Code

Apple Share Forecasting

Hide

```
#Importing packages dplyr and tidyr to assist data manipulation
library(dplyr)
library(tidyr)
library(zoo)

#Importing apple stock market values
apple <- read.csv('HistoricalData_1663834817479.csv')
apple <- as.data.frame(apple)
```

Apple Dataset:

Data Preview | Data Summary

Hide

```
head(apple)
```

	Date <chr>	Close.Last <chr>	Volume <int>	Open <chr>	High <chr>	Low <chr>
1	9/21/2022	$153.72	101696800	$157.34	$158.74	$153.60
2	9/20/2022	$156.90	107689800	$153.40	$158.08	$153.08
3	9/19/2022	$154.48	81474250	$149.31	$154.56	$149.10
4	9/16/2022	$150.70	162278800	$151.21	$151.35	$148.37
5	9/15/2022	$152.37	90481110	$154.65	$155.24	$151.38
6	9/14/2022	$155.31	87965410	$154.79	$157.10	$153.61

6 rows

Data contains dollar symbol in monetary values which should be removed to work with numerical values. The "Dates" column is also of class character and should be converted to a date type. Next step shows data cleaning process to remove symbol.

Hide

```
apple <- lapply(apple, sub, pattern = '\\$', replacement = '')
apple <- as.data.frame(apple)
apple$Date <- as.Date(apple$Date, "%m/%d/%Y")
apple$Date <- as.Date(apple$Date, "%Y-%m-%d")
head(apple)
```

	Date <date>	Close.Last <chr>	Volume <chr>	Open <chr>	High <chr>	Low <chr>
1	2022-09-21	153.72	101696800	157.34	158.74	153.60
2	2022-09-20	156.90	107689800	153.40	158.08	153.08
3	2022-09-19	154.48	81474250	149.31	154.56	149.10
4	2022-09-16	150.70	162278800	151.21	151.35	148.37
5	2022-09-15	152.37	90481110	154.65	155.24	151.38
6	2022-09-14	155.31	87965410	154.79	157.10	153.61

6 rows

Hide

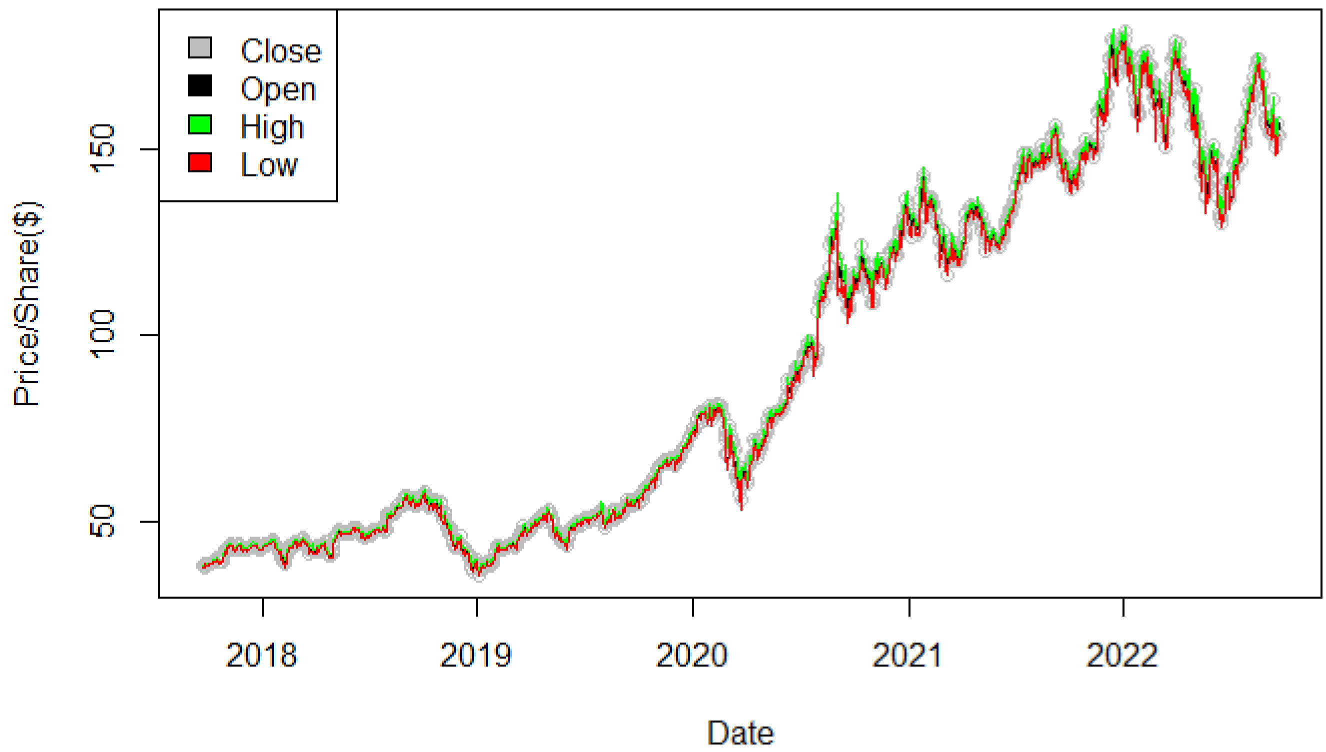
```
plot(apple$Date, apple$Close.Last,col = 'grey', xlab = 'Date', ylab = 'Price/Share($)')
lines(apple$Date, apple$Open, col = 'black')
```

Hide

```
lines(apple$Date, apple$High, col = 'green')
lines(apple$Date, apple$Low, col = 'red')
```

Hide

```
legend(x = 'topleft', y=150, legend = c('Close', 'Open', 'High', 'Low'), fill = c('grey',
'black', 'green', 'red'))
```



The stock market has a lot of volatility in it, so it may be beneficial to exponentially smooth the data set to ensure there are less random occurrences being analyzed. To accomplish this, the HoltWinters function will be used. First, the data must be converted to time series data and its components will be examined.

Hide

```
#Number of trading days in a year
Trading_Days <- 252

#Convert columns from type char to numeric
for(i in 2:length(names(apple))){
  apple[, i] <- as.numeric(apple[, i])
}

#Order data set by date
apple_ordered <- zoo(apple[,-1], order.by = apple[,1])
head(apple_ordered)
```

```
           Close.Last    Volume  Open  High   Low
2017-09-22      37.97 186301640 38.01 38.07 37.64
2017-09-25      37.64 177464560 37.50 37.96 37.29
2017-09-26      38.29 143631080 37.95 38.48 37.92
2017-09-27      38.56 101609080 38.45 38.68 38.39
2017-09-28      38.32  87933640 38.47 38.57 38.18
2017-09-29      38.53 104818680 38.30 38.53 38.00
```

Hide

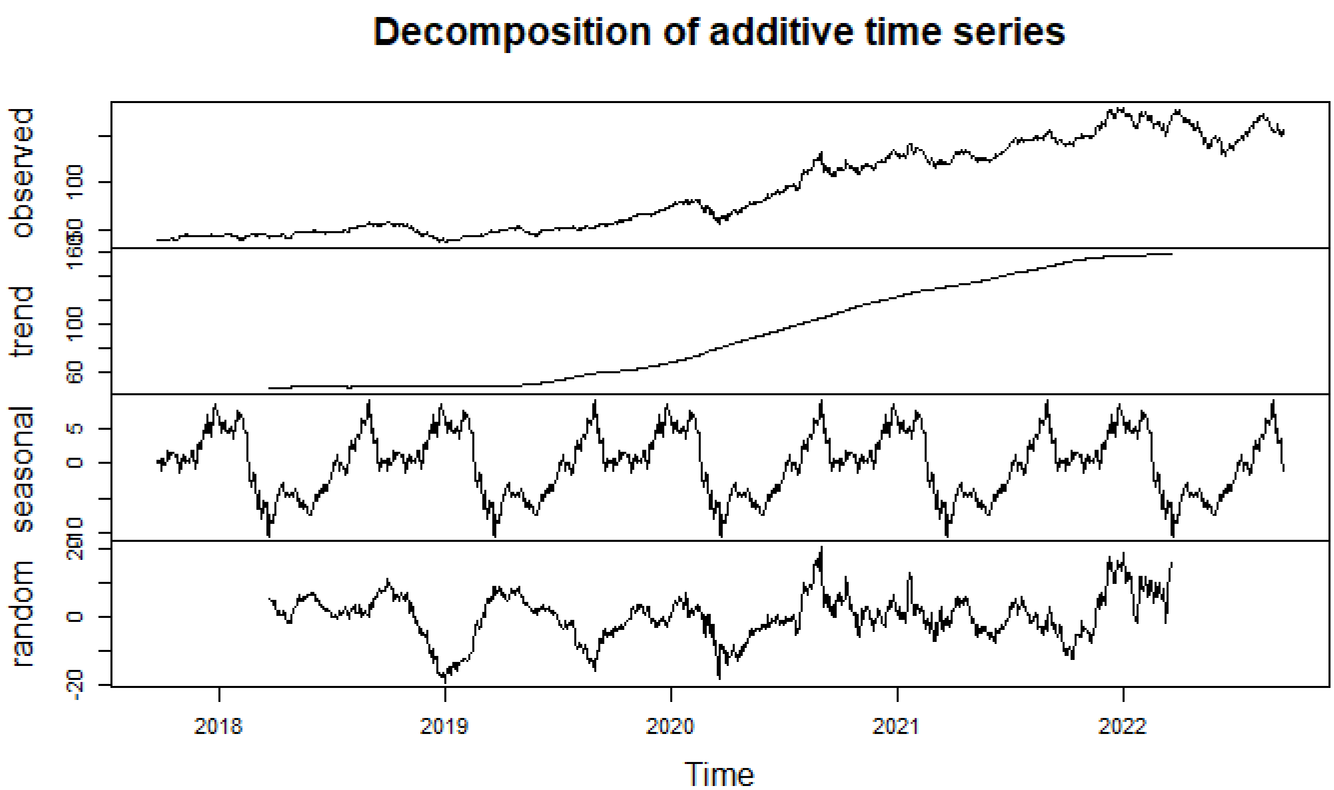
```
#Converting DF to time series data and starting on 184/252 trading day in 2017
apple_ts <- ts(apple_ordered, frequency = Trading_Days, start = c(2017, 184))

components_apple_close <- decompose(apple_ts[, 1])
components_apple_volume <- decompose(apple_ts[, 2])
components_apple_open <- decompose(apple_ts[, 3])
components_apple_high <- decompose(apple_ts[, 4])
components_apple_low <- decompose(apple_ts[, 5])
```

Apple Stock Price & Volume Changes Over Time:

Close Price | Trade Volume | Open Price | High Price | Low Price

Hide

```
plot(components_apple_close)
```



It is evident that there is randomness present in the data, but it is significantly lower in magnitude when compared to the trend and observed data. Therefore, a valid forecast is possible. To accomplish this the data will first be exponentially smoothed using the HoltWinters method.

Hide

```
#Applying HoltWinters function for exponential smoothing
apple_close_smoothed <- HoltWinters(apple_ts[,1], seasonal = 'multiplicative')
apple_volume_smoothed <- HoltWinters(apple_ts[,2], seasonal = 'multiplicative')
apple_open_smoothed <- HoltWinters(apple_ts[,3], seasonal = 'multiplicative')
apple_high_smoothed <- HoltWinters(apple_ts[,4], seasonal = 'multiplicative')
apple_low_smoothed <- HoltWinters(apple_ts[,5], seasonal = 'multiplicative')
```

Visual Inspection of Fits:

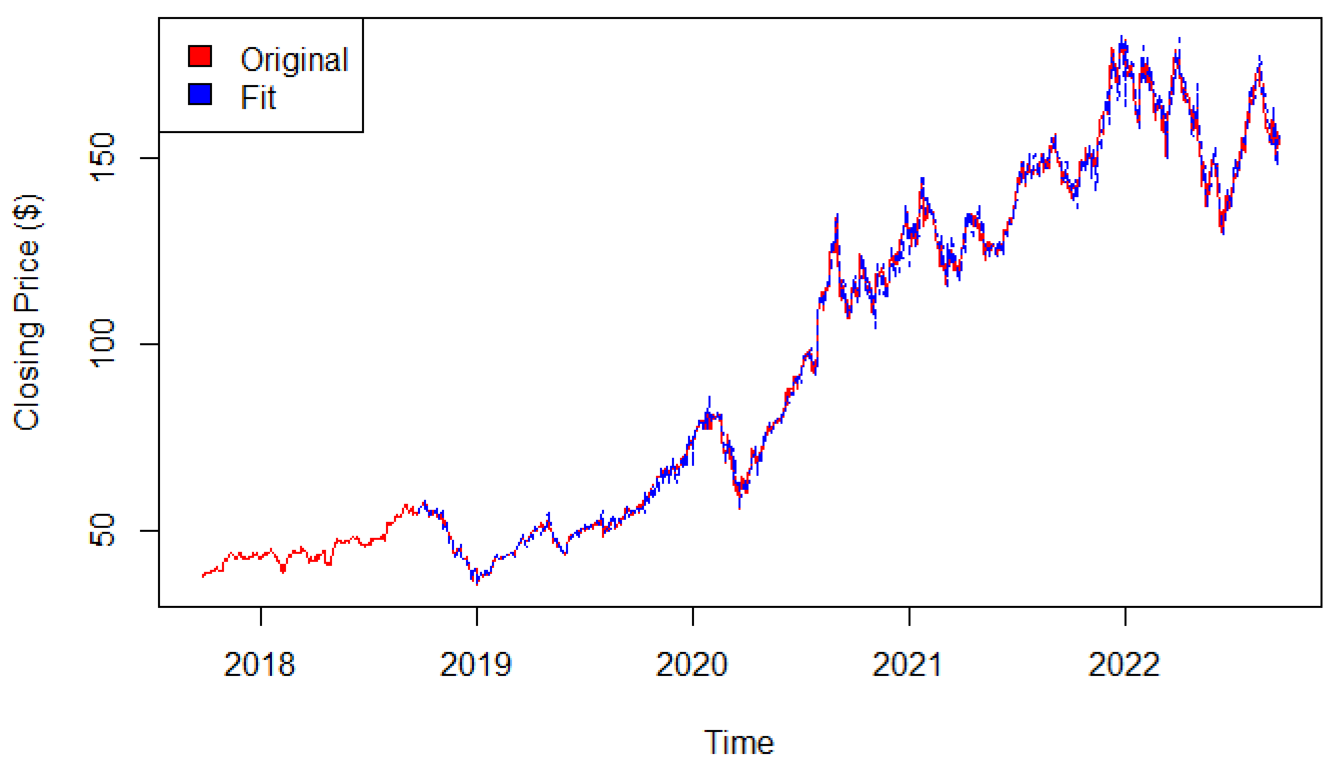
Close Price Fit | Trade Volume Fit | Open Price Fit | High Price Fit | Low Price Fit

Hide

```
#Visually evaluate the fits
plot(apple_ts[,1], ylab = 'Closing Price ($)', col = 'red')
lines(apple_close_smoothed$fitted[,1], lty = 2, col = 'blue')
```

Hide

```
legend(x = 'topleft', y=150, legend = c('Original', 'Fit'), fill = c('red', 'blue'))
```



Each fit appears to match well with the original data set. Now, this fit data can be used to forecast the share prices and trade volume over the next year. It will also be beneficial to include confidence intervals along with the predicted data.

Hide

```
#Forecasting next 252 trading days
apple_close_forecast <- predict(apple_close_smoothed, 252, prediction.interval = TRUE, le
vel=0.95)
apple_volume_forecast <- predict(apple_volume_smoothed, 252, prediction.interval = TRUE,
 level=0.95)
apple_open_forecast <- predict(apple_open_smoothed, 252, prediction.interval = TRUE, leve
l=0.95)
apple_high_forecast <- predict(apple_high_smoothed, 252, prediction.interval = TRUE, leve
l=0.95)
apple_low_forecast <- predict(apple_low_smoothed, 252, prediction.interval = TRUE, level=
0.95)
```

We can take a look at the predicted data over the next 252 trading days below.

Forecasts:

Close Price | Trade Volume | Open Price | High Price | Low Price

Hide

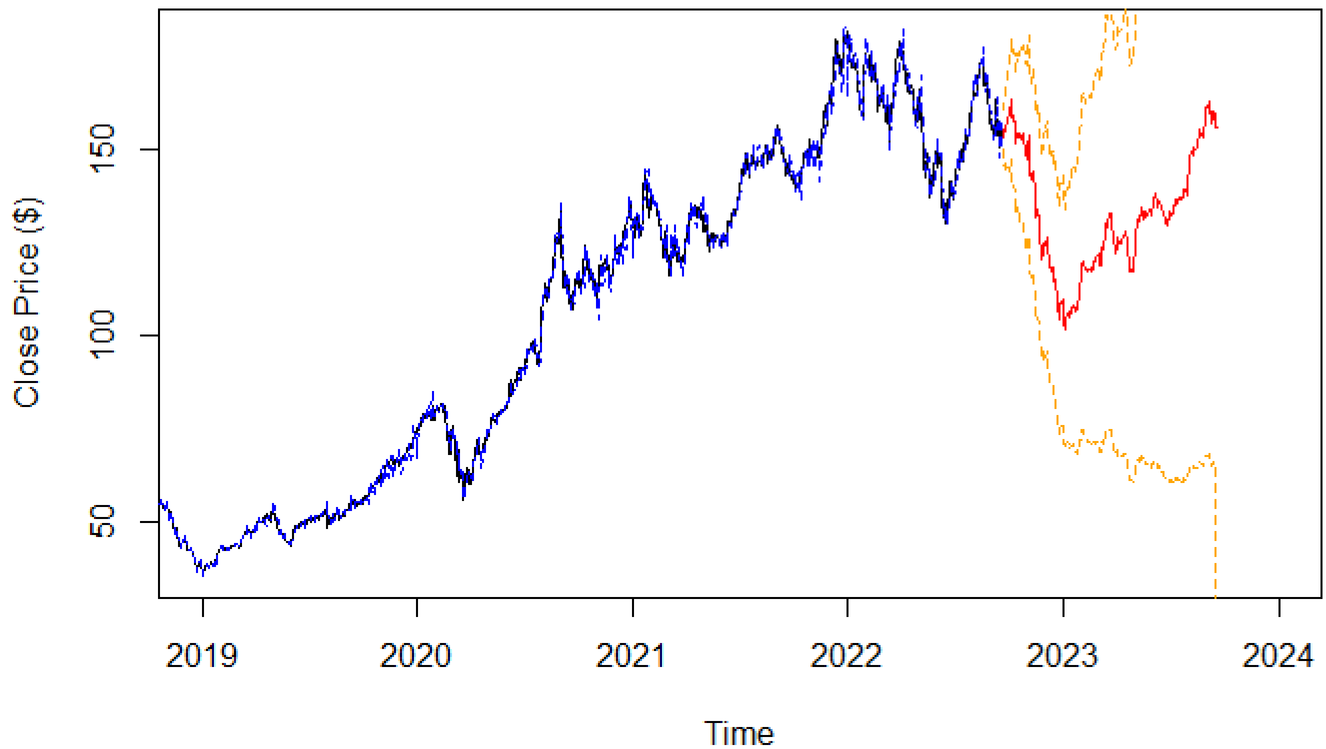
```
#Visually evaluate the prediction
plot(apple_ts[,1], ylab="Close Price ($)", xlim = c(2019, 2024))
lines(apple_close_smoothed$fitted[,1], lty=2, col="blue")
```

Hide

```
lines(apple_close_forecast[,1], col="red")
lines(apple_close_forecast[,2], lty=2, col="orange")
```

Hide

```
lines(apple_close_forecast[,3], lty=2, col="orange")
```



Looking at our predicted data can help give us insight into when may be a good time to invest our money into apple shares. Of course, this report is done for demonstration purposes and using such methods for predictions on share price is just about as good as gambling. Disregarding the completely unpredictable nature of the stock market, we can look at the minimum values of our data forecasts to get an idea for what might be a good value for a limit buy and a limit buy for our apple shares in the future.

Hide

```
#Determining lowest share prices for Close, Open, High, and Low in forecasts
low_buys <- c()
low_buys[1] <- min(apple_close_forecast[,1])
low_buys[2] <- min(apple_open_forecast[,1])
low_buys[3] <- min(apple_high_forecast[,1])
low_buys[4] <- min(apple_low_forecast[,1])
mean(low_buys)
```

```
[1] 102.5442
```

Above we can see that our lowest price forecasts can be averaged together to give us 102.54 dollars per share. It can also be seen that the lowest share price is occurring just shortly after the beginning of 2023. Therefore, we can set a conservative limit buy of about 110 dollars one week into December 2022.

Should we obtain our shares at the limit price we set, we can now go ahead and decide if we want to hold long term or turn around and set a limit sell for a price of our choice. A safe choice may be something like 140 dollars per share and holding our shares until we are able to sell them at this price.

Once again, this report is utilized to demonstrate the capabilities of exponential smoothing and forecasting future data based on past trends and randomness. Nothing in this report should be taken as financial advice.