Modelling Time Series Bitcoin (BTC) Adjusted Closing Price Data

Aarya Kulkarni - aarya_kulkarni@ucsb.edu

2023-05-17

Abstract

In this project, I'll be attempting to model time series data of BTC adjusted closing prices in an effort to gain insight on the nature of this type of data. I'll be using a Box-Jenkins approach as well as GARCH modelling to do so.

Introduction

For this project, I chose to model Bitcoin price because it has gained significant popularity as a decentralized digital currency and has experienced substantial price fluctuations throughout its history. By modeling and analyzing the adjusted price data, one can gain insights into the underlying trends, patterns, and potential factors influencing Bitcoin's price movements.

Understanding the dynamics of Bitcoin prices can be valuable for investors, traders, and financial institutions. Modeling the adjusted price data allows for the identification of important features such as trends, seasonality, and volatility. This information can aid in making informed decisions related to investment strategies, risk management, and portfolio diversification.

Moreover, the modeling of Bitcoin adjusted price data can provide insights into the overall cryptocurrency market. Bitcoin's price is often considered a good metric for other cryptocurrencies, and modeling its data can help in understanding the broader trends and dynamics of the cryptocurrency ecosystem.

Overall, modeling Bitcoin adjusted price data can play a crucial role in uncovering the underlying dynamics of the cryptocurrency market, aiding in investment decision-making, and risk management.

As mentioned in the abstract, I'll be using a Box-Jenkins approach as well as GARCH modelling to try and understand BTC price fluctuation as well as fluctuations in its volatility.

Data

The data I chose contains daily observations of Bitcoin (BTC) metrics in the cryptocurrency market. It contains 7 variables: `Date`, `Open`, `High`, `Low`, `Close`, `Adj.Close`, and `Volume`. For the purposes of this project, I'll be using the `Adj.Close` variable over time for time series analysis. The dates in this data range from 9/17/2014 to 5/05/2022. There are 2788 total daily observations during this period. The data was obtained from [https://www.kaggle.com/datasets/surajjha101/analyzing-and-prediction-of-bitcoin-pricing] where the author was able to use an API called Bitfinex to collect the final dataset. As mentioned above, I'm interested in studying this dataset to gain insight into the dynamics of BTC price and its volatility to better understand the cryptocurrency market as a whole.

```
btc_data <- read.csv('data/BTC-USD.csv')
head(btc_data)
```

```
##         Date    Open    High     Low   Close Adj.Close   Volume
## 1 2014-09-17 465.864 468.174 452.422 457.334   457.334 21056800
## 2 2014-09-18 456.860 456.860 413.104 424.440   424.440 34483200
## 3 2014-09-19 424.103 427.835 384.532 394.796   394.796 37919700
## 4 2014-09-20 394.673 423.296 389.883 408.904   408.904 36863600
## 5 2014-09-21 408.085 412.426 393.181 398.821   398.821 26580100
## 6 2014-09-22 399.100 406.916 397.130 402.152   402.152 24127600
```

Methodology

In this section, I'll briefly explain the methodology of the approaches I'll be using for this project.

Box-Jenkins Approach

The Box-Jenkins approach is a widely used methodology for modeling and forecasting time series data. It follows a systematic three-step process: model identification, estimation, and diagnostic checking.

In the model identification step, the Box-Jenkins approach analyzes the properties of the time series to determine the appropriate order of the Autoregressive (AR), Integrated (I), and Moving Average (MA) components of the model. This is achieved through the examination of autocorrelation and partial autocorrelation plots to identify potential AR and MA terms. Additionally, differencing is applied to achieve stationarity if the series is non-stationary.

Once the model is identified, the estimation step involves estimating the parameters of the ARIMA model using methods such as maximum likelihood estimation. This involves finding the values of the parameters that maximize the likelihood of observing the given data.

Finally, in the diagnostic checking step, the residuals of the estimated model are analyzed to ensure that they exhibit randomness and do not possess any remaining patterns or autocorrelations. Various statistical tests and graphical techniques, such as the Ljung-Box test and residual plots, are employed to assess the adequacy of the model. If the residuals exhibit significant patterns or correlations, adjustments may be made to the model.

The Box-Jenkins approach iterates through these three steps, refining the model until an appropriate and satisfactory model is obtained. This methodology allows for the identification of an optimal ARIMA model that captures the key characteristics and dynamics of the time series, providing a foundation for accurate forecasting and analysis.

GARCH Modelling

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model is a widely employed methodology for modeling time series data, particularly for capturing heteroskedasticity. The GARCH approach consists of some key steps.

Initially, in the specification stage, the GARCH model determines the appropriate order of the autoregressive (AR) and moving average (MA) components to describe the time series. This is typically done by examining the autocorrelation and partial autocorrelation plots of the time series. The selected ARMA model captures the mean dynamics of the data.

In the estimation phase, the GARCH model estimates the conditional variance of the series, which represents the time-varying volatility. It accomplishes this by fitting an autoregressive conditional heteroskedasticity (ARCH) model to the squared residuals of the mean equation. The ARCH model captures the dependency of the conditional variance on past squared residuals.

Next, the GARCH model extends the ARCH model by incorporating lagged conditional variances in addition to squared residuals. This is done to account for the persistence of volatility. The GARCH model estimates the parameters of the conditional variance equation, which include the ARCH and GARCH coefficients, using methods such as maximum likelihood estimation.

Once the GARCH model is estimated, model diagnostics are performed. This involves examining the residuals to ensure that they are uncorrelated and exhibit no remaining patterns. Various statistical tests, such as the Ljung-Box test, are used to assess the adequacy of the model.

In summary, the GARCH model approach involves specifying the mean equation, estimating the conditional variance, and conducting model diagnostics. This methodology allows for the modeling and forecasting of time series data with time-varying volatility.

Box-Jenkins Approach

Loading in Data / Setting up Workspace

Loading in Data

```
btc_data <- read.csv('data/BTC-USD.csv')
head(btc_data)
```

```
##         Date    Open    High     Low   Close Adj.Close   Volume
## 1 2014-09-17 465.864 468.174 452.422 457.334   457.334 21056800
## 2 2014-09-18 456.860 456.860 413.104 424.440   424.440 34483200
## 3 2014-09-19 424.103 427.835 384.532 394.796   394.796 37919700
## 4 2014-09-20 394.673 423.296 389.883 408.904   408.904 36863600
## 5 2014-09-21 408.085 412.426 393.181 398.821   398.821 26580100
## 6 2014-09-22 399.100 406.916 397.130 402.152   402.152 24127600
```

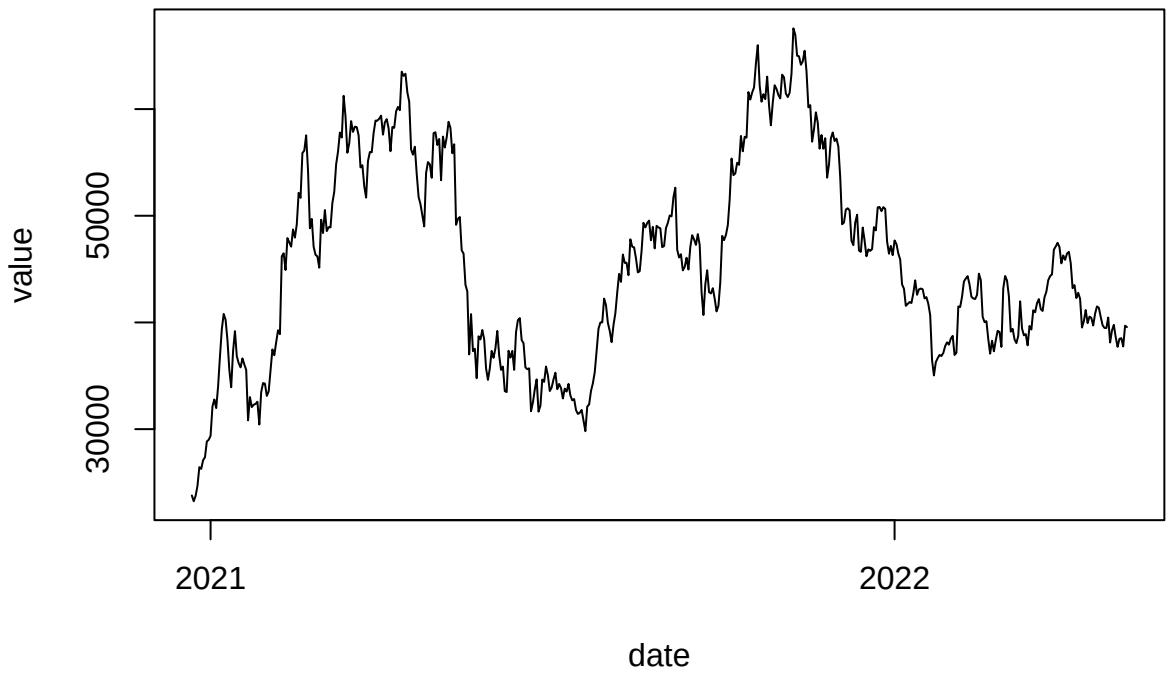
```
df1 = data.frame(date = btc_data$Date[2289:2788], adj_close = btc_data$Adj.Close[2289:2788])
head(df1)
```

```
##         date adj_close
## 1 2020-12-22  23783.03
## 2 2020-12-23  23241.35
## 3 2020-12-24  23735.95
## 4 2020-12-25  24664.79
## 5 2020-12-26  26437.04
## 6 2020-12-27  26272.29
```

In the above chunk, we're taking the 500 most recent observations of BTC adjusted closing price data in order to capture the most recent trends in the data. If we were to include the whole dataset, our estimated model's results would be drastically different as the shape of the data wouldn't be representative of recent BTC price, negatively influencing our time series modelling and producing less accurate results. Looking at the head of our new dataframe, we can see that the first observation takes place on 12/22/2020 rather than 9/17/2014.

Plotting Raw Data

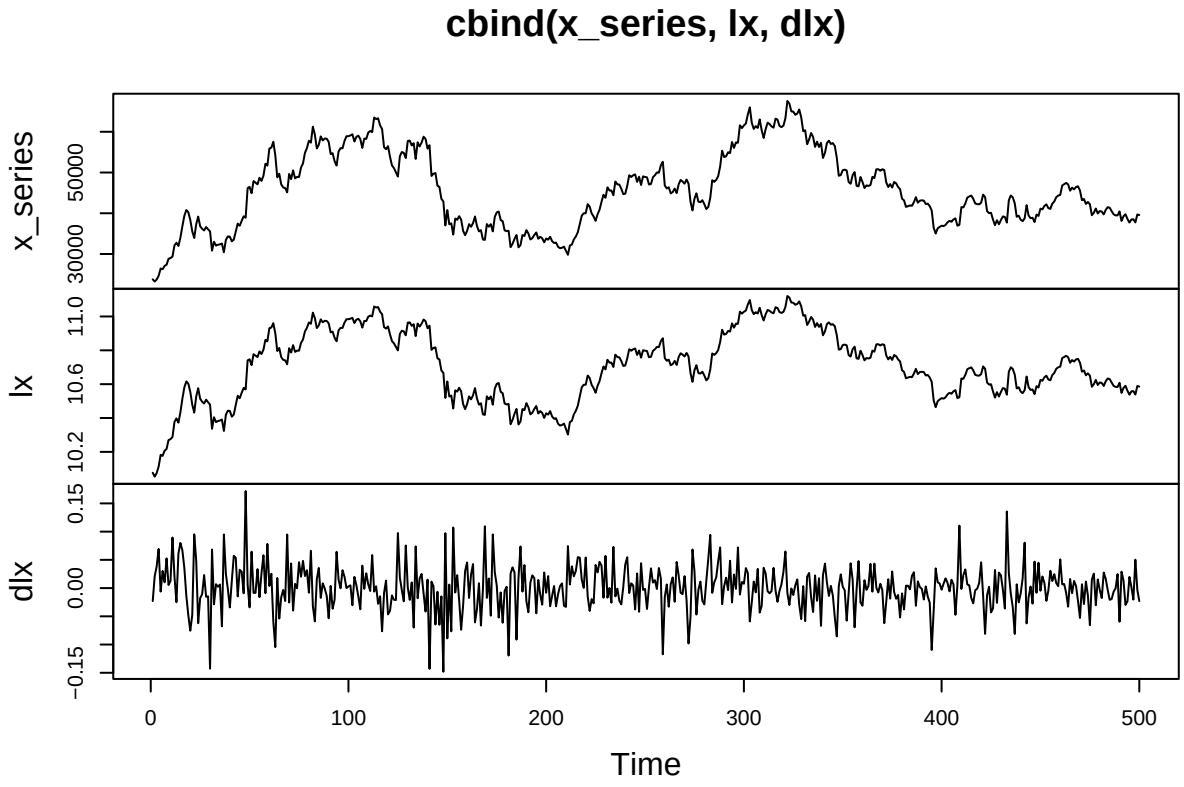
```
# Convert data frame to time series with appropriate frequency
ts_aclose_data <- ts(df1$adj_close, start = c(2020, 12, 22), frequency = 365)  # Assuming daily observa

# Create time series data frame with ordered dates
ts_aclose_df <- data.frame(date = as.Date(df1$date), value = ts_aclose_data)
ts_aclose_df <- ts_aclose_df[order(ts_aclose_df$date), ]

# Plot the time series
plot(ts_aclose_df, type = "l")
```



As we can see, the raw data appears to be rather volatile. We'll take the log of the data to stabilize its variance in order to continue with the next parts of the project.

Transforming the Data

```
x <- ts_aclose_df
x_series = x$value      # actual adjusted close price values
lx = log(x$value)       # logged values to stabilize variance
dlx = diff(lx)          # logged difference to make series stationary
plot.ts(cbind(x_series, lx, dlx))
```



After taking the log of the raw data, it still doesn't seem stationary. We difference the logged data in order to achieve stationarity. Due to the fact that our data doesn't look stationary after using a log transform, let's use the auto.arima() function to calculate the best ARIMA(p,d,q) model for the logged data based on values of AIC and BIC. The function searches over a range of possible p, d, q values in order to determine the ones that provide the best performance metrics. In order to not hinder model performance, let's increase the max p and max d values to 30.

```
auto.arima(lx, max.p = 30, max.d = 30)
```

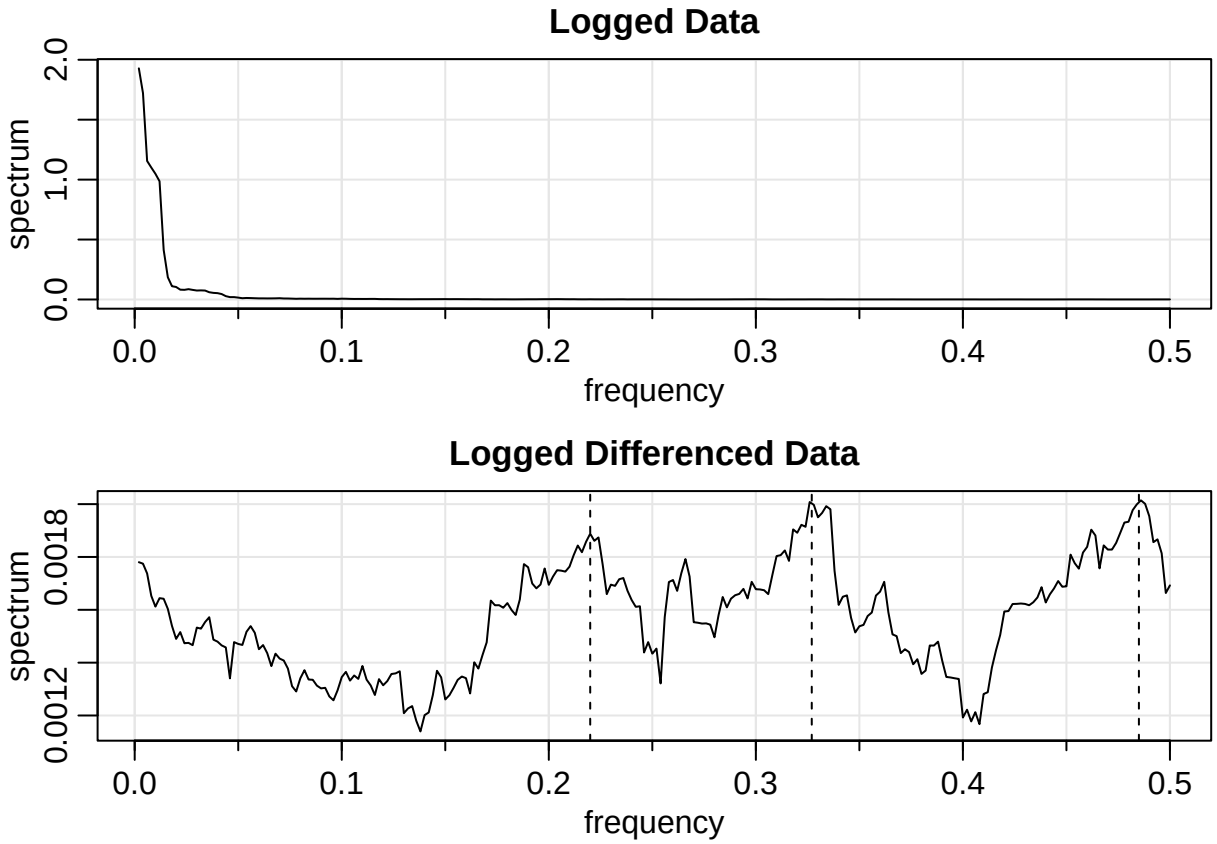
```
## Series: lx
## ARIMA(0,1,0)
##
## sigma^2 = 0.00158:  log likelihood = 901.4
## AIC=-1800.8   AICc=-1800.79   BIC=-1796.59
```

The output of the following chunk indicates that the time series data can best be described by a random walk, with i = 1 to achieve stationarity from the logged data. It can be described as a random walk as each observation in the data is the cumulative sum of random errors, there is no autoregressive or moving average components to the model. In the following chunks, let's look at some of the time series data's characteristics to see if an ARIMA(0,1,0) model is the best fit for our data.

Spectral Analysis

In this section, we're trying to assess the validity of modelling our data with an ARIMA(0,1,0) model. Here, we're trying to identify seasonal components through periodograms of the logged data and logged difference of data.

```r
# why is differenced logged appear to have significant peaks, whereas logged doesn't --> (differenced l
par(mfrow = c(2,1))
mvspec(lx, kernel('daniell', 4), main = 'Logged Data')
mvspec(dlx, kernel('daniell', 20), main = 'Logged Differenced Data')
abline(v = c(0.22,0.327,0.485), lty = 2)
```

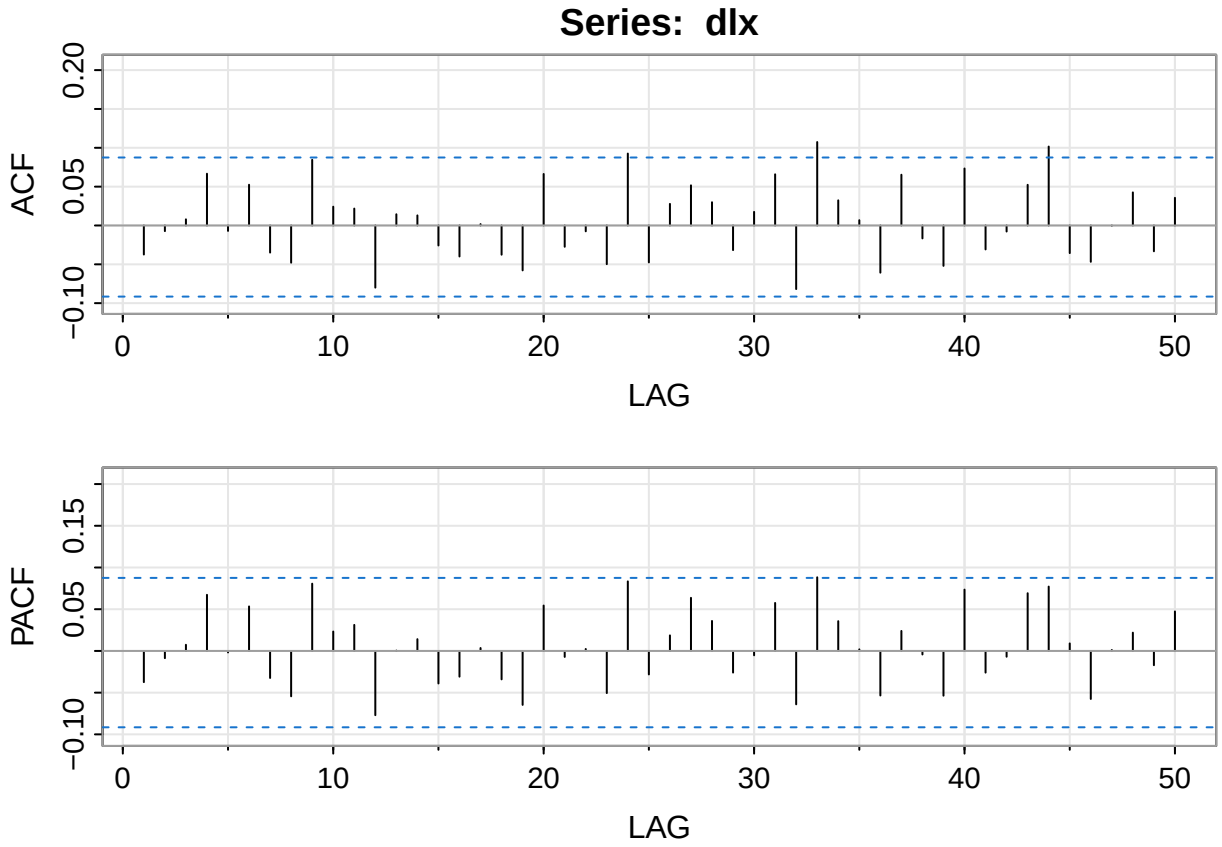


From the output of the above chunk, we can't see any significant peaks in the logged data which doesn't indicate seasonal components. However, the logged differenced data shows some peaks, but the magnitude of the strength is very low. We are unable to say that there exists any seasonal components in this time series data.

ACF & PACF

So far we've identified the lack of seasonality in our data. Let's look at the autocorrelation function (ACF) and partial autocorrelation function (PACF) of our data to see if we should incorporate any autoregressive or moving average components into our model.

```
acf2(dlx, 50)
```

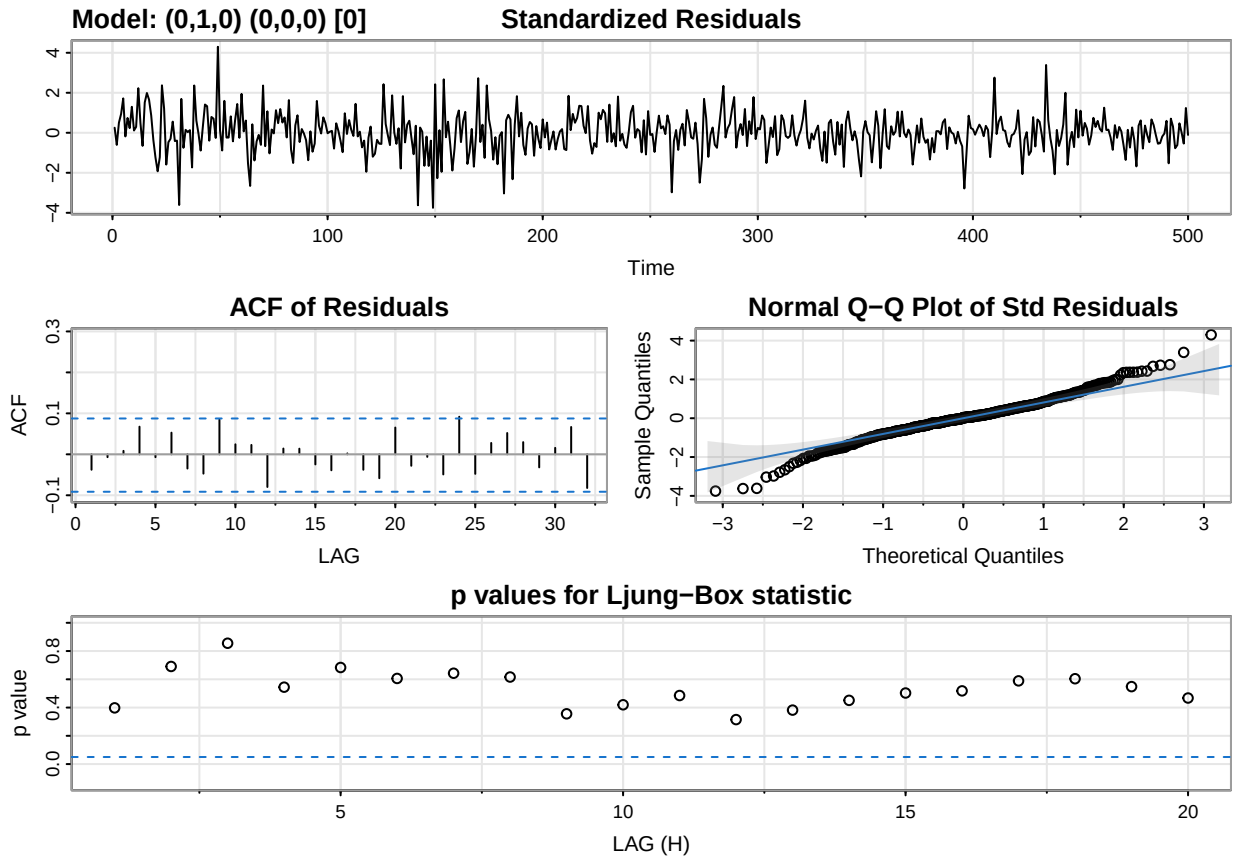


Looking the ACF and PACF of the transformed (logged & differenced) data, there's no significant lags before lag 24. Since there's no seasonal component to the data, it wouldn't make sense to incorporate at least 24 lags into the model. We can keep this in the back of our minds as we continue, but so far, it's looking like we'll stick with the ARIMA(0,1,0) model generated by the auto.arima() function.

Model Diagnostics

So far, we've seen no indication of seasonality in the time series data. From the ACF and PACF, we've also not seen any significant lags that would feasibly make sense to incorporate into our model. Everything points to our model being a random walk model, so let's look at the diagnostics of the residuals after we fit an ARIMA(0,1,0) model to our data.

```
f1 <- sarima(lx, 0, 1, 0, 0, 0, 0, 0)
```



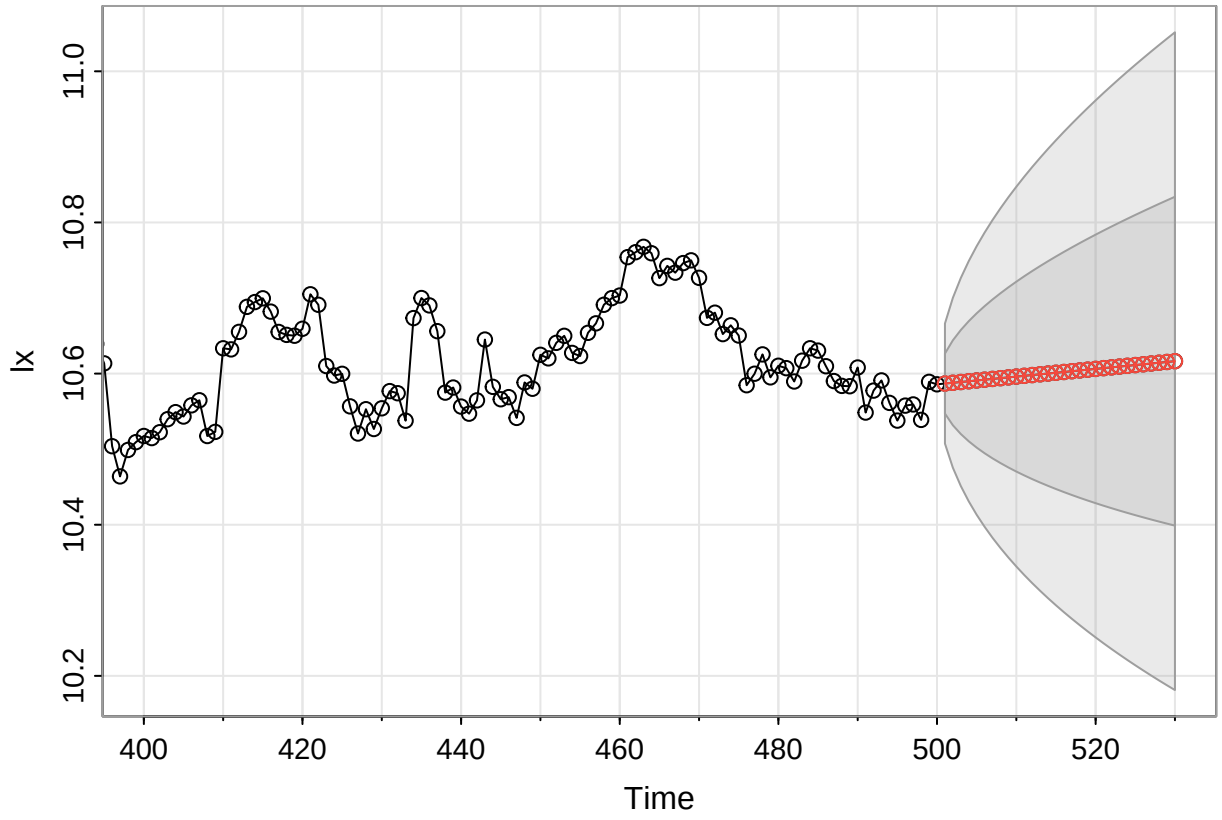

```
f1
```

From the output of the model diagnostic plot, we can see that the ACF of the residuals don't seem to exhibit significant spikes for almost all lags. The ACF however does seem to indicate that lag 24 is significant while examining the residuals of the model. However, the residuals seem to follow the theoretical quantiles on normal Q-Q plot for the most part, and the p-values for the Ljung-Box statistic are not significant- indicating that there doesn't seem to be a correlation among the errors, and that the ARIMA(0,1,0) is a good fit for the data.

Forecasting

Finally, we'll use our ARIMA(0,1,0) random walk process to forecast the next 30 days.

```
sarima.for(lx, 30, 0, 1, 0, 0, 0, 0, 0)
```



Looking at numeric values of forecast.

```
model <- arima(lx, order = c(0,1,0))
arima_forecast <- forecast(model, 30)
print(arima_forecast)
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 501        10.5858 10.53487 10.63673 10.50790 10.66369
## 502        10.5858 10.51377 10.65782 10.47564 10.69595
## 503        10.5858 10.49758 10.67401 10.45088 10.72071
## 504        10.5858 10.48393 10.68766 10.43001 10.74158
## 505        10.5858 10.47191 10.69968 10.41162 10.75997
## 506        10.5858 10.46104 10.71055 10.39500 10.77659
## 507        10.5858 10.45105 10.72055 10.37971 10.79188
## 508        10.5858 10.44174 10.72985 10.36548 10.80611
## 509        10.5858 10.43300 10.73859 10.35212 10.81947
## 510        10.5858 10.42474 10.74685 10.33948 10.83211
## 511        10.5858 10.41688 10.75472 10.32746 10.84414
## 512        10.5858 10.40937 10.76223 10.31597 10.85562
## 513        10.5858 10.40216 10.76943 10.30495 10.86664
## 514        10.5858 10.39523 10.77636 10.29435 10.87724
## 515        10.5858 10.38854 10.78305 10.28412 10.88747
## 516        10.5858 10.38207 10.78952 10.27423 10.89737
## 517        10.5858 10.37580 10.79579 10.26464 10.90696
## 518        10.5858 10.36971 10.80188 10.25533 10.91627
## 519        10.5858 10.36379 10.80780 10.24627 10.92532
```

```
## 520       10.5858 10.35803 10.81357 10.23745 10.93414
## 521       10.5858 10.35240 10.81919 10.22885 10.94274
## 522       10.5858 10.34691 10.82468 10.22045 10.95114
## 523       10.5858 10.34154 10.83005 10.21224 10.95936
## 524       10.5858 10.33629 10.83531 10.20420 10.96739
## 525       10.5858 10.33114 10.84045 10.19633 10.97526
## 526       10.5858 10.32610 10.84550 10.18862 10.98297
## 527       10.5858 10.32115 10.85044 10.18106 10.99054
## 528       10.5858 10.31629 10.85530 10.17363 10.99796
## 529       10.5858 10.31152 10.86007 10.16633 11.00526
## 530       10.5858 10.30684 10.86476 10.15916 11.01243
```

In all, a Box-Jenkins approach to modelling our data yields a random walk as the best model. As we can see from above, the ARIMA(0,1,0) process will continue to forecast the last value given by lx since there are no auto regressive (AR) components. However, the variance will increase as the forecast horizon increases, due to the accumulation of random error terms from the model. We can see this in the graphical representation of the forecast, as well as in the increasingly large confidence intervals of the numeric representation.

GARCH Modeling

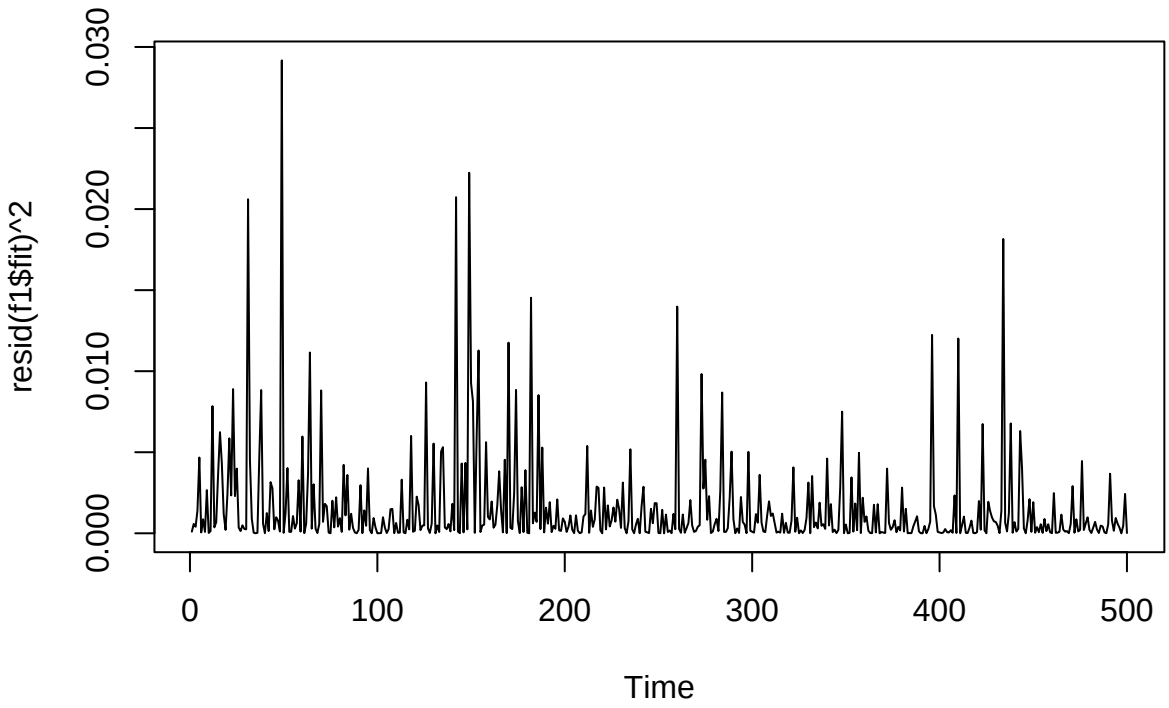
Testing for GARCH

Now that we've got our model from the earlier section, we're going to have to inspect the model's squared residuals in order to determine the presence or absence of correlation among the residuals.

Plot of Squared Residuals

Let's start off by plotting the squared residuals from the ARIMA(0,1,0) model that we got above.

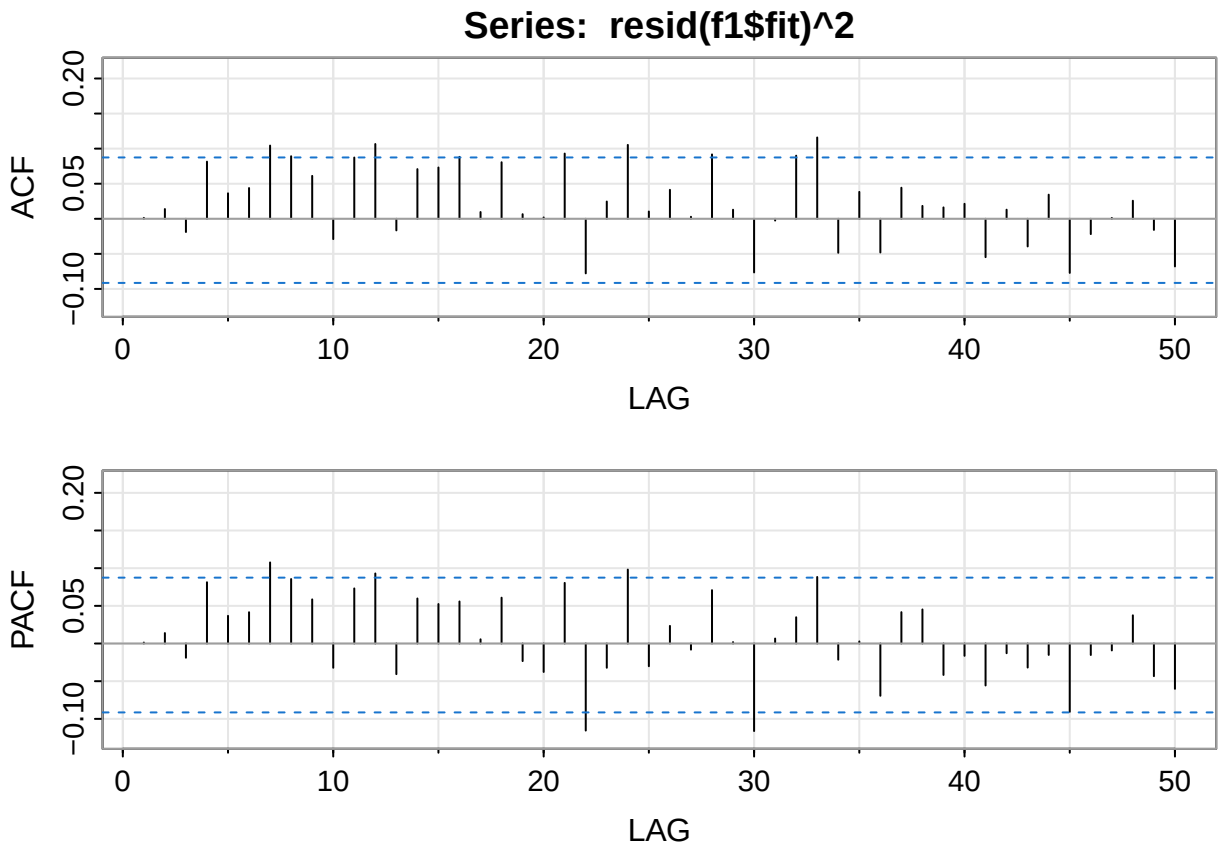
```
# note: f1 is the model we've fit above to the data
plot(resid(f1$fit)^2)
```



If the volatility of the returns was constant, we would expect the squared residuals of our fitted model to not follow any trends or patterns over time. Here, although the pattern isn't extremely clear, we can see that there appear to be some periods of higher volatility followed by periods of lower volatility. Typically, this is an indication that a GARCH model would suit the data, and that conditional variance can be modeled. Next, let's examine the acf and pacf of the data to see if we can uncover any trends about the dependent nature of the model's squared residuals.

ACF & PACF of Squared Residuals

```
acf2(resid(f1$fit)^2, 50)
```



From looking at the ACF and PACF of the squared residuals of our model, there's no immediate significant spikes to inform us about the nature of the squared residuals' dependency. We can see a significant spike at lag 7 in the ACF as well as the PACF. This could potentially indicate a GARCH(7,7) model would be a good fit for the residuals. Although this may be true, we don't see any significant lags before 7, indicating that a GARCH(7,7) fit may be introducing unnecessary coefficients into the model- so we'll not fit the GARCH(7,7) model to the residuals.

However, since there does seem to be some clusters of volatility looking at the plot of squared residuals above, we'll fit a GARCH(1,1) model on the residuals- a 'gold-standard' model. We choose the GARCH model over an ARCH model in this case because GARCH models incorporate lagged conditional variance as well as past errors to get a model that is smoother than an ARCH model. Whereas, the ARCH model only includes past squared error, capturing the auto regressive nature of volatility. Since there's not much to suggest our squared errors follow an auto regressive process (looking at the ACF and PACF), we'll use GARCH modelling to incorporate conditional variance into our model- making it more flexible in explaining volatility.

GARCH Model Fitting

Let's start off by fitting our GARCH(1,1) model to the residuals of the ARMA(0,0) fit to the data. Note that the p-values given in the estimation paragraph are two-sided, so they should be halved when considering the GARCH parameters according to the fGARCH package.

```r
g_fit1 <- garchFit(~arma(0,0) + garch(1,1), dlx)
```

```r
summary(g_fit1)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 0) + garch(1, 1), data = dlx)
##
## Mean and Variance Equation:
##  data ~ arma(0, 0) + garch(1, 1)
## <environment: 0x132880df0>
##  [data = dlx]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 7.8995e-04  4.4616e-05  3.0998e-02  9.3938e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     7.899e-04   1.720e-03    0.459   0.6461
## omega  4.462e-05   2.549e-05    1.750   0.0801 .
## alpha1 3.100e-02   1.217e-02    2.547   0.0109 *
## beta1  9.394e-01   2.264e-02   41.496   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  909.5369    normalized:  1.822719
##
## Description:
##  Tue Jun 13 18:42:20 2023 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  65.82667  5.107026e-15
##  Shapiro-Wilk Test  R    W      0.9780031 7.685863e-07
##  Ljung-Box Test     R    Q(10)  8.871457  0.5443448
##  Ljung-Box Test     R    Q(15)  10.91151  0.7588417
##  Ljung-Box Test     R    Q(20)  14.56245  0.8008732
##  Ljung-Box Test     R^2  Q(10)  9.719692  0.4654203
##  Ljung-Box Test     R^2  Q(15)  15.12811  0.4422293
##  Ljung-Box Test     R^2  Q(20)  18.95679  0.5246362
##  LM Arch Test       R    TR^2   11.97868  0.447393
```

```
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.629406 -3.595638 -3.629533 -3.616154
```

What we can draw from the output of this model fit is that our GARCH parameters are significant, and therefore do explain the variance of the residuals of our fitted model.

We'll also note that all of the Ljung-Box Tests are not statistically significant. These tests are used as a diagnostic tool in the analysis of residuals from a time series model, such as an ARIMA or a GARCH model. They test for the presence of autocorrelation in the series. In this case, we're testing for significant autocorrelation among the residuals up to a certain lag. Because none of the tests are statistically significant, we can see that there's no evidence of significant autocorrelation among the residuals of our GARCH model, indicating our GARCH model is a good fit for the data.

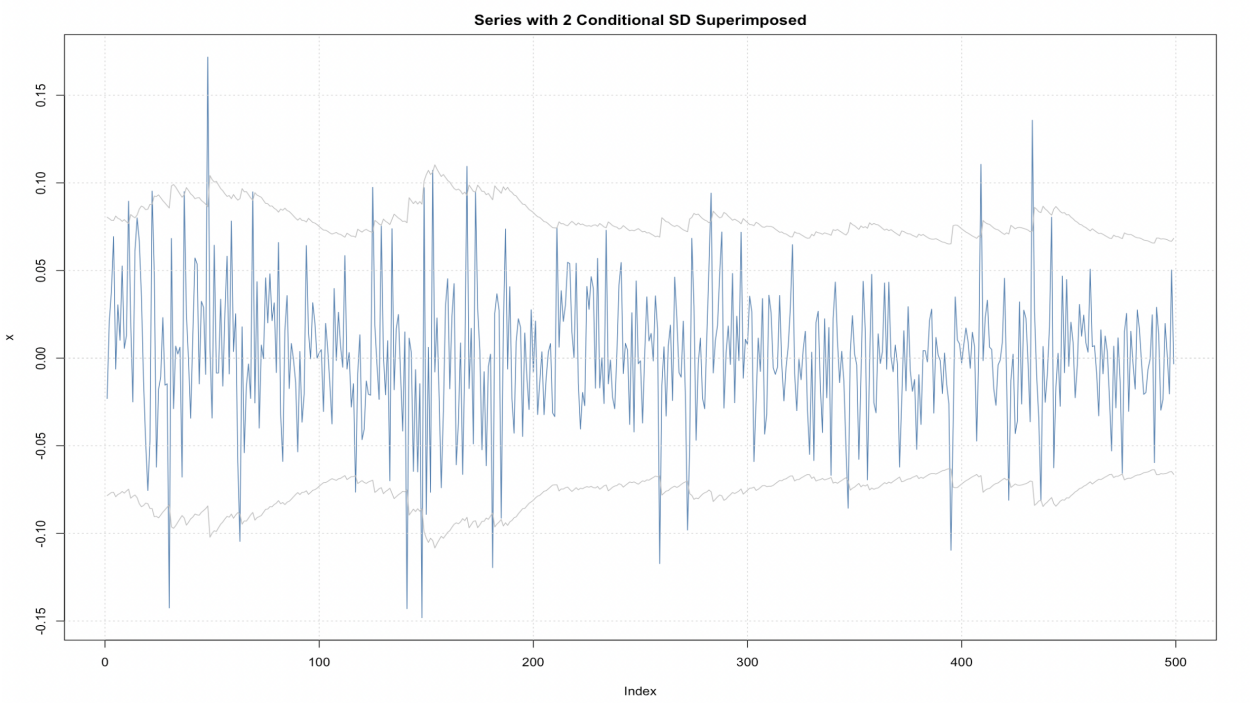
From the output of the table, we can see that both the autoregressive and moving average components of the ARMA fit aren't significant (to be expected because our best model fit is a random walk- ARMA(0,0)). We can also see that all the parameter estimates from the GARCH(1,1) model are significant in explaining the variance.

We can draw the following conclusions about our data from the GARCH(1,1) model we fit to the residuals of the ARIMA(0,1,0) model of our data.

- From ARIMA(0,1,0) fit on logged data, ARMA(0,0) fit on differenced, logged data (return)
  - $r_t = \sigma_t \epsilon_t = log(\nabla(x_t))$
  - $r_t = \epsilon_t$ where $\epsilon_t \sim N(0, 1)$
- Adding on GARCH parameter estimates to model variance (volatility)
  - $\sigma_t^2 = 0 + 0.031\epsilon_{t-1}^2 + 0.939\sigma_{t-1}^2$, where $\omega = 0$, $\alpha_1 = 0.031$, and $\beta_1 = 0.939$ from the output table.

Now, let's plot the conditional variance from our GARCH(1,1) model to visualize it

```
plot(g_fit1)
```



Here, we can see that our model does well to capture the smaller clusters in volatility that we discussed earlier. We can see that periods of higher volatility correspond with periods of higher standard deviation estimates. Because these volatility clusters tend to happen next to each other, we can also see that the standard deviation estimates decrease when the volatility decreases as well. Our model does well to capture the nature of the volatility of our data. This is especially apparent in the initial time periods where greater periods of volatility are followed by lower periods of volatility.

Conclusion and Future Study

Box-Jenkins Approach

In the end, when employing a Box-Jenkins approach to model our data, we have determined that a random walk is the most suitable model. As demonstrated earlier, the ARIMA(0,1,0) process predicts the latest value, lx, since there are no autoregressive (AR) components involved. However, it is important to note that as the forecast horizon extends, the variance of the forecasted values grows. This phenomenon arises due to the cumulative effect of random error terms from the model- as the only contribution to future estimates is the random error term.

As seen through the plot at the end of this section, as we extend our predictions further into the future, the spread of the forecast widens, indicating increasing uncertainty. This widening spread signifies that the confidence in the forecasted values decreases with time. Moreover, examining the numeric representation of the forecasted data reinforces this observation, as the confidence intervals accompanying the numeric values become progressively wider.

This pattern has significant implications for making predictions using this specific model. While the ARIMA(0,1,0) process may be appropriate for short-term forecasting, it doesn't seem to be reliable as the forecast horizon gets larger. The accumulating random errors lead to growing uncertainty, making it increasingly challenging to generate accurate predictions further into the future. Consequently, relying solely on this model for long-term forecasting may result in unreliable and imprecise projections.

GARCH Modeling

When utilizing GARCH models to describe the underlying trends in the volatility of our model, the GARCH(1,1) model best fit the residuals of the ARIMA(0,1,0) model we fit on the log transformed data.

First, we analyzed the behaviour of the squared residuals to assess whether or not volatility was constant.In the scenario where returns' volatility remains constant, we would expect the squared residuals from our fitted model to exhibit a lack of any trends or patterns over time. However, upon closer inspection, we observed periods of heightened volatility followed by periods of reduced volatility. Although the pattern was not overly pronounced, it suggested the presence of varying volatility- which we could address through fitting a GARCH model.

Our GARCH model did well in capturing the conditional variance of our data, however the ACF and PACF did not show enough significant lags to suggest the fitting of anything other than a GARCH(1,1) model. Fitting this GARCH(1,1) model, we observe that it performs well, as the estimated coefficients were significant, and the corresponding Ljung-Box tests were not statistically significant- indicating no autocorrelations between residuals.

Overall, as seen in the visual representation of our GARCH model, it did well in capturing the variance of the model. From our model, we're able to discern the trend of clusters with high volatility being accompanied by high estimated variance, and clusters of low volatility being accompanied by low estimated variances.

Future Study & Considerations

From the start, I was aware that modelling Bitcoin prices would be a difficult task given what I've learned in the course so far. This is true for various reasons. Two of the main reasons why that is are as follows:

Firstly, Bitcoin is a highly volatile and speculative asset, it is characterized by significant price swings and extreme fluctuations. This volatility presents challenges in identifying and capturing the underlying patterns and trends in the data.

Secondly, Bitcoin is a relatively young and evolving market, which means that historical data might not provide a comprehensive representation of its future behavior. The absence of long-term data limits the

accuracy and reliability of models, as they heavily rely on past observations for parameter estimation and forecasting. This is especially true in this project, as I had to omit the first 2200 observations due to the data exhibiting different patterns as compared to the most recent 500 observations. This is truly indicative of BTC's novelty and the issues it presents in trying to model its movement.

If I were to continue this project, I would definitely look into models that capture trends in volatility. I'd research and implement stochastic volatility models and other state space models, as treating the actual closing price as a state observation could help increase model performance. I would also research and implement models such as Long Short-Term Memory (LSTM) models that able to capture more complex, non-linear dynamics within the data.