DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite

Appendix – Individual Report- Evelina Ramoskaite

Contributions to the Team's Efforts

Meetings

I attended group meetings once a week to gauge progress and keep the project on track. These were more frequent in the last two weeks of the project.

Presentation

I set up the structure and was responsible for the style of the presentation. I contributed content to the introduction, exploratory, and AMD/NVDA slides. I made sure the presentation was aesthetically pleasing, fun, and organized.

Report

The exploratory section was written by me, as well as the Bitcoin series analysis. The graphs for these two sections were from my teammate, Mourad.

The AMD/NVDIA analysis was written by Gerardo and me together.

I also made a VAR model and wrote about it as a response to the punchlist.

Takeaways from the Project

This was the first project I had where the models showed little predictive value at the end of our work. That was a new experience and I had to explain a lot of the logic behind why we did not reach the conclusions we intended to. I felt that the project reinforced what we learned in class—that markets are generally efficient. I do believe that there may be more interesting behavior at higher frequencies or on a shorter time frame.

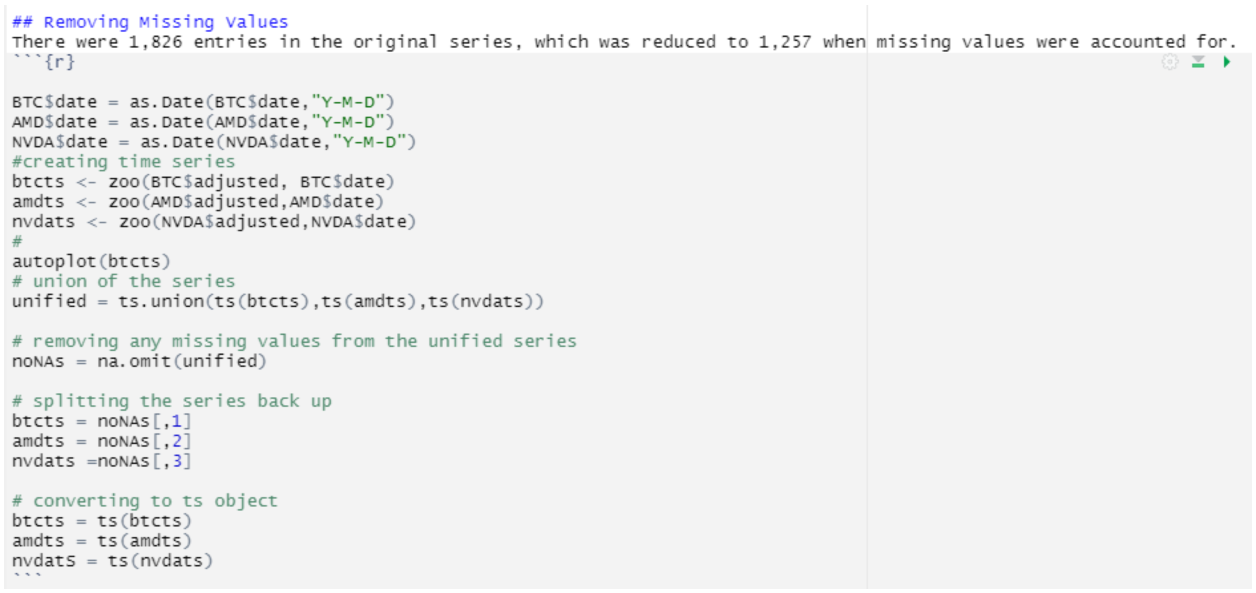
Working on the project gave me valuable practice applying all of the techniques we learned in class. I feel better about graphing, unit root testing, and validation. I feel more comfortable with graphing and indexing time series objects as well.

I learned a lot from this project through all the trials and mistakes I made along the way. In the VAR part specifically, I ran into a lot of issues with the data structure compatibility for that package and spent a long time making the forecasts work. I am much more comfortable with R and appreciate the practice because that will be the main language I will need for my new job. Trading cryptos and stocks is a hobby of mine, so I plan to continue to use these techniques in combination with other models, like social media sentiment analysis, for my personal use.

DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite

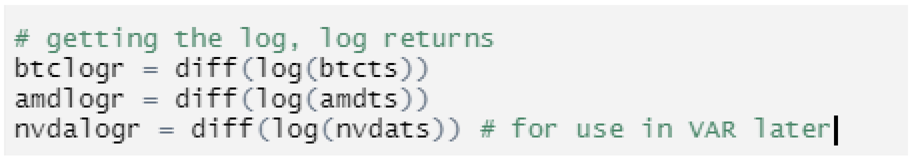
Pre-Processing Work

To remove missing values, I merged all the time series together, deleted the missing rows, and split them back up for individual use.



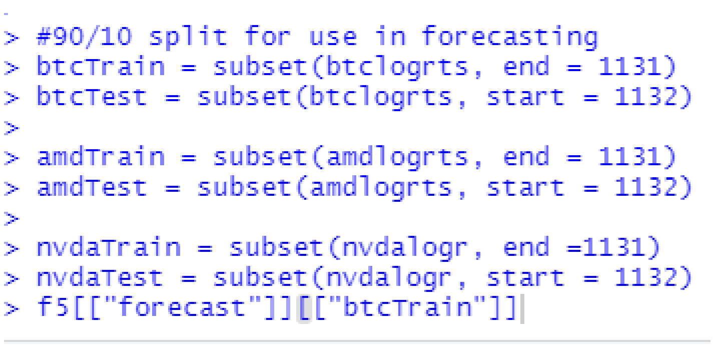
~~~
## Removing Missing Values
There were 1,826 entries in the original series, which was reduced to 1,257 when missing values were accounted for.
```{r}

BTC$date = as.Date(BTC$date,"Y-M-D")
AMD$date = as.Date(AMD$date,"Y-M-D")
NVDA$date = as.Date(NVDA$date,"Y-M-D")
#creating time series
btcts <- zoo(BTC$adjusted, BTC$date)
amdts <- zoo(AMD$adjusted,AMD$date)
nvdats <- zoo(NVDA$adjusted,NVDA$date)
#
autoplot(btcts)
# union of the series
unified = ts.union(ts(btcts),ts(amdts),ts(nvdats))

# removing any missing values from the unified series
noNAs = na.omit(unified)

# splitting the series back up
btcts = noNAs[,1]
amdts = noNAs[,2]
nvdats =noNAs[,3]

# converting to ts object
btcts = ts(btcts)
amdts = ts(amdts)
nvdats = ts(nvdats)
```
~~~

After the mismatched values were addressed, I got the log returns for every series.



```
# getting the log, log returns
btclogr = diff(log(btcts))
amdlogr = diff(log(amdts))
nvdalogr = diff(log(nvdats)) # for use in VAR later
```

Then, I created a train/test split with a 10% hold out set for forecast validation.

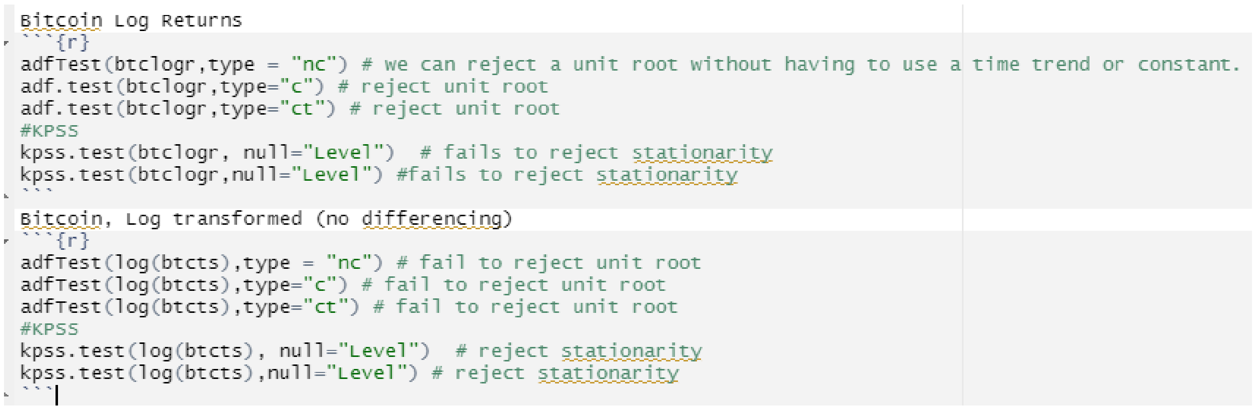


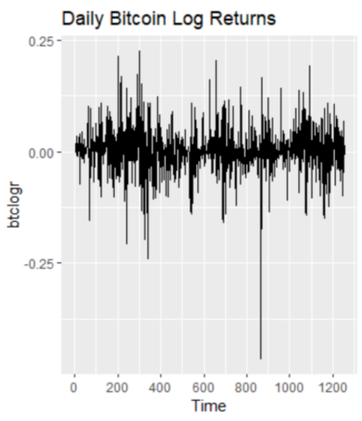
```
> #90/10 split for use in forecasting
> btcTrain = subset(btclogrts, end = 1131)
> btcTest = subset(btclogrts, start = 1132)
>
> amdTrain = subset(amdlogrts, end = 1131)
> amdTest = subset(amdlogrts, start = 1132)
>
> nvdaTrain = subset(nvdalogr, end =1131)
> nvdaTest = subset(nvdalogr, start = 1132)
> f5[["forecast"]][["btcTrain"]]
```

DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite

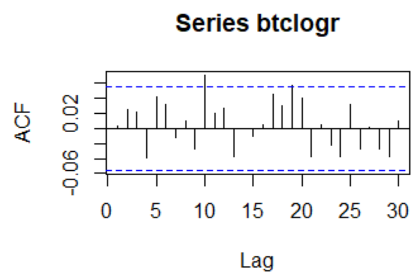
Analysis – Bitcoin

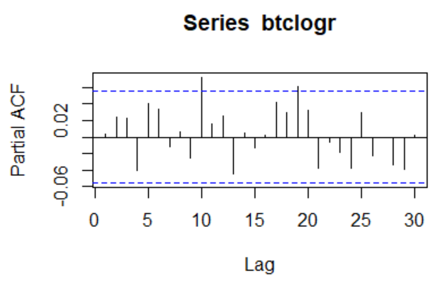
I began with unit root testing to confirm stationarity.



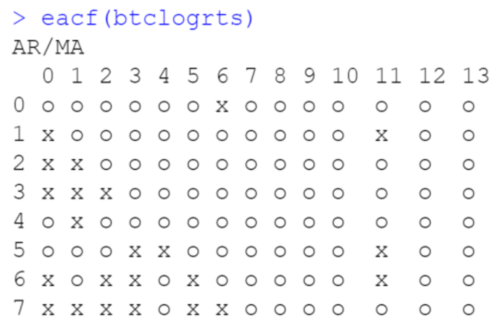
~~~
Bitcoin Log Returns
```{r}
adfTest(btclogr,type = "nc") # we can reject a unit root without having to use a time trend or constant.
adf.test(btclogr,type="c") # reject unit root
adf.test(btclogr,type="ct") # reject unit root
#KPSS
kpss.test(btclogr, null="Level")  # fails to reject stationarity
kpss.test(btclogr,null="Level") #fails to reject stationarity
~~~
Bitcoin, Log transformed (no differencing)
```{r}
adfTest(log(btcts),type = "nc") # fail to reject unit root
adfTest(log(btcts),type="c") # fail to reject unit root
adfTest(log(btcts),type="ct") # fail to reject unit root
#KPSS
kpss.test(log(btcts), null="Level")  # reject stationarity
kpss.test(log(btcts),null="Level") # reject stationarity
```
```

With the Dickey fuller tests on both the log transformed and log return series, I ruled out the possibility of a time regression being appropriate. The log return series was stationary/had no unit root, even without a constant or time trend. The kpss test failed to reject stationarity, agreeing with the dickey fuller tests. The log return time plot did not appear to have a time trend and looked approximately stationary. The log transformed series still had a unit root in every version of the dickey fuller test, ruling out those possibilities. So, I continued the analysis with the log returns.



I then proceeded to select an order for the ARIMA model using the ACF, PACF, and EACF.
```

DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite



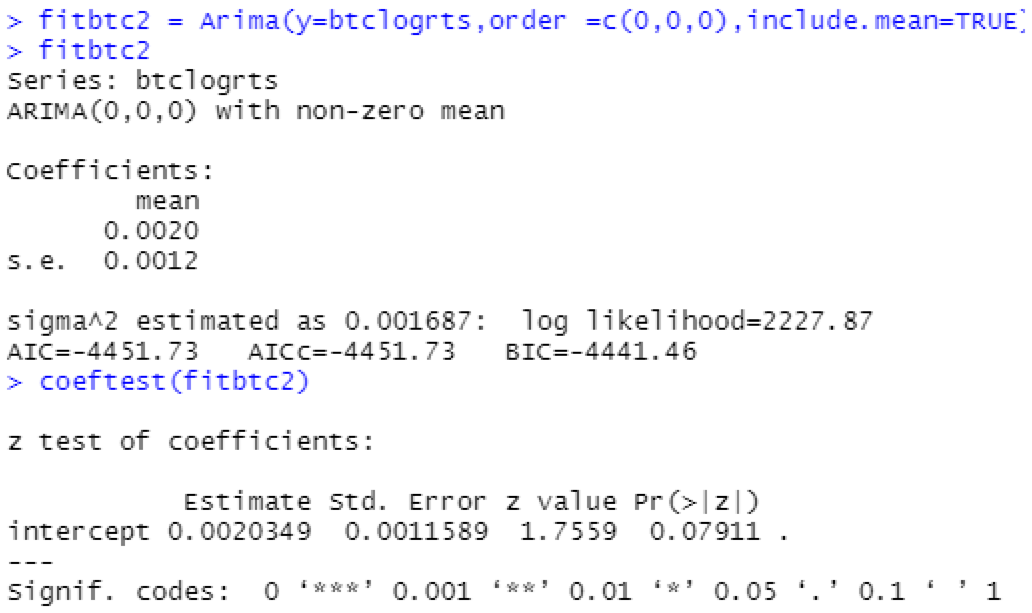


The ACF and PACF lacked a clear pattern, so I proceeded with the EACF. The EACF shows a clear triangle of 0s at the ARMA(0,0) point. I did attempt an AR(1) and MA(1) model just for comparison reasons, but the ARMA(0,0,0) performed the best. I also attempted to create a model with drift, but the drift term was insignificant. In other words, the best forecast that we can get with ARIMA is the mean of the series.



```
> eacf(btclogrts)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 o o o o o o x o o o o  o  o  o
1 x o o o o o o o o o o  x  o  o
2 x x o o o o o o o o o  o  o  o
3 x x x o o o o o o o o  o  o  o
4 o x o o o o o o o o o  o  o  o
5 o o o x x o o o o o o  x  o  o
6 x o x x o x o o o o o  x  o  o
7 x x x x o x x o o o o  o  o  o
```

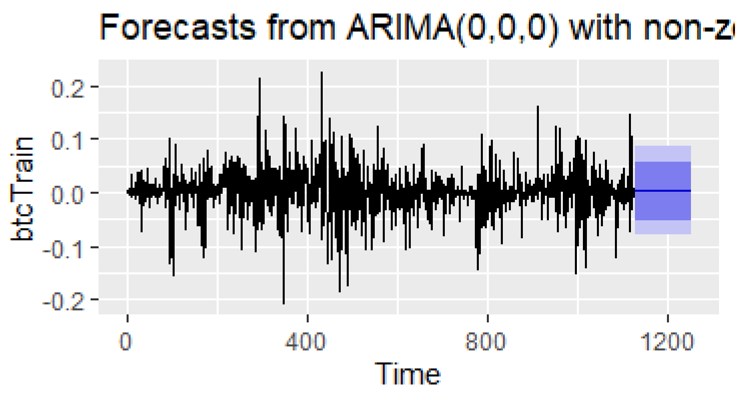
Here are the coefficients for the best model:



```
> fitbtc2 = Arima(y=btclogrts,order =c(0,0,0),include.mean=TRUE)
> fitbtc2
Series: btclogrts
ARIMA(0,0,0) with non-zero mean

Coefficients:
        mean
      0.0020
s.e.  0.0012

sigma^2 estimated as 0.001687:  log likelihood=2227.87
AIC=-4451.73   AICc=-4451.73   BIC=-4441.46
> coeftest(fitbtc2)

z test of coefficients:

           Estimate Std. Error z value Pr(>|z|)
intercept 0.0020349  0.0011589  1.7559  0.07911 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite

The forecast met my expectations. It is just a flat line at the mean of the series with a wide confidence interval.

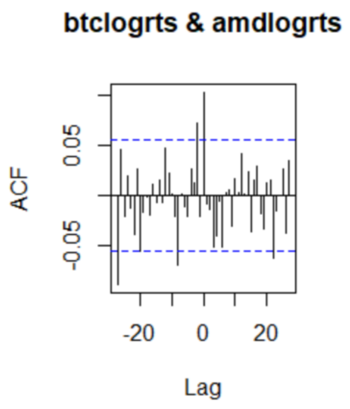


Analysis – Bitcoin and AMD Cross Correlated Model

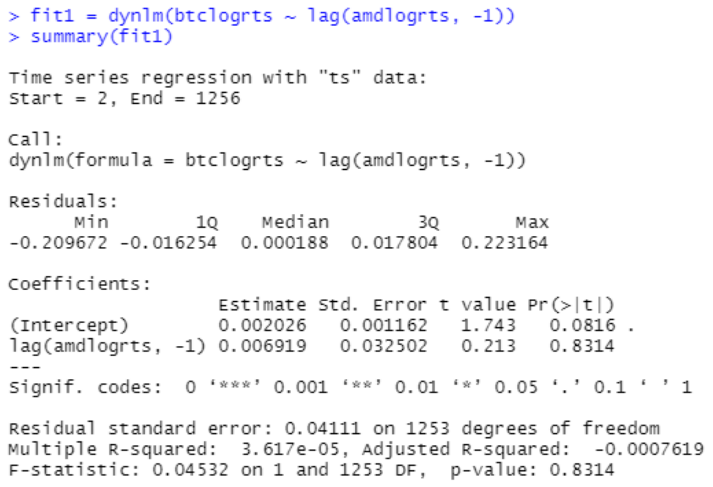
Next, I focused on exploring Bitcoin's potential relationship to a graphics card manufacturing company, AMD. AMD makes GPUs that are commonly used for cryptocurrency mining. It is speculated that bitcoin values have increased sales for graphics card producers, as more people want to get involved in mining. The two series follow a similar general trend.

There were missing values in the dataset, which we did not address in the prior exploratory analysis. We decided as a group to simply remove cryptocurrency values outside of regular trading hours. To make the bitcoin series compatible with AMD, I had to merge the two series and remove any rows with missing values, then split them back up into individual time series objects.

After handling the missing values, I explored the cross correlation between AMD and bitcoin. I started by looking at the acf plot comparing the two series.



DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite
There was a small degree of cross correlation at lag 1 in the log returns. I went ahead and modeled this relationship.



```
> fit1 = dynlm(btclogrts ~ lag(amdlogrts, -1))
> summary(fit1)

Time series regression with "ts" data:
Start = 2, End = 1256

Call:
dynlm(formula = btclogrts ~ lag(amdlogrts, -1))

Residuals:
      Min        1Q    Median        3Q       Max 
-0.209672 -0.016254  0.000188  0.017804  0.223164 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)  
(Intercept)        0.002026   0.001162   1.743   0.0816 .
lag(amdlogrts, -1) 0.006919   0.032502   0.213   0.8314  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04111 on 1253 degrees of freedom
Multiple R-squared:  3.617e-05,	Adjusted R-squared:  -0.0007619 
F-statistic: 0.04532 on 1 and 1253 DF,  p-value: 0.8314
```

```
> Box.test(fit1$residuals, type="Ljung")

	Box-Ljung test

data:  fit1$residuals
X-squared = 0.010776, df = 1, p-value = 0.9173
```

When I fit a regression model to predict Bitcoin log returns, the lag-1 AMD term was insignificant. The Ljung Box test failed to reject that there was no autocorrelation in the residuals, but the model does not have any predictive value over the base ARIMA(0,0,0). There was no meaningful relationship between the two series, so I moved on to other prospects.

Analysis – VAR Model with Bitcoin, AMD,NVDA

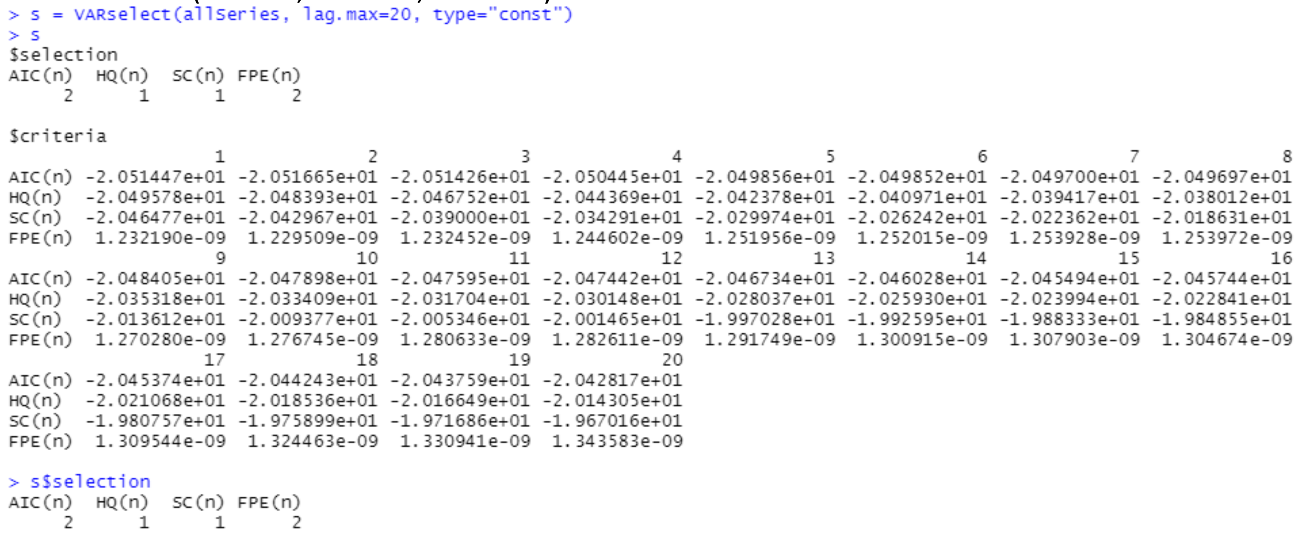
One of our punchlist items for the final week of class was to explore a vector autoregressive model for Bitcoin and the crypto mining stocks.

First, imported NVDA and removed the missing values. After this, I split every series into 90% training and 10% validation. I then used cbind() to merge all 3 time series.

```
allseries = cbind(btclogr,amdlogr,nvdalogr)

alltrain = cbind(btcTrain,amdTrain,nvdaTrain)
```

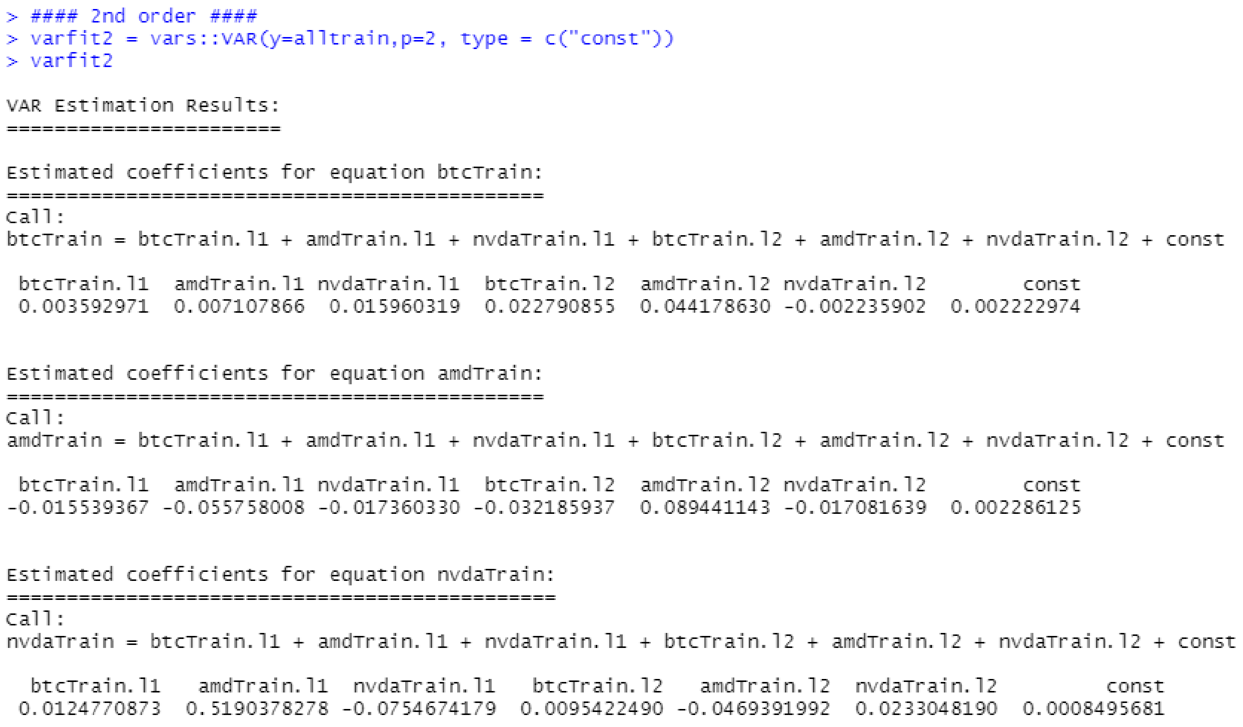
DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite

alltest = cbind(btcTest,amdTest,nvdaTest)



```
> s = VARselect(allSeries, lag.max=20, type="const")
> s
$selection
AIC(n)  HQ(n)  SC(n) FPE(n) 
     2      1      1      2 

$criteria
                   1             2             3             4             5             6             7             8
AIC(n) -2.051447e+01 -2.051665e+01 -2.051426e+01 -2.050445e+01 -2.049856e+01 -2.049852e+01 -2.049700e+01 -2.049697e+01
HQ(n)  -2.049578e+01 -2.048393e+01 -2.046752e+01 -2.044369e+01 -2.042378e+01 -2.040971e+01 -2.039417e+01 -2.038012e+01
SC(n)  -2.046477e+01 -2.042967e+01 -2.039000e+01 -2.034291e+01 -2.029974e+01 -2.026242e+01 -2.022362e+01 -2.018631e+01
FPE(n)  1.232190e-09  1.229509e-09  1.232452e-09  1.244602e-09  1.251956e-09  1.252015e-09  1.253928e-09  1.253972e-09
                   9            10            11            12            13            14            15            16
AIC(n) -2.048405e+01 -2.047898e+01 -2.047595e+01 -2.047442e+01 -2.046734e+01 -2.046028e+01 -2.045494e+01 -2.045744e+01
HQ(n)  -2.035318e+01 -2.033409e+01 -2.031704e+01 -2.030148e+01 -2.028037e+01 -2.025930e+01 -2.023994e+01 -2.022841e+01
SC(n)  -2.013612e+01 -2.009377e+01 -2.005346e+01 -2.001465e+01 -1.997028e+01 -1.992595e+01 -1.988333e+01 -1.984855e+01
FPE(n)  1.270280e-09  1.276745e-09  1.280633e-09  1.282611e-09  1.291749e-09  1.300915e-09  1.307903e-09  1.304674e-09
                  17            18            19            20
AIC(n) -2.045374e+01 -2.044243e+01 -2.043759e+01 -2.042817e+01
HQ(n)  -2.021068e+01 -2.018536e+01 -2.016649e+01 -2.014305e+01
SC(n)  -1.980757e+01 -1.975899e+01 -1.971686e+01 -1.967016e+01
FPE(n)  1.309544e-09  1.324463e-09  1.330941e-09  1.343583e-09

> s$selection
AIC(n)  HQ(n)  SC(n) FPE(n) 
     2      1      1      2 
```

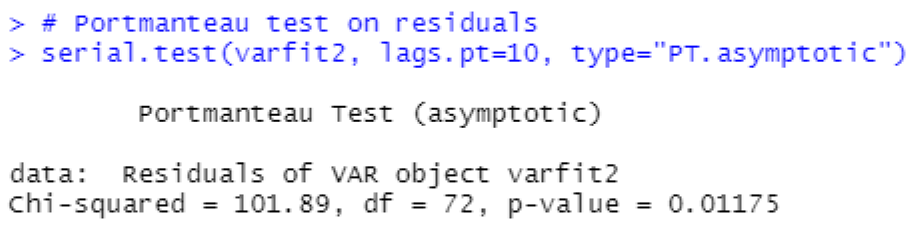
All of the loss metrics suggested an order of 1 or 2. I experimented with those, as well as a handful of higher order values to see if adding more complexity improved the model (it did not). Once I decided on an order of 2, I created a VAR model on the training set that contained the 3 series.



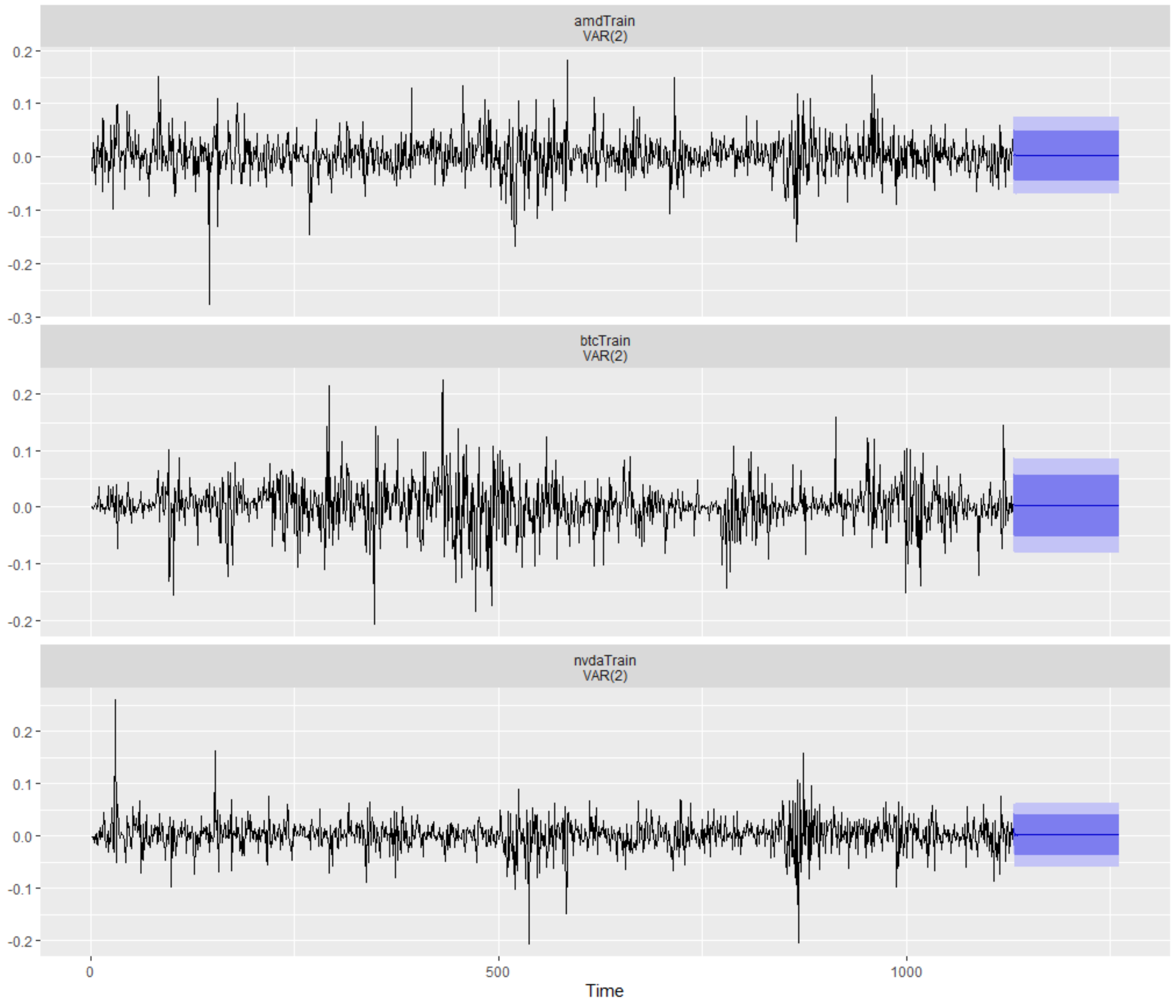
```
> #### 2nd order ####
> varfit2 = vars::VAR(y=alltrain,p=2, type = c("const"))
> varfit2

VAR Estimation Results:
======================= 

Estimated coefficients for equation btcTrain: 
============================================= 
Call:
btcTrain = btcTrain.l1 + amdTrain.l1 + nvdaTrain.l1 + btcTrain.l2 + amdTrain.l2 + nvdaTrain.l2 + const 

 btcTrain.l1  amdTrain.l1 nvdaTrain.l1  btcTrain.l2  amdTrain.l2 nvdaTrain.l2        const 
 0.003592971  0.007107866  0.015960319  0.022790855  0.044178630 -0.002235902  0.002222974 


Estimated coefficients for equation amdTrain: 
============================================= 
Call:
amdTrain = btcTrain.l1 + amdTrain.l1 + nvdaTrain.l1 + btcTrain.l2 + amdTrain.l2 + nvdaTrain.l2 + const 

 btcTrain.l1  amdTrain.l1 nvdaTrain.l1  btcTrain.l2  amdTrain.l2 nvdaTrain.l2        const 
-0.015539367 -0.055758008 -0.017360330 -0.032185937  0.089441143 -0.017081639  0.002286125 


Estimated coefficients for equation nvdaTrain: 
============================================== 
Call:
nvdaTrain = btcTrain.l1 + amdTrain.l1 + nvdaTrain.l1 + btcTrain.l2 + amdTrain.l2 + nvdaTrain.l2 + const 

  btcTrain.l1   amdTrain.l1  nvdaTrain.l1   btcTrain.l2   amdTrain.l2  nvdaTrain.l2         const 
 0.0124770873  0.5190378278 -0.0754674179  0.0095422490 -0.0469391992  0.0233048190  0.0008495681 
```

The P-value of the 2nd order model was the highest, but it still rejected no autocorrelation. I still moved forward with forecasting to see if the VAR model captured any of the variance in the test set.

DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite



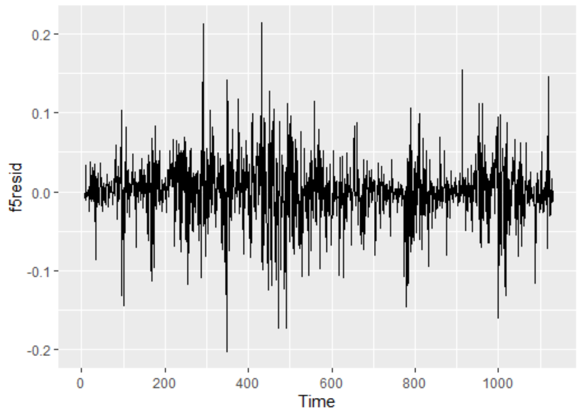
```
> # Portmanteau test on residuals
> serial.test(varfit2, lags.pt=10, type="PT.asymptotic")

        Portmanteau Test (asymptotic)

data:  Residuals of VAR object varfit2
Chi-squared = 101.89, df = 72, p-value = 0.01175
```



> f2 = forecast::forecast(varfit2,newdata=alltest,h=130)
> autoplot(f2)

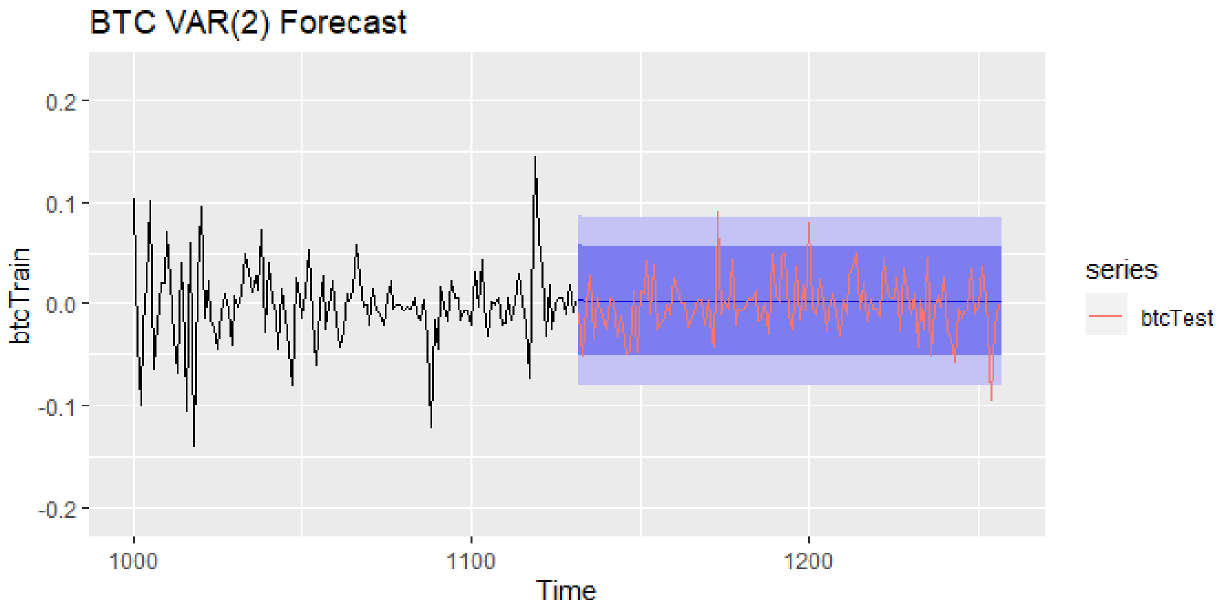
Residuals of the fitted model on the training set:

DSC 425 Time Series Analysis and Forecasting
Appendix to the Final Project Report (Appendix_EvelinaR.pdf)
Due 11/21/21
Evelina Ramoskaite



f2resid = f2[["forecast"]][["btcTrain"]][["residuals"]]
autoplot(f5resid)

Here is a screenshot of the bitcoin forecast compared to the actual values in the hold-out set:



The forecasts for every series quickly flatlined to the mean and failed to capture the variation in the test set. Also, the residuals were not stationary. Unfortunately, there did not seem to be a valuable relationship we could exploit in this line of analysis.