PSTAT 174 Final Project

S&P 500 Closing Values: Time Series Analysis

Authors: Ana Caklovic, Esther Hsu, Sophie Xu,

Arnold Wu, Hongshan Lin

Table of Contents

Abstract

Business has evolved from barter trade where goods were exchanged for goods in the past to the use of currency in the modern times. Today one does not necessarily need to be physically present during the exchange for the trade to be successful; the advancement of technology has fueled the global adoption of technology supported trades. One such instance is the stock market trading that continues to gain momentum world-over. Globally, stock traders in leading platforms depend on speculation to make trades in the hope that they will eventually earn profits. Many tools are available for such traders, and any other person interested in it, one being the market indices offered by leading platforms like the S&P 500, which is often assigned the symbol ^GSPC. It is important for traders to understand the behavior of markets if they have to make profits, which is why these indices are available to help them have better insight into the goings-on of the market under consideration.

To increase profit margins, business people involved in stock trading need to make winning trades; this is to say there speculation needs to be right most of the time. It is wise for traders to have as much insight of how the foreseeable future may look like by learning from present and past indices. For this project we are going to focus much on the market closing values for the 20 years beginning January 1997. The aim here is to use the available data to model a time series that will be used to forecast closing values of ^GSPC the last 10 months of 2017 and compare with the actual closing values recorded for that period at the index. This will be possible since we are going to examine the data for any seasonality and trend so we can remove them before coming up with a suitable model for the data and thereafter using AICc and BIC to find the best SARIMA model for the time series. We followed up with a diagnostic checking to check for any non-stationary remain. Finally, we made our forecast for March 1st to December 1st, 2017. Our predicted values are within the confidence level of 95% and approximately close to the true value in the original dataset.

Introduction

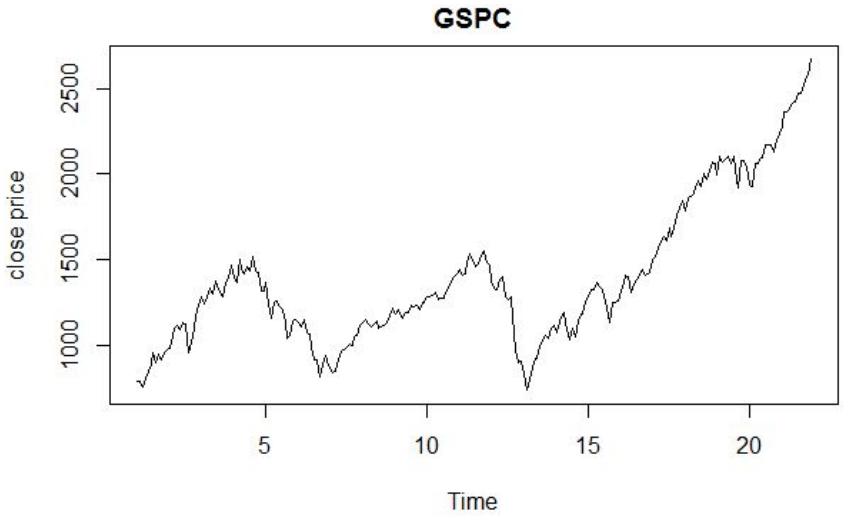
An investment becomes desirable only if it gives good returns in the long run, which is why traders find forecast information and historical data helpful in making trade decisions. It is of great importance for investors to have an idea how a time in the future may look or impact them. This information is useful as it plays a big role in the strategy employed for trading of stocks. We chose to work with market data because we found the behaviour of stock markets particularly interesting; a stock may record big values at the opening then plummet to a low mid-month and then close at an unexpected high. The data we will be working on contains 241 individual observations of the S&P 500 closing values, which is a good sample size to work with because the model developed based on it will be dependable.We plot the time series of the closing values as a first step in order to understand the behaviour of the time series, and possibly note the existence and nature of any unique components.

Data Exploratory Analysis

Preliminary Data Exploration

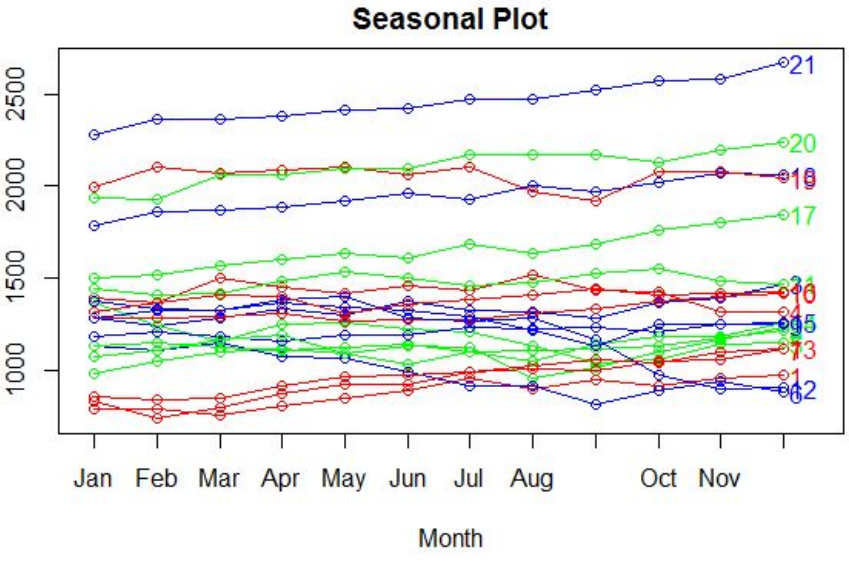
The data contains 241 monthly observations and we take out the last 10 observations for forecasting, so that we can examine our predicted values with actual observed values. We first plot a time series of the closing values by graphing the remaining 231 data points in our data set against time to formulate an idea about the data as in *Figure 1* below.

Figure 1



A quick examination of the initial plot reveals that the time series has a seasonal component, an upward trend, and a random component. We constructed a seasonal plot *(Figure 2)* to better understand the seasonality of the time series and how often it happens.

Figure 2



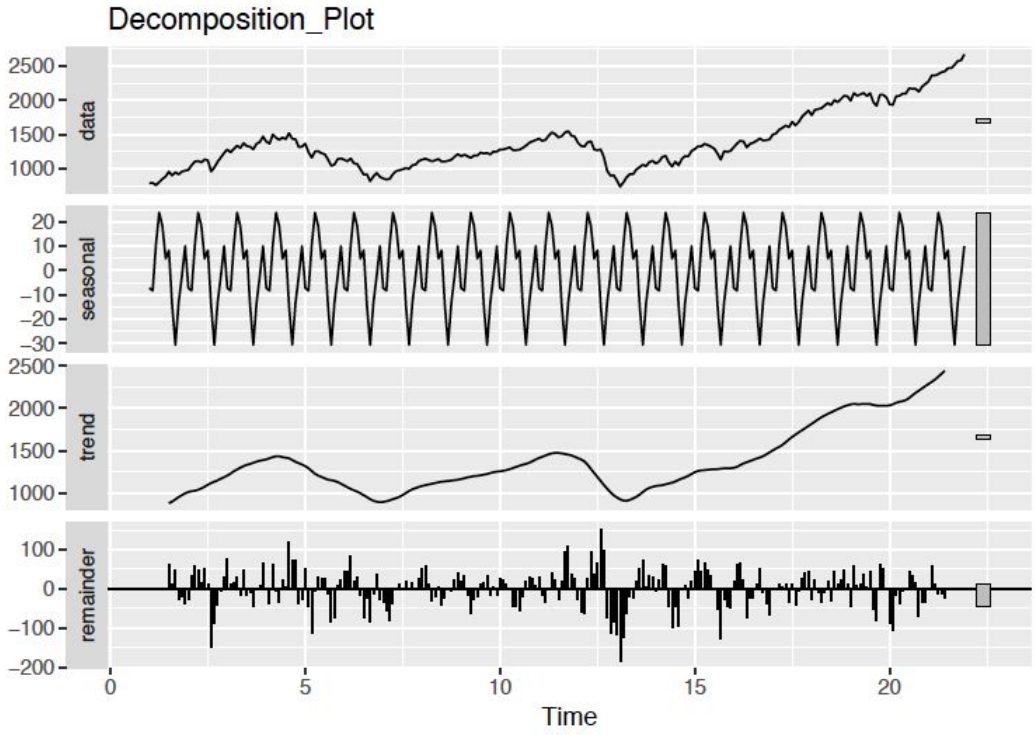
The seasonal plot above indicates a gradual increase from the beginning of the year to the end; values for December are generally higher than those recorded at the beginning of the year. Some

years recorded significantly lower closing values for the month of September compared to the other months. By looking at the plot, it is clear that our data will need to be made stationary before picking our model as the seasonality may strongly impact our model. To obtain a stationary time series for deciding our model, we need to determine the necessity of transformations and decompose the trend and seasonality.

Decomposition Model

To further examine if our data has seasonality and trend, we graphed the equation $Y_t=m_t+s_t+S_t$ where m_t represents trend, s_t represents seasonal component and S_t is a stationary process in a decomposition plot *(Figure 3)*.

Figure 3



The decomposition plot also implies that our data has annual seasonality and shows an upward trend.

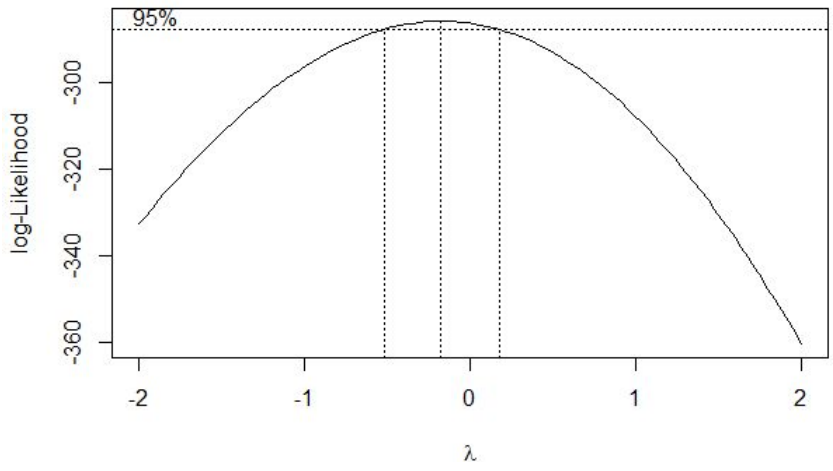
Based on *Figure 3*, we can conclude that our time series data is not stationary. Therefore, we will need to difference the time series at different lags to make the series stationary and determine if other transformations of the data should be applied.

Data Transformation

Stabilize Variance

We used the Box-Cox Transformation to find an appropriate lambda to transform our data and stabilize the variance. The lambda, found where the log-likelihood is highest, was calculated to be -0.1818182 as seen in the Box-Cox plot in *Figure 4*.

Figure 4



From the figure 6, we can see 0 is contained in the 95% confidence interval of the Box-Cox plot of the ^GSPC closing value (which allows for a log transformation), but we performed the box-cox transformation with the calculated -0.1818182 value.

As seen below in the ACF and PACF plots shown in *Figures 5* and *6* respectively, the time series data still shows signs of seasonality and trends. Thus, our next step will be to remove them from the data with the differencing method.

Figure 5

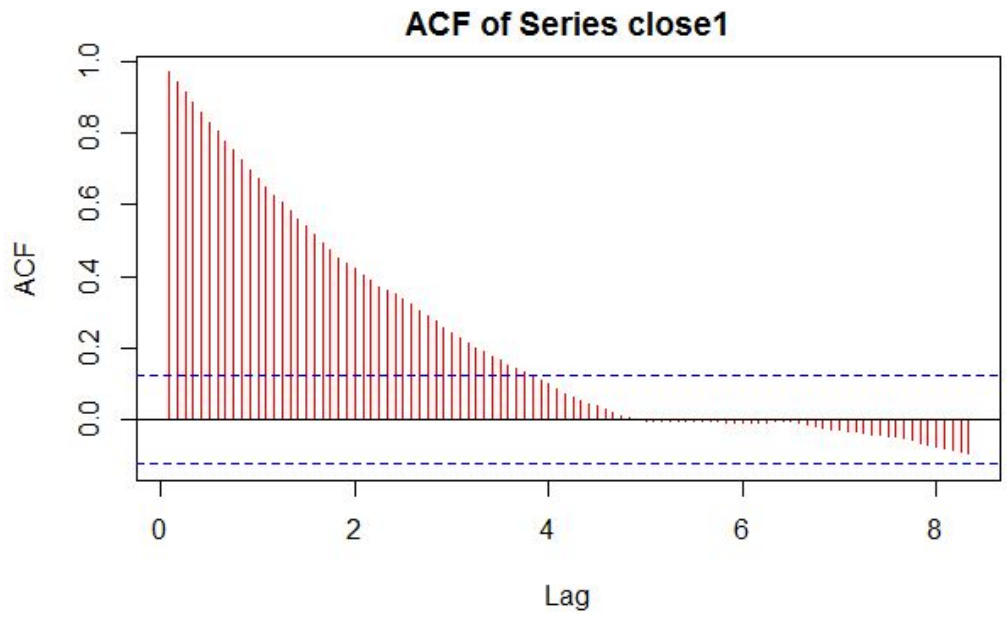
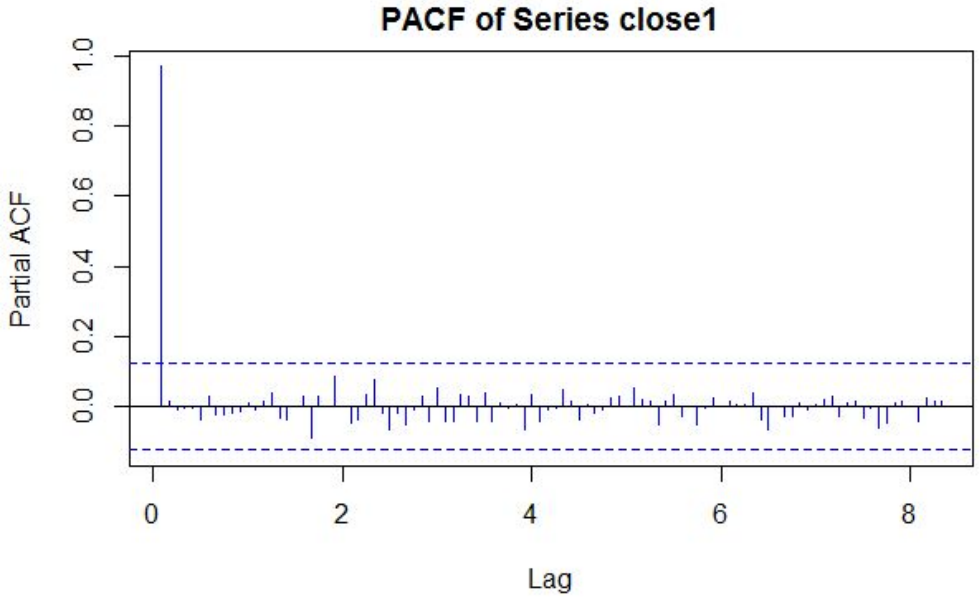


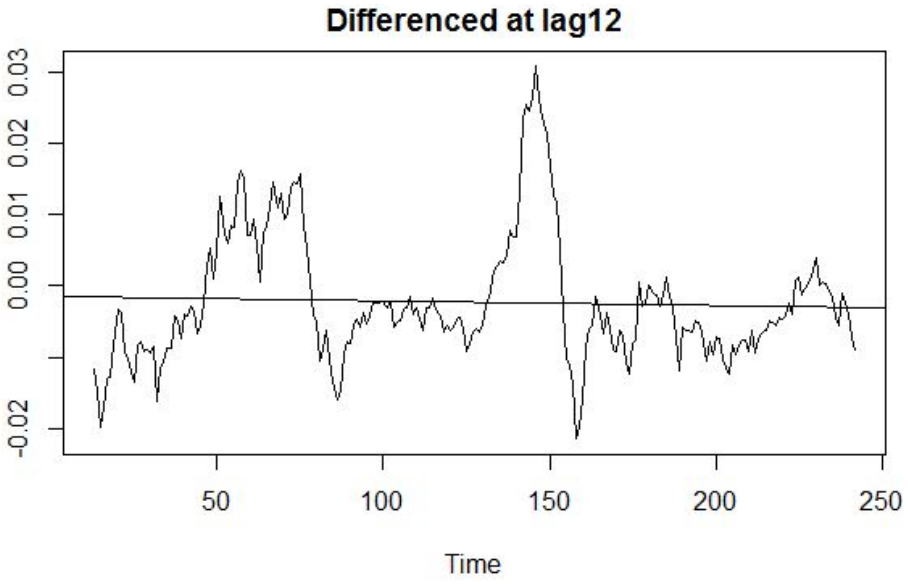
Figure 6



Remove Seasonality and Trend

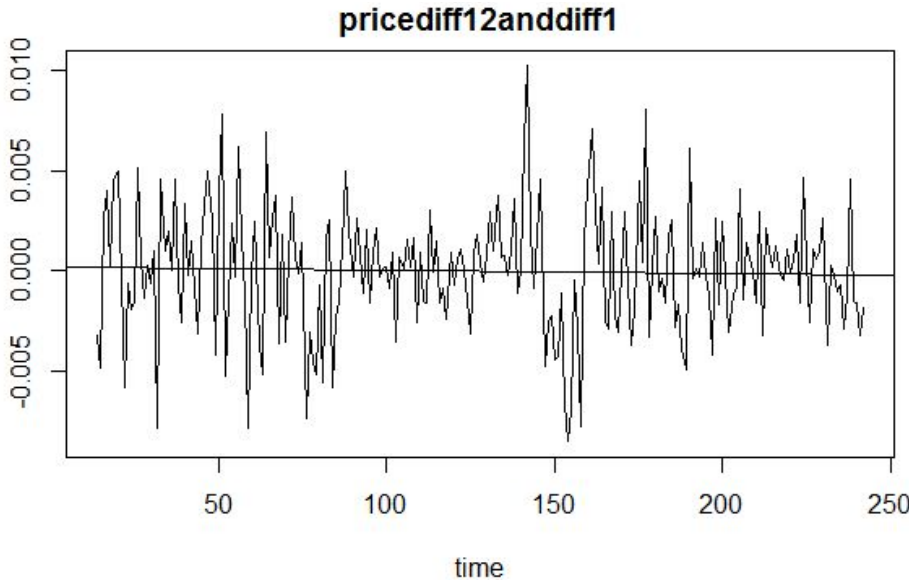
To remove seasonality and trend, we will use the differencing method where lag *d* differencing is defined as $\nabla_d Y_t = Y_t - Y_{t-d}$. We first difference the series at lag 12 to remove the seasonality of period 12 because the data recorded the monthly prices of the S&P 500. The variance of diff 12 is 8.088491e-06. From Graphing the non-seasonalized data (*Figure 7*), we see that the data still not stationary and has a downward trend as seen below. So, we need to persist in differencing the data.

Figure 7



We continue to difference at lag 1 as seen in *Figure 8* to get a relatively stationary time series with a variance of 9.41451e-06. We can observe that the trend line is horizontal, which indicates that we have removed the trend. But just to ensure that we differenced the data enough, once again we differenced at lag 1 to check if the time series is stationary. However this time, our calculated variance increased to 1.664501e-05, which is larger than the previous variance of 9.41451e-06 indicating that we are overdifferencing and should stop at the previous difference.

Figure 8



To verify stationarity of the time series, we performed the Augmented Dickey-Fuller test and got a p-value of 0.01, which is smaller than 0.05, so we reject the null hypothesis that the time series is not stationary and conclude with a 95% confidence interval that the transformed time series data is stationary. Next, we will perform the model identification and estimation based on the newly transformed data.

Model Identification and Estimation

Since our data recorded the monthly close price of the S&P 500, the first model we chose to analyze our data was the SARIMA model. The SARIMA model is the seasonal ARIMA model. The structure of the SARIMA is:

$$SARIMA(p,d,q) \times (P,D,Q)_s$$

(p, d, q) are the non-seasonal components: p = the order of non-seasonal AR process, d = non-seasonal differencing, q = the order of non-seasonal MA process. From the previous step we differenced data at lag 1 for detrending, so d is 1.

$(P,D,Q)_s$ are the seasonal components: P = the order of seasonal AR process, D = seasonal differencing, Q = the order of seasonal AR process and s = the period of the time lag. Since our data is monthly, s is 12. Also from the previous analysis, we differenced our data at lag 12 to remove the seasonality, so D is 1.

Hence, we get the SARIMA model (p,1,q) x (P,1,Q) S=12. Next, we will utilize the ACF/ PACF graphs as well as AIC/ BIC tests to find the remaining parameters (p,q) and (P,Q).

Preliminary Model Identification

ACF and PACF plots will help us identify our seasonal terms P and Q. Looking at the detrended and deseasonalized plots with lag in intervals of 12 (lag = 12, 24 , 36 …) in *Figure 9* and *10*, we can tell that the ACF cuts off after lag 12 and the PACF trail off after lag 24. After discussing with our TA, we can assume the parameter of P&Q should between 0-2. Based on this assumption, we tested different combinations of P&Q and finally we conclude when P=2 Q=2 we have most appropriate coefficients and result.

Figure 9

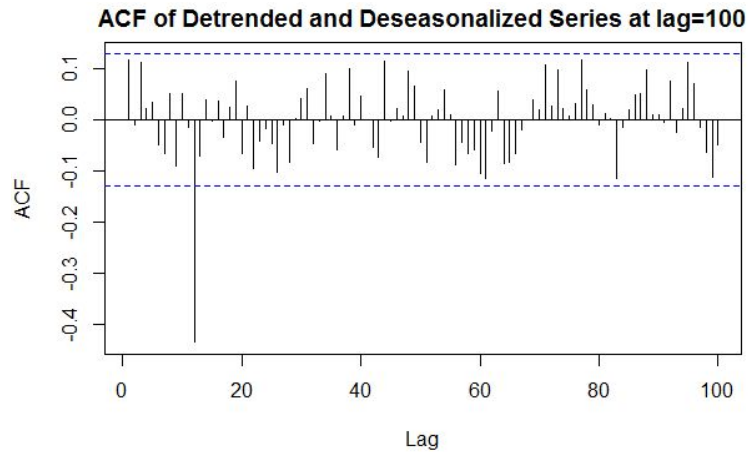
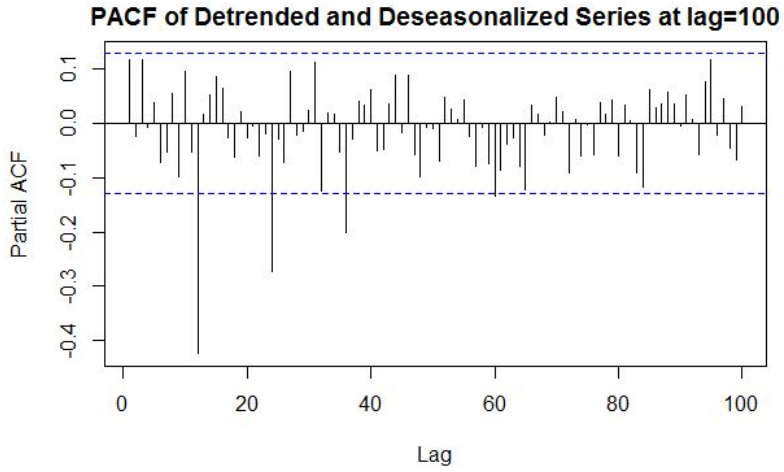


Figure 10



Our next step is to find the non-seasonal components (p, q). To determine the non-seasonal components, we first need to look at the graph when lag < 12 as the time series has a seasonality of period 12. Zooming in the ACF and PACF plots in *Figure 9* and *Figure 10*, we discovered that there are no extreme lags exceeding the confidence interval meaning that they are all close to zero. To further confirm, we zoom in further in *Figure 11* and *Figure 12.* This gives us white noise, but we can still use a for loop (q:0-1, p:0-1) to determine the parameters for the non-seasonal component. This gives us 4 models to consider as p and q can take in values between 0 and 1.

Figure 11

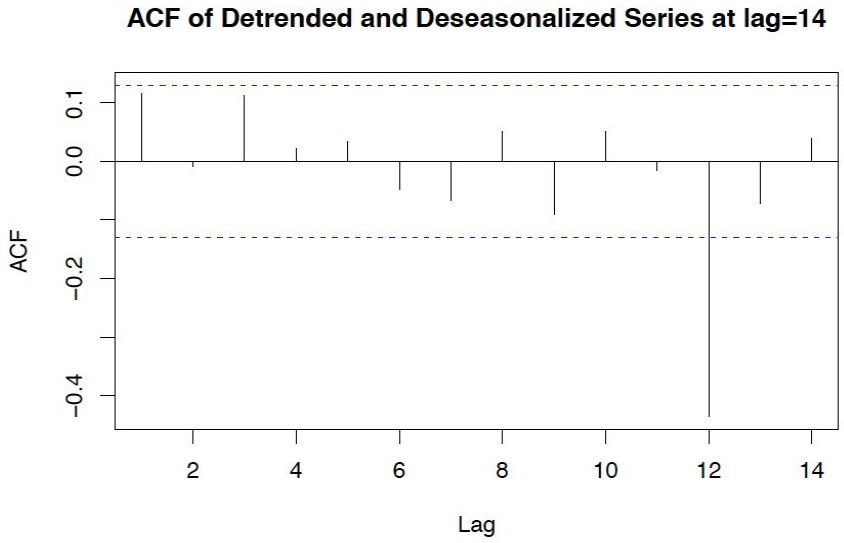
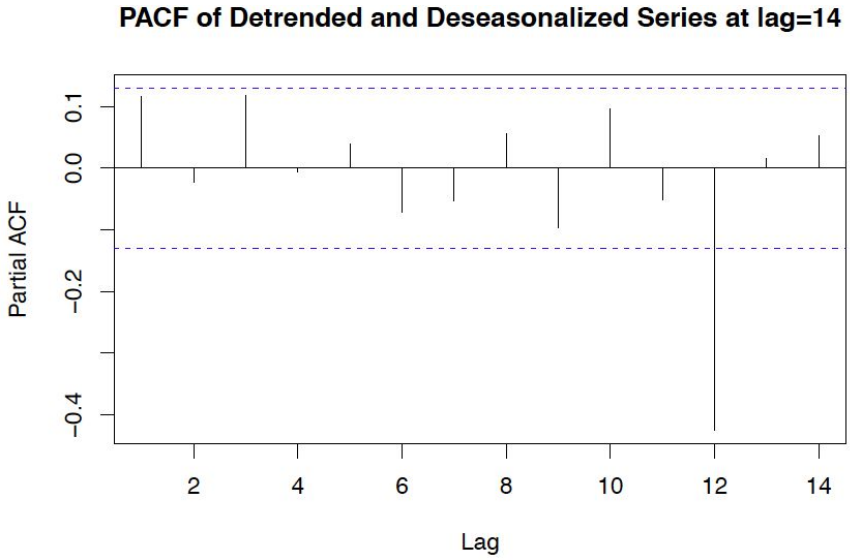


Figure 12



Model Selection

To determine the best model out of our 4 choices, we used a for loop to examine the Akaike information criterion (AICc) and the Bayesian information criterion (BIC) to decide the best models for the non-seasonal components. Since the lags from 1 to 11 are all within the

confidence bound, we can say the maximum values of both p and q are 1. Therefore, we fixed the SARIMA model (p,1,q) x $(2,1,2)_s$ with S=12 and run a for loop from (p=0~1, q=0~1) to select the smallest AICc and BIC values.

We used AICc and BIC to find the model that is the best fit for our data by choosing the lowest and second lowest AICc and BICc values.

Table 1

	q=0	**q=1**
p=0	**-11.25209**	**-11.25139**
p=1	**-11.16587**	**-11.25139**

AICc

Table 2

	q=0	**q=1**
p=0	**-12.20374**	**-12.18904**
p=1	**-12.10353**	**-12.17513**

BIC

From the tables above, the smallest AICc and BIC values are seen at p=0, q=0, and the second smallest values are at p=0, q=1. Thus, the two best models are SARIMA$(0,1,0)\times(2,1,2)_{12}$ with a standard error of 4.907e-06 and SARIMA$(0,1,1)\times(2,1,2)_{12}$ with a standard error of 4.872e-06. However, SARIMA$(0,1,0)\times(2,1,2)_{12}$ doesn't seem as applicable to this case since it is a white noise when p = 0 and q = 0, and is not very possible for realistic data.

Then we also ran the auto arima function in R with the MLE method and to confirm the selected values p=0, q=1 were correct.

```
# AIC
library(forecast)

# AICc (when P=2, Q=2) --> finding p,q (FIGURE 1)
auto.arima(newmodel)

## Series: newmodel
## ARIMA(0,1,1)
##
```

Model Estimation

We used MLE to fit the models and evaluate the coefficients as seen in *Table 3* below.

Table 3

	Model 1 **SARIMA(0,1,1)x(2,1,2)$_{12}$**	**Model 2** **SARIMA(0,1,0)x(2,1,2)$_{12}$**
MA(1)	0.0842	-
SAR(1)	-0.8521	-0.8486
SAR(2)	0.1209	0.1194
SMA(1)	-0.0542	-0.0619
SMA(2)	-0.9458	-0.9380

We also check the roots of the polynomials for the two models to see if they are causal and inevitable. We plot the roots as seen in the appendix figures; all roots of both models lie outside the unit circle and all coefficients are in the range of -1 to 1, that is the absolute values of the coefficients are less than 1.

Below we have our models, where X_t is transformed and differenced so that $X_t = \nabla\nabla_{12} Y_t^{-0.1818182}$:

Model 1: SARIMA(0,1,1)×(2,1,2)$_{12}$

(1 +0.8521B^12- 0.1209B^24)Xt = (1 + 0.0842B)(1-0.0542B^12-0.9458B^24)Zt

where Zt ~ N(0, 4.872e-06)

Model 2: SARIMA(0,1,0)×(2,1,2)$_{12}$

(1 + 0.8486B^12 - 0.1194B^24)Xt = (1+0.0619B^12+0.9380B^24)Zt

where Zt ~ N(0, 4.907e-06)

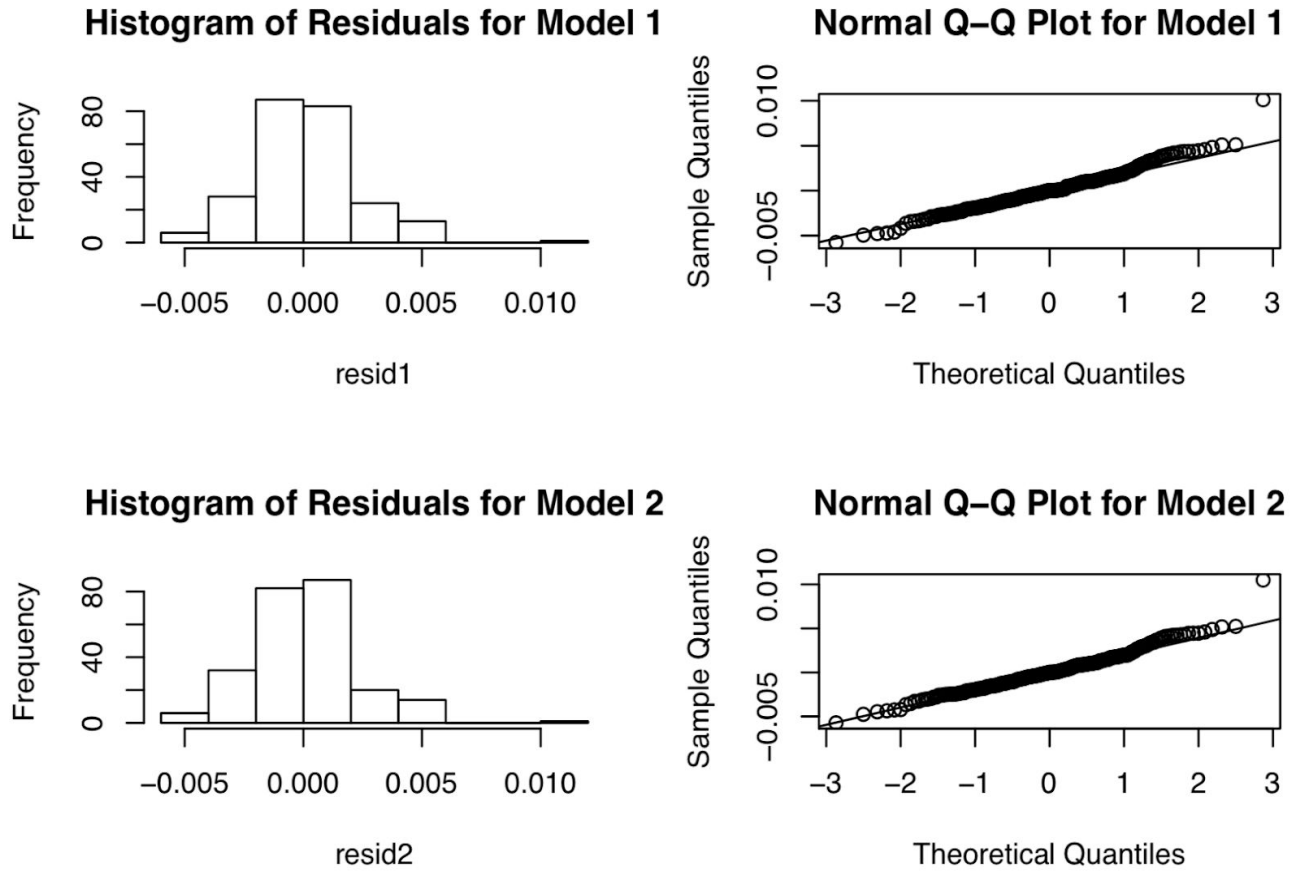
Diagnostics

Diagnostics are done to ensure that the assumption of the SARIMA model are followed, namely that the residuals are normally distributed, have no serial correlation, and are homoscedastic. After selecting our two models, we ran the residuals of the models through the Shapiro Wilks test to test for normality, the Ljung-Box and Box-Pierce tests to test for serial correlation, QQ plots and histograms to check normality of residuals, and examined their ACF and PACF plots to determine homoscedasticity.

Normality

Our first time in determining the normality of the residuals was to plot them in histograms and associated Q-Q plots.

Figure 13



As seen above in *Figure 13*, both histograms appear fairly normally distributed with a standard bell shape over the tops of the bins. Also, although there are a few outliers in the normal Q-Q plots of both models near the ends of the data, most of the points lie on the normality line.

To ascertain the normality of the residuals from the two models, we also performed the Shapiro-Wilk test. The null hypothesis states that the residuals from the model are normally distributed and the alternative hypothesis states that the residuals are not normally distributed. The tests were evaluated with $\alpha = 0.05$ level of significance and resulted in *Table 4*.

Table 4

	W.Statistic <dbl>	P.value <dbl>
Model 1	0.9780334	0.0008320613
Model 2	0.9747652	0.0002635252

The p-value for the test on model 1 is 8.3206e-04 and 2.6351e-04 on model 2. We reject the null hypothesis at the standard 0.05 level for both residuals as both p-values are found to be much less than 0.05. This rejection can be explained by outliers in the data set of residuals of both models and other factors that are not accounted for in the models. However since the Normal Q-Q plots show a mostly normal distribution when excluding outliers, we can still proceed to test the residuals for independence. Also, according to the histograms of residuals (Figure 13), they perform as normal with mean equal to zero. Therefore, it is reasonable to conclude our residuals are approximately normal and our assumption of normality holds for both models.

Serial Correlation (Independence)

Serial correlation of residual values can strongly impact the model and make the model not dependable when forecasting. To test for serial correlation and to confirm our assumption of uncorrelated residuals, we applied the Ljung-Box test and the Box-Pierce test when p = .05

Table 5

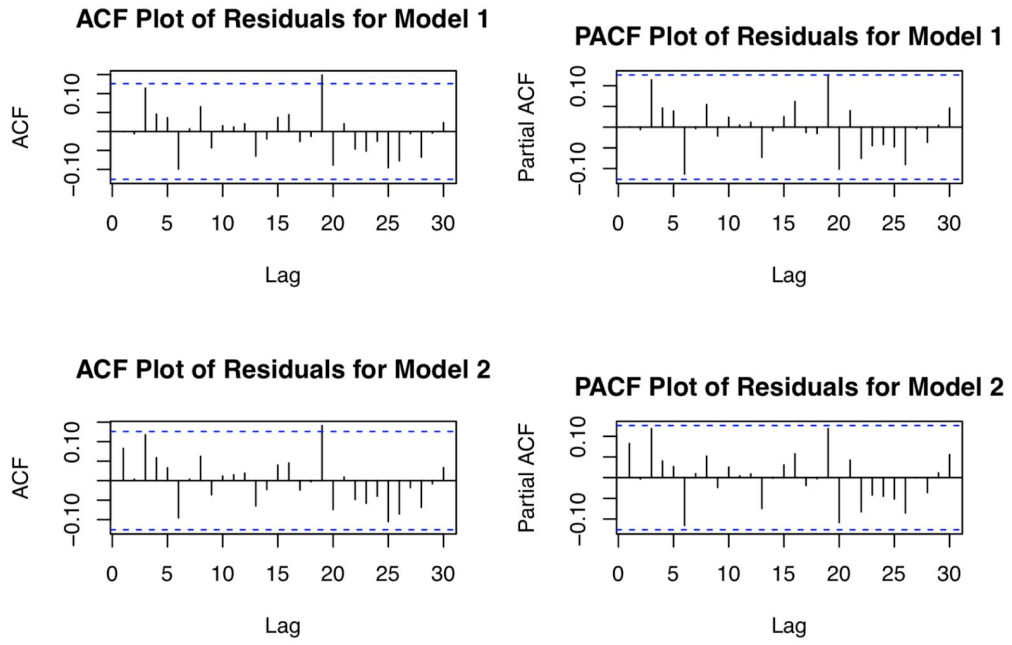
	Box-Pierce	**Ljung-Box**
Model 1	0.6227433	0.598754
Model 2	0.4634229	0.440355

The null hypothesis states that there is no serial correlation between residuals meaning that the residuals are independently distributed at a 95% confidence level and the alternative hypothesis states the residuals are serially correlated, implying that the residuals are not independently distributed. Looking at the results in the table above, both p-values are much larger than .05, thus we cannot reject the null hypothesis and confirm that our residuals are serially uncorrelated for models 1 and 2.

Homoscedasticity (Constant Variance)

It is also important to ensure that all our models are homoscedastic meaning that they have a constant variance. This is important as a variance that is unequal across the values range can have a large impact on forecasting and cause our model to be undependable. We plotted the ACF and PACF plots for residuals below in *Figure 14*. If most of the residual values lie within the 95% upper and lower bounds of the white noise, we can determine that our models are homoscedastic.

Figure 14



Apart from an outlier at lag = 19, the residual values lie well within the established confidence bounds. In conclusion, we can infer that the models display common variance and are homoscedastic.

Both models 1 and 2 satisfy all the assumptions of the SARIMA model. The residuals of both models demonstrate normality, independence and homoscedasticity. However in the end, we must select one model that best fits the data. Looking at the two models, it becomes clear that we should choose model 1, SARIMA(0,1,1)×(2,1,2)S=12, as our final model because model 2, SARIMA(0,1,0)×(2,1,2)S=12, is white noise as stated in the model selection.

Forecasting

The primary aim of this project was to model the ^GSPC monthly closing values as a time series and use the final model to forecast values for the last 10 months of 2017. By testing our accuracy in forecasting the last 10 months of 2017, we can determine the efficacy of our model in hopes of utilizing it for future predictions. The red dots in *Figures 15* and *16* and the dark green dots in *Figure 17* show the forecast values of the time series data from March 2017 to December 2017. The blue dots on the third plot shows the actual recorded values of March 2017 to December 2017 and the dotted lines in each of the figures show the 95% confidence boundaries of the predicted values.

Figure 15

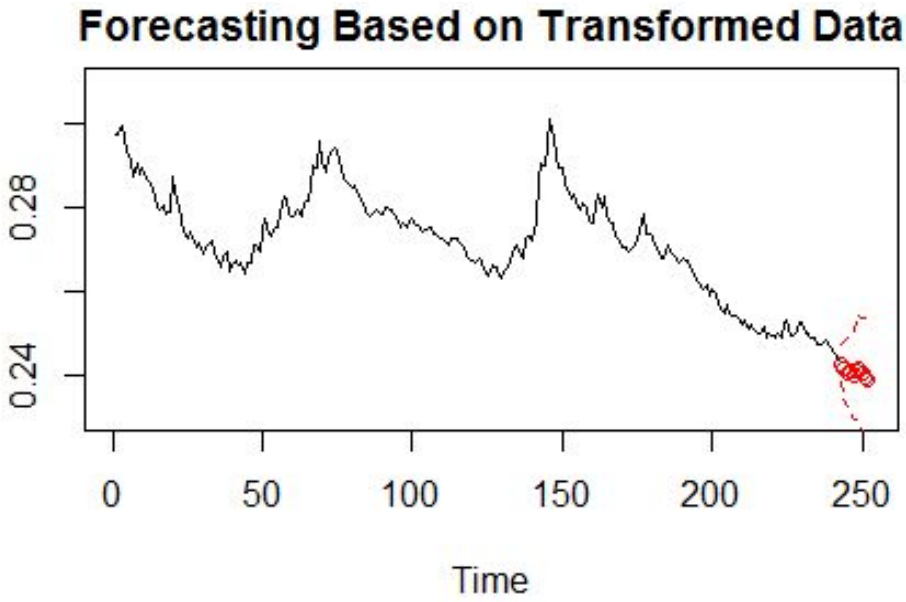


Figure 16

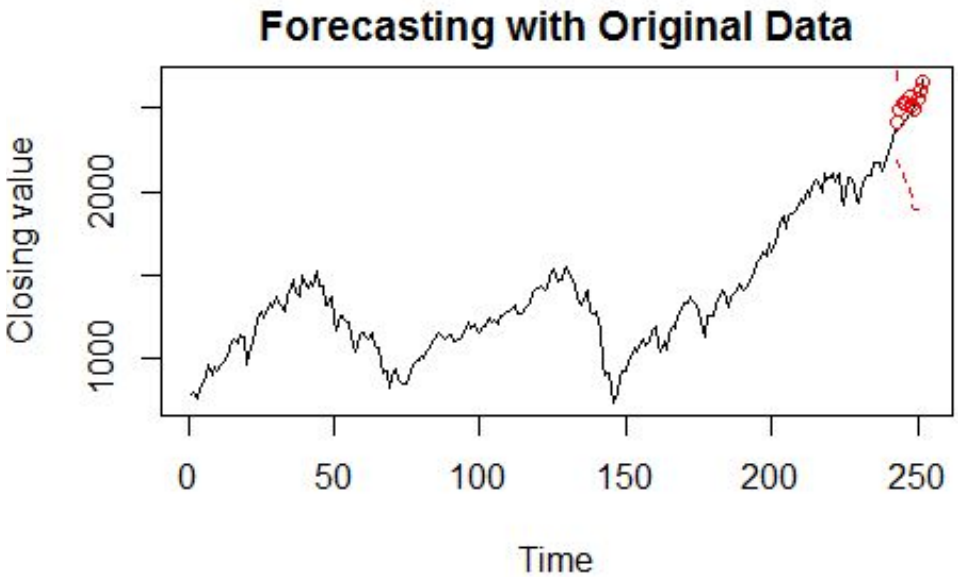
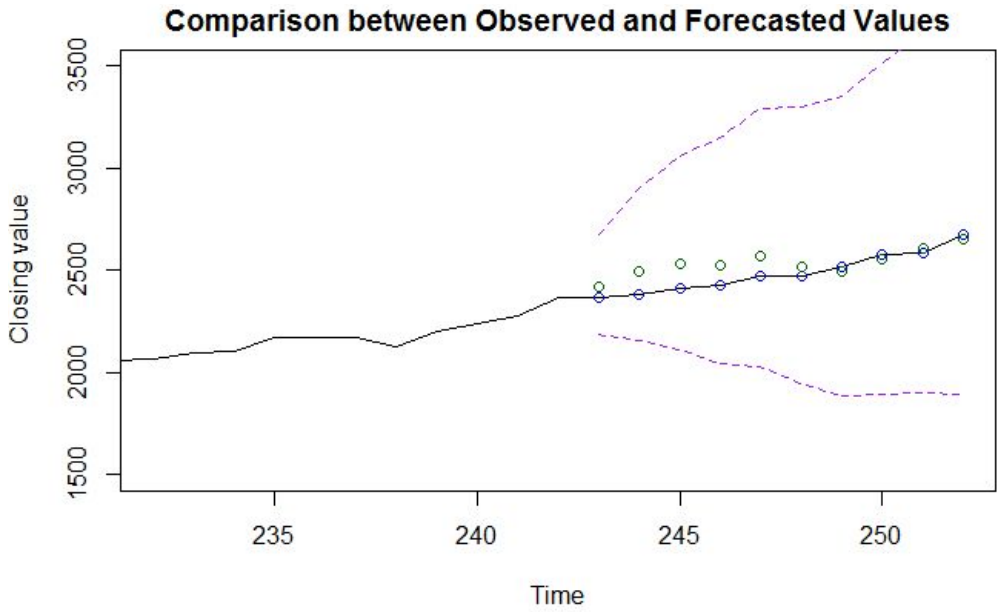


Figure 17



Looking at *Figure 17*, we note that the forecast values are close to the actual values and our model is fairly accurate in the 95% confidence interval. The values are increasing, which shows that they are following the pattern of the original data. The observed and forecasted values not only remain within the confidence interval, they also overlap at times. Thus, we can see that our final model is successful when it comes to forecasting.

Conclusion

Our main objective was to develop a time series model to use in forecasting closing stock values for the first 10 months of 2018 at the S&P 500, hereby referred to as the ^GSPC index. The project focused on using data recorded for the index between January 1997 and December 2017 which has 252 observations. From the analysis we noted a gradual increase in values from year to year as we observed in the upward trend of the plotted time series. We also observed that there was seasonality in our data; January often recorded small values which increased going to

August before slightly dropping in October and later increasing to a high in December. We made the data stationary by differencing twice, initially at lag 12 and later at lag 1, to make it stationary before trying to fit into different models and selecting the model that is best fit by the data. Then we used model selection process by AICC and BICC to find the best two models with the lowest AIC and BIC value. We also test these models through diagnostics. Although we found the models had outliers and failed the Shapiro test, they still follow a normal distribution in the histogram and Q-Q plot and passed the Wilk test. The models also passed tests for identicality by Ljung-Box test and constant variance by constructing an interval on the ACF and PACF of the model residuals. Based on AIC and diagnostics, the final model we conclude was a SARIMA process

SARIMA (0,1,1)×(2,1,2)S=12

(1 +0.8521B^12- 0.1209B^24)Xt= (1 + 0.0842B)(1-0.0542B^12-0.9458B^24)Zt

We used the final model to forecast closing values at ^GSPC for the last 10 months of 2017 and noted that all forecast values lied within the 95% confidence interval. The forecast values are, in fact, close to the observed values. This validates the efficacy of our model in forecasting monthly closing values of ^GSPC.

Future Study

Although we got a great result to predict the future value (our forecasted values are very close to the true value in the original dataset), we still have some limitations. For example, there may be potential effects such as policy problems and a financial crisis happened that impacted the stock price which lead to some prediction errors in our forecasting. If possible, we may use more advanced methods such as machine learning and Long Short Term Memory to avoid these errors and get a better prediction.

References

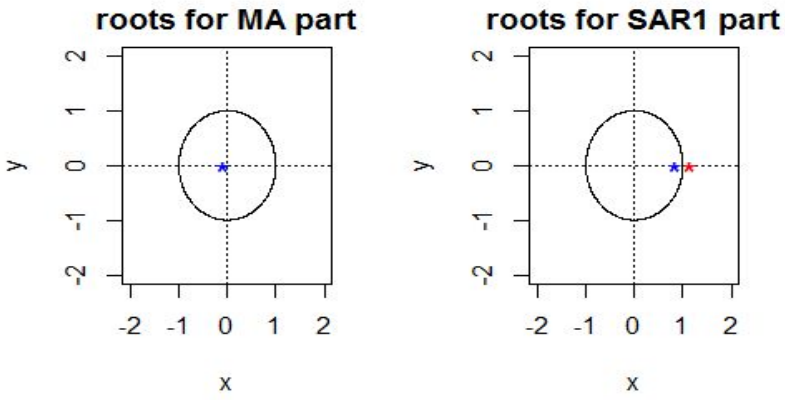
G. Russon, M., & F. Vakil, A. (2017). On the non-linear relationship between VIX and realized SP500 volatility. *Investment Management and Financial Innovations*, *14*(2), 200-206. doi:10.21511/imfi.14(2-1).2017.05

"^GSPC : Summary for S&P 500." *Yahoo! Finance*, Yahoo!, 2 Mar. 2019, finance.yahoo.com/quote/^GSPC?p=^GSPC.

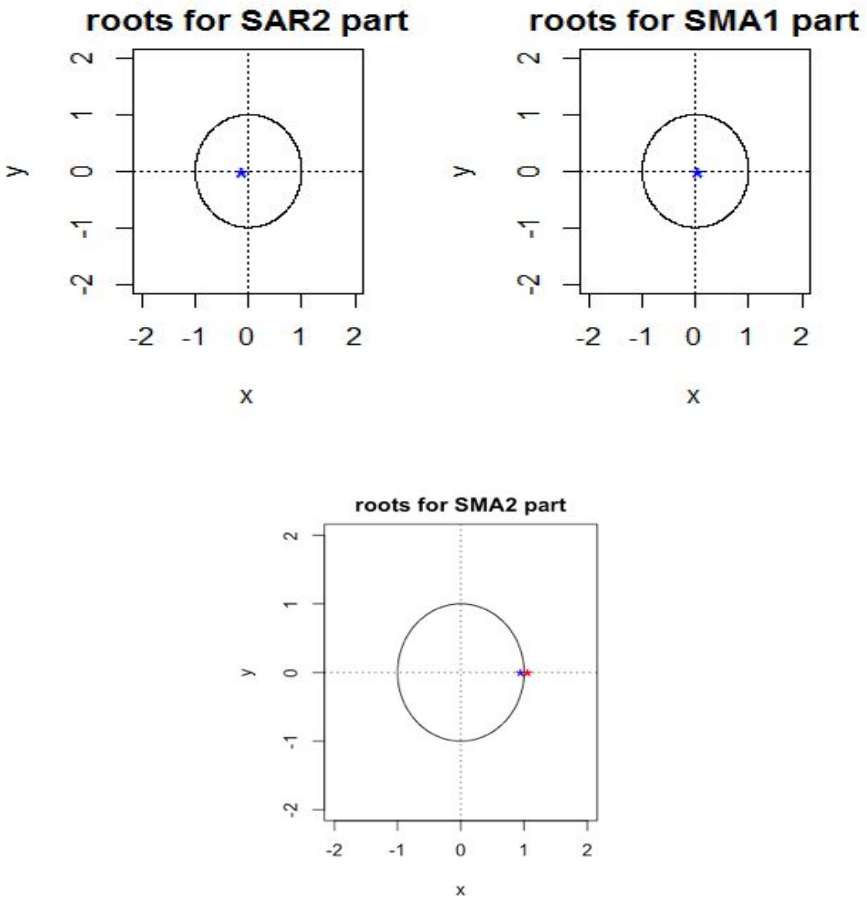
Appendix

Model 1

Appendix Figure 1

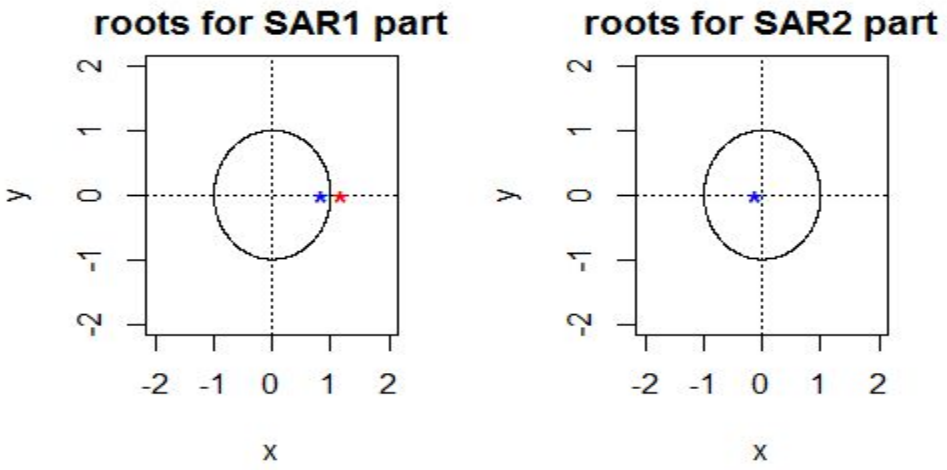


Appendix Figure 2

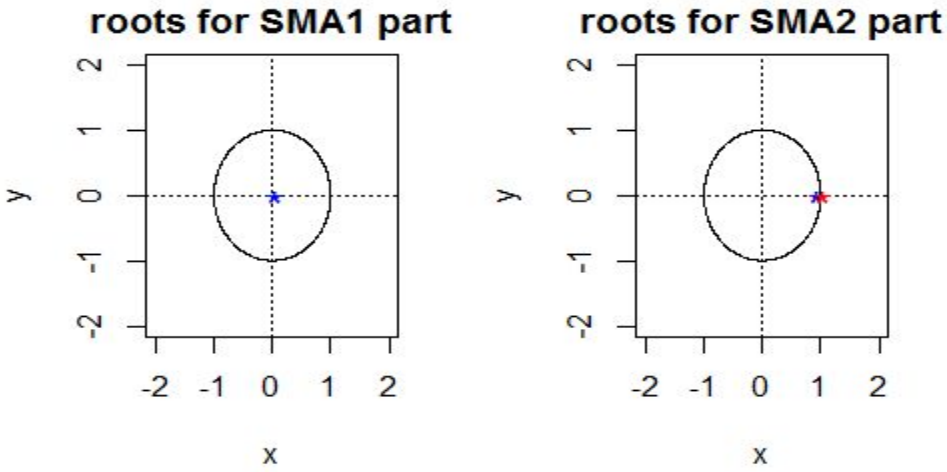


Model 2

Appendix Figure 3



Appendix Figure 4



Appendix Figure 5

```
Call:
arima(x = newmodel, order = c(0, 1, 1), seasonal = list(order = c(2, 1, 2),
    period = 12), method = "ML")

Coefficients:
          ma1     sar1    sar2     sma1     sma2
      0.0842  -0.8521  0.1209  -0.0542  -0.9458
s.e.  0.0678   0.1417  0.0725   0.1576   0.1563

sigma^2 estimated as 4.872e-06:  log likelihood = 1057.98,  aic = -2105.95
```

Appendix: R code

---

title: "PSTAT 174 Final Project (final edit)"

author: "Esther Hsu"

date: "3/9/2019"

output: pdf_document

```{r}
# Import packages
library(MASS)
library(tseries)
library(TSA)
library(astsa)
library(ggplot2)
library(lmtest)
library(forecast)
```

```{r}
# plot original data
gspc <- read.csv(file.choose())
closepr <- gspc$Close
close1 <- ts(gspc$Close, frequency=12)
summary(close1)
var(close1)
plot(close1, xlab="Time", ylab="close price", main="GSPC")
seasonplot(close1, 12, col=rainbow(3), year.labels=TRUE, main="Seasonal Plot")
```

```{r}
decomp <- decompose(close1)
autoplot(decomp, main="Decomposition_Plot")
```

```{r}
acf(close1, lag.max=100, col="red", main="ACF of Series close1")
pacf(close1, lag.max=100, col="blue")
title(main="PACF of Series close1")
```

```{r}
# Stablize variance using box-cox
require(MASS)
bxTransform <- boxcox(closepr~as.numeric(1:length(closepr)))
lambda <- bxTransform$x[which.max(bxTransform$y)]
lambda

# Transformation Model
Trans <- closepr^lambda
```

```
newmodel <- ts(Trans[1:(length(closepr)-10)])
# TS plot of closepr^lamda
plot(newmodel, xlab="Time", ylab="Close Price", main="lamda, GSPC")

# ACF and PACF
acf(newmodel, lag.max=100, col="red", main="ACF of Box-Cox Transformation")
pacf(newmodel, lag.max=100, col="blue")
title(main="PACF of Box-Cox Transformation")
```

```{r}
# De-seasonalize
pricediff12 <- diff(newmodel, lag=12)
plot(pricediff12, xlab="Time", ylab=" ", main="Differenced at lag12")
abline(lm(pricediff12~as.numeric(1:length(pricediff12))))
var(pricediff12)
```

#downward trend line on the graph. Next, we need to remove the trend based on the deseasonalized data we obtained.

```{r}
# De-Trend
pricediff12diff1 <- diff(pricediff12, lag=1)
plot(pricediff12diff1, xlab='time', ylab='', main='pricediff12anddiff1')
abline(lm(pricediff12diff1~as.numeric((1:length(pricediff12diff1)))))
var(pricediff12diff1)
```

```{r}
# De-trend Again
```

```
pricediff12diff2 <- diff(pricediff12diff1, lag=1)
var(pricediff12diff2)
#variance increases after one more difference so the model should be D=1,d=1
```

```
# variance increases after one more difference so the model should be D=1,d=1
```{r}
adf.test(pricediff12diff1, k=12)
```

#p-value in the test is equal to 0.01, which is smaller than 0.05 in the confidence interval of 95%.
#Thus, we can reject the null hypothesis, and the time series is proven to be stationary

#Model Identification
#Identify P,Q
```{r}
acf(pricediff12diff1,lag.max=100, main="ACF of Detrended and Deseasonalized Series at lag=100")
pacf(pricediff12diff1, lag.max=100)
title(main="PACF of Detrended and Deseasonalized Series at lag=100")
```

#probably P=0~2, Q=0~2
```{r}
acf(pricediff12diff1, lag.max=14, main="Detrended and Deseasonalized ACF at lag=14")
pacf(pricediff12diff1, lag.max=14)
title(main="Detrended and Deseasonalized PACF at lag=14")
# lag p=1, q=1 small p and q
```

looks as if p=0 and q=1
#- Possible models after differencing and transforming: MA(1)
Analysis of ACF and PACF: P=2, Q=2
```{r}
```
```

```
# AIC
library(forecast)
# AICc (when P=2, Q=2) --> finding p,q (FIGURE 1)
auto.arima(newmodel)
AICc <- numeric()
for (p in 0:1){
  for (q in 0:1){
    AICc <- c(AICc, sarima(newmodel, p, 1, q, 2, 1, 2, 12, details=FALSE)$AICc)
  }
}
AICc <- matrix(AICc, nrow=2, byrow=TRUE)
rownames(AICc) <- c("p=0", "p=1")
colnames(AICc) <- c("q=0", "q=1")
AICc <- data.frame(AICc)
AICc
# smallest: p=0, q=0; second smallest: p=0, q=1
```

```{r}
# BIC (P=2, Q=2), find p,q (FIGURE 2)
BIC <- numeric()
for (p in 0:1){
  for (q in 0:1){
    BIC <- c(BIC, sarima(newmodel, p, 1, q, 2, 1, 2, 12, details = FALSE)$BIC)
  }
}
BIC <- matrix(BIC, nrow=2, byrow=TRUE)
rownames(BIC) <- c("p=0", "p=1")
colnames(BIC) <- c("q=0", "q=1")
BIC <- data.frame(BIC)
```

BIC

#smallest: p=0, q=0, second smallest: p=0, q=1

```

# 3) BEST MODELS (FINAL) are:

## Model 1: SARIMA (0,1,1) x (2,1,2)s=12

## Model 2: SARIMA (0,1,0) x (2,1,2)s=12

#4) Estimate the coefficients

```{r}
# Model 01: SARIMA (0,1,1) x (2,1,2)s=12
model01 <- arima(newmodel, order=c(0,1,1), seasonal=list(order=c(2,1,2), period=12), method="ML")
model01
```

```{r}
# Model 02: SARIMA (0,1,0) x (2,1,2)s=12
model02 <- arima(newmodel, order=c(0,1,0), seasonal=list(order=c(2,1,2),period=12), method="ML")
model02
```

We choose model 1 as our final model (0,1,1)x(2,1,2)s=12

5) Diagnostic checks on our chosen model

```{r}
# Inputting the plot.roots function
plot.roots <- function(ar.roots=NULL, ma.roots=NULL, size=2, angles=FALSE, special=NULL,
                       sqecial=NULL, my.pch=1, first.col="blue", second.col="red",
                       main=NULL)
{
   xylims <- c(-size, size)
```
```

```
  omegas <- seq(0,2*pi, pi/500)
  temp <- exp(complex(real=rep(0,length(omegas)), imag=omegas))
  plot(Re(temp), Im(temp), typ="l", xlab="x", ylab="y",
       xlim=xylims, ylim=xylims, main=main)
  abline(v=0, lty="dotted")
  abline(h=0, lty="dotted")
  if(!is.null(ar.roots)){
     points(Re(1/ar.roots), Im(1/ar.roots), col=first.col, pch=my.pch)
     points(Re(ar.roots), Im(ar.roots), col=second.col, pch=my.pch)
  }
  if(!is.null(ma.roots)){
     points(Re(1/ma.roots), Im(1/ma.roots), pch="*", cex=1.5, col=first.col)
     points(Re(ma.roots), Im(ma.roots), pch="*", cex=1.5, col=second.col)
  }
  if(angles){
     if(!is.null(ar.roots)){
        abline(a=0, b=Im(ar.roots[1])/Re(ar.roots[1]), lty="dotted")
        abline(a=0, b=Im(ar.roots[2])/Re(ar.roots[2]), lty="dotted")
     }
     if(!is.null(ma.roots)){
        sapply(1:length(ma.roots), function(j)abline(a=0,
                                          b=Im(ma.roots[j])/Re(ma.roots[j]),
                                          lty="dotted"))
     }
  }
  if(!is.null(special)){
     lines(Re(special), Im(special), lwd=2)
  }
  if(!is.null(sqecial)){
```

```
      lines(Re(sqecial), Im(sqecial), lwd=2)
   }
}
```

```{r}
# MODEL 1:
# plotting roots --> checking causality and invertability
model01 <- arima(newmodel, order=c(0,1,1), seasonal=list(order=c(2,1,2), period=12), method="ML")
model01
#source("plot.roots.R")
par(mfrow = c(1,2))
plot.roots(NULL, polyroot(c(1, 0.0842)), main="roots for MA part")
plot.roots(NULL, polyroot(c(1, -0.8522)), main="roots for SAR1 part")
plot.roots(NULL, polyroot(c(1, 0.1209)), main="roots for SAR2 part")
plot.roots(NULL, polyroot(c(1, -0.0542)), main="roots for SMA1 part")
plot.roots(NULL, polyroot(c(1, -0.9458)), main="roots for SMA2 part")
```

```{r}
# MODEL 2:
# plotting roots --> checking causality and invertability
model02 <- arima(newmodel, order=c(0,1,0), seasonal=list(order=c(2,1,2), period=12), method="ML")
model02

#source("plot.roots.R")
par(mfrow = c(1,2))
plot.roots(NULL, polyroot(c(1, -0.8486)), main="roots for SAR1 part")
plot.roots(NULL, polyroot(c(1, 0.1194)), main="roots for SAR2 part")
plot.roots(NULL, polyroot(c(1, -0.0619)), main="roots for SMA1 part")
```

```
plot.roots(NULL, polyroot(c(1, -0.9380)), main="roots for SMA2 part")
```

```{r}
# To check NORMALITY
# residuals for model 1:
resid1 <- residuals(model01)
# residuals for model 2:
resid2 <- residuals(model02)

# MODEL 1
op <- par(mfrow=c(2,2))
hist(resid1, main="Histogram of Residuals for Model 1")
qqnorm(resid1, main="Normal Q-Q Plot for Model 1")
qqline(resid1)

# MODEL 2
hist(resid2, main="Histogram of Residuals for Model 2")
qqnorm(resid2, main="Normal Q-Q Plot for Model 2")
qqline(resid2)
par(op)
```

```{r}
# Shapiro Test for Model 1 and 2
shapiro <- matrix(c(shapiro.test(resid1)$statistic, shapiro.test(resid1)$p.value,
                    shapiro.test(resid2)$statistic, shapiro.test(resid2)$p.value),
                  nrow=2, byrow=T)
# want a p-value greater than 0.05 (since H0: residuals are normal)
rownames(shapiro) <- c("Model 1", "Model 2")
```

```
colnames(shapiro) <- c("W Statistic", "P-value")

shapiro_test <- data.frame(shapiro)

shapiro_test
```

#From the Shapiro-Wilk test we found a p-value of 8.3206e-04 which rejects the null hypothesis at the standard 0.05 level. This is due to some outliers and other components that we cannot use to analyze this model. But as seen in the histogram and QQ plot the residuals lie on the normality line.

```{r}
# INDEPENDENCE/CORRELATION diagnostics


# Model 1:

b_1 <- Box.test(resid1, lag=12, type="Box-Pierce", fitdf=2)$p.value

b_2 <- Box.test(resid1, lag=12, type="Ljung-Box", fitdf=2)$p.value

b_1 # p-value is greater than 0.05; it's good

b_2 # p-value is greater than 0.05; it's good


# Model 2:

b_3 <- Box.test(resid2, lag=12, type="Box-Pierce", fitdf=2)$p.value

b_4 <- Box.test(resid2, lag=12, type="Ljung-Box", fitdf=2)$p.value

b_3 # p-value is greater than 0.05; it's good

b_4 # p-value is greater than 0.05; it's good
```

#Both p-values are above our standard 0.05 significance level, thus we confirm our assumption that our residuals are uncorrelated for our model.

```{r}
# CONSTANT VARIANCE of residuals diagnostics:

# model 1:

par(mfrow=c(2,2))

# acf

acf(resid1, main="ACF Plot of Residuals for Model 1", lag.max=30)
```

```
# pacf
pacf(resid1, lag.max=30)
title(main="PACF Plot of Residuals for Model 1", outer=FALSE, line=1)

# model 2:
# acf
acf(resid2, main="ACF Plot of Residuals for Model 2", lag.max=30)
# pacf
pacf(resid2, lag.max = 30)
title(main="PACF Plot of Residuals for Model 2", outer=FALSE, line=1)
```

Without some outliers, the residuals all lie in the confidence bounds.

#6) Forecasting

```{r}
# Forecasting based on Final Model
pred.ts <- predict(model01, n.ahead=10)

upper.ts <- pred.ts$pred + 1.96 * pred.ts$se # upper bound for CI for transformed data
lower.ts <- pred.ts$pred - 1.96 * pred.ts$se # lower bound for CI for transformed data

ts.plot(newmodel, xlim=c(1, length(newmodel) + 10), main="Forecasting Based on Transformed Data",
        ylim=c(0.23, 0.31), ylab="")
lines(upper.ts, col="red", lty="dashed")
lines(lower.ts, col="red", lty="dashed")
points ((length(newmodel) + 1):(length(newmodel) + 10), pred.ts$pred, col="red")
```

```{r}
```

```
predict.origin <- pred.ts$pred^(1/lambda) # back-transform in order to return to get predictions of the original time series
# CI for original data
upper.or <- upper.ts^(1/lambda) # upper bound of the CI
lower.or <- lower.ts^(1/lambda)  # lower bound of the CI
# Plot of forecast with original data
close2 = ts(closepr)
ts.plot(close2, xlim=c(1, length(close2)), main="Forecasting with Original Data",
      ylab="Closing value")
lines(upper.or, col="red", lty="dashed")
lines(lower.or, col="red", lty="dashed")
points ((length(newmodel) + 1):(length(newmodel) + 10), predict.origin, col="red")
```

```{r}
# zooming in
ts.plot(close2, xlim=c(length(close2) - 20, length(close2)), ylim=c(1500, 3500),
      main="Comparison between Observed and Forecasted Values", ylab="Closing value")
points((length(newmodel)+1):(length(newmodel)+10), close2[243:252], col="blue")
points((length(newmodel)+1):(length(newmodel)+10), predict.origin, col="dark green")
lines((length(newmodel)+1):(length(newmodel)+10),upper.or,lty=2, col="purple")
lines((length(newmodel)+1):(length(newmodel)+10),lower.or,lty=2, col="purple")
```