Jostein Barry-Straume
Laura Ludwig
David Tran

Time Series Analysis of Bitcoin

Data Science @ Southern Methodist University

Source: bitcoin.com

Team Members

- Jostein Barry-Straume
- Laura Ludwig
- David Tran

Table of Contents

Introduction

Cryptocurrency is a digital currency and acts as a medium for exchanges/transactions. Cryptocurrencies are decentralized, which means it is not processed by any banking system and goes straight to the consumers. The transactions are posted on an online ledger for transparency. Users' identities are protected through an encryption key, which is a feature that Bitcoin has. Bitcoin is one of the popular choices of cryptocurrency. Since its introduction into the market in 2009, it has drastically increased and decreased in value. The analysis below will offer insights on the characteristics of the cryptocurrency and its projected value and trend.

Problem Statement

Develop a time series model based on an observed set of explanatory variables that can be utilized to predict future price of Bitcoin.

Constraints and Limitations

Bitcoin was created in 2009, and the available data in the dataset begins in April 2013. We are constrained by not seeing all of the history of this currency within the dataset. The data is sourced from Kaggle, which is ultimately sourced from another site that tracks Bitcoin and other cryptocurrencies. There are limitations on the amount of metadata available from this source, particularly around how the market-level breakdown is sourced into one cohesive price in the Historical data. There are some potentially confounding variables inherent in an analysis of Bitcoin. The market valuation is consistently changing on a daily basis with the mining of coins, and the nature of the market is highly dependent on supply and demand. There is also one owner who has 5% of the market share, whose actions may contribute to the behavior of the market prices. There is some data missing, particularly in the Volume variable. The subsequent analysis does not rely on Volume due to lack of collinearity with this variable, but this may have been due to missing data.

Jostein Barry-Straume
Laura Ludwig
David Tran

Data Set Description

The dataset for this analysis was pulled from Kaggle: Cryptocurrency Historical Prices. The data is taken from the historical data available on coinmarketcap.

Variable	Variable Type	Summary
Date	Factor	Date for summary info
Open	Numeric	Opening market price for Bitcoin
High	Numeric	Daily high price for Bitcoin
Low	Numeric	Daily low price for Bitcoin
Close	Numeric	Closing market price for Bitcoin
Volume	Numeric	Total amount of Bitcoin available
Market Cap	Numeric	Market Capitalization (valuation of the overall currency market)
Time	Date	Conversion of original Date variable for analysis use

It is unclear from the sources exactly how the terms Open and Close are defined in the data source. In general, Bitcoin is always open, as a market. The timestamp to mark Opening price and Closing price are based on the timezone of a market, and it turns over at midnight each day. With global markets and multiple time zones, there is no standard time across all markets. Without a clear description available from the source, it is not possible to clearly articulate the exact variable meaning in the real world.

Snapshot of the data set:

```
## 'data.frame':    1620 obs. of  8 variables:
##  $ Date      : Factor w/ 1620 levels "Apr 01, 2014",..: 109 114 119 1069 1074 1079 1084 1089 1094 1099 ...
##  $ Open      : num  135 134 144 139 116 ...
##  $ High      : num  136 147 147 140 126 ...
##  $ Low       : num  132.1 134 134.1 107.7 92.3 ...
##  $ Close     : num  134 145 139 117 105 ...
##  $ Volume    : Factor w/ 1378 levels "-","1,002,120,000",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Market.Cap: Factor w/ 1616 levels "1,000,070,000",..: 130 125 158 142 75 37 16 64 74 62 ...
##  $ Time      : Date, format: "2013-04-28" "2013-04-29" ...

## [1] 1620    8

##              Date   Open   High    Low  Close Volume    Market.Cap
## 1620 Apr 28, 2013 135.30 135.98 132.10 134.21      - 1,500,520,000
## 1619 Apr 29, 2013 134.44 147.49 134.00 144.54      - 1,491,160,000
## 1618 Apr 30, 2013 144.00 146.93 134.05 139.00      - 1,597,780,000
## 1617 May 01, 2013 139.00 139.89 107.72 116.99      - 1,542,820,000
## 1616 May 02, 2013 116.38 125.60  92.28 105.21      - 1,292,190,000
## 1615 May 03, 2013 106.25 108.13  79.10  97.75      - 1,180,070,000
##            Time
## 1620 2013-04-28
## 1619 2013-04-29
## 1618 2013-04-30
## 1617 2013-05-01
## 1616 2013-05-02
## 1615 2013-05-03
```

The above output shows the structure, dimension, and head of the data set. There are 1,620 observations with 8 explanatory variables.

Summary statistics of daily closing price of bitcoin:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   68.43  261.41  448.19  718.80  705.28 4892.01
```

Summary statistics of daily high price of bitcoin:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   74.56  266.46  454.64  736.27  724.55 4975.04
```

Additional summary statistics and subsequent analysis indicated that most other variables are similar in trend to the closing price, which was selected as the response variable for the analysis.

Exploratory Data Analysis

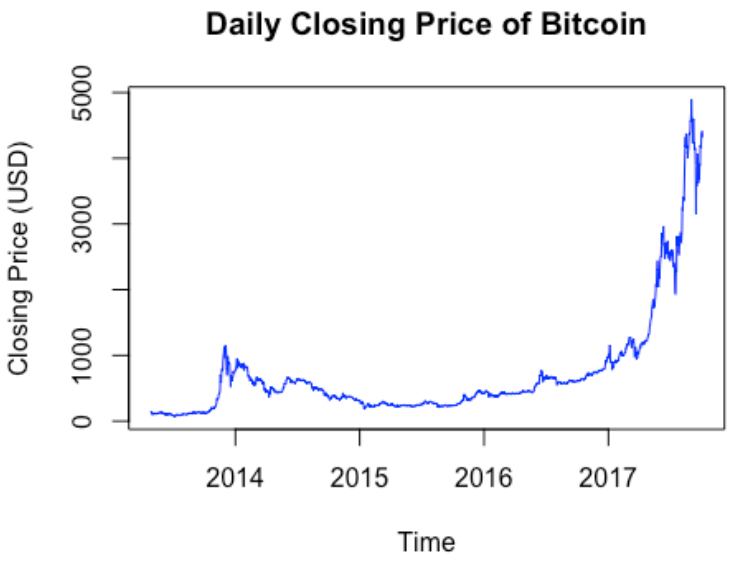

Figure 1: Line Plot of Daily Closing Price of Bitcoin

Figure 1 reflects the daily closing prices of bitcoin from to April 28th, 2013 to October 3rd, 2017. Although there appears to be no pattern in the change of the closing price, a general increase in price over time is apparent. Increasing variance over time necessitates transformation of the original data.

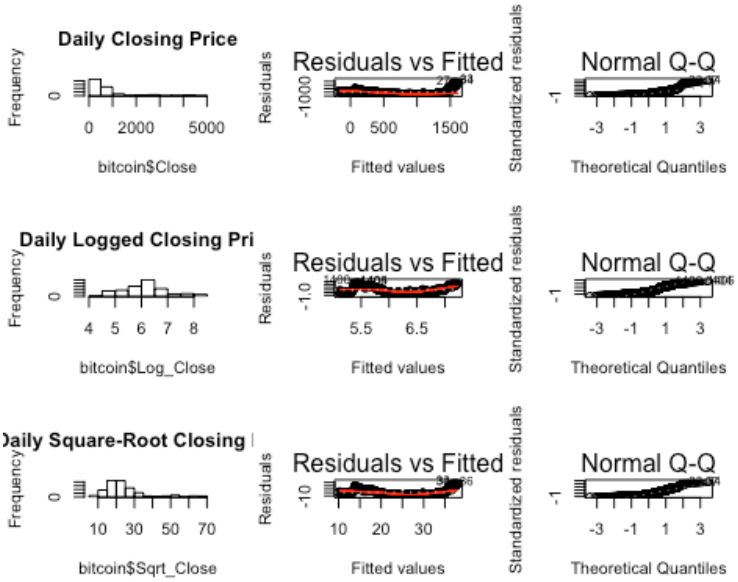

Figure 2: Diagnostic Plots of Original and Transformed Data

The above diagnostic plots (Figure 2) confirm the need for transformation, as well as give insight into which transformation is most appropriate. The histograms of both the original data and square-root data are heavily right skewed, with the former to a larger degree. Additionally, the Q-Q plots for the original data and logged data venture far of the path of diagonal line. In contrast, the logged data displays a normal distribution for its histogram, as well as a fairly good Q-Q plot. The tail ends of the logged Q-Q plot indicate some skewness at both ends, which the corresponding histogram supports. However, the size of our data set should ease

any concern we might have. The residual diagnostic plot of the logged data reflects non-constant variance. This will be addressed by taking the first-degree difference of the logged daily closing price.

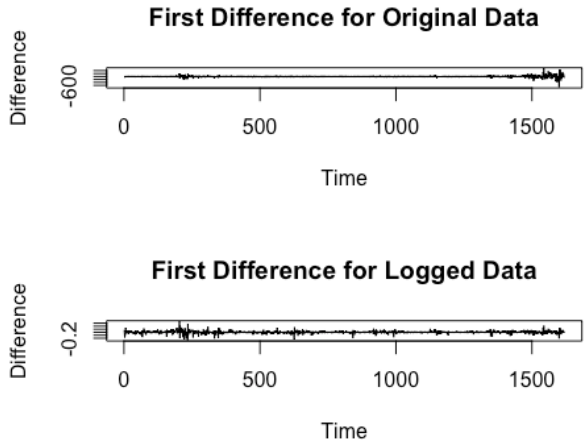

Figure 3: First Differences of Original and Logged Data

The variance of the first difference between the original and logged data are vastly different. In the original data, the increasing variance as time goes on is visually clear, whereas the variance of the logged data is reasonably constant with no apparent patterns.

Variable Screening

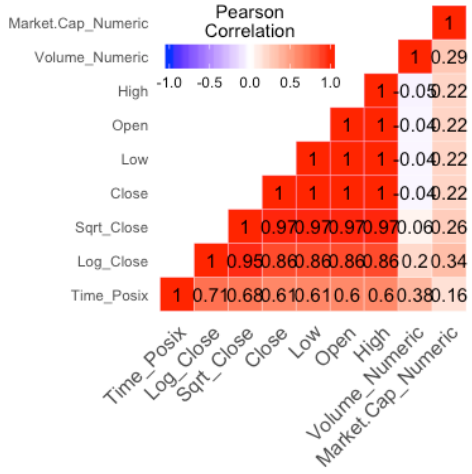

Figure 4: Heatmap Correlation Matrix

The above heat map correlation matrix (Figure 4) offers limited new comprehension of the bitcoin data set, but is still helpful nonetheless. Volume of daily bitcoin trades has a weak positive correlation (R = 0.20) with logged closing price, and a moderate positive correlation (R = 0.38) with time. This suggests that as time goes on, the volume of trades increases and might have an impact on the closing price of bitcoin. Of note, the total market cap of mined bitcoins has a moderate positive correlation with logged closing prices. In other words, the total value of mined bitcoins possibly influences the closing price.

The original and logged closing prices have strong positive correlations with time (R = 0.61, and R = 0.71 respectively). This suggests the need to address auto correlation issues. Moreover, the following variables are 100% collinear with each other: High, Low, Open, and Close. This makes sense as all of the said variables pertain to the price of bitcoin. To reduce redundancy, only the closing price of bitcoin will be utilized for a time series model.

Model Selection

Analysis of the daily closing price of bitcoin can now be carried out. Instead of manually testing various models, the computer will select the best from a plethora of models based on their respective Akaike Information Criterion (AIC) value. Invoking the trace option in the auto.arima() function allows the user to see which exact models the computer is testing, as seen below:

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)(1,0,1)[365] with drift         : Inf
##  ARIMA(0,1,0)             with drift         : -5585.415
##  ARIMA(1,1,0)(1,0,0)[365] with drift         : Inf
##  ARIMA(0,1,1)(0,0,1)[365] with drift         : Inf
##  ARIMA(0,1,0)                                : -5583.391
##  ARIMA(0,1,0)(1,0,0)[365] with drift         : Inf
##  ARIMA(0,1,0)(0,0,1)[365] with drift         : Inf
##  ARIMA(0,1,0)(1,0,1)[365] with drift         : Inf
##  ARIMA(1,1,0)             with drift         : -5585.338
##  ARIMA(0,1,1)             with drift         : -5583.537
##  ARIMA(1,1,1)             with drift         : -5583.41
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,1,0)             with drift         : -5591.706
##
##  Best model: ARIMA(0,1,0)             with drift

## Series: myts
## ARIMA(0,1,0)             with drift
##
## Coefficients:
##        drift
##       0.0021
## s.e.  0.0011
##
## sigma^2 estimated as 0.001848:  log likelihood=2797.85
## AIC=-5591.71   AICc=-5591.7   BIC=-5580.93
##
## Training set error measures:
##                      ME       RMSE        MAE           MPE      MAPE
## Training set 3.023e-06 0.04296455 0.02617695 -0.004801344 0.4298044
##                    MASE         ACF1
## Training set 0.02767789 -0.008218823
```

It appears that an ARIMA model with an order of (0, 1, 0) with a constant has been selected, which corresponds to a random walk model with drift. In other words, the best forecast for tomorrow's closing price of bitcoin is based on today's closing price plus a drift term. The general historical trend of bitcoin's closing price determines the drift term.

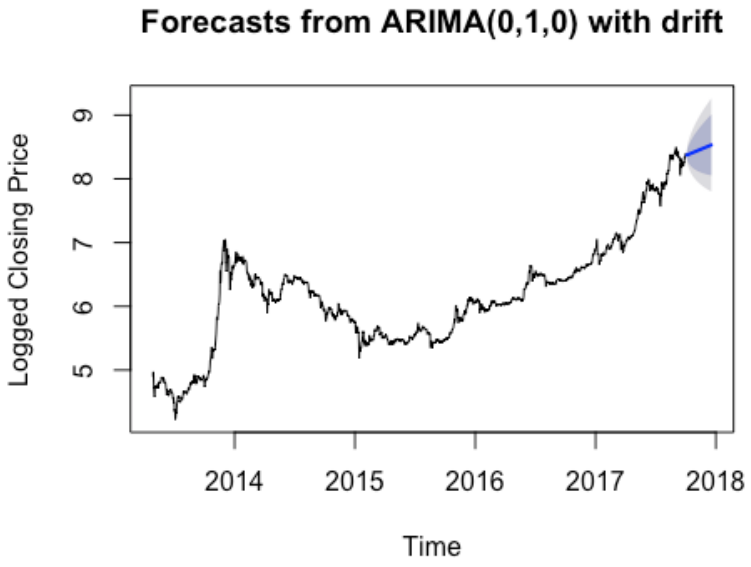

Figure 5: Forecast Plot from ARIMA model

Figure 5 reflects the forecast of bitcoin's logged closing price for the next 75 days ahead of October 3rd, 2017. The predicted forecast values are represented as the blue line, and the drift terms appear as the grey fan shape around the forecasted values. The accuracy of the forecast model is best seen by testing it against already known observations.

```
##                        ME       RMSE        MAE          MPE      MAPE
## Training set  0.000003023 0.04296455 0.02617695 -0.004801344 0.4298044
## Test set     -0.223949838 0.27269486 0.22482297 -2.764754812 2.7750326
##                    MASE         ACF1
## Training set 0.9965053 -0.008218823
## Test set     8.5585688           NA
```

Fortunately, the accuracy() function in the forecast library of R provides a convenient vehicle in which to discern the precision of the forecast model. The above table output shows, among many things, the Root Mean Square Error (RMSE) for both the training and test set. In this scenario, the last 75 days of the data set were used as the test set. The training set had a RMSE value of 0.04296455, whereas the test set had a RMSE value of 0.27269486. So, the difference in the standard deviation of the residuals between the training and test set was 0.2297303. The fact that the test RMSE value is over 6 times greater than the training set may indicate the presence of overfitting. Even so, it may very well be the case that the best model is simply over-fit.

Serial Correlation

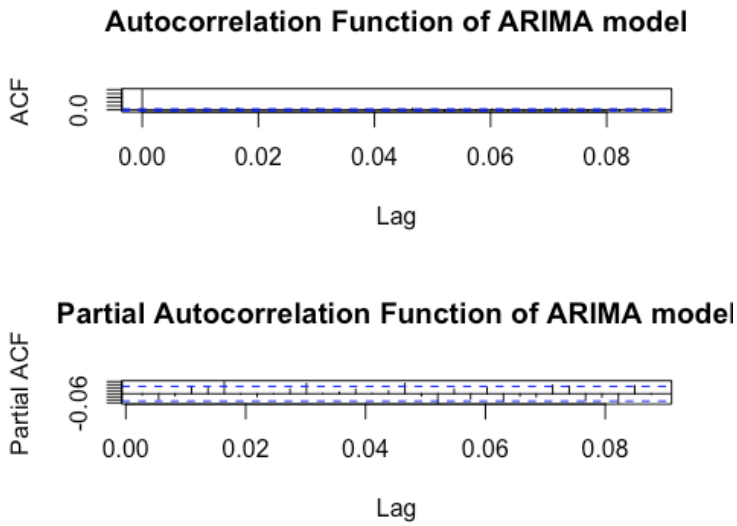

Figure 6: ACF and PACF Plots of Residuals

Upon examination, both the ACF and PACF plots (Figure 6) exhibit "white noise" behavior. In the ACF plot, residuals decay exponentially to zero after lag of zero. Likewise, the residuals in the PACF plot are either well within or near the two-standard deviation boundary. These characteristics are indicative that the residuals are behaving like uncorrelated data, which in turn means that the data is behaving as if stationary under first degree differences.

```
##  Box-Ljung test
##
## data:  residuals(forecast_arima_fit)
## X-squared = 0.10963, df = 1, p-value = 0.7406
```

The Ljung-Box test allows for the examination of independence. With a resulting p-value of 0.7406, we fail to reject the null hypothesis that any group of autocorrelations of this time series data is zero. The data is plausible under the null hypothesis, so there is reason to believe that there is no autocorrelation.

Conclusion

The time series analysis model above was developed to determine the relationship of the response variable of the Closing price of Bitcoin with the explanatory variables of Date, Open, High, Low, Volume, Market Cap, and Time. Based on our diagnostic plots, a log transformation on the data was performed to obtain a normal distribution of the data. In our explanatory data analysis, the closing price of Bitcoin did not appear to have a pattern, but a general price increase was observed over time. It is expected that prices will generally increase as a positive linear relationship with time. In our variable screening, the correlation between volume of daily bitcoin trades and the logged closing price has a low positive correlation value (R = 0.20) and a moderate positive correlation (R = 0.38) with time.

According to CoinDesk.com, the price of Bitcoin is $7,351.53 as of November 4th, 2017. Since the last closing price of our dataset, the price of Bitcoin has increased roughly 70%. With the analysis above, it is recommended that the forecasting should be used in a short-term range and kept up-to-date.

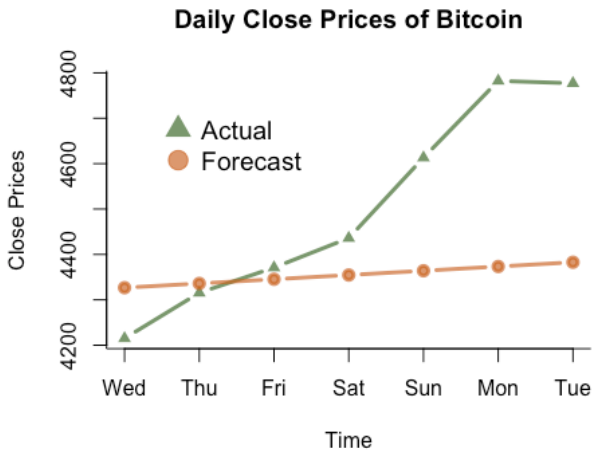

Figure 7: Forecast Vs. Actual Close Prices

To put the forecast model to test in a real-world environment, actual closing prices of bitcoin were gathered to compare against the forecast model (Figure 7). Specifically, October 4th to October 10th were examined as it is a week beyond where the data set leaves off. It appears the forecast model does fairly well within the time span of 3 days, but beyond that point the realized closing prices venture far from those of the forecast model. When looking at the larger trend up until today's current date, there have been significant exponential increases in the closing price of bitcoin. This recent trend may indicate that a new model may be necessary to capture the new behaviors of bitcoin itself.

In particular, additional research into forecasting bitcoin prices would do well in focusing on exploring the nature between news about bitcoin and its respective price. The second chapter of this case study could combine Google search trends with the logged daily closing price to potentially construct a more accurate forecasting model.

Ways to improve the analysis include drilling down deeper into the sporadic changes in bitcoin's value. A bitcoin price per minute data set is available on Kaggle for analysis, but due to hardware and time constraints the daily data set was chosen. Moreover, employing additional data gathering techniques to capture the full scope of bitcoin's financial history could improve the forecasting of bitcoin's future financial value. However, it is unclear how helpful the initial years would be, as the current price of one bitcoin is many magnitudes greater than its initial value in the beginning years.

Appendix

```
# Load lubridate package to convert dates from factor to date class
library(lubridate)
library(reshape2)
library(ggplot2)
library(forecast)
library(DescTools)

# File path of the data set
path <- "/Users/Jostein/Grad School/SMU/6372/project2/bitcoin/data/bitcoin_price.csv"

# Read in the CSV file of the data set
bitcoin <- read.csv(path, header = TRUE)

# Create new variable time via Lubridate, then order data set by ascending time
bitcoin$Time <- mdy(bitcoin$Date)
bitcoin <- bitcoin[order(bitcoin$Time),]

# Snapshot of the data set
str(bitcoin)
dim(bitcoin)
head(bitcoin)

summary(bitcoin$Close)
summary(bitcoin$High)

# Line plot of daily closing price of bitcoin
# Source: https://stackoverflow.com/questions/9053437/r-plot-with-an-x-time-axis-how-to-force-the-ticks-labels-to-be-the-days
tsPlot <- ts(data = bitcoin$Close, start = c(2013, 118), frequency = 365)
plot(tsPlot, type = "l", ylab = "Closing Price (USD)", main = "Daily Closing Price of Bitcoin", xlab = "Time", col = "blue")

# Transformation appears to be needed
bitcoin$Log_Close <- log(bitcoin$Close)
bitcoin$Sqrt_Close <- sqrt(bitcoin$Close)

# How do we get to stationary?
# First differences for original and transformed data sets
diff1 <- diff(bitcoin$Close, lag = 1)
logDiff1 <- diff(bitcoin$Log_Close, lag = 1)
sqrtDiff1 <- diff(bitcoin$Sqrt_Close, lag = 1)

fitClose <- lm(Close ~ Time, data = bitcoin)
fitLogClose <- lm(Log_Close ~ Time, data = bitcoin)
fitSqrtClose <- lm(Sqrt_Close ~ Time, data = bitcoin)
par(mfrow = c(3, 3))
hist(bitcoin$Close, main = "Daily Closing Price")
plot(fitClose, which = 1:2)
hist(bitcoin$Log_Close, main = "Daily Logged Closing Price")
plot(fitLogClose, which = 1:2)
hist(bitcoin$Sqrt_Close, main = "Daily Square-Root Closing Price")
plot(fitSqrtClose, which = 1:2)

par(mfrow = c(2, 1))
plot(diff1, type = "l", xlab = "Time", ylab = "Difference", main = "First Difference for Original Data")
plot(logDiff1, type = "l", xlab = "Time", ylab = "Difference", main = "First Difference for Logged Data")

# Correlation matrix heatmap
# Source: http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization
```

```
# Source: https://stackoverflow.com/questions/3571909/calculate-correlation-cor-for-only-a-subset-of-columns
# Source: https://stackoverflow.com/questions/20077944/changing-dates-into-numeric-form-to-do-a-correlation
# Change variables from factors and date into numeric class
bitcoin$Time_Posix <- as.POSIXct(bitcoin$Time, format="%Y-%m-%d %H:%M:%S")
bitcoin$Time_Posix <- as.numeric(bitcoin$Time_Posix)
bitcoin$Volume_Numeric <- as.numeric(bitcoin$Volume)
bitcoin$Market.Cap_Numeric <- as.numeric(bitcoin$Market.Cap)
cormat <- round(cor(bitcoin[sapply(bitcoin, is.numeric)]), 2)

# Get Lower triangle of the correlation matrix
get_lower_tri <- function(cormat) {
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat) {
  cormat[lower.tri(cormat)] <- NA
  return(cormat)
}

# Organize the correlation matrix
reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}

cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)

# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))+
  coord_fixed()

ggheatmap +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    legend.justification = c(1, 0),
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                               title.position = "top", title.hjust = 0.5))

# Arima Modeling
# Source: https://stats.stackexchange.com/questions/207473/how-to-set-the-prediction-range-of-arima-model-in-
```

```r
# Source: https://www.otexts.org/fpp/8/7
myts <-ts(data = bitcoin$Log_Close, start = c(2013, 118), frequency = 365)
arima_fit <- auto.arima(myts, d = 1, ic = "aic", trace = TRUE)
summary(arima_fit)

forecast_arima_fit <- forecast(arima_fit, h = 75)
plot(forecast_arima_fit, xlab = "Time", ylab = "Logged Closing Price")

accuracy(f = forecast_arima_fit, x = myts[1546:1620])

par(mfrow = c(2, 1))
acf(residuals(forecast_arima_fit), main = "Autocorrelation Function of ARIMA model")
pacf(residuals(forecast_arima_fit), main = "Partial Autocorrelation Function of ARIMA model")

# Source: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/box.test.html
# Source: https://stats.stackexchange.com/questions/64711/ljung-box-statistics-for-arima-residuals-in-r-confu
sing-test-results
Box.test(residuals(forecast_arima_fit), type = "Ljung")

df <- read.csv("/Users/Jostein/Grad School/SMU/6372/project2/bitcoin/data/conclusion_comparison.csv", header
= TRUE)
df$Date <- ymd(df$Date)
# Source: http://www.r-graph-gallery.com/119-add-a-legend-to-a-plot/
plot(y = df$Actual, x = df$Date, col=rgb(0.2,0.4,0.1,0.7), type = "b", bty="l", lwd=3, pch=17, xlab = "Time",
ylab = "Close Prices", main = "Daily Close Prices of Bitcoin")
lines(y = df$Forecast, x = df$Date, col=rgb(0.8,0.4,0.1,0.7), lwd=3 , pch=19 , type="b")
legend("topleft",
       legend = c("Actual", "Forecast"),
       col = c(rgb(0.2,0.4,0.1,0.7),
               rgb(0.8,0.4,0.1,0.7)),
       pch = c(17,19),
       bty = "n",
       pt.cex = 2,
       cex = 1.2,
       text.col = "black",
       horiz = F ,
       inset = c(0.1, 0.1))
```