

Time Series Analysis of Dominos Stock Price Data

Arjun Sahasrabuddhe (arjun981@ucsb.edu)

2024-05-21

```
knitr::opts_chunk$set(echo = TRUE)
library(astsa)
library(forecast) # for forecasting data
library(fGarch) # for garch modeling
```

Abstract

Stock market data is used today all across the world, but especially in the US. According to the Statista Research Department, stock markets in the US account for roughly 60 percent of world stocks in 2024. The US has been involved in the stock market since the late 1700s, with the well known New York Stock Exchange (NYSE) on Wall Street being founded in 1792. Stock data can be used on both a macro or micro scale. We can study them to better understand how well a country is performing economically, but also when an individual stock broker should buy or sell a stock. Generally, stock prices tend to increase over time due to inflation, but could also be through increased profits in the company. Time series analysis has always been a great way to model financial data because it has tools that can help us with projections based on patterns from past data. In my project, I fit a SARIMA model, but also a GARCH model to see what fits the data best. I also use a SARIMA model to forecast future data. These models will be explained in detail later.

Introduction

Domino's is a well known pizza restaurant chain that is in more than 90 international markets today. It is the largest pizza chain in the world as well as the US, and is a commonly traded stock on the market. I also value their products and have been a fairly regular customer. In this project, I will be examining the Dominos Stock Price using a dataset retrieved from Kaggle, titled "Dominos Pizza Stock Data". I chose this dataset because it contains daily data that is tidied and relatively recent (2019-2021, also includes the onset of COVID), and for which we can conduct time series forecasting/modeling. Modeling the stock price can be a good indicator of how the company is performing financially. If profits are lost or gained during a specific period, we should expect to see a change in the stock price. I will specifically be measuring the Adjusted Stock Price across time because this is the price after any corporate actions are made, which gives us a more accurate representation of the "true" stock price. We found that the best model was a ARMA(1,8)-GARCH (1,1) model with an AIC of -4.922.

Data

The Dominos Pizza Stock Data starts in 16th October 2019 and ends in 15th October 2021. The frequency of the dataset is daily, where each observation is a new day. The units are US dollars, and the values are non-negative. The sample size of the time series dataset is 505. This dataset was found on Kaggle, and the webpage link is: <https://kaggle.com/datasets/pratmo/dominos-pizza-stock-data>. The data was collected by Prathik Mohan, and was extracted from finance.yahoo.com. It is a useful dataset because we can perform various tasks on the data such as modeling how stock price changes over time, and whether it might be a valuable stock to invest in. Also, we can notice any sudden abnormalities or changes in the data, and identify times where it may be smart to buy or sell this stock. Next, we will describe our models and fit them on our data to help forecast and predict the stock price for future days, which can be quite useful.

Methodology

SARIMA Model

The first model we apply is a SARIMA (p, d, q) X (P, D, Q) model. This method is similar to an ARIMA model but also takes into account any seasonal patterns in the data. The p is the autoregressive order, d is the difference order, and q is the moving average order. The P, A, Q are the autoregressive, difference, and moving average orders respectively but seasonal components. The S refers to the length of the seasonal period. If there is any seasonality in our data, this model will pick up on it as stock market data is known to be seasonal. If the data was not likely to be seasonal, using an ARIMA model could be more efficient as it is simpler and easier to interpret.

ARCH/GARCH Model

The second model we apply is a ARCH/GARCH model. It's commonly used to model heteroskedasticity for financial data. Heteroskedasticity means that the variance of the errors is not constant across observations. It has an ARMA as well as a GARCH component, so it can be seen as an extension of the ARMA model. An ARMA model assumes that there is constant volatility, or variance, unlike the GARCH model. It uses the concept of volatility clustering, which says that large movements are followed by large movements, and small movements are followed by small movements. The p in the GARCH (p, q) model refers to the spikiness in the volatility, and the q refers to how long it takes for the volatility to change, or the persistency.

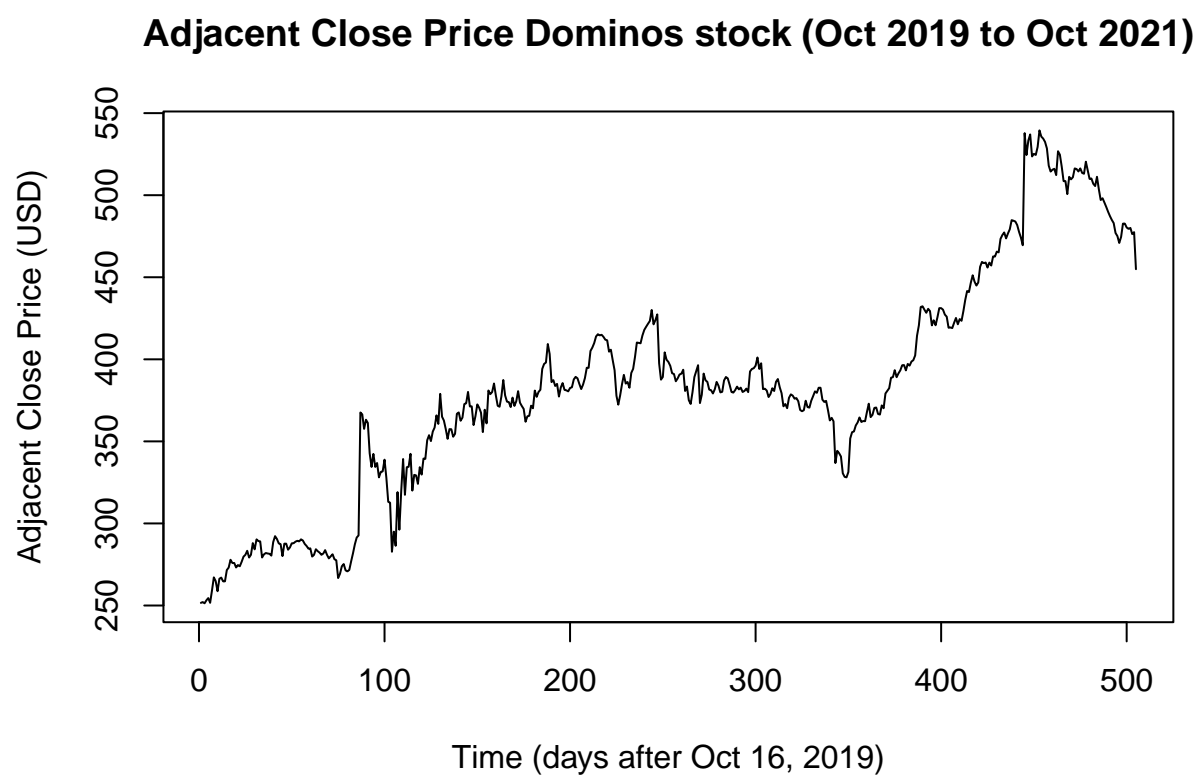
Results

Let us first see a time series visualization of adjusted stock price with time, where time is measured as the number of days after the first observation on October 16th, 2019. We will also visualize ACF/PACF plots.

```
setwd("~/R Studio Courses/PSTAT174 - Time Series")
Dominos_Stock_Data <- read.csv("Dominos_Stock_Data.csv")

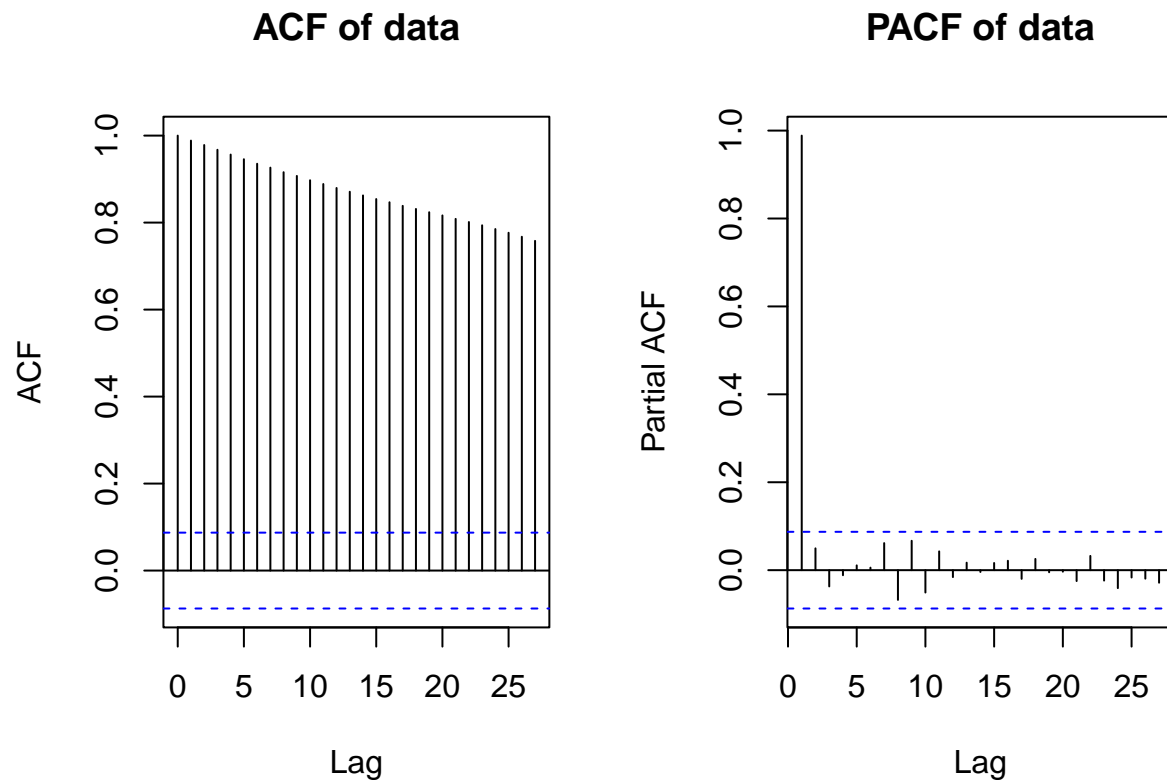
# I've decided to measure the adjacent close price with time.
data <- Dominos_Stock_Data$Adj.Close

plot.ts(data, xlab = 'Time (days after Oct 16, 2019)', ylab = 'Adjacent Close Price (USD)', main = 'Adj
```



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

acf(data, main = 'ACF of data')
pacf(data, main = 'PACF of data')
```

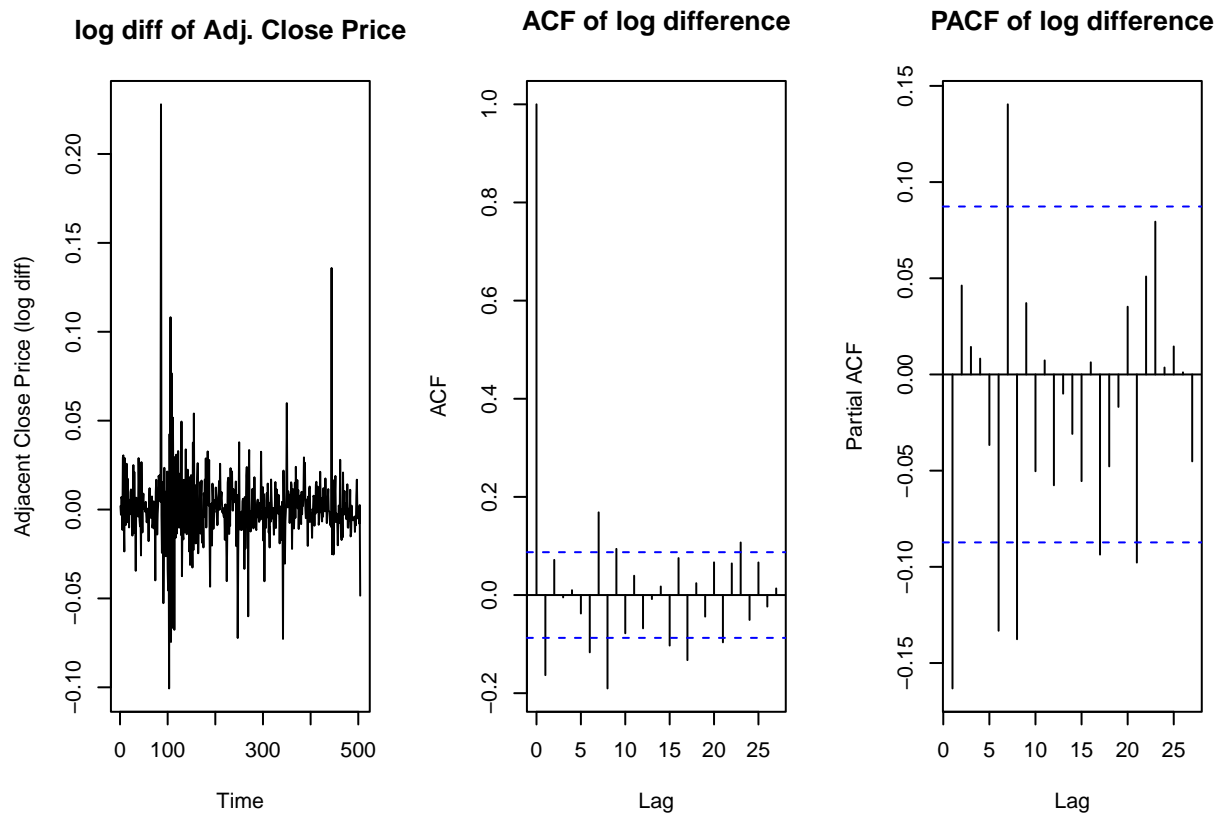


As we can see above, we are dealing with non-stationary data and this is true for most stock market data. Like we expect, as time increases our stock price increases as well due to economic growth. Our ACF/PACF plots also don't tell us much, because our ACF plots has lags that are all highly correlated with one another, and our PACF plot has no significant lags. Let's take the log difference of this data to make the data stationary, and to make it easier to visualize trends. When there are changes in our data, they will stand out more clearly.

```
diff_fn = diff(log(data), 1) # take log difference once.

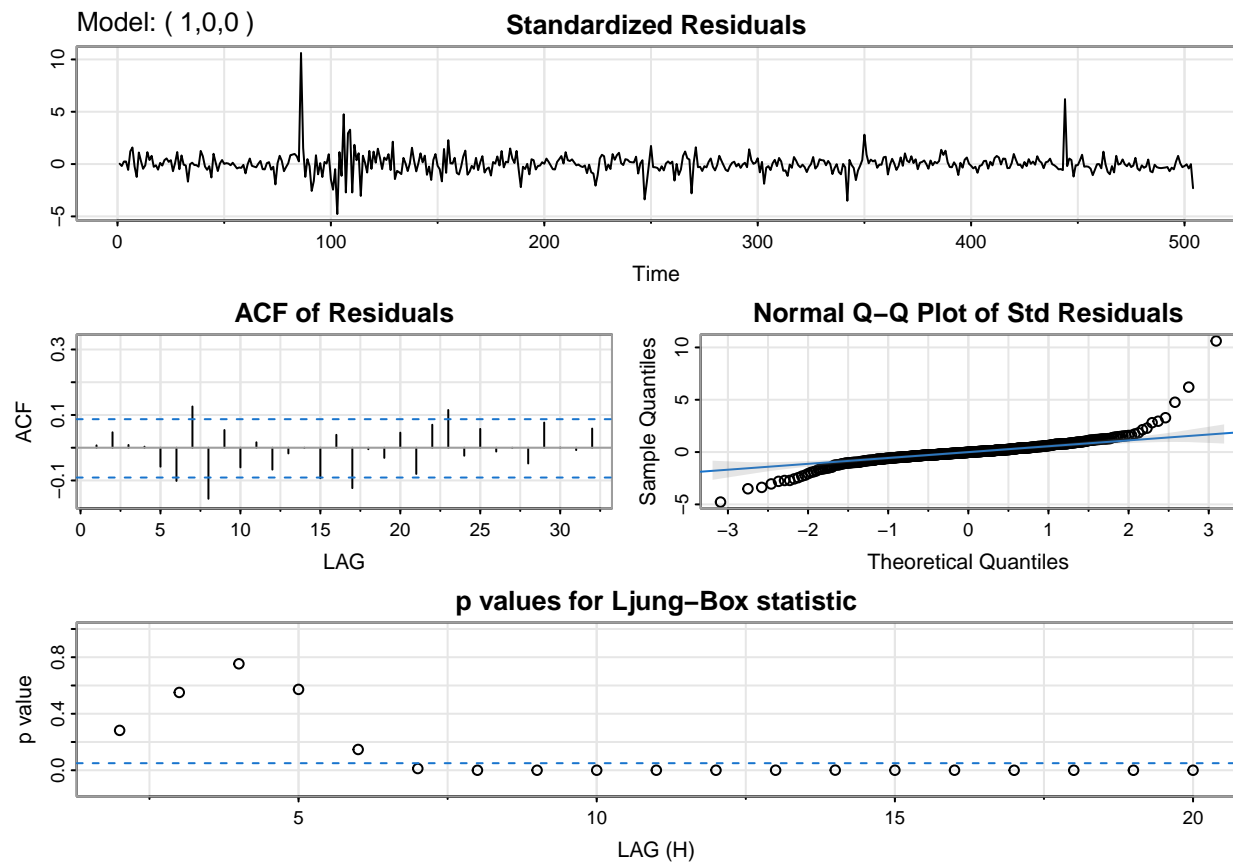
par(mfrow = c(1, 3))
plot.ts(diff_fn, ylab = 'Adjacent Close Price (log diff)', main = 'log diff of Adj. Close Price')

acf(diff_fn, main = 'ACF of log difference')
pacf(diff_fn, main = 'PACF of log difference')
```



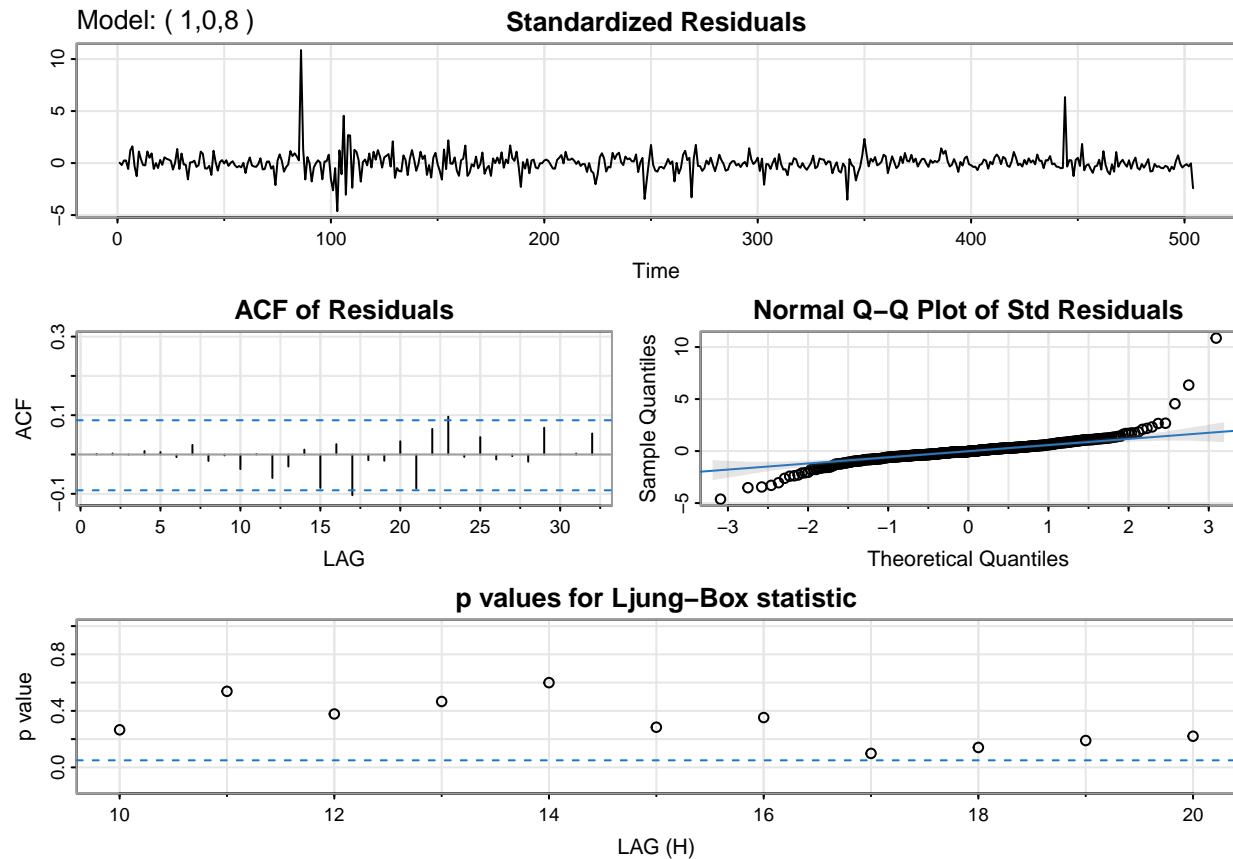
Now our graph looks stationary. We can notice sudden changes in variances that weren't visible before. Our ACF plot has multiple significant lags, the most significant being at lag 8. Our PACF plot also has multiple significant lags, where the most significant is at lag 1. We will use this knowledge when building our SARIMA model.

```
auto.arima(diff_fn)
sarima(diff_fn, 1, 0, 0, P = 0, D = 0, Q = 0, S = -1)
```



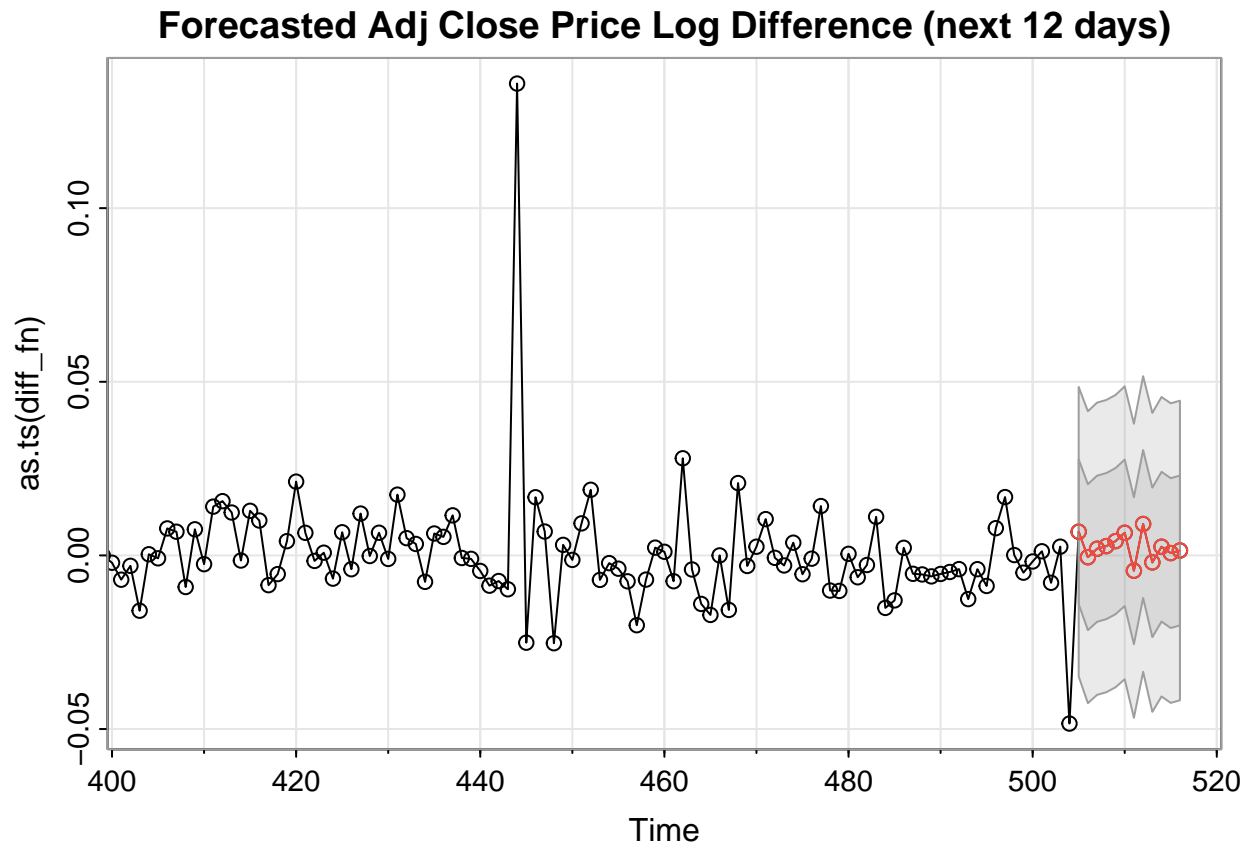
First, we apply the auto arima function to identify the best ARIMA model according to the AIC value and we get ARIMA (1, 0, 0). However our p values are below the threshold line, which means that there is autocorrelation between residuals. We also try using ARIMA (1, 0, 1), ARIMA (0, 0, 1), ARIMA (1, 0, 1), and have a similar result. This may not be the best model to use, as the residuals should be normally distributed and have a zero mean. Let's proceed by using our p=1, q=8 model.

```
sarima(diff_fn, 1, 0, 8, P = 0, D = 0, Q = 0, S = -1) # sarima chosen model
```



We can see that our standard residuals plot appears to follow a stationary pattern, and that the ACF of residuals plot has no significant peaks. We can also see that our p-values are all above the blue significance line, which suggests that there is no significant autocorrelation between residuals. The values that we chose for our SARIMA model, $p=1$ and $q=8$ seem like a good choice based on our plots. We tried adjusting the seasonal component “S” but this didn’t affect our plots which tells us our data doesn’t have a seasonal element. From our ACF residuals plot, we can see that the residuals follow a white noise process. Our Q-Q plot has some outliers that don’t follow the trend line. This is likely due to the peaks we see in our standardized residuals plot just before time 100 and 450. We received a sigma squared value of almost zero, and our AIC is -4.83.

```
sarima.for(as.ts(diff_fn), 12, 1, 0, 8, main = 'Forecasted Adj Close Price Log Difference (next 12 days)
```



We also forecasted the next 12 days for our data after the last observation (October 15, 2021) which we can see above. This is obviously going to be very useful for a stock trader, or even as an economist trying to predict future trends.

To account for volatility, or sudden changes in our data like we can see at time 444, let's see how a GARCH model does and whether it outperforms our SARIMA model.


```

# GARCH model
fit <- garchFit(~arma(1, 8) + garch(1, 1), data = diff_fn) # best garch model based on what we tried.

##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(1, 8) + garch(1, 1), data = diff_fn)
##
## Mean and Variance Equation:
## data ~ arma(1, 8) + garch(1, 1)
## <environment: 0x7fb240073008>
## [data = diff_fn]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ar1      ma1      ma2      ma3      ma4
## 6.6875e-04 2.4545e-02 -3.2745e-01 -6.7503e-02 -1.2519e-01 -4.7886e-02
##      ma5      ma6      ma7      ma8      omega      alpha1
## 5.6659e-02 1.7350e-02 1.2785e-01 -8.7632e-02 7.3895e-05 7.1908e-01
##      beta1
## 4.2686e-01
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      6.687e-04 4.423e-04 1.512 0.1306
## ar1      2.454e-02 2.712e-01 0.091 0.9279
## ma1     -3.275e-01 2.649e-01 -1.236 0.2165
## ma2     -6.750e-02 1.029e-01 -0.656 0.5118
## ma3     -1.252e-01 5.793e-02 -2.161 0.0307 *
## ma4     -4.789e-02 6.084e-02 -0.787 0.4312
## ma5      5.666e-02 4.873e-02 1.163 0.2449
## ma6      1.735e-02 5.359e-02 0.324 0.7461
## ma7      1.279e-01 5.022e-02 2.546 0.0109 *
## ma8     -8.763e-02 4.454e-02 -1.967 0.0492 *
## omega    7.390e-05 1.829e-05 4.040 5.34e-05 ***
## alpha1   7.191e-01 1.443e-01 4.982 6.29e-07 ***
## beta1    4.269e-01 7.311e-02 5.838 5.27e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 1265.559 normalized: 2.511031
##
## Description:
## Sat Jun 22 17:59:40 2024 by user:
##

```

```
##
## Standardised Residuals Tests:
##
##           Statistic      p-Value
## Jarque-Bera Test    R    Chi^2  5827.5687399 0.000000e+00
## Shapiro-Wilk Test   R     W      0.8593671 0.000000e+00
## Ljung-Box Test      R    Q(10)   40.2519787 1.529795e-05
## Ljung-Box Test      R    Q(15)   47.7156001 2.826030e-05
## Ljung-Box Test      R    Q(20)   50.3581963 1.967741e-04
## Ljung-Box Test      R^2  Q(10)    1.3753792 9.992739e-01
## Ljung-Box Test      R^2  Q(15)    2.7602046 9.997612e-01
## Ljung-Box Test      R^2  Q(20)    5.5679218 9.993678e-01
## LM Arch Test        R    TR^2     2.6071118 9.977393e-01
##
## Information Criterion Statistics:
##           AIC      BIC      SIC      HQIC
## -4.970474 -4.861558 -4.971760 -4.927750
```

In our GARCH model above, we utilized our arma (1,8) that we selected from above, and tried 2 different GARCH models: garch (1, 1), and garch (1, 0). Garch (1, 1) had the lowest AIC of -4.97 so we chose a ARMA (1, 8) - GARCH (1, 1) model.

Conclusion and Future Study

In this project, we visualized the Dominos stock price data, modeling the Adjusted Stock Price. We made our data stationary, and used our ACF/PACF plots to help us choose the best SARIMA model, and we also applied a GARCH model. At the end, we can say that we were able to model Dominos stock data effectively through the use of our models. Our ARMA(1,8) - GARCH(1,1) model fit slightly better than our SARIMA model, giving us a lower AIC value of -4.97 versus -4.83. The GARCH (1,1) model also had a high log-likelihood value of 1265.559, which supports its effectiveness. The likely reason for this is that our data had a considerable amount of volatility that our sarima model was unable to pick up on but our GARCH (1,1) model did.

I do think we had enough observations to work with in this dataset, but our time range could have been longer. We are only looking at data from 2019 to 2021 and for a lot of stock forecasting, data from many years ago is often used.

In the future, we plan on expanding on these models, and there are many variations of the SARIMA and GARCH models that we haven't tried. There are always so many other models out there that I'm sure may work better than what we've tried, but we're happy with our results. I felt I learned a lot exploring this dataset, and I'm excited to dive deeper into time series analysis in the future.

References

1. <https://www.kaggle.com/datasets/pratmo/dominos-pizza-stock-data>
2. Time Series Analysis and its Applications with R Examples by R. Links to an external site. H. Shumway and D. S. Stoffer
3. <https://www.statista.com/statistics/710680/global-stock-markets-by-country/#:~:text=Countries%20with%20largest>
4. <https://www.ucl.ac.uk/~uctp41a/b203/lecture9.pdf>
5. <https://fastercapital.com/topics/interpreting-garch-results.html#:~:text=The%20residuals%20from%20the%20GARCH>

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(astsa)
library(forecast) # for forecasting data
library(fGarch) # for garch modeling

setwd("~/R Studio Courses/PSTAT174 - Time Series")
Dominos_Stock_Data <- read.csv("Dominos_Stock_Data.csv")

# I've decided to measure the adjacent close price with time.
data <- Dominos_Stock_Data$Adj.Close

plot.ts(data, xlab = 'Time (days after Oct 16, 2019)', ylab = 'Adjacent Close Price (USD)', main = 'Adj. Close Price')

par(mfrow = c(1, 2))
acf(data, main = 'ACF of data')
pacf(data, main = 'PACF of data')

diff_fn = diff(log(data), 1) # take log difference once.

plot.ts(diff_fn, ylab = 'Adjacent Close Price (log diff)', main = 'log difference of Adj. Close Price')

par(mfrow = c(1, 2))

acf(diff_fn, main = 'ACF of log difference')
pacf(diff_fn, main = 'PACF of log difference')

auto.arima(diff_fn)
sarima(diff_fn, 1, 0, 0, P = 0, D = 0, Q = 0, S = -1)

sarima(diff_fn, 1, 0, 1, P = 0, D = 0, Q = 0, S = -1)

sarima(diff_fn, 0, 0, 1, P = 0, D = 0, Q = 0, S = -1)

sarima(diff_fn, 0, 0, 0, P = 0, D = 0, Q = 0, S = -1)
```

```
sarima(diff_fn, 1, 0, 8, P = 0, D = 0, Q = 0, S = -1) # sarima chosen model
```

```
sarima.for(as.ts(diff_fn), 12, 1, 0, 8, main = 'Forecasted Adj Close Price Log Difference (next 12 days)
```

```
fit1 <- garchFit(~arma(1, 8) + garch(1, 0), data = diff_fn) # garch model  
summary(fit1)
```

```
fit <- garchFit(~arma(1, 8) + garch(1, 1), data = diff_fn) # best garch model based on what we tried.  
summary(fit) # fit a garch model and compare your results. Compare the AIC value, and see which is small
```