New York Stock Exchange

Yen Hung Shen 301301268
Bowen Zhang 301336980
Xiaoying Zhao 301297789

Table of Content

Abstract

This article uses data from the Yahoo Finance database for analysis, and selects Amazon and Wal-Mart as samples. Our sample includes daily stock price data from 2010 to 2016, and we select closing price data for analysis and forecasting. We use the ARIMA model for stock price prediction. Because Amazon and Wal-Mart data are both unstable, we convert stock price data into stock return data. According to the ADF unit root test, stock returns are sequence stationary. We determined the order of the ARIMA model according to the ACF diagram and the PACF diagram. The p value is 0, the q value is 1, and the I value is 1. Finally, we use the ARIMA(0,1,1) model to fit the stock price data, And forecast the data for 30 periods. In addition, we also drew the forecast map and calculated the forecast error MSE. All true values fall within the 95% prediction confidence interval, indicating that the prediction performance of the model in this paper is good. Furthermore, we can also define the linearity from the regression model in the fundamental files. In all of the statements, we selected the "Total equity" as the response variable to see if the other factors on the financial balance would efficited it.

Introduction

Due to the development of the Internet, the retail industry began to shift to online sales. Amazon and Wal-Mart are two representative retail companies. Furthermore, most of the capital inflow to valuable companies and the fundamentals file lists out each of S&P500 company's financial statement indicators. For example, Cash-flow statement, Equity and Liabilities. From all the financial statements, finding the right indicator to predict and analyze a company's Total equity has a high difficulty. The stock price doesn't have a direct correlation to any factors. We use the stock prices of these two companies to explore the predicted performance of the ARIMA model in the price file and we also create a regression model from the data set. More and more investors are paying attention to the stock investment of Amazon and Walmart. In order to increase investment returns, we can predict stock prices. Based on possible changes in stock prices, the investment team reacts in advance. Sell in advance when the stock price may fall, and buy in advance when the stock price may rise, thereby increasing investors' income. The ARIMA model is a classic time series model, which is stable in predicting stock prices. However, a significant shortcoming of the ARIMA model is that it can only predict short-term stock prices, and the accuracy of long-term stock price fluctuation forecasts is poor. Therefore, we only forecast the stock price for the next 30 periods.

Data Description

The selecting variables of the fundamental file, we choose "Total Equity" , "Net Income", "Total Liabilities" , "Common stocks", "Total Assets" , and "ROE". The data in this article comes from the Yahoo Finance database. In our data set, a total of 140 stocks daily price data from January 2010 to December 2016 are included. We only selected stock price data of Amazon and Walmart for analysis and time series forecasting. According to the descriptive statistics, the average stock price of Wal-Mart is $67.04 and the standard deviation is 10.27. Therefore, we calculated the coefficient of variation: 67.04/10.27=6.53. For Amazon, the average stock price is $337.9 and the standard deviation is 189.11. Therefore, the coefficient of variation is calculated as follows: 337.9/189.11=1.79. Since Wal-Mart's coefficient of variation is greater than that of Amazon, Wal-Mart's investment risk is lower, that is, the return per unit risk is higher.

Then, we drew Wal-Mart and Amazon's stock price time series diagrams, as shown in Figure 1 and Figure 2. Amazon's stock price has risen significantly more than Wal-Mart. The results in the figure are different from the results of the coefficient of variation. This is because we use stock prices instead of stock returns for comparison. In the latter part, we will use stock returns for analysis. In addition, we have drawn histograms of the stock prices of the two companies, as shown in Figures 3 and 4. Neither stock price data conforms to a normal distribution.

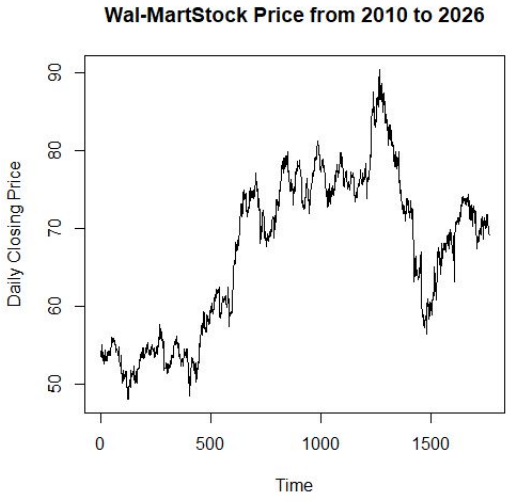


Figure 1

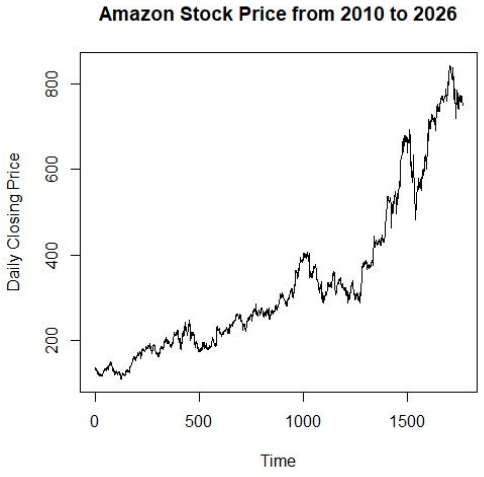


Figure 2

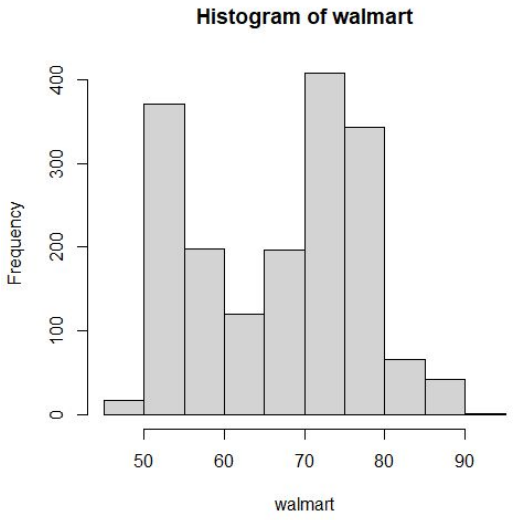


Figure 3

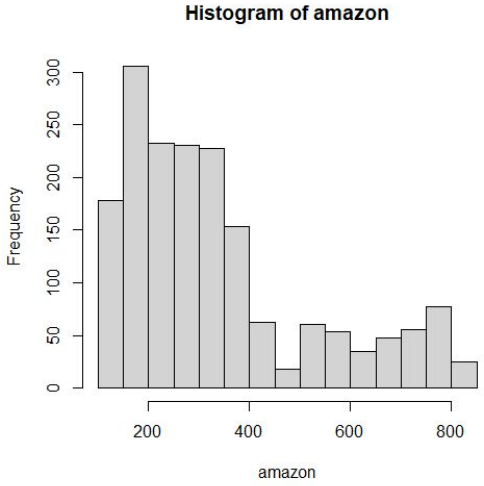


Figure 4.

	Symbol	TE	PE	TL	CS	TA	NI	ROE
1	AAL	-7987000000	2012-12-31	2.489100e+10	1.27000e+08	2.351000e+10	-1876000000	23
2	AAL	-2731000000	2013-12-31	4.500900e+10	5.00000e+06	4.227800e+10	-1834000000	67
3	AAL	2021000000	2014-12-31	4.120400e+10	7.00000e+06	4.322500e+10	2882000000	143
4	AAL	5635000000	2015-12-31	4.278000e+10	6.00000e+06	4.841500e+10	7610000000	135
5	AAP	1210694000	2012-12-29	3.403120e+09	7.00000e+03	4.613814e+09	387670000	32
6	AAP	1516205000	2013-12-28	4.048569e+09	7.00000e+03	5.564774e+09	391758000	26
7	AAP	2002912000	2015-01-03	5.959446e+09	7.00000e+03	7.962358e+09	493825000	25
8	AAP	2460648000	2016-01-02	5.673917e+09	7.00000e+03	8.134565e+09	473398000	19
9	AAPL	123549000000	2013-09-28	8.345100e+10	1.97640e+10	2.070000e+11	37037000000	30
10	AAPL	111547000000	2014-09-27	1.202920e+11	2.33130e+10	2.318390e+11	39510000000	35

Figure 5.

Response variable

Y= TE : Total equity of the company

Explanatory Variables

X1= NI : Net Income
X2= TL : Total Liabilities
X3= CS : Common Stocks
X4= TA : Total Assets
X5= ROE: return on earned

```
Call:
lm(formula = TE ~ +TL + CS + TA + NI + ROE, data = df1)

Residuals:
       Min         1Q     Median         3Q        Max
-2.637e+09  1.199e+07  1.314e+07  1.431e+07  8.082e+07

Coefficients:
              Estimate Std. Error   t value Pr(>|t|)
(Intercept) -1.293e+07  4.380e+06    -2.952   0.0032 **
TL          -9.994e-01  4.045e-04 -2470.986   <2e-16 ***
CS           1.638e-04  4.652e-04     0.352   0.7248
TA           9.994e-01  3.724e-04  2683.971   <2e-16 ***
NI           2.185e-03  1.461e-03     1.496   0.1350
ROE          3.638e+03  2.015e+04     0.181   0.8568
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 153800000 on 1628 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 9.838e+06 on 5 and 1628 DF,  p-value: < 2.2e-16
```

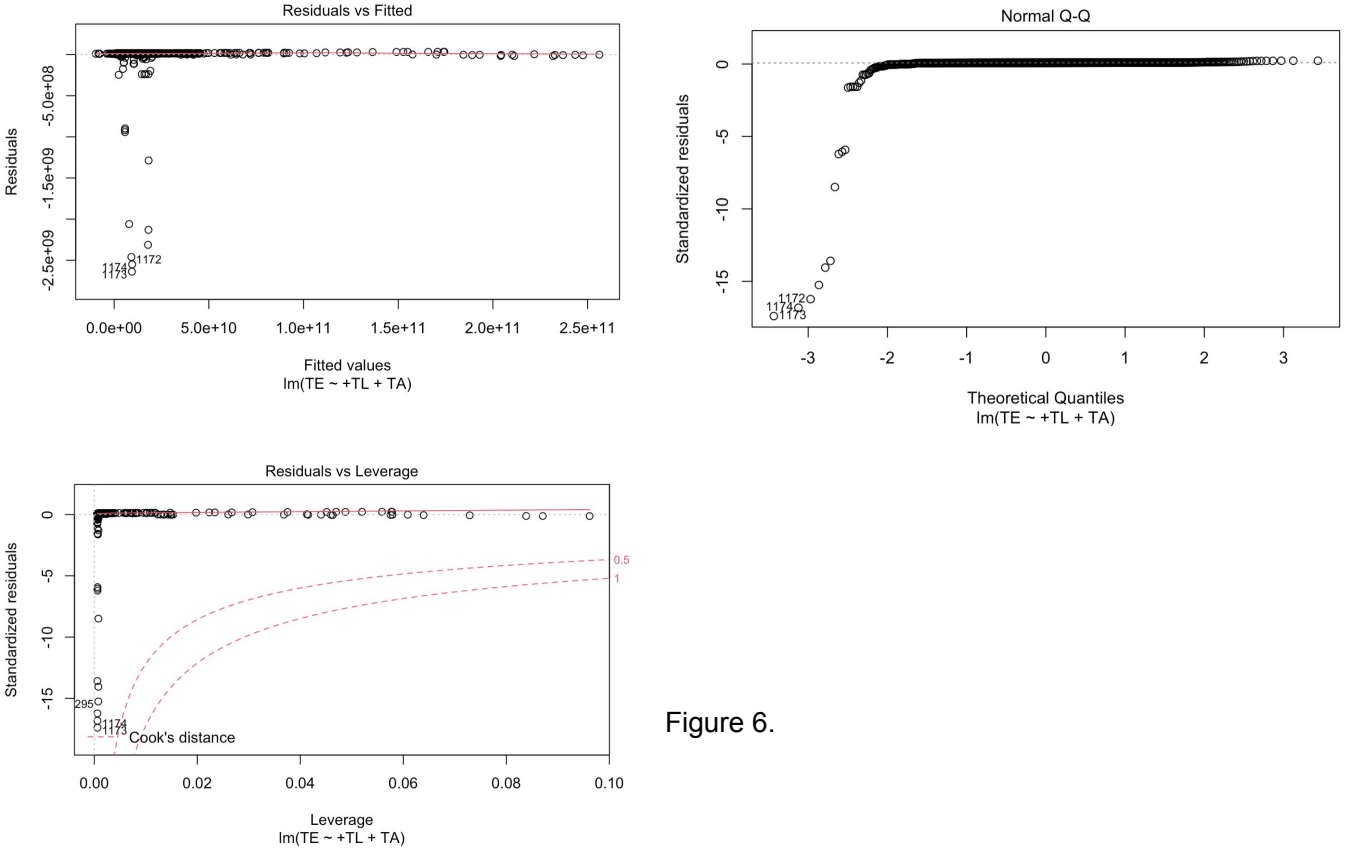


Figure 6.

Methods

The time series model ARIMA for data analysis and stock price forecasting. The formula is as follows:

$$Y_t = \alpha_0 + \beta_1 Y_{t-1} + \ldots + \beta_p Y_{t-p} + \varepsilon_t + r_1 Y_{t-1} + r_q Y_{t-q}$$

Among them, Y_t is the stock price in time t. The p and q values are the order of the ARIMA (p, i, q) model. We determine the order of the ARIMA model based on the characteristics of the data, that is, the p value and the q value, and then estimate each β and r. Finally, we predict the future stock price data based on the fitted model and compare it with the real value to calculate the prediction error. We use mean square error (MSE) to characterize the prediction error. The smaller the MSE, the higher the prediction accuracy of the model.

Regression model analysis the model selection, multicollinearity, and cross validation:

$$\hat{TE} = \beta_0 + \beta_1 * NI + \beta_2 * TL + \beta_3 * CS + \beta_4 * TA + \beta_5 * ROE$$

Formula1.

Multicollinearity and Variable Selection

Recall the figure 6 we can see that TL and TA would be more effective for the model, so we can use the multicollinearity and variable selection method to choose a better regression model from all the variables. Therefore, we reduce the formula 1 to formula 2 in the following:

NI	TL	CS	TA	ROE
2.488477	398.127706	1.368659	422.540294	1.004491

	Model Selected	AIC
Forward	TE~NI+TL+TA+CS+ROE	63262.16
Backward	TE~ROE+CS	63260
Both	TE~NI+TL+TA	63258.36

Table1.

Reduce model

$$\hat{TE} = \beta_0 + \beta_1 * TA + \beta_2 * TL + \beta_3 * NI$$

Formula2.

Results

Because the basic assumption of the ARIMA model is sequence stationarity. First, we need to test whether the stock price data of Amazon and Wal-Mart meets sequence stationarity. According to the ADF unit root test, as shown in Figure 7, the p-value is greater than 0.05, and the null hypothesis cannot be rejected. Therefore, the stock price data of Wal-Mart and Amazon are not sequence stationary.

```
> adf.test(walmart)

        Augmented Dickey-Fuller Test

data:  walmart
Dickey-Fuller = -1.5694, Lag order = 12, p-value = 0.7606
alternative hypothesis: stationary

> adf.test(amazon)

        Augmented Dickey-Fuller Test

data:  amazon
Dickey-Fuller = -1.6485, Lag order = 12, p-value = 0.7271
alternative hypothesis: stationary
```

Figure 7

Return function

Then, we calculated the stock price return based on the stock price, the formula is as follows:

$$Return_t = \frac{P_t}{P_{t-1}} - 1$$

Among them, P_t is the stock price in time t. Based on the stock return data, we have drawn time series diagrams of Amazon and Walmart, as shown in Figure 8 and Figure 9. The figure shows that the two stock return data fluctuate around 0, indicating that the sequence is close to stationary. To prove it statistically, we also calculated the ADF unit root test. The results are shown in Figure 8. All P values are less than 0.05, indicating that the null hypothesis is rejected. Therefore, the stock returns of Amazon and Wal-Mart are sequence stationary.

Then, we combine the ACF diagram and the PACF diagram to determine the order of the ARIMA model. At the same time, we also use the auto.arima command to calculate the smallest AIC value to determine the final model. In the end, we all chose the ARIMA(0,1,1) model to predict the prices of the two stocks. We divide the data into two parts, including the training group and the test group. The test group data includes stock price data for the last 30 trading days, while the training group includes all other data. We use the ARIMA (0, 1, 1) model to predict the stock price of Wal-Mart and Amazon respectively, as shown in Figure 11 and Figure 12. Among them, the red line represents the predicted value, the green line represents the true value, and the blue line represents the 95% prediction confidence interval. As shown in the figure, all the true values fall within the prediction interval, indicating that the ARIMA model performs well in prediction. Furthermore, we also calculated the forecast error MSE of the two stocks separately. As Wal-Mart's average stock price is lower than Amazon's, the final MSE is also lower.

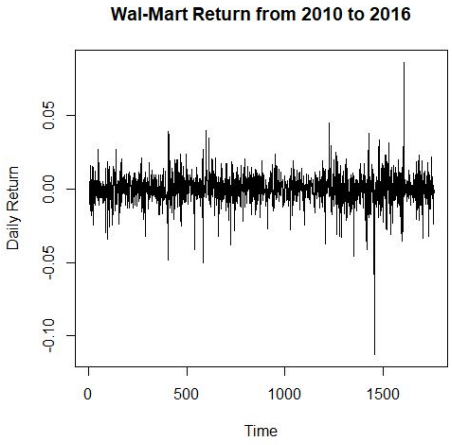


Figure 8

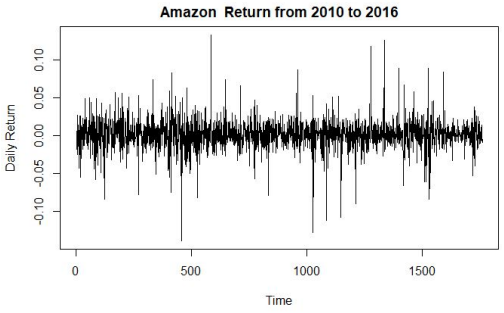


Figure 9

```
> adf.test(r_walmart)

        Augmented Dickey-Fuller Test

data:  r_walmart
Dickey-Fuller = -11.586, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(r_walmart) : p-value smaller than printed p-value
> adf.test(r_amazon)

        Augmented Dickey-Fuller Test

data:  r_amazon
Dickey-Fuller = -12.494, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary
```

Figure 10

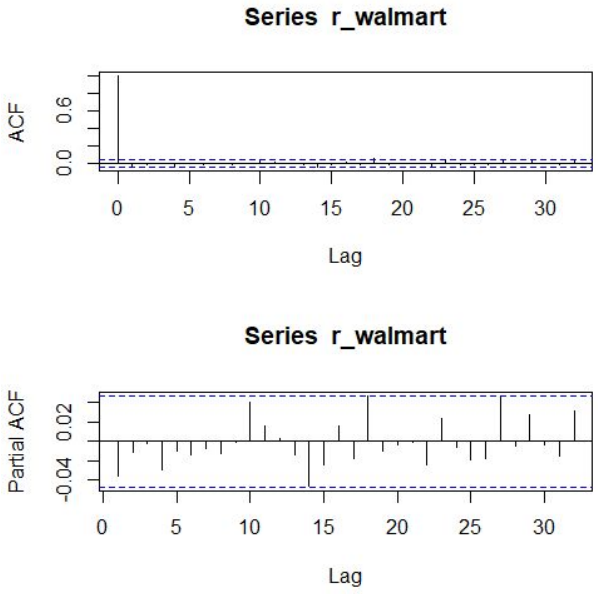


Figure 11

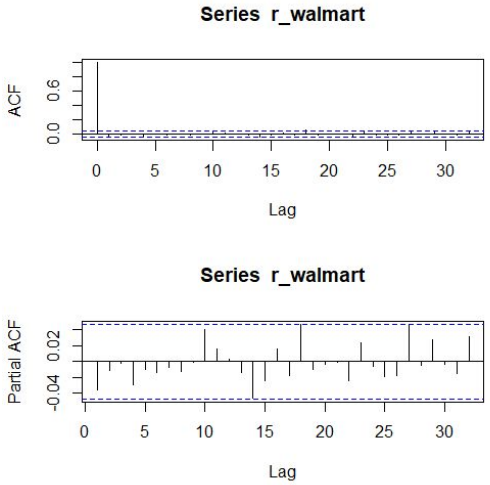


Figure 12

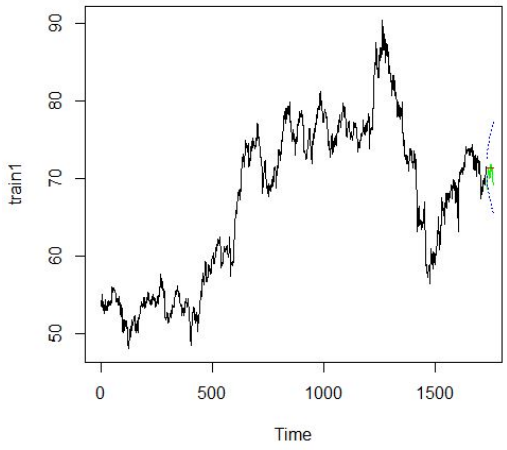


Figure 13 Wal-Mart

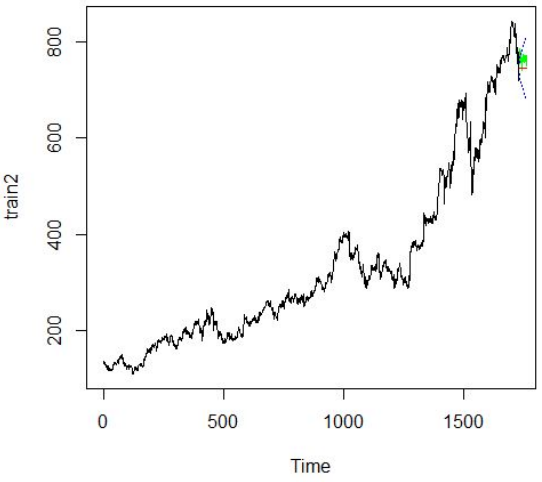


Figure 14 Amazon

Transformation

After the model selection, we can use it to define its transformation. In this case, we found the log method is more likely a better way to transfer the model. In the "residual vs fitted" it looks more linear compared to the figure 6 same as the Q-Q plot. At the same time, all of them have less outliers. Cook's distances are in the good range on the leverage plot.

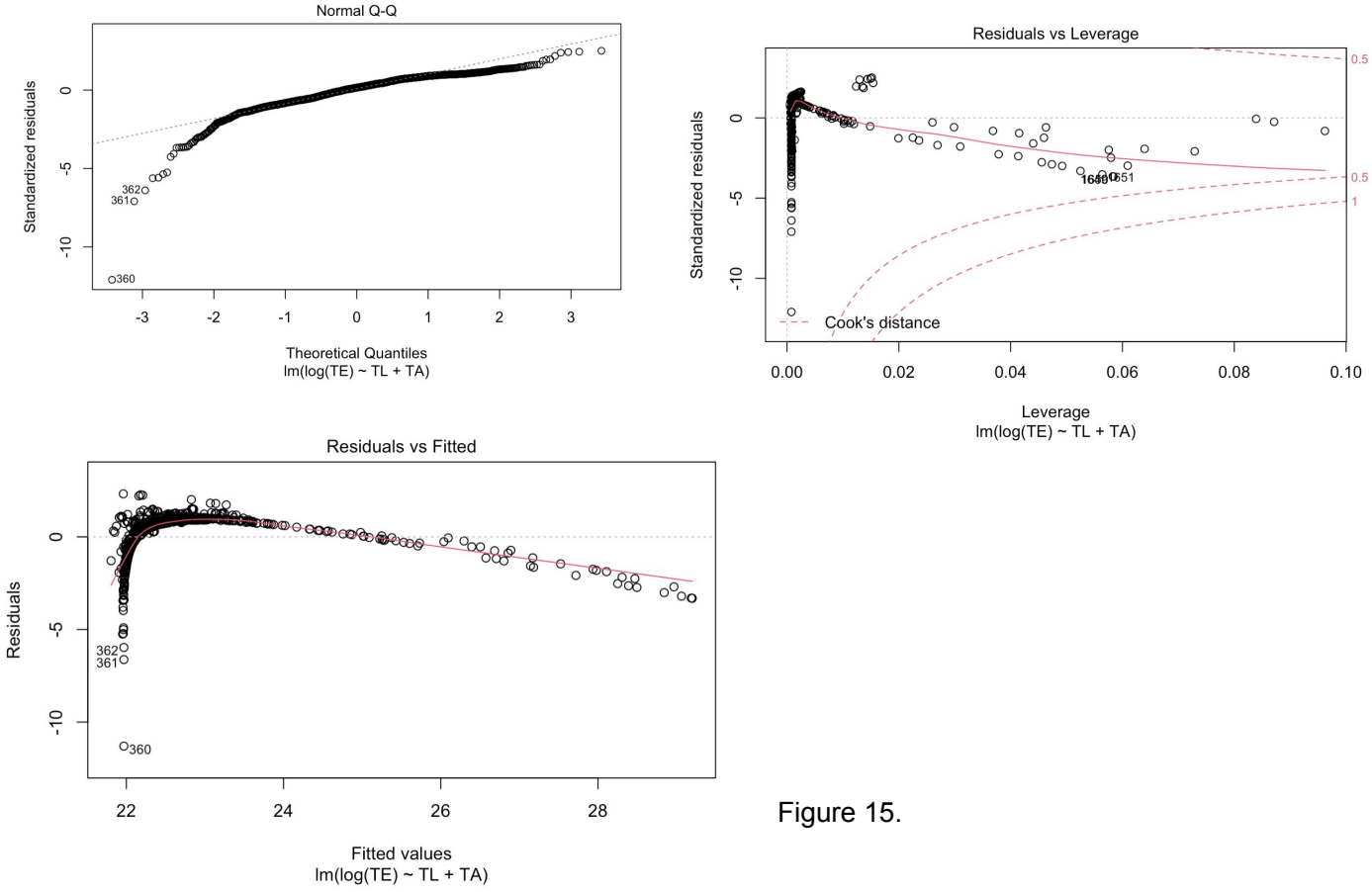


Figure 15.

Conclusion

The development of the Internet has promoted the expansion of the online retail industry. Amazon and Wal-Mart are examples of the Internet retail industry. We use the stock prices of these two companies for analysis and forecasting. We selected daily stock closing price data from 2010 to 2016 for analysis. According to the ACF chart and the PACF chart, we use the ARIMA(0,1,1) model to fit the stock price data and predict the 30-period data. The results show that all the true values fall within the 95% prediction confidence interval, indicating that the prediction performance of the model in this paper is good.

The performance after the all the test from all the methods like the multicollinearity , transformation, and variable selection. We found out the Equity is quite efficient by the assets of the company and the liabilities. However, we couldn't analyze the correlation between total equity and stock price of each company because there are only 4 years of period in the data. It would have a better performance if there is a more continuous time line so the model would be more precise.

Final model

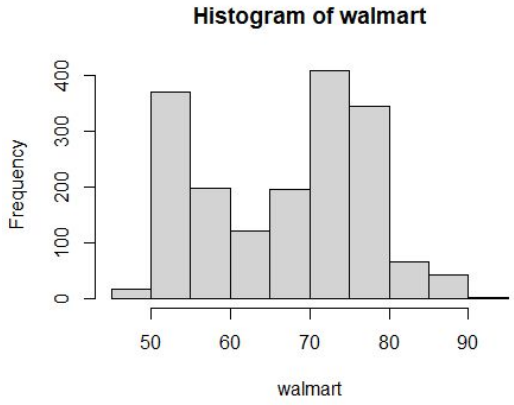
$$\hat{TE} = \beta_0 + (9.995e-01) * TA + (-9.994e-01) * TL + (2.206e-03) * TA$$

Appendix

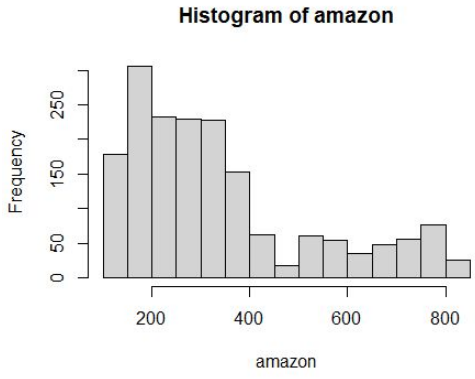
##R code

```
head(prices)

##                 date symbol   open  close    low   high  volume
## 1 2016-01-05 00:00:00   WLTW 123.43 125.84 122.31 126.25 2163600
## 2 2016-01-06 00:00:00   WLTW 125.24 119.98 119.94 125.54 2386400
## 3 2016-01-07 00:00:00   WLTW 116.38 114.95 114.93 119.74 2489500
## 4 2016-01-08 00:00:00   WLTW 115.48 116.62 113.50 117.44 2006300
## 5 2016-01-11 00:00:00   WLTW 117.01 114.97 114.09 117.33 1408600
## 6 2016-01-12 00:00:00   WLTW 115.51 115.55 114.50 116.06 1098000


head(walmart)

##            date symbol  open close   low  high    volume
## 702  2010-01-04    WMT 53.74 54.23 53.67 54.67 20753100
## 1170 2010-01-05    WMT 54.09 53.69 53.57 54.19 15648400
## 1638 2010-01-06    WMT 53.50 53.57 53.42 53.83 12517200
## 2106 2010-01-07    WMT 53.72 53.60 53.26 53.75 10662700
## 2574 2010-01-08    WMT 53.43 53.33 53.02 53.53 11363200
## 3042 2010-01-11    WMT 53.33 54.21 53.10 54.44 13987700

head(amazon)

##            date symbol   open  close    low   high   volume
## 285  2010-01-04   AMZN 136.25 133.90 133.14 136.61  7599900
## 752  2010-01-05   AMZN 133.43 134.69 131.81 135.48  8851900
## 1220 2010-01-06   AMZN 134.60 132.25 131.65 134.73  7178800
```

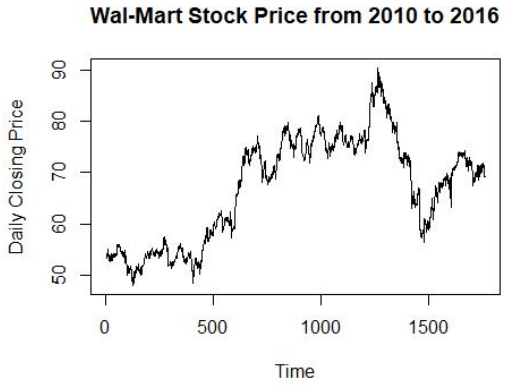
```
## 1688 2010-01-07   AMZN 132.01 130.00 128.80 132.32 11030200
## 2156 2010-01-08   AMZN 130.56 133.52 129.03 133.68  9830500
## 2624 2010-01-11   AMZN 132.62 130.31 129.21 132.80  8779400


walmart<-ts(walmart$close)
amazon<-ts(amazon$close)

describe(walmart)

##    vars    n  mean    sd median trimmed   mad min   max range  skew kurtosis
## X1    1 1762 67.04 10.27  69.56   67.07 11.79  48 90.47 42.47 -0.14    -1.22
##      se
## X1 0.24

describe(amazon)

##    vars    n  mean     sd median trimmed    mad    min    max  range skew
## X1    1 1762 337.9 189.11 282.92  312.14 142.43 108.61 844.36 735.75 1.09
##    kurtosis   se
## X1     0.14 4.51

hist(walmart)
```



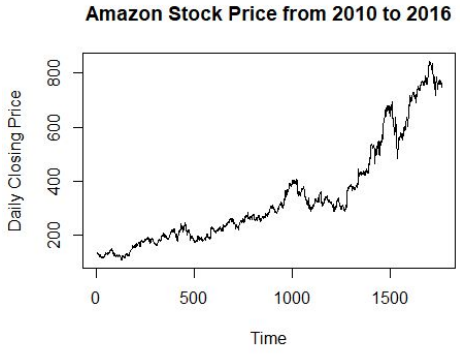

```
hist(amazon)
```




```
plot(walmart,ylab="Daily Closing Price", main="Wal-Mart Stock Price from 2010 to 2016")
```




```
plot(amazon,ylab="Daily Closing Price", main="Amazon Stock Price from 2010 to 2016")
```




```
adf.test(walmart)
```

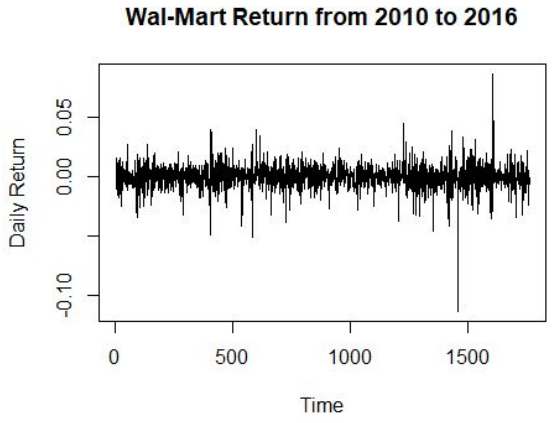
```
##
##  Augmented Dickey-Fuller Test
##
## data:  walmart
## Dickey-Fuller = -1.5694, Lag order = 12, p-value = 0.7606
## alternative hypothesis: stationary
```

```
adf.test(amazon)
```

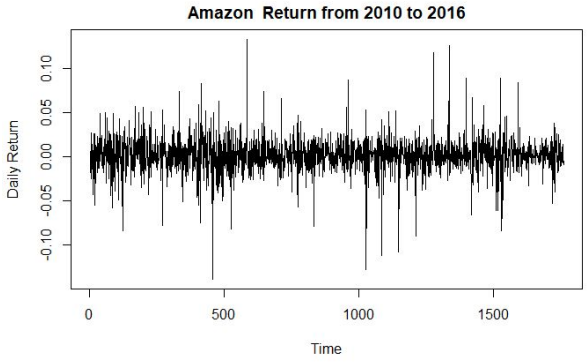
```
##
##  Augmented Dickey-Fuller Test
##
## data:  amazon
## Dickey-Fuller = -1.6485, Lag order = 12, p-value = 0.7271
## alternative hypothesis: stationary
```

```
r_walmart <- diff(walmart )/lag(walmart )
r_amazon<- diff(amazon)/lag(amazon)
```

```
plot(r_walmart,ylab="Daily Return", main="Wal-Mart Return from 2010 to 2016")
```




```
plot(r_walmart,ylab="Daily Return", main="Amazon  Return from 2010 to 2016")
```




```
adf.test(r_walmart)
```

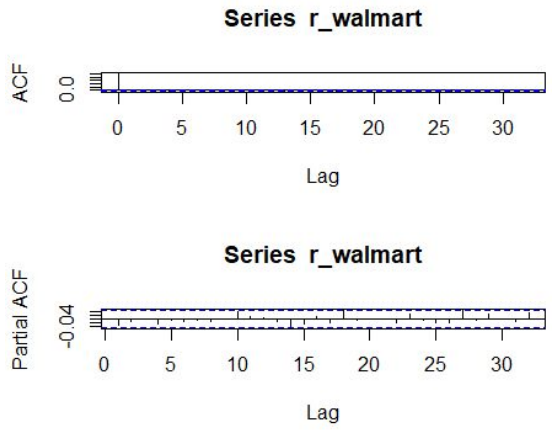
```
## Warning in adf.test(r_walmart): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  r_walmart
## Dickey-Fuller = -11.586, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(r_amazon)
```

```
## Warning in adf.test(r_amazon): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  r_amazon
## Dickey-Fuller = -12.494, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

```
par(mfrow=c(2,1))
acf(r_walmart)
pacf(r_walmart)

par(mfrow=c(2,1))
```

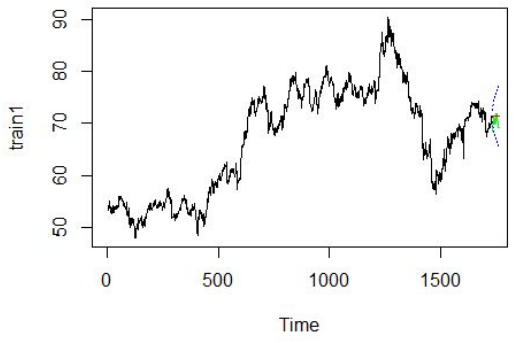
```
acf(r_walmart)
pacf(r_walmart)
```




```
train1<-walmart[1:(length(walmart)-30)]
test1<-walmart[(length(walmart)-29):length(walmart)]

train2<-amazon[1:(length(amazon)-30)]
test2<-amazon[(length(amazon)-29):length(amazon)]

auto.arima(train1,trace = TRUE)
```

```
      ##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2) with drift         : 3625.589
##  ARIMA(0,1,0) with drift         : 3619.188
##  ARIMA(1,1,0) with drift         : 3618.856
##  ARIMA(0,1,1) with drift         : 3618.445
##  ARIMA(0,1,0)                    : 3617.544
##  ARIMA(1,1,1) with drift         : 3620.597
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,1,0)                    : 3616.794
##
##  Best model: ARIMA(0,1,0)

      ## Series: train1
## ARIMA(0,1,0)
##
## sigma^2 estimated as 0.4726:  log likelihood=-1807.4
## AIC=3616.79   AICc=3616.79   BIC=3622.25

      fit1<-arima(train1,order=c(0,1,1))
fit1

      ##
## Call:
## arima(x = train1, order = c(0, 1, 1))
##
## Coefficients:
```

```
##          ma1
##      -0.0400
## s.e.  0.0242
##
## sigma^2 estimated as 0.4718:  log likelihood = -1806.04,  aic = 3616.07


fore<-predict(fit1, n.ahead=30)
fore

      ## $pred
## Time Series:
## Start = 1733
## End = 1762
## Frequency = 1
##  [1] 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976
##  [9] 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976
## [17] 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976
## [25] 71.38976 71.38976 71.38976 71.38976 71.38976 71.38976
##
## $se
## Time Series:
## Start = 1733
## End = 1762
## Frequency = 1
##  [1] 0.6868839 0.9521838 1.1582308 1.3327952 1.4870063 1.6266629 1.7552426
##  [8] 1.8750257 1.9876030 2.0941371 2.1955079 2.2924003 2.3853603 2.4748310
## [15] 2.5611780 2.6447074 2.7256783 2.8043121 2.8808004 2.9553097 3.0279861
## [22] 3.0989586 3.1683417 3.2362376 3.3027380 3.3679256 3.4318752 3.4946547
## [29] 3.5563262 3.6169463



par(mfrow=c(1,1))
ts.plot(train1)
real<-ts(test1,start=length(train1)+1,end=length(walmart))
lines(real,col="green")
lines(fore$pred,col="red")
lines(U,col="blue", lty=3)
lines(L,col="blue", lty=3)
```

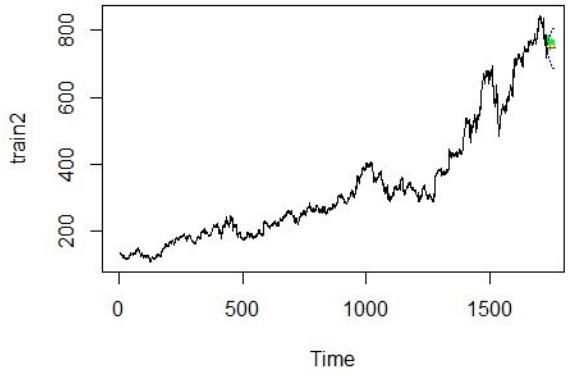



```
MSE1<-sum((fore$pred-real)^2)/length(real)
MSE1
```

```
      ## [1] 1.720646

auto.arima(train2,trace = TRUE)

      ##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2) with drift         : 11748.98
##  ARIMA(0,1,0) with drift         : 11739.1
##  ARIMA(1,1,0) with drift         : 11742.1
##  ARIMA(0,1,1) with drift         : 11741.1
##  ARIMA(0,1,0)                    : 11741.29
##  ARIMA(1,1,1) with drift         : 11744.11
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,1,0) with drift         : 11743.04
##
##  Best model: ARIMA(0,1,0) with drift

      ## Series: train2
## ARIMA(0,1,0) with drift
##
## Coefficients:
##         drift
##        0.3539
## s.e.  0.1727
##
## sigma^2 estimated as 51.64:  log likelihood=-5869.52
## AIC=11743.04   AICc=11743.04   BIC=11753.95

      fit2<-arima(train2,order=c(0,1,1))
fit2

      ##
## Call:
## arima(x = train2, order = c(0, 1, 1))
##
## Coefficients:
##          ma1
##       0.0023
## s.e.  0.0240
##
## sigma^2 estimated as 51.74:  log likelihood = -5871.61,  aic = 11747.22


fore<-predict(fit2, n.ahead=30)
fore

      ## $pred
## Time Series:
## Start = 1733
## End = 1762
## Frequency = 1
##  [1] 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974
##  [9] 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974
## [17] 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974
## [25] 746.4974 746.4974 746.4974 746.4974 746.4974 746.4974
```

```
##
## $se
## Time Series:
## Start = 1733
## End = 1762
## Frequency = 1
##  [1]  7.19291 10.18411 12.47774 14.41083 16.11367 17.65300 19.06847 20.38589
##  [9] 21.62319 22.79343 23.90646 24.96992 25.98990 26.97133 27.91829 28.83416
## [17] 29.72183 30.58374 31.42202 32.23851 33.03482 33.81239 34.57247 35.31619
## [25] 36.04458 36.75853 37.45888 38.14637 38.82169 39.48547
```

```
lines(real,col="green")
lines(fore$pred,col="red")
lines(U,col="blue", lty=3)
lines(L,col="blue", lty=3)
```




```
MSE2<-sum((fore$pred-real)^2)/length(real)
MSE2
```

```
## [1] 450.5341
```