Shu-Ya Chiang

The data of the stock market is often used to analyze time-series because there are investors who want to profit from the investment and forecasting is very important to them. Even the stock market is hard to predict; there are still some algorithms of machine learning that can gather information from the past pattern to forecast the future. Time series analysis is used to predict the short-term stock market. We can use our prediction model to make assumptions on market price. Although we can use time series to predict the future stock trend, the prediction is not safe to apply on the real market.

Dataset:

Source from Kaggle:
https://www.kaggle.com/szrlee/stock-time-series-20050101-to-20171231

Data type / Variables:
This whole dataset contains 20 companies and 12 years of stock data, but we will focus on Google['GOOGL'] to do time series analysis.
There are 3019 observations and 7 variables:
Date (Date) from 2006-01-03 to 2017-12-29
Open (Numeric) is the opening price in a trading day
High((Numeric) is the highest stock price in a trading day
Low (Numeric) is the lowest stock price in a trading day
Close(Numeric) is the closing price in a trading day
Volume(Numeric) means the number of shares is traded in a trading day
Name (String) means stock ticker
The trading days are usually from Monday to Friday, excluding holidays.

Introduction

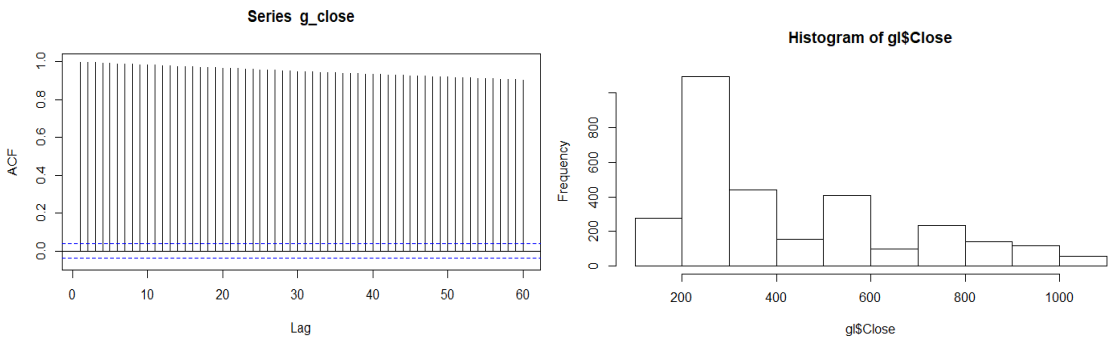
The goal is to forecast the closing stock price using existing dataset. The dataset named 'S&P 500 stock data' is from Kaggle. The original source of the dataset is scrapped from google finance. The data will be preprocessed, conducted exploration analysis and visualization, and built several types of models to evaluate the forecasting results.

Data exploratory:
After using *is.na() function*, there's no missing value. The frequency in each year is different.
There are 251 observations in 2010 and 252 observations in 2012.
There are 253 observations in 2006, 2007, 2009, 2013, 2015, and 2016.
There are 254 observations in 2017.
There are 255 observations in 2008 and 2011.
There are 256 observations in 2014.

The five-number summary of Close shows that the minimum is 128.8, 1st quartile is 247.6, the median is 310.1, and the mean is 428, 3rd quartile is 570.8, and the max is 1085.1.

4 price time plots:

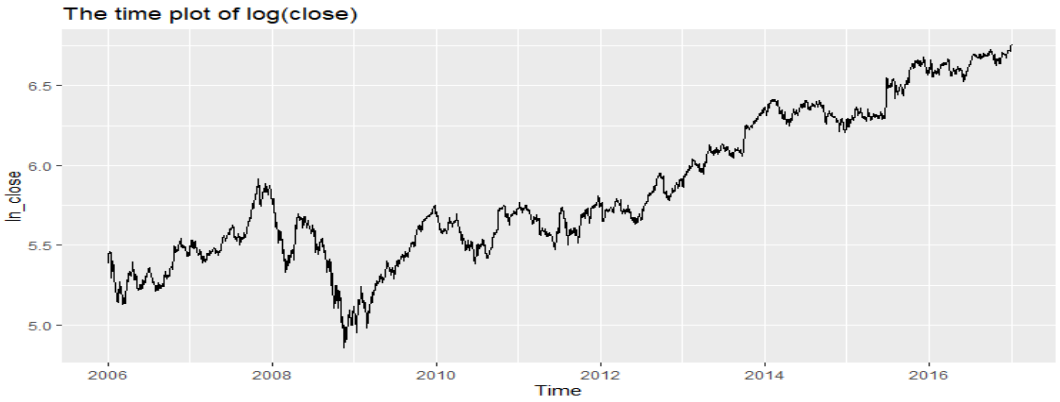
After compare 4 price plots, since all plot has the same trends we decided to use close price as our primary target, for the rest of the project close price will be our predict value. Transforming our dataset of closing price to a time-series data, we decided to set frequency equal to 253 and start with 2006 and end in 2017. In the time series plot of closing price, there're fluctuations. The closing price decreased sharply between 2008 and 2009, which was the great recession. There is an uptrend after 2012. It's not stationary because the trends don't move back to the mean where the closing price is 428. The autocorrelation plot shows that the series is non-stationary because the lags decay very slow and they are above the blue dotted line and also far from zero.

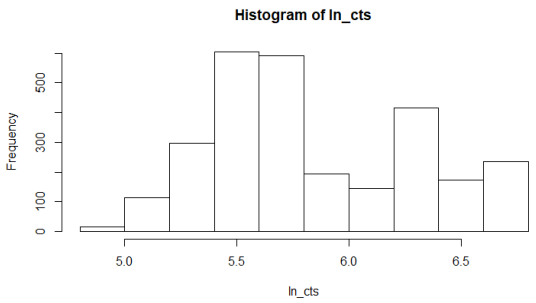


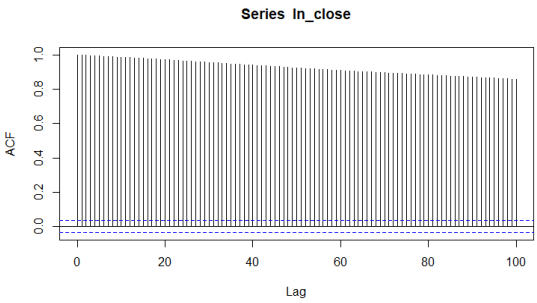
```
        D'Agostino skewness test

data:  g_close
skew = 0.902, z = 16.830, p-value < 2.2e-16
alternative hypothesis: data have a skewness
```

The histogram of the closing price is a right-skewed distribution, which is not normal. We also use D'Agostino Skewness Test to check normality, and if the p-value of this test is greater than 0.05, it means data don't have skewness. However, the result of the D'Agostino Skewness Test states that data have skewness. Now, both results prove the data aren't normal.

Next, we decided to use log transformation to smooth the data. In the time plot of log(close), the uptrend from 2010 to 2017 is smoother than the series before the transformation. The summary of min, first quartile, median, mean, third quartile, and the max of log(close) is 4.859, 5.492, 5.709, 5.841, 6.294, and 6.755. However, the inflections in the time plot are not around 5.709, which is the value of the mean. The autocorrelation plot of time series of log(close) shows that the series is non-stationary because the lags decay very slow and a hundred lags are above the blue dotted lines. The histogram is still not normal distributed.







KPSS Test for Trend Stationarity and Level Stationarity

data: g_close KPSS Trend = 5.4608, Truncation lag parameter = 9, p-value = 0.01

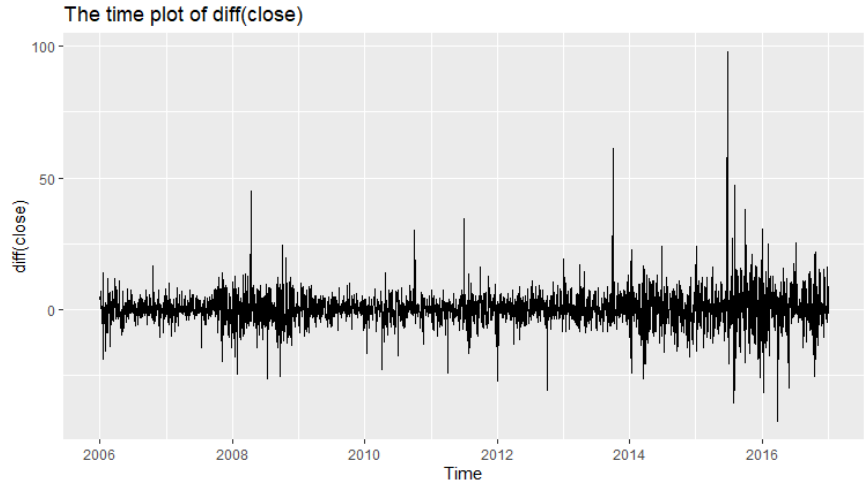
data: g_close KPSS Level = 23.052, Truncation lag parameter = 9, p-value = 0.01

data: ln_cts KPSS Trend = 3.6502, Truncation lag parameter = 9, p-value = 0.01

data: ln_cts KPSS Level = 23.825, Truncation lag parameter = 9, p-value = 0.01

In the result of the KPSS test, regardless of the original closing price or log-transformed closing price, the p-values are all less than 0.05, which means data are not stationary.

Next, we created a time plot by using the first-order differencing of the closing price.



There are fluctuations but it's stationary since trends usually move back to zero.

KPSS Test for Trend Stationarity and Level Stationarity

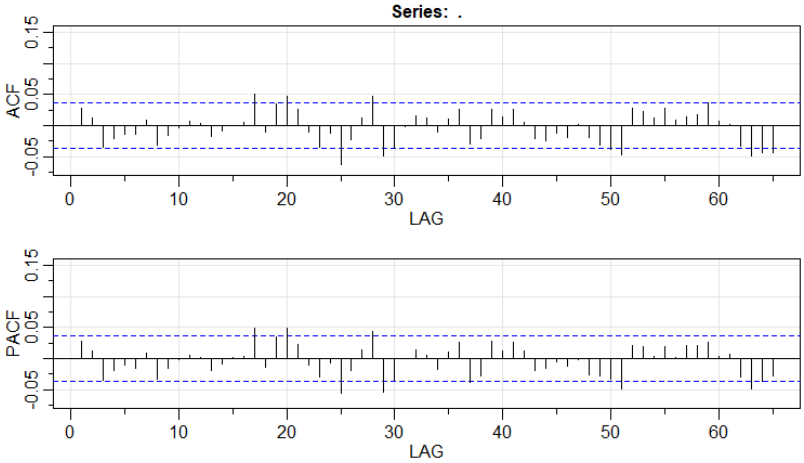
data: diff1_ts

KPSS Trend = 0.020922, Truncation lag parameter = 9, p-value = 0.1

KPSS Level = 0.2684, Truncation lag parameter = 9, p-value = 0.1

After we used the KPSS test to test trend and level stationary, both p-values are 0.1, which means that the first-order differencing of this time series is stationary.

ACF plot and PACF plot of the first order differencing of the closing price:



The ACF and PACF plot show that there's no huge spike lag. The first 10 lags are all below the blue dotted lines.

Auto ARIMA

Applying auto Arima on the whole dataset and will return the best ARIMA model based on either AIC, AICc or BIC value. I set the information criterion as BIC, so the best ARIMA model will have the lowest BIC.

> fit<-auto.arima(g_close, trace=TRUE, allowmean=FALSE, allowdrift = FALSE, max.p=1,

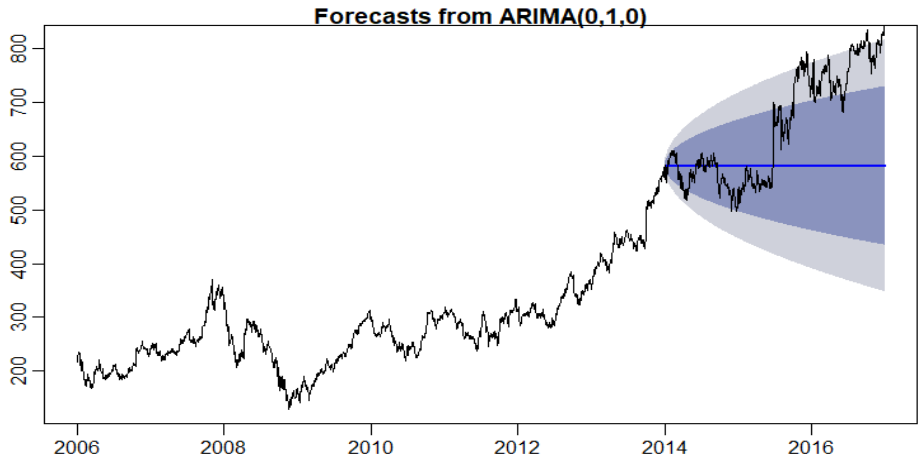
+ max.q=1, max.P=1, max.Q=1,max.d=1,max.D=1, ic="bic")

```
ARIMA(1,1,1)(1,0,1)[253]                : Inf
ARIMA(0,1,0)                            : 18466.27
ARIMA(1,1,0)(1,0,0)[253]                : Inf
ARIMA(0,1,1)(0,0,1)[253]                : Inf
ARIMA(0,1,0)(1,0,0)[253]                : Inf
ARIMA(0,1,0)(0,0,1)[253]                : 18474.17
ARIMA(0,1,0)(1,0,1)[253]                : Inf
ARIMA(1,1,0)                            : 18472.92
ARIMA(0,1,1)                            : 18472.51
ARIMA(1,1,1)                            : 18480.77
```

The auto.arima shows that the best model is ARIMA (0,1,0) that the AIC of the model is 18464.13 and BIC is 18470.07.

> backtest(fit,g_close,2100,1)
[1] "RMSE of out-of-sample forecasts"
[1] 9.624985
[1] "Mean absolute error of out-of-sample forecasts"
[1] 6.638275

Next, forecasting by using window to split the data to training and test.
xtr= window(g_close, start=2006, end=2014)
xtest=window(g_close,start=2014)
> fit2 <-auto.arima(xtr, ic=c("bic"))
Series: xtr
ARIMA(0,1,0)
sigma^2 estimated as 26.77: log likelihood=-6198.53
AIC=12399.06 AICc=12399.06 BIC=12404.67



SARIMA

Seasonal Autoregressive Integrated Moving Average (Seasonal ARIMA) supports time-series data with a seasonal component.
Applying SARIMA on the whole data set, the results as following:
sa1 = sarima(g_close, p=0, d=1, q=0, no.constant = TRUE)
sigma^2 estimated as 44.52: log likelihood = -9231.07, aic = 18464.13
$AIC [1] 6.63461 $AICc [1] 6.634615 $BIC [1] 6.636746

The standard residuals show that most data points are above blue dotted line in the p-value for Ljung-Box Statistic. There is no spike lag in the ACF of residuals plot.

sa2 = sarima(g_close, p=0, d=1, q=0,P = 0, D = 0, Q = 1,S=253, no.constant = TRUE)

sigma^2 estimated as 44.52: log likelihood = -9231.05, aic = 18466.11

$AIC [1] 6.635324 $AICc [1] 6.635325 $BIC[1] 6.639587

The standard residuals show that most data points aftter lag =100 are below blue dotted line in the p-value for Ljung-Box Statistic. There is no spike lag in the ACF of residuals plot.

sa3 = sarima(g_close, p=1, d=1, q=0, no.constant = TRUE)

sigma^2 estimated as 44.5: log likelihood = -9230.21, aic = 18464.42

$AIC [1] 6.634717 $AICc [1] 6.634718 $BIC[1] 6.63898

The standard residuals show that most data points are above blue dotted line in the p-value for Ljung-Box Statistic. There is no spike lag in the ACF of residuals plot.

sa4 = sarima(g_close, p=0, d=1, q=1, no.constant = TRUE)

sigma^2 estimated as 44.5: log likelihood = -9230.22, aic = 18464.45

$AIC [1] 6.634727 $AICc [1] 6.634728 $BIC[1] 6.63899

The standard residuals show that most data points are above the blue dotted line in the p-value for Ljung-Box Statistic. There is no lag above the blue dotted line in the ACF of residuals plot.

sa5 = sarima(g_close, p=1, d=1, q=1, no.constant = TRUE)

sigma^2 estimated as 44.5: log likelihood = -9230.2, aic = 18466.4

$AIC [1] 6.635428 $AICc[1] 6.635429 $BIC [1] 6.641822

After training the models on the whole dataset and comparing aic of the models, sarima(g_close, p=0, d=1, q=0) has the lowest aic.

Next, we split the data to training and test set. The training dataset is from 2006 up to 2014, which isn't including 2014. The test set is start in 2014.

```
xtr= window(g_close, start=2006, end=2014)
xtest=window(g_close,start=2014)
```

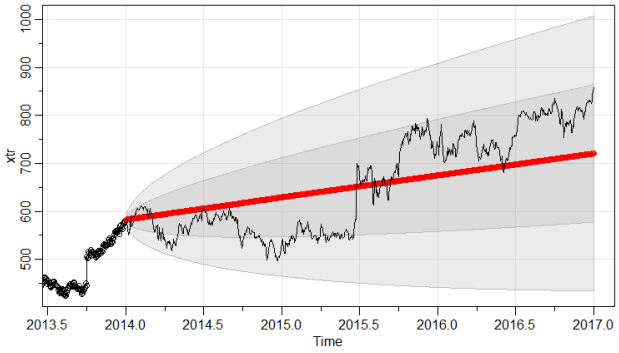
Then, we compared five models' performance by RMSE. the fc_s2 has the lowest RMSE.

```
fc_s1=sarima.for(xtr,759, p=0, d=1, q=0)
> RMSE( fc_s1$pred,xtest)   = 70.98797
fc_s2=sarima.for(xtr,759, p=0, d=1, q=0,P = 0, D = 0, Q = 1,S=253)
> RMSE( fc_s2$pred,xtest)  =  70.7973
fc_s3=sarima.for(xtr,759, p=1, d=1, q=0)
> RMSE( fc_s3$pred,xtest) =   71.05244
fc_s4=sarima.for(xtr,759, p=0, d=1, q=1)
> RMSE( fc_s4$pred,xtest)   = 70.99442
fc_s5=sarima.for(xtr,759, p=1, d=1, q=1)
> RMSE( fc_s5$pred,xtest)   = 71.06175
```

Therefore, the best model is sarima(xtr, p=0, d=1, q=0, P=0,D = 0, Q = 1,S=253)

Sma1 isn't significant.

```
Coefficients:
        sma1  constant
      0.0065    0.1803
s.e.  0.0225    0.1156

sigma^2 estimated as 26.73:  log likelihood = -6197.26,  aic = 12400.52

$degrees_of_freedom
[1] 2022

$ttable
         Estimate     SE t.value p.value
sma1       0.0065 0.0225  0.2867  0.7744
constant   0.1803 0.1156  1.5597  0.1190

$AIC
[1] 6.126739

$AICc
[1] 6.126741

$BIC
[1] 6.135058
```



The forecast didn't capture the fluctuation. It shows an uptrend, which is similar to the real data.