Time Series Modelling

Kaustubh Saraf

4/19/2020

Loading packages

library("forecast")  
library(tidyverse)  
library(vctrs)  
library(lubridate)  
library("tseries")  
library(TSstudio)

Import data

amazon <- read.csv("C:/Users/kaustubh\_14/Documents/GitHub/Time-Series/AMZN.csv")

amazon$month <- as.factor(month.abb[(month(amazon$Date))])  
  
levels(amazon$Months)

## NULL

amazon$year <- as.factor(year(amazon$Date))  
  
amazon <- amazon %>%   
 mutate(Months = fct\_relevel(month,"Jan",  
 "Feb",  
 "Mar",  
 "Apr",  
 "May",  
 "Jun",  
 "Jul",  
 "Aug",  
 "Sep",  
 "Oct",  
 "Nov",  
 "Dec"))

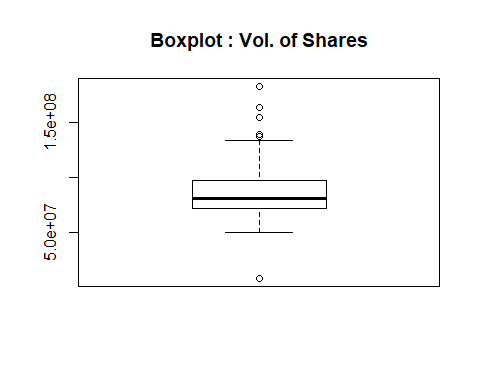
Summary statistics

summary(amazon)

## Date Open High Low   
## 2015-05-01: 1 Min. : 423.8 Min. : 439 Min. : 414.6   
## 2015-06-01: 1 1st Qu.: 752.4 1st Qu.: 775 1st Qu.: 716.5   
## 2015-07-01: 1 Median :1105.4 Median :1195 Median :1086.9   
## 2015-08-01: 1 Mean :1227.9 Mean :1315 Mean :1157.6   
## 2015-09-01: 1 3rd Qu.:1769.5 3rd Qu.:1854 3rd Qu.:1672.0   
## 2015-10-01: 1 Max. :2372.3 Max. :2461 Max. :2316.3   
## (Other) :55   
## Close Adj.Close Volume month year   
## Min. : 429.2 Min. : 429.2 Min. : 7930010 Apr : 6 2015: 8   
## 1st Qu.: 758.8 1st Qu.: 758.8 1st Qu.: 71748300 Aug : 5 2016:12   
## Median :1169.5 Median :1169.5 Median : 80936900 Dec : 5 2017:12   
## Mean :1258.0 Mean :1258.0 Mean : 88281005 Feb : 5 2018:12   
## 3rd Qu.:1776.3 3rd Qu.:1776.3 3rd Qu.: 97126100 Jan : 5 2019:12   
## Max. :2375.0 Max. :2375.0 Max. :183220800 Jul : 5 2020: 5   
## (Other):30   
## Months   
## Apr : 6   
## Jan : 5   
## Feb : 5   
## Mar : 5   
## May : 5   
## Jun : 5   
## (Other):30

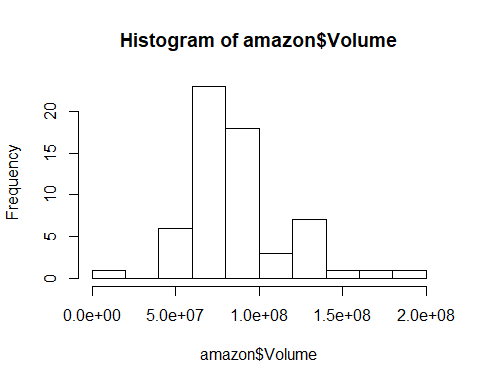
Checking the range of Volume

boxplot(amazon$Volume,main = "Boxplot : Vol. of Shares")



Checking the distribution of Volume

hist(amazon$Volume)



The Series is the volume of shares sold

amazon.vol <- amazon %>%   
 filter(year!=2020 & year!=2015) %>%   
 select (Volume)   
  
amazon.vol.ts<-ts(amazon.vol,frequency = 12, start = c(2016,1))  
  
amazon.vol.ts

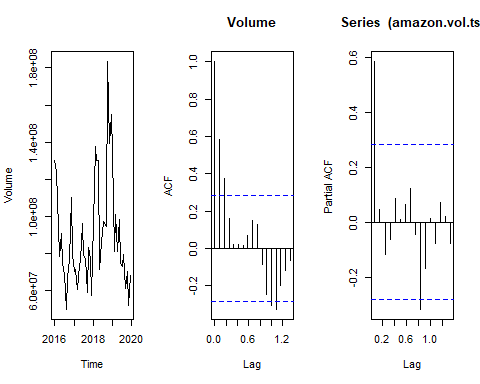
## Jan Feb Mar Apr May Jun Jul  
## 2016 130200900 124144800 94009500 78464200 90614500 74540900 68635500  
## 2017 70614000 71748300 60710700 73539700 76202000 96135400 78812400  
## 2018 96371200 137784000 130400100 129919600 71615500 85941300 97521100  
## 2019 134001700 80936900 100832200 81293700 98214400 74742700 73177400  
## Aug Sep Oct Nov Dec  
## 2016 50000400 67335700 77063800 110085900 79308600  
## 2017 77391800 59291800 83334100 77165000 57760200  
## 2018 96575800 94445500 183220800 139290000 154812700  
## 2019 79771200 61172900 70360500 52076200 68149600

Splitting the dataset

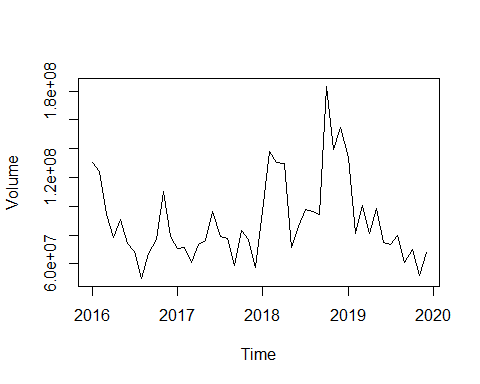
trainData = window(amazon.vol.ts, start=2016, end=c(2018,12))  
  
testData = window(amazon.vol.ts, start=2019, end=c(2019,12))

# Vol. forecasting

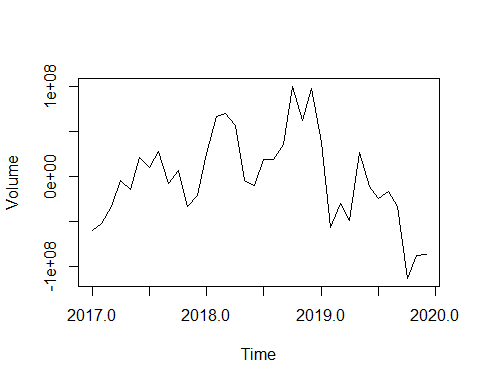
par(mfrow=c(1,3))  
  
plot((amazon.vol.ts))  
acf((amazon.vol.ts))  
pacf((amazon.vol.ts))



plot.ts(amazon.vol.ts)



diff1<-diff(amazon.vol.ts, lag=12)  
plot.ts(diff1)



* The plots suggests that the series is stationary
* The PACF plot suggests that the series might be generated from MA1
* The ACF plot suggests that the series might be generated from AR1 or AR2

\*\* Lets start with checking if the series is stationary \*\* # ADF Test

adf.test(amazon.vol.ts,k=0)

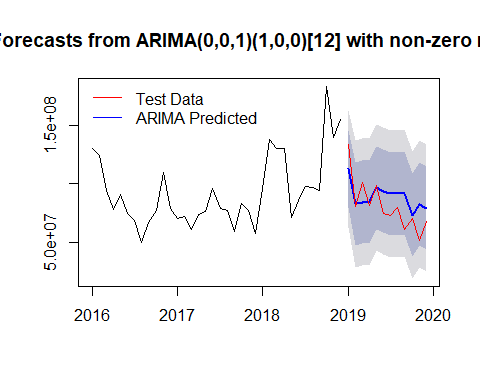
##   
## Augmented Dickey-Fuller Test  
##   
## data: amazon.vol.ts  
## Dickey-Fuller = -3.4874, Lag order = 0, p-value = 0.05379  
## alternative hypothesis: stationary

The ADF suggests that the series is stationary and we do not need to take difference

# Model Fitting

Model1 MA1 Season=1,0,0

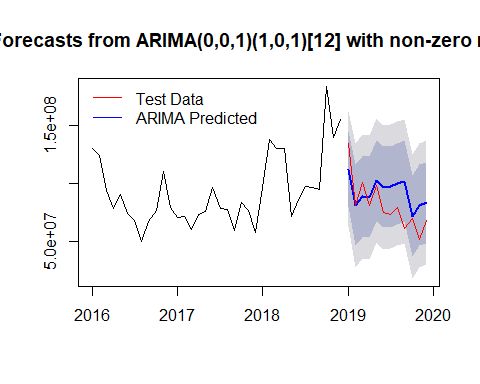
ma1.100 <- arima(trainData, order=c(0,0,1), season=list(order=c(1,0,0), period=12))  
ma1.100.fc <-forecast(ma1.100,h=12)  
  
plot(ma1.100.fc)  
lines(testData, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("Test Data","ARIMA Predicted"))



forecasted.val11 <-forecast(ma1.100.fc,h=12)  
  
model1.accuracy11 <- accuracy(forecasted.val11,testData)  
  
model1.accuracy11 <- data.frame(model1.accuracy11)  
  
model1.train.MAPE11 <- model1.accuracy11[1, 'MAPE']  
model1.test.MAPE11 <- model1.accuracy11[2, 'MAPE']

Model1 MA1 Season=1,0,1

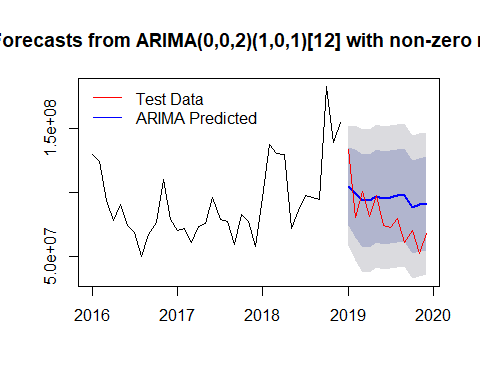
ma1.101 <- arima(trainData, order=c(0,0,1), season=list(order=c(1,0,1), period=12))  
ma1.101.fc <-forecast(ma1.101,h=12)  
  
plot(ma1.101.fc)  
lines(testData, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("Test Data","ARIMA Predicted"))



forecasted.val <-forecast(ma1.101.fc,h=12)  
  
model1.accuracy <- accuracy(forecasted.val,testData)  
  
model1.accuracy <- data.frame(model1.accuracy)  
  
model1.train.MAPE <- model1.accuracy[1, 'MAPE']  
model1.test.MAPE <- model1.accuracy[2, 'MAPE']

Model1 MA2 Season=1,0,1

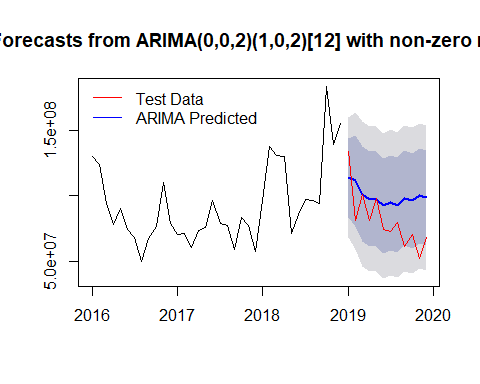
ma2.101 <- arima(trainData, order=c(0,0,2), season=list(order=c(1,0,1), period=12))  
ma2.101.fc <-forecast(ma2.101,h=12)  
  
plot(ma2.101.fc)  
lines(testData, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("Test Data","ARIMA Predicted"))



forecasted.val1 <-forecast(ma2.101.fc,h=12)  
  
model1.accuracy1 <- accuracy(forecasted.val1,testData)  
  
model1.accuracy1 <- data.frame(model1.accuracy1)  
  
model1.train.MAPE1 <- model1.accuracy1[1, 'MAPE']  
model1.test.MAPE1 <- model1.accuracy1[2, 'MAPE']

Model1 MA2 Season=1,0,2

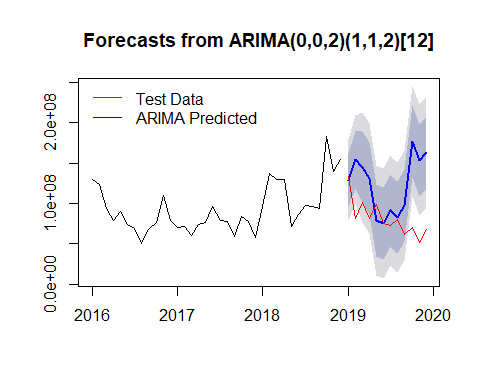
ma2.102 <- arima(trainData, order=c(0,0,2), season=list(order=c(1,0,2), period=12))  
ma2.102.fc <-forecast(ma2.102,h=12)  
  
plot(ma2.102.fc)  
lines(testData, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("Test Data","ARIMA Predicted"))



forecasted.val2 <-forecast(ma2.102.fc,h=12)  
  
model1.accuracy2 <- accuracy(forecasted.val2,testData)  
  
model1.accuracy2 <- data.frame(model1.accuracy2)  
  
model1.train.MAPE2 <- model1.accuracy2[1, 'MAPE']  
model1.test.MAPE2 <- model1.accuracy2[2, 'MAPE']

Model1 MA2 Season=1,1,2

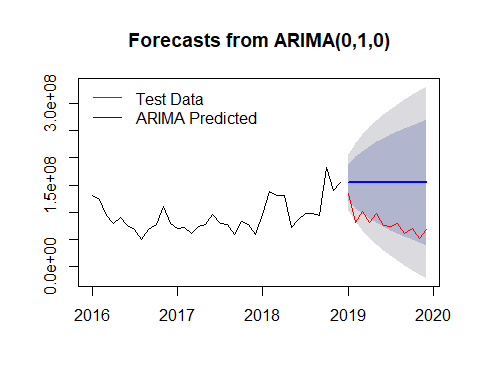
ma2.112 <- arima(trainData, order=c(0,0,2), season=list(order=c(1,1,2), period=12))  
ma2.112.fc <-forecast(ma2.112,h=12)  
  
plot(ma2.112.fc)  
lines(testData, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("Test Data","ARIMA Predicted"))



forecasted.val3 <-forecast(ma2.112.fc,h=12)  
  
model1.accuracy3 <- accuracy(forecasted.val3,testData)  
  
model1.accuracy3 <- data.frame(model1.accuracy3)  
  
model1.train.MAPE3 <- model1.accuracy3[1, 'MAPE']  
model1.test.MAPE3 <- model1.accuracy3[2, 'MAPE']

AutoArima

auto.arima.model <- auto.arima(trainData)  
auto.arima.model.fc <-forecast(auto.arima.model,h=12)  
  
plot(auto.arima.model.fc)  
lines(testData, col="red")  
legend("topleft",lty=1,bty = "n",col=c("red","blue"),c("Test Data","ARIMA Predicted"))



forecasted.val4 <-forecast(auto.arima.model,h=12)  
  
model1.accuracy4 <- accuracy(forecasted.val4,testData)  
  
model1.accuracy4 <- data.frame(model1.accuracy4)  
  
model1.train.MAPE4 <- model1.accuracy4[1, 'MAPE']  
model1.test.MAPE4 <- model1.accuracy4[2, 'MAPE']

AutoARIMA suggests that the model is generated from AR1 series which was our initial understanding.

Let’s compare the different AIC values

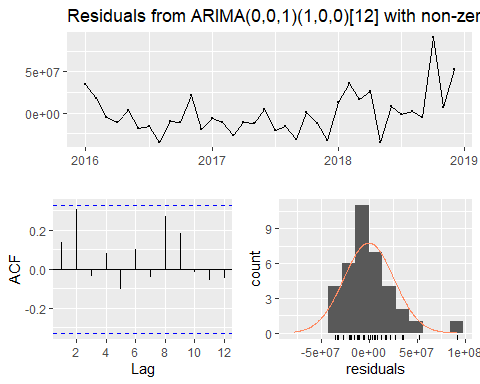
train.mape <- c(model1.train.MAPE11, model1.train.MAPE, model1.train.MAPE1,   
 model1.train.MAPE2, model1.train.MAPE3,model1.train.MAPE4)  
test.mape <- c(model1.test.MAPE11, model1.test.MAPE, model1.test.MAPE1,   
 model1.test.MAPE2, model1.test.MAPE3,model1.test.MAPE4)  
  
mape <- data.frame(train.mape, test.mape)  
  
rownames(mape) <- c("ma1.100","ma1.101", "ma2.101", "ma2.102", "ma2.112","Auto Arima")  
  
mape

## train.mape test.mape  
## ma1.100 21.174764 19.56398  
## ma1.101 21.085901 22.98210  
## ma2.101 19.145619 28.76997  
## ma2.102 17.979521 31.46609  
## ma2.112 9.588964 66.07467  
## Auto Arima 20.076593 101.55918

The AIC metric suggests that MA1 with seasonality AR=1 has smallest out of sample MAPE hence we go ahead with this model.

Residual Diagonastics of the Best Model Let’s try the auto-arima function and check the output

checkresiduals(ma1.100)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,0,1)(1,0,0)[12] with non-zero mean  
## Q\* = 5.8527, df = 4, p-value = 0.2104  
##   
## Model df: 3. Total lags used: 7

# VAR

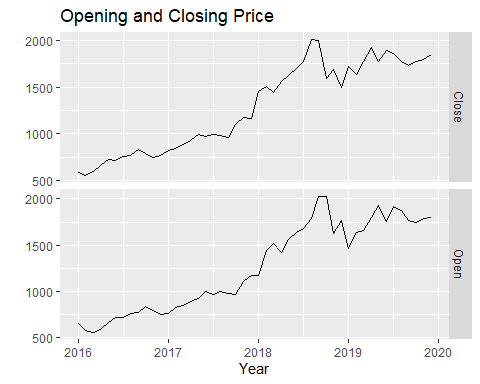
Data Prep

amazon.vol.close <- amazon %>%   
 filter(year!=2020 & year!=2015) %>%   
 transmute(Close,Open)  
   
   
  
amazon.vol.close.ts<-ts(amazon.vol.close,frequency = 12, start = c(2016,1))  
  
amazon.vol.close.ts

## Close Open  
## Jan 2016 587.00 656.29  
## Feb 2016 552.52 578.15  
## Mar 2016 593.64 556.29  
## Apr 2016 659.59 590.49  
## May 2016 722.79 663.92  
## Jun 2016 715.62 720.90  
## Jul 2016 758.81 717.32  
## Aug 2016 769.16 759.87  
## Sep 2016 837.31 770.90  
## Oct 2016 789.82 836.00  
## Nov 2016 750.57 799.00  
## Dec 2016 768.66 752.41  
## Jan 2017 823.48 757.92  
## Feb 2017 845.04 829.21  
## Mar 2017 886.54 853.05  
## Apr 2017 924.99 888.00  
## May 2017 994.62 927.80  
## Jun 2017 968.00 998.59  
## Jul 2017 987.78 972.79  
## Aug 2017 980.60 996.11  
## Sep 2017 961.35 984.20  
## Oct 2017 1105.28 964.00  
## Nov 2017 1176.75 1105.40  
## Dec 2017 1169.47 1172.05  
## Jan 2018 1450.89 1172.00  
## Feb 2018 1512.45 1445.00  
## Mar 2018 1447.34 1513.60  
## Apr 2018 1566.13 1417.62  
## May 2018 1629.62 1563.22  
## Jun 2018 1699.80 1637.03  
## Jul 2018 1777.44 1682.70  
## Aug 2018 2012.71 1784.00  
## Sep 2018 2003.00 2026.50  
## Oct 2018 1598.01 2021.99  
## Nov 2018 1690.17 1623.53  
## Dec 2018 1501.97 1769.46  
## Jan 2019 1718.73 1465.20  
## Feb 2019 1639.83 1638.88  
## Mar 2019 1780.75 1655.13  
## Apr 2019 1926.52 1800.11  
## May 2019 1775.07 1933.09  
## Jun 2019 1893.63 1760.01  
## Jul 2019 1866.78 1922.98  
## Aug 2019 1776.29 1871.72  
## Sep 2019 1735.91 1770.00  
## Oct 2019 1776.66 1746.00  
## Nov 2019 1800.80 1788.01  
## Dec 2019 1847.84 1804.40

Plot

autoplot(amazon.vol.close.ts[,1:2], facets=TRUE) +  
 xlab("Year") + ylab("") +  
 ggtitle("Opening and Closing Price")



lm.fit <- lm(Close~Open,data=amazon.vol.close.ts)  
summary(lm.fit)

##   
## Call:  
## lm(formula = Close ~ Open, data = amazon.vol.close.ts)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -418.83 -49.82 -2.97 51.92 252.69   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 69.42273 47.65039 1.457 0.152   
## Open 0.96312 0.03532 27.272 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 115.6 on 46 degrees of freedom  
## Multiple R-squared: 0.9418, Adjusted R-squared: 0.9405   
## F-statistic: 743.7 on 1 and 46 DF, p-value: < 2.2e-16

(fit <- auto.arima(amazon.vol.close.ts[,"Close"], xreg=amazon.vol.close.ts[,"Open"]))

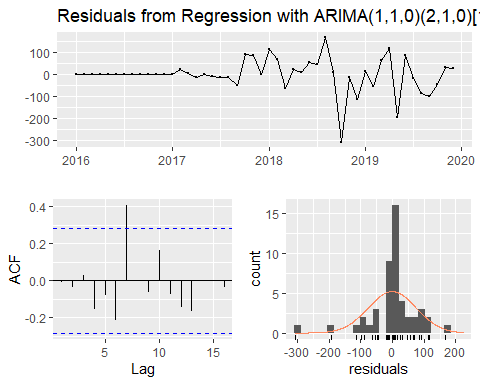
## Series: amazon.vol.close.ts[, "Close"]   
## Regression with ARIMA(0,0,0) errors   
##   
## Coefficients:  
## xreg  
## 1.0113  
## s.e. 0.0124  
##   
## sigma^2 estimated as 13689: log likelihood=-296.19  
## AIC=596.38 AICc=596.64 BIC=600.12

(fit <- Arima(amazon.vol.close.ts[,"Close"], order=c(1,1,0), season=list(order=c(2,1,0)), xreg=amazon.vol.close.ts[,"Open"]))

## Series: amazon.vol.close.ts[, "Close"]   
## Regression with ARIMA(1,1,0)(2,1,0)[12] errors   
##   
## Coefficients:  
## ar1 sar1 sar2 xreg  
## -0.3130 -1.1043 -0.7208 0.0530  
## s.e. 0.2653 0.1361 0.1613 0.2292  
##   
## sigma^2 estimated as 8678: log likelihood=-218.27  
## AIC=446.54 AICc=448.61 BIC=454.32

Plot

checkresiduals(fit)



##   
## Ljung-Box test  
##   
## data: Residuals from Regression with ARIMA(1,1,0)(2,1,0)[12] errors  
## Q\* = 16.005, df = 6, p-value = 0.01373  
##   
## Model df: 4. Total lags used: 10