STAT 3504 Time Series Forecasting: Bitcoin

Andrew O'Drain

December 6th 2022

Abstract

In this project I attempted to build a time series model of Bitcoins price action to forecast its possible future values. My main finding is that Bitcoins price is very hard or almost impossible to forecast using time-series methods. The primary reason for this is that financial time series, when broken down, transformed, and analyzed, reveals itself to be nothing more than a random walk that cannot be modeled appropriately, and reliably. This is because differencing is used to de-trend and make the data stationary. The first difference of a random walk model is white noise, and white noise is stochastic in nature. This implies very little to no auto-correlation and if we have none to very little auto-correlation, there is nothing to build a predictive model on. However, during the analysis the ACF and $PACF$ of the logged first difference of Bitcoins price showed some significant spikes at *lags 6*, *10*, and *33*. Therefore, I attempted a model anyway.

Introduction

My primary motivation for building a time series model for Bitcoin is my interest in the domain of finance and digital assets. The data was downloaded from the yfinance API in the daily interval format. My approach to building the models was general, meaning that even if I thought a model wouldn't be significant I built it anyway. This helped me focus and absorb more of the information that was being relayed to me. It also helped me narrow down and intuitively see what time series analysis is all about. You could say that it was a series of "Aha!" moments. Those are the moments I wait for when learning a new topic. It what makes all the work worthwhile.

The Linear Model

The first model that I ruled out while searching for a trending component was the linear model. Even thought the summary view of the data showed the price action to be non-linear, I built the model anyway and used it as a reference to compare to other models. The equation for the model is very basic

$$y = \alpha + \beta t$$

$$\alpha = intercept$$
$$x = time$$

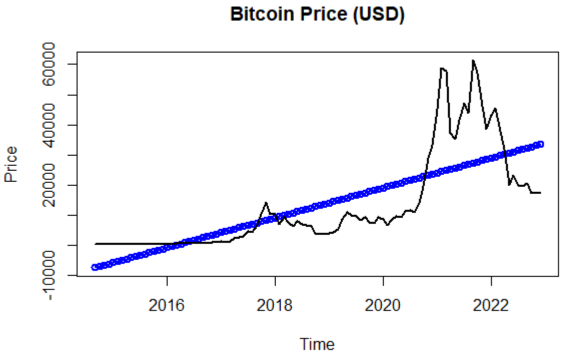

Figure 1: A plot of the linear model

As we can see the line approximates price very poorly. However, the coefficients are significant and R^2 reports that *55%* of the variation in price can be explained by time, but the residual standard error is large at 10,740.

The Quadratic Model

After performing the linear regression I decided to perform a log transform of the data to close the large gaps in price, and to make the outputs more consistent. I found that Bitcoin has a very prominent trend that can be modeled quadratically. I took the natural logarithm of the output and added a square-term.

$$ln_e[y] = \alpha + \beta_1 t + \beta_2 t^2$$

$$\alpha = intercept$$
$$t = time$$

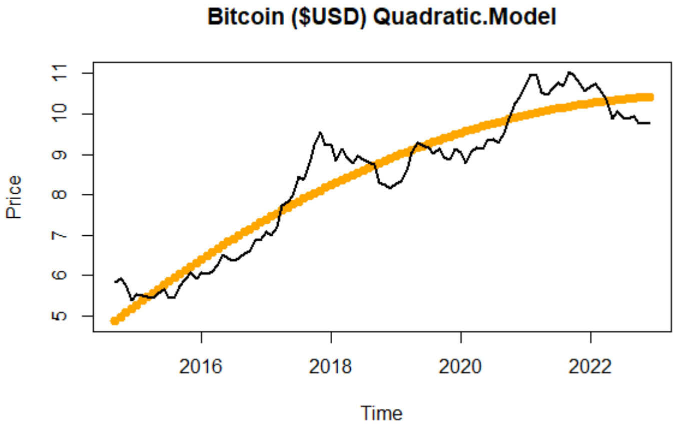

Figure 2: Coefficients: $-281098 + 277.81t - 0.069t^2$

This model was a much better fit to the data. The residual standard error is 0.54, multiple $R^2 = .906$ and the adjusted $R^2 = .904$ This implies that *90%* of the variation in price can be explained by time. The standard error for time-squared is only *.01*, while it is *42.30* for linear time.

The Harmonic Model

Just to make sure that I had the trend component figured our I tried a harmonic model. I first made a pair of harmonic functions then I added them into the regression equation. Here is the model equation

$$y = \alpha + \cos(2\pi t) + \sin(2\pi t) + t$$

$$\alpha = intercept$$
$$t = time$$

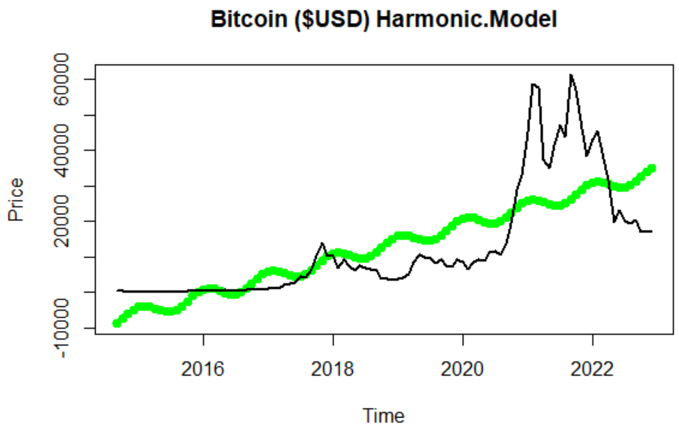

Figure 3: Coefficients: $-352740.74 + 4.159\cos(2\pi t) - 45.1\sin(2\pi t) + 170.92$

As we can see the trend that most closely models our logged data is the quadratic trend. This is confirmed with the lowest *AIC* of *166.29* and *BIC* of *176.71*. To give you an idea of how much better the quadratic trend is compared to the others, the *AIC's* and *BIC's* of the other models were in the *2000's* range.

Exploratory Analysis for fitting AR, ARIMA, ARMA, or MA

The Mann-Kendall test revealed that we are correct about there being a major global trend in the data. The test concluded that we can reject the null hypothesis with a p-value of $2.22x10^{-10}$. This implies a strong global monotonic trend in the data. Furthermore, an ADF test was performed to check for a unit-root. The ADF test revealed that we fail to reject the null and conclude the logged time series of Bitcoin price action contains a unit-root and is not stationary with a *p-value* of $p = .341$

Stationarity: Detrending and Differencing

As part of the exploratory analysis the first difference was taken and yielded some good results. It was found that after taking the first difference of the data that it was successfully de-trended and was stationary. Another Mann-Kendall test was performed and showed that we could fail to reject the null and conclude that there was no longer a significant global trend with a *p-value = .19*. Another ADF test was performed and revealed that the data was also now stationary at the .05 *significance level* with a *p-value of .01*.

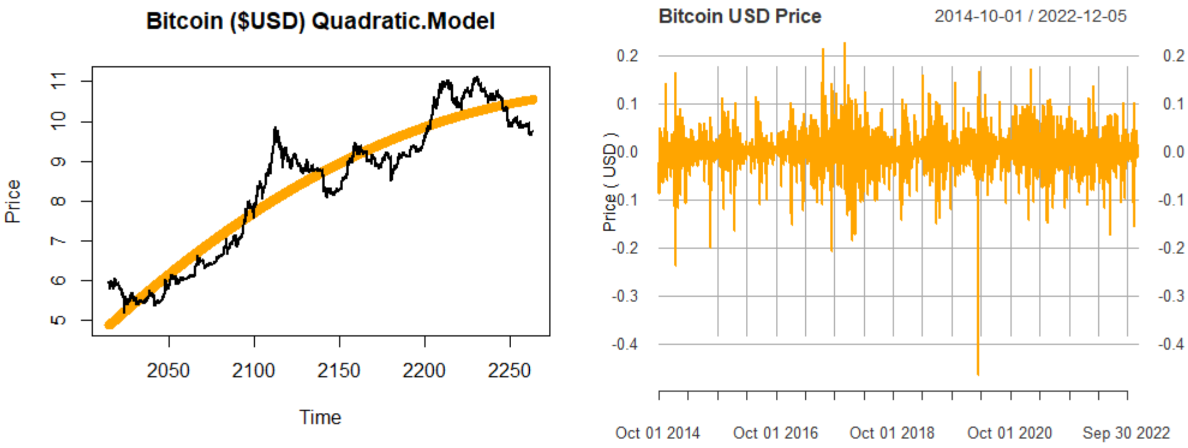

Figure 4: Left: quadratic trend, Right: de-trended stationary Bitcoin price

ACF & PACF for ARIMA(p,d,q) parameter estimation

After confirming that the data no longer carried a trend once the first difference was taken the *ACF* and the *PACF's* of the stationary data was evaluated to ensure that there was auto-correlation to build a model off of.

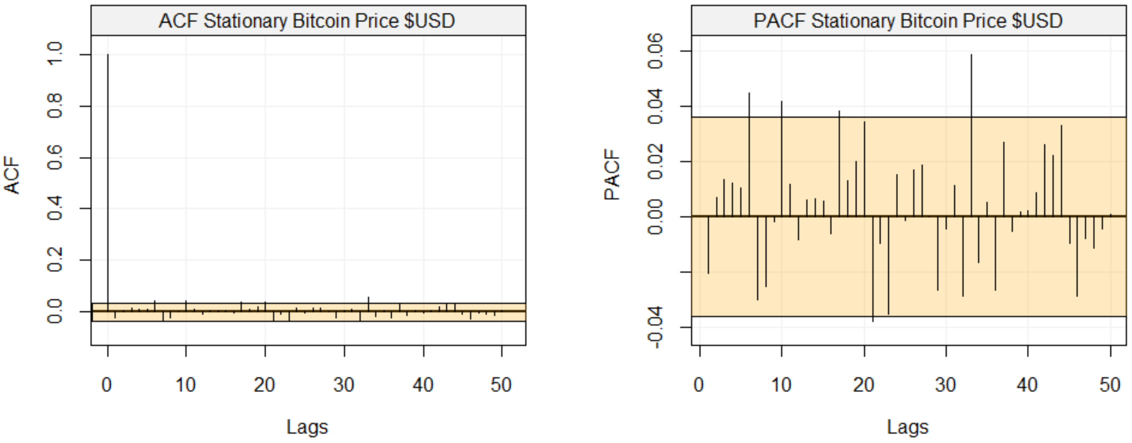

Figure 5: Left: Auto-Correlation Function, Right: Partial Auto-Correlation Function

Looking at the above plots we can see that our ACF almost looks like there is zero auto-correlation. However, this is only because of the size of the plot. Looking at the precise correlations it was found that on both the PACF and the ACF plots that lag-6, lag-10, and lag-33 were significant. There respective values are *.044*, *.042*, and *.059*. There is no tail-off on either of the plots suggesting that and *ARMA* model could be used to model the data. Please note that the *ACF* and *PACF* in the above plots are of the first difference.

Model Coefficients

The next step was to run the auto.arima function on our first differenced and logged data. The result was the coefficients of our model. Auto.arima specified that the series was an *ARMA(2,2)* with *AR1 = 1.29, AR2 = -0.9194, MA1 = -1.30, MA2 = .9449, AIC = -10938, and BIC = -10902*. I decided to run the undifferenced data through the auto.arima function, the output was *ARIMA(2, 1, 2)* with drift and had the same exact coefficients of the *ARMA(2,2)* model. After fitting the model, I ran the model diagnostics. If you look below you can see the results.

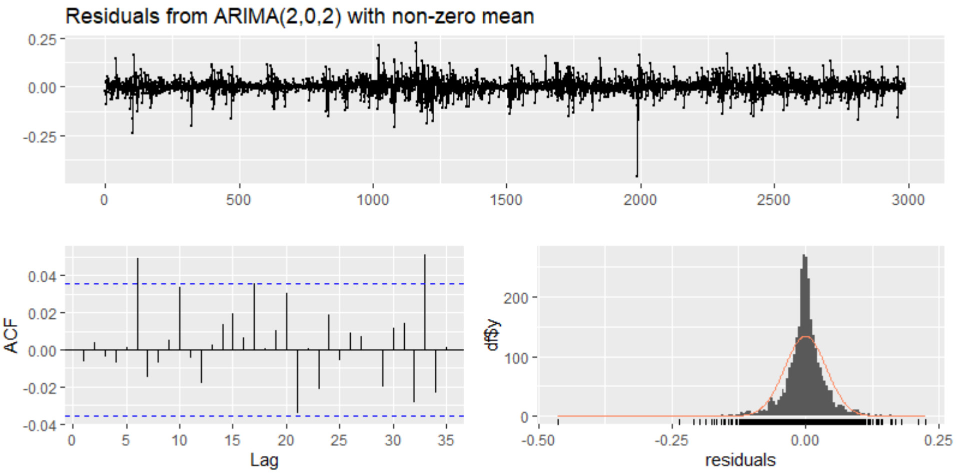

Figure 6: Residual Analysis of ARIMA(2,0,2), Top: residuals, Bottom Left: ACF, Bottom Right: normality check

I also ran the Ljung-Box test to test the independence of the residuals. It was concluded that the null hypothesis could not be rejected at the .05 significance level with a *p-value* of *.06* implying that the residuals were indeed independent.

Building the ARIMA & Forecasting

The next step was to actually build the *ARIMA* model. This is when I introduced the data that was not differenced first. Therefore, the model that was built was an *ARIMA(2, 1, 2)* with drift. I decided to create a forecast for the next 7 days. The forecast has a constant mean of $e^{9.74} = \$16,983.54$, and could drop as low as $e^{9.54} = \$13,904.94$ or go as high as $e^{9.95} = \$20,952.22$. In other words, with *95% confidence* price, over the next 7 days will be 9.7455 ± 0.0444. You can see the plot below

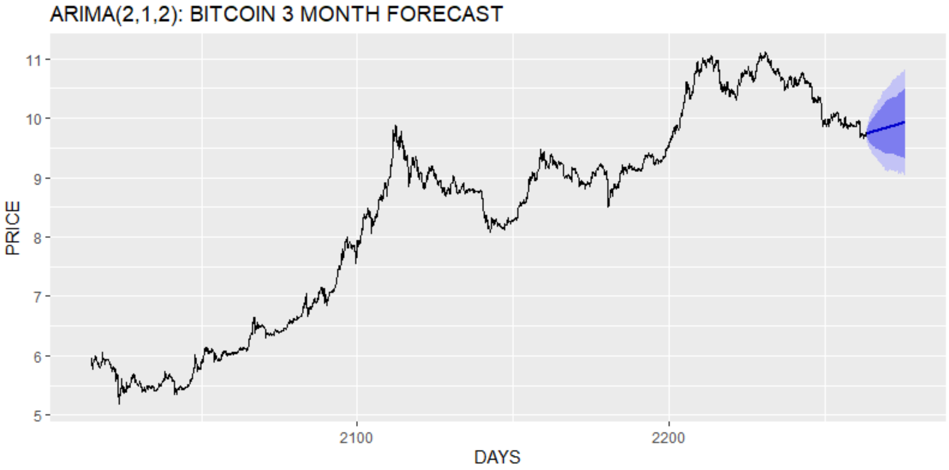

Figure 7: 150-Day Forecast ARIMA(2,0,2): A 150-Day forecast was performed for visual purposes

LINEAR MODEL: L.I.N.E. ASSUMPTIONS

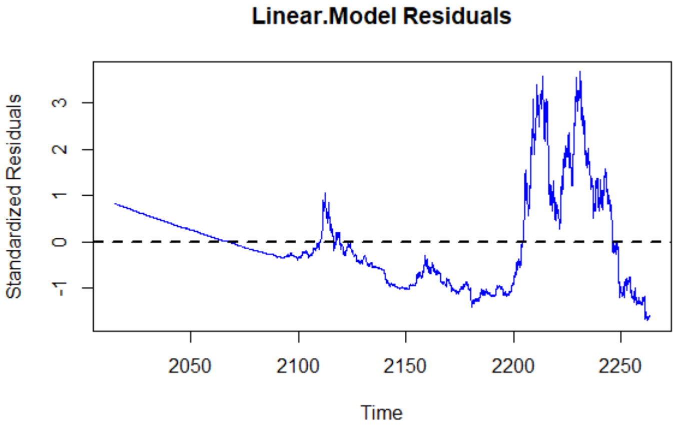

Figure 8: Standardized Residual Analysis: Residuals are unevenly distributed

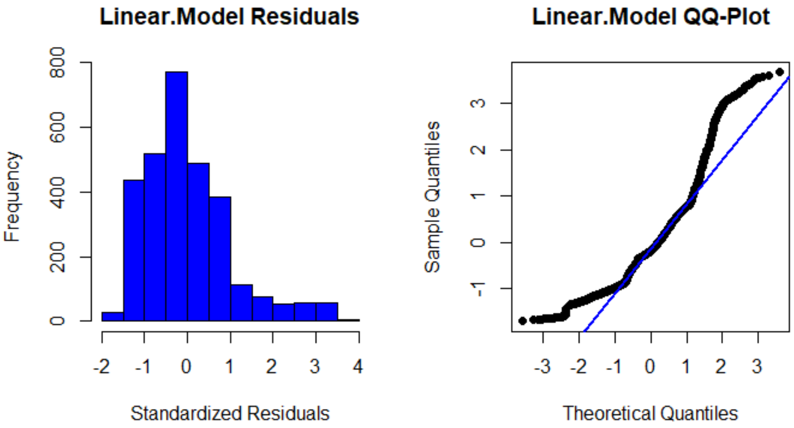

Figure 9: Normality Check: residuals not normally distributed

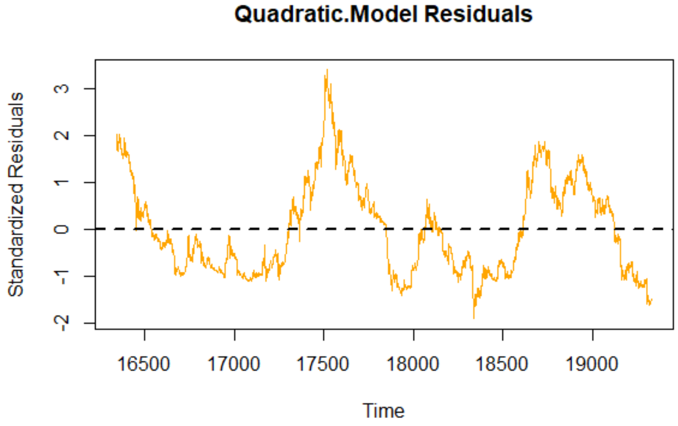

Figure 10: Standardized Residual Analysis: Residuals are unevenly distributed, but the most evenly distributed out of all models

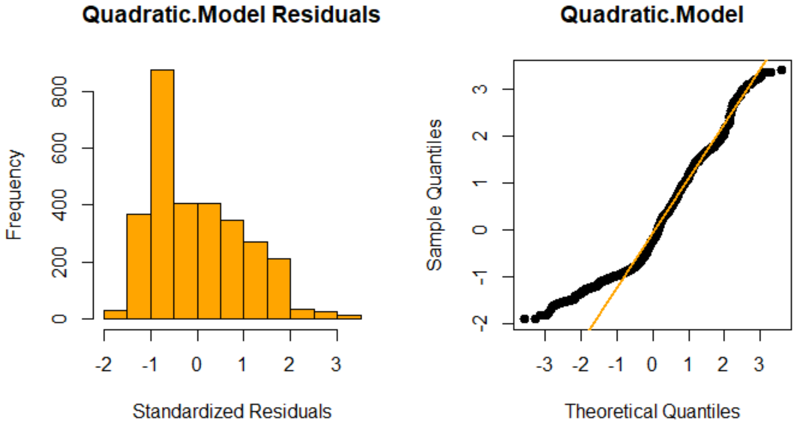

Figure 11: Normality Check: residuals not normally distributed, but show the most normality out of all models

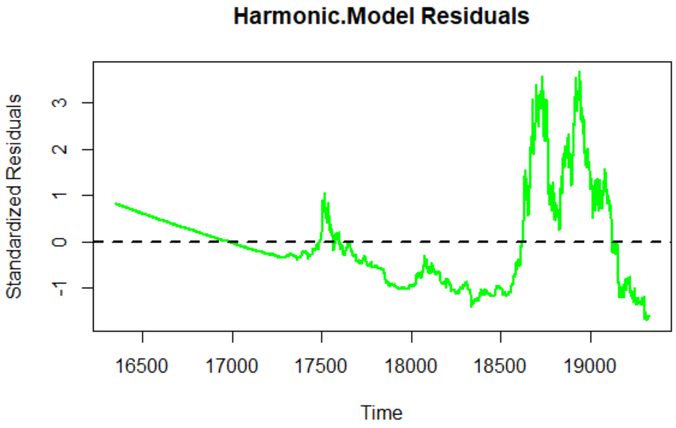

Figure 12: Standardized Residual Analysis: Residuals are unevenly distributed

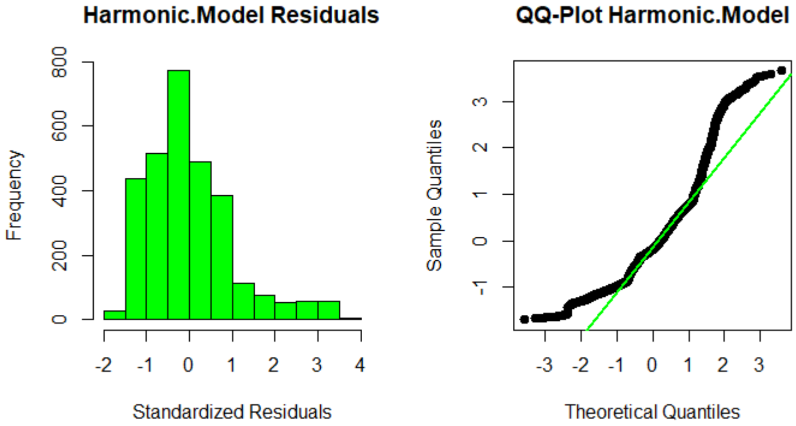

Figure 13: Normality Check: residuals not normally distributed