Final Project: Milestone 3

Evelina Ramoskaite[1,2]

[1] DSC 425 - Time Series Analysis and Forecasting

[2] DePaul University

Final Project: Milestone 3

Crypto-Currency and Stock Data - Bitcoin and AMD

As a group, we are exploring the relationships between several cryptocurrencies and stocks over time. This includes looking at cryptocurrency prices over time as they relate to each other, possible cryptocurrency-related stocks, and common index funds.

For my portion of milestone 3, I decided to explore cryptos as they relate to chip manufacturing companies, specifically Bitcoin and AMD. High-end GPUs are commonly utilized for mining cryptocurrencies, because they have faster processing times than CPUs. This has lead some investors to speculate that their sales may be positively correlated with high crypto prices. Chips for crypto mining made up a smaller chunk of AMD's sales relative to NVDA, so I expect that the relationship will be smaller, if a link between crypto prices and GPU sales exists.

The data we used was retrieved using the Yahoo Finance API with tidyquant, and includes the adjusted daily prices of stocks between 10/01/2016 and 09/20/2021. AMD is only traded on standard trading days, while Bitcoin is available for purchase around the clock.

Pulling Bitcoin and AMD from the CSV

5- Number Summaries

There are 1826 observations of bitcoin, compared to 1257 observations of AMD, because cryptocurrencies are traded 24/7. The adjusted price of Bitcoin ranged from 610 t0 63,503. The mean is higher than the median, showing that the data is positively skewed.

The adjusted AMD price ranged from 6.30 to 118.77, and the distribution is also positively skewed. The volume of AMD is considerably smaller, which is to be expected.

```
##       date               symbol              volume              adjusted
##  Min.   :2016-10-01   Length:1826        Min.   :3.904e+07   Min.   :  610.2
##  1st Qu.:2017-12-31   Class :character   1st Qu.:3.782e+09   1st Qu.: 4110.7
##  Median :2019-04-01   Mode  :character   Median :1.343e+10   Median : 8040.3
##  Mean   :2019-04-01                      Mean   :1.911e+10   Mean   :12728.8
##  3rd Qu.:2020-06-30                      3rd Qu.:2.942e+10   3rd Qu.:11346.4
##  Max.   :2021-09-30                      Max.   :3.510e+11   Max.   :63503.5
##                                          NA's   :4           NA's   :4
```

```
##       date               symbol              volume              adjusted
##  Min.   :2016-10-03   Length:1257        Min.   : 11035800   Min.   :  6.30
##  1st Qu.:2018-01-02   Class :character   1st Qu.: 41925300   1st Qu.: 13.12
##  Median :2019-04-03   Mode  :character   Median : 54413200   Median : 27.79
##  Mean   :2019-04-02                      Mean   : 65451146   Mean   : 38.90
##  3rd Qu.:2020-07-01                      3rd Qu.: 78366500   3rd Qu.: 56.18
##  Max.   :2021-09-29                      Max.   :325058400   Max.   :118.77
```

Distribution Historgrams

```
ggplot(data=BTC,aes(x=adjusted)) + geom_histogram(stat_bin=100) + ylab('Frequency')+ xlab('Pri
```

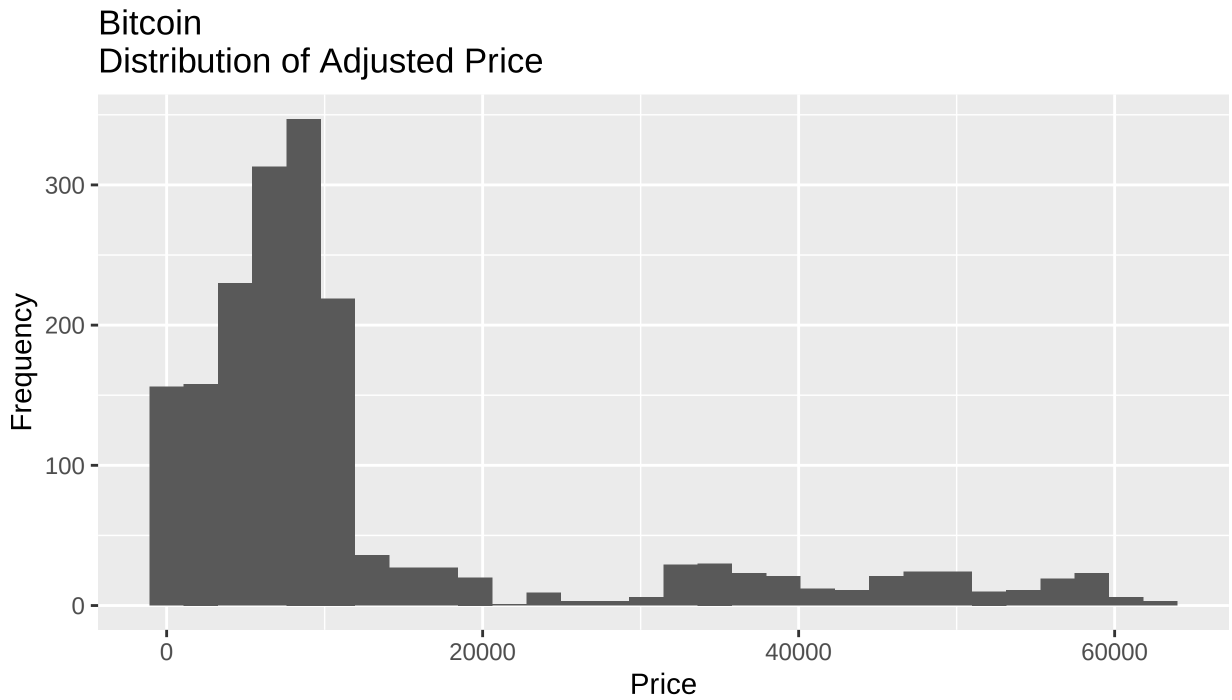


Figure 1

```
ggplot(data=AMD,aes(x=adjusted)) + geom_histogram(stat_bin=100) + ylab('Frequency')+  xlab('Pr
```

As expected given the 5-number summary, the distribution of the Bitcoin prices is skewed heavily to the right.

AMD shares a similar distribution to Bitcoin, although there are more high value than low value days for the period.

Here are the distributions after taking a log.

```
ggplot(data=BTC,aes(x=log(adjusted))) + geom_histogram() + ggtitle('Bitcoin\nDistribution of Ad
```

```
ggplot(data=AMD,aes(x=log(adjusted))) + geom_histogram() + ggtitle('AMD\nDistribution of Adjus
```

There is a high degree of kurtosis, but the distributions look more normal relative to the prior graphs.

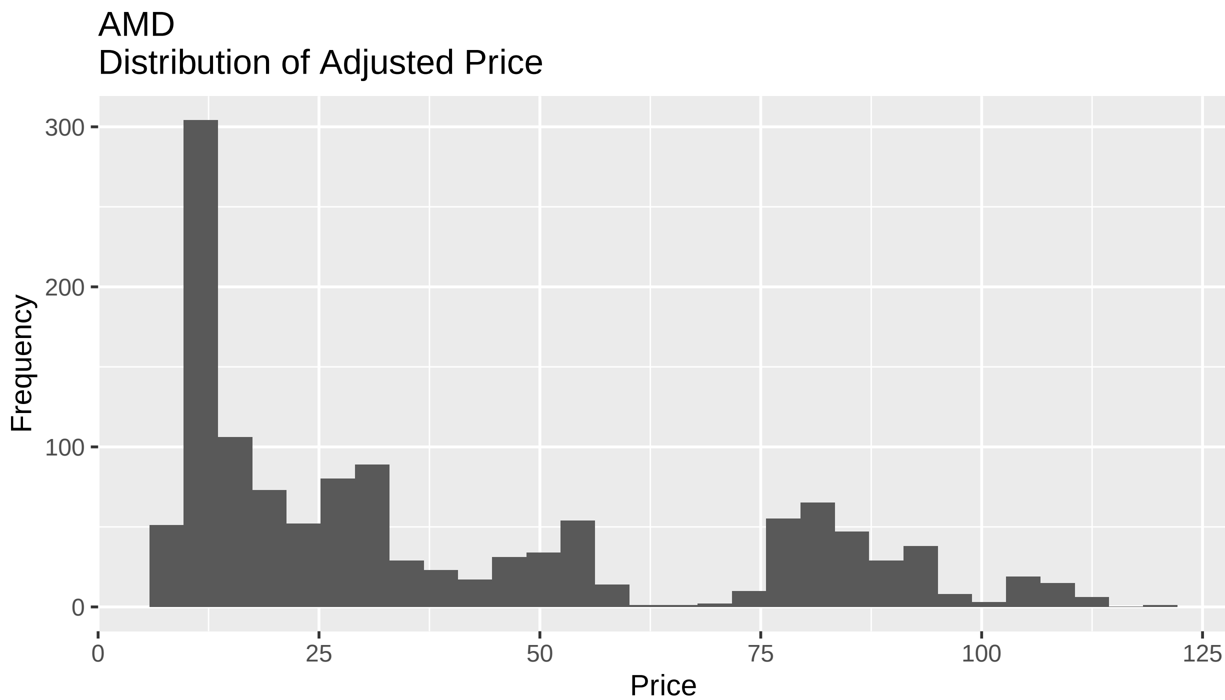


Figure 2

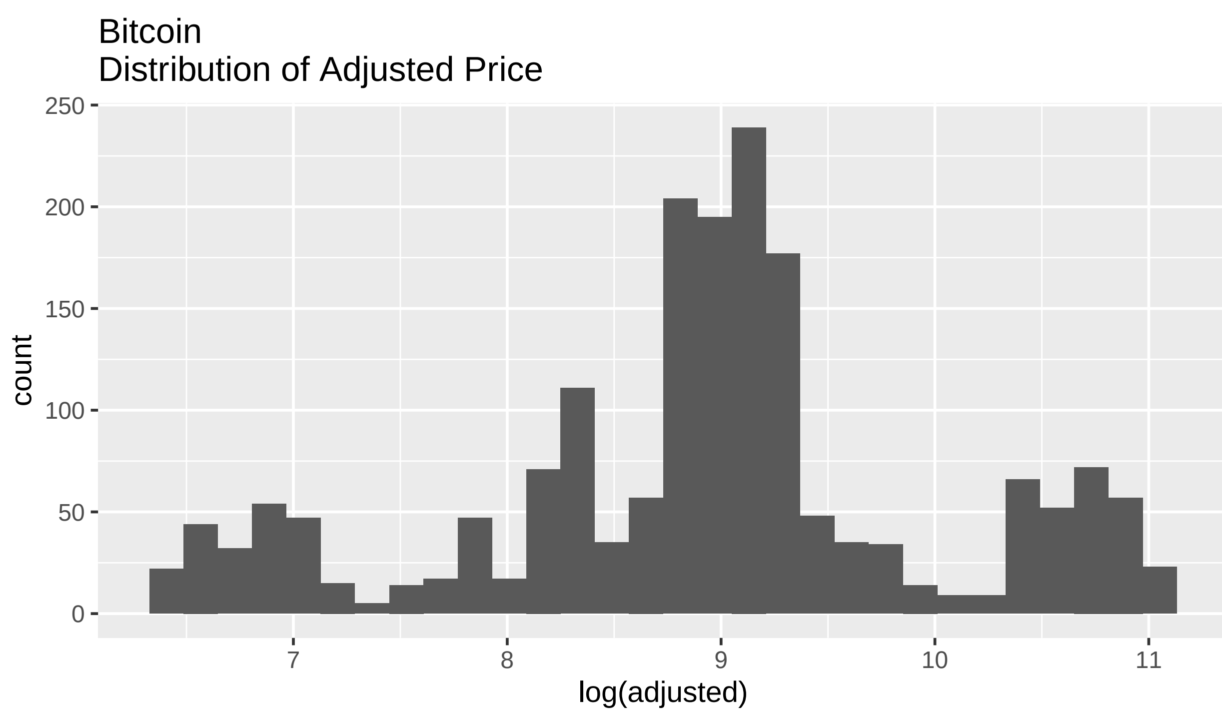


Figure 3

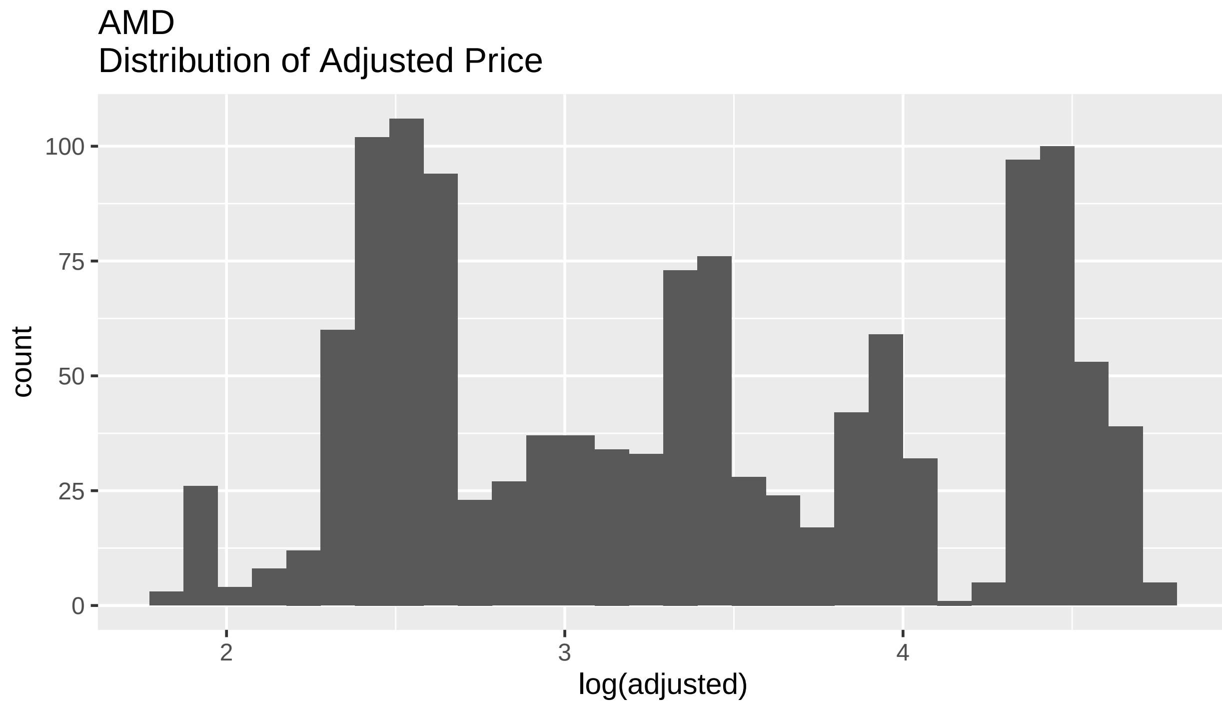
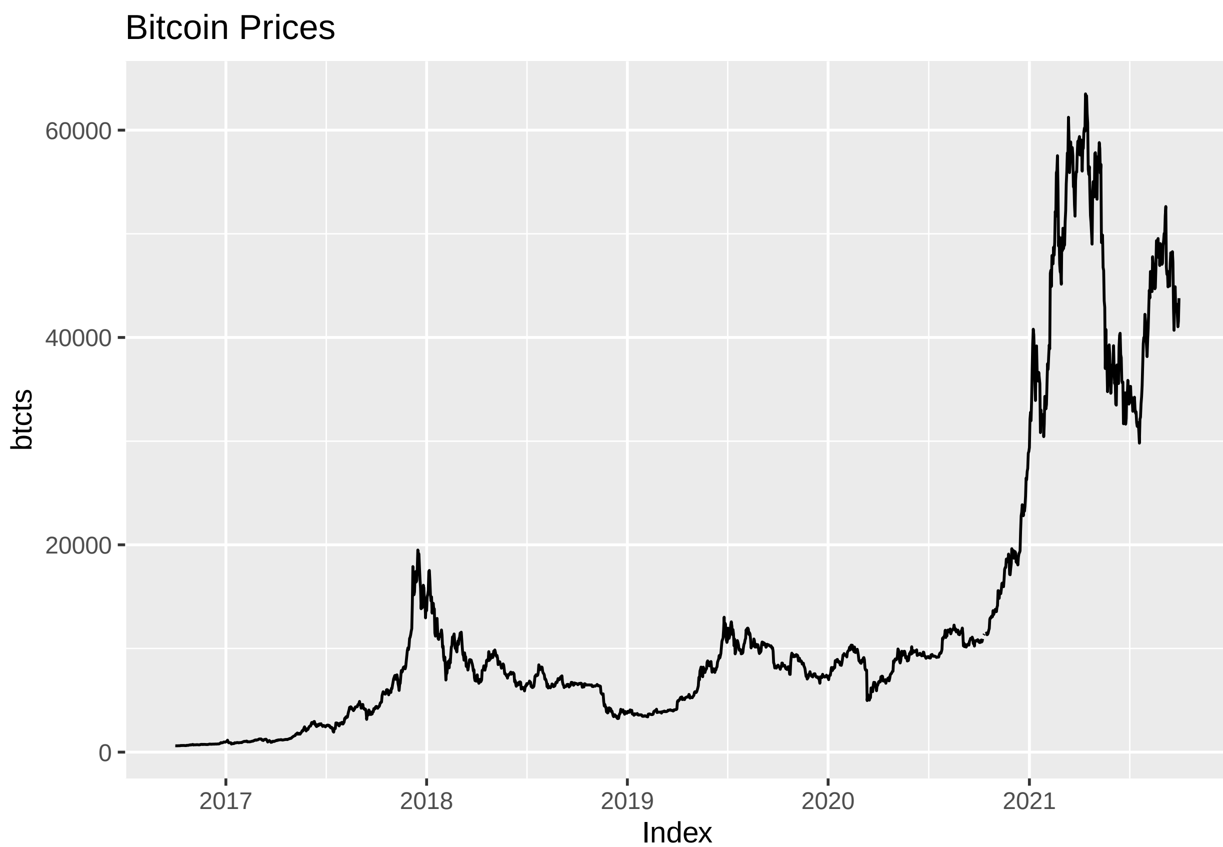


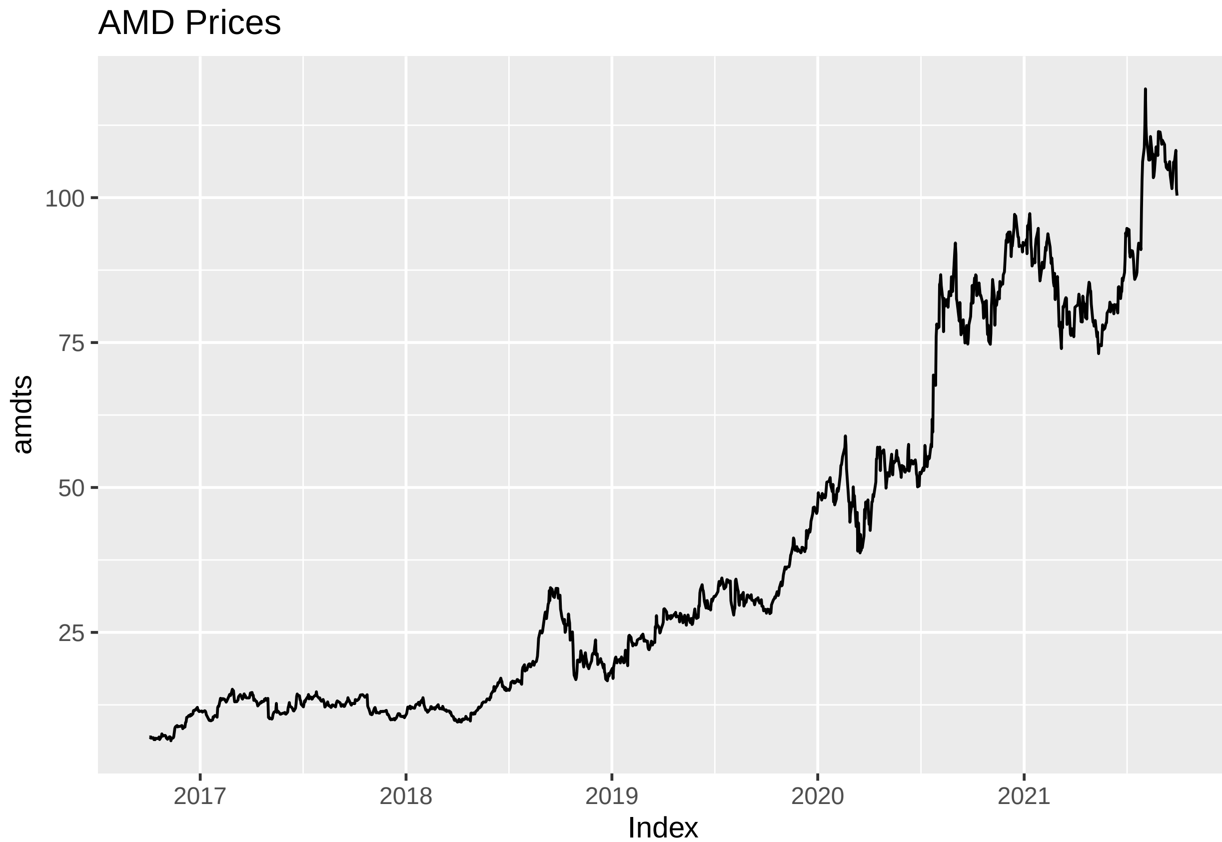
Figure 4

Converting to Zoo Objects

Graphing the Series

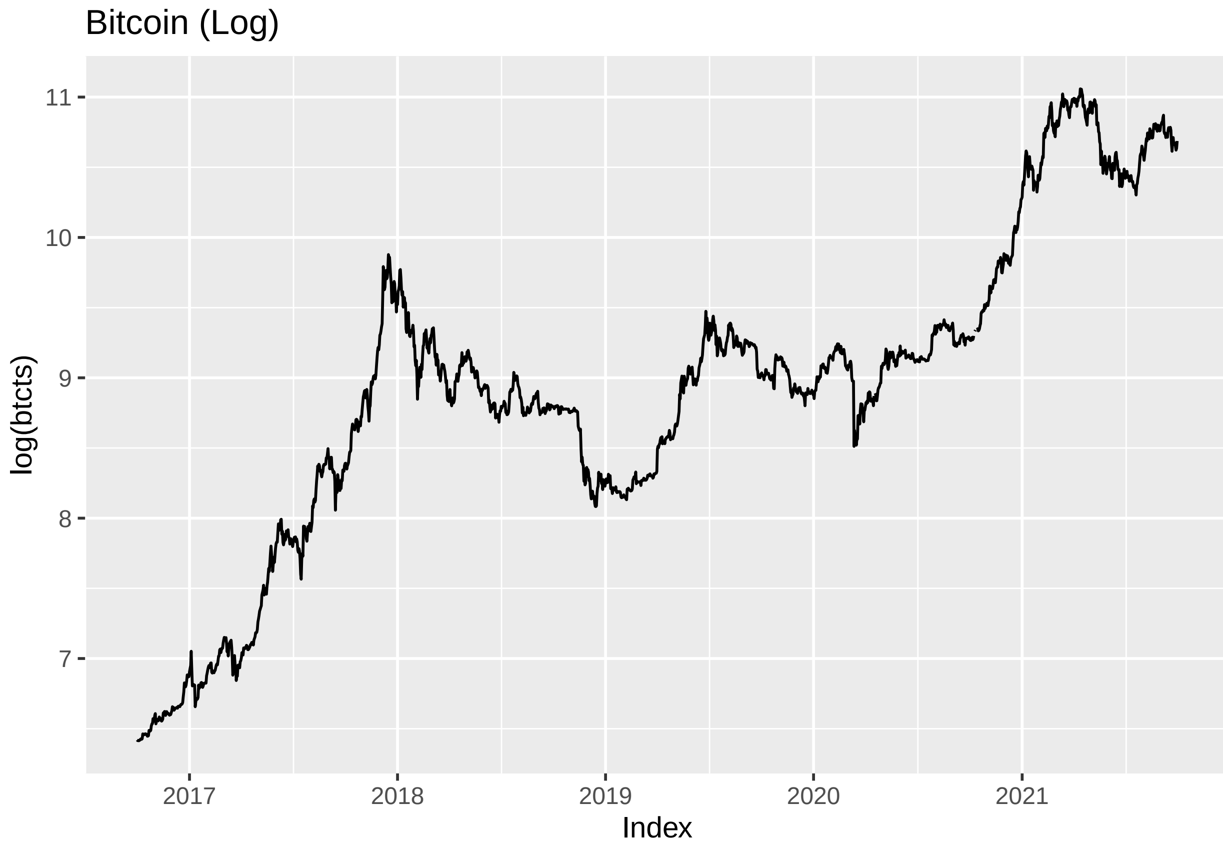
Looking at both series side by side, they appear to have very similar growth and behavior over time. Both of the series are multiplicative in nature and will benefit from a log transformation, as well as from differencing when we apply an ARIMA model.

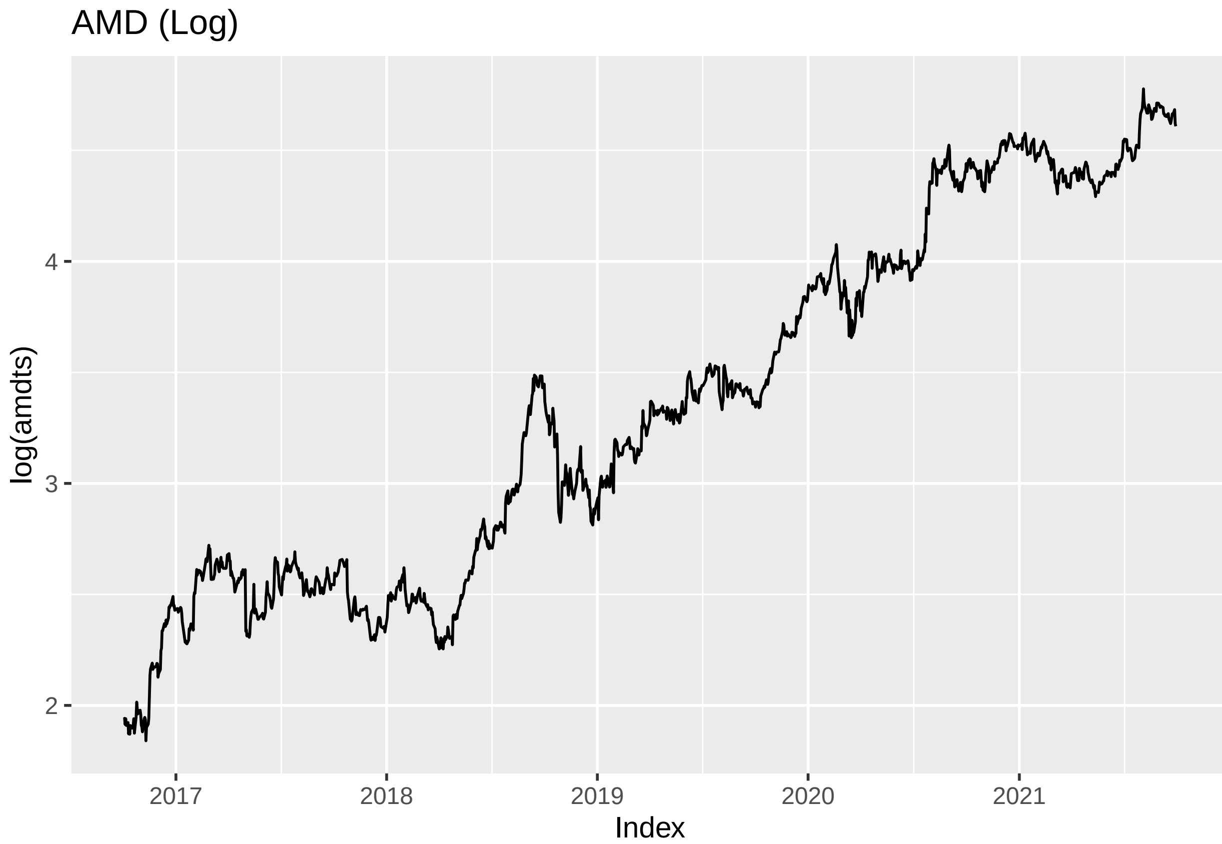

Bitcoin Prices
60000
40000
20000
0
btcts
2017
2018
2019
2020
2021
Index

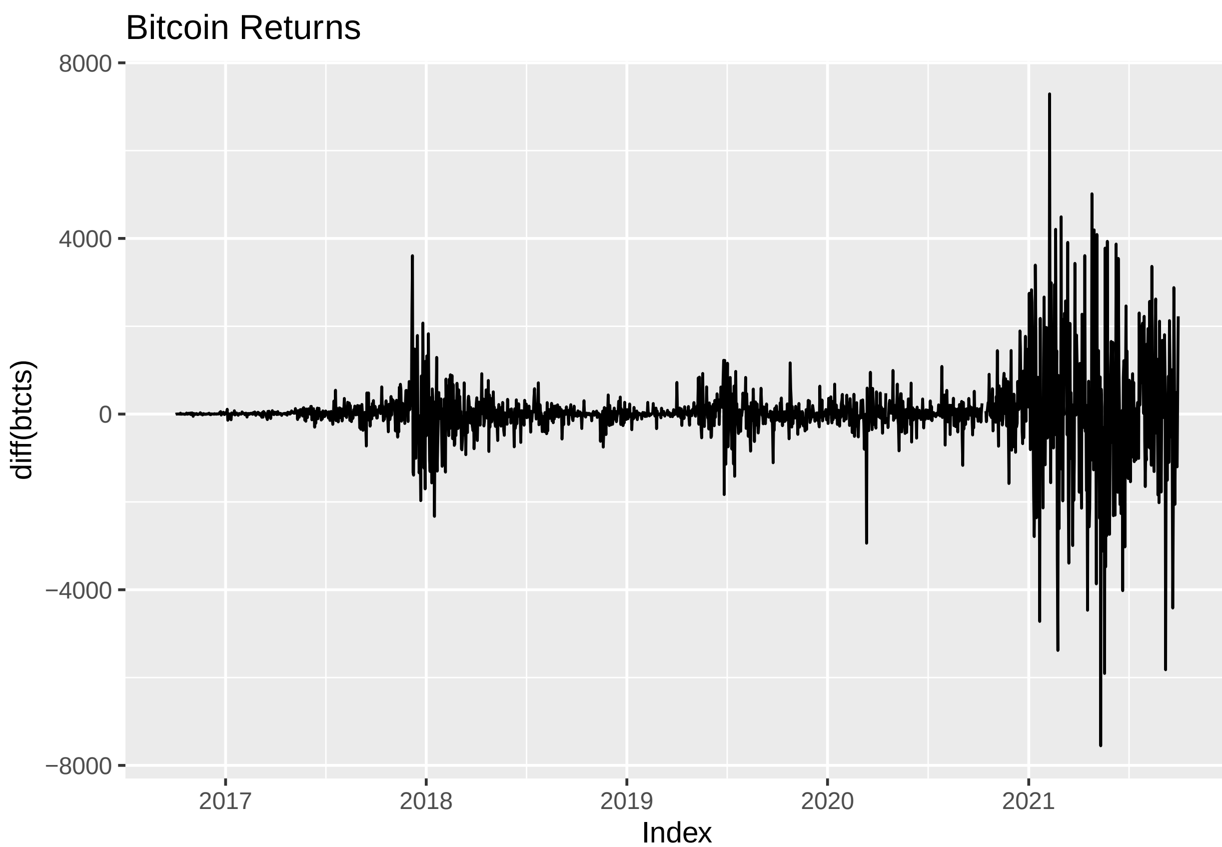


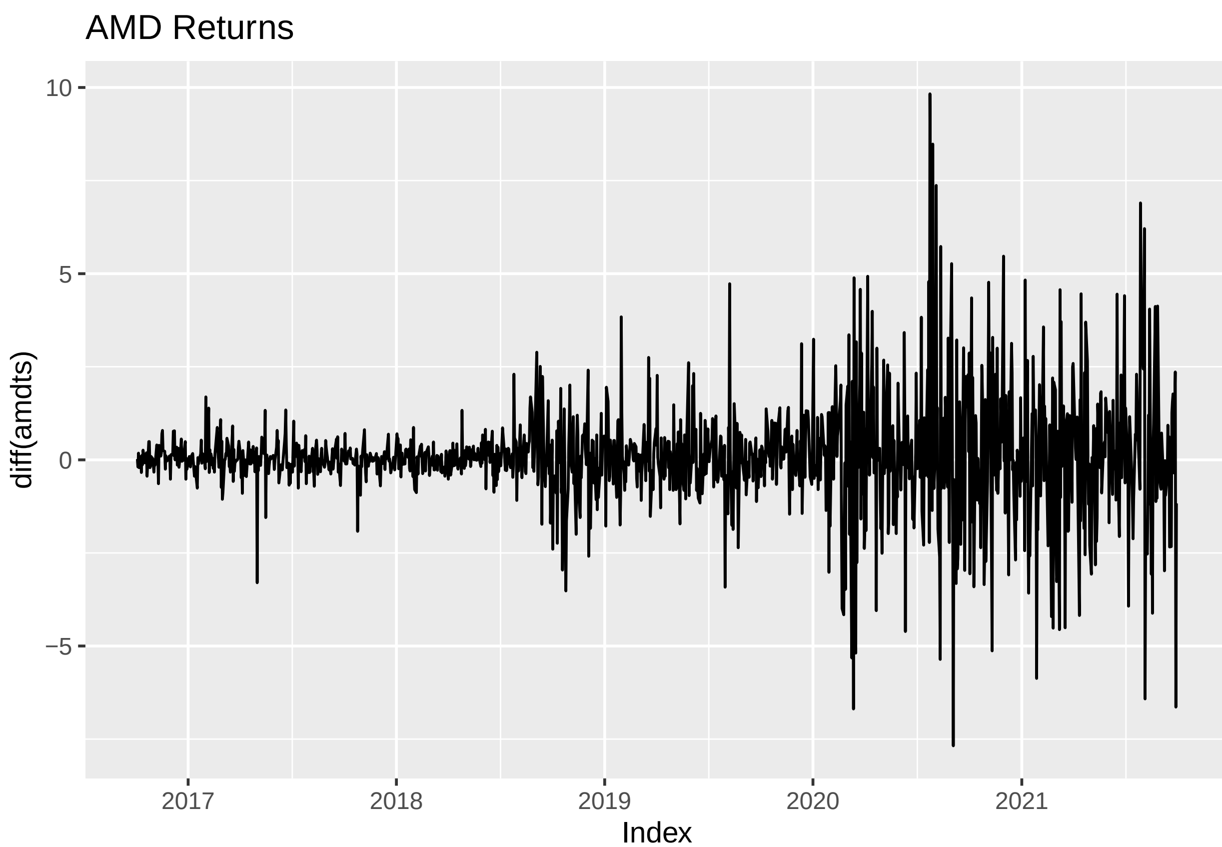
Graphing Transformations on the Series

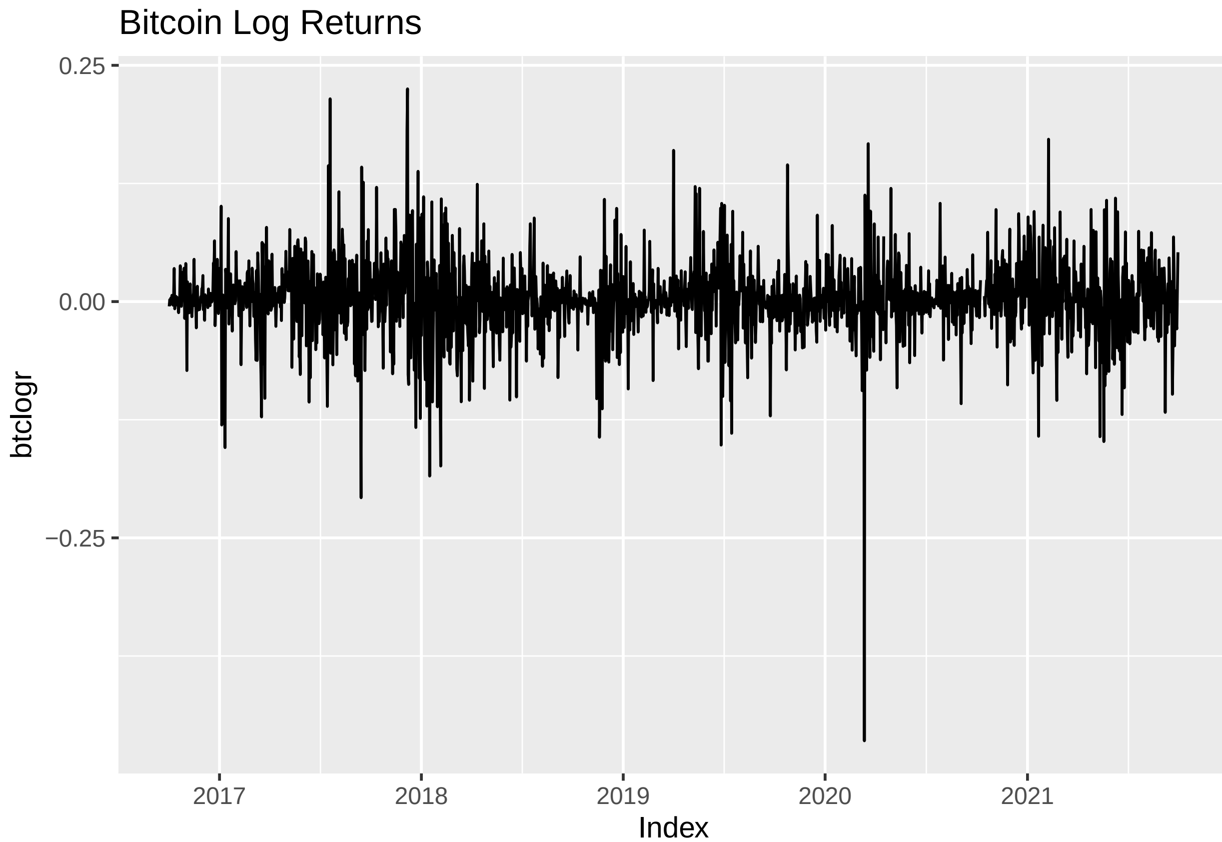
I experimented with a variety of transformations on the data set.By far, taking the log return of the series does the most to create a uniform distributions accross time, making the series look similar to white noise.

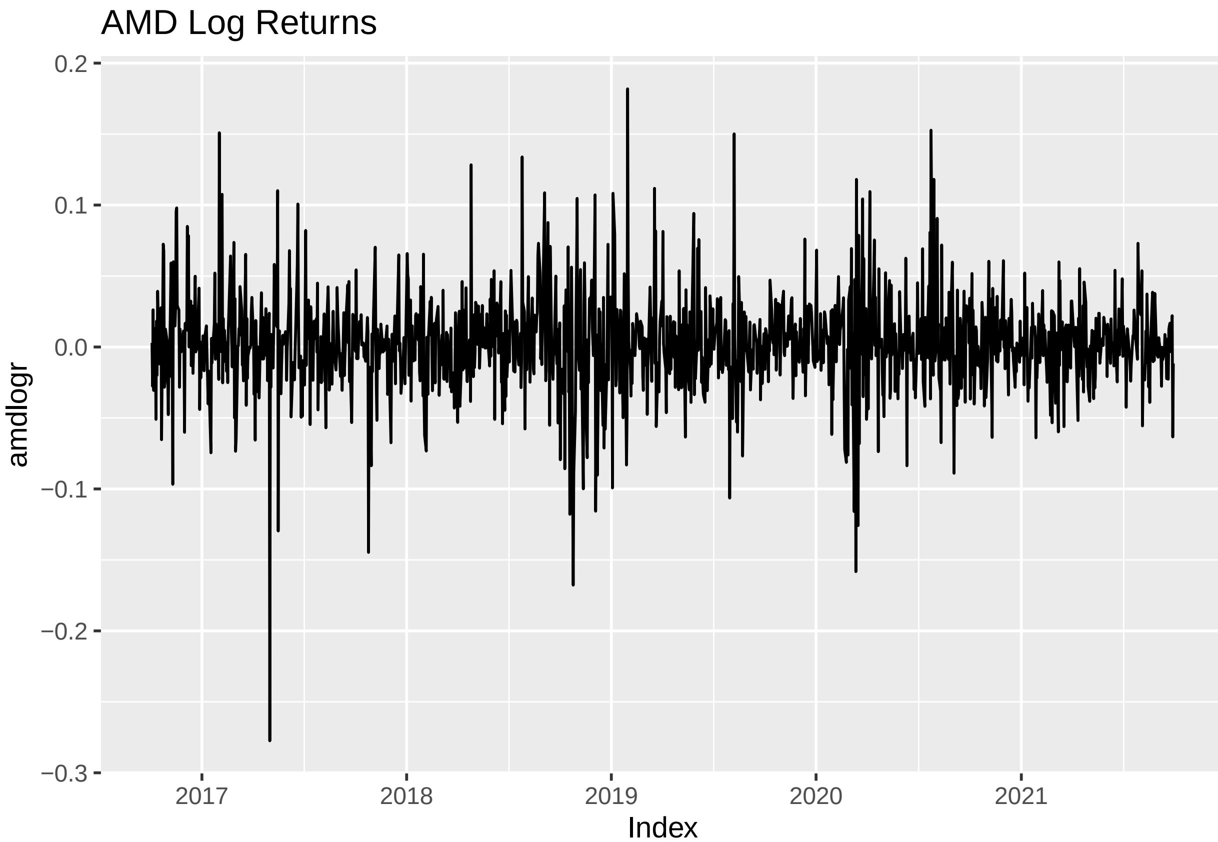

Bitcoin (Log)
11
10
9
8
7
log(btcts)
2017
2018
2019
2020
2021
Index


AMD (Log)
log(amdts)
2
3
4
2017
2018
2019
2020
2021
Index


Bitcoin Returns
8000
4000
0
−4000
−8000
diff(btcts)
2017
2018
2019
2020
2021
Index


AMD Returns
10
5
0
−5
diff(amdts)
2017
2018
2019
2020
2021
Index


Bitcoin Log Returns
0.25
0.00
−0.25
btclogr
2017
2018
2019
2020
2021
Index



ACFs of the Log Returns

It appears that an AR model could potentially fit the AMD time series well. I am unsure about the Bitcoin series at this time.

```
#acf of the original series

#Bitcoin
acf(btclogr, na.action = na.pass)
```

```
#AMD
acf(amdlogr, na.action= na.pass)
```

PACFs of the Log Returns

```
#acf of the original series

#Bitcoin
pacf(btclogr, na.action = na.pass)
```

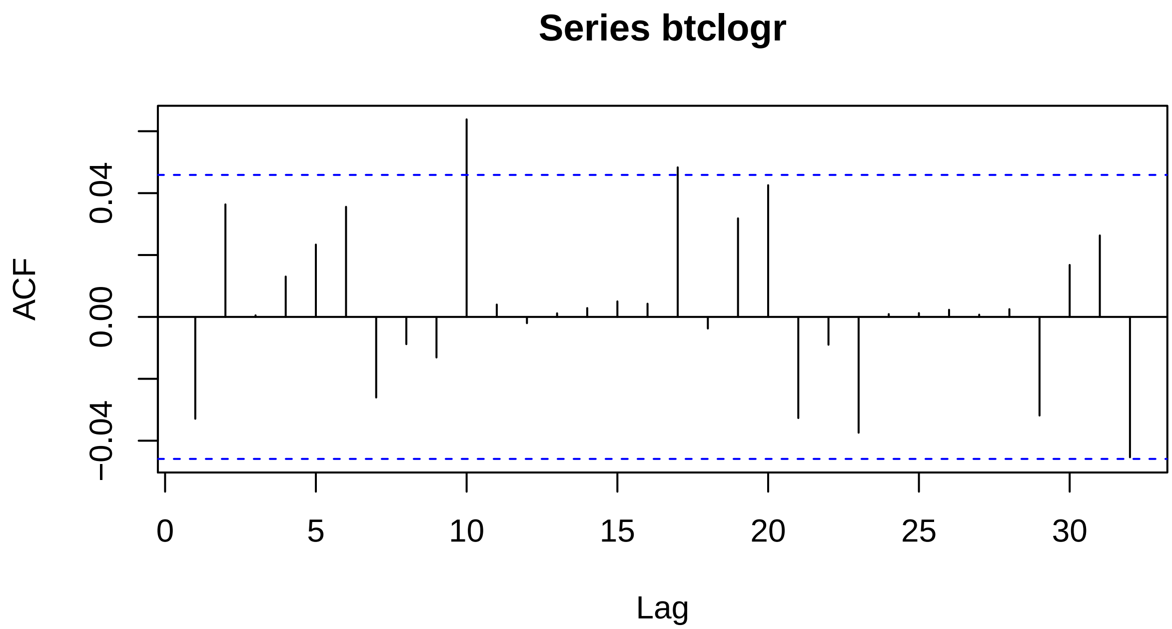


Figure 5

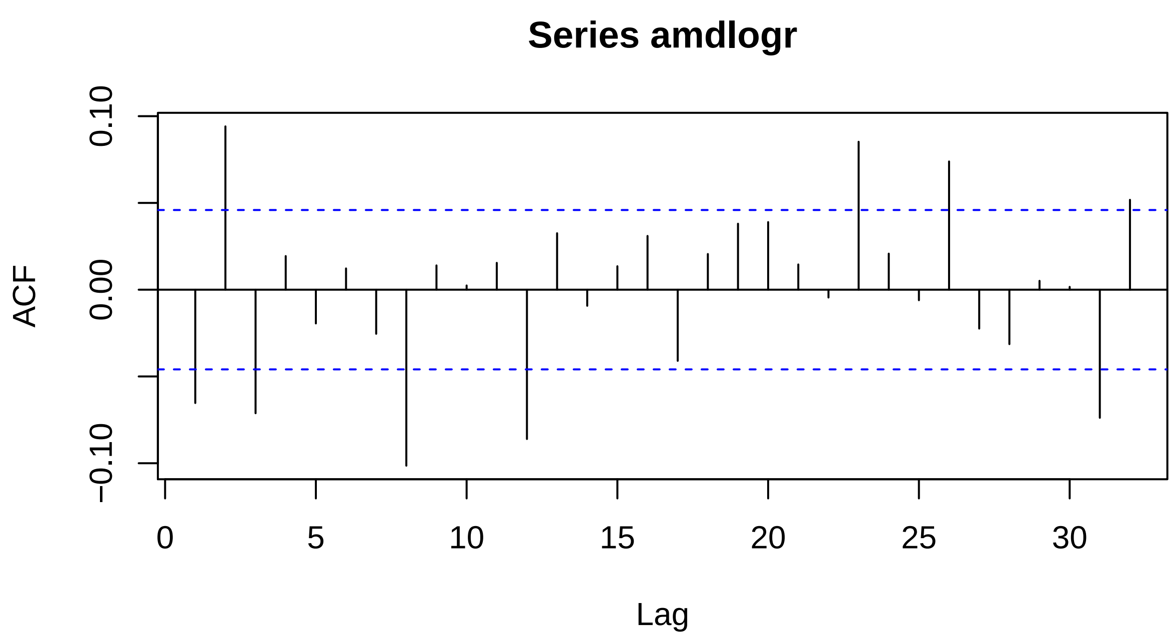


Figure 6

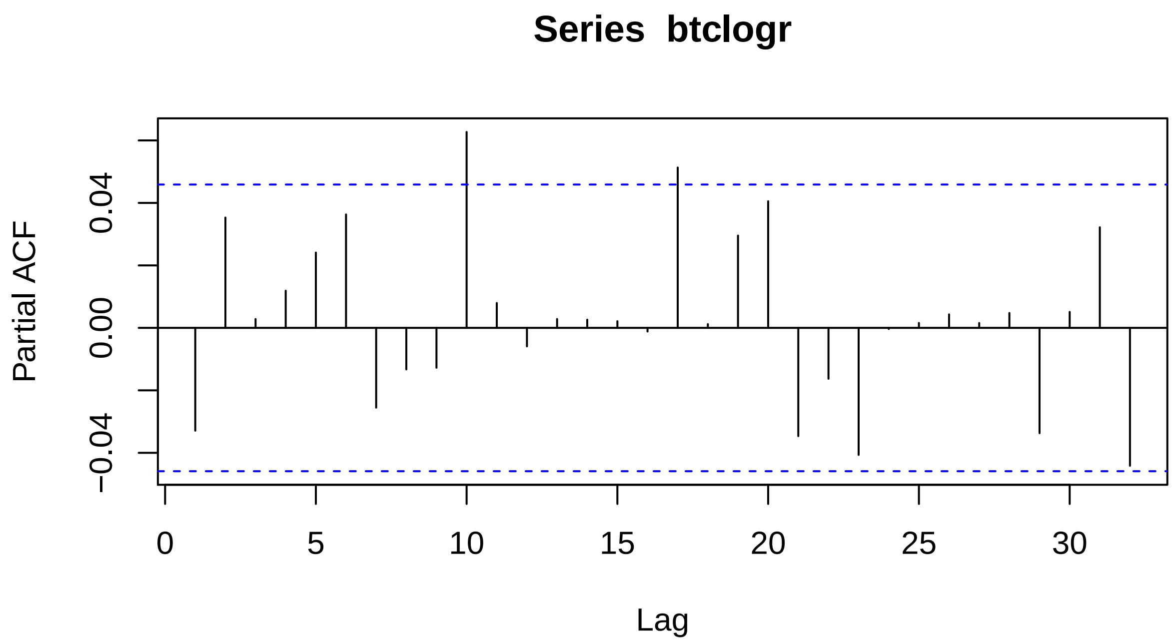


Figure 7

```
#AMD
pacf(amdlogr, na.action= na.pass)
```

Ljung Box Test

The P values for the log transormed series are very low, rejecting the null hypothesis that the data is stationary. The P values are much higher when looking at the log returns, failing to reject the null hypotheisis at the 99% confidence interval for both series.

```
# log transformed
Box.test(log(btcts), lag = 1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  log(btcts)
## X-squared = 1819.9, df = 1, p-value < 2.2e-16
```

```
Box.test(log(amdts),lag=1, type='Ljung-Box')
```

```
##
##  Box-Ljung test
```

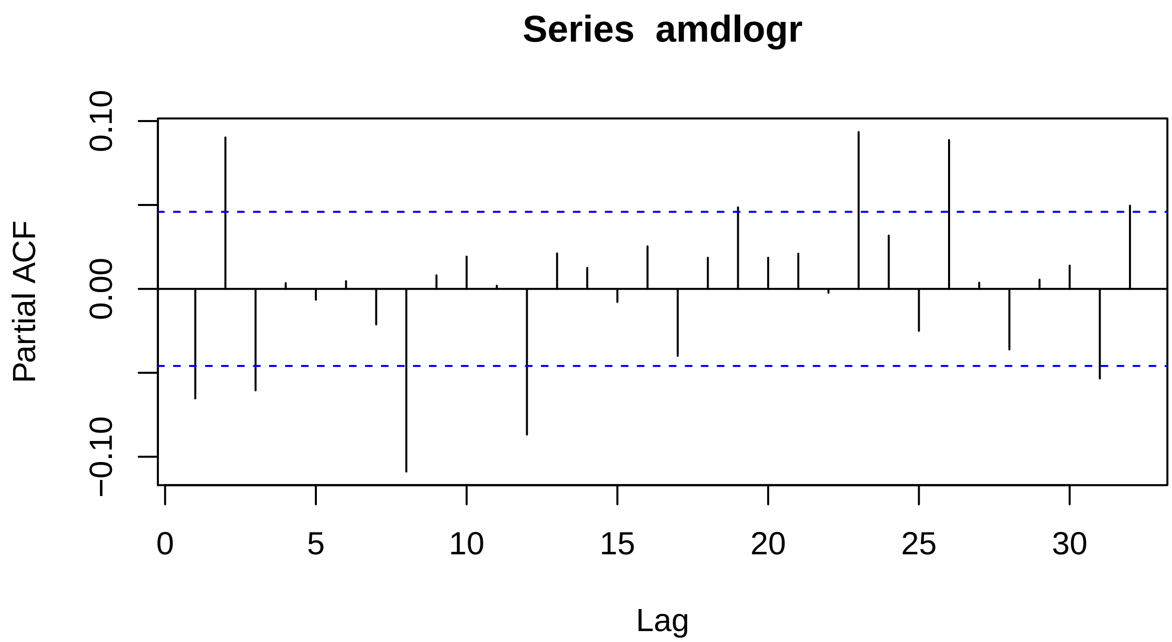


Figure 8

```
##
## data:  log(amdts)
## X-squared = 1260, df = 1, p-value < 2.2e-16
```

```
# log returns
Box.test(btclogr, lag = 1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  btclogr
## X-squared = 1.9714, df = 1, p-value = 0.1603
```

```
Box.test(amdlogr,lag=1, type='Ljung-Box')
```

```
##
##  Box-Ljung test
##
## data:  amdlogr
## X-squared = 5.369, df = 1, p-value = 0.0205
```

Preliminary Arima Model

```r
# Auto Arima on BTC
fitbtc=auto.arima(log(btcts))
fitbtc
```

```
## Series: log(btcts)
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       0.0023
## s.e.  0.0010
##
## sigma^2 estimated as 0.001752:  log likelihood=3194.47
## AIC=-6384.93   AICc=-6384.93   BIC=-6373.91
```

```r
coeftest(fitbtc)
```

```
##
## z test of coefficients:
##
##         Estimate Std. Error z value Pr(>|z|)
## drift 0.00231702 0.00097972   2.365  0.01803 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Auto Arima on AMD
fitamd=auto.arima(log(amdts))
fitamd
```

```
## Series: log(amdts)
## ARIMA(3,1,0)
##
## Coefficients:
##           ar1    ar2      ar3
##       -0.1379  0.059  -0.0216
## s.e.   0.0296  0.037   0.0409
##
## sigma^2 estimated as 0.001087:  log likelihood=2367.94
## AIC=-4727.87   AICc=-4727.85   BIC=-4705.84
```

```r
coeftest(fitamd)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.137856   0.029599 -4.6574 3.202e-06 ***
## ar2  0.058977   0.037038  1.5923    0.1113
## ar3 -0.021555   0.040894 -0.5271    0.5981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```