BUSINESS FORECASTING

FINAL PROJECT

TESLA STOCK PRICE PREDICTION

Presented By:

Parth A Patel – 190000189

Urvi Vipani – 189007018

Dipti Kalathiya – 188009829

1. Introduction

Tesla, Inc. (formerly **Tesla Motors, Inc.**) is an American automotive and energy company based in Palo Alto, California. The company specializes in electric car manufacturing and, through its SolarCity subsidiary, solar panel manufacturing. It operates multiple production and assembly plants, notably Gigafactory 1 near Reno, Nevada, and its main vehicle manufacturing facility at Tesla Factory in Fremont, California. As of June 2018, Tesla sells the Model S, Model X and Model 3 vehicles, Powerwall and Powerpack batteries, solar panels, solar roof tiles, and some related products.

Tesla was founded in July 2003, by engineers Martin Eberhard and Marc Tarpenning, under the name Tesla Motors. The company's name was derived from physicist Nikola Tesla. In early Series A funding, Tesla Motors was joined by Elon Musk, J. B. Straubel and Ian Wright, all of whom are retrospectively allowed to call themselves co-founders of the company. Musk, who formerly served as chairman and is the current chief executive officer, said that he envisioned Tesla Motors as a technology company and independent automaker, aimed at eventually offering electric cars at prices affordable to the average consumer. Tesla Motors shortened its name to Tesla in February 2017.

The company received funding at various levels and eventually launched its IPO on NASDAQ with 13,300,000 shares each priced at $17.The company recognizes itself as an independent automaker, aimed at eventually offering electric cars at affordable price. Apart from the electric cars (Model S, Model X, and Model 3) the tesla also sells Powerwall and Powerpack batteries, solar panels, solar roof tiles and some related products. Elon Musk serves as Chairman & Chief Executive Officer, although he was asked to step down from chairman position by the SEC after recent controversy.

2. Background

For any investor, stock prices play an important role in decision making to buy the stock or not. We have tried and forecasted the stock price of tesla using around 1300 past observations of stock details. This kind of analysis and forecast results help understanding and prepare in advance to either buy or sell the shares.

In our project, we have used the **quantmod** package to download the data for Tesla. It is basically designed to assist the quantitative trader in the development, testing, and deployment of statistically based trading models. Also, a rapid prototyping environment, with comprehensive tools for data management and visualization, where quant traders can quickly and cleanly explore and build trading models. A replacement for anything statistical. It has no ’new’ modelling routines or analysis tool to speak of. It does now offer charting not currently available elsewhere in R, but most everything else is more of a wrapper to what you already know and love about the language and packages you currently use.

The quantmod-library makes modelling easier by removing the repetitive workflow issues surrounding data management, modelling interfaces, and performance analysis.

Following are the variables in the dataset:

- **TSLA.Open**: This variable stores the opening price of a stock on a particular day.
- **TSLA.High**: This stores the highest value of a particular stock on that particular day.
- **TSLA.Low**: This stores the lowest value of the stock for that particular day.
- **TSLA.Close**: This stores the closing price of the stock for that particular day.
- **TSLA.Volume**: This variable stores the amount of stocks traded on that particular day
- **TSLA.Adjusted**: This variable stores the adjusted closing price after adjusting with some other market scenarios pertaining to that particular stock.

3. Data Exploration

Data exploration is an approach similar to initial data analysis, whereby a data analyst uses visual exploration to understand what is in a dataset and the characteristics of the data, rather than through traditional data management systems. These characteristics can include size or amount of data, completeness of the data, correctness of the data, possible relationships amongst data elements or files/tables in the data.

Data exploration is typically conducted using a combination of automated and manual activities. Automated activities can include data profiling or data visualization or tabular reports to give the analyst an initial view into the data and an understanding of key characteristics.

It is basically required to understand the dataset better since it helps choose appropriate variables in the dataset. The Tesla Stock Price data has 6 variables. In data exploration we'll try to determine which platforms are the best for prediction of Tesla Stock price.

We will be using following libraries in our project:

Code:

```
library(ggplot2)

library(fpp2)

library(forecast)

library(fma)

library(expsmooth)

library(quantmod) #install.packages("quantmod")
```

4. Data Analysis

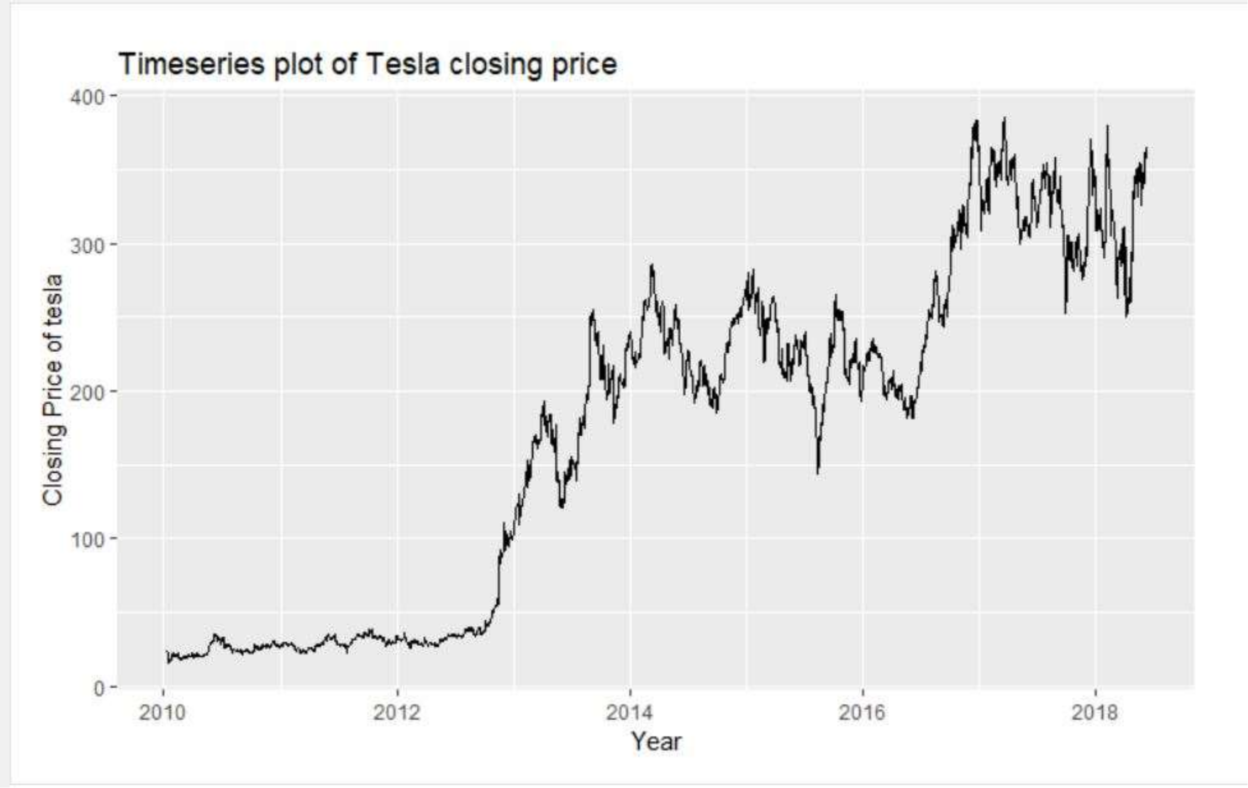
4.1 Timeseries plot of Tesla closing price:

We will start by plotting all the data points that we have. We observe upon plotting that there's an overall increasing trend and stock rose from mere $17 to around $350 in span of 8 years with two major jump in prices.

R Code:

```
autoplot(tsla_close) + xlab("Year") + ylab("Closing Price of tesla")+
  ggtitle("Timeseries plot of Tesla closing price")
```

Plot:



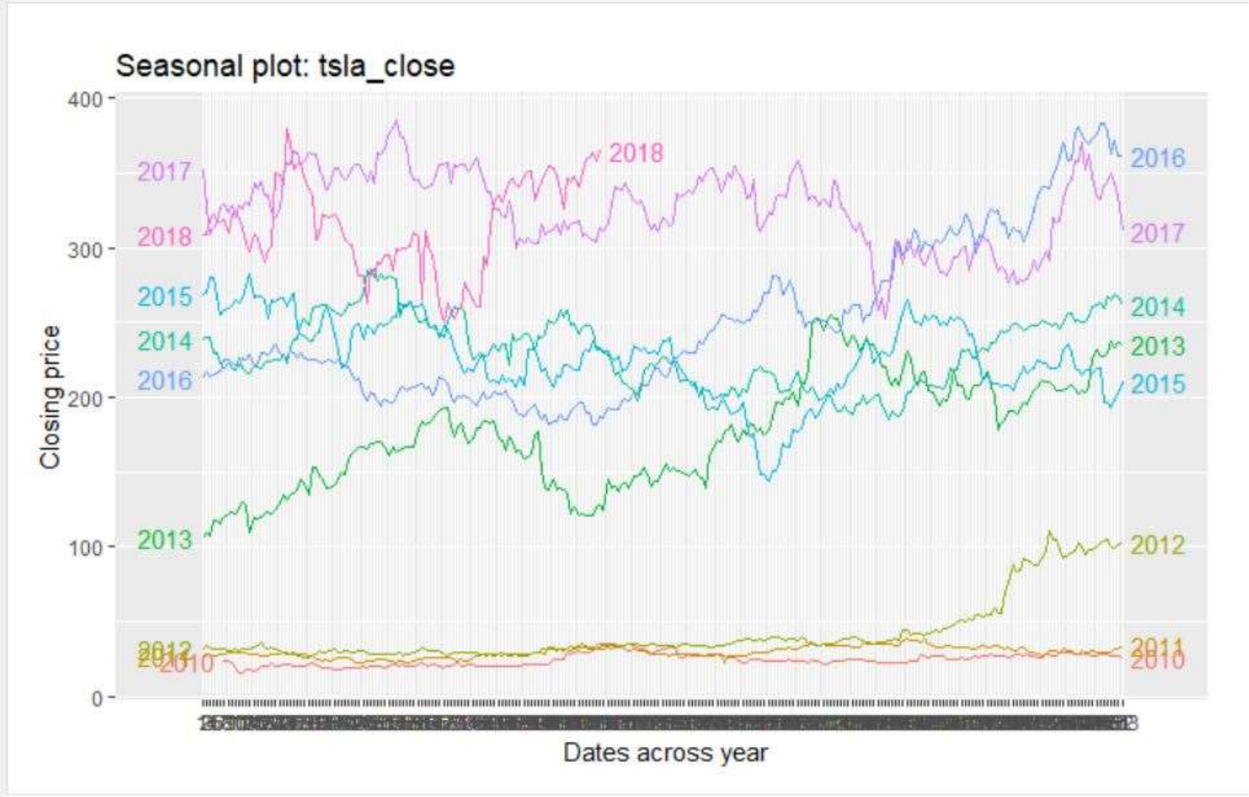
4.2 Seasonal Plot with Tesla Close:

Now we will also look at the graph season wise and that can be done by the below given code. A seasonal plot is similar to a time plot except that the data are plotted against the individual "seasons" in which the data were observed.

R Code:

```
ggseasonplot(tsla_close, year.labels = TRUE, year.labels.left = TRUE) + ylab("Closing price") +
          xlab(" Dates across year")
```

Plot:



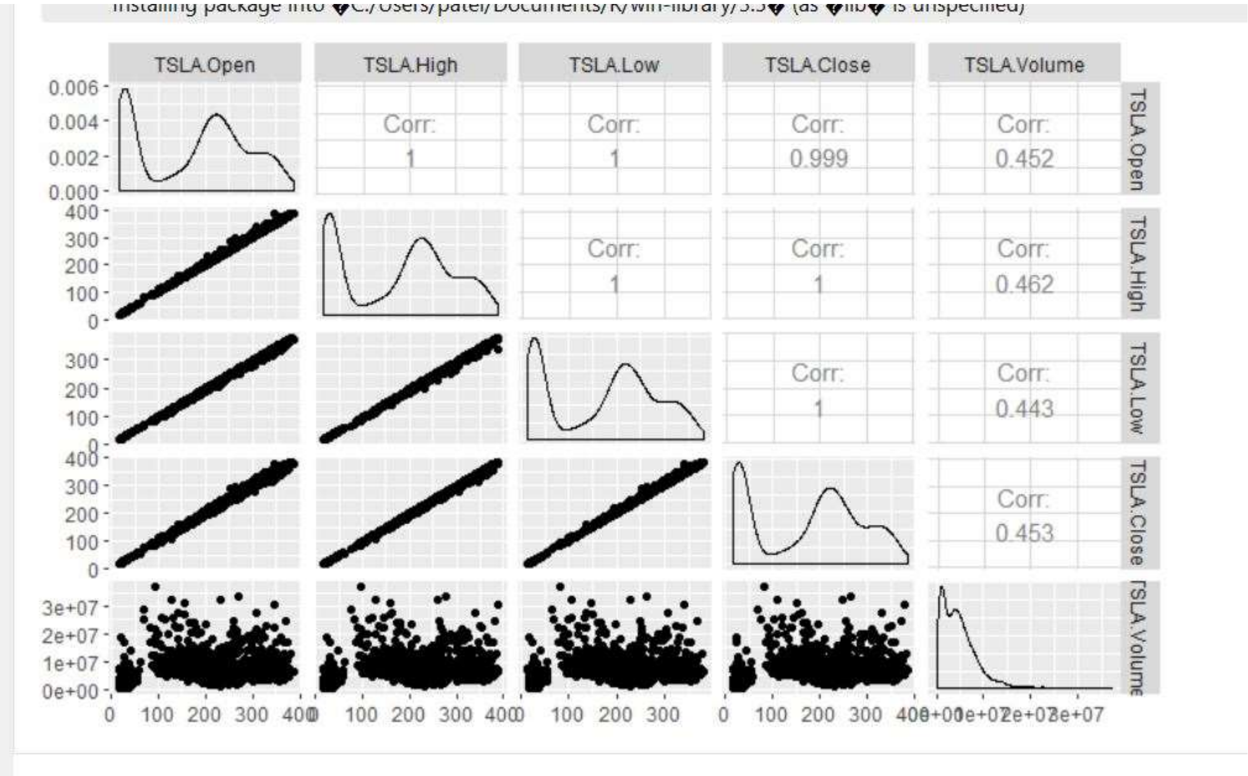
4.3 Correlation Plot:

The plot which describes how correlated 2 variables is known as correlation plot. This plot is plotted for all variables against all variables. We can generate such plots by the below given code.

Code:

```
library(GGally)

ggpairs(tsla[,1:5])
```

Plot:



installing package into �C:/Users/patel/Documents/R/win-library/3.5� (as �lib� is unspecified)

From the plot we can conclude that the correlations which tend tobe equal to 1 or are greater than 0.5 are highly correlated, it means that a change in one will cause similar kind of change in another.

5. Forecasting Methods

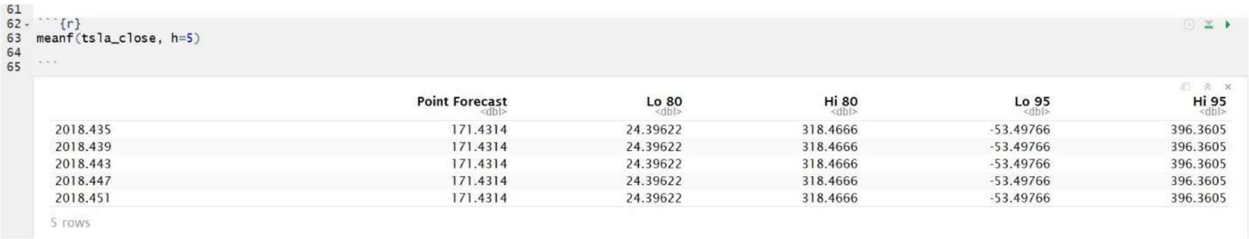
5.1 Average Method

In this method we forecast the future values of a variables using the average or mean of the historical data.

Code:

meanf(tsla_close, h=5)

Plot:



~~~
```{r}
meanf(tsla_close, h=5)

```
~~~

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
2018.435	171.4314	24.39622	318.4666	-53.49766	396.3605
2018.439	171.4314	24.39622	318.4666	-53.49766	396.3605
2018.443	171.4314	24.39622	318.4666	-53.49766	396.3605
2018.447	171.4314	24.39622	318.4666	-53.49766	396.3605
2018.451	171.4314	24.39622	318.4666	-53.49766	396.3605

5 rows

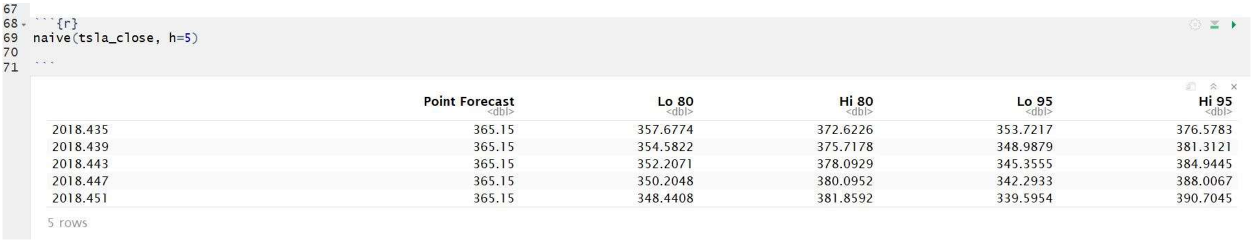
5.2 Naïve Method

In the naïve method we are simply using the last observed value or observation to forecast the future values.

Code:

naive(tsla_close, h=5)

Plot:



~~~
```{r}
naive(tsla_close, h=5)

```
~~~

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
2018.435	365.15	357.6774	372.6226	353.7217	376.5783
2018.439	365.15	354.5822	375.7178	348.9879	381.3121
2018.443	365.15	352.2071	378.0929	345.3555	384.9445
2018.447	365.15	350.2048	380.0952	342.2933	388.0067
2018.451	365.15	348.4408	381.8592	339.5954	390.7045

5 rows

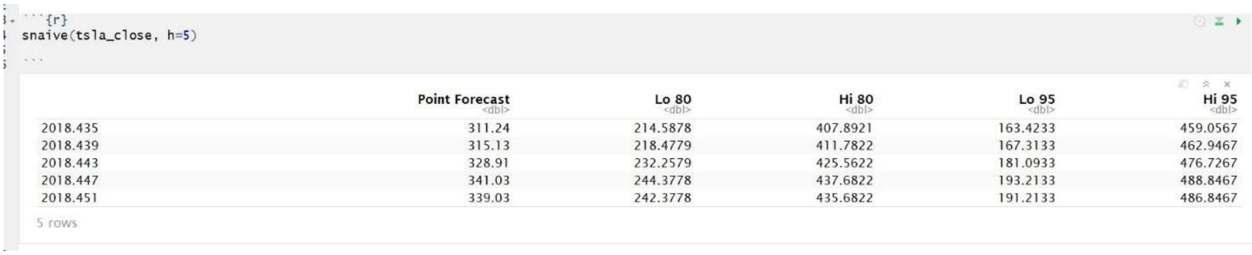
5.3 Seasonal Naïve Method

Similar to the naïve method, here we consider values from same season of last year to forecast the values.

Code:

```
snaive(tsla_close, h=5)
```

Plot:



~~~
```{r}
snaive(tsla_close, h=5)
```
~~~

	Point Forecast <dbl>	Lo 80 <dbl>	Hi 80 <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
2018.435	311.24	214.5878	407.8921	163.4233	459.0567
2018.439	315.13	218.4779	411.7822	167.3133	462.9467
2018.443	328.91	232.2579	425.5622	181.0933	476.7267
2018.447	341.03	244.3778	437.6822	193.2133	488.8467
2018.451	339.03	242.3778	435.6822	191.2133	486.8467

5 rows

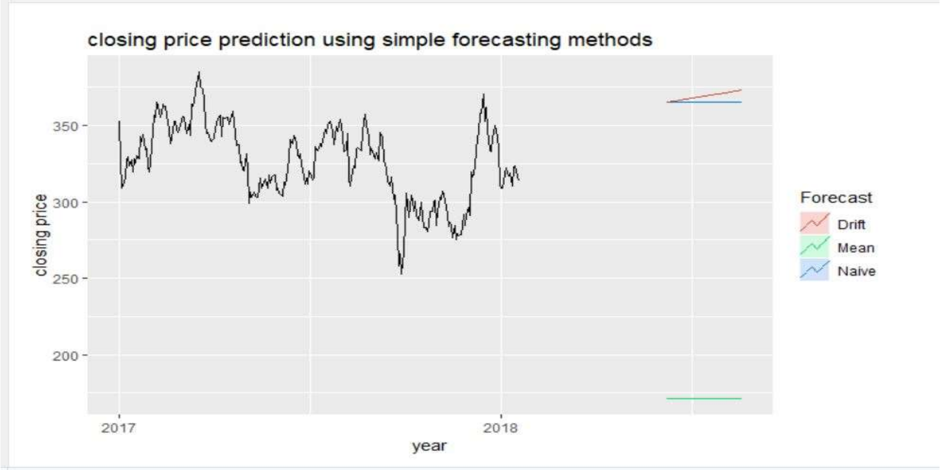
5.4 Plotting the above mentioned methods using a single graph

We can simply take the complete forecast of all the datapoints and then layer it with the three methods of forecasting.

Code:

```
abc = window(tsla_close, start=2017,end=c(2018,12))

autoplot(abc)+

  autolayer(meanf(tsla_close, h=50), series="Mean",PI=FALSE) +

  autolayer(rwf(tsla_close, h=50), series="Naive",PI=FALSE)+

  autolayer(rwf(tsla_close,drift = TRUE, h=50), series="Drift",PI=FALSE) +

  ggtitle("closing price prediction using simple forecasting methods") +

  xlab("year")+ ylab("closing price")+

  guides(colour=guide_legend(title="Forecast"))
```

Plot:



5.5 Simple Linear Regression

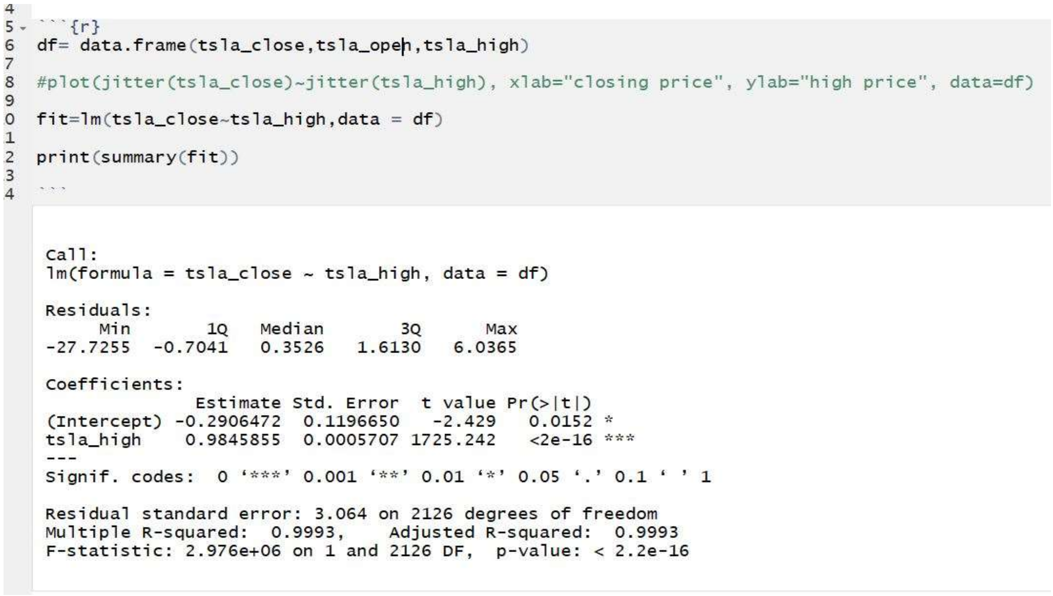
We use the linear regression model here to fit the data to the model and predict the value of the forecast.

Code:

df= data.frame(tsla_close,tsla_open,tsla_high)

fit=lm(tsla_close~tsla_high,data = df)

print(summary(fit))



~~~
```{r}
df= data.frame(tsla_close,tsla_open,tsla_high)

#plot(jitter(tsla_close)~jitter(tsla_high), xlab="closing price", ylab="high price", data=df)

fit=lm(tsla_close~tsla_high,data = df)

print(summary(fit))

```
~~~

```

Call:
lm(formula = tsla_close ~ tsla_high, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-27.7255  -0.7041   0.3526   1.6130   6.0365 

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -0.2906472  0.1196650   -2.429   0.0152 *  
tsla_high    0.9845855  0.0005707 1725.242   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.064 on 2126 degrees of freedom
Multiple R-squared:  0.9993,	Adjusted R-squared:  0.9993 
F-statistic: 2.976e+06 on 1 and 2126 DF,  p-value: < 2.2e-16
```

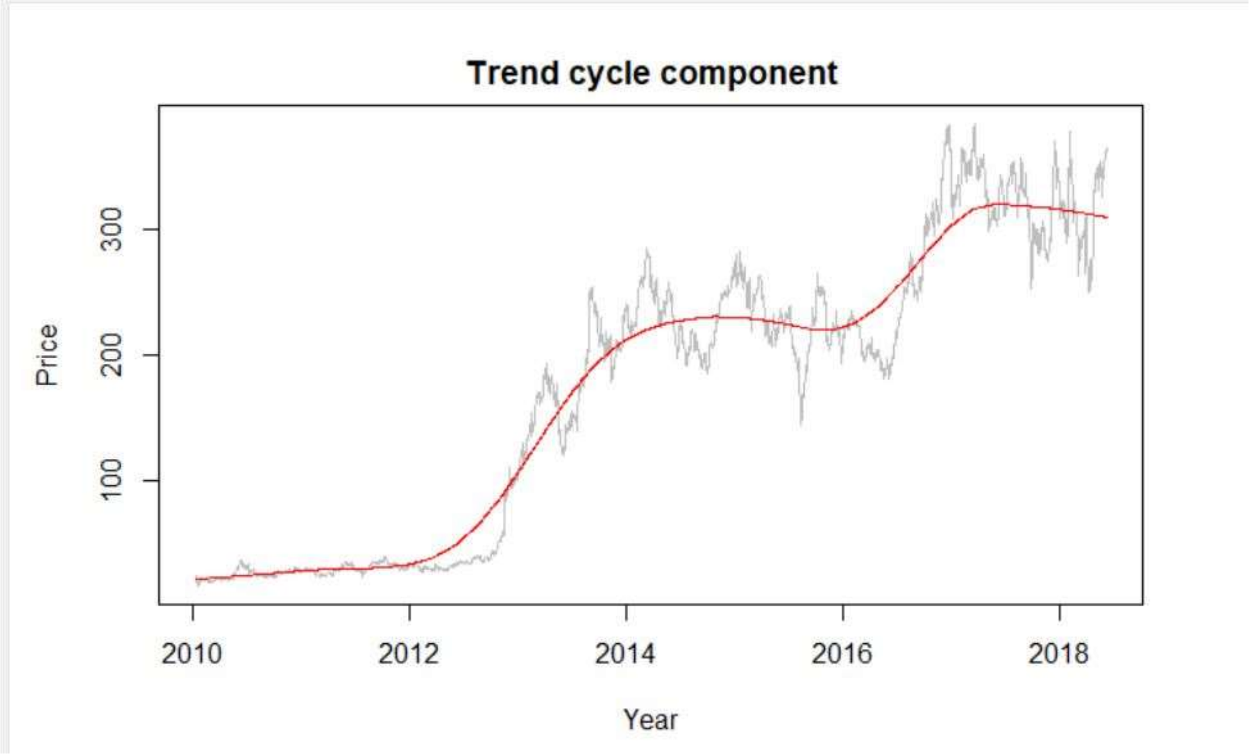
5.6 Trend Cycle Component

In here we try to plot the trend component by smoothening the original plot and we layer it using a red line.

Code:

```
fit1 = stl(tsla_close, s.window = 5)

plot(tsla_close, col="gray", main="Trend cycle component", ylab="Price", xlab="Year")

lines(fit1$time.series[,2], col="red", ylab="trend")
```

Plot:



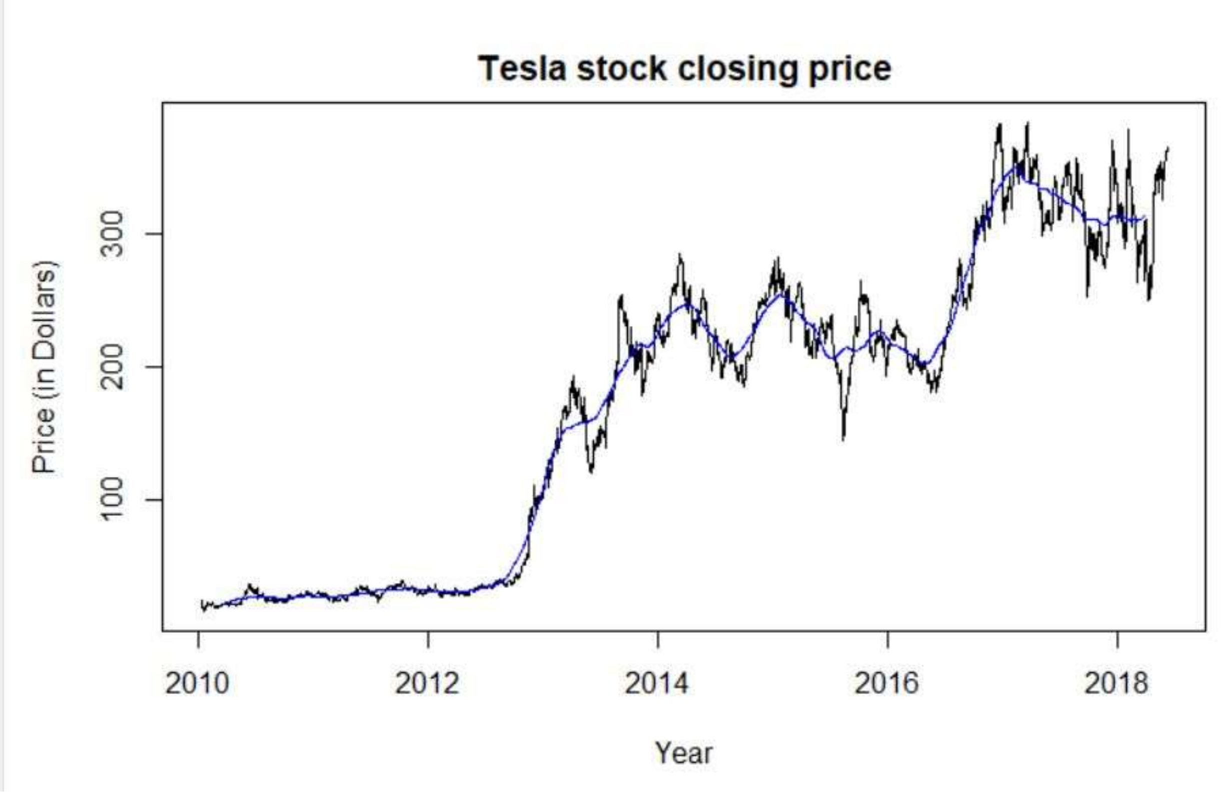
5.7 Moving Average

We can also use the moving average method to estimate all the existing values of the data points and that can be done using the below given code.

Code:

```
plot(tsla_close, main="Tesla stock closing price", ylab="Price (in Dollars)", xlab="Year")

lines(ma(tsla_close,100),col="blue")
```

Plot:



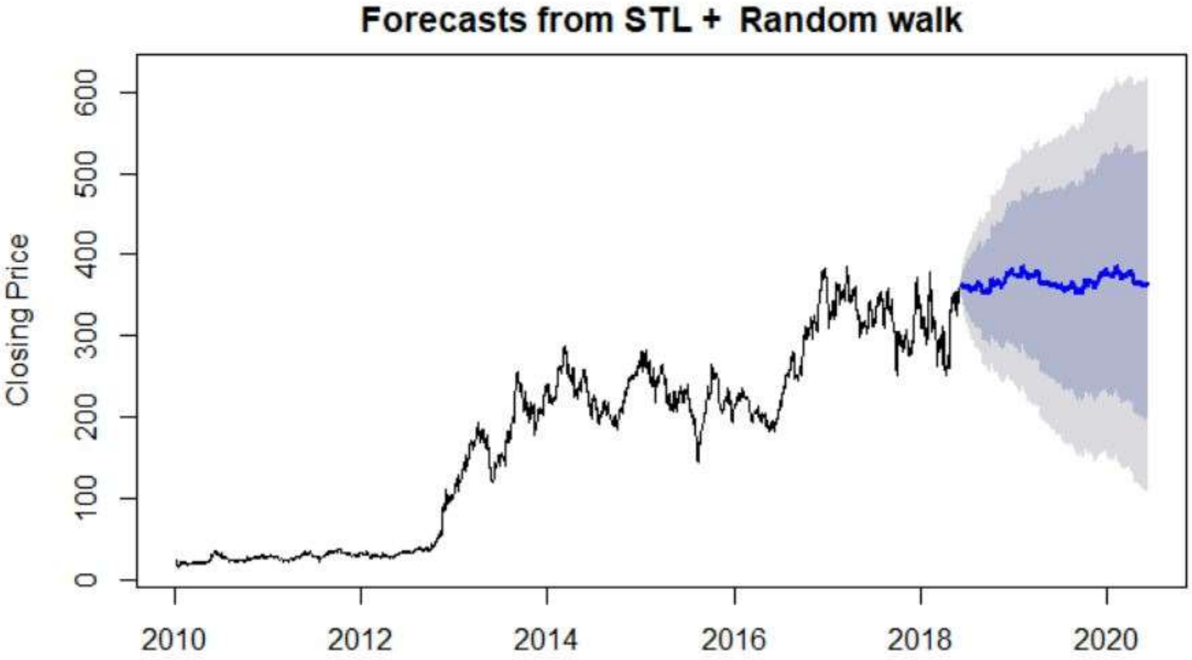
5.8 STL Decomposition

Forecasting the seasonal component and seasonally adjusted component separately. In other words, a seasonal naive method is used for the seasonal component after an STL decomposition of the data. Seasonal and Trend Decomposition using Loess

Code:

```
fcast = forecast(fit2, method = "naive")

plot(fcast, ylab="Closing Price")
```

Plot:



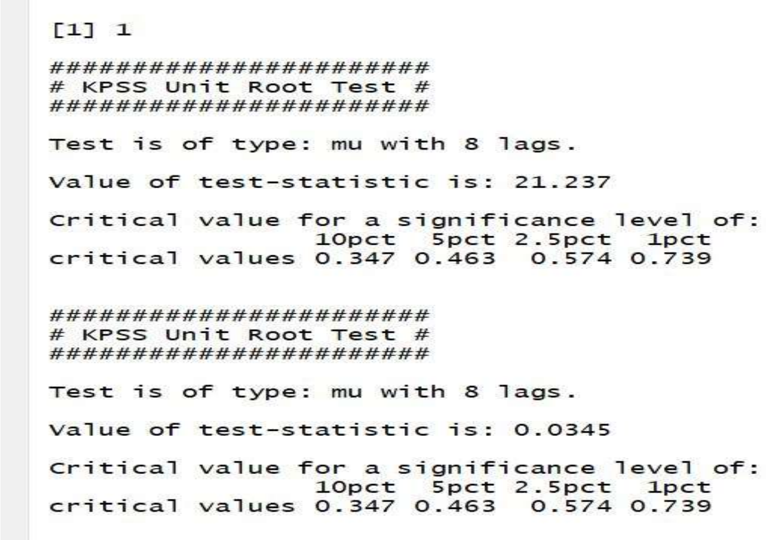
5.9 Stationarity Tests

Our null hypothesis is that the data is ststionary, and we look for evidence in form of p-values Whenever we have high or significant p-values, that means that our data needs differencing, On the controrary, after the first differencing, if we find out that the p-values becomes significantly small that means that we have proved the null hypothesis to be false and now our data is stationary. The test for differencing can be computed using the ur.kpss() function and that is available in the ‘urca’ package.

Code:

```
library(urca)

n = ndiffs(tsla_close)

print(n)

test=ur.kpss(tsla_close)

print(summary(test))

test2= ur.kpss(diff(tsla_close))

summary(test2)
```

Plot:



```
[1] 1

#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 8 lags.

Value of test-statistic is: 21.237

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739

#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 8 lags.

Value of test-statistic is: 0.0345

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

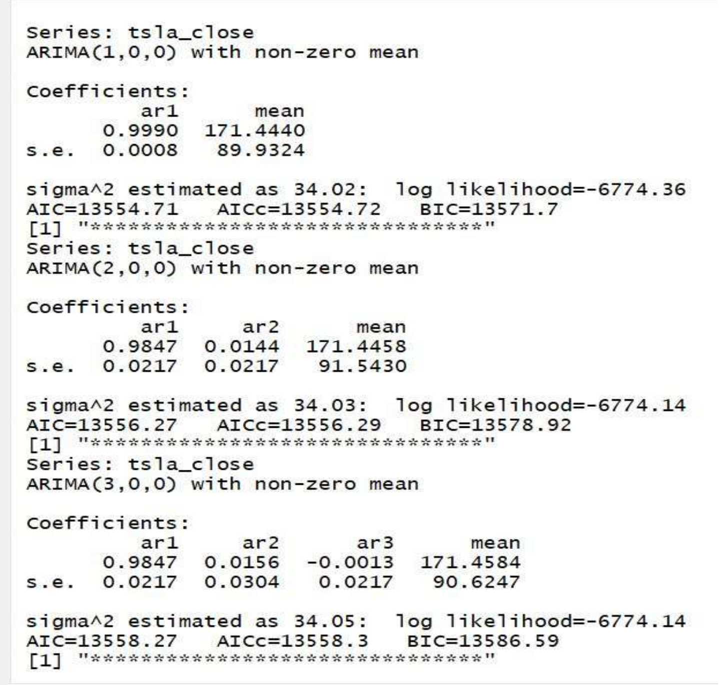
From the kpss unit root test we can see that our data needed differencing and that can be observed by high test-statistic. After the differencing and again putting the differenced dataset in the kpss unit root test we can observe that the test-statistic value has drastically fallen, hence we can conclude that our data is stationary.

5.10 AR(p) Model

In multiple regression model, we forecasted variable of interest using a linear combination of predictor variables. Auto-Regression is slightly different because in autoregression we forecast the variable using the linear combination of the past values of the variables. The name itself suggests that the regression of the forecast variable is against itself. AR(p) is called as autoregression model of order 'p' that means that the model uses p past values of the target variable.

Code:

```
Arima(tsla_close, order = c(1,0,0))

print("********************************")

Arima(tsla_close, order = c(2,0,0))

print("********************************")

Arima(tsla_close, order = c(3,0,0))

print("********************************")
```



```
Series: tsla_close
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1      mean
      0.9990  171.4440
s.e.  0.0008   89.9324

sigma^2 estimated as 34.02:  log likelihood=-6774.36
AIC=13554.71   AICc=13554.72   BIC=13571.7
[1] "********************************"
Series: tsla_close
ARIMA(2,0,0) with non-zero mean

Coefficients:
         ar1     ar2      mean
      0.9847  0.0144  171.4458
s.e.  0.0217  0.0217   91.5430

sigma^2 estimated as 34.03:  log likelihood=-6774.14
AIC=13556.27   AICc=13556.29   BIC=13578.92
[1] "********************************"
Series: tsla_close
ARIMA(3,0,0) with non-zero mean

Coefficients:
         ar1     ar2      ar3      mean
      0.9847  0.0156  -0.0013  171.4584
s.e.  0.0217  0.0304   0.0217   90.6247

sigma^2 estimated as 34.05:  log likelihood=-6774.14
AIC=13558.27   AICc=13558.3   BIC=13586.59
[1] "********************************"
```

5.11 MA(q) Model

The Moving Average Models. The first thing we need to keep in mind is that we should not confuse the moving average models with moving-average smoothing. Reason: The moving average model is used for forecasting future values while the moving-average smoothing is used to estimate the trend-cycle of the past values. In this model instead of using the past values to forecast values of the variable, we use the past forecast errors to forecast in a model similar to regression.

Code:

Arima(tsla_close, order = c(0,0,1))

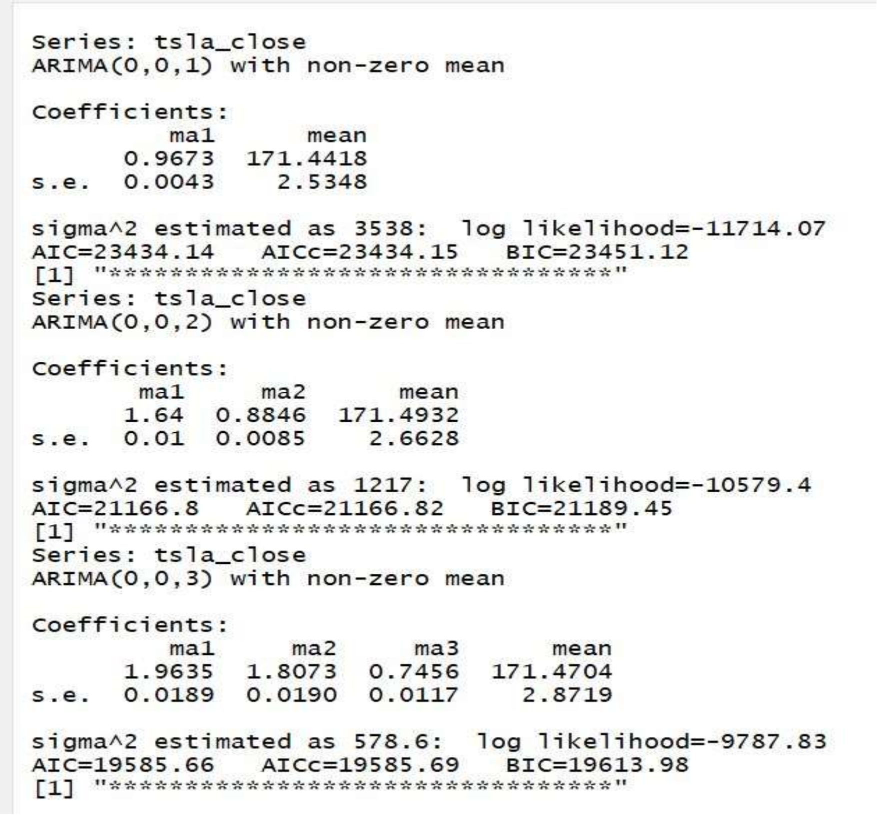
print("*********************************")

Arima(tsla_close, order = c(0,0,2))

print("*********************************")

Arima(tsla_close, order = c(0,0,3))

print("*********************************")



```
Series: tsla_close
ARIMA(0,0,1) with non-zero mean

Coefficients:
         ma1      mean
      0.9673  171.4418
s.e.  0.0043    2.5348

sigma^2 estimated as 3538:  log likelihood=-11714.07
AIC=23434.14   AICc=23434.15   BIC=23451.12
[1] "*********************************"
Series: tsla_close
ARIMA(0,0,2) with non-zero mean

Coefficients:
       ma1     ma2      mean
      1.64  0.8846  171.4932
s.e.  0.01  0.0085    2.6628

sigma^2 estimated as 1217:  log likelihood=-10579.4
AIC=21166.8   AICc=21166.82   BIC=21189.45
[1] "*********************************"
Series: tsla_close
ARIMA(0,0,3) with non-zero mean

Coefficients:
         ma1     ma2     ma3      mean
      1.9635  1.8073  0.7456  171.4704
s.e.  0.0189  0.0190  0.0117    2.8719

sigma^2 estimated as 578.6:  log likelihood=-9787.83
AIC=19585.66   AICc=19585.69   BIC=19613.98
[1] "*********************************"
```

6. CONCLUSION

- The data we took of the Tesla INC. Contained trend which we were able to realize.
- The data was also not stationary.
- Some forecast methods would be inappropriate to use here considering the amount data points we had in our dataset.
- Some forecast methods were good upto certain extent but they lack some or other condition that makes them miss certain aspects of the forecast.