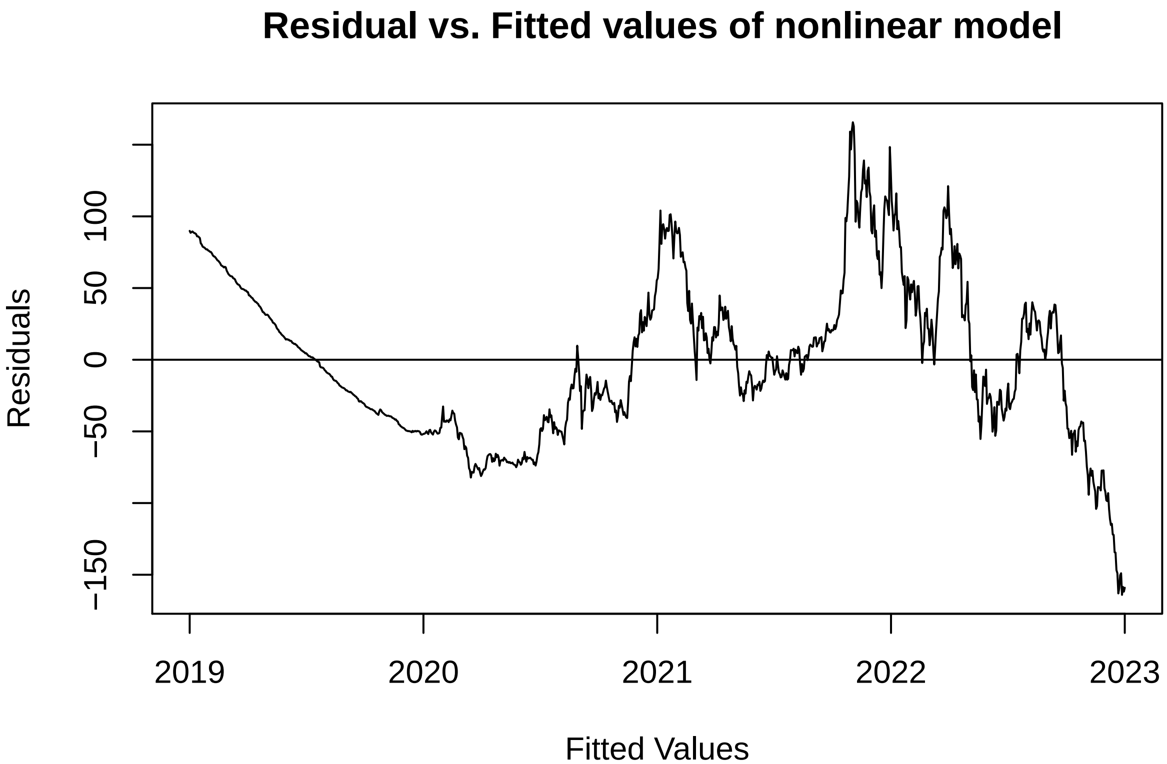
(e) For each model, plot the respective residuals vs. fitted values and discuss your observations.

```
plot(t, linear$res, main = " Residual vs. Fitted values of linear model",
     ylab = "Residuals", type = 'l', xlab = "Fitted Values")
abline(0,0)
```



The linear model's fitted values and residual graph shows that the model overpredicts, underpredicts, and then overpredicts again. And the n shape shows that the time series trend does not follow a linear pathway. The range goes from -200 to 200. There is a horizontal line representing the mean of residuals at zero essentially.

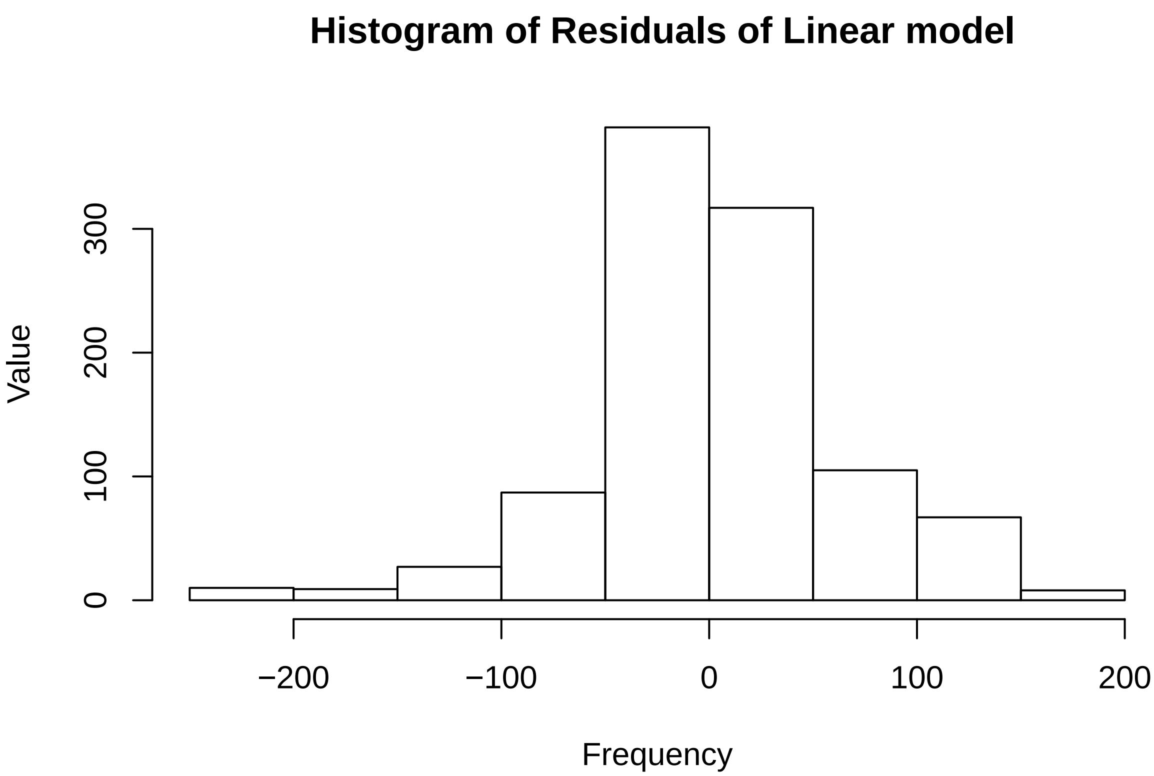
```
plot(t, nonlinear$res, main = " Residual vs. Fitted values of nonlinear model",
     ylab = "Residuals", type = 'l', xlab = "Fitted Values")
abline(0,0)
```



The non-linear model's fitted values and residual graph shows a small improvement compared to the linear model. There is still an n-shape in the plot but is much better than the linear plot showing an overall better trend fitting. The range of the residuals gets narrower from -150 to 150. The polynomial model explains the overall trend of Tesla's stock price significantly better than a plain linear model but still fails to explain seasonality with the trend.

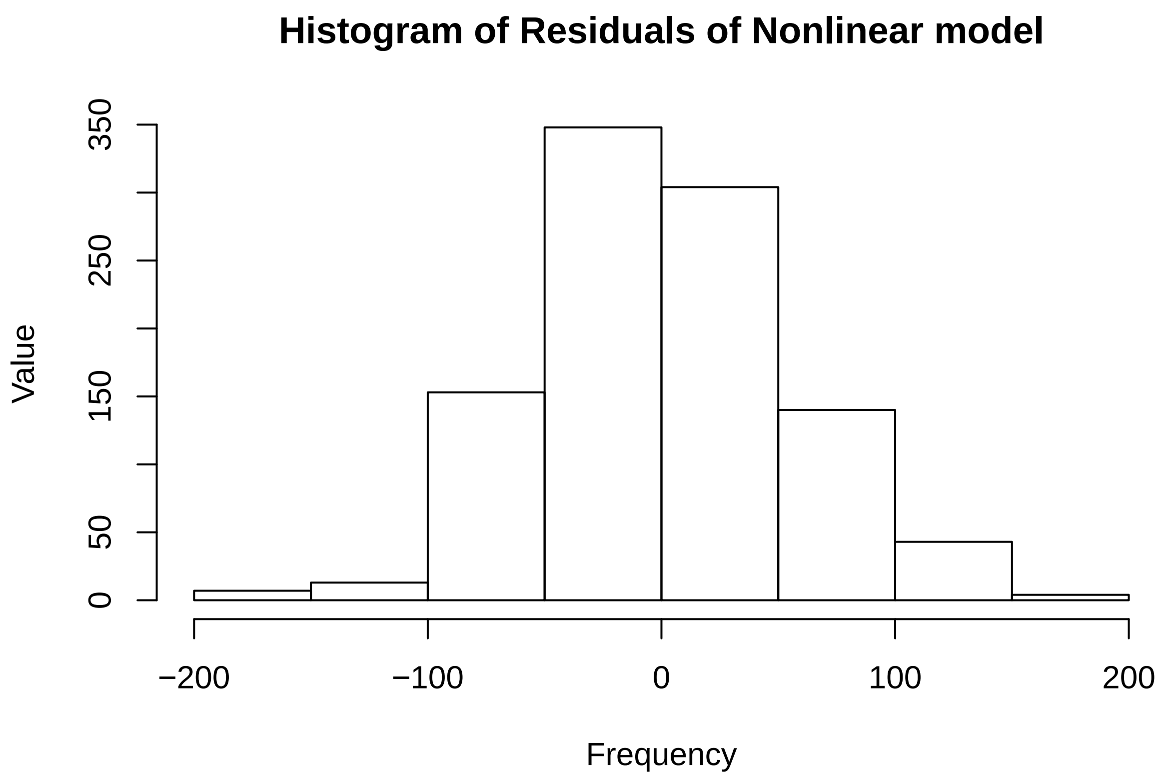
(f) For each model, plot a histogram of the residuals and discuss your observations.

```
hist(linear$residuals, main = " Histogram of Residuals of Linear model"
     , ylab = "Value", xlab = "Frequency")
```



We can see that the linear model residuals are a little bit left-skewed. It fails to capture the seasonal components. However, from the time series and fitted values vs. residual plot, the under and over-predictions canceled each other out in this histogram showing that the linear model is somewhat an ok model.

```
hist(nonlinear$residuals, main = "Histogram of Residuals of Nonlinear model"
     , ylab = "Value", xlab = "Frequency")
```



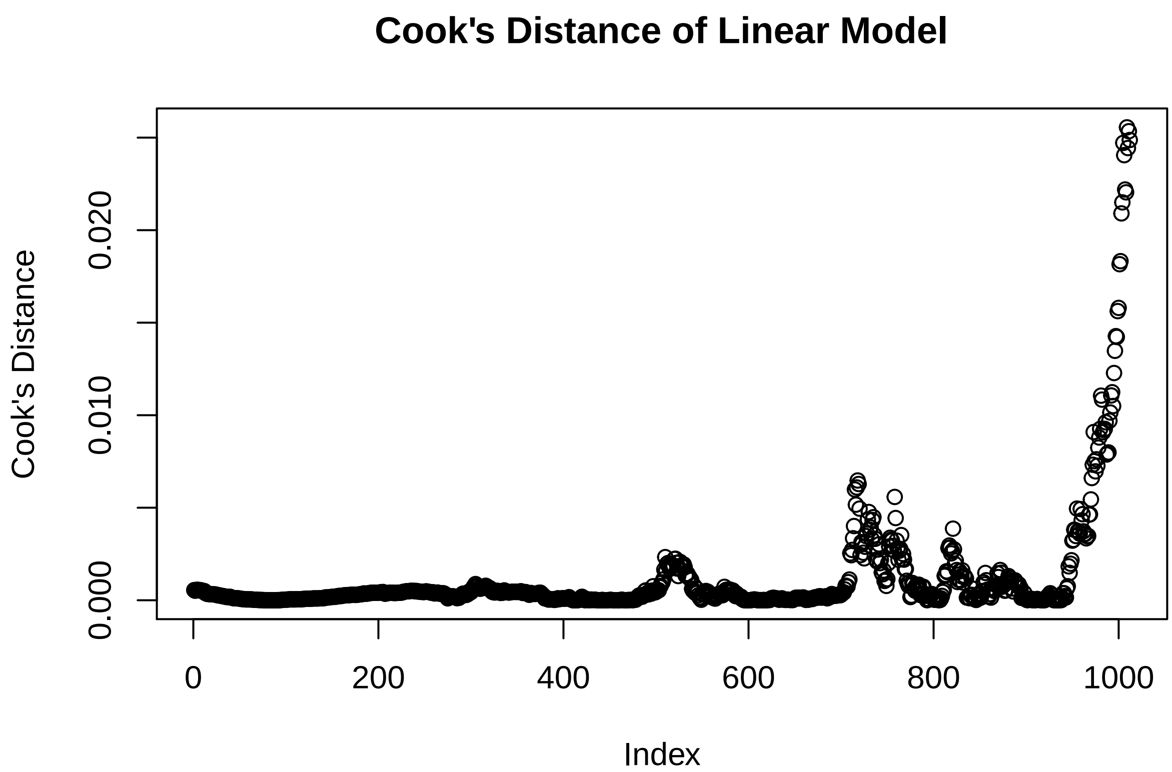
The non-linear model residuals are uniformly distributed showing it is a better model compared to the linear model. It is closer to normality, which means that our estimates are more unbiased and consistent.

(g) For each model, discuss the associated diagnostic statistics (R^2, t-distribution, F-distribution, etc.)

```r
summary(linear)
```

```
##
## Call:
## lm(formula = TSLA_ts ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -220.693  -38.660   -0.702   31.579  179.422
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.723e+05  3.369e+03  -51.14   <2e-16 ***
## t            8.533e+01  1.667e+00   51.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.29 on 1010 degrees of freedom
## Multiple R-squared:  0.7218, Adjusted R-squared:  0.7215
## F-statistic:  2620 on 1 and 1010 DF,  p-value: < 2.2e-16
```

```r
cooks_lr <- cooks.distance(linear)
plot(cooks_lr, main = "Cook's Distance of Linear Model", ylab = "Cook's Distance")
```

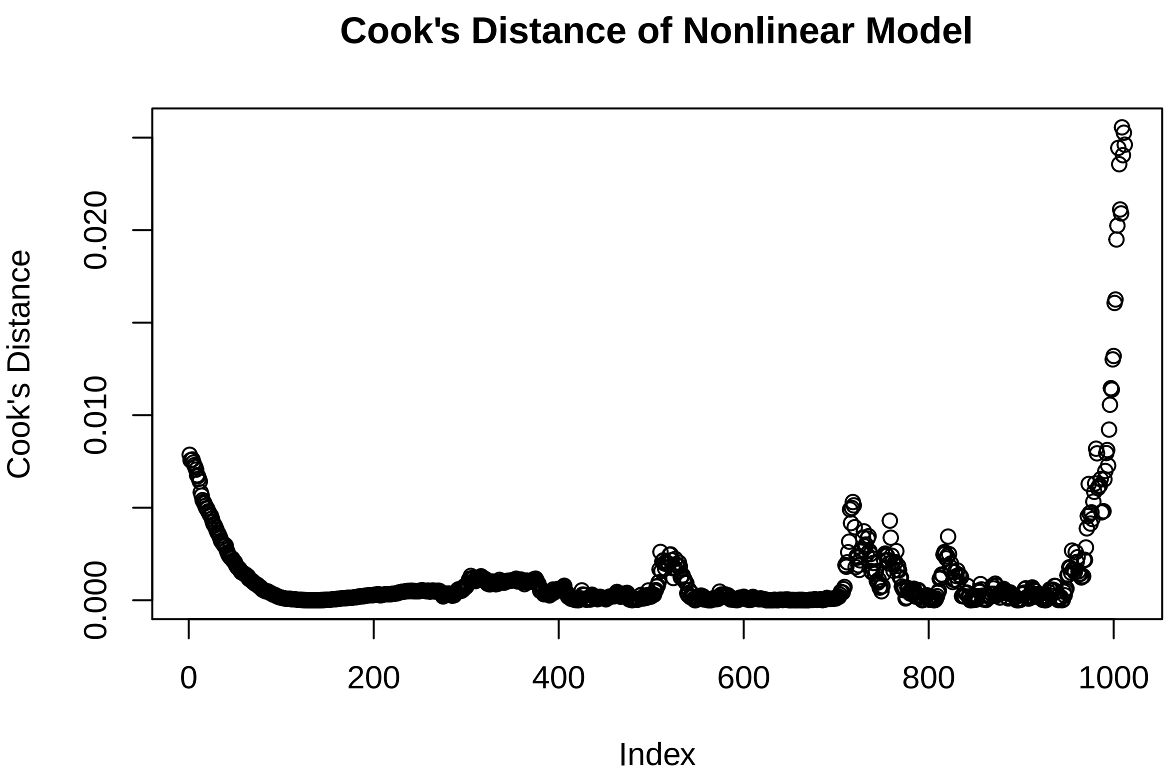


We can see from the summary that the ts value and intercept are statistically significant because of the high t values and the low p-value. The adjusted R-squared value is 72.15% showing that the linear model is performing ok. The F-statistic also has a statistically significant p-value. The cook's distances are all pretty low (below 0.025) showing there are no super extreme values in data that influence the model.

```
summary(nonlinear)
```

```
##
## Call:
## lm(formula = TSLA_ts ~ t + I(t^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -164.033  -40.273   -3.464   33.333  165.656
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.861e+07  5.972e+06  -14.84   <2e-16 ***
## t            8.761e+04  5.910e+03   14.82   <2e-16 ***
## I(t^2)      -2.165e+01  1.462e+00  -14.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.58 on 1009 degrees of freedom
## Multiple R-squared:  0.7715, Adjusted R-squared:  0.771
## F-statistic:  1703 on 2 and 1009 DF,  p-value: < 2.2e-16
```

```
cooks_nlr <- cooks.distance(nonlinear)
plot(cooks_nlr, main = "Cook's Distance of Nonlinear Model", ylab = "Cook's Distance")
```



We can see from the summary that the ts value and intercept are statistically significant because of the high t values and the low p-value. The adjusted R-squared value is 77.1% compared to the linear model showing that the nonlinear model is better. The F-statistic also has a statistically significant p-value. The cook's distances are also pretty low (below 0.025), showing that there are also no super extreme values in data that influence the model.

(h) Select a trend model using AIC and one using BIC (show the values obtained from each criterion). Do the selected models agree?

```
aic <- AIC(linear, nonlinear)
bic <- BIC(linear, nonlinear)
tscores <- data.frame(aic[,2], bic[,2], row.names=c('Linear', 'Nonlinear'))
scores <- setNames(tscores, c('AIC', 'BIC'))
grid.table(scores)
```

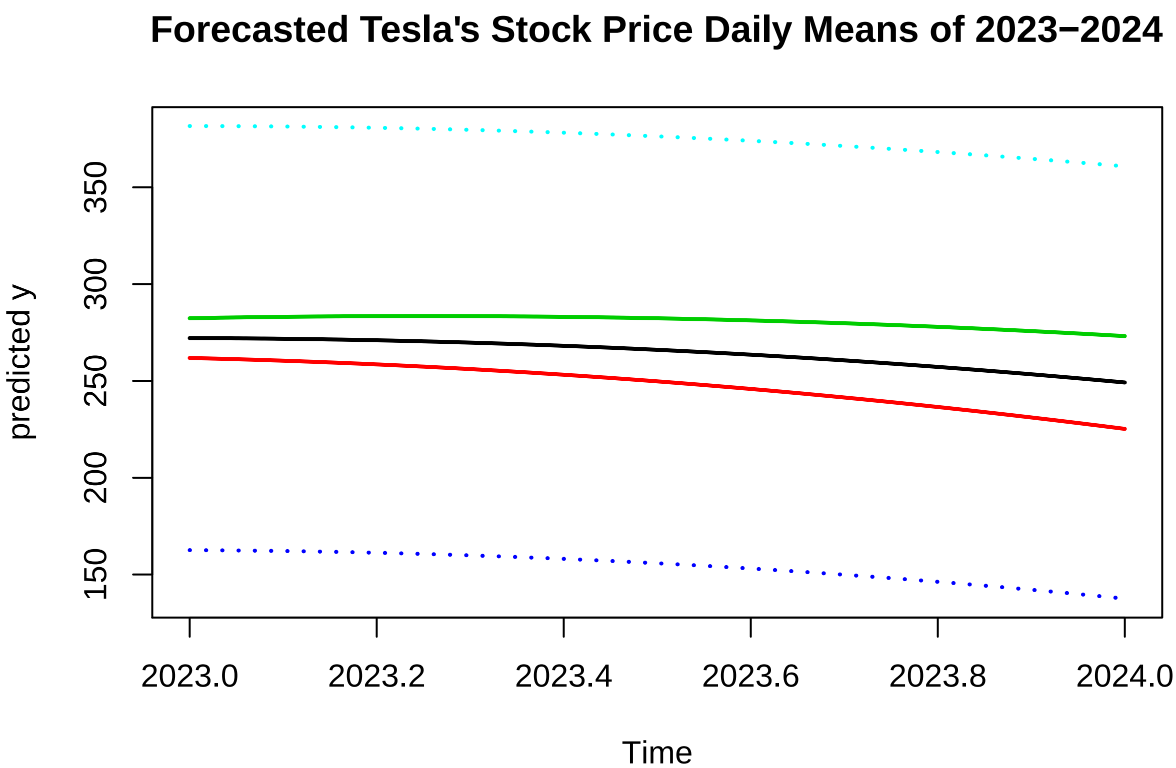
	AIC	BIC
Linear	11206.0436955209	11220.8027470705
Nonlinear	11008.9986975907	11028.6774329901

We can see from the table that both the AIC and BIC scores show that we should select the nonlinear model because the values are both smaller.

(i) Use your preferred model to forecast h-steps (at least 16) ahead. Your forecast should include the respective uncertainty prediction interval. Depending on your data, h will be in days, months, years, etc.

```
tn=data.frame(t=seq(2023,2024, length = 365))
pred=predict(nonlinear, tn, se.fit = TRUE)
pred.plim = predict(nonlinear, tn, level =0.95, interval="prediction")
pred.clim = predict(nonlinear, tn,level=0.95, interval="confidence")
```

```r
matplot(tn$t,cbind(pred.clim, pred.plim[,-1]),
        lty=c(1,1,1,3,3), type="l", lwd=2, ylab="predicted y",xlab="Time",
        main = "Forecasted Tesla's Stock Price Daily Means of 2023-2024")
```

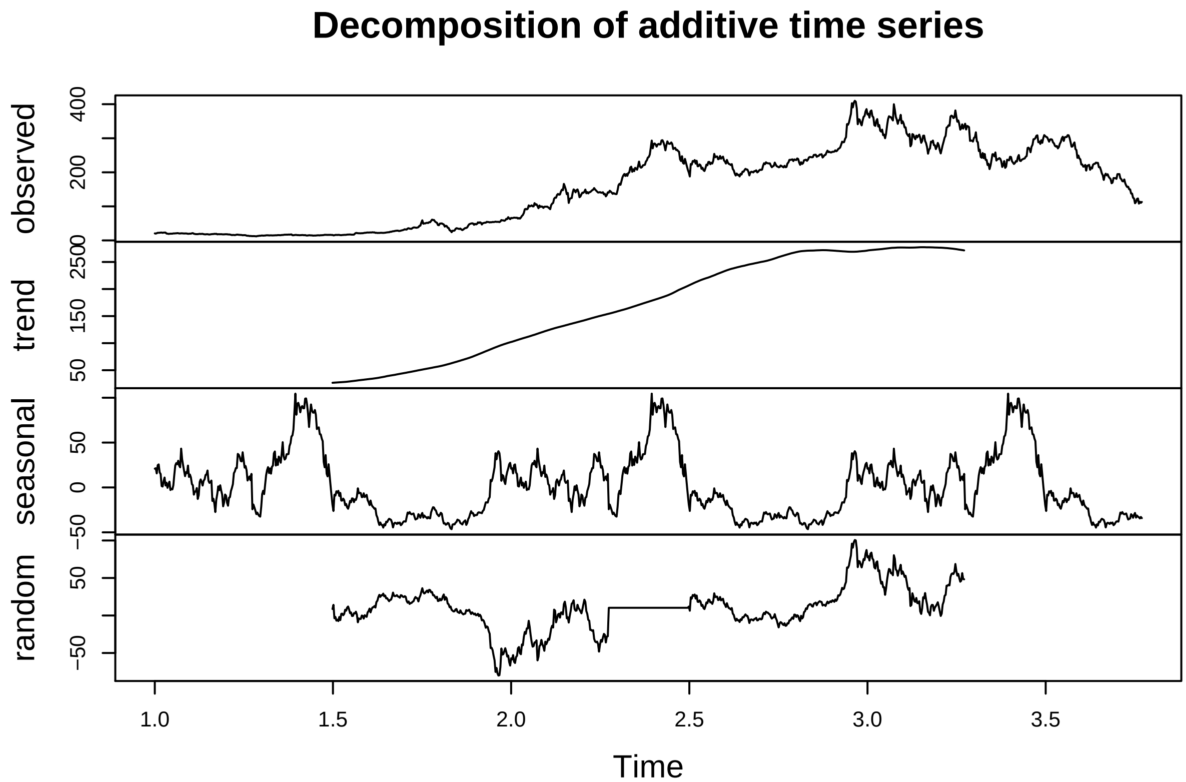


The forecast plot shows the nonlinear model's daily forecast for the next whole year. As we can see from the graph, the forecast follows a path that goes down to continue from the fitted values.

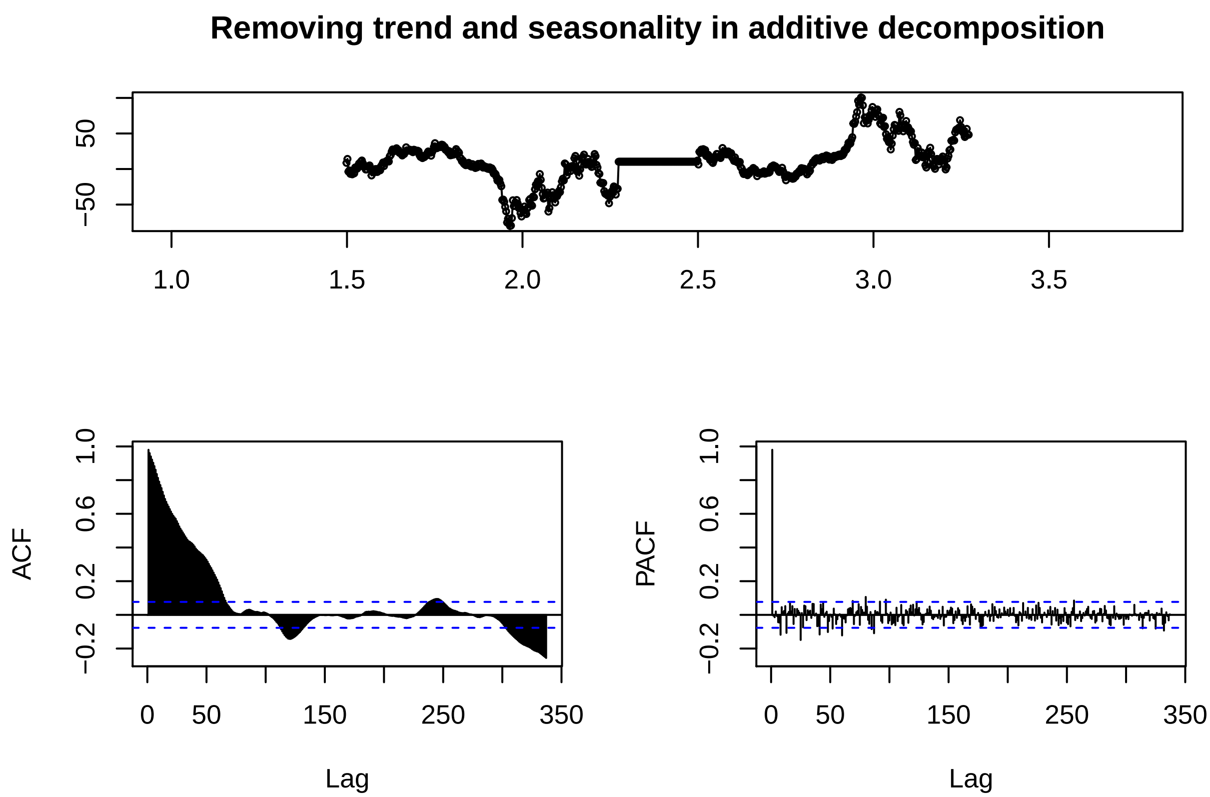
2. Trend and Seasonal Adjustmennts

(a) Perform an additive decomposition of your series. Remove the trend and seasonality, and comment on the ACF and PACF of the residuals (i.e., what is left after detrending and seasonally adjusting the series). Comment on the results.

```
dcmp_add = decompose(ts(TSLA_ts, frequency=365), "additive")
plot(dcmp_add)
```




```
cost_add = ts(TSLA_ts, frequency=365)
detrend_seas_adj_add = cost_add - dcmp_add$trend - dcmp_add$seasonal
tsdisplay(detrend_seas_adj_add, main = "Removing trend and seasonality in additive decomposition")
```

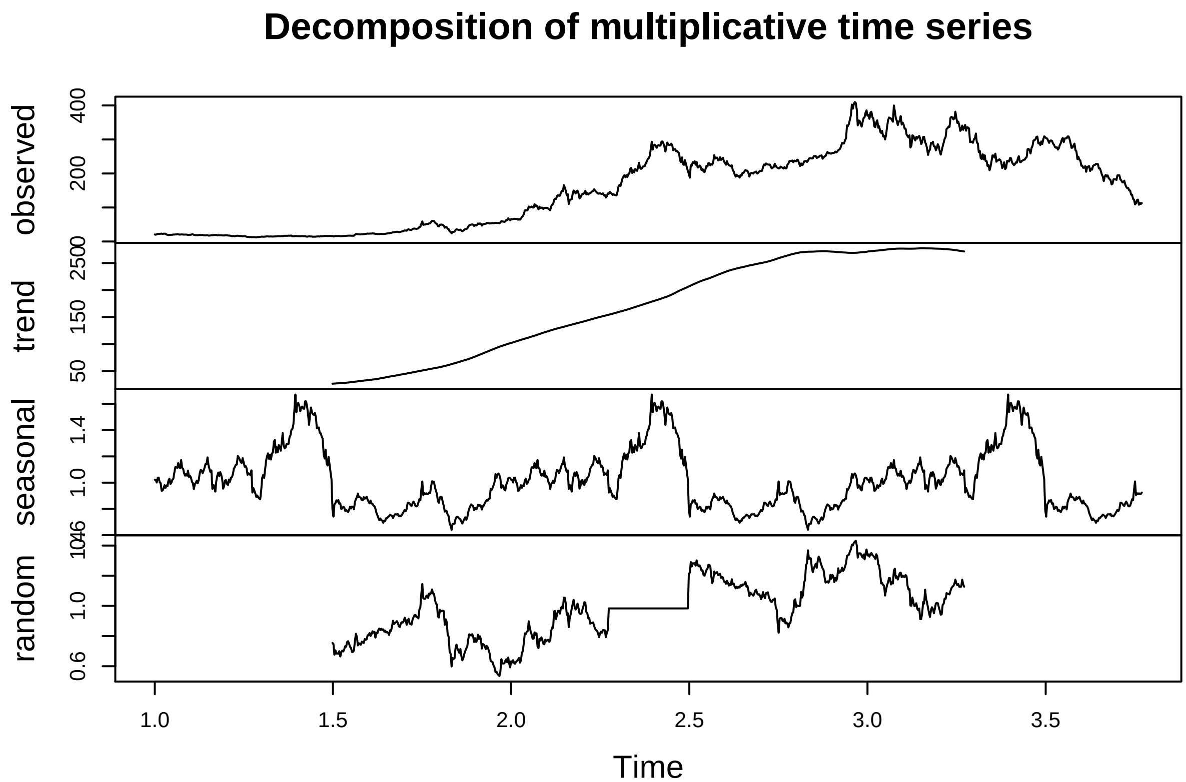


In the additive decomposition plot, we can see that the trend is mostly well-captured as an increasing line that behaves similarly with the data, except for the drop at the end that is shown in the time series plot but not on the trend plot. The seasonal component shows that there is a peak in the middle of each year, which is not shown in our previous models. Moreover, the random component shows that there are many small positive residuals at the beginning, and then there are many negative residuals which then become mostly zero, and finally mostly positive residuals at the end. It is worth noting that the time when the residuals are almost 0 corresponds to approximately the start of 2021 when the economy was starting to recover and there was less outside interference.

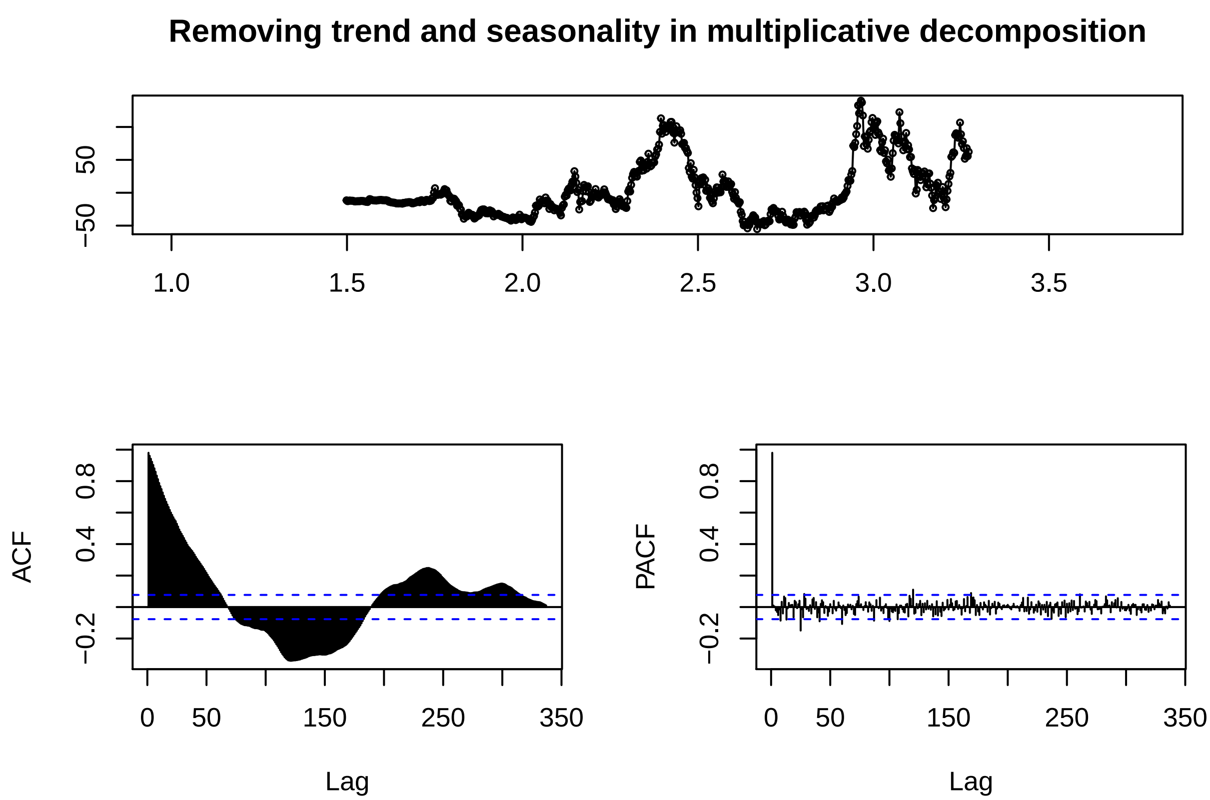
The ACF plot here shows that most times are significantly non-zero (hence time series nonrandom). There is a strong positive autocorrelation over the lags that decreases for the first 100 lags and then a negative autocorrelation from lags 100 to 150. There is another strong negative autocorrelation near the end. The PACF plot shows that there is a strong autocorrelation at the lags of around 10, 15, 25, 60, 80, 90.

(b) Perform a multiplicative decomposition of your series. Remove the trend and seasonality, and comment on the ACF and PACF of the residuals (i.e., what is left after detrending and seasonally adjusting the series). Comment on the results.

```
dcmp_multi = decompose(ts(TSLA_ts, frequency=365), "multiplicative")
plot(dcmp_multi)
```




```
cost_multi = ts(TSLA_ts, frequency=365)
detrend_seas_adj_multi = cost_multi - dcmp_multi$trend - dcmp_multi$seasonal
tsdisplay(detrend_seas_adj_multi, main = "Removing trend and seasonality in multiplicative decompositio
```



In the multiplicative decomposition plot, we can see a similar trend, seasonal component, and random plot. However, the peak in the seasonal component here lasts a bit longer, and the residues seem to be closer to 0. The ACF plot shows that most times are significantly non-zero (hence time series nonrandom). There is a strong positive autocorrelation over the lags that decreases for the first 100 lags and then a negative autocorrelation from lags 75 to 175. There are a few strong negative autocorrelations near lag 230, but the autocorrelation decreases at the end. The PACF plot shows that there is a strong autocorrelation at the lags of around 25 and 80, fewer than what we see in the previous PACF plot.

(c) Which decomposition is better, additive or multiplicative? Why?

Multiplicative decomposition is better, as the overall residues are closer to 0 and the sum of squared ACF is smaller.

(d) Based on the two decompositions, and interpretation of the random components, would your models for the cycles be similar (additive vs. multiplicative) or very different? Why?

The cycles of both decompositions are similar, as the random components are very similar.

III. Conclusions and Future Work

In conclusion, we identified our multiplicative decomposition as the best-fit model to forecast Tesla's stock prices. In comparison to our previous linear and non-linear models, it not only captures the trend effectively but also shows the seasonality that is ignored in both models. In comparison to the additive decomposition, it shows fewer deviations in terms of residuals.

Nevertheless, our linear and non-linear models give us initial insights into the data, and our forecast using the non-linear model shows that Tesla's stock price will decrease in the next year. We recognize that this intermediate conclusion seems to contradict our results from the best-fit multiplicative decomposition, but this also reflects the importance of seasonality in time-series prediction, and intuitively speaking, Tesla does have the potential to keep developing as the world is switching to renewable energy and gradually recovering from COVID-19.

For our future work, we would suggest taking white noise into account in our future forecasting models, and more data collection such as hourly stock prices, which may give more information prediction.

IV. References

https://www.kaggle.com/datasets/rajkumarpandey02/tesla-inc-tsla-stock-price