Jian Hui Mai
MA 641: Time Series
Forecasting AAPL Stock and Inventories to Sales Ratio for Retailers
May 2022

Forecasting of AAPL (Apple) Stock

Nonseasonal Time Series Dataset

Motivation and Introduction to the Problem

Everyone wants to know what the best time is to buy or sell a stock. Everyone wants to buy low and sell high but what is the low and what is the high? There are many ways we can approach this problem; we can utilize financial analysis or create models. Concerning models, we can create LSTM or ARIMA and GARCH models. In this project, I will create ARIMA and GARCH models to forecast adjusted closing prices.

Data

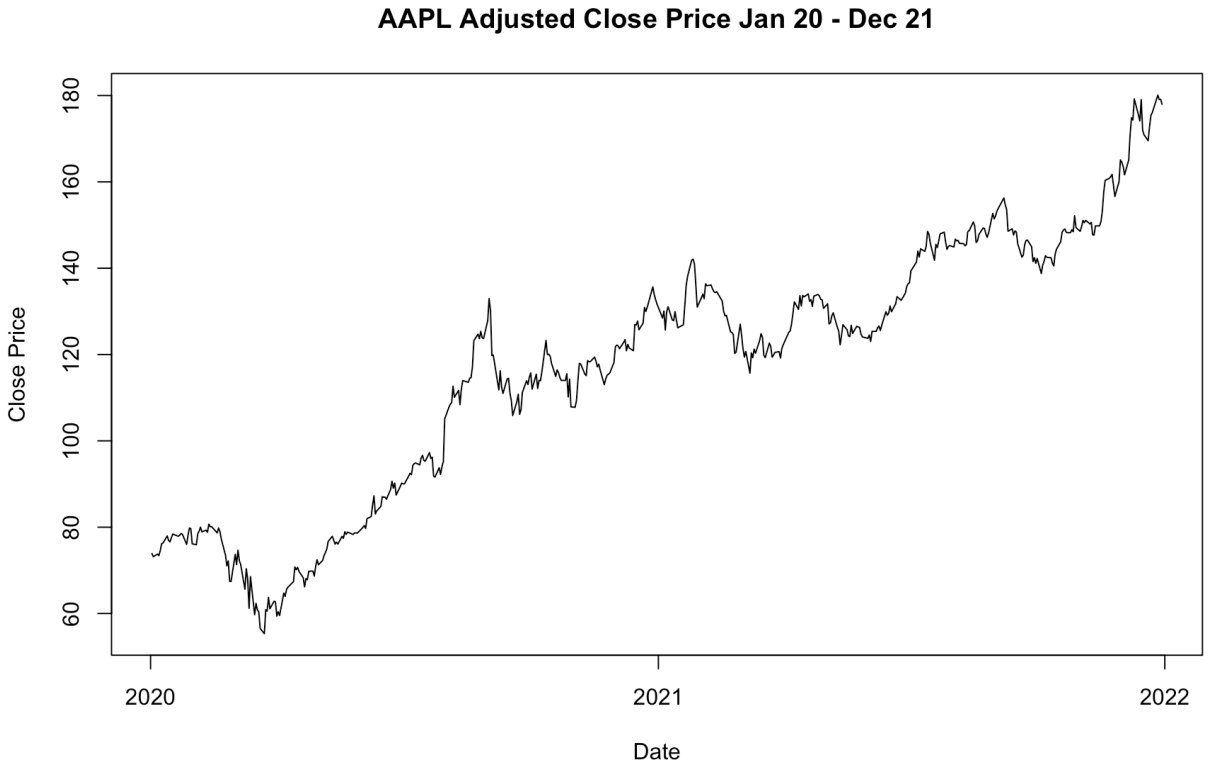
I retrieved the Apple stock data for January 1, 2020, to December 31, 2021, from Yahoo Finance. Due to holidays and weekends, the dataset contains 504 only rows. The dataset is in the format of a CSV (Comma Separated Values) file and contains the date, open, high, low, close, adj. close, and volume columns. I will only be working with the adjusted close prices for simplicity stake. The adjusted closing price is the adjusted last trade price of the day before the market closes at 4:00 P.M. Eastern Time. Due to the speed of trades, this is the true last trade of the day before the market is closed and is added later. Since the dataset is in a structured format without any missing values, data cleaning would not be required. Additionally, given the type of this dataset, we can say there are not any outliers.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	1/2/2020	74.06	75.15	73.7975	75.0875	73.89433	135480400
3	1/3/2020	74.2875	75.145	74.125	74.3575	73.17593	146322800
4	1/6/2020	73.4475	74.99	73.1875	74.95	73.759	118387200
5	1/7/2020	74.96	75.225	74.37	74.5975	73.41212	108872000
6	1/8/2020	74.29	76.11	74.29	75.7975	74.59304	132079200
7	1/9/2020	76.81	77.6075	76.55	77.4075	76.17746	170108400
8	1/10/2020	77.65	78.1675	77.0625	77.5825	76.34969	140644800
9	1/13/2020	77.91	79.2675	77.7875	79.24	77.98084	121532000
10	1/14/2020	79.175	79.3925	78.0425	78.17	76.92785	161954400
11	1/15/2020	77.9625	78.875	77.3875	77.835	76.59817	121923600
12	1/16/2020	78.3975	78.925	78.0225	78.81	77.55767	108829200
13	1/17/2020	79.0675	79.685	78.75	79.6825	78.41631	137816400
14	1/21/2020	79.2975	79.755	79	79.1425	77.88489	110843200
15	1/22/2020	79.645	79.9975	79.3275	79.425	78.16291	101832400
16	1/23/2020	79.48	79.89	78.9125	79.8075	78.53933	104472000
17	1/24/2020	80.0625	80.8325	79.38	79.5775	78.31297	146537600
18	1/27/2020	77.515	77.9425	76.22	77.2375	76.01017	161940000
19	1/28/2020	78.15	79.6	78.0475	79.4225	78.16044	162234000
20	1/29/2020	81.1125	81.9625	80.345	81.085	79.79652	216229200
21	1/30/2020	80.135	81.0225	79.6875	80.9675	79.68088	126743200
22	1/31/2020	80.2325	80.67	77.0725	77.3775	76.14793	199588400
23	2/3/2020	76.075	78.3725	75.555	77.165	75.93881	173788400
24	2/4/2020	78.8275	79.91	78.4075	79.7125	78.44583	136616400
25	2/5/2020	80.88	81.19	79.7375	80.3625	79.0855	118826800

The picture above is a sample of the dataset.

Plot of Apple Stock Dataset

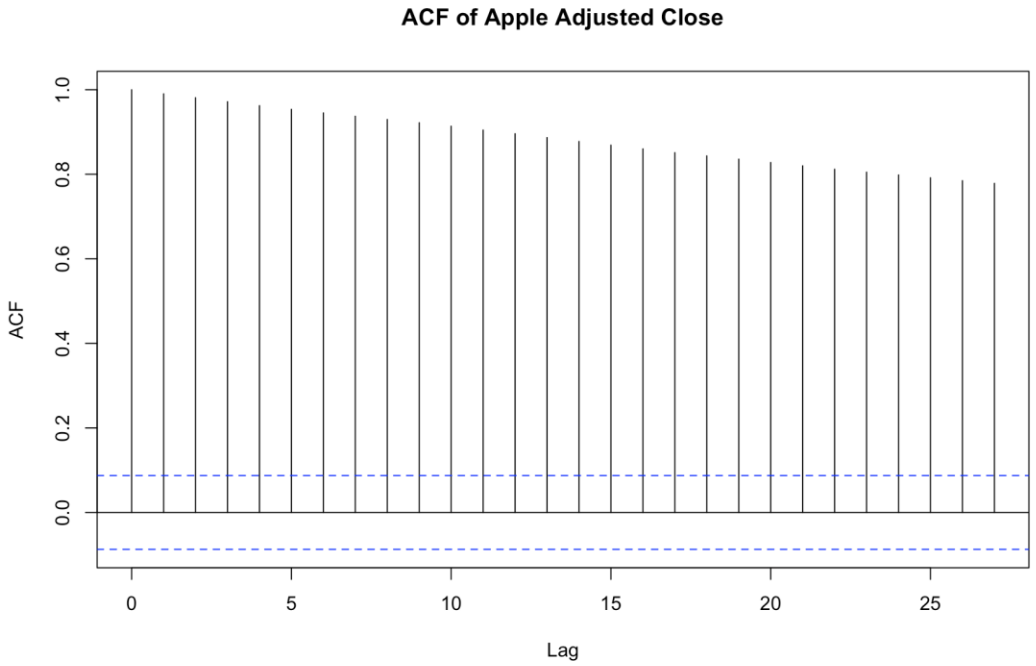
```
> AAPL <- read.csv("AAPL.csv", header = TRUE)
> AAPL$Date <- as.Date(AAPL$Date, format = '%Y-%m-%d')
> plot(AAPL$Adj.Close ~ AAPL$Date, type = "l", xlab = 'Date', ylab = "Close Price", main =
"AAPL Adjusted Close Price Jan 20 - Dec 21")
```



According to the graph above, there does not appear to be a seasonality trend. We would need to confirm by the ACF plot. The graph also appears to be trending upwards therefore not stationary.

Seasonality: ACF Plot

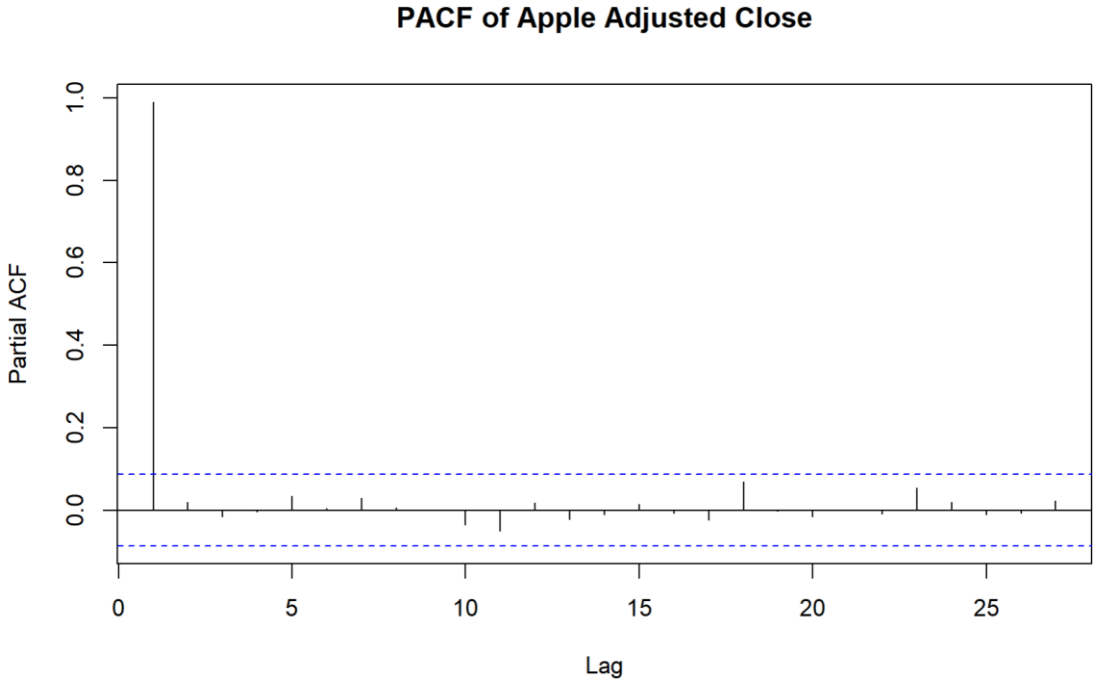
> acf(AAPL$Adj.Close, main = 'ACF of Apple Adjusted Close')



According to the ACF plot, we can confirm that this dataset is not seasonal since there is no yearly trend between the lags. In other words, we cannot see a year-to-year trend.

PACF Plot

> pacf(AAPL$Adj.Close, main = 'PACF of Apple Adjusted Close')



Box-Jenkins Models

Stationary: Dickey-Fuller Test

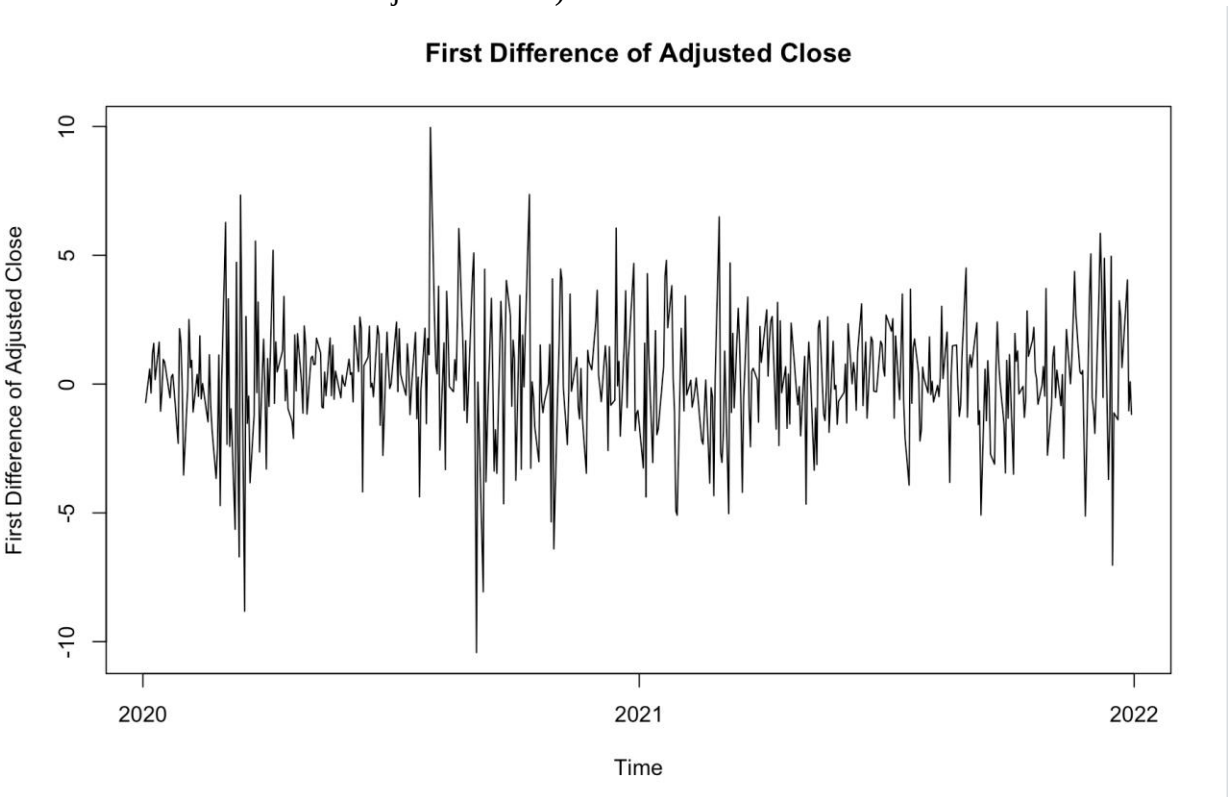
> library(tseries)
> adf.test(AAPL$Adj.Close)

```
        Augmented Dickey-Fuller Test

data:  AAPL$Adj.Close
Dickey-Fuller = -2.3566, Lag order = 7, p-value = 0.4274
alternative hypothesis: stationary
```

Since the p-value from the Dickey-Fuller Test is greater than 0.05, we accept the null hypothesis that the dataset is not stationary. With this in mind, we must apply differencing to make the dataset stationary.

Stationarity through Differencing

> first_diff <- diff(AAPL$Adj.Close)
> date_diff <- AAPL$Date[2:504]
> plot(first_diff ~ date_diff, ylab = 'First Difference of Adjusted Close', xlab = 'Time', type = 'l', main = 'First Difference of Adjusted Close')



According to the graph above, the dataset (after taking the first difference) appears to be stationary. We need to use the Dickey-Fuller test again to confirm.

Dickey Fuller Test after Differencing

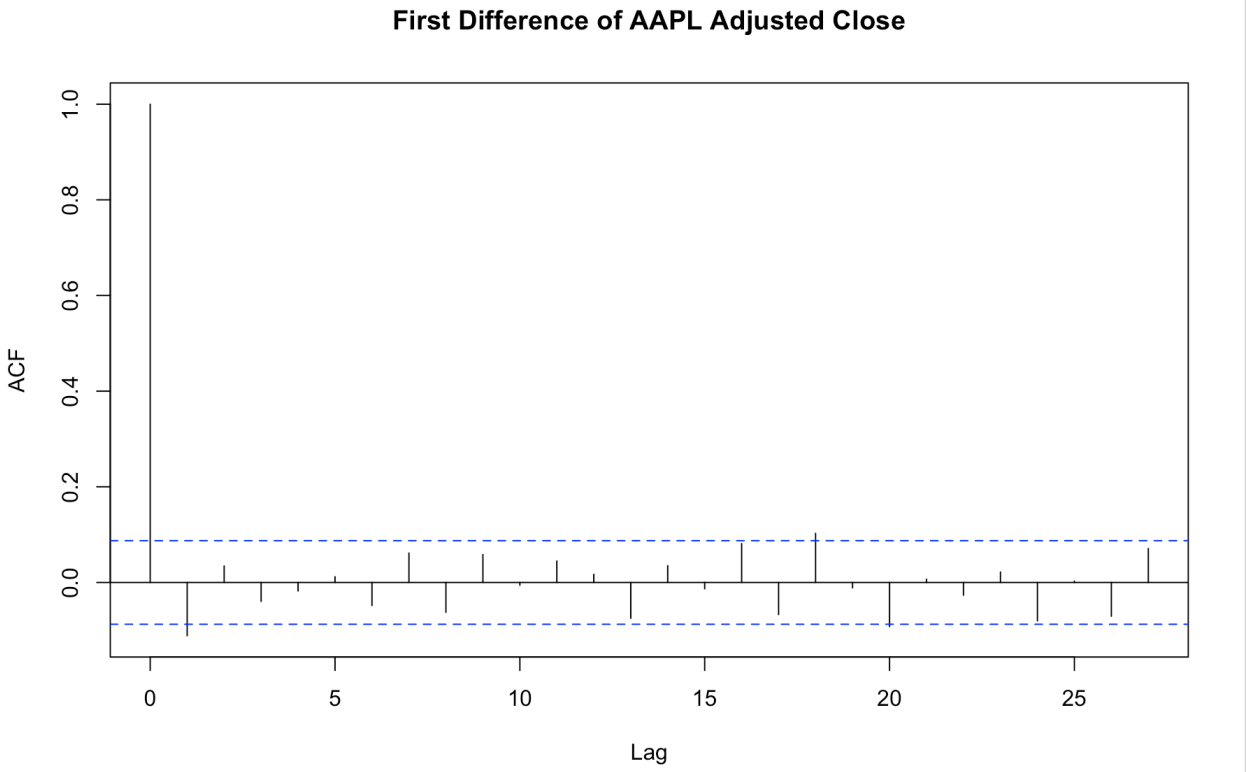
> library(tseries)
> adf.test(first_diff)

```
        Augmented Dickey-Fuller Test

data:  first_diff
Dickey-Fuller = -8.3341, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(first_diff) : p-value smaller than printed p-value
```

Since the p-value from the Dickey-Fuller Test is less than 0.05, we reject the null hypothesis that the data (after taking the first difference) is not stationary. In other words, there is not enough evidence to say that the dataset (after taking the first difference) is not stationary therefore the dataset is stationary.

ACF of First Difference of AAPL Adjusted Close

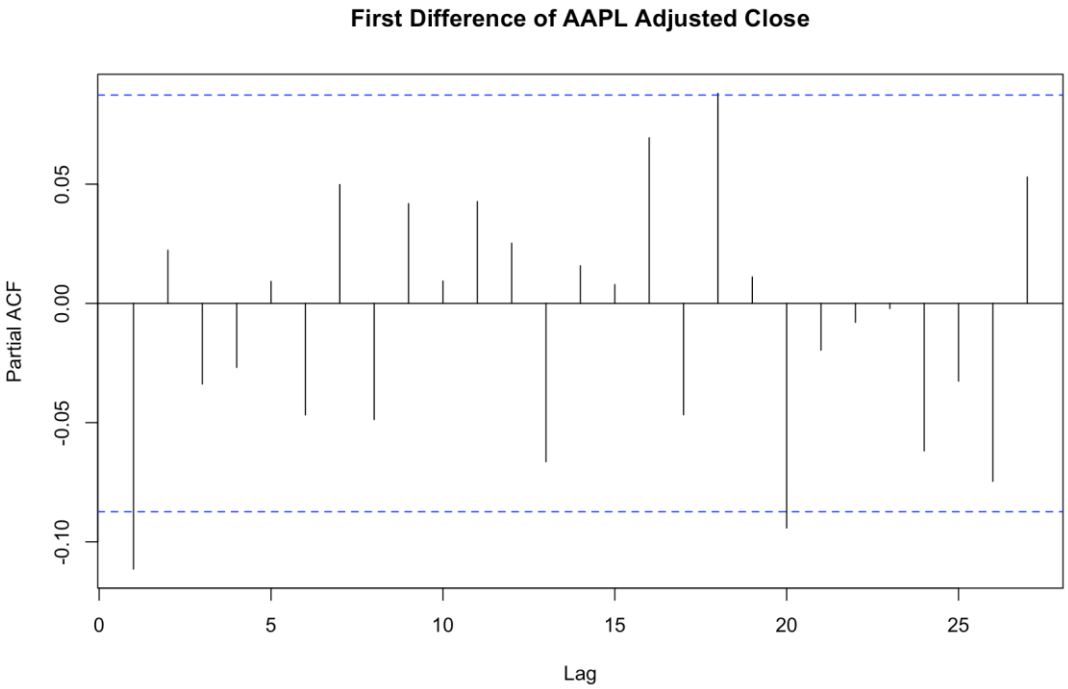
> acf(first_diff, main = "First Difference of AAPL Adjusted Close")



This ACF plot suggests that the dataset follows a MA (3). I believe it is an MA (3) because there are three significant lags (after lag 0) above the confidence interval.

PACF of First Difference of AAPL Adjusted Close

```
> pacf(first_diff, main = "First Difference of AAPL Adjusted Close")
```



This PACF plot suggests that the dataset follows AR (2). I believe it is an AR (2) because there are two significant lags above the confidence interval.

EACF of First Difference of AAPL Adjusted Closed

```
> library("TSA")
> eacf(first_diff)
```

```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o o o o o o o o o  o  o  o
1 x o o o o o o o o o o  o  o  o
2 x x o o o o o o o o o  o  o  o
3 x x x o o o o o o o o  o  o  o
4 x x x x o o o o o o o  o  o  o
5 x x x x o o o o o o o  o  o  o
6 x o o x o x o o o o o  o  o  o
7 x x o o o x o o o o o  o  o  o
```

The EACF suggests both an ARMA (0,1) and ARMA (1,1) model. Since the PACF plot suggests an AR (2) model, I believe this dataset might be ARMA (1,1) and not ARMA (0,1). I will fit both options to see which has the lowest AIC scores.

Parameter Redundancy

```
> arima(x = AAPL$Adj.Close, order = c(2,1,3))

Call:
arima(x = AAPL$Adj.Close, order = c(2, 1, 3))

Coefficients:
          ar1      ar2      ma1     ma2      ma3
      -0.0267  -0.0504  -0.0724  0.0871  -0.0391
s.e.   2.9337   1.0897   2.9325  0.8478   0.0664

sigma^2 estimated as 5.717:  log likelihood = -1152.23,  aic = 2314.46
> arima(x = AAPL$Adj.Close, order = c(0, 1, 1))

Call:
arima(x = AAPL$Adj.Close, order = c(0, 1, 1))

Coefficients:
          ma1
      -0.0968
s.e.   0.0430

sigma^2 estimated as 5.732:  log likelihood = -1152.86,  aic = 2307.73
> arima(x = AAPL$Adj.Close, order = c(1, 1, 1))

Call:
arima(x = AAPL$Adj.Close, order = c(1, 1, 1))

Coefficients:
          ar1     ma1
      -0.4027  0.3027
s.e.   0.2940  0.3054

sigma^2 estimated as 5.721:  log likelihood = -1152.4,  aic = 2308.79
```

I tried three different models, ARIMA (2,1,3), ARIMA (0,1,1) and ARIMA (1,1,1). In other to determine the best model, I used the AIC. The lower the AIC values, the better the fit of the model to the data. According to this standpoint, I would say that ARIMA (0,1,1) is the best model given that it has the lowest AIC of the three.

Parameter Estimation

```
> arima(AAPL$Adj.Close, order = c(0,1,1), method = 'ML')$coef
        ma1
-0.09675169
> arima(AAPL$Adj.Close, order = c(0,1,1), method = 'CSS')$coef
       ma1
-0.0969195
```

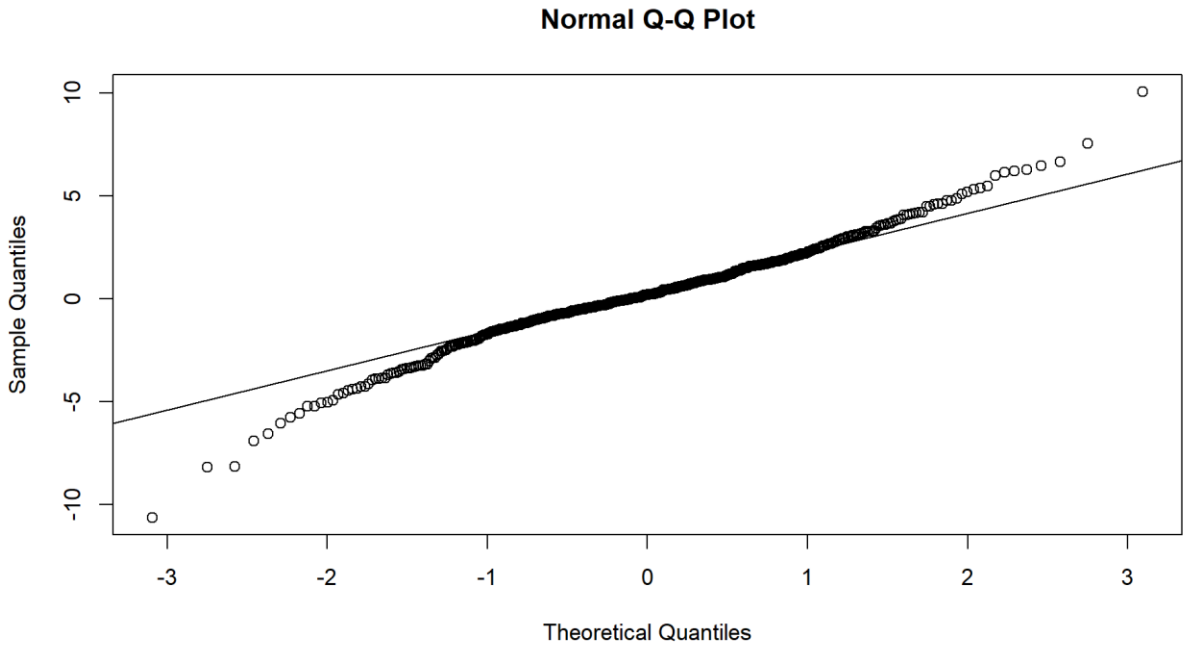
The estimation of theta using Maximum Likelihood Estimation and Conditional Sum of Square Estimator is similar.

Final ARIMA Model

```
> final_model <- arima(x = AAPL$Adj.Close, order = c(0, 1, 1))
```

QQ Plot

```
> qqnorm(residuals(final_model))
> qqline(residuals(final_model))
```



The residuals of the ARIMA model do not appear to follow the normal distribution. We need to use the Shapiro Wilk Test to confirm.

Shapiro-Wilk Test for Normality

```
> shapiro.test(residuals(final_model))
```

```
        Shapiro-Wilk normality test

data:  residuals(final_model)
W = 0.9782, p-value = 7.611e-07
```

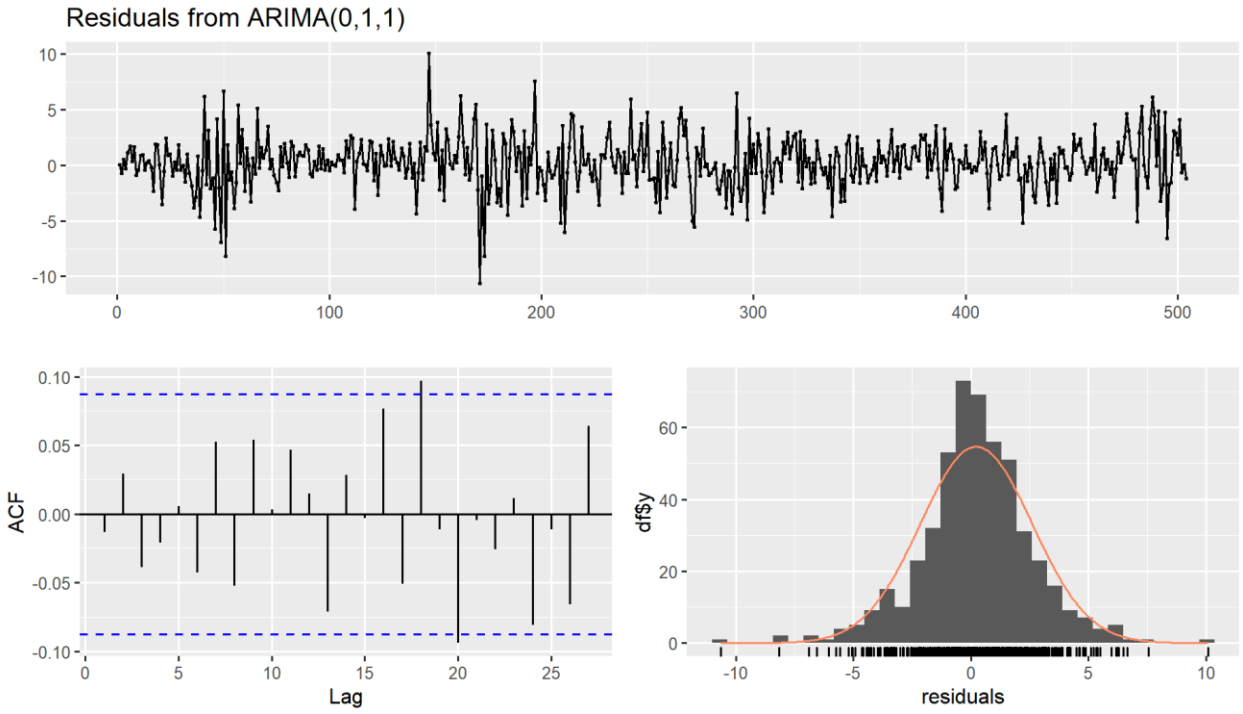
Since the P-value is less than 0.05, we must reject the null hypothesis. Therefore, the residuals do not follow the normal distribution.

Residuals Analysis

```
> library(forecast)
> checkresiduals(final_model)
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,1)
Q* = 6.7951, df = 9, p-value = 0.6584

Model df: 1.   Total lags used: 10
```

Given the P value is greater than 0.05, we accept the null hypothesis that error terms are uncorrelated.



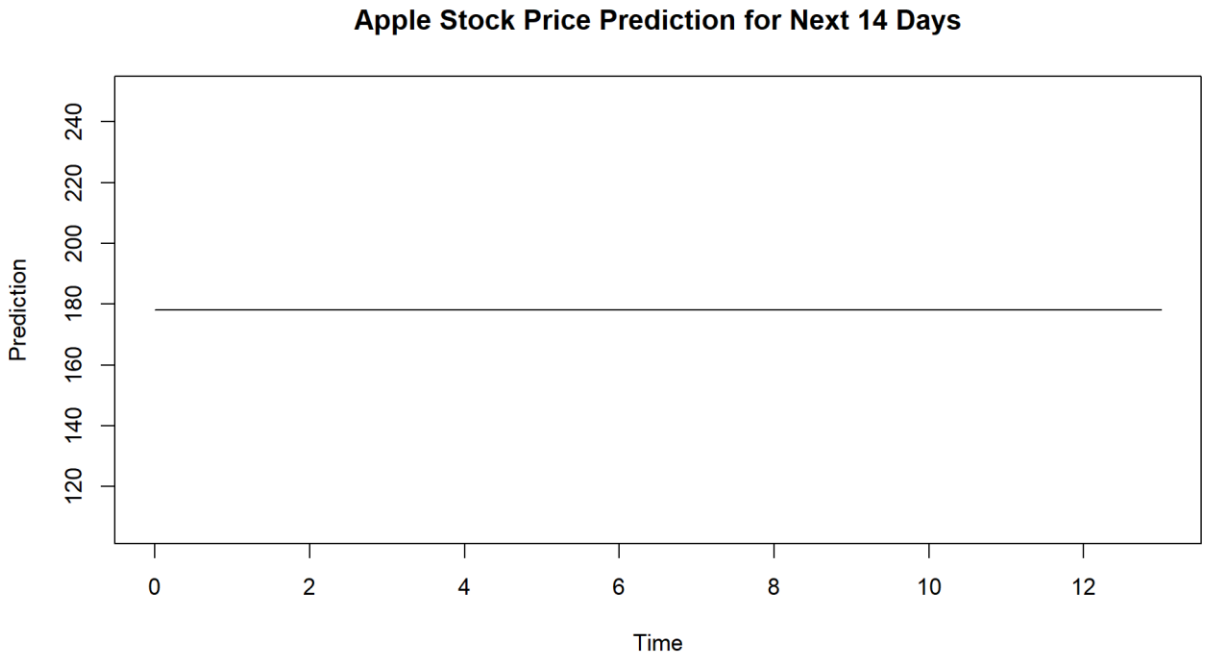
According to the ACF plot, there does not appear to be any significant autocorrelations in the residuals.

Forecasting

> predict(final_model, n.ahead=14)

```
$pred
Time Series:
Start = 505
End = 518
Frequency = 1
 [1] 178.087 178.087 178.087 178.087 178.087 178.087
 [7] 178.087 178.087 178.087 178.087 178.087 178.087
[13] 178.087 178.087

$se
Time Series:
Start = 505
End = 518
Frequency = 1
 [1] 2.394150 3.226207 3.883924 4.445371 4.943458
 [6] 5.395761 5.812976 6.202189 6.568379 6.915205
[11] 7.245448 7.561281 7.864441 8.156340
```

> plot(x = c(0:13), y = predict(final_model, n.ahead = 14)$pred, type = 'l', ylab = 'Prediction', xlab = 'Time', main = 'Apple Stock Price Prediction for Next 14 Days')



Since the predictions generated by the ARIMA(0,1,1) model are linear, we must apply to the GARCH model.

GARCH Model

> new_data <- residuals(final_model)**2

We generate a GARCH Model using the residuals of the final ARIMA model.

Check for Stationary

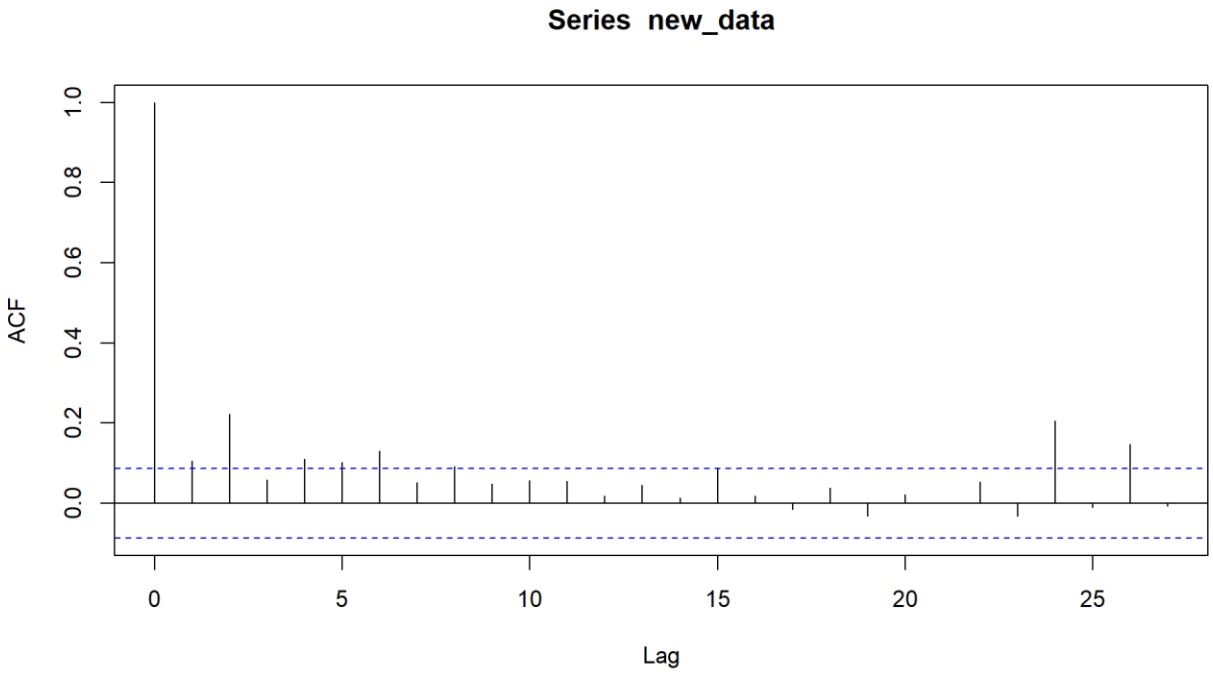
```
> library(tseries)
> adf.test(new_data)
```

```
        Augmented Dickey-Fuller Test

data:  new_data
Dickey-Fuller = -5.8841, Lag order = 7,
p-value = 0.01
alternative hypothesis: stationary
```

According to the Dickey-Fuller Test, the data containing the residual is stationary because the P value is less than 0.05.

ACF

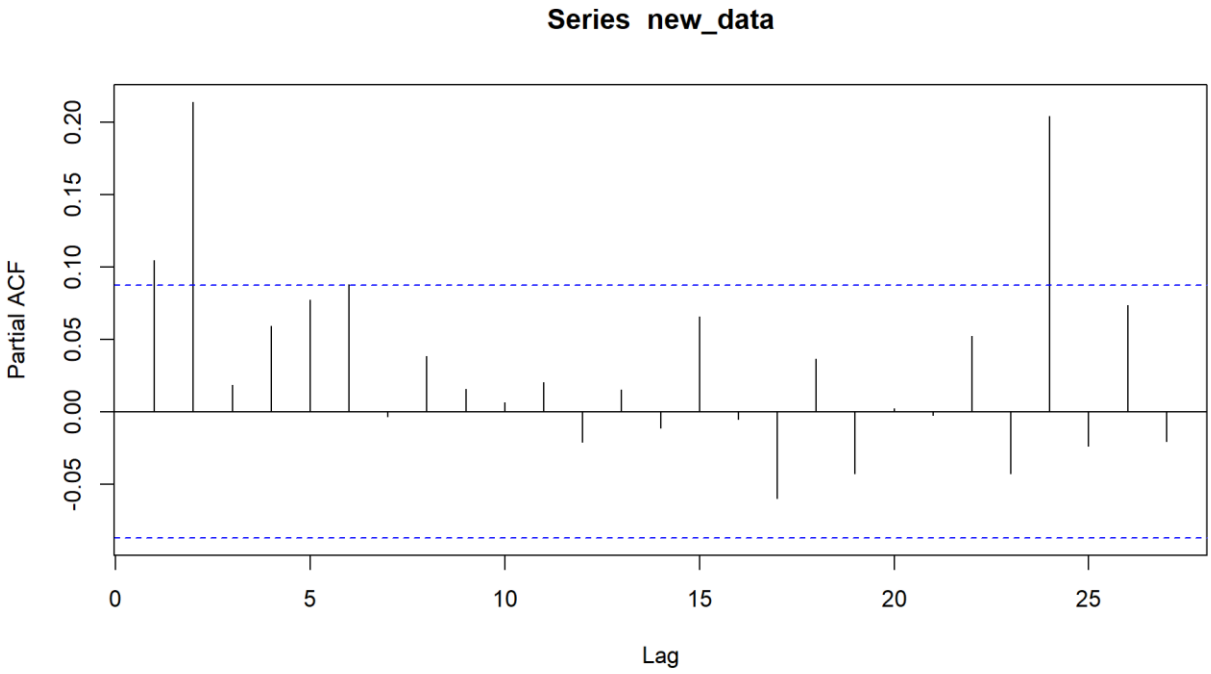
```
> acf(new_data)
```



According to the ACF above, it appears that there are two significant lags above the confidence interval. Therefore, I believe this is a MA (2) model.

PACF

> pacf(new_data)



According to the PACF above, there are two significant lags (two bars above the dotted lines). Therefore, the graph suggests it is an AR (2) model.

EACF

> eacf(new_data)

```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x o x x x o x o o o  o  o  o
1 x x o o o o o o o o o  o  o  o
2 o x x o o o o o o o o  o  o  o
3 x o o o o o o o o o o  o  o  o
4 x x o x o o o o o o o  o  o  o
5 x o x x x o o o o o o  o  o  o
6 o x x o o o o o o o o  o  o  o
7 o x x x o x o o o o o  o  o  o
```

According to the EACF above, it suggests an ARMA (0,2), ARMA(1,2) and ARMA(2,0).

Parameter Redundancy

> library(rugarch)

GARCH(2,2) ARIMA(0,1,1)

> ugarchfit(ugarchspec(variance.model = list(garchOrder = c(2, 2)), mean.model=list(armaOrder=c(0,1)), fixed.pars=list(arfima = 1)), data = AAPL$Adj.Close)

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(2,2)
Mean Model       : ARFIMA(0,0,1)
Distribution     : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu     124.007076    0.716654 173.036085 0.000000
ma1      0.869174    0.013218  65.759018 0.000000
omega    3.310561    1.071140   3.090690 0.001997
alpha1   0.012404    0.021571   0.575010 0.565285
alpha2   0.986595    0.134245   7.349230 0.000000
beta1    0.000000    0.020558   0.000000 1.000000
beta2    0.000001    0.098231   0.000013 0.999989

Robust Standard Errors:
        Estimate  Std. Error   t value Pr(>|t|)
mu     124.007076    2.155658 57.526315 0.000000
ma1      0.869174    0.024218 35.889338 0.000000
omega    3.310561    1.290587  2.565158 0.010313
alpha1   0.012404    0.024109  0.514481 0.606916
alpha2   0.986595    0.215548  4.577150 0.000005
beta1    0.000000    0.023963  0.000000 1.000000
beta2    0.000001    0.172893  0.000008 0.999994

LogLikelihood : -1851.735

Information Criteria
------------------------------------

Akaike       7.3759
Bayes        7.4346
Shibata      7.3756
Hannan-Quinn 7.3989

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      214.6       0
Lag[2*(p+q)+(p+q)-1][2]     389.0       0
Lag[4*(p+q)+(p+q)-1][5]     796.3       0
d.o.f=1
H0 : No serial correlation
```

GARCH(0,2) ARIMA(0,1,1)

> ugarchfit(ugarchspec(variance.model = list(garchOrder = c(0, 2)), mean.model=list(armaOrder=c(0,1)), fixed.pars=list(arfima = 1)), data = AAPL$Adj.Close)

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model      : sGARCH(0,2)
Mean Model       : ARFIMA(0,0,1)
Distribution     : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu     119.982210    1.772715 6.7683e+01  0.00000
ma1      0.926263    0.012119 7.6434e+01  0.00000
omega    1.306836    0.142300 9.1837e+00  0.00000
beta1    0.992573    0.000481 2.0622e+03  0.00000
beta2    0.000019    0.001508 1.2621e-02  0.98993

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu     119.982210    8.383005 1.4313e+01 0.000000
ma1      0.926263    0.014755 6.2775e+01 0.000000
omega    1.306836    0.433853 3.0122e+00 0.002594
beta1    0.992573    0.000714 1.3901e+03 0.000000
beta2    0.000019    0.001543 1.2332e-02 0.990161

LogLikelihood : -2104.739

Information Criteria
------------------------------------

Akaike       8.3720
Bayes        8.4139
Shibata      8.3718
Hannan-Quinn 8.3884

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      367.9       0
Lag[2*(p+q)+(p+q)-1][2]     608.1       0
Lag[4*(p+q)+(p+q)-1][5]    1223.2       0
d.o.f=1
H0 : No serial correlation
```

GARCH(1,2) ARIMA(0,1,1)

> ugarchfit(ugarchspec(variance.model = list(garchOrder = c(1, 2)), mean.model=list(armaOrder=c(0,1)), fixed.pars=list(arfima = 1)), data = AAPL$Adj.Close)

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,2)
Mean Model      : ARFIMA(0,0,1)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu     125.72144    0.861250 145.975532 0.000000
ma1      0.79179    0.022784  34.751502 0.000000
omega    1.25031    0.645406   1.937238 0.052716
alpha1   0.35155    0.077030   4.563791 0.000005
beta1    0.64346    0.162302   3.964560 0.000074
beta2    0.00000    0.167209   0.000001 0.999999

Robust Standard Errors:
        Estimate  Std. Error   t value Pr(>|t|)
mu     125.72144    3.321354 37.852462 0.000000
ma1      0.79179    0.031218 25.363312 0.000000
omega    1.25031    0.632377  1.977154 0.048024
alpha1   0.35155    0.084908  4.140329 0.000035
beta1    0.64346    0.171307  3.756158 0.000173
beta2    0.00000    0.244203  0.000001 0.999999

LogLikelihood : -1892.175

Information Criteria
------------------------------------

Akaike       7.5324
Bayes        7.5827
Shibata      7.5322
Hannan-Quinn 7.5522

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      247.3       0
Lag[2*(p+q)+(p+q)-1][2]     445.7       0
Lag[4*(p+q)+(p+q)-1][5]     891.2       0
d.o.f=1
H0 : No serial correlation
```

GARCH(2,0) ARIMA(0,1,1)

> ugarchfit(ugarchspec(variance.model = list(garchOrder = c(2, 0)), mean.model=list(armaOrder=c(0,1)), fixed.pars=list(arfima = 1)), data = AAPL$Adj.Close)

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(2,0)
Mean Model      : ARFIMA(0,0,1)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu     124.007101    0.655730 189.11315 0.000000
ma1      0.869174    0.012391  70.14543 0.000000
omega    3.310288    1.035823   3.19580 0.001394
alpha1   0.012407    0.019069   0.65066 0.515268
alpha2   0.986592    0.076848  12.83824 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu     124.007101    1.749964 70.86266 0.000000
ma1      0.869174    0.020425 42.55342 0.000000
omega    3.310288    1.467370  2.25593 0.024075
alpha1   0.012407    0.019737  0.62863 0.529590
alpha2   0.986592    0.045831 21.52674 0.000000

LogLikelihood : -1851.735

Information Criteria
------------------------------------

Akaike       7.3680
Bayes        7.4099
Shibata      7.3678
Hannan-Quinn 7.3844

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      214.6       0
Lag[2*(p+q)+(p+q)-1][2]     389.0       0
Lag[4*(p+q)+(p+q)-1][5]     796.3       0
d.o.f=1
H0 : No serial correlation
```

*** The second portion of these model summaries are not included ***

Since GARCH(2,0) ARIMA(0,1,1) has the lowest AIC, that would be our final model.

Forecasting

```
> final_model2 <- ugarchfit(ugarchspec(variance.model = list(garchOrder = c(2, 0)),
mean.model=list(armaOrder=c(0,1)), fixed.pars=list(arfima = 1)), data = AAPL$Adj.Close)
> ugarchforecast(final_model_2, n.ahead=20)
```

```
*----------------------------------*
*       GARCH Model Forecast       *
*----------------------------------*
Model: sGARCH
Horizon: 20
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=1971-05-19 20:00:00]:
     Series Sigma
T+1   148.2 30.14
T+2   124.0 27.86
T+3   124.0 30.15
T+4   124.0 27.93
T+5   124.0 30.16
T+6   124.0 28.01
T+7   124.0 30.18
T+8   124.0 28.08
T+9   124.0 30.19
T+10  124.0 28.15
T+11  124.0 30.21
T+12  124.0 28.22
T+13  124.0 30.22
T+14  124.0 28.29
T+15  124.0 30.24
T+16  124.0 28.36
T+17  124.0 30.26
T+18  124.0 28.43
T+19  124.0 30.27
T+20  124.0 28.50
```

Conclusions in the Context of the Problem

With the GARCH model I created from the Apple adjusted closing price dataset, we can forecast the adjusted closing stock price for the next twenty days. This forecast can provide a guide for traders to determine the “best” time to sell or buy Apple stock.

Forecasting Inventories to Sales Ratio for Retailers
Seasonal Time Series Dataset

Motivation and Introduction to the Problem

Retailers all over the country want to maintain a certain level of inventory throughout the year. For some months, they want to have more inventory given the higher demand in the months ahead. For other months, they would want to have less inventory given the lower demand in the months ahead. Retailers want to stock as much inventory as possible to prevent selling out. However, they also do not want to stock too much if there is lower demand since they will have to clear them when a new season arrives. In other words, retailers must maintain the delicate balance of supply and demand. With this in mind, retailers must know the best amount of inventory they must have on hand. They can turn to machine learning or mathematical models to answer this question.

Data

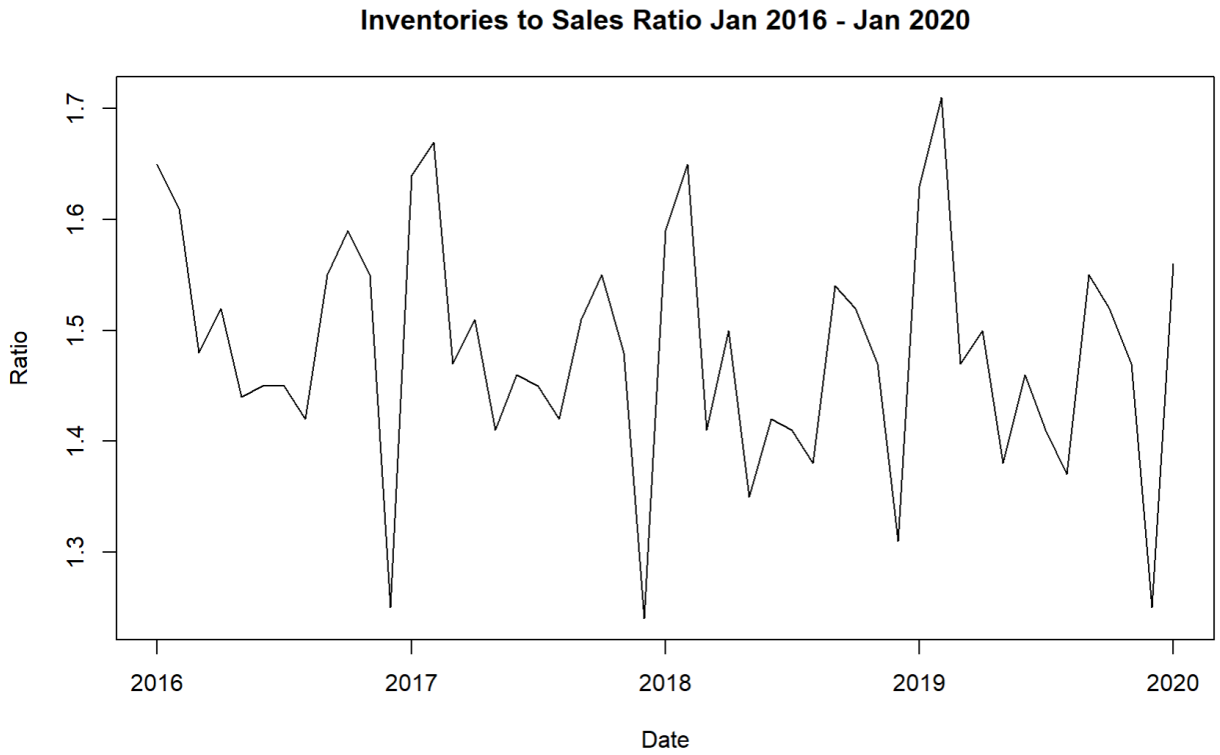
The dataset is retrieved from the St. Louis Fed. It includes the monthly inventories to sales ratio from 1992 to 2022. The inventories to sale ratio represent the amount of inventory (in months) retailers have on hand in relation to the sales for that month. Additionally, the data is not seasonally adjusted. The dataset is in a CSV (comma-separated values) format with a column for the date and "RETAILIRNSA". The "RETAILIRNSA" column is the inventories to sales ratio. For simplicity's sake, I only took a subset of the original data namely from January 2016 to January 2020 due to COVID's impact on store closures and inventories. With this in mind, the dataset only contains 49 rows (one for each month). I did not do any data cleansing for this dataset since there are not any missing values or outliers.

	A	B
1	DATE	RETAILIRNSA
2	1/1/2016	1.65
3	2/1/2016	1.61
4	3/1/2016	1.48
5	4/1/2016	1.52
6	5/1/2016	1.44
7	6/1/2016	1.45
8	7/1/2016	1.45
9	8/1/2016	1.42
10	9/1/2016	1.55
11	10/1/2016	1.59
12	11/1/2016	1.55
13	12/1/2016	1.25
14	1/1/2017	1.64
15	2/1/2017	1.67
16	3/1/2017	1.47
17	4/1/2017	1.51
18	5/1/2017	1.41
19	6/1/2017	1.46
20	7/1/2017	1.45
21	8/1/2017	1.42
22	9/1/2017	1.51
23	10/1/2017	1.55
24	11/1/2017	1.48
25	12/1/2017	1.24

This is a brief sample of the dataset.

Plot of Inventories to Sales Ratio Dataset

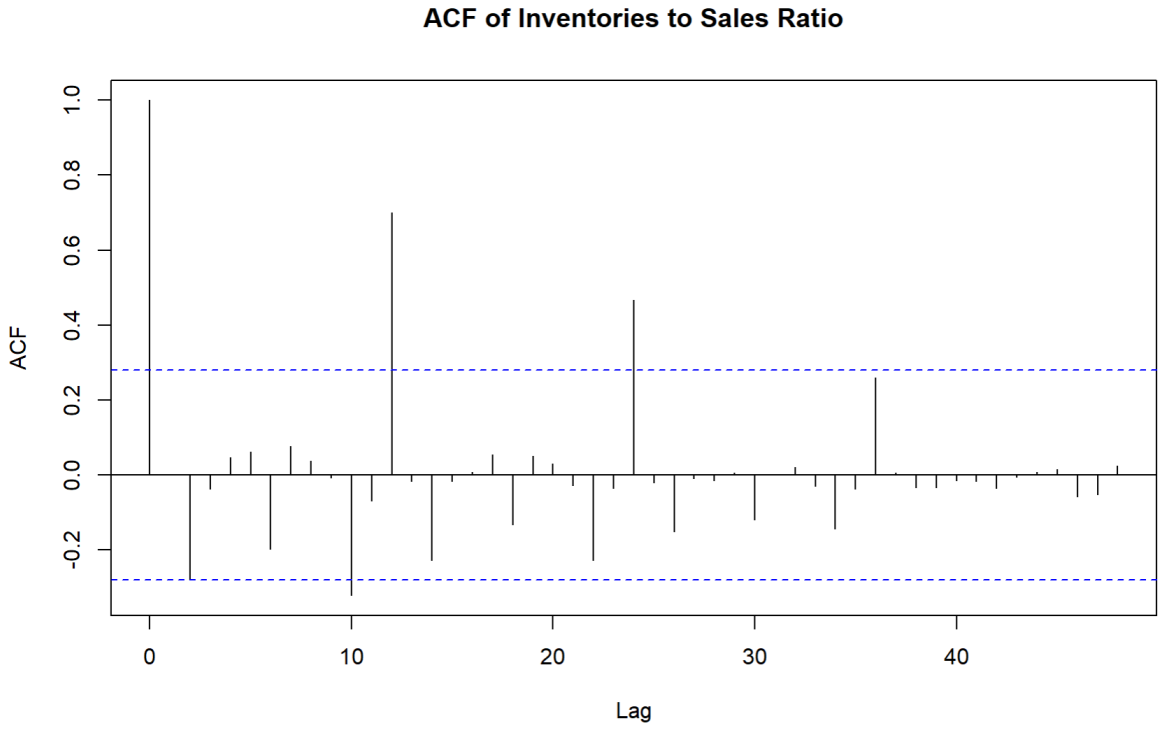
```
> retail <- read.csv("Retail.csv", header = TRUE)
> retail$DATE <- as.Date(retail$DATE, format = '%Y-%m-%d')
> plot(retail$RETAILIRNSA ~ retail$DATE, type = "l", xlab = 'Date', ylab = "Ratio", main =
"Inventories to Sales Ratio Jan 2016 - Jan 2020")
```



According to this plot, the dataset appears to be seasonal because the lowest point is just before the new year and the highest point is right after the new year. You see that pattern in the four years shown. We would need to confirm seasonality through the ACF plot. Also, the plot does appear stationary. We will confirm stationary with the Dickey-Fuller Test.

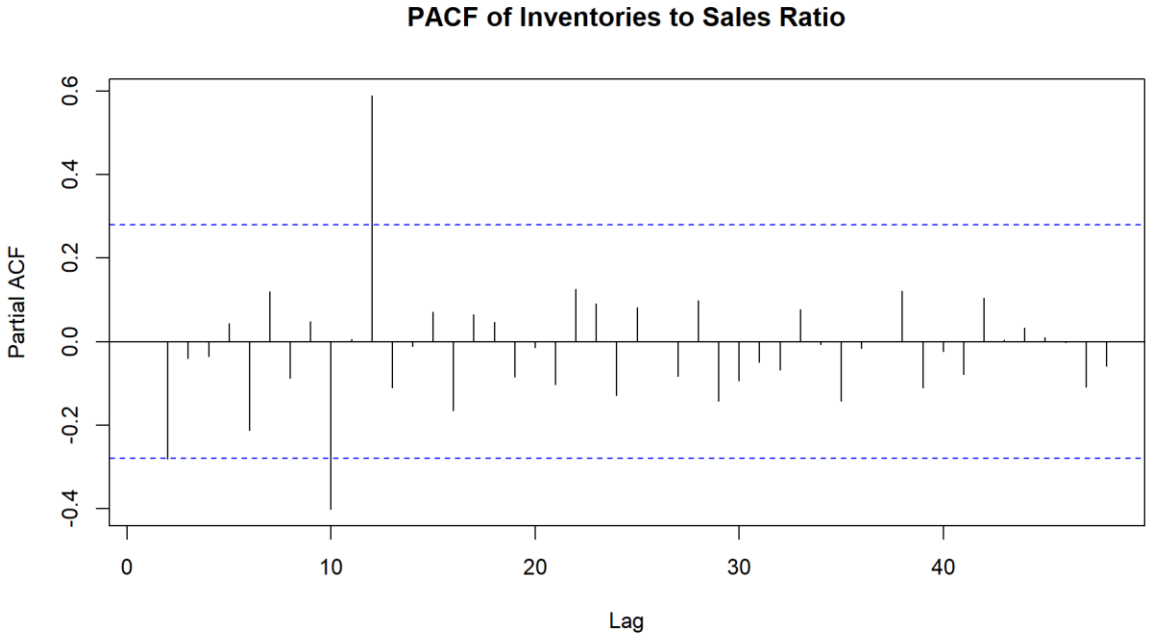
Seasonality: ACF Plot

> acf(retail$RETAILIRNSA, main = 'ACF of Inventories to Sales Ratio', lag.max = 50)



According to the ACF plot, we can say there is seasonality. You can see that the highest lag of each year exists at lags 12, 24, and 36.

PACF Plot



Box-Jenkins Models

Stationary: Dickey-Fuller Test

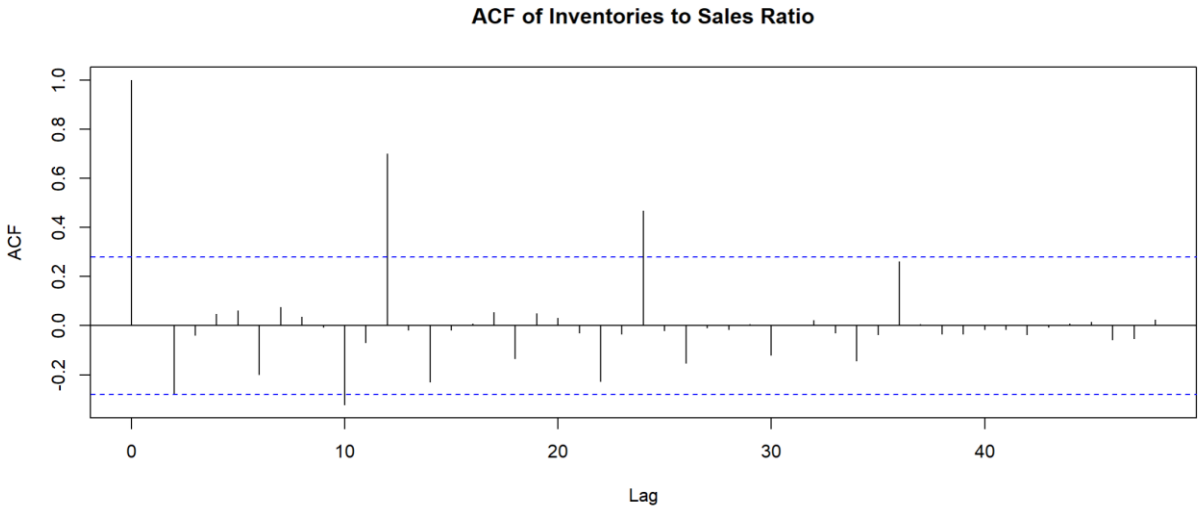
```
> library(tseries)
> adf.test(retail$RETAILIRNSA)
```

```
        Augmented Dickey-Fuller Test

data:  retail$RETAILIRNSA
Dickey-Fuller = -4.3939, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

The P-value of the Dickey-Fuller Test is less than 0.05, therefore we reject the null hypothesis that the dataset is not stationary. Since the dataset is stationary, we do not need to apply any differencing, transforming, or detrending.

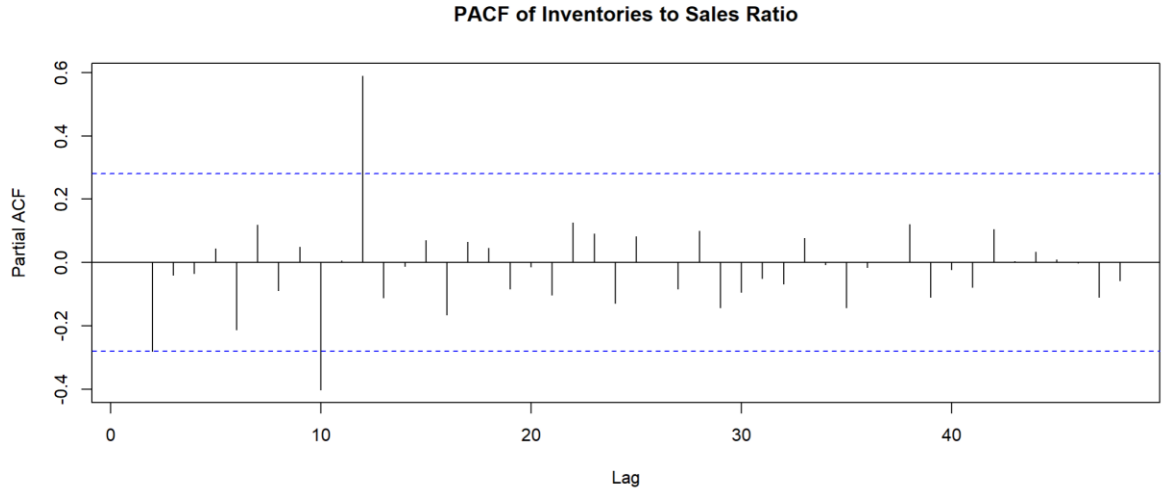
ACF of Inventories to Sales Ratio

```
> acf(retail$RETAILIRNSA, lag.max=50, main = "ACF of Inventories to Sales Ratio")
```



PACF of Inventories to Sales Ratio

```
> pacf(retail$RETAILIRNSA, lag.max=50, main = "PACF of Inventories to Sales Ratio")
```



Since it is a seasonal dataset, we must use the SARIMA models. According to the ACF and PACF, we can infer that this is a SARIMA(1,0,1) x (1,0,3) s = 12 or SARIMA (1,0,1) x (1,0,4) s = 12. The middle value is 0 because we did not take the difference to make the dataset stationary.

Parameter Redundancy

```
> arima(x = retail$RETAILIRNSA, order=c(1,0,1), seasonal=c(1,0,3))

Call:
arima(x = retail$RETAILIRNSA, order = c(1, 0, 1), seasonal = c(1, 0, 3))

Coefficients:
          ar1     ma1     sar1    sma1     sma2
      -0.5058  0.4012  -0.5058  0.6295  -0.4653
s.e.   0.7979  0.8155   0.7979  0.8593   0.2670
         sma3  intercept
      -0.2269     1.4804
s.e.   0.4907     0.0083

sigma^2 estimated as 0.009457:  log likelihood = 44.31,  aic = -72.62
> arima(x = retail$RETAILIRNSA, order=c(1,0,1), seasonal=c(1,0,4))

Call:
arima(x = retail$RETAILIRNSA, order = c(1, 0, 1), seasonal = c(1, 0, 4))

Coefficients:
          ar1      ma1     sar1    sma1     sma2     sma3     sma4  intercept
      -0.2709  -0.2629  -0.2709  0.8494  -0.1325  -0.3372  -0.2201     1.4804
s.e.   0.8594   0.6964   0.8594  0.7560   0.6558   0.4046   0.1751     0.0077

sigma^2 estimated as 0.009326:  log likelihood = 44.64,  aic = -71.29
```

Given the lower AIC from the SARIMA (1,0,1) x (1,0,3) s = 12 (first model), we will use that model to make predictions.

Parameter Estimation

```
> arima(x = retail$RETAILIRNSA, order = c(1,0,1), season = c(1,0,3), method = 'ML')$coef
       ar1        ma1       sar1       sma1       sma2       sma3  intercept
-0.5080499  0.4092189 -0.5080499  0.6260301 -0.4665994 -0.2251812  1.4804449
> arima(x = retail$RETAILIRNSA, order = c(1,0,1), season = c(1,0,3), method = 'CSS')$coef
         ar1          ma1         sar1         sma1         sma2         sma3    intercept
 0.002845099 -0.090055831  0.002845099  0.062926926 -0.581858378  0.051919253  1.476741202
```

The estimations provided by both Maximum Likelihood Estimator and Conditional Sum of Squares Estimator appear vastly different.

Statistical Conclusions

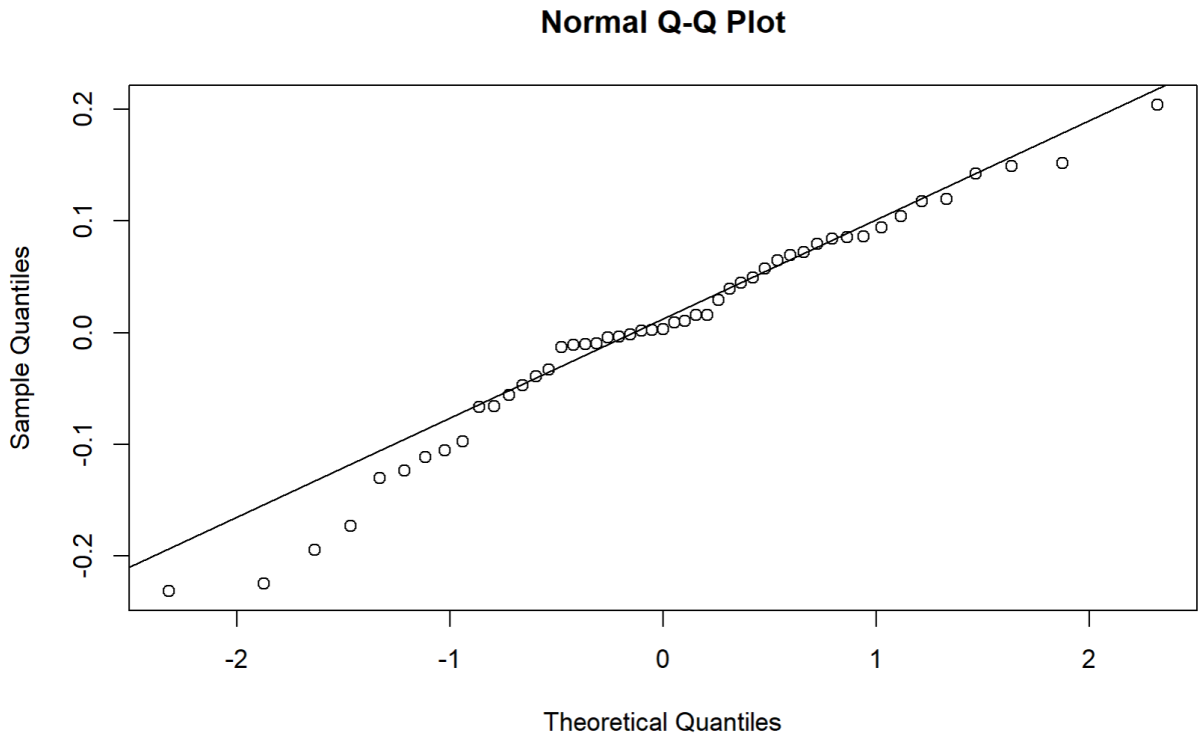
Final Model

```
> final_model <- arima(x = retail$RETAILIRNSA, order=c(1,0,1), seasonal=c(1,0,3))
```

The final model is chosen based on the lowest AIC score of the two models generated from "Parameter Redundancy".

QQ Plot

```
> qqnorm(residuals(final_model))
> qqline(residuals(final_model))
```



According to the QQ Plot, the residual of the final model appears not to be normally distributed since the tails appear far from the QQ line. We would need to use the Shapiro-Wilk test to confirm or deny this claim.

Shapiro-Wilk Test for Normality

```
> shapiro.test(residuals(final_model))
```

```
        Shapiro-Wilk normality test

data:  residuals(final_model)
W = 0.97188, p-value = 0.2872
```

According to the P-value from the Shapiro-Wilk Test, we can say the residuals of the model follow the normal distribution. Since the P-value is greater than 0.05, we can say that we accept

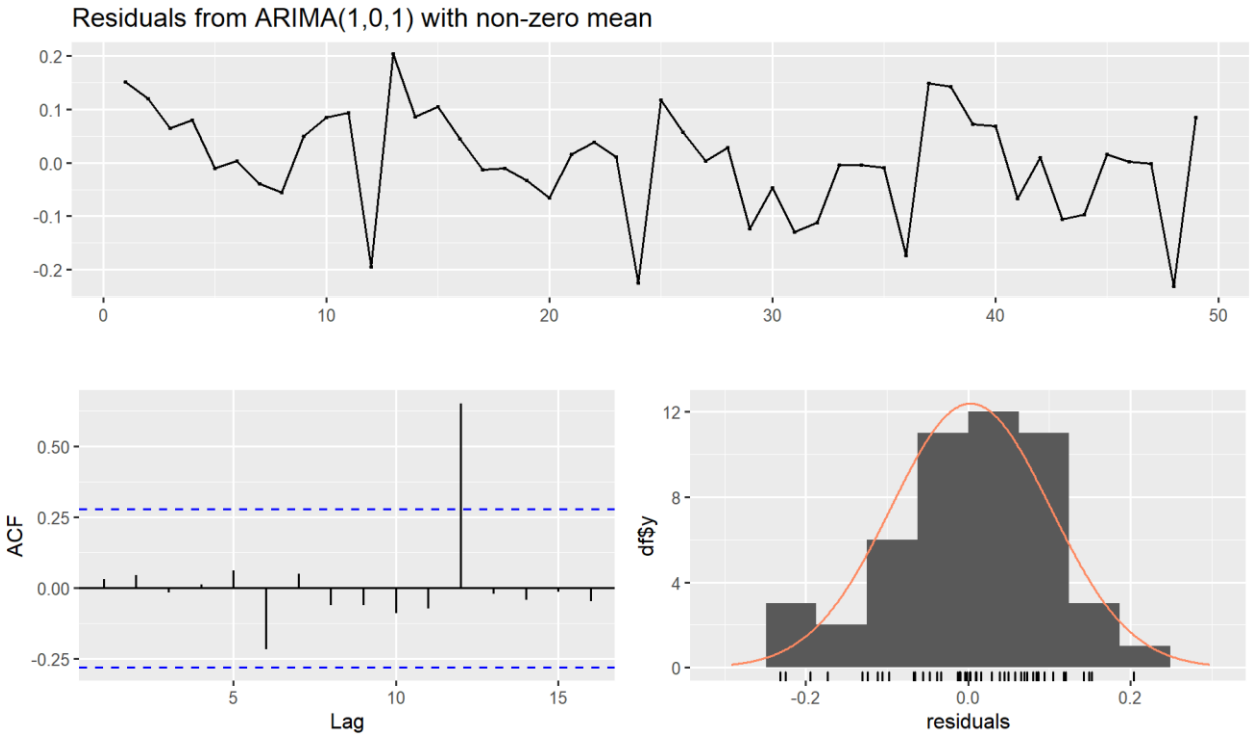
the null hypothesis (residuals following the normal distribution). With this in mind, we can say for certain that the residuals are normally distributed.

Residuals Analysis

```
> library(forecast)
> checkresiduals(final_model)
```

```
        Ljung-Box test

data:  Residuals from ARIMA(1,0,1) with non-zero mean
Q* = 4.222, df = 3, p-value = 0.2385

Model df: 7.   Total lags used: 10
```

Given the P value is greater than 0.05, we accept the null hypothesis that error terms are uncorrelated.



According to the ACF plot, there does appear to be a significant autocorrelation in the residuals.

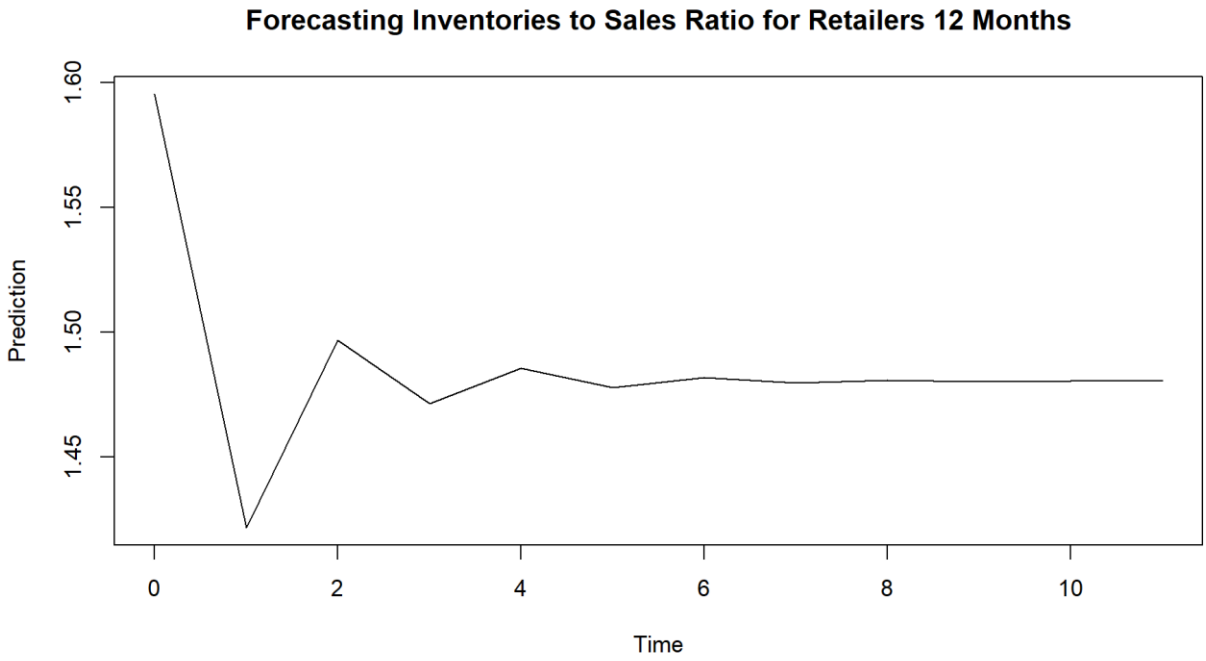
Forecasting

> predict(final_model, n.ahead = 12)

```
$pred
Time Series:
Start = 50
End = 61
Frequency = 1
 [1] 1.595497 1.421601 1.496679 1.471369 1.485451 1.477681 1.481938 1.479620 1.480876 1.480198 1.480562
[12] 1.480367

$se
Time Series:
Start = 50
End = 61
Frequency = 1
 [1] 0.09724548 0.09726340 0.10822036 0.10846632 0.10854377 0.10856771 0.10857499 0.10857717 0.10857782
[10] 0.10857801 0.10857806 0.10857808
```

The first row represents the predictions of the next 12 months of inventory to sale ratios.

> plot(x = c(0:11), y = predict(final_model, n.ahead = 12)$pred, type = 'l', xlab = 'Time', ylab = 'Prediction', main = 'Forecasting Inventories to Sales Ratio for Retailers 12 Months')



Conclusions in the Context of the Problem

With the ARIMA model I created above, retailers can now predict how much inventory they should keep on hand for the next twelve months. It would take the guess work out that analysis and ensure retailers maintain the delicate balance of supply and demand.