Final Project Report

Group 8 : Electricity

Chinmay Patil

Vaidehi Madhu

Ashay Kargaonkar

Pramathesh Shukla

TABLE OF CONTENTS :

NON-TECHNICAL SUMMARY

For this project we took 4 cryptocurrencies into consideration. The currencies we are working on are Ethererum , Tether , LiteCoin and Bitcoin Cash. Each member from the group is modeling on different currencies. Data entries are recorded on a daily basis.We are working on a highest value of the currency for the day.Our aim is to identify and compare the patterns in above cryptocurrencies and to forecast them to see their behaviour.

Ethereum was first launched in 2013. It is the second famous cryptocurrency launched after bitcoin.In 2018 Ethereum boosted around $1400.But after that it dropped to $100 in late 2019 which is almost consistent for 2 years and then in pandemic it again boosetd and now the its price is $1700. The total market capitalization of the currency is $206.90B.

Litecoin was first launched in 2011. In mid 2017 litecoin updated their framework and reduced the transaction time.In mid 2019, litecoin was boosted around $140 from $40 at early 2019. The pandemic also boosted this currency and now its price is $199.94 with market capitalization of $13.42B

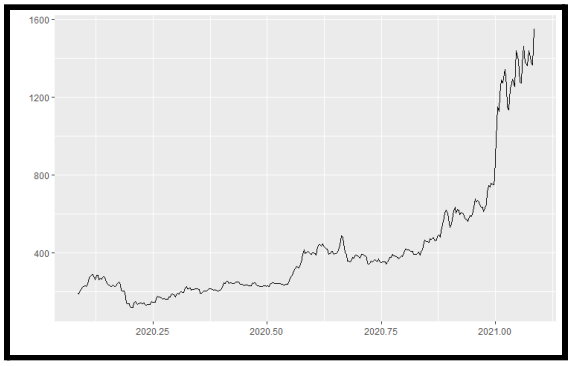
Tether, launched in 2014 started at $1 cost and still the price is $1. This shows that this is the most consistent cryptocurrency. But It showed a huge drop in late 2019 which was less than $0.96.This currency has market capitalization of $40.06B

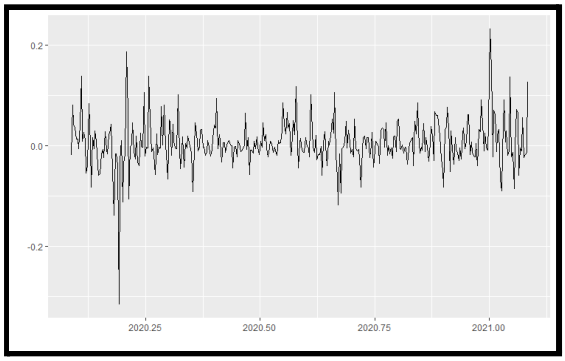
Bitcoin cash is the one of the new cryptocurrency launched in 2017 by forking the bitcoin blockchain.In mid 2018 bitcoin cash was on peak of $1000+ but again in 2019 it dropped by almost 90% and the price at the moment was $174 which states that it has high volatility rate. But again in pandemic it started boosting and now the price of the currency is $523.53 with market capitalization of 9.78B.

Exploratory Data Analysis:

Ethereum:

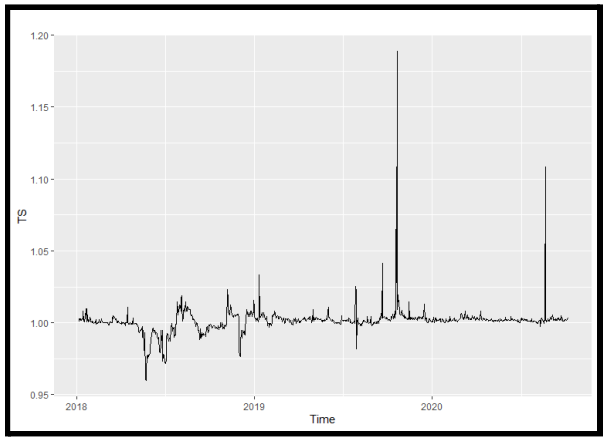
Initial plot of the ethereum shows that the plot is multiplicative.So converting it to the log-return so that we can use that time series for further analysis.There is non-stationary behavior in the graph so we will have to make it stationary.

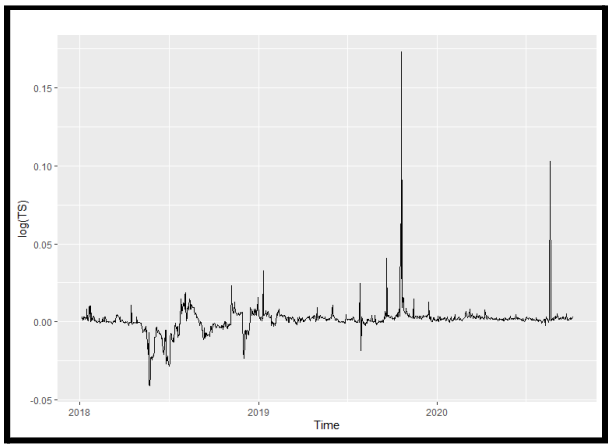




Tether:

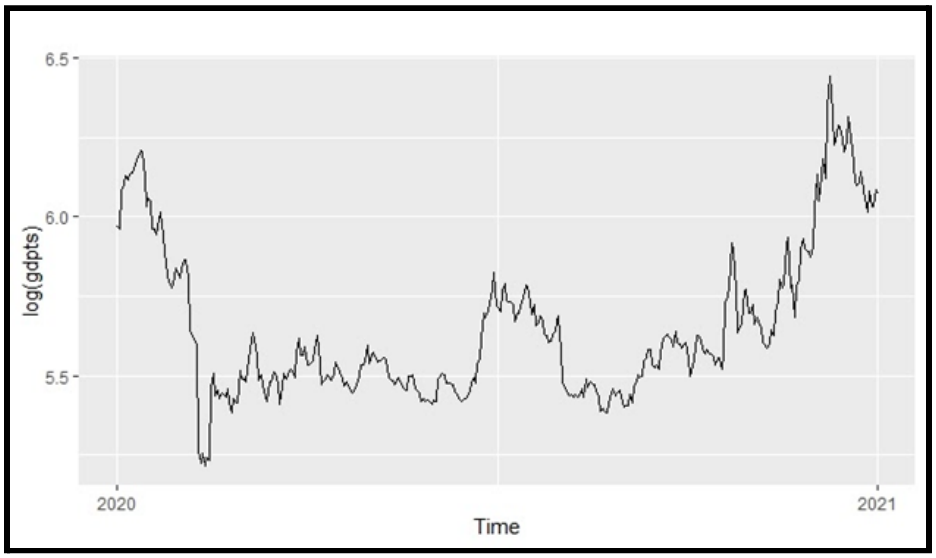
Created a time series of a High variable ranging from last year data from June 2018 to February 2021 using daily frequency. The time series plot looks non stationary with no trend or no seasonal component. The plot shows the time series is Multiplicative. This time series has a random factor which shows it has moderately randomness. To make the time series additive series, performed a log transformation on the time series.





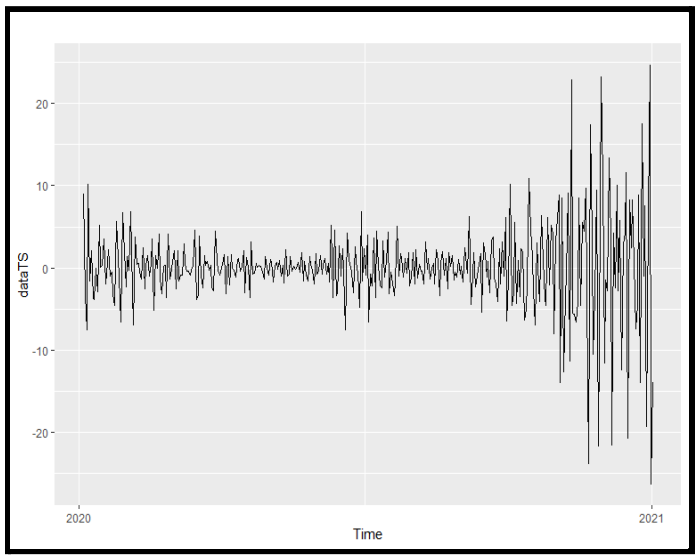
BitCoin Cash:

The initial plot of bitcoin cash is shown below. The data was ranging from 2020-2021. The plot was multiplicative. For further analysis it converted to additive. It is showing the non-stationary behavior. We also converted to stationary for further analysis.



LiteCoin:

The time-series was already additive and I wasn't required to take the log of it. Later I did the ADF and KPSS test to check its stationarity and I saw it wasn’t. Therefore I took the difference of time series twice and got the series stationary. Below image shows the series after differencing and its adf and kpss test result.

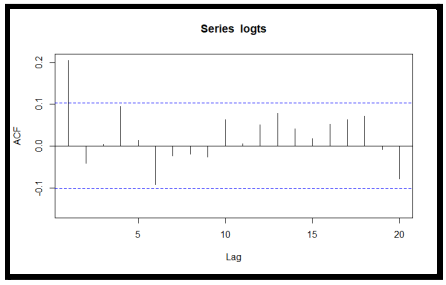


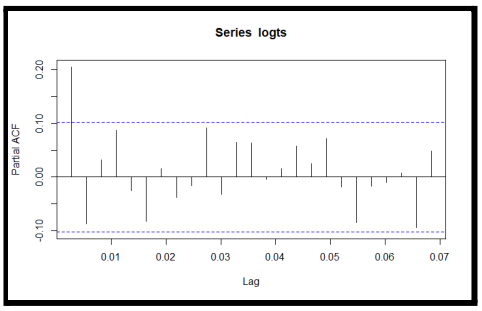

```
> adf.test(diff2)

        Augmented Dickey-Fuller Test

data:  diff2
Dickey-Fuller = -10.948, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(diff2) : p-value smaller than printed p-value
> kpss.test(diff2)

        KPSS Test for Level Stationarity

data:  diff2
KPSS Level = 0.035664, Truncation lag parameter = 5, p-value = 0.1
```

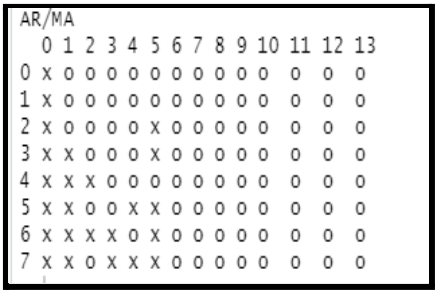
Model Fitting - Residual analysis - Model diagnostics

Ethereum:

The First model used is ARIMA. To serve the conditions of arima model order ACF, PACF and EACF plots are generated.ACF plots show correlation at lag-1 only so for the ARIMA model considering AR(1) or MA(1) order might get significant values. Eacf confirms that there is some autocorrelation in ARMA(1,1) order. After creating a few models, I finally decided to use the ARMA(0,1) model in which only a significant term is MA.

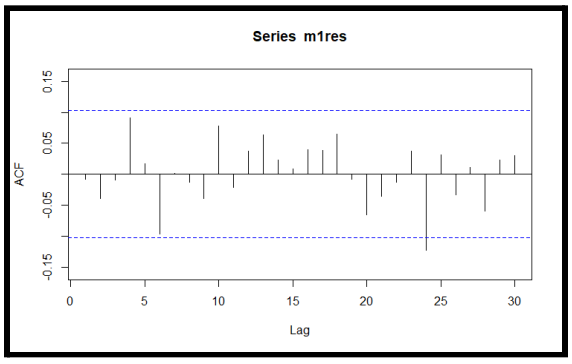


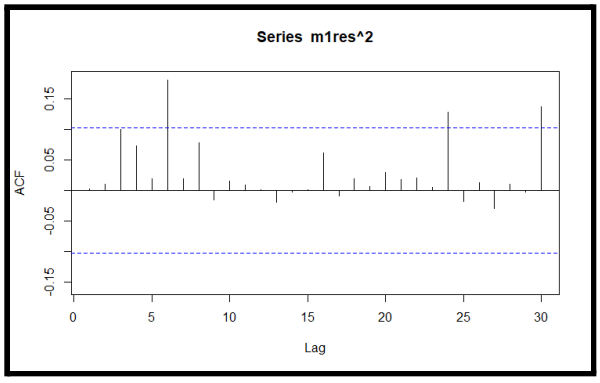




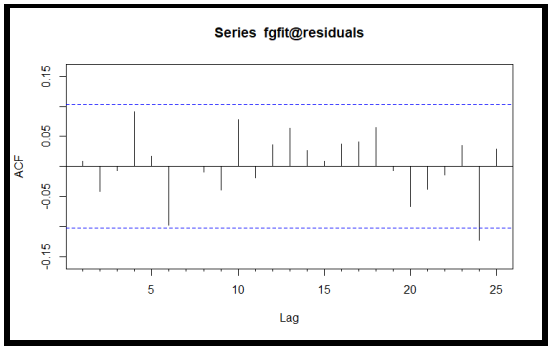
After generating the ARIMA model to check the efficiency of the model we have applied residual analysis to check the remaining residuals are just white noise or not, and after plotting the ACF of the ARIMA model residuals it proves that the remaining residuals are just white noise. Also, the Lung Box test cannot reject the null hypothesis suggesting the residuals are white noise with high confidence.

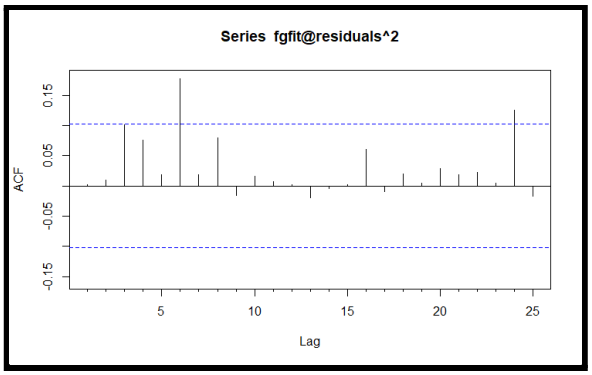
The next step was to check the autocorrelation and ARCH effect in the residuals of the ARIMA model and in order to do so the ACF plot of the residuals squares was built and there we found the slow decay in residuals, that was the indication of ARCH effect in the series. The graph of residuals squares is displayed below.





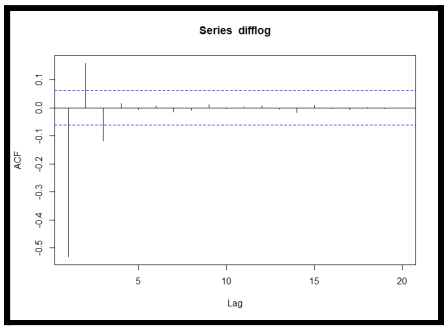
Another step was to capture the volatility of the series as the ARCH effect was present in the series, and to do so the GARCH model was constructed on the series. The order of the model was GARCH(1,1) – ARMA(0,1). This was the best model selected based on the residual analysis and the backtest error scores. For the residual analysis both the residuals and the squared residuals ACF were built and that were white noise only, this is also proved by the Ljung box test whit did not reject the null hypothesis, suggesting mostly white noise**.**

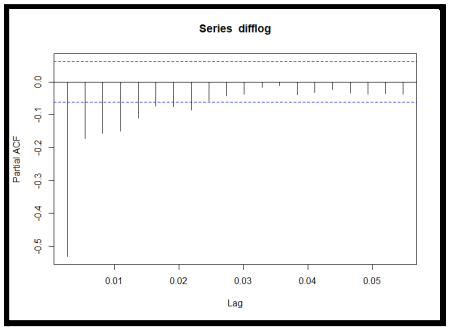




Tether:

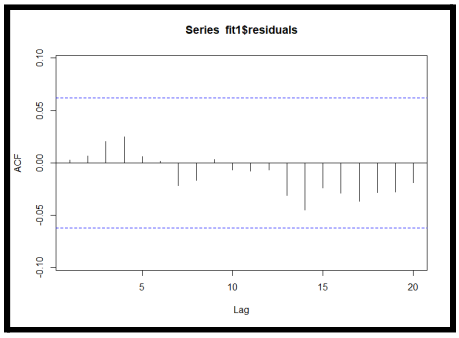
The eacf plot (VM 1) shows ARIMA(0,0,3) or ARIMA(1,0,3). It was observed that AR(1) coefficient is insignificant in ARIMA(1,0,3) hence, fit the model with ARIMA(0,0,3). The acf plot of squared residuals (VM 2) shows no autocorrelation. The plot of squared residuals (VM 3) show volatility indicating arch effect. Then fitted model 3 using GARCH(1,1) (VM 4). Residual analysis was performed on all the models to select the final model. The backtesting of model MA(3) is shown in the VM 6.

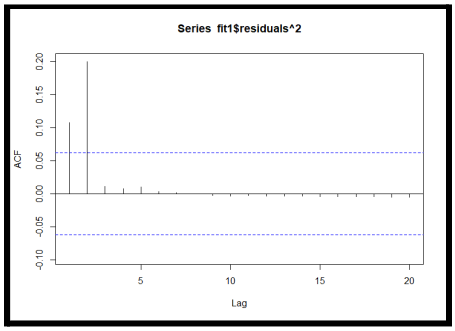





```
> eacf(difflog)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x o o o o o o o  o  o  o
1 x x x o o o o o o o  o  o  o
2 x o x x o o o o o o  o  o  o
3 x o x x o o o o o o  o  o  o
4 x x x o o o o o o o  o  o  o
5 x x x x o o o o o o  o  o  o
6 x x x x o o o o o o  o  o  o
7 x x x x o o o o o o  o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

(VM1)

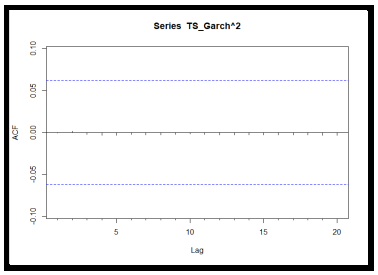


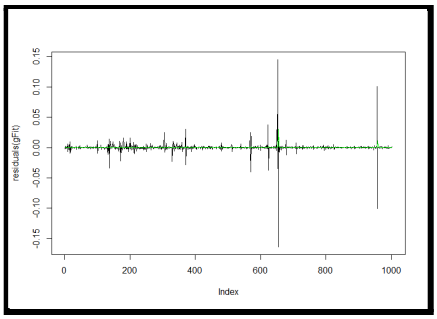



```
Error Analysis:
       Estimate  Std. Error  t value Pr(>|t|)
mu    1.485e-06   1.420e-04    0.010 0.991655
omega 1.598e-05   1.315e-06   12.150  < 2e-16 ***
alpha1 1.000e+00   1.784e-01    5.606 2.07e-08 ***
beta1  1.472e-01   4.259e-02    3.457 0.000547 ***
---
```

(VM2) **(VM3)** **(VM4)**

The residuals of model (0,0,3) show slight normality but have outliers. The residuals also show noticeable bias and significant autocorrelation. The residuals of the GARCH model (VM 4) exhibit white noise behaviour. From squared residuals plot, It is observed that volatility was significantly reduced.



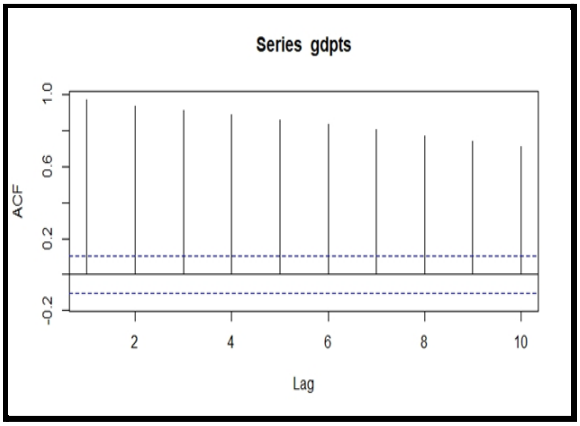


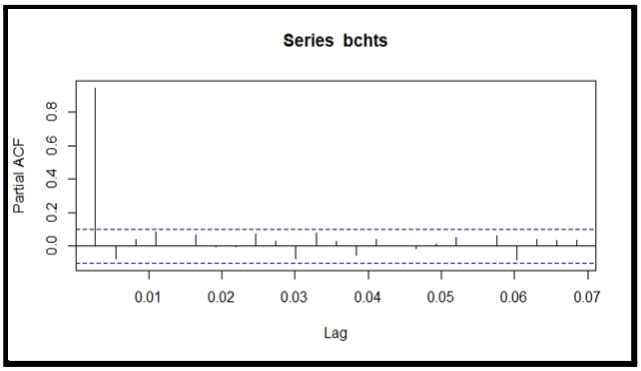

```
> b1 = backtest(fit1, TS, h=1, orig=.9*n)
[1] "RMSE of out-of-sample forecasts"
[1] 0.01195868
[1] "Mean absolute error of out-of-sample forecasts"
[1] 0.002711003
[1] "Mean Absolute Percentage error"
[1] 0.002603906
[1] "Symmetric Mean Absolute Percentage error"
[1] 0.002641389
```

(VM5) **(VM6)**

Bitcoin Cash:

For model building first check the ACF, EACF and PACF. In ACF, I found that the series has AR behavior so after performing EACF and PACF and model building, I finally decided to choose the AR(1) model. Series is non stationary so in the arima model I took integrated term I as 1.

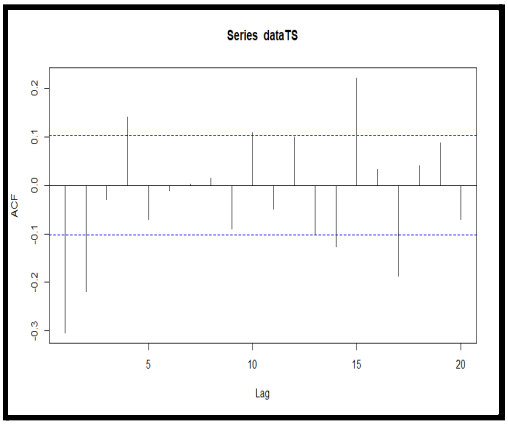


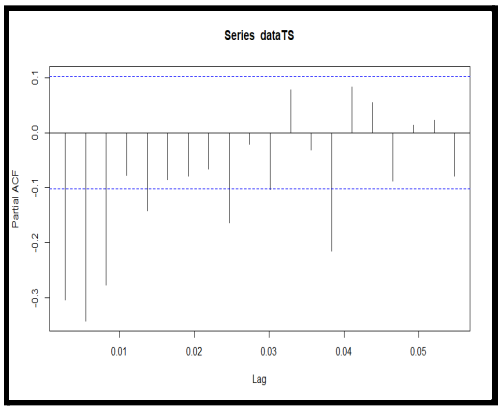



```
> eacf(dataTs)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x x x x x x x x  x  x  x
1 x x o o o o o o o x o  o  o  o
2 x x x o o o o o o o x  o  o  o
3 x x o o o o o o o o o  o  o  o
4 x x o o o o o o o o o  o  o  o
5 x x o x o o o o o o o  o  o  o
6 x o o x o o o o o o o  o  o  o
7 x o x o o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

Litecoin:

For making a model I first checked the ACF, PACF and the EACF of the graph and tried to get to know about the order of the best model.






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

By looking at the ACF graph it doesn't show any AR order model but can't confirm at this stage. Later by looking at ACF and PACF graphs there is an ARCH effect seen because there are multiple lag values above confidence level. Therefore, we will have to use the GARCH model for it. Moreover, by looking at EACF plot, it suggests that the ARMA model of order (2,1) might be a good fit. But to confirm it we have to look for auto arima and manual model building.

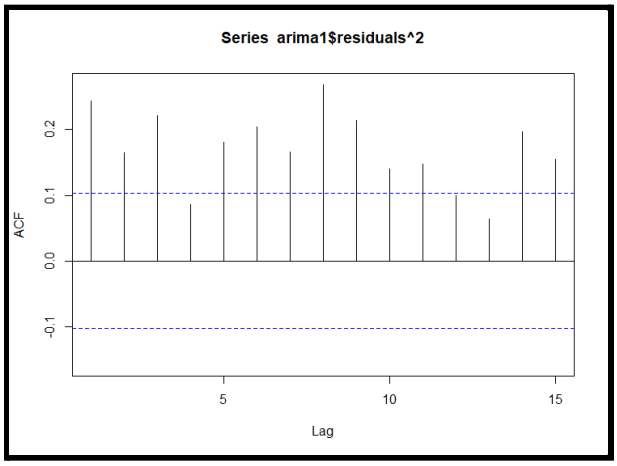
```
> autoarima = auto.arima(dataTS)
> autoarima
Series: dataTS
ARIMA(2,0,1) with zero mean

Coefficients:
         ar1      ar2      ma1
      0.2218  -0.1545  -0.9806
s.e.  0.0534   0.0534   0.0115

sigma^2 estimated as 18.23:  log likelihood=-1044.99
AIC=2097.98   AICc=2098.09   BIC=2113.57
> coeftest(autoarima)

z test of coefficients:

     Estimate Std. Error  z value  Pr(>|z|)
ar1  0.221843   0.053397   4.1546 3.259e-05 ***
ar2 -0.154535   0.053358  -2.8962  0.003777 **
ma1 -0.980584   0.011523 -85.0989 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By looking at the auto-arima model it suggests that the arima model of order (2, 0, 1) might be a good fit for the model. We can see that the sigma-square estimate value is low which is good. Also, the p-values in coeftest are less than 0.05 which states that AR1, AR2, and MA1 are significant and can be considered for further analysis. Also, standard error of these variables are less. Moreover, the rest of values like AIC, AICc and BIC values are significant.

Below 3 values display the values of RMSE (Root Mean Square Error), MABSO (Mean absolute error of out-of-sample) and MAPE (Mean Absolute Percentage Error) and they are also significant.

```
$rmse
[1] 4.786135

$mabso
[1] 2.878273

$mape
[1] 1.642369
```

After trying the manual model, I came to the conclusion that the model built by auto-arima is the best fit and I will consider this model for further analysis in later parts.



By looking at the above ACF plot of residuals-square of the model, I think there is definitely an ARCH effect. Therefore I applied the GARCH model and saw that ARMA model of order (2, 2) fits the best. Below model shows the GARCH model fitted with ARMA (2, 2) model.

By looking at the values below, it looks like all parameters, i.e. mu, ar1, ar2, ma1, ma2, omega, alpha1 and beta1 have less p-value and are therefore significant to use.

```
         Estimate  Std. Error    t value Pr(>|t|)
mu      3.140e-03   1.382e-04     22.730  < 2e-16 ***
ar1     1.987e-01   6.513e-05   3051.322  < 2e-16 ***
ar2    -1.152e-01   6.512e-05  -1769.493  < 2e-16 ***
ma1    -9.992e-01   3.823e-05 -26134.752  < 2e-16 ***
ma2    -4.986e-02   3.856e-05  -1292.852  < 2e-16 ***
omega   4.646e-02   2.232e-02      2.081   0.0374 *
alpha1  1.143e-01   2.331e-02      4.903 9.45e-07 ***
beta1   8.958e-01   1.818e-02     49.264  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 -860.5168    normalized:  -2.364057

Description:
 Tue Mar 16 14:38:40 2021 by user: ashay
```

Forecast Analysis:

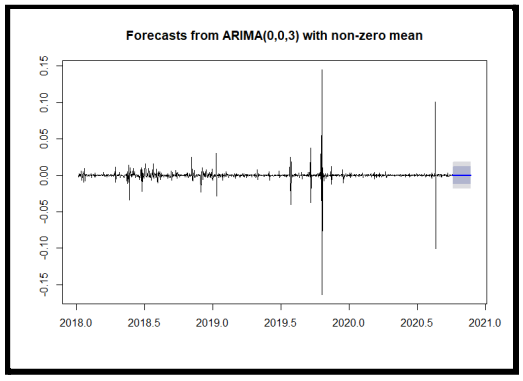
Ethereum :

The best model here is a GARCH model.So, I used that model to predict the forecast.The 5-step ahead forecast of the model is displayed below.

	meanForecast	meanError	standardDeviation
1	0.034429684	0.05918662	0.05918662
2	0.006637042	0.06049748	0.05902366
3	0.006637042	0.06033354	0.05886385
4	0.006637042	0.06017278	0.05870713
5	0.006637042	0.06001513	0.05855344

Tether :

The forecast of the ARIMA(0,0,3) (VM 7) model and GARCH model (VM 8) shown below:




```
> f = predict(gFit, n.ahead=5)
> f
  meanForecast   meanError standardDeviation
1  1.48483e-06 0.004473815       0.004473815
2  1.48483e-06 0.006240033       0.006240033
3  1.48483e-06 0.007787604       0.007787604
4  1.48483e-06 0.009249404       0.009249404
5  1.48483e-06 0.010682816       0.010682816
```

(VM7) **(VM8)**

Bitcoin Cash:

Below graph predicts five steps ahead values after using garch models.By looking at the below garch model fit Table we can see that except omega Other values are significant.

```
> predict = predict(gfit,n.ahead =5)
> predict
  meanForecast meanError standardDeviation
1     426.6411  20.51227          20.51227
2     420.7471  28.68304          20.73020
3     415.0510  34.74507          20.94798
4     409.5462  39.69263          21.16563
5     404.2261  43.91733          21.38318
```

Litecoin:

I predicted the forecast value for 5 steps and below figure shows the values of those.

```
> predict
  meanForecast meanError standardDeviation
1  10.30915869  10.88505          10.88505
2   4.14868101  13.98757          10.94209
3  -0.36032864  14.49684          10.99941
4  -0.54652201  14.58963          11.05700
5  -0.06394822  14.66667          11.11488
```

Analysis of the results and discussion:

Ethereum :

After applying backtesting both models ARIMA and GARCH with 98% of n. The results show that ARIMA performs slightly better than the GARCH model.The results of the model are displayed below.

Model	MAPE	SMAPE
ARIMA(0,0,1)	1.33	1.64
GARCH(1,1) -ARMA(0,1)	1.34	1.80

Tether:

After applying backtesting both models ARIMA and GARCH with 98% of n. The results show that GARCH performs way better than the ARIMA model.The results of the model are displayed below.

Model	MAPE	SMAPE
ARIMA(0,0,3)	4.093124	1.22
GARCH(1,1)	1.00922	1.917231

Bitcoin Cash:

After applying backtesting both models ARIMA and GARCH. The results show that GARCH performs slightly better than the ARIMA model.The results of the model are displayed below.

Model	MAPE	SMAPE
ARIMA(0,0,1)	1.64	1.03
GARCH(1,1)	0.73	0.92

LiteCoin:

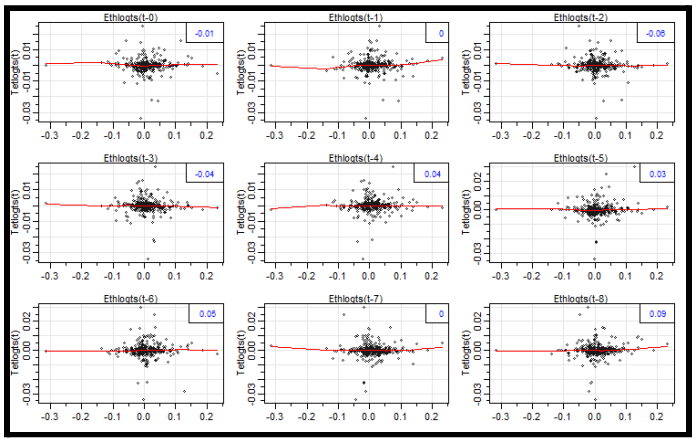
By applying backtest on ARIMA and GARCH models, we can see that both models are showing similar values but the ARIMA model is slightly better than the GARCH model but still GARCH is good and is significant enough to consider for future analysis.

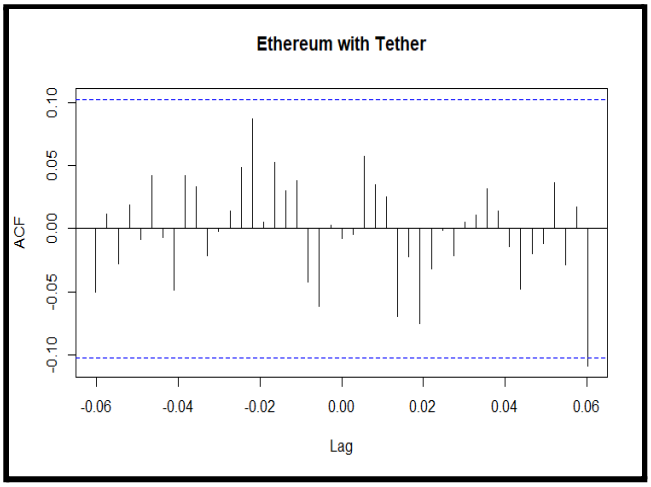
Model	MAPE	SMAPE
ARIMA(2,0,1)	1.56	1.05
GARCH(1,1) -ARMA(2,1)	1.77	1.15

CryptoCurrency Analysis

Ethereum - Tether:

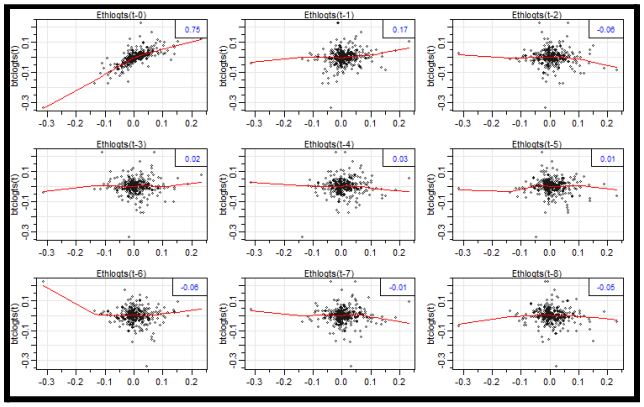
The analysis is done on the log returns of ethereum and log returns of tether. The highest correlation captured at lag-8 which is 0.08

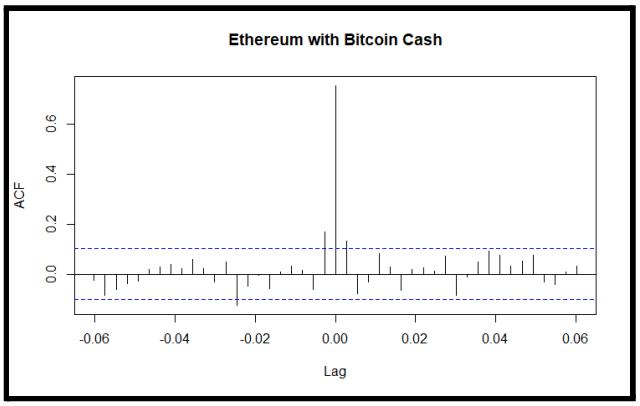




Ethereum -Bitcoin Cash:

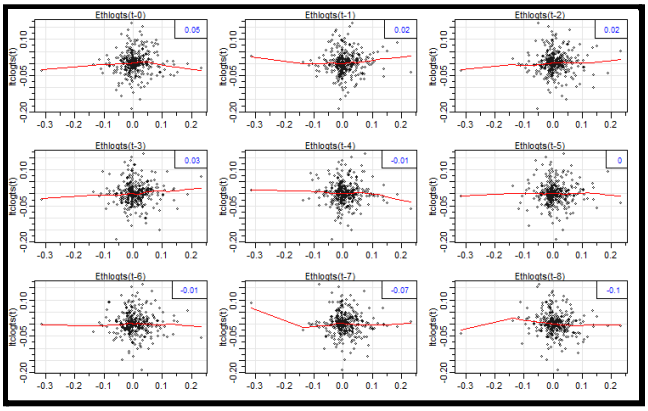
Ethereum shows maximum correlation with the bitcoin cash at lag-0 which is 0.75 after that at lag-1 the correlation is 0.17

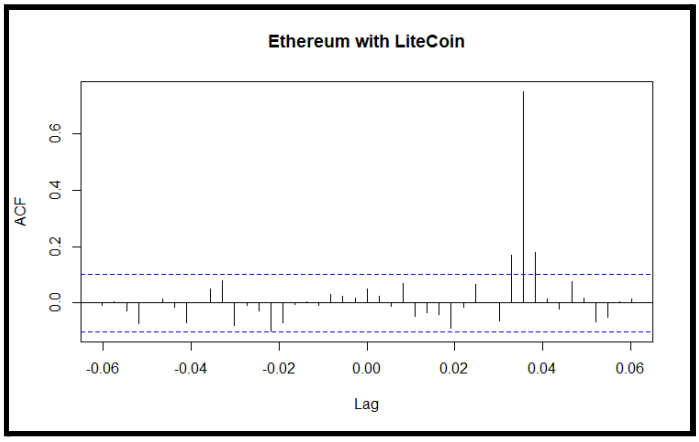




Ethereum -LiteCoin:

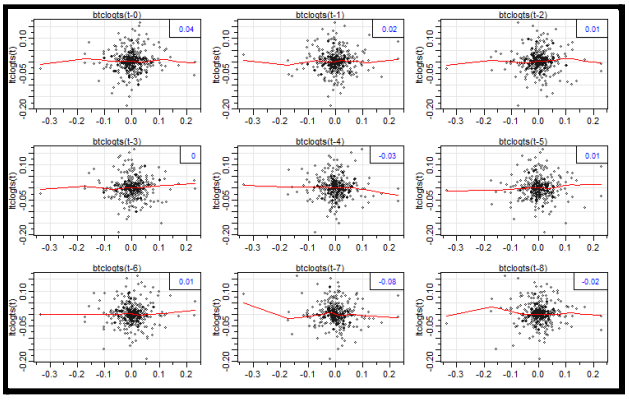
Next analysis is the correlation of log returns of Ethereum and Litecoin. This analysis shows maximum correlation at lag-0 which is 0.05

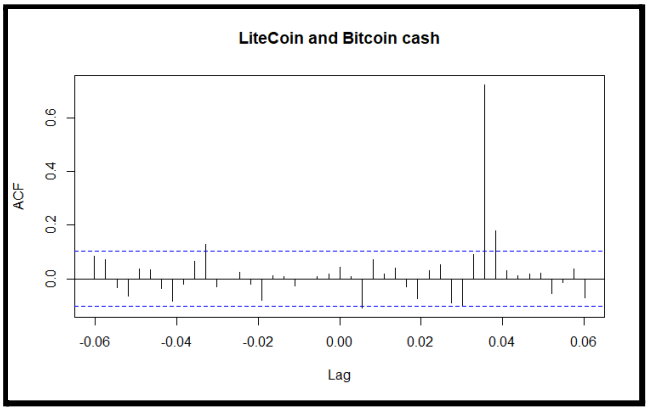




LiteCoin - Bitcoin Cash :

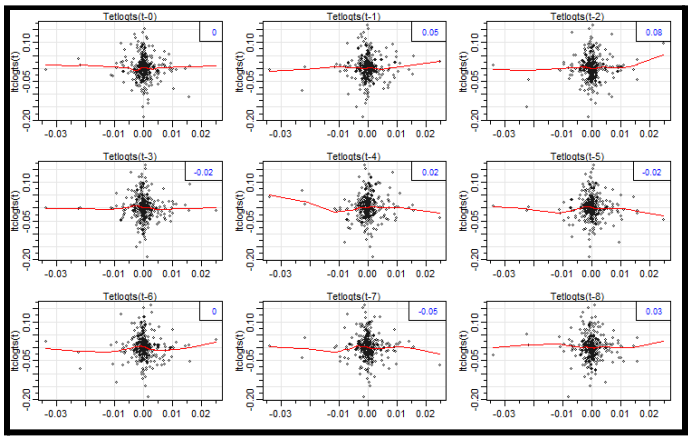
Log returns of bitcoin cash and litecoin shows maximum correlation at lag-0 only which is 0.04 and high negative correlation at lag-7 which is -0.08

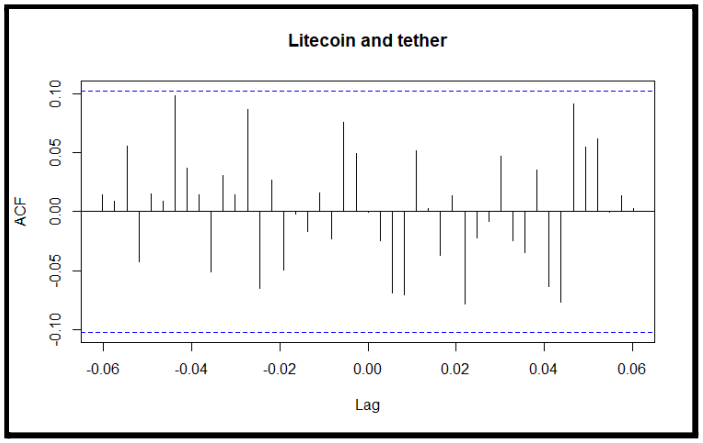




LiteCoin - Tether :

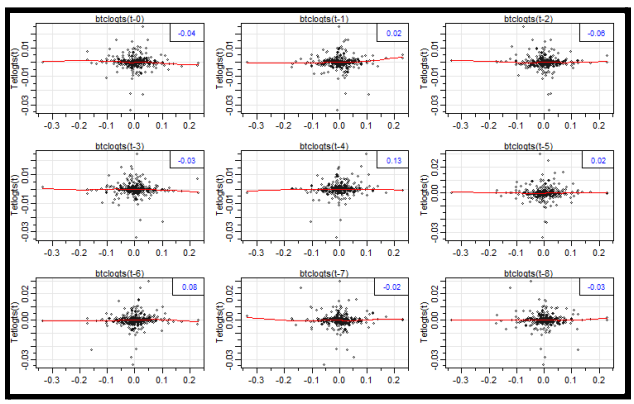
When litecoin is compared with tether it shows maximum correlation at lag-2 which is 0.08.Lag2 plot and CCF plots are displayed below.

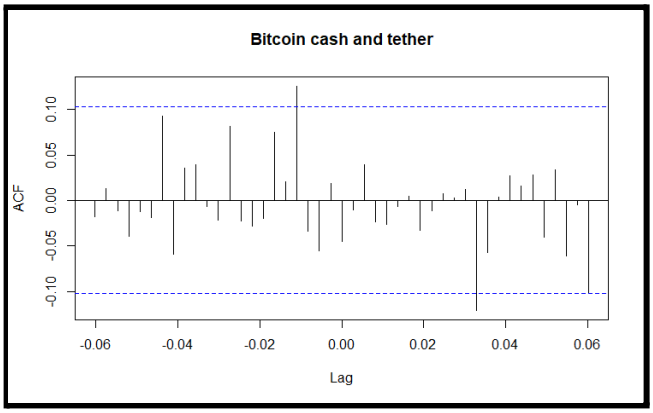




Bitcoin Cash - Tether :

Log returns of the bitcoin cash shows maximum correlation at lag-4 which is 0.13 . The results of the same are displayed below.





Vector Autoregressive Model:

All 4 cryptocurrencies are further studied via VAM. ACF and Cross correlation is applied on all the cryptocurrencies.From that it can be said that data has maximum correlation till at lag-4. But when automatic suggestion VARselect applied on all 4 series it gives the best lag for correlation is 1, both AIC and BIC suggest this lag. So further the model is built using the cbind of all 4 series with the lag 1.when the significance is tested on the model using the serial test it rejects the null hypothesis that the residuals are white noise and accepts the alternative not being white noise suggesting some autocorrelation. So, this suggests that the series of all 4 currencies does not have in common that much that can be modeled.

Individual Reporting

Chinmay Patil (Ethereum) :

Data Preprocessing :

For this project I worked on an ethereum dataset. The dataset is taken from coindesk. The data ranges from Feb2020 - Feb 2021 in which all the entries are recorded on a daily basis.The Dataset contains the highest price of the day , opening price of the day and closing price of the day. We are considering the highest price of the day for the analysis.The data is already in cleaned format so we did not perform any cleaning on the dataset.

Time Series Analysis:

The first step is to convert the date field in date format so that time series can understand.After that the highest price is taken to construct the time series. Our data is recorded on daily basis so the frequency i chose Is 362.25 as the standard way for daily time series data. Building auto plot suggests that the series is multiplicative and hence the Log transform has been applied to the series. After the first order difference has been applied to the series.

The next step is to calculate ACF ,PACF and eacf of the series which suggested the ARMA order of 1,1.EACF was not giving the clear indication; instead it was giving the 0,4 order of the ARMA. In order to build the best fitting model, I have iterated through various orders of the AR and MA. At last the best model generated manually was of order in ARMA 0,1.which has the least sigma squared and AIC scores as well as compared to other iterations. The coefficients are also significant which was checked using the coeftest. To add to it the residuals of the generated model are also white noise means the model is capturing the maximum autocorrelations. Which is also proved using the Ljung box test, in which the passed model cannot reject the null hypothesis with high confidence. At last the Backtest has been applied to estimate the errors, and the Mean Absolute Percentage Error is 1.33.

Next the ARCH effect was checked using plotting the ACF of the residual squares of the ARIMA fitted model. AS a result, the residual squares were having the slow decay which suggested the presence of the ARCH effect in the series and to capture it the GARCH model has been generated by passing the residuals of the ARIMA model. For the GARCH model too the order of ARMA was a tough nut to crack. Many iterations have been applied to the get best ARMA order. Lastly the ARMA 0,1 was the best and significant model to capture the volatility of the series. Talking about the residual analysis the residuals are white noise, which also proves using the Ljung box test that the applied residuals cannot reject the null hypothesis. To add to it the squared residuals of the GARCH model are also white noise there is no trace of autocorrelation in it.Lastly the back test is applied on the model to check the error estimates of the model and the Mean Absolute Percentage Error of the GARCH model is 1.34 which is almost same as the ARIMA model.The GARCH model is SAMPE 1.8 .At last the forecast is applied on the model.

Comparison with the Other Cryptocurrencies:

Further analysis is done by comparing each cryptocurrency with one another. The CCF plot and the lag2 function is used to see the correlation among the currencies. The results showed most of the correlation is present at lag-1 but some correlation found at lag-4 too.
All 4 companies stock data is further studied via VAM. ACF and the cross correlation is applied on all 4 currencies. From that it can be said the data has max correlation till lag 3. But when the automatic suggestion technique VARselect is applied on all 4 series it gives the best lag for correlation is 2, both AIC and BIC suggest this lag. So further the model is built using the cbind of all 4 series with the lag 2. But when the significance is tested on the model using the serial test it rejects the null hypothesis that the residuals are white noise and accepts the alternative not being white noise suggesting some autocorrelation. So, this suggests that the series of all 4 currencies does not have in common that much that can be modeled.

Vector Autoregressive Models:

All 4 cryptocurrencies are further studied via VAM. ACF and the cross correlation is applied on all 4 currencies. From that it can be said the data has max correlation till lag 4. But when the automatic suggestion technique VARselect is applied on all 4 series it gives the best lag for correlation is 1, both AIC and BIC suggest this lag. So further the model is built using the cbind of all 4 series with the lag 2. But when the significance is tested on the model using the serial test it rejects the null hypothesis that the residuals are white noise and accepts the alternative not being white noise suggesting some autocorrelation. So, this suggests that the series of all 4 currencies does not have in common that much that can be modeled**.**

Take Away from the course:

At First place , I thought this course would be tough but as we went through the course this course is not challenging if you study on a daily basis. The most important thing in this course is to understand concepts you can code or you can get the materials online but you need to properly understand the basic concepts of the time series like smoothing , date factor etc. These are some basic things but they are very important. Manual model building can give you better results than the auto arima model. I got confused many times with many concepts but the lecture helped me to clear them.This course took me to the next level as a data scientist. Data scientist must have time series skill to land a in good company with good job

Vaidehi Madhu (Tether) :

I worked on Tether Cryptocurrency. After doing some background research about the variables. I decided to analyze the High variable of Tether dataset. Firstly, I performed data preprocessing to check the correctness of the columns and analyze it. During the data cleaning stage, there were no null values in the dataset. After the data cleaning stage I created a time series for High variable and analyzed it's distribution using time series and auto plot. The series is nonstationary and does not have any trend or seasonality in it. The time series looks like a multiplicative series just like any other financial dataset. I took a log transformation to convert it to an additive time series. Since the series is non stationary, I perform the first difference. The time series is ready to perform a modeling.

The order of the model is chosen by analyzing acf, pacf and eacf. I performed 2 models (1,0,3) and (0,0,3). The best model was (0,0,3) since it has all variables significant. Model(1,0,3) does have AR(1) insignificant. The residual analysis of model (0,0,3) indicates that the residuals show white noise behavior and it was a good fit. The residuals of the fitted model show ARCH effect hence, I fit model3 using GARCH(1,1). The order of the GARCH model is selected by analyzing acf, pacf and eacf plots. After fitting all the models, the model performance of all the models are evaluated by performing residual analysis and goodness of fit. Of all the three models, GARCH has less AIC, BIC values and high likelihood ratio, so I selected GARCH as the final model and performed 5 step forecasting. Finally, analyzed the forecast and concluded that the volatility of the High variable will persist for a short duration after reaching its peak. Since the volatility persists for a short time it is better to buy the currencies after when it returns to normal.

Take Away from the course:-

Time series forecasting plays a major part in most of the domains. This interest made me take a time series analysis course. The course was well organized and had all the resources we needed to understand the topic. Overall, the class was very informative and had a lot of fun. It was a good learning experience.

Pramthesh Shukla (Bitcoin Cash) :

Preprocessing:

We worked on the Cryptocurrency data analysis which was recorded from 2020 to 2021. The dataset consists of a total of 20 assets, so each of the members picked different assets to apply the analysis on.
The first step is to differentiate the data from the main aggregated dataset. Second step after generating a separate dataset was to deal with the missing values of each data. The NA values were handled using the forward fill technique. After unwanted columns were removed from each dataset. This was the preprocessing done on the main dataset.

ANALYSIS:

I have worked on a Bitcoin cash dataset derived from Cryptocurrency. It was ranging from 2020-2021. After some research I selected the High variable of bitcoin cash dataset. Firstly, I analyzed the dataset. Converted it to month, date, and yearly format. Secondly, I checked for the null values. After all cleaning and converting the dataset into proper date format, I created the time series for the high variable and created the autoplot. When I did the analysis, I found that the series was multiplicative and non-stationary. I converted both multiplicative and non-stationary to additive and stationery using the difference.I have performed ACF, PACF and EACF. I have performed a total of 5 models. After all models I found the best model was (0,0,1). Because all the variables were significant. The residual analysis of model (0,0,1) indicates that the residuals show white noise behavior, and it was a good fit. The order of the GARCH model is selected by analyzing ACF, PACF and EACF plots. After fitting all the models, the model performance of all the models is evaluated by performing residual analysis and goodness of fit. Of all the three models, GARCH has less AIC, BIC values. After that I performed the 5-step forecasting.

Take Away from the course:-

I have learned a lot from this course. How we can use time series analysis on data which is time related. For instance I learned how statistical methods like AR, MA, ARIMA, ARCH, GRACH, forecasting can apply to the data to make models and it can help in future prediction. All things to learn were made easy to learn by the examples provided by the professor.

Ashay Kargaonkar (LiteCoin) :

Pre-processing:

I have worked on LiteCoin cryptocurrency which ranges from **16th Feb 2020 to 15th Feb 2021**. This database contains variables like Date, Closing Price, Open Price, Low Price and High Price of a day. Our project members have selected **High Price** and later will compare other 3 cryptocurrency High Price values with mine.

Firstly, I saw that the series was additive and not multiplicative so there was no reason to take log transformation. But still the adf test and kpss test were suggesting that the series is stationary. Therefore, I took differencing twice and made the series stationary. Also, adf and kpss tests supported this notion.

Also, I converted the date column into “Date” format for creating time series. I have selected the frequency as 365.25 because I was going to perform analysis on a year's data.

As the download data was already cleaned I don't have to clean it explicitly.

Time-Series Analysis:

Firstly, I tried to find the best ARIMA model for the data. I looked at the ACF graph and it doesn’t show any AR behaviour. Later by looking at the EACF graph, it states that the ARMA model of order (1, 2) may be significant. Then I performed auto-arima on the time series and auto-arima suggests that the ARIMA model of order (2, 0, 1) is the best model for this dataset. But still to get more clarity I performed manual model building and it resulted that the ARIMA model of order (2, 0, 1) is the best fit, which is the same as that of auto-arima mode. Therefore I will go forward for the further analysis with the auto-arima model i.e. (2, 0, 1). Also, by looking at the coeftest of the auto-arima, it has less value of sigma-square estimate which is good. Also, AIC, BIC values are significant to consider. Moreover, the estimated standard values of these 3 parameters are also less which is good. In addition to this, the p-values of ar1, ar2 and ma1 is less than 0.05 which is good and I think that ARIMA model of order (2, 0, 1) is the best fit. Lastly, backtest was applied by using this model on the time series and saw that the value of MSPE is 1.5 which indicates that this model is a good fit.

Secondly, after looking at the PACF graph of the residual - square value it seems that the series has an ARCH value because there were multiple lag values which were greater than the confidence level. Therefore I applied the GARCH model of order (1, 1) and the ARMA model of order (2, 1), but the values were not significant enough to consider. Therefore, I tried different ARMA orders for this GARCH model,and saw that the ARMA model order of order (2,2) with GARCH order of (1,1) gives very good significant values. The p-values of mu, ar1, ar2, ma1, ma2, omega, alpha1 and beta1 are less than 0.05 which states that all the values are significant to consider. Also, the standard errors of these variables are very less which is good.

Lastly, I checked the backtest by using this model and saw that the value of MAPE is 1.7 and SMAPE is 1.15 which are low. Therefore, I think that GARCh model of order (1,1) with the ARMA model of order (2, 2) is the best fit model for the data and can be used for further forecasting or any further analysis.

Take Away from the course:

This course was very instructive as I have learned a lot about a different kind of analysis i.e. time series analysis. In this course, I learned how to pre-process the data effectively, by altering the time-format in specific format and how different formats can be visualised and help for different kinds of analysis.

Also, I learned about various model building types like autoarima, manual model building, GARCH model building and how to optimise and find a better model for the dataset. Because after finding one model, we can't just rely on that model. But we have to fine tune the model by experimenting with different order values and find the best one. Also, I have learned how to use the ACF, PACF and EACF graphs to find a way for the best model.

Also, even after finding the best model, I learned about the backtesting and forecasting of the model which tells whether the model is good or not. There were a lot of things to learn in this course and I have learned almost everything. Also this course has motivated me to look for a new field in the data analytics world and I will certainly be looking for my future opportunities which will contain this type of time-series analysis.

APPENDIX

Chinmay Patil

```
rm(list = ls())

library(ggplot2)
library(ggfortify)
library(zoo)
library(forecast)
library(fBasics)
library(fpp2)
library(xts)
library(lubridate)
library(Hmisc)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(ggfortify)
library(zoo)
library(forecast)
library(fBasics)
library(fpp2)
library(xts)
library(lubridate)
library(Hmisc)
library(lmtest)
library(tidyquant)
library(fUnitRoots)
library(tseries)
source("eacf.R")
source("Backtest.R")

##########Convert Date format####################
dat = read.csv('C:/Users/patil/OneDrive/Documents/425-Project/ETH_USD_2020-02-03_2021-02-02-CoinDesk.csv')
head(dat)
tail(dat)
class(dat$Date)
dat$Date =  as.Date(dat$Date, format="%Y-%m-%d")

################Time Series#################

DataTS = ts(dat$X24h.High..USD., start = c(2020,32), frequency = 365.25)
autoplot(DataTS)

##########Convert time series to additive by log transform########

logts = diff(log(DataTS))
autoplot(logts)
adf.test(logts)
```

```
###################### docent look stationary so to convert to stationary ########

adf.test(logts)


autoplot(logts)
adf.test(logts)
adfTest(logts) #Statioanry
adfTest(logts, type= 'c') #trend-statioanry
adfTest(logts, type ='ct')#random walk with drift
#########################

Acf(logts,lag.max = 20)
pacf(logts)
eacf(logts)


m1 = auto.arima(logts)
coeftest(m1)

m1res= m1$residuals
Acf(m1res,lag.max = 30)
Acf(m1res^2,lag.max = 30)
pm1 = backtest(m1,logts,orig = nTest,h=1)

######################
nTest = 0.80*length(logts)
####################

m1 = Arima(logts,order =c(2,0,2))
m1
coeftest

Acf(m1$residuals^2,lag.max = 20)
Acf(abs(m1$residuals),lag.max = 20)
Box.test(m1$residuals,lag=15,type= "Ljung-Box")


Acf(m1$residuals,lag.max = 20)
pm1 = backtest(m1,logts,orig = nTest,h=1)
Box.test(m1$residuals,lag = 15)
```

```
############Auto arima####
m2 = auto.arima(logts)
m2
coeftest(m2)
Acf(m2$residuals,lag.max = 20)
pm2 = backtest(m2,logts,orig = nTest,h=2)
Box.test(m2$residuals,lag = 15)
###############GARCH Modeling

res1 = m1$residuals
res2 = m2$residuals
####
archfit1 = garch(res1,order = c(1,1))
archfit1
coeftest(archfit1)
#####

archfit2 = garch(res2,order = c(1,1))
archfit2
coeftest(archfit2)
####
autoplot(res1)
gfit = na.omit(archfit1$residuals)
autoplot(gfit)
acf(gfit)
#####
skewness(gfit)
kurtosis(gfit)
#####
jarque.bera.test(gfit) ####Rejecting normality

##########

Box.test(gfit,lag=15,type= "Ljung-Box") #not rehjecting white noise

#########
```

```
######################fgarch

library(fGarch)
fgfit = garchFit(~ arma(0,1)+garch(1,1), data = logts,trace =F)
fgfit
Acf(fgfit@residuals)
Acf(fgfit@residuals^2)


f = predict(fgfit, n.ahead=10)
f


source("BacktestGarch.R")
nTest = 0.80*length(logts)
testLen = floor(length(Ethlogts) * .98)

backtestGarch(gFit, Ethlogts, testLen, 1)
```

```
############ Multiple Datasets#############
library(astsa)
Eth = read.csv('C:/Users/patil/OneDrive/Documents/425-Project/ETH_USD_2020-02-03_2021-02-02-CoinDesk.csv')
tether = read.csv('C:/Users/patil/OneDrive/Documents/425-Project/CoinDesk2.csv')
Btc = read.csv('C:/Users/patil/OneDrive/Documents/425-Project/Bitcoincash.csv')
Ltc = read.csv('C:/Users/patil/OneDrive/Documents/425-Project/Litecoin.csv')



Eth$Date =  as.Date(Eth$Date, format="%Y-%m-%d")
tether$Date = as.Date(tether$Date, format="%Y-%m-%d")
Btc$Date = as.Date(Btc$Date,format ="%m/%d/%Y" )
Ltc$Date = as.Date(Ltc$Date, format="%m/%d/%Y")



Ethts =  ts(Eth$X24h.High..USD., start = c(2020,32), frequency = 365.25)
Ethlogts = diff(log(Ethts))
TethtS = ts(tether$X24h.High..USD., start = c(2020,32), frequency = 365.25)
Tetlogts = diff(log(TethtS))
btchtS = ts(Btc$X24h.High..USD., start = c(2020,32), frequency = 365.25)
btclogts = diff(log(btchtS))
ltctS = ts(Ltc$X24h.High..USD., start = c(2020,32), frequency = 365.25)
ltclogts = diff(log(ltctS))

#eth and teth
lag2.plot(Ethlogts, Tetlogts,8)
ccf(Ethlogts,Tetlogts,main = "Ethereum with Tether")

#eth and btc
lag2.plot(Ethlogts, btclogts,8)
ccf(Ethlogts,btclogts,main = "Ethereum with Bitcoin Cash")

#eth and lite

lag2.plot(Ethlogts, ltclogts,8)
ccf(Ethlogts,ltclogts,main = "Ethereum with LiteCoin")

#lite and bitconcash
lag2.plot(btclogts, ltclogts,8)
ccf(btclogts,ltclogts,main = "LiteCoin and Bitcoin cash")

#litecoin and tether
lag2.plot(Tetlogts, ltclogts,8)
ccf(Tetlogts,ltclogts,main = "Litecoin and tether")
```

```
#litecoin and tether
lag2.plot(Tetlogts, ltclogts,8)
ccf(Tetlogts,ltclogts,main = "Litecoin and tether")

#bitcoin and tether
lag2.plot(btclogts, Tetlogts,8)
ccf(btclogts,Tetlogts,main = "Bitcoin cash and tether")


############VAR
library(vars)


s = VARselect(cbind(btclogts,ltclogts,Ethlogts), lag.max=5, type="const")
s

fit = VAR(cbind(btclogts,ltclogts,Ethlogts), p=3, type="const")
fit
serial.test(fit, lags.pt=10, type="PT.asymptotic")
coeftest(fit)
######################d############
autoplot(forecast(fit, h=15))
```

Vaidehi Madhu

```
library(ggplot2)
library(ggfortify)
library(lmtest)
library(fpp2)
library(tseries)
library(fBasics)
library(zoo)
library(forecast)
library(fUnitRoots)
library(fGarch)
library(tseries)
library(rugarch)
library(svs)
library(astsa)
library(TSA)
source("eacf.R")
source("backtest.R")


df = read.csv("CoinDesk2.csv")
head(df)
tail(df)
colnames(df) = c("Currency", "Date", "Closing", "Open", "High", "Low")
head(df)

df$Date = as.Date(df$Date, format="%Y-%m-%d")

class(df$Date)
head(df)


TS = ts(df$High, start = c(2018,6), frequency = 365.25)
autoplot(TS)   #Multiplicative series, no trend or seasonality
autoplot(log(TS))
basicStats(TS)
Acf(TS, lag.max = 20)
pacf(TS, lag.max = 20)
eacf(TS)
lag.plot(TS, do.lines = FALSE) #non stationary

qqnorm(TS)
Box.test(TS, lag = 20, type="Ljung-Box")

adfTest(TS, lags=20, type='ct')
kpss.test(TS, null= c('Level', 'Trend'))
```

```
difflog = diff(log(TS))
autoplot(difflog)
basicStats(difflog)
Acf(difflog, lag.max = 20)
pacf(difflog, lag.max = 20)
eacf(difflog)
lag.plot(TS, do.lines = FALSE)

adfTest(difflog)
kpss.test(difflog)

jarque.bera.test(TS)  #not normally distributed


# Model 1

fit1 = Arima(difflog, order = c(0,0,3))
fit1
coeftest(fit1)

#Residual Analysis

plot(fit1$residuals)
plot(fit1$residuals^2)
Acf(fit1$residuals, lag=20)
Acf(fit1$residuals^2, lag=20)
Box.test(fit1$residuals^2, lag=10, type="Ljung-Box")
```

```r
# Model 2

fit2 = Arima(difflog, order = c(1,0,3))
fit2
coeftest(fit2)

#Residual Analysis

plot(fit2$residuals)
plot(fit2$residuals^2)
Acf(fit2$residuals^2, lag=20)
Box.test(fit2$residuals^2, lag=10, type="Ljung-Box


# Garch Model

gFit = garchFit(~garch(1,1), data = difflog, trace
summary(gFit)
gFit

TS_Garch = ts(residuals(gFit, standardize=T))
jarque.bera.test(TS_Garch)
Box.test(TS_Garch, lag = 20, type="Ljung-Box")

Acf(TS_Garch, lag.max = 20)
Acf(TS_Garch^2, lag.max = 20)

jarque.bera.test(TS_Garch^2)
Box.test(TS_Garch^2, lag = 20, type="Ljung-Box")
Acf(TS_Garch^2)

plot(residuals(gFit), type="l")
lines(gFit@h.t, col="green")

# Backtesting and forecasting

n = length(difflog)
b1 = backtest(fit1, difflog, h=1, orig=.7*n)

h = forecast(fit1, h=5)
plot(forecast(fit1, h=5))
h$mean

f = predict(gFit, n.ahead=5)
plot(forecast(gFit, f=5))
f

p = 0.7*length(difflog)
testLengh = floor(length(difflog)*.98)

backtestGarch(gFit, difflog, testLengh,1)
```

Pramathesh Shukla

```r
setwd("C:/Users/PBS/Desktop/DSC 425")

library(tidyverse)
library(dplyr)
library(ggplot2)
library(ggfortify)
library(zoo)
library(fBasics)
library(xts)
library(lubridate)
library(Hmisc)
library(fUnitRoots)
library(tidyquant)
library(fUnitRoots)

source("Backtest.R")
source("eacf.R")
source("backtestGarch.R")

library(dynlm)
library(fGarch)
library(forecast)
library(fGarch)
library(rugarch)
library(tseries)
library(lmtest)


data = read.csv("BCH_USD_2020-02-04_2021-02-03-CoinDesk (1).csv")
head(data)

data$Date = as.Date(data$Date,format="%m/%d/%Y")

dataOrig = ts(data$X24h.Open..USD., start = c(2020, 02), frequency = 365.25)
autoplot(dataOrig)
```

```r
#below both tests say that the series is non-stationary, therefore we have to try to conve
adf.test(dataOrig)
kpss.test(dataOrig)

kpss.test(lag(dataOrig))

diff1 = diff(dataOrig)

#still the series is non-stationary
adf.test(diff1)
kpss.test(diff1)

#after taking diff 2 times, the series becomes stationary and useful for further analysis
diff2 = diff(diff1)
kpss.test(diff2)
adf.test(diff2)

dataTS = diff2


#There are multiple significant lag values. Maybe have to use GARCH
Acf(dataTS, lag.max = 20)

pacf(dataTS, lag.max = 20)

#by looking at eacf, it looks like there might be a (1, 2) ARMA model
eacf(dataTS)


autoarima = auto.arima(dataTS)
autoarima
coeftest(autoarima)

#looking at the coeftest ma1 has good significance.
#but ar1 has bad significance
```

```r
arima1 = Arima(dataTS, order = c(1,0,2))
arima1
coeftest(arima1)

arima2 = Arima(dataTS, order = c(0,0,1)) #I think this is the best model to fit.
arima2
coeftest(arima2)

arima3 = Arima(dataTS, order = c(1,0,1))
arima3
coeftest(arima3)

arima4 = Arima(dataTS, order = c(1,0,3))
arima4
coeftest(arima1)

arima4 = Arima(dataTS, order = c(0,0,2))
arima4
coeftest(arima4)


b1 = backtest(arima2, dataTS, h=1, orig = 0.80*length(dataTS))
b1


########arch efftects
Acf(arima2$residuals,lag.max=15)
Acf(arima2$residuals^2,lag.max=15)
###############Arch model
gfit = garchFit(~ arma(0,1) + garch(1,1), data = dataTS, trace = F)
gfit

predict = predict(gfit,n.ahead =5)
predict


nTest = 0.80*length(diffts)
testLen = floor(length(dataTS) * .95)

backtestGarch(gfit, dataTS, testLen, 1)
```

Ashay Kargaonkar

```
setwd("D:/_College/6th Quarter/DSC 465/Project/Milestone 4")

library(tidyverse)
library(dplyr)
library(ggplot2)
library(ggfortify)
library(zoo)
library(fBasics)
library(xts)
library(lubridate)
library(Hmisc)
library(fUnitRoots)
library(tidyquant)
library(fUnitRoots)

source("Backtest.R")
source("eacf.R")
source("backtestGarch.R")

library(dynlm)
library(fGarch)
library(forecast)
library(fGarch)
library(rugarch)
library(tseries)
library(lmtest)


data = read.csv("LTC_USD_2020-02-16_2021-02-15-CoinDesk.csv")
head(data)
tail(data)


data$Date = as.Date(data$Date,format="%m/%d/%Y")

dataOrig = ts(data$X24h.High..USD., start = c(2020, 02), frequency = 365.25)
autoplot(dataOrig)



#below both tests say that the series is non-stationary, therefore we have to try to convert it to stationary.
adf.test(dataOrig)
kpss.test(dataOrig)

kpss.test(lag(dataOrig))

diff1 = diff(dataOrig)

#still the series is non-staionary
adf.test(diff1)
kpss.test(diff1)
```

```
#after taking diff 2 times, the series becomes stationary and useful for further analysis
diff2 = diff(diff1)
kpss.test(diff2)
adf.test(diff2)

dataTS = diff2


autoplot(dataTS)

#There are multiple significant lag values. Maybe have to use GARCH
Acf(dataTS, lag.max = 20)

pacf(dataTS, lag.max = 20)

#by looking at eacf, it looks like there might be a (1, 2) ARMA model
eacf(dataTS)


autoarima = auto.arima(dataTS)
autoarima
coeftest(autoarima)

#looking at the coeftest ar1, ar2 and ma1 have high significant values:
#good estimate std values, less p value, error is also less.
#less sigma-square estimate value which is also good.

#AUTOARIMA gives ARIMA model of order (2, 0, 1) as the best fit.


#trying different orders of arima

arima1 = Arima(dataTS, order = c(2,0,1))  #best model, same as the autoarima
arima1
coeftest(arima1)

arima2 = Arima(dataTS, order = c(2,0,2))
arima2
coeftest(arima2)

#ma2 order is not significant

arima3 = Arima(dataTS, order = c(1,0,1))
arima3
coeftest(arima3)

arima4 = Arima(dataTS, order = c(3,0,1))
arima4
coeftest(arima1)
```

```
#by doing some manual testing we can say that model suggested by auto-arima model is the be


b1 = backtest(autoarima, dataTS, h=1, orig = 0.9*length(dataTS))
b1


########arch efftects
Acf(arima1$residuals,lag.max=15)
Acf(arima1$residuals^2,lag.max=15)
###############Arch model
gfit = garchFit(~ arma(2,2) + garch(1,1), data = dataTS, trace = F)
gfit


predict = predict(gfit,n.ahead =5)
predict


testLen = floor(length(dataTS) * .95)

bgarch = backtestGarch(gfit, dataTS, testLen, 1)
bgarch

barima = backtest(autoarima, dataTS, testLen, 1)
barima
```