CARLETON UNIVERSITY

SCHOOL OF
MATHEMATICS AND STATISTICS

HONOURS PROJECT

TITLE: Stock Market Analysis and Predictions Using Time Series Approach

AUTHOR: Danxi(Daisy) Wu

SUPERVISOR: Patrick Farrell

DATE: May 5, 2020

Abstract

The stock market, unpredictable but powerful indicators of global or national economies, enables the buying and selling of company stocks and thus attracts many investors. The ability to forecast the stock market can maximize the profit and minimize the loss. This paper performed a time series analysis to predict the short-term stock market. First, we check for stationarity of the data and apply to eliminate trends by using log transformation and difference equations. Then, an Autoregressive Integrated Moving Average (ARIMA) model is employed to analyze the weekly historical data of Toronto Stock Exchange Index for past five years and forecast, forecast the future values based on the predicted model and compare the results to actual data. A similar approach is performed on the Shanghai Stock Exchange Index. The model based on historical data in early 2003 is used as the training model. The model is then applied to the dataset from January 2015 to December 2019 to predict the current stock trend. The models, in general, are too simple and ideal to represent the overall performances. A more advanced or hybrid model may yield a better prediction.

Contents

1 Introduction

1.1 Motivation

The stock market is one of the most powerful indicators of world economics and finances. It can be influenced by market activities, government policies, or any public events. The stock market reflects all currently available information, as well as any price changes that are based on the efficient-market hypothesis and not on newly relevant information. Thus, the stock market is unpredictable. In contrast, the actual market experience differs from the hypothesis, and some people are unable to predict better than others based on some material nonpublic information. Many stock market studies focus heavily on the recommendation of whether to buy or sell the stock but fail to address when. However, investors would benefit, and would secure greater profits while minimizing investment risks if they could know stock trends in advance. It would thus be helpful to be able to predict the short-term stock prices.

Recently, the global outbreak of Novel Coronavirus (COVID-19) has affected daily activities and caused economic disruptions due to the shutdown of major countries and high transmission and mortality rates in some metropolises. It has caused great panic among both domestic and oversea investors. Many people around the world have begun to sell their stock investment, which has caused the major stock indexes to tumble. The U.S. Stock Market has been crushed several times within a short period. It is worthwhile to forecast the future stock prices to predict when the society will recover from this event and how the pandemic may influence the economy in general.

1.2 Objective

This thesis is part of the graduation requirements for obtaining a Bachelor of Mathematics Honors degree at Carleton University. The methodology section is organized in a way that follows Shumway's book of *Time Series Analysis and Its Applications*. The purpose of this study is to develop some time series and machine learning models to analyze the stock market and forecast its performance for the coming four months by using some of the major indexes around the world for past historical data. First, historical data from the Toronto Stock Exchange from 2015 to 2020 was used to develop an ARIMA model to forecast the performances from August 2019 to December 2019. The prediction was compared to the actual stock prices to verify that model was working. Then, I apply a similar technique to analyze the stock performance of the Shanghai Composite Index (SSE) from 2015 to 2020. Next, I modified the SSE model to reflect the possible impact of novel coronavirus on the stock market, based on a similar situation in China during early 2003. The new model was then used to predict the stock performances of the index for the following four months from December 2019 to March 2020. The predicted model is then compared to the actual values.

1.3 Literature Review

Many researchers and analysts have been studying and predicting stock markets for years. People have begun to develop algorithms to help them make more accurate and working predictions. Neural networks (NN) are commonly used in solving various business problems, for example, sales forecasting, stock market prediction, and risk management. NN use a feature called an iterative learning process to adjust the weights of input items at each iteration until the correct weights are assigned to the input

variables. A peer review conducted by Soni & Kumar in 2005 discovered that Neural Networks and Neuro Fuzzy systems are effective in forecasting of the stock market. The advantage of neural networks is that they have high tolerance of noisy, insufficient, and volatile data. Other models such as Extreme Learning Machine and Support Vector Machine (SVM) may have higher prediction accuracy and faster speed than the Back-Propagation Neural Networks (Li, et. al., 2016).

To predict major Chinese stock market indices, Chen, Y. & Hao, Y in 2007 proposed a hybridized model using the a Feature Weighted SVM (FWSVM) and K-Nearest Neighbor (KNN) to predict the SSE Composite Index and Shenzhen Stock Exchange Component Index. They discovered that the FWSVM outperformed the SVM in the short, medium, and long run. For example, when predicting stock prices one day ahead, the FWSVM was 1.8% more accurate than the SVM; when predicting stock indices ten days ahead, the FWSVM was 1.0% more accurate than the SVM. The AUC Scores in percentage for predicting profit loss were 98.2% for FWSVM and 96.4% for SVM in forecasting one day ahead. The SVM in general provided a solid prediction of the stock prices.

Furthermore, people may use a hybrid methodology that combines several models to improve forecasting accuracy compared to the use of one model. For example, ARIMA is a linear model commonly used to forecast a time series. Artificial neural networks (ANNs) are one of the alternatives to the linear model. When the two models are combined, people can take advantage of both models and obtain more accurate results (Zhang, G., 2003).

In this paper, we exam only how the simplest model, ARIMA, helps to predict the future stock prices.

1.4 Data Source

Two indices are analyzed in these paper. S&P/TSX Composite Index. The weekly historical data for past 5 years from 2015-01-01 to 2020 is downloaded from Yahoo Finance. Shanghai Stock Exchange Composite Index. The weekly historical data from December 2002 to 2003, and that from 2015-01-01 to 2020 is downloaded from Yahoo Finance.

2 Methodology

2.1 Basic Time Series Models

A **time series** can be defined as a series of random variables obtained according to the order they appeared in time. In general, a **stochastic process** refers to a collection random variables x_t indexed by time t $t = 0, \pm 1, \pm 2, ...$ Stock market data can be interpreted as both discrete and continuous. In the business hour, the data is continuous since it can be observed at any time during its opening hour. When we consider, for example, the daily closing prices during a week, the data is discrete since each price is a point recorded at a specific date of the week. In this paper, we mainly look at the adjusted closing prices on a weekly or daily basis. A time series is univariate when it contains data from only one single variable, and multivariate when it contains data from more than one variables. A simple form of the series is the white noise.

Definition 2.1. A series w_t of random variables, w_t, is called a **white noise** if it has a mean of 0 and a finite variance of σ^2

We denote this process as $w_t \sim wn(0, \sigma_w^2)$. Figure 1(a) illustrates a series of 200 iid random variables, wn, with mean 0 and $\sigma_w^2 = 1$ (R codes attached to the end of the paper). In this paper, we assume that the white noises are Gaussian white noises.

Definition 2.2. A **Gaussian white noise** is defined as $w_t \sim iidN(0, \sigma_w^2)$. Here, w_t are independent, identically distributed normal variables.

A white noise is a mixture of various types of oscillations. It can not be the ideal model representing other series. We can improve the smoothness by introducing two new models, **moving average** and **autoregression**.

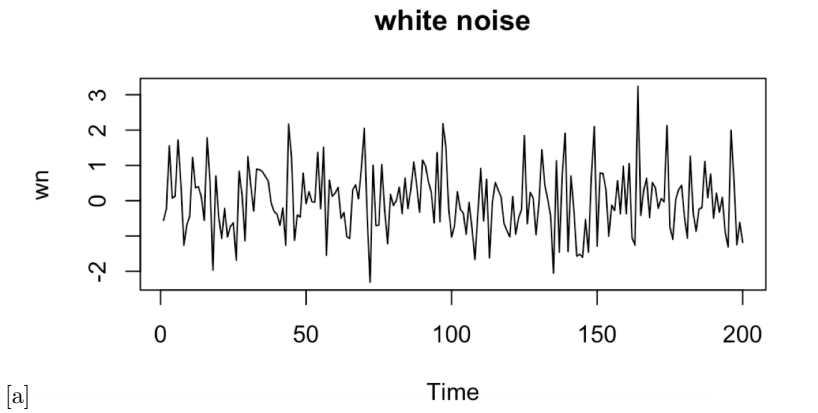
We can use a moving average model to smooth a white noise series. For example, we replace a white noise series $w_t \sim N(0, 1)$, denoted as wn, by a moving average model $ma = \frac{1}{3}(w_{t-1} + w_t + w_{t+1})$ (see Figure 1(b)). Such a linear combination of values is referred to as a filtered series.

If we consider the series wn as an input and calculate the output using $x_t = 0.5x_{t-1} + w_t$, we have an autoregression function. This equation is the current values as a regression of the past value and white noise of the series. Figure 1(c) illustrates a simple autoregression of 250 data with a coefficient of 0.5. Here, we ignore the initial 50 values to minimize the startup problem, which occurs when the data depend on the initial conditions x_0 and x_{-1}. Two ways to simulate a moving average or an autoregression data are to use the filter command or arima.sim command. To plot a time series data, we traditionally place the observed values on y-axis and time on the x-axis using the function plot.ts. Other functions to plot the series include ggplot, dygraph.

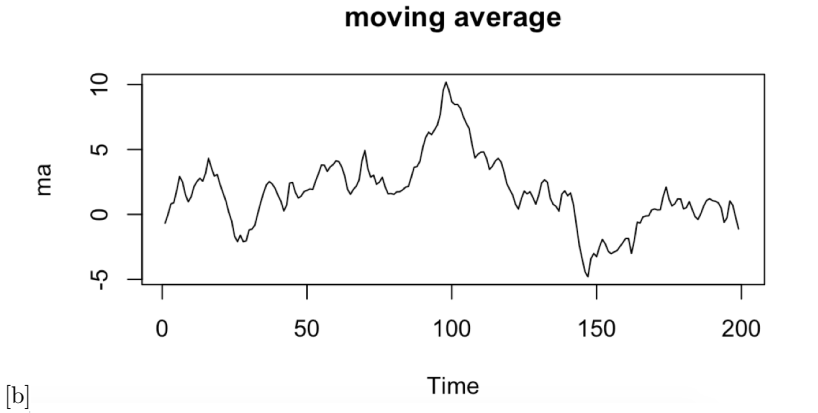
Another simple and classic time series model is the random walk model.

Definition 2.3. We define **random walk with drift** model as $x_t = \alpha + x_{t-1} + w_t$ for $t \in \mathbb{Z}_{\geq 0}$, where w_t represents the white noise, and α represents the drift. We obtain a **random walk** model when $\alpha = 0$

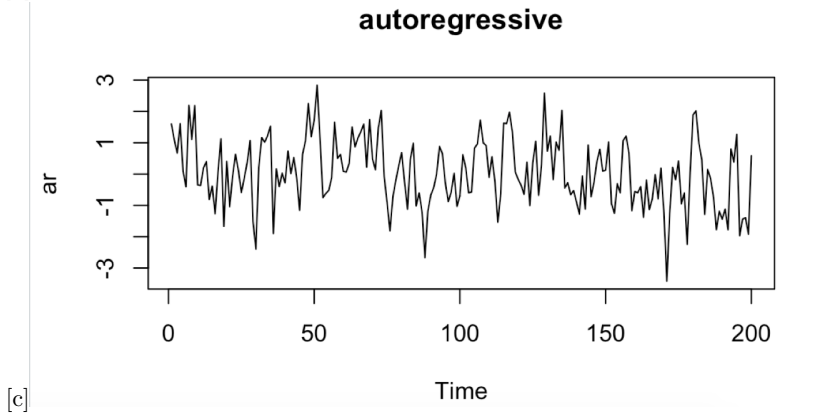
Here, the current value at time t is determined by the combination of the previous value at time t-1 and a random white noise movement. A random walk model of 200 observations is shown on Figure 1(d).



[a]



[b]



[c]

[d]

Figure 1: [a] A white noise with mean 0 and variance 1 of 200 observations. [b] A moving average of 200 observations. [c] An autoregressive model of 200 observations. [d] A random walk model with drift of 0.5 (black solid line), without drift (blue solid line).

A time series may have some outputs, x_t depend on the inputs values, y_{ti}, for $t = 1, ..., n, i = 1, ..., q$. In some time domain applications, for example, we may present x_t by a linear regression model

$$x_t = \sum_{i=1}^{q} \beta_i y_{ti} + w_t \tag{1}$$

where $\beta_i, i = 1, ..., q$ are unknown regression coefficients, w_t is the white noise. It is easy to forecast future values when the predictions models are expressed in this way. Suppose we consider a regression model with m coefficients. The Maximum Likelihood Estiomator (MLE) for σ_w^2 in this model is denoted as

$$\hat{\sigma}_m^2 = \frac{SSE_m}{n}$$

where SSE_m is the residual sum of squares (RSS) under the model with k coefficients. We can measure the goodness of fit for model (1) by balancing the number of parameters and the error of fit.

Definition 2.4. Akaike's Information Criterion (AIC) is defined as:

$$AIC = log\ \hat{\sigma}_m^2 + \frac{n + 2m}{n} \tag{2}$$

where $\hat{\sigma}_m^2$ is the MLE defined above, m is the number of parameters, n is the sample size. The AIC estimates the Kullback-Leibler discrepancy between the candidate and the true model. We choose the best model based on the value of m that yields lowest AIC. The issue is that as m increases, the AIC value decreases monotonically. To reduce the error variance, we consider a corrected model that is based on small sample distributions for the linear regression model.

Definition 2.5. The corrected model, **AIC, Bias Corrected (AICc)** is defined as

$$AICc = log\ \hat{\sigma}_m^2 + \frac{n + m}{n - m - 2} \tag{3}$$

When we search for the best model using the R function *auto.arima*, the function suggests the optimal model based on the one with the lowest AICc value.

Similarly, we can find the best model based on a minimum Bayesian argument.

Definition 2.6. Bayesian Information Criterion (BIC), also known as Schwarz Information Criterion (SIC), is defined as:

$$BIC = log\ \hat{\sigma}_m^2 + \frac{m\ logn}{n} \tag{4}$$

BIC measures the trade-off between model fit and its complexity. It is derived from a large sample approximation to the Bayesian model. While AICc usually does well in smaller samples which the number of parameters is relatively large, BIC usually works well in large samples.

We can also express a time series using Fourier transformation, or periodic variations of some phenomenon that produces the series, if we treat sines and cosines as inputs.

2.2 Components

A time series is determined by a combination of trend, seasonal, cyclical, and random variations. Seasonal variations are the fluctuations in the data within a specific period of time. Several factors may cause such fluctuation. Cyclical variations often refer to the changes within economic or business cycles. A typical business cycle consists of four phases: expansion, recession, depression, and recovery, as shown in Figure 2. In the expansion stage, GDP first rapidly increases and then reaches the peak where the growth rate approaches zero. After GDP growth rate is below zero, the business enters the recession stage. Recession can create shocks in stock markets. Once the GDP drops to the point where the change in the GDP becomes positive, it passes the trough point and enters the recovery and then expansion stages.

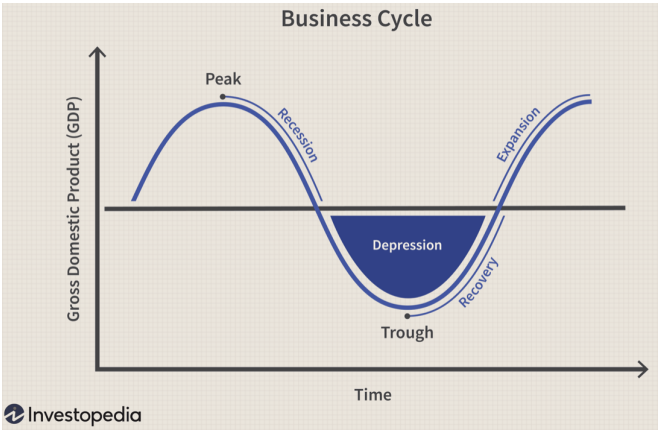


Figure 2: Four Stages of Business Cycle With Peak and Trough.

2.3 Stationarity

Stationarity is an essential property of a time series that does not change over time. Successful applications of many statistical tests and analytical tools depend on the stationarity of the data. The most common ways of checking for it are by visualization and parametric tests. For time series, x_s, with a finite variance to be defined as weakly stationary.

Definition 2.7. A time series, x_s, with finite variance is **weakly stationary** if it satisfies two conditions:

(i) its **mean value function**, defined as

$$\mu_{xs} = E(x_s) = \int_{-\infty}^{\infty} x f_s(x) ds \tag{5}$$

where $E(x_s)$ is the expected value of the series, independent of s.

(ii) its **autocovariance function**, defined as

$$\gamma(i) = cov(x_{s+i}, x_s) = E[(x_{s+i} - \mu_{s+i})(x_s - \mu_s)] \tag{6}$$

only depends on i - the difference between $s + i$ and s.

The autocovariance function measures the linear dependence between two points in a series at different times. The autocovariance is large for smooth series even when two points are far apart, while it is small, nearly zero for oscillating series.

A time series is considered **strictly stationary** if it satisfies the following relationship:

$P\{x_{s_i} \leq c_i\} = P\{x_{s_{i+h}} \leq c_i\}$,

$\forall 1 \leq i \leq k$ where $k \in \mathbb{Z}_{>0}$, for all time points s_i, $\forall c_i$, all time shifts $h \in \mathbb{Z}$. However, such criteria are usually too strong for most applications. Thus, we deal with weakly stationary time series in this paper and refer the term stationary as weakly stationary.

2.3.1 Visualization

The most basic method is to plot the data and determine whether the plot shows any signs of stationary data. Although it is sometimes difficult to detect a stationary process by simply looking at the graphs, we are able to rule out some apparent non-stationary process. Consider the following plots (Figure 3) extracted from *Forecasting Principles and Practice* by Hyndman & Athanasopoulos (2018). We can see that series (d), (h), and (i) follow prominent seasonality; series (a), (c), (e), (f), and (i) have noticeable trends; series (i) has an increasing variance. We can then conclude that series (b) and (g) are stationary series.

The second method is to examine the autocorrelation function plot.

Definition 2.8. The **autocorrelation function (ACF)** of a stationary process is defined as

$$\rho(i) = \frac{\gamma(s+i, s)}{\sqrt{\gamma(s+i, s+i)\gamma(s, s)}} = \frac{\gamma(i)}{\gamma(0)} \tag{7}$$

where the autocovariance function is defined as $\gamma(i) = cov(x_{s+i}, x_s) = E[(x_{s+i} - \mu)(x_s - \mu)]$ for a stationary time series process.

If a series is stationary, we may estimate its mean function, μ_s, using the sample mean $\overline{x} = \frac{1}{n}\sum_{s=1}^{n} x_s$, where $\mu_s = \mu$ is a constant. Also, we may estimate the autocovariance function using the sample autocovarianace function.

Definition 2.9. A **sample autocorrelation function (PACF)** is defined, similar to definition of ACF, as

$$\hat{\rho}(i) = \frac{\hat{\gamma}(i)}{\hat{\gamma}(0)} \tag{8}$$

where

$$\hat{\gamma}(i) = \frac{1}{n}\sum_{s=1}^{n-i}(x_{s+i} - \overline{x})(x_s - \overline{x}) \tag{9}$$

is the sample autocovariance function with $\hat{\gamma}(-i) = \hat{\gamma}(i)$ for $i = 0, ..., n-1$.

Autocorrelation refers to the correlation of a signal with a lag, a delayed signal. It is a function of the lag. When plotting ACF with increasing lags, the values are positive and tend to slowly lay down to zero for non-stationary data while fluctuating around zero for stationary data (see Figure 4 below for an example of Google stock price).

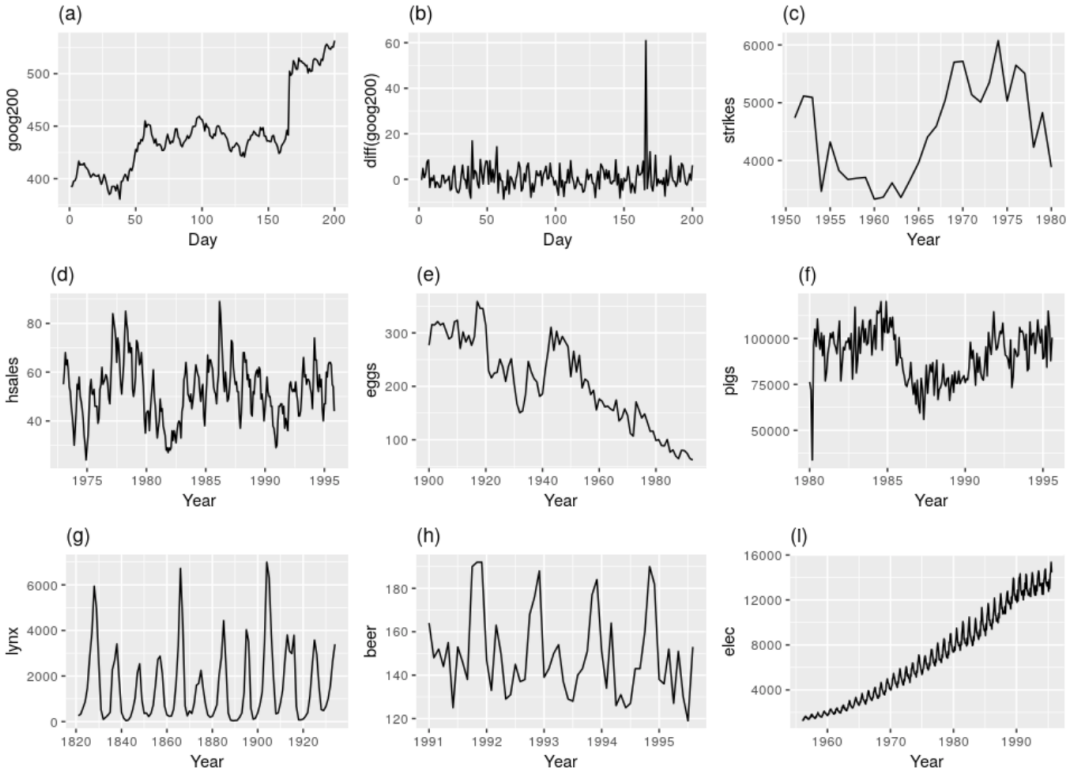


Figure 1: Nine examples of time series data; (a) Google stock price for 200 consecutive days; (b) Daily change in the Google stock price for 200 consecutive days; (c) Annual number of strikes in the US; (d) Monthly sales of new one-family houses sold in the US; (e) Annual price of a dozen eggs in the US (constant dollars); (f) Monthly total of pigs slaughtered in Victoria, Australia; (g) Annual total of lynx trapped in the McKenzie River district of north-west Canada; (h) Monthly Australian beer production; (i) Monthly Australian electricity production. [Hyndman & Athanasopoulos, 2018]

Figure 3: Examples of time series data.

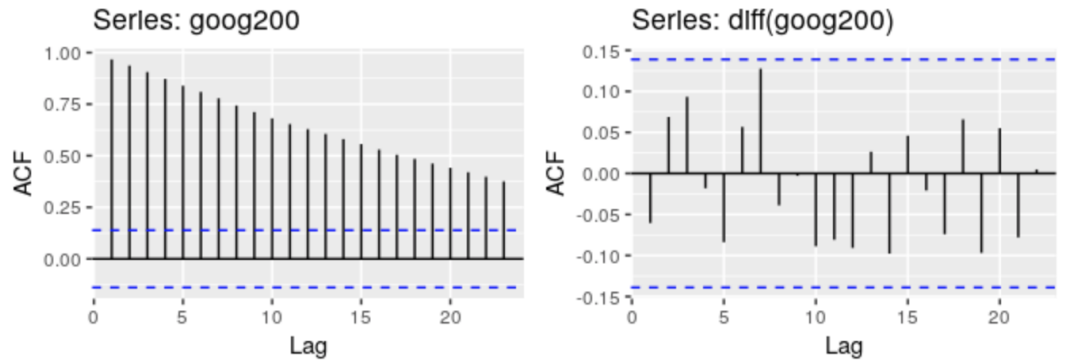


Figure 4: Left: The ACF of a non-stationary data - Google stock price. Right: The ACF of a stationary data - daily changes in Google stock price.

2.3.2 Parametric Tests

Another method to test for specific types of stationarity is to use statistical tests. Some useful parametric approaches include Augmented Dickey-Fuller test (ADF), Phillips-Perron (PP) Test, and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.

The ADF Test

The DF test was developed by David Dickey and Wayne Fuller in 1979. It tests the hypothesis that a unit root is present in an autoregressive model. Contrast to most hypothesis tests, the null hypothesis of the Dickey-Fuller test is non-stationary while the alternative hypothesis is stationary or trend-stationary (see Shumway & Stoffer, 2017, page 277-280). Consider a simple AR(1) process (autoregressive model with order 1, see Section 2.4.1 for detailed definitions and explanations),

$$x_t = \phi x_{t-1} + w_t. \tag{10}$$

We want to test whether the process is a random walk model (the null hypothesis) or a causal process - trend stationary (the althernate hypothesis). That is, we are testing,

$$H_0 : \phi = 1 \ (unit\ root\ presents)\ versus\ H_a : |\phi| < 1 \ (no\ unit\ root).$$

Under null hypothesis, the model is a random walk, that is, $x_s = x_{s-1} + w_s$. We can use the least square estimator (LSE) of ϕ to study $(\phi - 1)$ under the null. That is,

$$\hat{\phi} = \frac{\sum_{i=2}^{n} x_i x_{i-1}}{\sum_{i=1}^{n} x_i^2} \tag{11}$$

with

$$n(\hat{\phi} - 1) \xrightarrow{d} \frac{\frac{1}{2}(\chi_1^2 - 1)}{\int_0^1 B^2(s)ds}$$

where $\{B(s) : s \geq 0\}$ is a Brownian motion and χ_1^2 is a chi-square distribution with 1 degree of freedom.

Definition 2.10. A **standard Brownian motion** is a continuous process $\{(B(s) : s \geq 0\}$ which satisfies:

(i) B(0) = 0;
(ii) B follows independent increments;
(iii) $B(s + \Delta s) - B(s) \sim N(0, \Delta s)\ for\ \Delta s > 0$.

The ADF test has three main versions, and each version has its own critical value and the power of the test. When we test for a unit root, we use the model

$$\Delta y_t = \delta y_{t-1} + w_t.$$

When we test for a unit root with drift, we use

$$\Delta y_t = \alpha_0 + \delta y_{t-1} + w_t.$$

When we test for a unit root with drift and deterministic time trend, we use

$$\Delta y_t = \alpha_0 + \alpha_1 t + \delta y_{t-1} + w_t,$$

where α is a constant, δ is the drift, w_t is the white noise. If we set $\alpha_1 = 0$, and under the null hypothesis, we obtain a random walk with a drift α_0.

When we deal with more complex model such as

$$\Delta y_t = \alpha_0 + \alpha_1 t + \delta \Delta y_{t-1} + ... + \delta \Delta y_{y-p+1} + w_t,$$

we use the ADF test, where α_0, α_1 are constants, and p is the lagged order of the AR process.

Once we obtain the critical value and test statistic - t-test, we have decision whether to accept or reject the null hypothesis. We reject the null and conclude that the process is stationary when $t < t_{critical}$. We fail to reject the null and conclude that the process is non-stationary when $t > t_{critical}$

The PP Test

The Phillips-Perron unit root test become popular in financial time series analysis. Contrast to the ADF test which uses an autoregression model, the PP test ignores serial correlations. The model for the PP test is

$$\Delta y_t = \beta D_t + \pi y_{t-1} + w_t.$$

The test statistics are denoted by Z_t and Z_π with

$$Z_t = (\frac{\hat{\sigma}^2}{\hat{\lambda}^2})^{1/2} \times t_{\pi=0} - \frac{1}{2}(\frac{\hat{\lambda}^2 - \hat{\sigma}^2}{\hat{\lambda}^2}) \times (\frac{T \times SE(\hat{\pi})}{\hat{\sigma}^2}) \; and \; Z_\pi = T\hat{\pi} - \frac{1}{2}\frac{T^2 \times SE(\hat{\pi})}{\hat{\sigma}^2}(\hat{\lambda}^2 - \hat{\sigma}^2).$$

Under the null hypothesis ($\pi = 0$), both test statistics have the same asymptotic distributions as the ADF test. Contrast to the ADF test where we need to specific the lag value, the PP test does not require a lag length.

The KPSS Test

Constrast to the ADF test, the KPSS tests for a null hypothesis of a trend-stationary time series against the alternative of a unit root. If $t < t_{critical}$, the series is non-stationary. The test result is often misinterpreted when people check for the stationary. It can be the reason why the KPSS test is less popular than the ADF test.

2.3.3 Decomposition

We can decompose a time series into trend, seasonal effects, and random error. A trend is the variations, or increases and decreases during a long-term period of time. It is the general tendency of the series. One way to illustrate a trend, the extracted signal, is via linear filters, $m_t = \sum_{j=-\infty}^{\infty} \lambda_j x_{t+1}$. Consider the monthly airline passengers from 1949 to 1960 shown in Figure 5 (Holmes, Scheurell, & Ward, 2020). The smaller λ is, the smoother the trend line becomes (Figure 6). If the λ is small enough, we will end up with a straight line.

We mentioned using a moving average model to smooth a white noise series in 2.1. This method is useful to discover long-term trend or seasonal components. In some applications, we may also apply a a periodic regression smoother to bring out the seasonal component and trends. Another approach is to use nearest neighbor regression to smoothing a time series plot.

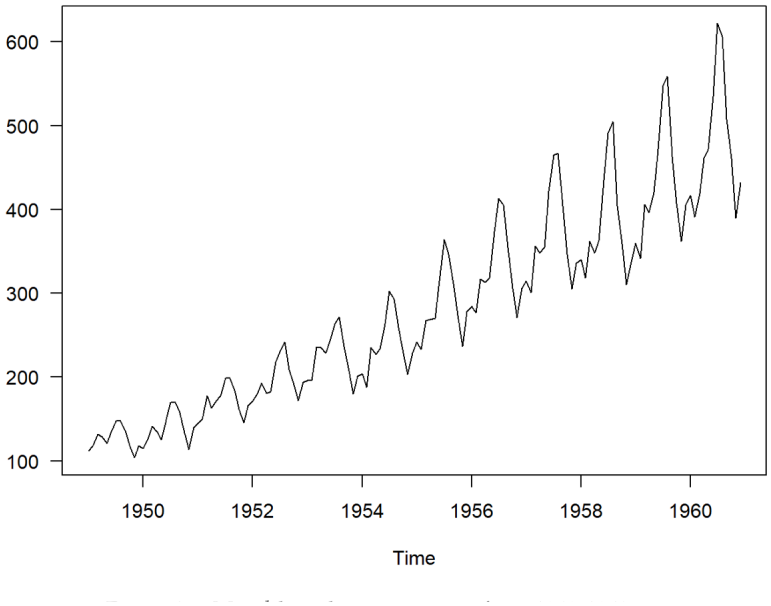


Figure 5: Monthly airline passengers (Holmes, Scheurell, & Ward, 2020).

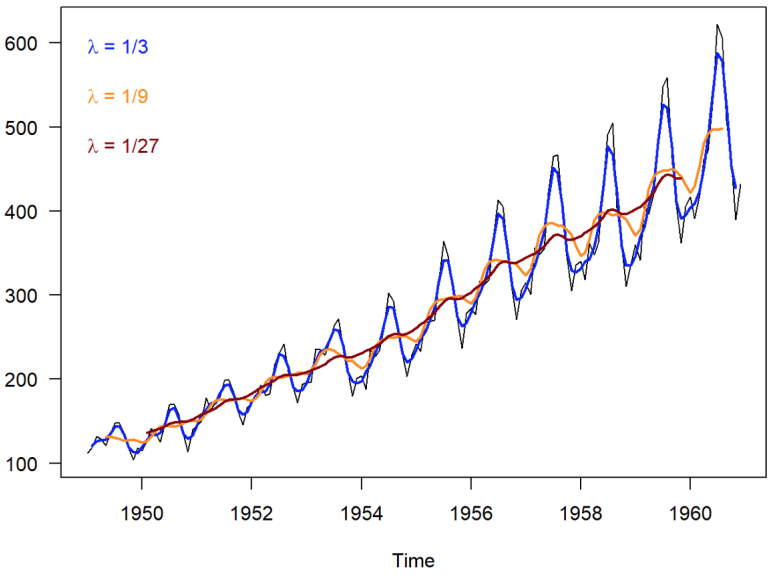


Figure 6: Monthly airline passengers with different filter values.

A seasonal effect is the repeated short-term cycle in the series. Some data may exhibit seasonal trends, that is, lags at an integer multiple of a period t.

These seasonal effects can be removed by, for example, differencing.

2.3.4 Difference Equations

Difference equations are the discrete version of linear differential equations that are sometimes the ideal model for certain series. They can be used to remove trends on a regression model.

Definition 2.11. We define **backshift operator** as

$$\Delta x_t = (1 - B)x_t \tag{12}$$

and **differences on order d** as

$$\Delta^d x_t = (1 - B)^d x_t \tag{13}$$

for any positive integer values of d.

Consider a model $x_t = \beta_1 + \beta_2 t + y_t$. We apply the first difference and the resulting equation is:

$$\Delta x_t = x_t - x_{t-1} = \beta_2 + y_t - y_{t-1}. \tag{14}$$

The first difference equation eliminates a linear trend, while the second difference - the difference of the first difference equation - eliminates a quadratic trend.

2.3.5 Logarithmic Transformations

We sometimes perform a logarithmic transformation to make the distributions less skewed and smooth a time series. A non-linear model can be transformed into a linear one by log transformations. The airline passengers data has a non-linear trend and increase its variability over time (see Figure 7). After a log transformation is applied, the new series increases following a linear trend, and the variability of the data seem to become constant over time.

2.3.6 Power Transformations

Some transformations are effective in reducing the variations across time. Such transformations include taking the square ($x :\to x^2$) or performing a square root ($x :\to \sqrt{x}$) of the variables. The former approach can be used to reduce left skewness. Squaring usually make sense if the variables is non-negative, given that x^2 and $(-x)^2$ produce identical results. The latter approach may have moderate effects on the distribution but usually works for data with non-constant variances.

2.4 ARIMA Model

Classical regressions are sometimes not sufficient. When we deal with nonstationary time series data, the time domains approach are more appropriate compared to the classical regression theory. More

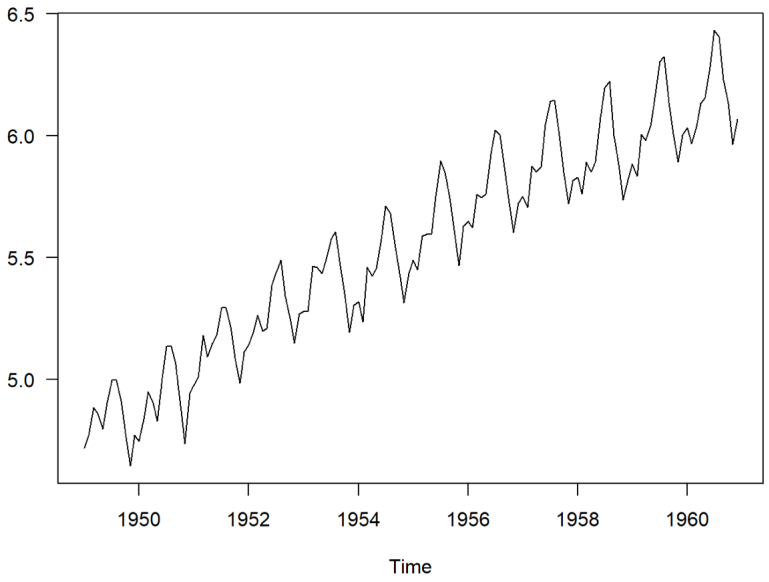


Figure 7: Monthly airline passengers in log.

advanced models such as autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models may be better fit and more accurate. In classical regressions, only current values of independent variables can influence the values of dependent variables. However, in time series, we allow past values of independent variables to influence the values of dependent variables. It becomes possible to forecast the future outputs if we can model the present data using solely past values of the independent variables.

2.4.1 AR Models

We can express the value of a series as a linear combination of its past value.

Definition 2.12. We denote **AR(p)** as an order p **AR model** by

$$x_s = \sum_{i=1}^{p} \phi_i x_{s-i} + w_s \tag{15}$$

where $w_s \sim N(0, \sigma^2)$ and $\phi_i \neq 0$ are constant.

We can rewrite the model using the backshift operator as:

$$w_s = (1 - \sum_{i=1}^{p} \phi_i B^i) x_s \tag{16}$$

Definition 2.13. An **autoregressive operator** is the linear combination of all $\phi_i B^i$, that is,

$$\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i \tag{17}$$

If we set $\phi(B) = 0$, we obtain a **characteristic equation**.

Consider the AR(1) model, $x_s = \phi_0 + \phi_1 x_{s-1} + w_s$. Set $\phi_0 = 0$ for a linear process with y-intercept at 0. We then apply backward iterations j times, we obtain:

$$x_s = \phi_1 x_{s-1} + w_s = \phi_1(\phi_1 x_{s-2} + w_{s-1}) + w_s = ... = \phi_1^j x_{s-j} + \sum_{j=0}^{j-1} \phi_1^j w_{s-j} = \sum_{j=0}^{\infty} \phi_1^j w_{s-j} \tag{18}$$

Assume the series is weakly stationary, from Equations (7) and (8), we know that $E(x_j) = \mu$, and $cov(x_{t+j}, x_t) = \gamma(j) \forall j$. So, the AR(1) process has the mean of

$$E(x_s) = \sum_{j=0}^{\infty} \phi_1^j w_{s-j} = 0;$$

the autocovariance function of

$$\gamma(j) = \frac{\sigma_w^2 \phi_1^j}{1 - \phi_1^2}, j \geq 0;$$

the ACF of

$$\rho(j) = \frac{\gamma(j)}{\gamma(0)} = \phi_1^j;$$

and the PACF of

$$\phi(j) = \phi_{jj} = 0 \ for \ j > 1, \ and \ \phi(1) = \phi_{11} = \phi$$

For the model to be truly weakly stationary, $|\phi_1| < 1$.

2.4.2 MA Models

While in a AR model, x_s is a linear combination of itself; in a MA model, x_s is a linear combination of the white noise w_s.

Definition 2.14. We denote **MA(q)** as an order q **MA model** by:

$$x_s = w_s + \sum_{i=1}^{q} \theta_i w_{s-i}, \tag{19}$$

where $\theta_q \neq 0$ are constant parameter, and w_s is a Gaussian white noise. Or by the backshift operation,

$$x_s = (1 + \sum_{i=1}^{q} \theta_i B^q) w_s. \tag{20}$$

Definition 2.15. An **moving average operator** is the linear combination of all $\theta_i B^q$, that is,

$$\theta(B) = 1 + \sum_{i=1}^{q} \theta_i B^i. \tag{21}$$

Consider the simplest model MA(1), given by $x_s = w_s + \theta w_{s-1}$. The mean is $E(x_s) = 0$, the autocovariance function is $\gamma(1) = \theta\sigma_w^2$ and $\gamma(0) = var(x_s) = (1 + \theta^2)\sigma_w^2$. The ACF for this model is

$$\rho(s) = \frac{\theta}{(1 + \theta^2)}, s = 1.$$

and the PACF is

$$\phi(s) = \phi_{ss} = -\frac{(-\theta^s)(1-\theta^2)}{1-\theta^{2(s+1)}},\ s \geq 1.$$

For the generalized model MA(q) defined on Equation (19), the mean is $E(x_s) = 0$ and the variance $var(x_s) = (1 + \theta_1^2 + ... + \theta_q)^2\sigma_w^2$.

2.4.3 ARMA Models

Definition 2.16. We denote **ARMA(p,q)** as the general expression of an order (p,q) ARMA model by:

$$x_s = \alpha + \sum_{k=1}^{p} \phi_k x_{s-k} + w_s + \sum_{k=1}^{q} \theta_i w_{s-k} \tag{22}$$

where w_k is Gaussian white noises, and p is the AR order and q is the MA order, $\theta_p \neq 0$ and $\phi_p \neq 0$. Similar to AR and MA models, we can express the ARMA model using the backshift operation and obtain

$$(1 - \sum_{k=1}^{p} \phi_k B^k)x_s = (1 + \sum_{k=1}^{q} \theta_k B^k)w_s. \tag{23}$$

We take the ARMA(1,1) as an example. The ACF for the process $x_s = \phi x_{s-1} + w_s + \theta w_{s-1}$ is

$$\frac{(1+\theta\phi)(\theta+\phi)}{1+2\theta\phi+\theta^2}\phi^{k-1}, k \geq 1.$$

We can identify AR and MA models using the ACF and PACF plots. For a AR(p) model, the ACF slowly decreases and PACF cuts off after lag p. For a MA(q) model, the PACF slowly goes down and ACF cuts off after lag q. For a ARMA(p,q) model, both the ACF and PACF tails off.

2.4.4 ARIMA Models

A time series may have two trends, a zero-mean stationary one and a nonstationary one. In this case, we need to difference the series to produce a stationary process.

Definition 2.17. We denote a model as **ARIMA(p,d,q)** if it satisfies the equation

$$\phi(B)(1-B)^d x_s = \theta(B)w_s \tag{24}$$

where $\Delta^d x_s = (1-B)^d x_s$.

Consider a simple process

$$x_s = (1 + i_s)x_{s-1}$$

which is a growth model of an investment, where i_s is the investment interest rate from time $s-1$ to s. Apply log on both sides, we obtain

$$log(x_s) = log(1 + i_s) + log(x_{s-1}) \implies \Delta log(x_s) = log(1 + i_s) \approx i_s$$

Here, $log(x_s)$ is the growth rate, which is relatively stable. Thus, we frequently apply log transformations to smooth and stabilize an economic or financial time series.

2.4.5 Box-Jenkins Method

The Box-Jenkins Method in time series is to choose the best forecasting model to capture the patterns of an observed data through several iterations. It follows three steps (Aljandali, 2017):

1. Model selection
2. Parameter/Model estimation and verification
3. Model forecasting

3 Application in Stock Market Analysis

3.1 S&P/TSX Composite Index

We began by analyzing the S&P/TSX Composite Index. It includes 250 companies trading in and accounts for 70% of total capitalization in the Toronto Stock Exchange. The weekly historical data for the past five years, in CSV format, is downloaded from Yahoo Finance. We used the Date and AdjClose (Adjusted closing) columns to analyze the behaviors.

3.1.1 Stationarity

First, the weekly stock prices were plotted using the dygraph() and ts.plot() functions. From the plots, it was clear that the data set followed an upward trend, which indicated that the series was non-stationary (Figure 8). To confirm the observation, several statistical tests are performed to determine the non-stationarity (see Table 1)

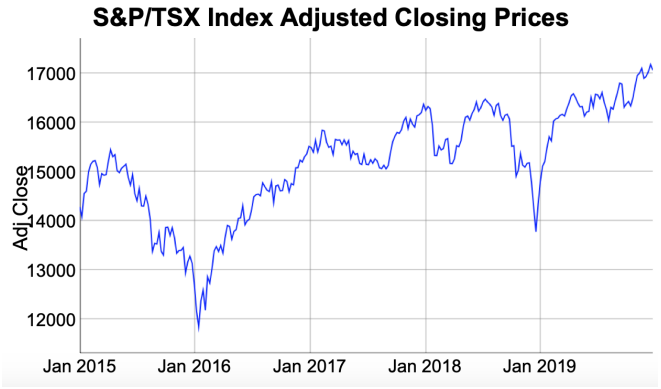


Figure 8: S&P/TSX Index

Test statistics	Lag order/ Truncation lag parameter	p-value
Augmented Dickey-Fuller Test	lag order = 6	0.2447
Augmented Dickey-Fuller Test	lag order = 0	0.3714
Phillips-Perron Unit Root Test	truncation lag parameter = 5	0.4081
KPSS Test for Trend Stationarity	truncation lag parameter = 5	0.01237

Table 1: Statistical Tests

Based on the ADF test and PP Test, the stock price was non-stationary since the p-values were greater than 0.05. The KPSS test for trend stationarity indicated that the null hypothesis of stationarity around a trend was rejected, and the stationarity was not supported. All the results support the graphical observations that the adjusted closing index was non-stationary.

To make the dataset stationary, we applied difference equations and log transformations. Figure 9 shows the new logarithmic series with seasonality and trends removed. The ADF test for the transformed dataset had a p-value below 0.01, which meant the series was now stationary.

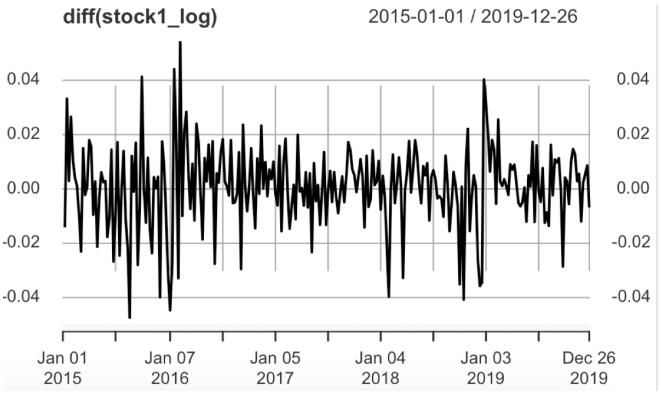


Figure 9: S&P/TSX Index after differencing and log transformation.

3.1.2 Model fitting

We then searched for an optimal model that captured the changes in the log-transformed data set using ACF and PACF plots. We first separated the dataset into a training set and a test set. The training set was used to determined the best fit model, while the test set was used to verify the model. The ACF and PACF for the training set is shown in Figure 10. We note that the ACF is slowly decaying and the PACF cuts of after lag 1. Two plots indicate a nonstationary behavior.

We took a first difference equation, and the ACF and PACF plot did not seem to follow any AR or MA model (Figure 11). We used the auto.arima() function to search for the best model, which was found to be ARIMA(0,1,0), or a random walk model.

The diagnostics for the fit are shown in Figure 12. The null hypothesis in Ljung-Box Q statistics states that the data are independently distributed, or has no correlation. The null indicates that model is lack of fit. Based on ACF of residuals, only a small portion of autocorrelation fell outside the critical range, the variations of the residuals were close to zero across the data and there was no correlation in the residuals series. Therefore, we can treat the residual variance as constant. If two sets of quantiles are from the same distribution - normal distribution, the points will form a roughly straight line. Based on the Normal Q-Q Plot, a few outliers existed in both ends of the plot but the straight line appeared to be a good fit. Ljung-Box Q statistic were mostly above the critical value so we rejected the null and concluded that the model was a good fit.

We also considered both squaring the variables and taking the square root to reduce the spike happening around 2016. The result did not differ much from that of the logarithmic transformations. Based on the R output, the best fit model in both cases were also ARIMA (0,1,0).

3.1.3 Forecasting

Finally, forecasts for four months based on the fitted model and the actual values are shown in Figure 13. The black solid line represents the actual stock prices. The points on the red fitted line are the predicted values. The blue dashed lines are the upper and lower bound of the prediction regions. The fitted model exhibits a slight downward trend with an increasing prediction region. The actual data falls within the upper prediction region and moves upward.

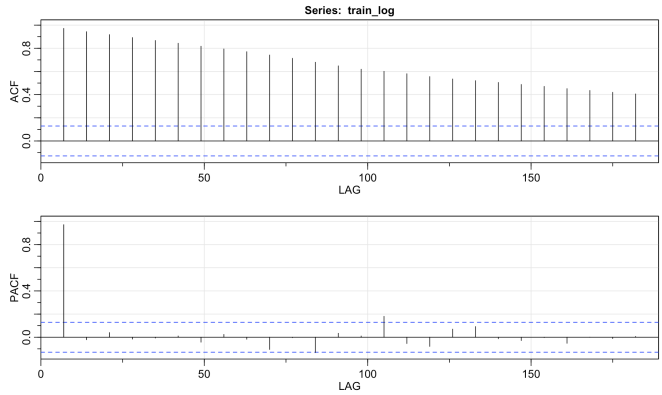


Figure 10: ACF and PACF for the training set.

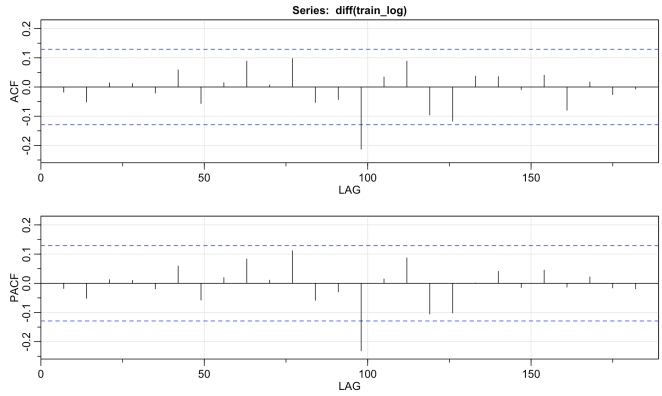


Figure 11: ACF and PACF plots for first difference of the training set.

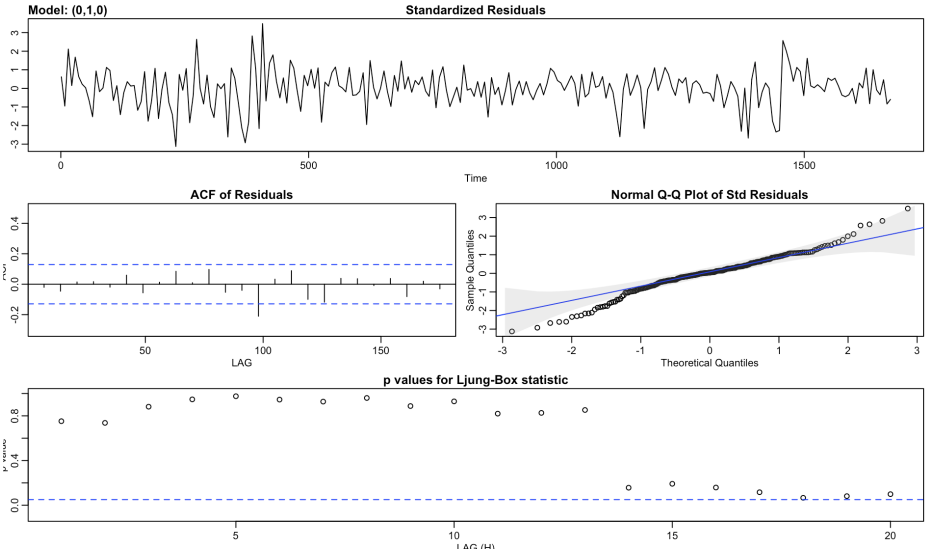


Figure 12: The diagnostics test for the fitted model of the training set.

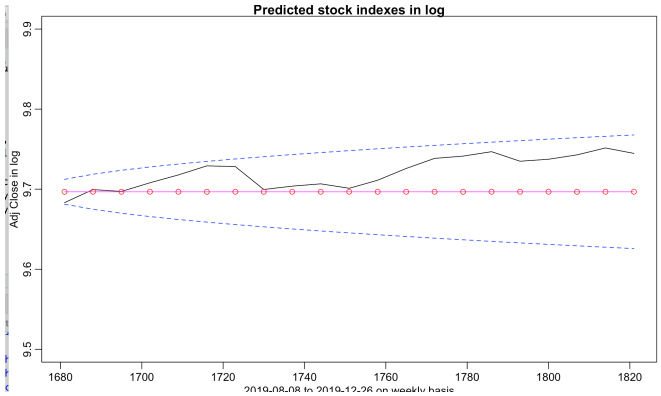


Figure 13: Forecasted and actual values of S&P/TSX Index from 2018-08-08 to 2019-12-26.

3.2 SSE Composite Index

We may use a similar approach to analyze the Shanghai Securities Composite Index. The SSE Index is a stock market index of stocks including A shares and B shares, traded in the Shanghai Stock Exchange. The SSE is one of the two stock exchanges operating independently in China and one of the world's largest stock markets by market capitalization. During the outbreak of COVID-19, many factories were closed for an extended period, and people have started to return to work recently. The stock market will be impacted during this time. To model the stock trend, we looked at the market values in early 2003 after the outbreak of SARS. The weekly historical data from December 2002 to August 2003, denoted as the training set, were used to develop an ARIMA model.

First, we checked the training set for stationarity. Based on both the original plot (Figure 14) and ACF plots (Figure 17), the ACF slowly decreased while the PACF cut off quickly and oscillated slightly around zero. We could tell that the original data set was nonstationary. Thus, we first attempted to remove the nonstationary part using log transformation and differencing equation (Figure 16). The ACF and PACF plots for the first difference did not seem to follow behaviors of AR or MA model (Figure 18). We tested for a best-fit model using the auto.arima() function and noted the model to be ARIMA(0,1,0).

Figure 19 illustrated the diagnostics for the fit. The ACF of residuals were within the critical region which indicated no correlation in the residuals series, so we can assume the variance were constant. Points in the Q-Q plot were on the fitted line indicating the normality of the distribution. The p-values for Ljung-Box Q statistics were all above 0.05, so we rejected the null and concluded that the model ARIMA(0,1,0) was a good fit. We also took the square and the square root of the variables to reduce the spikes, and noted that the best-fit model remain unchanged.

We then used the ARIMA(0,1,0) model to forecast the SSE Composite Index for four months starting from December 2019 to March 2020. The actual adjusted closing plot is shown in Figure 20. We applied the model on SSE Index from December 2019 to March 2020 and make the prediction as shown in Figure 21.

The predicted values and historical data is shown on the same plot. It seems the model is not ideal. It does capture the decreasing trend of the original data. Only predictions around March 2020 fall on the actual lines.

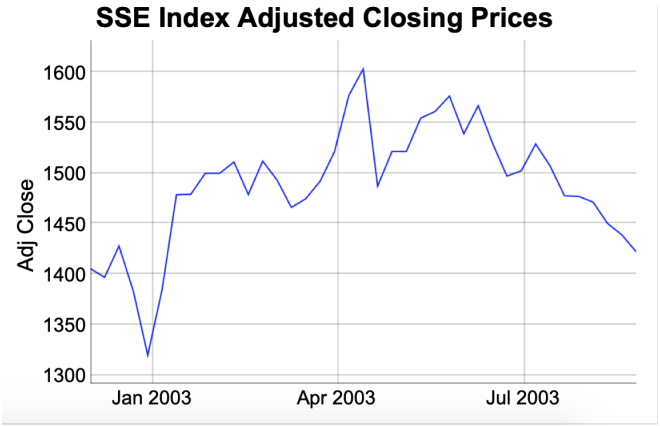


Figure 14: Actual Adjusted Closing from December 2002 to August 2003

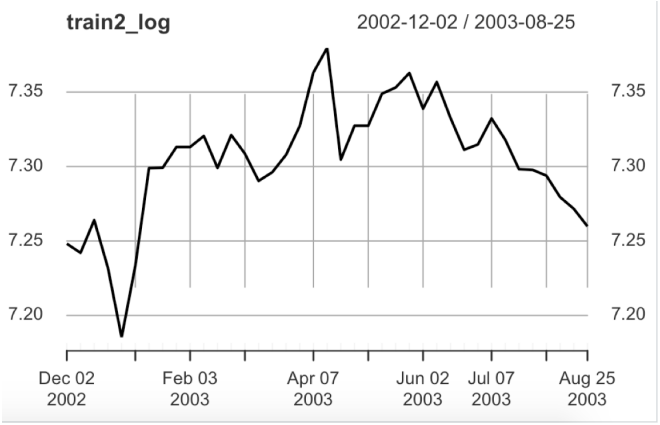


Figure 15: Log Adjusted Closing from December 2002 to August 2003

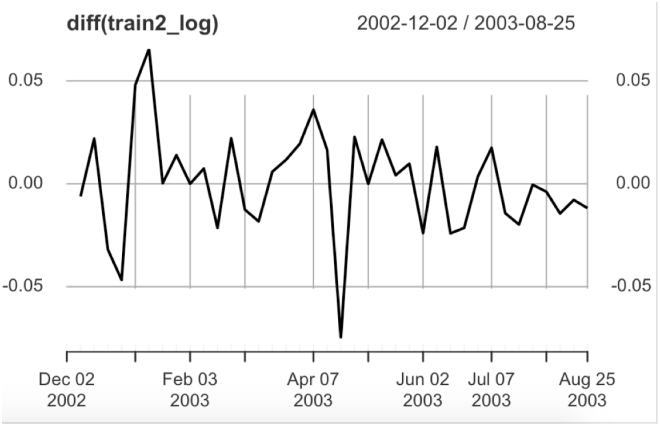


Figure 16: First Difference of Adjusted Closing from December 2002 to August 2003

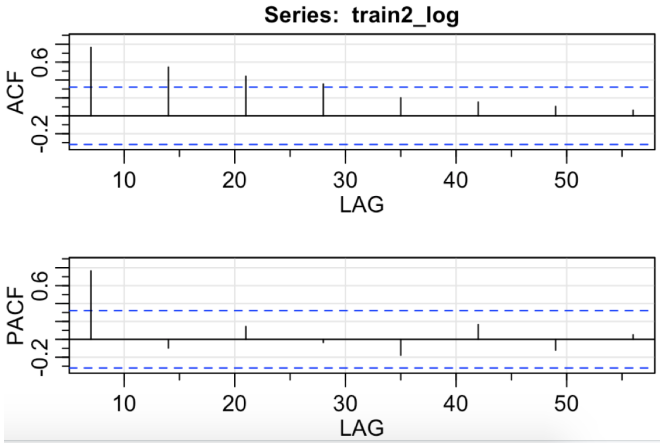


Figure 17: The ACF and PACF for log adjusted closing from December 2002 to August 2003

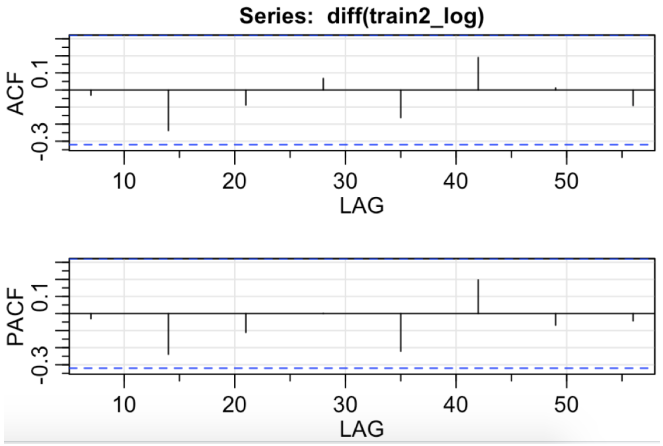


Figure 18: The ACF and PACF for differencing the log data.

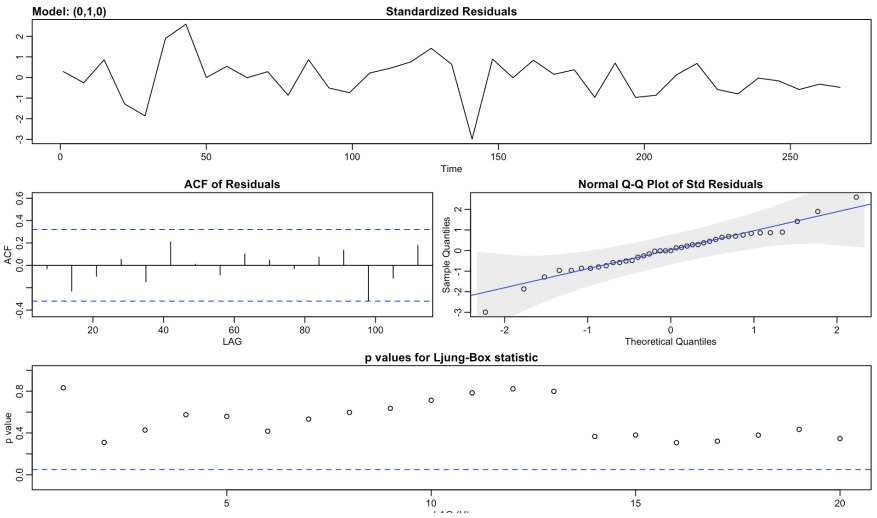


Figure 19: The diagnostics test for the fitted model of the training set.

Figure 20: Actual Adjusted Closing from 2015 to 2019.

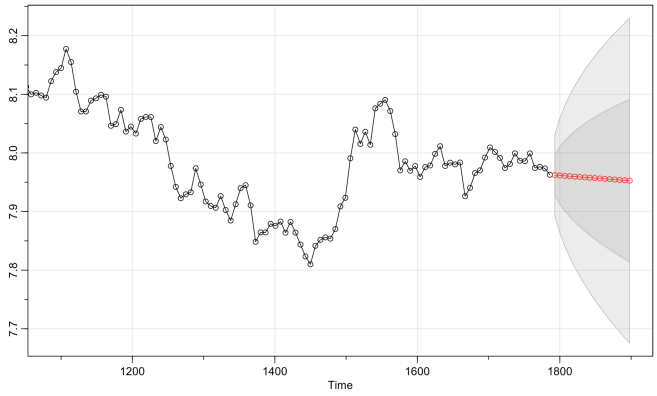


Figure 21: Log Adjusted Closing from 2015 to 2019 and forecast of future 16 weeks based on ARIMA(0,1,0) model

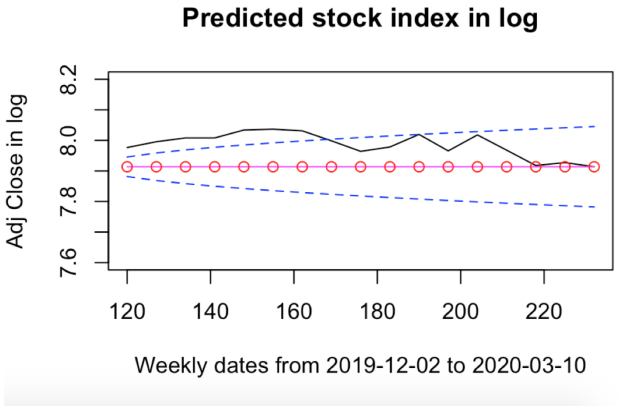


Figure 22: Predicted and actual values of log adjusted closings from 2019-12 to 2020-03. Black solid line is the actual data. Red dotted line is the predicted values. Blue dashed lines are the upper and lower bound of the prediction regions

4 Discussion

In this study, we attempted to forecast the stock indexes using simple time series models. We applied ARIMA models to investigate the trends of selected stock indexes. We collected weekly stock prices from the past several years and removed any possible patterns to produce stationary datasets. We used auto.arima techniques to determine the best fit model for each stock and visualized the future values based on the best-fit model. We discovered that both data are the random walk with drift model. The predictions, however, were not ideal. Although the actual data fell within the critical region, the predicted trends did not follow the actual ones. Thus, these models did not reflect the actual stock prices successfully. In particular, we were unable to remove the sudden bullish or bearish behaviors using simply log/differeing or power transformations. More advanced machine learning models may do a better job with stock market predictions. Additionally, in this paper we used weekly stock market data, which may be subject to higher volatility compared to monthly data. If we analyzed monthly data instead, we might obtain a better forecasting model. In order to derive a better forecasting model, we might also need to consider factors such as investor behaviors, economic conditions, other external factors and possible capital strengths. Furthermore, the stock indexes we chose in this study are were indexes, which means they were a mixture of stock prices of a large number of publicly traded companies. This makes it more challenging to analyze the overall trend because the price variations in different companies or industries may depend on or offset each other's price, and government policies are likely to affect the stock prices, making the future prices more unpredictable.

With the aid of advanced analytical models, we were able to forecast the future stock prices better. However, there are no such algorithms that can provide an accurate and successful prediction. In general, the stock price depends on demand and supply. When a person sells a stock, the seller and buyer exchange the ownership of the capital. The bid price, the price of the purchase of the stock, becomes the new market price. During some major or unexpected events, people may lose faith in certain stocks and start to sell the stock at a lower price; the stock price may and potentially plunge. One example is the US stock market crash that happened on March 2020. In this case, we unfortunately were unable to predict future events, and thus unable to develop a sound model. Many economists assume that people are rational beings. By extension, people will make decisions to buy or sell stocks rationally. When the stock prices start to rise, people will buy more of that stock. When the prices start to fall, people will sell more of that stock. However, based on social psychology, people can influence each other. If influential people or simply people around them start to buy or sell a stock, the majority of the investors may imitate such behaviors without actually analyzing the market. Groupthink may occur when people blindly follow what an influential person or the media proposes about the rise or fall of a stock, and make ask/bid decisions accordingly. These irrational behaviors will for sure impact the stock prices and make the prediction model less accurate.

In future works, we will consider analyzing a particular company stock instead of the market as a whole. We may also conduct research on investor behaviors and public opinions retrieved from social media. The combination of mathematical models developed from historical data and studies on behavioral finance may provide a better estimates and predictions of future prices.

5 References

Amadeo, K. (2020, March 17). Is the 2020 Stock Market Crash One of the Worst? https://www.thebalance.com/fundamentals-of-the-2020-market-crash-4799950. Accessed on April 5, 2020.

Aljandali A. (2017) The Box-Jenkins Methodology. In: *Multivariate Methods and Forecasting with IBM® SPSS® Statistics. Statistics and Econometrics for Finance* (pp. 57-79). Springer, Cham. doi: https://doi.org/10.1007/978-3-319-56481-4_3. Accessed on April 1, 2020.

Augmented Dickey–Fuller test. (2020, March 28). https://en.wikipedia.org/wiki/Augmented_Dickey–Fuller_test. Accessed on April 1, 2020.

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications, 80*, 340–355. doi: 10.1016/j.eswa.2017.02.044

Choi, I. (2015). *Almost all about unit roots: foundations, developments, and applications.* New York, NY: Cambridge University Press. e-ISBN 9781316157824.

Dickey–Fuller test. (2019, December 12). https://en.wikipedia.org/wiki/Dickey–Fuller_test. Accessed on April 1, 2020.

Dyrhovden, S. B. (2016, June). Stochastic unit-root processes. University of Bergen. http://bora.uib.no/bitstream/handle/1956/12650/144801902.pdf?sequence=1. Accessed on April 1, 2020

Hayem, M.O. Time Series & Forecasting (LEC) Fall 2019. https://carleton.ca/culearn/. Accessed on April 5, 2020.

Holmes, E. E., M. D. Scheuerell, & E. J. Ward. *Applied time series analysis for fisheries and environmental data.* NOAA Fisheries, Northwest Fisheries Science Center, 2725 Montlake Blvd E., Seattle, WA. https://nwfsc-timeseries.github.io/ atsa-labs/index.html#book-package. Accessed on April 1, 2020

Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. Accessed on April 1, 2020.

Kenton, W. (2020, March 2). Business Cycle. https://www.investopedia.com/terms/b/businesscycle.asp. Accessed on April 5, 2020.

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., ... Deng, X. (2014). Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications, 27(1).* doi: 10.1007/s00521-014-1550-z

Palachy, S. (2019, November 12). Detecting stationarity in time series data. https://towardsdatascience.com/detecting-stationarity-in-time-series-data-d29e0a21e638. Accessed on March 10, 2020.

Palachy, S. (2019, September 22). Stationarity in time series analysis. https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322. Accessed on March

10, 2020.

Shanghai Stock Exchange. (2020, March 19). https://en.wikipedia.org/wiki/Shanghai_Stock_Exchange. Accessed on April 5, 2020.

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: with R examples*, 3rd edition. New York: Springer. ISBN 978-1-4419-7864-6

Soni, N., Kumar, T. (2015). Cloud based Financial Market Prediction through Genetic Algorithms: A Review. *International Journal of Computer Applications*, 123(8), 18–20. doi: 10.5120/ijca2015905413

Unit Root Tests. (n.d.). https://faculty.washington.edu/ezivot/econ584/notes/unitroot.pdf. Accessed on April 5, 2020.

What are time series? definition and meaning. (n.d.). http://www.businessdictionary.com/definition/time-series.html. Accessed on April 1, 2020.

Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159–175. doi: 10.1016/s0925-2312(01)00702-0

6 R Codes

Libraries and Functions

```
library(ggplot2)
library(xts)
library(dygraphs)
library(astsa)
library(forecast)
library(caTools)
library(tseries)
library(fGarch)
read = function(x) {
    dat = read.csv(x, header = TRUE)
    df = dat[,c(1,6)]
    df$Date = as.Date(as.character(df$Date))
    return(df)
}
```

Methodology

```
graphics.off()
set.seed(123)
wn = rnorm(200,0,1)
ma = filter(wn, sides = 2, rep(1/3,3))
ar = arima.sim(list(order=c(1,0,0), ar=.5), n=250)

x=cumsum(wn); wd=wn+.5; xd=cumsum(wd)
plot.ts(xd,ylim=c(-10,100), main="random walk", ylab="with and without drift")
lines(x,col=4)
abline(h=0,col=4,lty=2)
abline(a=0,b=.5,lty=2)

plot.ts(wn, main="white noise")
plot.ts(ma, main="moving average")
plot.ts(ar[51:250], main="autoregressive", ylab="ar")
```

S&P/TSX Index

```
> stock1 <- read("SPTSE.csv")
> stock1_xts <- xts(stock1[,2], order.by = stock1\$Date, frequency=365)
> training_set <- stock1_xts[1:240, ]
> test_set <- stock1_xts[241:261, ]

> dygraph(stock1_xts, ylab="Adj Close",
+         main="S&P/TSX Index Adjusted Closing Prices") %>%
```

```
+   dySeries(label = "TSE", color = "blue")

> adf.test(stock1_xts)
Augmented Dickey-Fuller Test
    data:  stock1_xts
    Dickey-Fuller = -2.7865, Lag order = 6, p-value = 0.2447
    alternative hypothesis: stationary

> adf.test(stock1_xts,k=0)
Augmented Dickey-Fuller Test
    data:  stock1_xts
    Dickey-Fuller = -2.4855, Lag order = 0, p-value = 0.3714
    alternative hypothesis: stationary

> pp.test(stock1_xts)
Phillips-Perron Unit Root Test
    data:  stock1_xts
    Dickey-Fuller Z(alpha) = -12.515, Truncation lag parameter = 5,
    p-value = 0.4081
    alternative hypothesis: stationary

> kpss.test(stock1_xts)
KPSS Test for Level Stationarity
data:  stock1_xts
KPSS Level = 2.8392, Truncation lag parameter = 5, p-value = 0.01

> kpss.test(stock1_xts, null="Trend")
KPSS Test for Trend Stationarity
data:  stock1_xts
KPSS Trend = 0.20967, Truncation lag parameter = 5, p-value =
0.01237

> plot(stock1_xts)
> plot(diff(stock1_xts))
> acf2(stock1_xts)
> acf2(diff(stock1_xts))
> model2 <- auto.arima(training_set, trace = TRUE, approximation = FALSE) # Arima (0,1,0)
> sarima(training_set,0,1,0)
> sarima(stock1_xts,0,1,0)
> sarima.for(training_set, 21, 0,1,0)
> lines(pred$pred, type="p", col=2)
> lines(pred$pred+pred$se, lty="dashed", col=4)
> lines(pred$pred-pred$se, lty="dashed", col=4)
> ts.plot(test_log, pred$pred, col=c(1,6), ylim=c(1500, 1800),
+         ylab="Adj Close in log", main="Predicted stock indexes ",
+         xlab="2019-08-08 to 2019-12-26 on weekly basis")
> lines(pred$pred, type="p", col=2)
```

```
> lines(pred$pred+pred$se, lty="dashed", col=4)
> lines(pred$pred-pred$se, lty="dashed", col=4)
```

Log Transformation

```
> stock1_log <- log(stock1_xts)
> train_log = stock1_log[1:240, ]
> test_log = stock1_log[241:261, ]
> dygraph(stock1_log, ylab="Log Adj Close",
+         main="S&P/TSX Index Adjusted Closing in Log") %>%
+   dySeries(label = "TSX", color = "red")
> adf.test(stock1_log,k=0)

> plot(stock1_log)
> plot(diff(stock1_log))
> adf.test(na.omit(diff(stock1_log)))
> acf2(train_log)
> acf2(diff(train_log))

 ARIMA(2,1,2) with drift         : -1303.396
 ARIMA(0,1,0) with drift         : -1310.528
 ARIMA(1,1,0) with drift         : -1308.553
 ARIMA(0,1,1) with drift         : -1308.561
 ARIMA(0,1,0)                    : -1312.267
 ARIMA(1,1,1) with drift         : -1306.514

 Best model: ARIMA(0,1,0)

> a=sarima(train_log,0,1,0)
> sarima.for(train_log, 21, 0,1,0)
$pred
Time Series:
Start = 1681
End = 1821
Frequency = 0.142857142857143
[1] 9.697326 9.697869 9.698413 9.698956 9.699499 9.700042 9.700585
[8] 9.701129 9.701672 9.702215 9.702758 9.703301 9.703844 9.704388
[15] 9.704931 9.705474 9.706017 9.706560 9.707103 9.707647 9.708190

$se
[1] 0.01546612 0.02187240 0.02678811 0.03093225 0.03458330 0.03788411
[7] 0.04091952 0.04374480 0.04639837 0.04890818 0.05129533 0.05357623
[13] 0.05576390 0.05786894 0.05990004 0.06186450 0.06376846 0.06561721
[19] 0.06741527 0.06916661 0.07087468

> summary(model)
Series: train_log
```

```
ARIMA(0,1,0)

sigma^2 estimated as 0.0002399:  log likelihood=657.14
AIC=-1312.28   AICc=-1312.27   BIC=-1308.81

Training set error measures:
             ME       RMSE        MAE         MPE         MAPE
Training set 0.00058077 0.01545573 0.01134464 0.005902135 0.1183073
             MASE        ACF1
Training set 0.001179495 -0.02012327

> pred = predict(model, n.ahead=21, prediction.invertal = TRUE)
> l=pred$pred-pred$se
> m=pred$pred
> u=pred$pred+pred$se
>
> ts.plot(test_log, pred$pred, col=c(1,6), ylim=c(9.5, 9.9),
+         ylab="Adj Close in log", main="Predicted stock indexes in log",
+         xlab="2019-08-08 to 2019-12-26 on weekly basis")
> lines(pred$pred, type="p", col=2)
> lines(pred$pred+pred$se, lty="dashed", col=4)
> lines(pred$pred-pred$se, lty="dashed", col=4)
>
```

Square of Variables

```
> stock1_sq <- (stock1_xts)^2
> train_sq = stock1_sq[1:240, ]
> test_sq = stock1_sq[241:261, ]
> plot(stock1_sq)
> model3 <- auto.arima(train_sq, trace = TRUE, approximation=FALSE)

 ARIMA(2,1,2) with drift         : 8185.105
 ARIMA(0,1,0) with drift         : 8176.817
 ARIMA(1,1,0) with drift         : 8178.856
 ARIMA(0,1,1) with drift         : 8178.855
 ARIMA(0,1,0)                    : 8175.151
 ARIMA(1,1,1) with drift         : 8180.848

 Best model: ARIMA(0,1,0)
```

Square Root of Variables

```
> stock1_sr <- (stock1_xts)^(1/2)
```

```
> train_sr = stock1_sr[1:240, ]
> test_sr = stock1_sr[241:261, ]
> model4 <- auto.arima(train_sr, trace = TRUE, approximation=FALSE)

 ARIMA(2,1,2) with drift         : 648.4403
 ARIMA(0,1,0) with drift         : 644.6824
 ARIMA(1,1,0) with drift         : 646.6819
 ARIMA(0,1,1) with drift         : 646.6768
 ARIMA(0,1,0)                    : 642.9628
 ARIMA(1,1,1) with drift         : 648.7049

 Best model: ARIMA(0,1,0)
```

SSE Index

```
> train2 <- read("sse_train.csv")
> train2_xts <- xts(train2[,2], order.by = train2$Date, frequency=365)
> train2_log <- log(train2_xts)
> dygraph(train2_xts, ylab="Adj Close",
+ main="SSE Index Adjusted Closing Prices") %>%
+ dySeries(label = "SSE", color = "blue")
>
> plot(train2_log)
> plot(diff(train2_log))
> acf2(train2_log)
> acf2(diff(train2_log))
> model5 <- auto.arima(train2_log, trace = TRUE, approximation = FALSE)

ARIMA(2,1,2) with drift : -160.0301
ARIMA(0,1,0) with drift : -167.566
ARIMA(1,1,0) with drift : -165.2378
ARIMA(0,1,1) with drift : -165.2698
ARIMA(0,1,0) : -169.7921
ARIMA(1,1,1) with drift : -164.1761
Best model: ARIMA(0,1,0)

> sarima(train2_log,0,1,0)
$fit
Coefficients:
constant
0.0003
s.e. 0.0041
sigma^2 estimated as 0.0006351: log likelihood = 85.95, aic = -167.91
$degrees_of_freedom
[1] 37
$ttable
```

```
         Estimate SE t.value p.value
constant 3e-04 0.0041 0.0749 0.9407
$AIC
[1] -4.418653
$AICc
[1] -4.415729
$BIC
[1] -4.332464
>
> sarima.for(train2_log, 21, 1,1,1)
$pred
Time Series:
Start = 274
End = 414
Frequency = 0.142857142857143
[1] 7.266619 7.270874 7.273636 7.275527 7.276909 7.277993 7.278904
[8] 7.279714 7.280464 7.281180 7.281876 7.282560 7.283237 7.283910
[15] 7.284581 7.285250 7.285919 7.286587 7.287255 7.287923 7.288590
$se
[1] 0.02468512 0.03215121 0.03695202 0.04060050 0.04364225 0.04632399
[7] 0.04877130 0.05105343 0.05321109 0.05526965 0.05724584 0.05915130
[13] 0.06099459 0.06278226 0.06451955 0.06621079 0.06785960 0.06946914
[19] 0.07104213 0.07258098 0.07408785
>
> stock2 <- read("sse_test.csv")
> stock2_xts <- xts(stock2[,2], order.by = stock2$Date, frequency=365)
> stock2_log <- log(stock2_xts)
> a = stock2_log[1:256, ] #data from 2015 to 2019
> b = stock2_log[257:273, ] # data from 2019 to 2020
> dygraph(stock2_xts, ylab="Adj Close",
+ main="SSE Index Adjusted Closing Prices") %>%
+ dySeries(label = "SSE", color = "blue")
>
> sarima.for(a, 24, 0,1,0)
$pred
Time Series:
Start = 1793
End = 1954
Frequency = 0.142857142857143
[1] 7.962125 7.961494 7.960862 7.960230 7.959598 7.958967 7.958335
[8] 7.957703 7.957072 7.956440 7.955808 7.955177 7.954545 7.953913
[15] 7.953281 7.952650 7.952018 7.951386 7.950755 7.950123 7.949491
[22] 7.948860 7.948228 7.947596
$se
[1] 0.03464756 0.04899905 0.06001133 0.06929511 0.07747429 0.08486884
[7] 0.09166882 0.09799809 0.10394267 0.10956520 0.11491295 0.12002266
[13] 0.12492354 0.12963929 0.13418941 0.13859023 0.14285554 0.14699714
```

```
[19] 0.15102520 0.15494859 0.15877505 0.16251145 0.16616385 0.16973767
> ts.plot(b, pred$pred, col=c(1,6), ylim=c(7.5, 8.5),
+ ylab="Adj Close in log", main="Predicted stock index in log",
+ xlab="2019-12-02 to 2020-03-10 on weekly basis")
> lines(pred$pred, type="p", col=2)
> lines(pred$pred+pred$se, lty="dashed", col=4)
> lines(pred$pred-pred$se, lty="dashed", col=4)
>
```

Squaring and Square Root

```
> train2_sq <- (train2_xts)^2
> acf2(train2_sq)
> acf2(diff(train2_sq))
> plot(train2_sq)
> plot(diff(train2_sq))
> model7 <- auto.arima(train2_sq, trace = TRUE, approximation = FALSE)

 ARIMA(2,1,2) with drift         : 1001.137
 ARIMA(0,1,0) with drift         : 994.4177
 ARIMA(1,1,0) with drift         : 996.6461
 ARIMA(0,1,1) with drift         : 996.5498
 ARIMA(0,1,0)                    : 992.1906
 ARIMA(1,1,1) with drift         : 997.6607

 Best model: ARIMA(0,1,0)

>
> train2_sr <- (train2_xts)^(1/2)
> model8 <- auto.arima(train2_sr, trace = TRUE, approximation = FALSE)

 ARIMA(2,1,2) with drift         : 64.56275
 ARIMA(0,1,0) with drift         : 56.9287
 ARIMA(1,1,0) with drift         : 59.23862
 ARIMA(0,1,1) with drift         : 59.19311
 ARIMA(0,1,0)                    : 54.70235
 ARIMA(1,1,1) with drift         : 60.29263

 Best model: ARIMA(0,1,0)

>
```