Hugo Joncour - 40139130
Zié Coulibaly - 40046155

ECON 422: Econometrics 2

Machine Learning and Economics

Project:

Can we predict Tether's market cap evolution using other cryptocurrencies's demand ?

Index:

[1] Research Proposal is mostly composed of the first assignment
[2] First Results is for the most part the content of the second assignment

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

1. Introduction

a) Cryptocurrencies

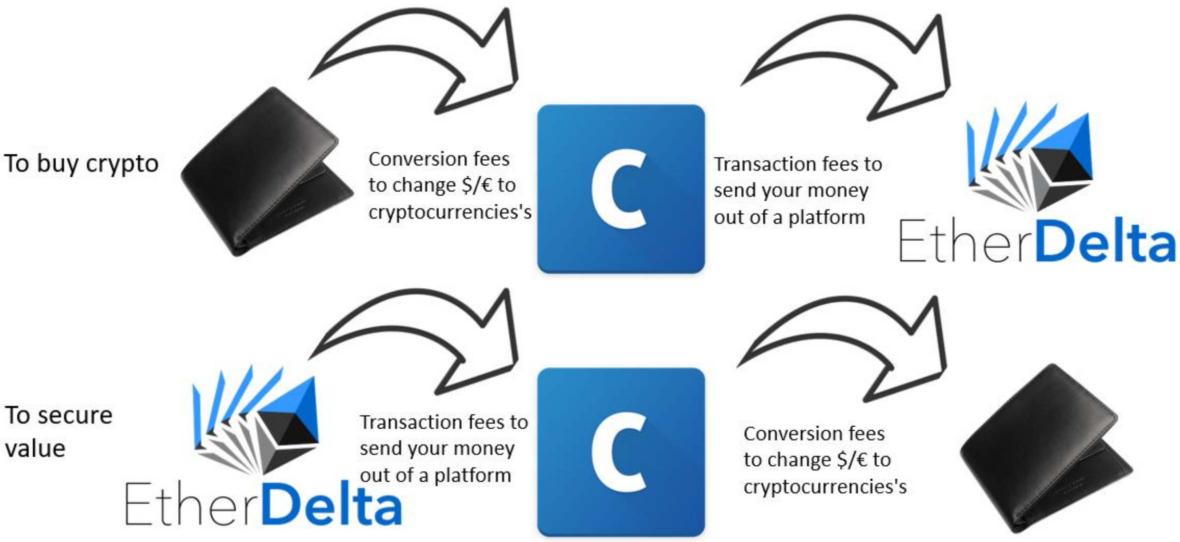
Cryptocurrencies are digital assets used as currencies for online transactions, using technology to secure money creation, money movement, limit money emission, or add features such as smart contracts. Their price is determined only by offer, demand and trust that people have in it; thus, their value is very volatile, which makes them both a digital currency that can be used for shopping (if accepted) and a very speculative asset that can be traded against any other cryptocurrency on online platforms.

Every unit of cryptocurrency is assigned to a wallet, an address composed of many letters and numbers. Many online exchanges offer people to trade one crypto against another, but few accept normal currencies such as USD or Euros to buy cryptos, so in order to buy any cryptocurrency, you first have to buy it on one platform that accept regulated currencies, then send it to a wallet on an exchange platform where you can trade it, which causes transaction costs.

Anyone wanting to trade cryptos must go through this process:

1) Create an account on a platform that sells cryptocurrencies against real currencies[3]
2) Create an account on a platform where you can trade cryptocurrencies
3) Send what you bought from the first platform to the second one, which causes transaction costs[4]

And the process is the same in the other direction for anyone who would like to exchange his cryptocurrency to get real money: send the crypto from the trading platform to the conversion platform (which causes transaction costs) and convert it back (which causes conversion costs).



[3] Few platforms accept to sell cryptocurrencies against real money (called fiat). Most platform accepting fiat make money on conversion rate euro/bitcoin, usd/btc while exchange platforms, where people trade cryptocurrencies make money on transaction fees

[4] Every time money is moved from a wallet to another, a transaction fee must be paid to secure the transaction. A computer will record the movement through a secured process and add the amount, the origin, the destination, the date and the time to a public register that will make sure this transaction can't be undone.

The process isn't immediate and even if the amount of bitcoin sent from one platform to another doesn't change, the value in dollar of the asset may have changed by the time it is finally converted to dollars.

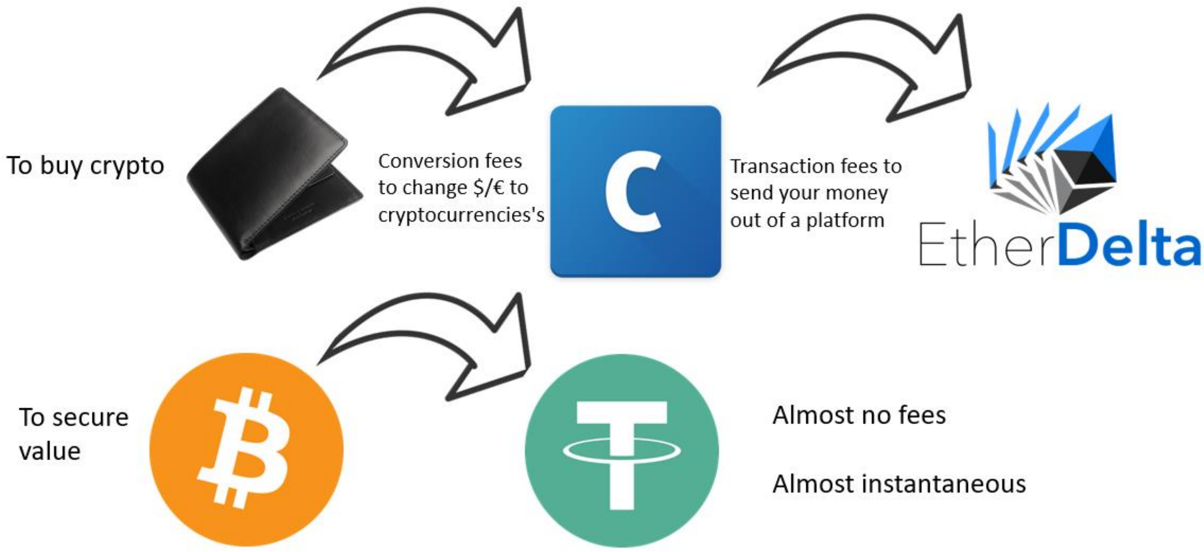
These intermediate steps make it long and costly to exchange a cryptocurrency asset against a safe and stable value such as a normal currency, so a platform thought about a way that would make it possible for people to move much faster from a "speculative" asset to a stable value asset and opposite: Tether

b) The Tether

The Tether[5] (USDT) is a cryptocurrency created by Bitfinex, one of the biggest cryptocurrency trading platform, that (supposedly) has every unit of tether backed by its original currency (for the USDT, the original currency is the US Dollar), and guarantees a fixed rate of:

$$1 \text{ USDT} = 1\text{USD}$$

This not only makes it much faster to secure an investment, but also avoid both conversion and transaction costs. The Tether is called a "stable-coin" (its value is stable), and thus the ultimate safe-haven when people feel like other cryptocurrencies's values are about to go down[6].



Bitfinex can create at any moment any amount of Tether they want, and sell it against any unit of cryptocurrency for their price in United States Dollar.

[5] We will be talking about the USDT for "United State Dollar Tether", the full name of this asset

[6] The Tether is not the only stable coin, but its market cap and volume traded everyday are respectively 10 times and 100 times more important than the second biggest stable coin, so I didn't judge necessary to make it part of the research proposal

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

2. Research Proposal

a) The Data

We built a scraping program in java that can scrape Coinmarketcap.com and a parser in Python that can reorganize it in a CSV containing for each asset: the day, the price at the opening, the price at the closure, the highest price, the lowest price, the volume traded and the market cap:

Date (character)	OpensAt (double)	ClosesAt (double)	HighestAt (double)	LowestAt (double)	VolumeTraded (double)	MarketCap (double)
2020-5-2	1.92	2.14	2.22	1.91	111859155	1486335955
2020-4-2	2.00	1.92	2.03	1.89	88198795	1336199969
2020-3-2	1.91	2.00	2.10	1.87	112154610	1391386055
2020-2-2	1.74	1.91	1.98	1.70	83037458	1326090827
2020-1-2	1.66	1.74	1.74	1.65	37226254	1209947862
2020-31-1	1.68	1.66	1.72	1.65	49262411	1151165578

10_Tezos.csv - Bloc-notes

Fichier Edition Format Affichage Aide

```
Date,OpensAt,ClosesAt,HighestAt,LowestAt,VolumeTraded,MarketCap
2020-5-2,1.92,2.14,2.22,1.91,111859155,1486335955
2020-4-2,2.00,1.92,2.03,1.89,88198795,1336199969
2020-3-2,1.91,2.00,2.10,1.87,112154610,1391386055
2020-2-2,1.74,1.91,1.98,1.70,83037458,1326090827
2020-1-2,1.66,1.74,1.74,1.65,37226254,1209947862
2020-31-1,1.68,1.66,1.72,1.65,49262411,1151165578
2020-30-1,1.57,1.68,1.70,1.56,54565714,1164729726
2020-29-1,1.55,1.57,1.68,1.54,51375642,1088132652
```

b) Method

We know that the Tether is a safe haven for anyone who wants to freeze or secure the value of his investment in a cryptocurrency. If someone exchanges an asset for something stable, he expects the value of the asset going down.

We can assume if the demand in Tether goes up, the market is going down and if the demand in Tether goes down, people are selling their Tethers for riskier assets. Since the price of the Tether is 1USD$ we can only rely on its market cap to know if its demand goes up or down, however for other cryptocurrencies we can look at their price, market cap and volume traded to know how the market evolves.

Using this data and what we know about the Tether we want to predict the evolution of the Tether's Market Cap: If we can predict the Tether's Market Cap given the trend of other cryptocurrencies we will know whether the Tether is bullish or bearish and from this we will know if we should buy or sell.

c) Why ?

We think this question may be more than just a way to predict markets on this domain but a general behavior about risk.

We think this this topic is interesting because even though cryptocurrencies are unregulated and for many of them heavily influenced by big actors, they follow human behaviors that most of them replicated in regulated market finance and other domains:

We will learn through this project if we can predict Tether's value, but we may also learn about people's behavior toward a "safe-value" which is more than just a way to predict markets on this domain but a general behavior about risk aversion.

This topic is also interesting because the US Justice Department is investigating Bitfinex for illegally manipulating the price of Bitcoin using the Tether, and because Bitfinex failed to provide an audit showing reserves correctly backing every Tether "printed" or emitted, meaning that this cryptocurrency could be the trigger of a crash if Bitfinex failed at some point to guarantee the fixed rate.

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

3. First Results

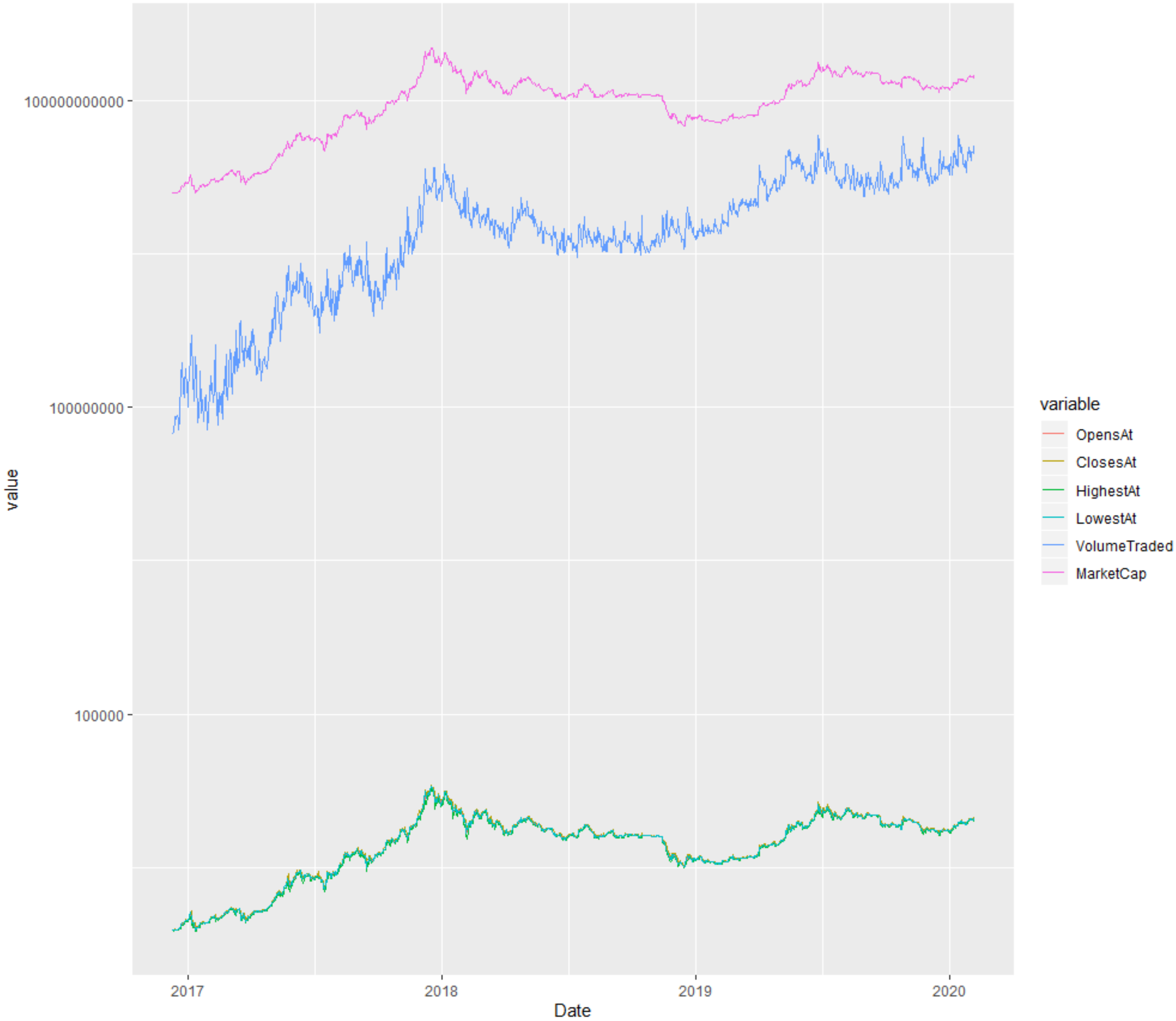
a) Observations

For this part we only used a sample of all the data we could gather to plot and compare the different trends through time. These graphs allow us to observe correlation between the different variables and the currencies themselves.

To plot these graphs, we used a logarithmic scale on the y axis for clarity purpose and limited the time windows from 2017 to January 2020 (included) so we could compare the most important cryptocurrencies[7]:

- Bitcoin (BTC):

The Bitcoin is the biggest and most important cryptocurrency by most metrics. It is worth 63% of the global market cap of all cryptocurrencies.

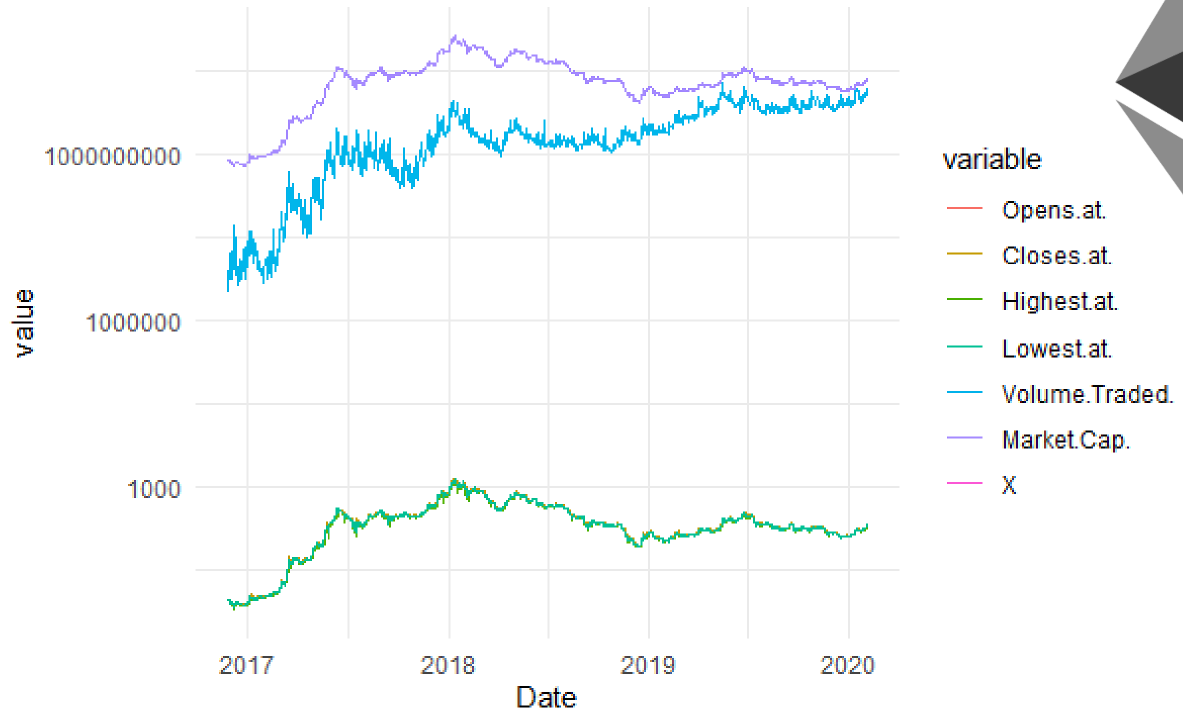


[7] many of the 30 biggest were created after 2018, this is why we only used a few ones

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

- Ethereum (ETH):

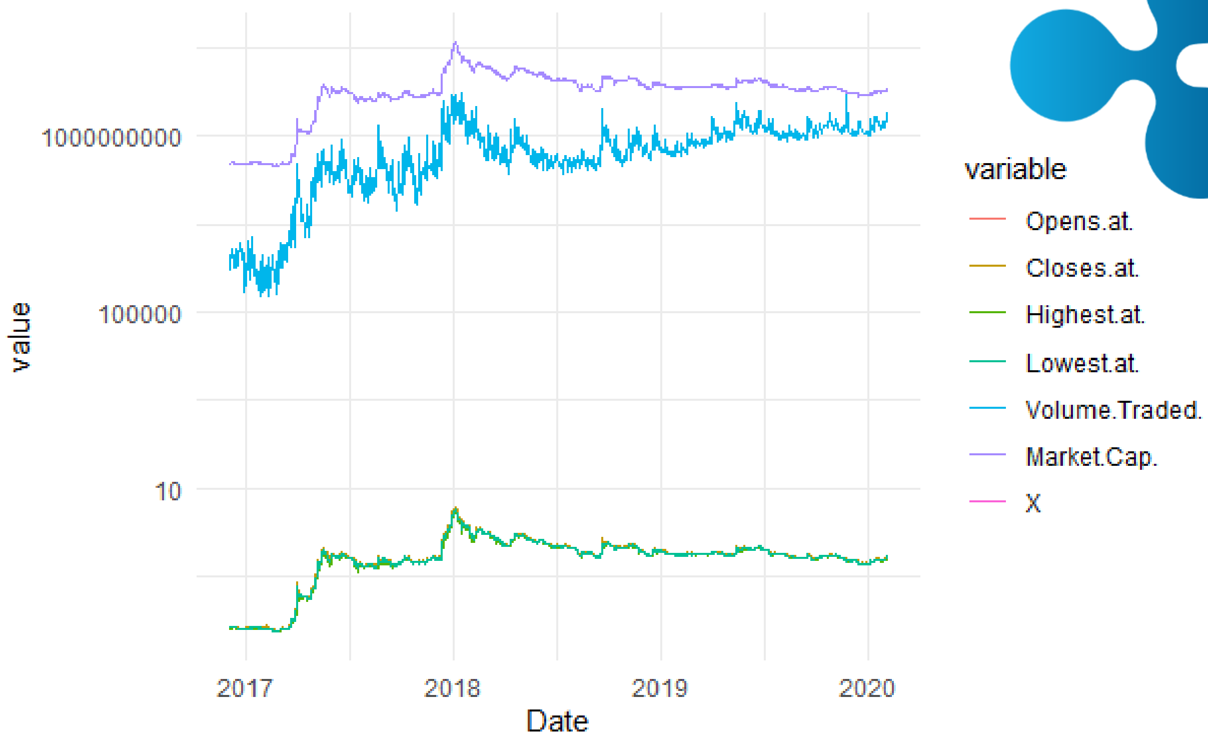
Ethereum is the second biggest cryptocurrency, similar in many aspects to the Bitcoin it differs mostly by the possibility to program smart-contracts (example: allowing transactions without third party).



- Ripple (XRP):

Ripple was created to facilitate financial settlement and money transfers through a cryptocurrency protocol providing security through encryption, cost efficiency and no third-party transaction fee.

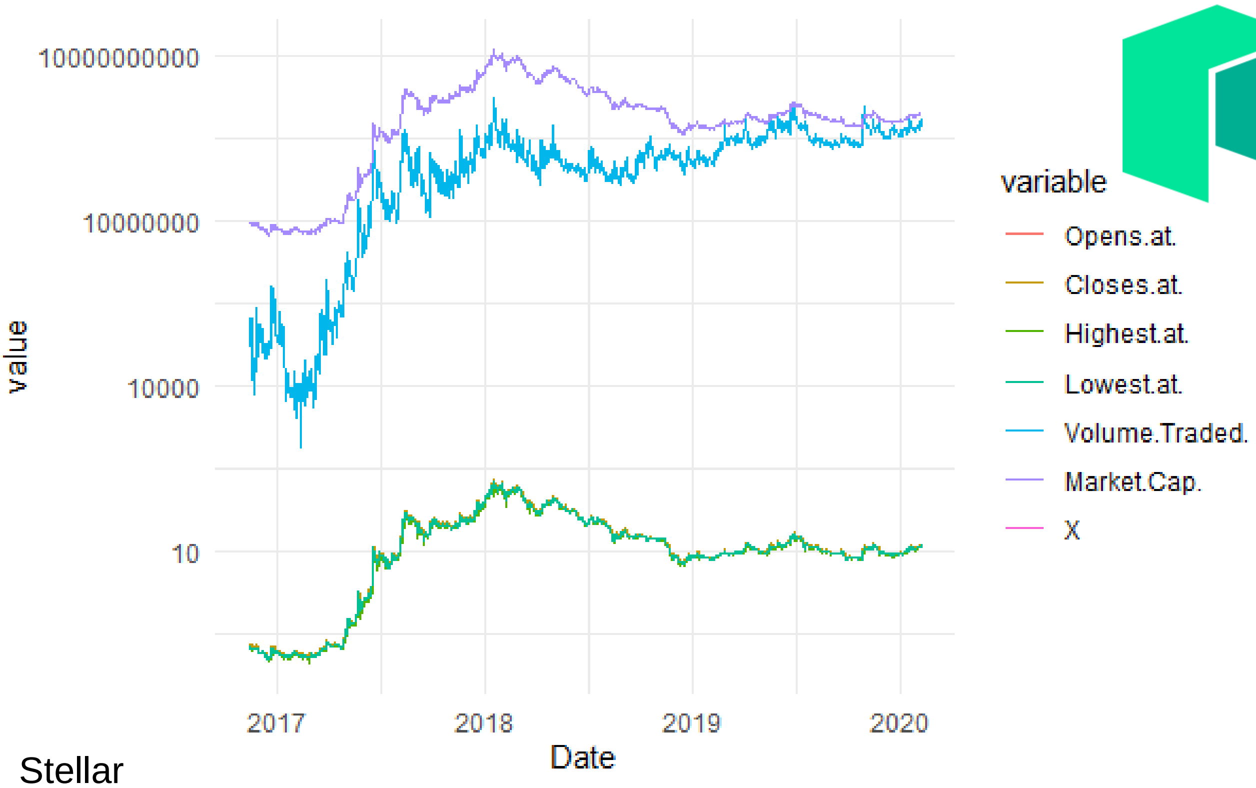
It is used by many financial institutions for real-time gross settlement, and unlike most cryptocurrencies, has no limit to the number of units that can be emitted.



Hugo Joncour - 40139130
Zié Coulibaly - 40046155

- NEO (NEO):

Regulator friendly cryptocurrency created in China. Deploys smart contract applications and helps manage digitized assets.



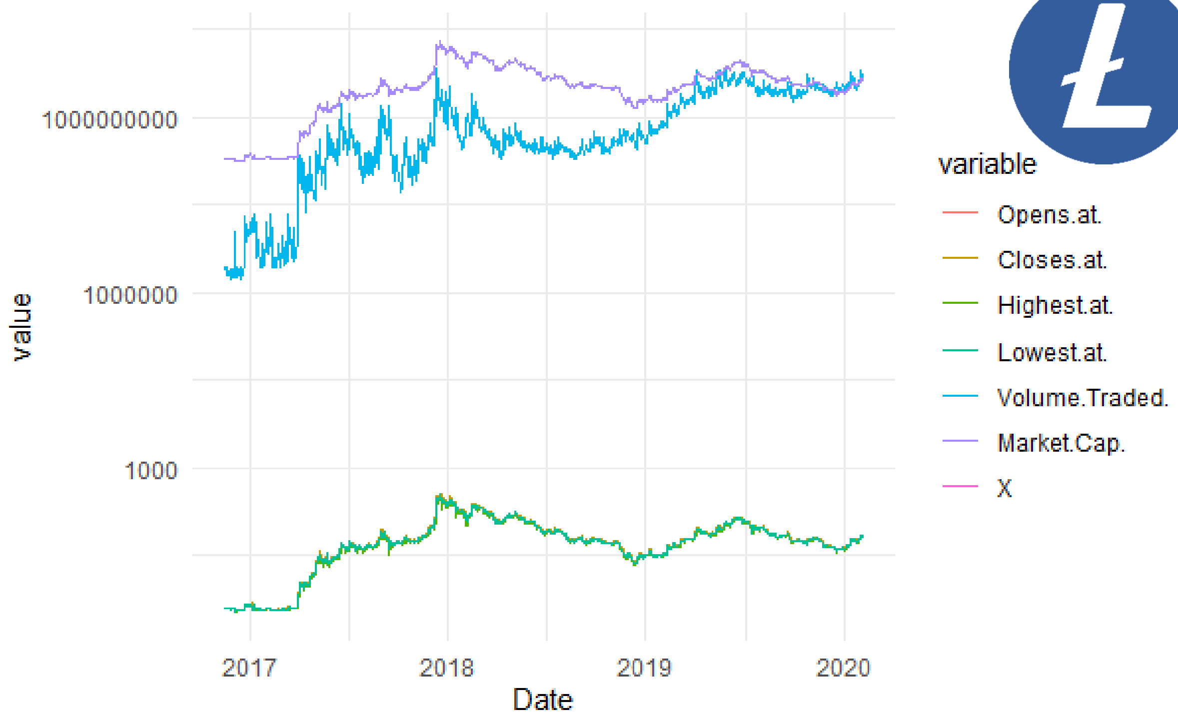
Stellar

Cryptocurrency used for a protocol allowing cross border transactions between real world currencies. Most notably used by Deloitte for its Deloitte Digital Bank and IBM

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

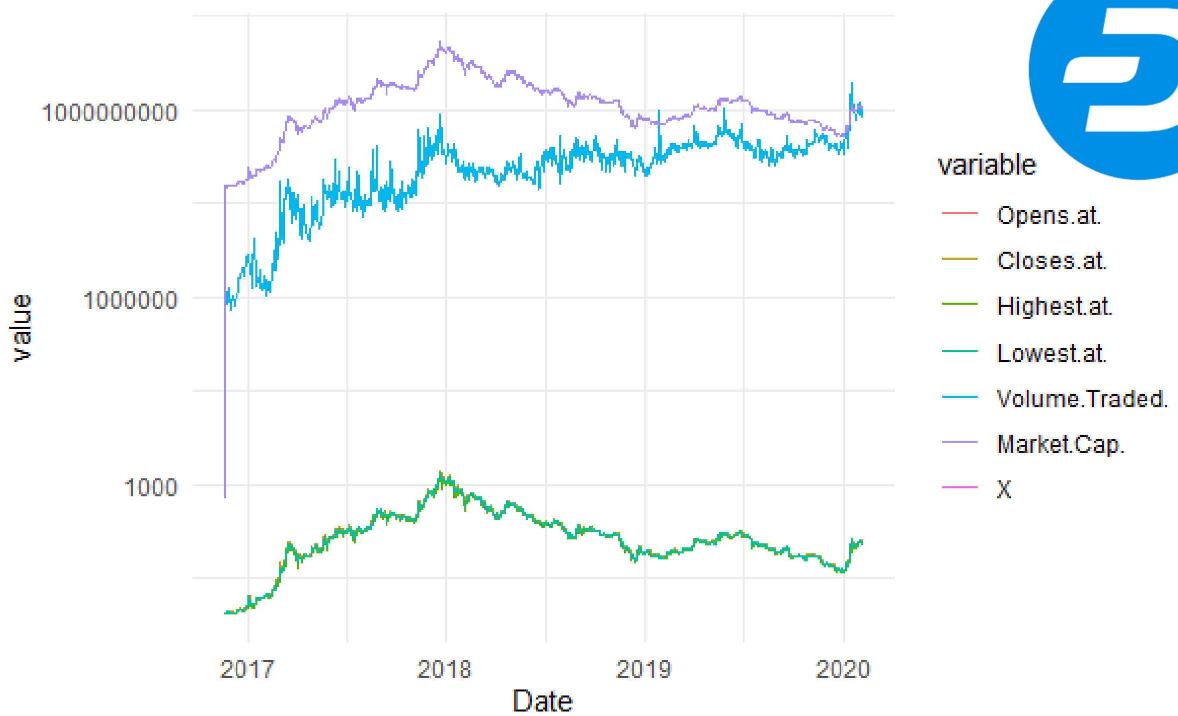
- Litecoin (LTC):

Litecoin (LTC) is very similar to the Bitcoin but confirms transactions must faster because of a different system of encryption



- Dash (DASH):

Started as a copy of the Bitcoin, it is most notably different from other cryptocurrencies because transactions in Dash are untraceable. It became the most popular cryptocurrency in Venezuela after the Bolivar's value collapsed.



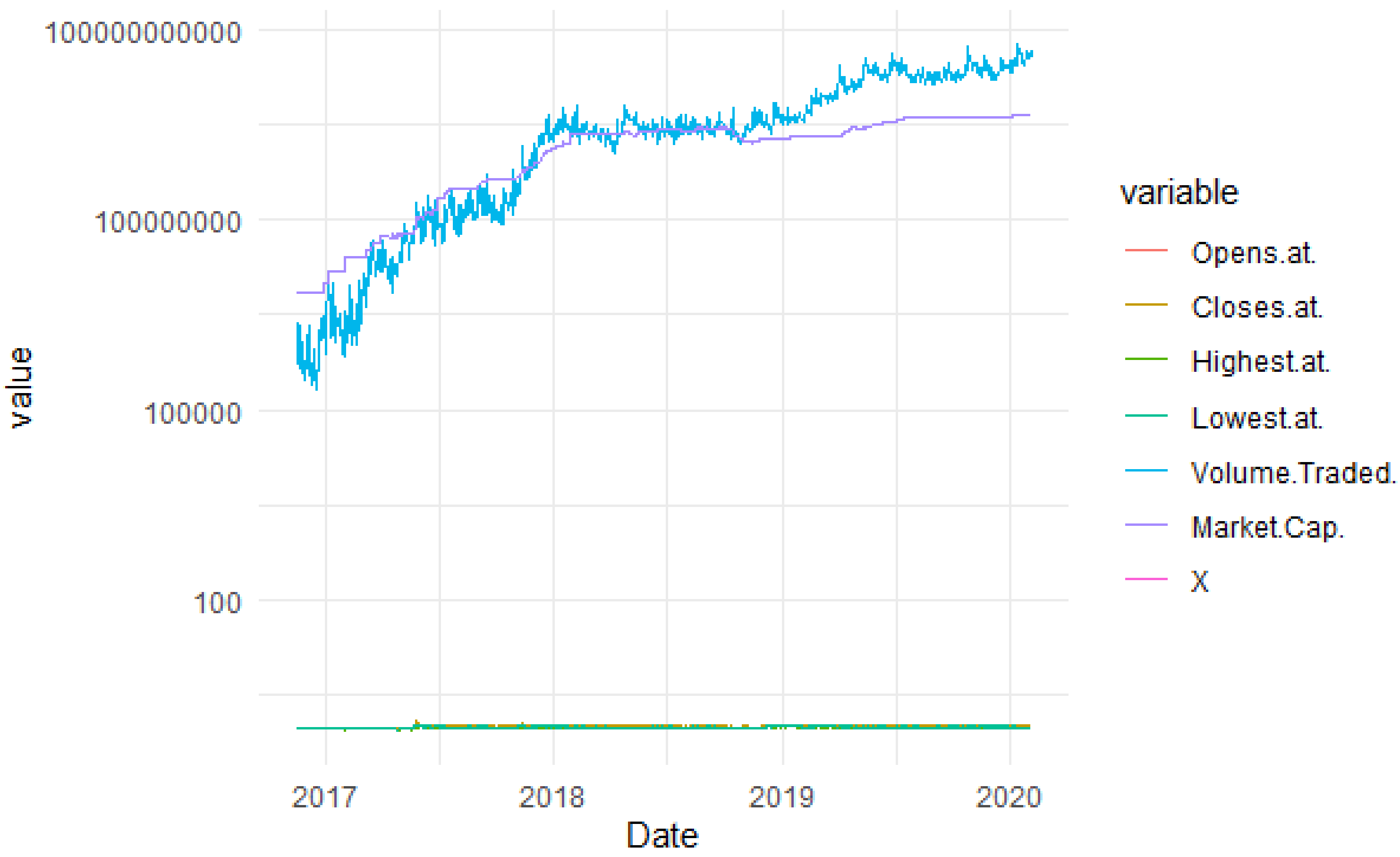
Hugo Joncour - 40139130
Zié Coulibaly - 40046155

- Tether (USDT):

The Tether is our variable of interest. As explained early, it has parity with the dollar, every Tether emitted is supposed to be backed by a dollar and can be exchanged at any time for a 1$USD on the Bitfinex platform.

What makes it special is that since its price can't go down, its market cap can only go down if people exchange it against a dollar (a Tether exchanged for a dollar is deleted), and it can only go up if people buy newly printed Tether (or if Bitfinex buys assets with the Tether they print like a Central Bank doing Quantitative Easing).

As we can see on this graph, its volume traded is by far the highest of all cryptocurrencies, and we can also observe that unlike other cryptocurrencies its demand is much less affected by exogenous shocks: even when other cryptocurrencies's market cap goes down (because of the price going down), Tether demand goes up or doesn't change because people want to secure the value of their assets which confirms the hypothesis we had in the previous part.



Hugo Joncour - 40139130
Zié Coulibaly - 40046155

b) First Conclusions

Cryptocurrencies can be created with a specific purpose in mind but as we can see on the graphs, their prices, volume traded, and market cap have the same spikes at the same events: they seem to be equally affected by exogenous shocks.

We used covariance matrices to observe this:

Market Cap:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	2.384102e+21	6.410551e+20	3.354263e+20	2.953990e+20	1.266103e+19	1.134423e+20
[2,]	6.410551e+20	5.899837e+20	2.837552e+20	2.085586e+20	-8.731644e+18	6.326402e+19
[3,]	3.354263e+20	2.837552e+20	2.125945e+20	1.110426e+20	-4.232475e+18	3.377801e+19
[4,]	2.953990e+20	2.085586e+20	1.110426e+20	9.566095e+19	-4.390558e+18	2.539125e+19
[5,]	1.266103e+19	-8.731644e+18	-4.232475e+18	-4.390558e+18	1.196380e+18	-5.632953e+17
[6,]	1.134423e+20	6.326402e+19	3.377801e+19	2.539125e+19	-5.632953e+17	9.733276e+18
[7,]	3.354481e+19	4.110833e+19	1.920425e+19	1.326382e+19	1.180160e+17	4.468126e+18
[8,]	2.047979e+19	-7.719876e+18	-2.792810e+18	-3.962501e+18	1.053517e+18	2.121627e+17
[9,]	3.498544e+19	4.849201e+19	2.914468e+19	1.650922e+19	-7.398377e+17	4.748859e+18
[10,]	2.535342e+19	2.621885e+19	1.787643e+19	9.102720e+18	-1.703787e+17	2.973410e+18
[11,]	4.656583e+19	5.071007e+19	2.427180e+19	1.689448e+19	-8.050920e+17	5.220948e+18
[12,]	5.819478e+19	4.142639e+19	2.247455e+19	1.853952e+19	-1.081557e+18	5.197601e+18

	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]
[1,]	3.354481e+19	2.047979e+19	3.498544e+19	2.535342e+19	4.656583e+19	5.819478e+19
[2,]	4.110833e+19	-7.719876e+18	4.849201e+19	2.621885e+19	5.071007e+19	4.142639e+19
[3,]	1.920425e+19	-2.792810e+18	2.914468e+19	1.787643e+19	2.427180e+19	2.247455e+19
[4,]	1.326382e+19	-3.962501e+18	1.650922e+19	9.102720e+18	1.689448e+19	1.853952e+19
[5,]	1.180160e+17	1.053517e+18	-7.398377e+17	-1.703787e+17	-8.050920e+17	-1.081557e+18
[6,]	4.468126e+18	2.121627e+17	4.748859e+18	2.973410e+18	5.220948e+18	5.197601e+18
[7,]	7.834798e+18	4.175219e+17	4.704941e+18	3.053226e+18	3.152210e+18	1.872131e+18
[8,]	4.175219e+17	1.670466e+18	-8.256872e+17	2.998361e+16	-8.618919e+17	-9.348365e+17
[9,]	4.704941e+18	-8.256872e+17	5.657571e+18	2.860813e+18	4.278559e+18	3.147959e+18
[10,]	3.053226e+18	2.998361e+16	2.860813e+18	2.169485e+18	2.123465e+18	1.650647e+18
[11,]	3.152210e+18	-8.618919e+17	4.278559e+18	2.123465e+18	4.731930e+18	3.473734e+18
[12,]	1.872131e+18	-9.348365e+17	3.147959e+18	1.650647e+18	3.473734e+18	3.872977e+18

On 144 observations, 28 are correlated

Volume Traded:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	7.065249e+19	2.727197e+19	5.605474e+18	6.884347e+18	8.419076e+19	1.175717e+19
[2,]	2.727197e+19	1.157185e+19	2.148614e+18	2.548551e+18	3.440961e+19	4.801735e+18
[3,]	5.605474e+18	2.148614e+18	1.184311e+18	5.989699e+17	5.074413e+18	8.328215e+17
[4,]	6.884347e+18	2.548551e+18	5.989699e+17	1.203901e+18	7.750228e+18	1.109931e+18
[5,]	8.419076e+19	3.440961e+19	5.074413e+18	7.750228e+18	1.147136e+20	1.462879e+19
[6,]	1.175717e+19	4.801735e+18	8.328215e+17	1.109931e+18	1.462879e+19	2.333193e+18
[7,]	6.921795e+18	2.978955e+18	5.128867e+17	6.506201e+17	8.982691e+18	1.231165e+18
[8,]	8.338763e+17	3.501490e+17	6.611941e+16	7.029641e+16	9.906209e+17	1.603951e+17
[9,]	9.306191e+17	3.792120e+17	1.189279e+17	9.370451e+16	9.833316e+17	1.610314e+17
[10,]	2.936515e+18	1.159380e+18	2.981960e+17	2.912081e+17	3.431081e+18	4.554602e+17
[11,]	1.264104e+18	5.148565e+17	1.265599e+17	1.146096e+17	1.446519e+18	2.132992e+17
[12,]	1.066638e+18	4.349768e+17	7.212786e+16	1.371825e+17	1.409108e+18	1.822314e+17

	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]
[1,]	6.921795e+18	8.338763e+17	9.306191e+17	2.936515e+18	1.264104e+18	1.066638e+18
[2,]	2.978955e+18	3.501490e+17	3.792120e+17	1.159380e+18	5.148565e+17	4.349768e+17
[3,]	5.128867e+17	6.611941e+16	1.189279e+17	2.981960e+17	1.265599e+17	7.212786e+16
[4,]	6.506201e+17	7.029641e+16	9.370451e+16	2.912081e+17	1.146096e+17	1.371825e+17
[5,]	8.982691e+18	9.906209e+17	9.833316e+17	3.431081e+18	1.446519e+18	1.409108e+18
[6,]	1.231165e+18	1.603951e+17	1.610314e+17	4.554602e+17	2.132992e+17	1.822314e+17
[7,]	1.072607e+18	9.445412e+16	1.018221e+17	3.097807e+17	1.371556e+17	1.254581e+17
[8,]	9.445412e+16	1.679515e+16	1.275584e+16	3.688852e+16	1.713980e+16	1.168910e+16
[9,]	1.018221e+17	1.275584e+16	2.349918e+16	4.387738e+16	2.025082e+16	1.514114e+16
[10,]	3.097807e+17	3.688852e+16	4.387738e+16	2.036202e+17	5.557382e+16	4.483645e+16
[11,]	1.371556e+17	1.713980e+16	2.025082e+16	5.557382e+16	3.713641e+16	1.842158e+16
[12,]	1.254581e+17	1.168910e+16	1.514114e+16	4.483645e+16	1.842158e+16	4.057914e+16

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

The 144 observations are all positively correlated. We can observe from this that demand in cryptocurrencies is exogenous and affects all of them at the same time in similar ways, however we cannot determine yet how much

We tested the Multilinear Regression Model on a limited sample using their volume traded and market cap:

```
Call:
lm(formula = X5_Tether$MarketCap[1:852] ~ bnb_vt + bcc_vt + btc_vt +
    dash_vt + eos_vt + eth_vt + ltc_vt + neo_vt + trx_vt + xlm_vt +
    xrp_vt + xtz_vt + bnb_mc + bcc_mc + btc_mc + dash_mc + eos_mc +
    eth_mc + ltc_mc + neo_mc + trx_mc + xlm_mc + xrp_mc + xtz_mc)

Residuals:
       Min         1Q     Median         3Q        Max
-1018198832  -185561366   35292130  216960097 1835945144

Coefficients:
                  Estimate       Std. Error t value              Pr(>|t|)
(Intercept) 1150826119.240497  55002733.764391  20.923 < 0.0000000000000002 ***
bnb_vt              -0.836247         0.231205  -3.617             0.000316 ***
bcc_vt              -0.100375         0.024604  -4.080   0.0000494777621656 ***
btc_vt               0.005754         0.007290   0.789             0.430127
dash_vt              0.578256         0.098400   5.877   0.0000000060696631 ***
eos_vt              -0.200295         0.033878  -5.912   0.0000000049336277 ***
eth_vt               0.144303         0.018691   7.720   0.0000000000000335 ***
ltc_vt              -0.082940         0.034230  -2.423             0.015607 *
neo_vt              -0.938465         0.128484  -7.304   0.0000000000006568 ***
trx_vt               0.495821         0.072424   6.846   0.0000000000147904 ***
xlm_vt              -0.557734         0.169072  -3.299             0.001013 **
xrp_vt               0.025602         0.021747   1.177             0.239431
xtz_vt               9.266279         1.306541   7.092   0.0000000000028345 ***
bnb_mc               0.215938         0.034523   6.255   0.0000000006373372 ***
bcc_mc               0.053982         0.008892   6.071   0.0000000019406290 ***
btc_mc               0.008762         0.000737  11.889 < 0.0000000000000002 ***
dash_mc             -0.708892         0.049094 -14.439 < 0.0000000000000002 ***
eos_mc               0.046847         0.015316   3.059             0.002295 **
eth_mc              -0.016050         0.003408  -4.710   0.0000029073587522 ***
ltc_mc               0.064160         0.015839   4.051   0.0000558570667524 ***
neo_mc               0.225566         0.033619   6.710   0.0000000000361958 ***
trx_mc              -0.135210         0.031969  -4.229   0.0000260437432899 ***
xlm_mc               0.041753         0.025396   1.644             0.100539
xrp_mc               0.006856         0.003634   1.887             0.059524 .
xtz_mc               0.117685         0.063185   1.863             0.062880 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 360000000 on 827 degrees of freedom
Multiple R-squared:  0.8928,	Adjusted R-squared:  0.8897
F-statistic: 286.9 on 24 and 827 DF,  p-value: < 0.00000000000000022
```

```
> cor(bnb_mc,bnb_vt, method="pearson" )
[1] 0.8422204
> cor(bcc_mc,bcc_vt, method="pearson" )
[1] 0.1578615
> cor(btc_mc,btc_vt, method="pearson" )
[1] 0.46154
> cor(dash_mc,dash_vt, method="pearson" )
[1] -0.1457533
> cor(eos_mc,eos_vt, method="pearson" )
[1] 0.1900146
> cor(eth_mc,eth_vt, method="pearson" )
[1] -0.2457992
> cor(ltc_mc,ltc_vt, method="pearson" )
[1] 0.06909263
> cor(neo_mc,neo_vt, method="pearson" )
[1] 0.03783648
> cor(trx_mc,trx_vt, method="pearson" )
[1] 0.3389256
> cor(xrp_mc,xrp_vt, method="pearson" )
[1] 0.4611258
> cor(xlm_mc,xlm_vt, method="pearson" )
[1] 0.02233195
> cor(xtz_mc,xtz_vt, method="pearson" )
[1] 0.4873249
> confint(model1, conf.level=0.95)
```

A 0.8928 may look like a very good R^2 but for some currencies the market cap and the volume traded are highly correlated to each other:

Adding to this that there are 24 explanatory variables we have good reasons to doubt the ability of this model to efficiently predict the market cap of the Tether:

it might simply be overfitting.

From what we saw in the previous part a regression using many variables, both market cap and volume traded would overfit so we decided to compare which of the two is the most correlated reducing the number of

variables, but on a longer window of time (not all data have an historic that long):

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

```
Call:
lm(formula = X5_Tether$MarketCap[1:919] ~ volt)

Residuals:
       Min         1Q     Median         3Q        Max 
-2.222e+09 -4.374e+08  5.075e+07  5.523e+08  1.681e+09 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.676e+09  4.319e+07  38.802  < 2e-16 ***
volt1        3.112e-02  1.135e-02   2.743  0.00622 ** 
volt2        2.642e-01  2.855e-02   9.253  < 2e-16 ***
volt3       -3.265e-01  2.862e-02 -11.408  < 2e-16 ***
volt4       -2.739e-01  3.272e-02  -8.372 2.32e-16 ***
volt5       -7.361e-03  4.681e-02  -0.157  0.87507    
volt6       -1.067e-01  4.276e-02  -2.496  0.01276 *  
volt7       -4.702e-01  3.234e-01  -1.454  0.14626    
volt8        5.993e-01  8.827e-02   6.789 2.11e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 680300000 on 851 degrees of freedom
  (59 observations deleted due to missingness)
Multiple R-squared:  0.6208,	Adjusted R-squared:  0.6172 
F-statistic: 174.1 on 8 and 851 DF,  p-value: < 2.2e-16
```

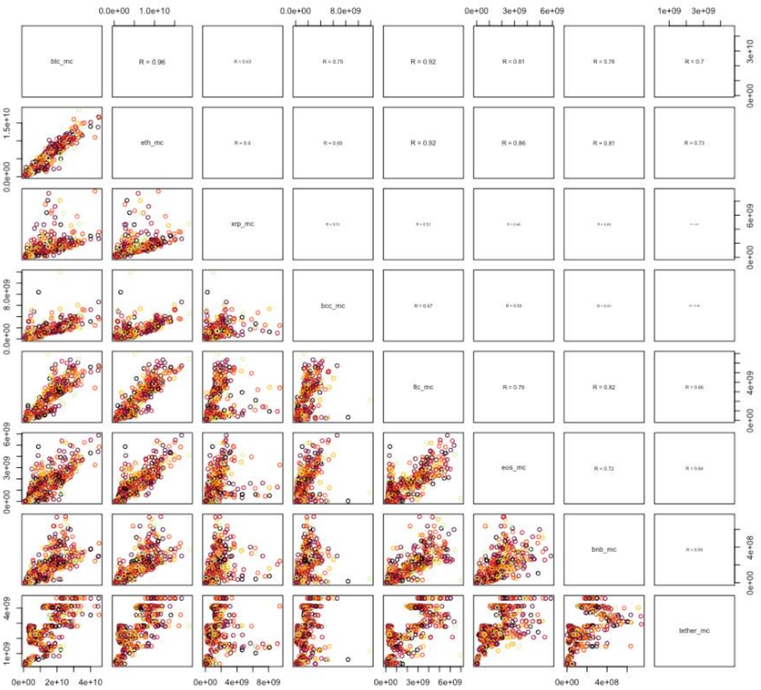
```
Call:
lm(formula = X5_Tether$MarketCap[1:919] ~ marketc)

Residuals:
       Min         1Q     Median         3Q        Max 
-2.747e+09 -2.926e+08  2.695e+06  3.424e+08  1.995e+09 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.790e+08  7.201e+07  10.818   <2e-16 ***
marketc1     1.591e-02  8.580e-04  18.542   <2e-16 ***
marketc2     6.605e-04  2.290e-03   0.288    0.773    
marketc3     9.133e-03  3.670e-03   2.489    0.013 *  
marketc4    -7.014e-02  6.345e-03 -11.056   <2e-16 ***
marketc5    -1.521e-01  1.673e-02  -9.092   <2e-16 ***
marketc6     1.179e-01  1.375e-02   8.578   <2e-16 ***
marketc7     2.802e-01  2.977e-02   9.415   <2e-16 ***
marketc8    -1.266e-02  3.674e-02  -0.345    0.730    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 584600000 on 851 degrees of freedom
  (59 observations deleted due to missingness)
Multiple R-squared:  0.7199,	Adjusted R-squared:  0.7173 
F-statistic: 273.4 on 8 and 851 DF,  p-value: < 2.2e-16
```

Color palet:

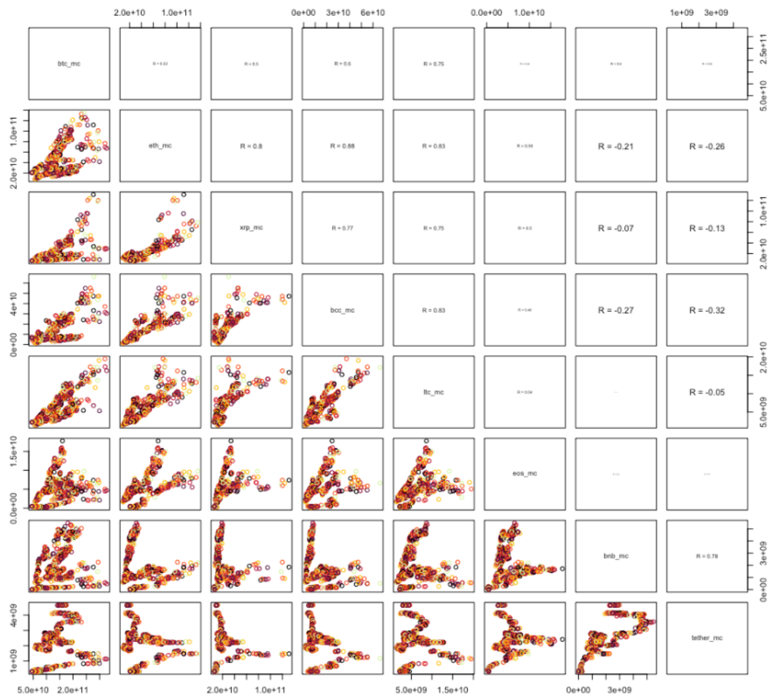
- Bitcoin	- Litecoin
- Ethereum	- EOS
- Ripple	- Binance Coin
- Bitcoin Cash	- Tether

(hyperlink to full size pictures of the matrices included)

[Volume Traded](#):



[Market Cap](#):



Hugo Joncour - 40139130
Zié Coulibaly - 40046155

From these matrices we learn that the Volume Traded is much more correlated than the Market Cap, which confirms our previous assumption about the exogeneity of demand, and from the comparison of the linear models we also learn that Market Cap is a slightly better estimator than the Volume Traded. More important, the R^2 is much lower than it was with the previous linear model, which might indicate that the high R^2 was caused more because of the high number of variables than because of an accurate model.

In the next part we will proceed to the regularization and prediction of the data.

4. Machine Learning Models

From what we saw in the previous part a regression using many variables, both market cap and volume traded would overfit. In the next part we will compare different Regularization methods to select which variable should we use in the prediction part.

A) Regularization:

1) Lasso

We want to select among our variables the best predictors for the Tether's market cap, so we perform a lasso regression:

```
volt <- cbind(X1_Bitcoin$VolumeTraded[1:860], X2_Ethereum$VolumeTraded[1:860], X3_XRP$VolumeTraded[1:860], X4_Bitcoin_Cash$VolumeTraded[1:860],
              X7_Litecoin$VolumeTraded[1:860], X8_EOS$VolumeTraded[1:860], X9_Binance_Coin$VolumeTraded[1:860], X15_TRON$VolumeTraded[1:860])
marketc <- cbind(X1_Bitcoin$MarketCap[1:860], X2_Ethereum$MarketCap[1:860], X3_XRP$MarketCap[1:860], X4_Bitcoin_Cash$MarketCap[1:860],
                 X7_Litecoin$MarketCap[1:860], X8_EOS$MarketCap[1:860], X9_Binance_Coin$MarketCap[1:860], X15_TRON$MarketCap[1:860])

predictor_matrix <- cbind(volt, marketc)
colnames(predictor_matrix) <- c("btc_vt", "eth_vt", "xrp_vt", "bcc_vt", "ltc_vt", "eos_vt","bnb_vt", "tron_vt",
                                "btc_mc", "eth_mc", "xrp_mc", "bcc_mc", "ltc_mc", "eos_mc","bnb_mc", "tron_mc")

# FIT LASSO MODEL
lasso_model <- cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]), lambda = 10^seq(9, 6, length = 80), alpha = 1, )

lasso_model$lambda.1se
plot(lasso_model)

best_lambda <- lasso_model$lambda.1se
lasso_coef <- lasso_model$glmnet.fit$beta[, lasso_model$glmnet.fit$lambda == best_lambda]
lasso_coef
```

With output:

```
> lasso_model <- cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]), lambda = 10^seq(9, 6, length = 80), alpha = 1, )
>
> lasso_model$lambda.1se
[1] 10600258
> plot(lasso_model)
>
> best_lambda <- lasso_model$lambda.1se
> lasso_coef <- lasso_model$glmnet.fit$beta[, lasso_model$glmnet.fit$lambda == best_lambda]
> lasso_coef
       btc_vt        eth_vt        xrp_vt        bcc_vt        ltc_vt        eos_vt        bnb_vt 
 0.0002786177  0.1496243603 -0.0434659542 -0.0204344364 -0.0364224952 -0.2783660574 -1.4830860827 
      tron_vt        btc_mc        eth_mc        xrp_mc        bcc_mc        ltc_mc        eos_mc 
 0.7878807896  0.0077520865  0.0000000000  0.0000000000 -0.0491572992 -0.0563833718  0.1568421707 
       bnb_mc       tron_mc 
 0.2681956178 -0.1341808080 
```

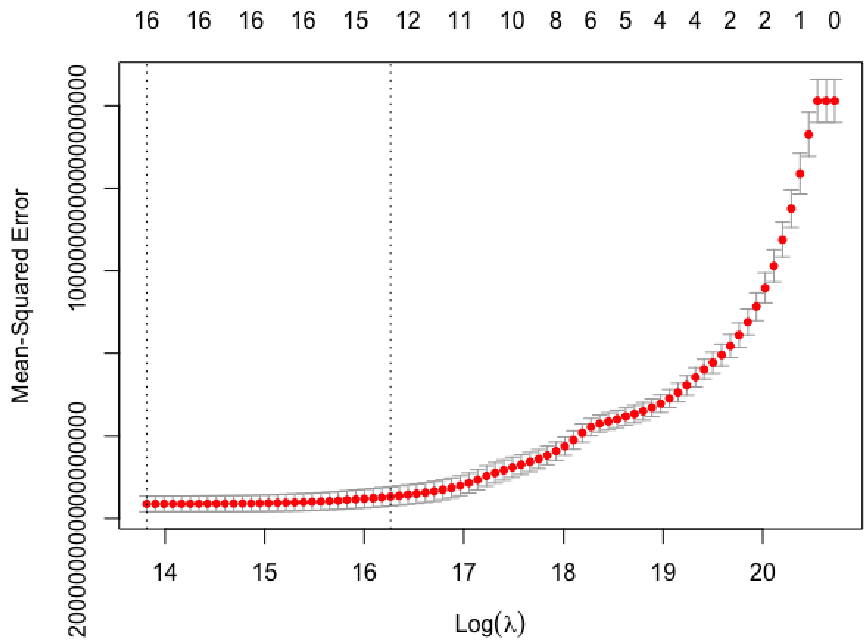
With lasso_model$Lambda.1se we receive the best λ : 10600258

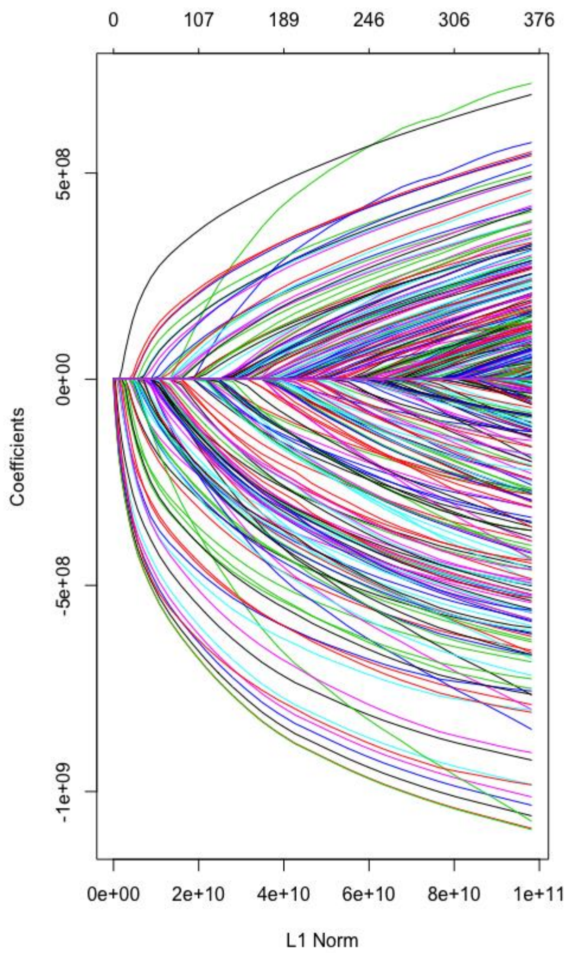
Hugo Joncour - 40139130
Zié Coulibaly - 40046155

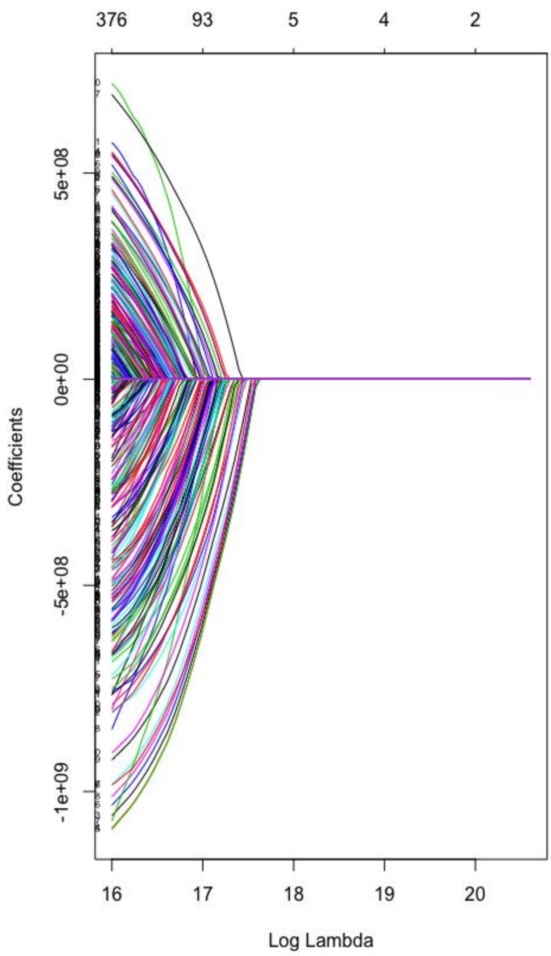
The big difference between Ridge and Lasso Regression is that Ridge Regression can only shrink the slope asymptotically close to 0 while Lasso can shrink all the way to 0 (sum of squared residuals).

The lasso_coef gives us the corrected variable coefficients, and as expect some variable's effect on the Tether's market cap is null: The Ethereum and the Ripple's market cap has no effect on the Tether.

And here is the plot of the Mean-Squared Error







Hugo Joncour - 40139130
Zié Coulibaly - 40046155

2) Ridge

With the same data we performed a regularization using the ridge model:

```
# FIT ON A RIDGE MODEL
ridge_model <- cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]), lambda = 10^seq(12, 7, length = 80), alpha = 0, )
plot(ridge_model)

best_lambda <- ridge_model$lambda.1se
ridge_coef <- ridge_model$glmnet.fit$beta[, ridge_model$glmnet.fit$lambda == best_lambda]
ridge_coef
```

The coefficients:

```
> best_lambda <- ridge_model$lambda.1se
> ridge_coef <- ridge_model$glmnet.fit$beta[, ridge_model$glmnet.fit$lambda == best_lambda]
> ridge_coef
      btc_vt       eth_vt       xrp_vt       bcc_vt       ltc_vt       eos_vt       bnb_vt      tron_vt       btc_mc       eth_mc 
 0.023830085  0.131929526 -0.082115004 -0.063202235 -0.085173569 -0.279164337 -1.610245313  0.796052044  0.006849442 -0.001343681 
      xrp_mc       bcc_mc       ltc_mc       eos_mc       bnb_mc      tron_mc 
 0.004399633 -0.042450651 -0.060003543  0.160094448  0.302707698 -0.169468167 
```

The model is working well, small impact variables aren't set to 0 like the lasso does, moderated marginal impact of the biggest variables

3) Elastic net

The elastic net is a simple combination of ridge and lasso:

```
# FIT ON ELASTICNET MODEL
elastic_net_model <- cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]), lambda = 10^seq(10, 6, length = 80), alpha = 0.5, )
plot(elastic_net_model)

best_lambda <- elastic_net_model$lambda.1se
elastic_net_coef <- elastic_net_model$glmnet.fit$beta[, elastic_net_model$glmnet.fit$lambda == best_lambda]
elastic_net_coef
```

The coefficients:

```
> best_lambda <- elastic_net_model$lambda.1se
> elastic_net_coef <- elastic_net_model$glmnet.fit$beta[, elastic_net_model$glmnet.fit$lambda == best_lambda]
> elastic_net_coef
      btc_vt       eth_vt       xrp_vt       bcc_vt       ltc_vt       eos_vt       bnb_vt      tron_vt       btc_mc       eth_mc 
 0.011652742  0.155767924 -0.063595485 -0.035643302 -0.082977795 -0.312076357 -1.642570424  0.827503970  0.007260463  0.000000000 
      xrp_mc       bcc_mc       ltc_mc       eos_mc       bnb_mc      tron_mc 
 0.002654930 -0.048855705 -0.053918241  0.167941862  0.293435337 -0.173373974 
```

It confirms what we thought: it is 'less severe' than the Lasso because the Ethereum's Market Cap is screened out, but the Ripple's Market Cap isn't.

4) Comparison:

These 3 models return different coefficients and mean-squared error we will see below:

```
# COMPARE COEFFICIENTS

coef = data.table(lasso = lasso_coef,
                  elastic_net = elastic_net_coef,
                  ridge = ridge_coef)
coef

to_plot = melt(coef, id.vars = "feature", variable.name = "model", value.name = "coefficient")


ggplot(to_plot, aes(x=feature, y = coefficient, fill = model)) + coord_flip() + geom_bar(stat = "identity")
  facet_wrap( ~ model) + guides(fill = FALSE)


ggplot(to_plot[grepl('mc', feature), ], aes(x = feature, y = coefficient, fill = model)) + coord_flip() +
  geom_bar(stat="identity") + facet_wrap( ~ model) + guides(fill = FALSE)

ggplot(to_plot[grepl('vt', feature), ], aes(x = feature, y = coefficient, fill = model)) + coord_flip() +
  geom_bar(stat="identity") + facet_wrap( ~ model) + guides(fill = FALSE)
```

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

Coefficients:

Upon a quick glance, it might seem that the three methods don't agree much. However, when taking a deeper dive, it would seem that they are in alliance.

In fact, we can see that they move in the same direction and differ by very little; some more negligible than others.

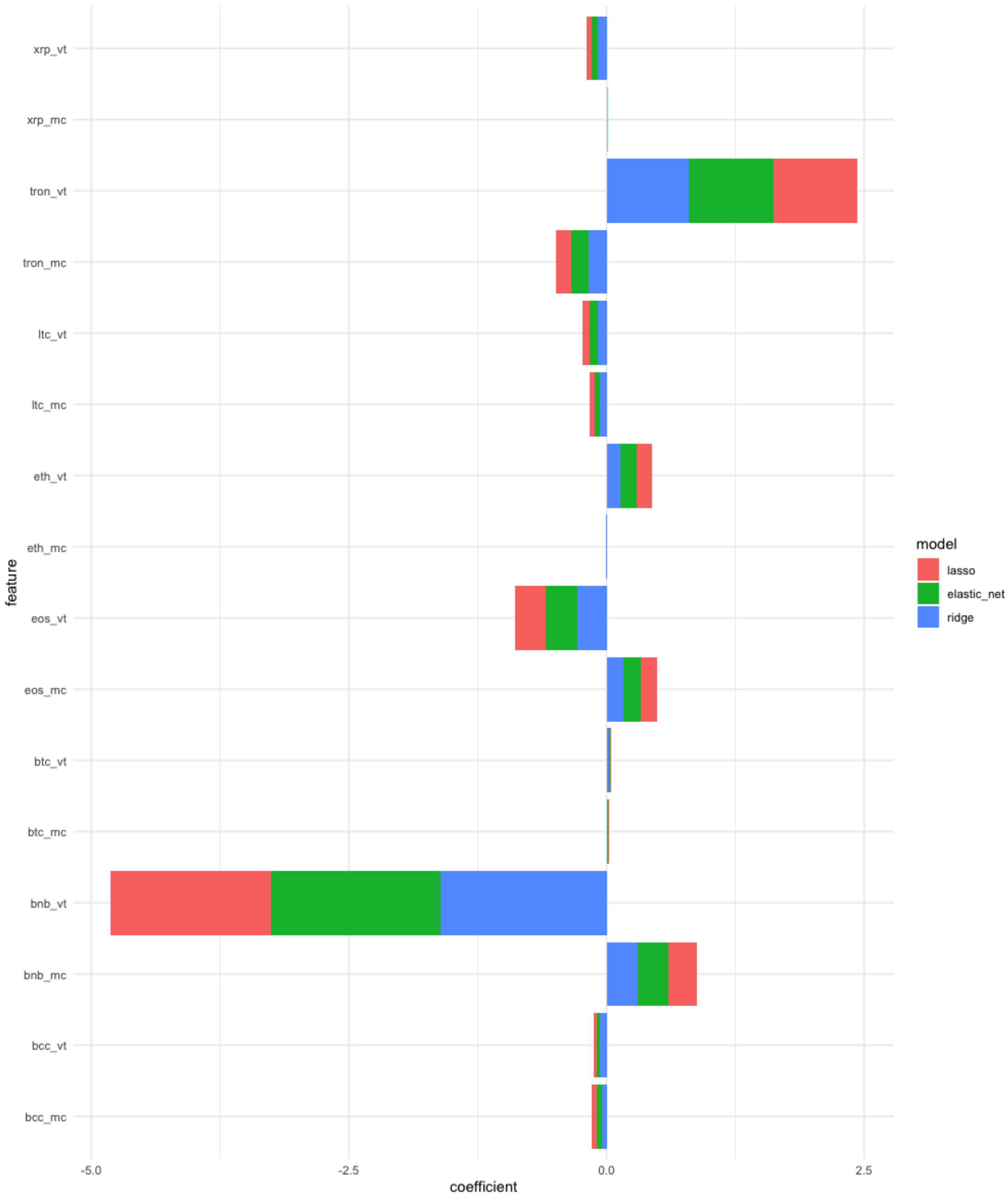
Despite this, it seems to be the case that the Lasso and the Elastic Net methods agree amongst themselves more than they do with the Ridge method.

Although it is close, the Ridge method is not as consistent as the other two methods.

```
> coef = data.table(lasso = lasso_coef,
+                   elastic_net = elastic_net_coef,
+                   ridge = ridge_coef)
> coef
          lasso  elastic_net        ridge
 1:  0.004463166  0.011652742  0.023830085
 2:  0.156680648  0.155767924  0.131929526
 3: -0.047010880 -0.063595485 -0.082115004
 4: -0.025955896 -0.035643302 -0.063202235
 5: -0.060692434 -0.082977795 -0.085173569
 6: -0.301121869 -0.312076357 -0.279164337
 7: -1.560806117 -1.642570424 -1.610245313
 8:  0.805731132  0.827503970  0.796052044
 9:  0.007571739  0.007260463  0.006849442
10:  0.000000000  0.000000000 -0.001343681
11:  0.000000000  0.002654930  0.004399633
12: -0.049462686 -0.048855705 -0.042450651
13: -0.052968288 -0.053918241 -0.060003543
14:  0.162187923  0.167941862  0.160094448
15:  0.277487028  0.293435337  0.302707698
16: -0.145268723 -0.173373974 -0.169468167
```

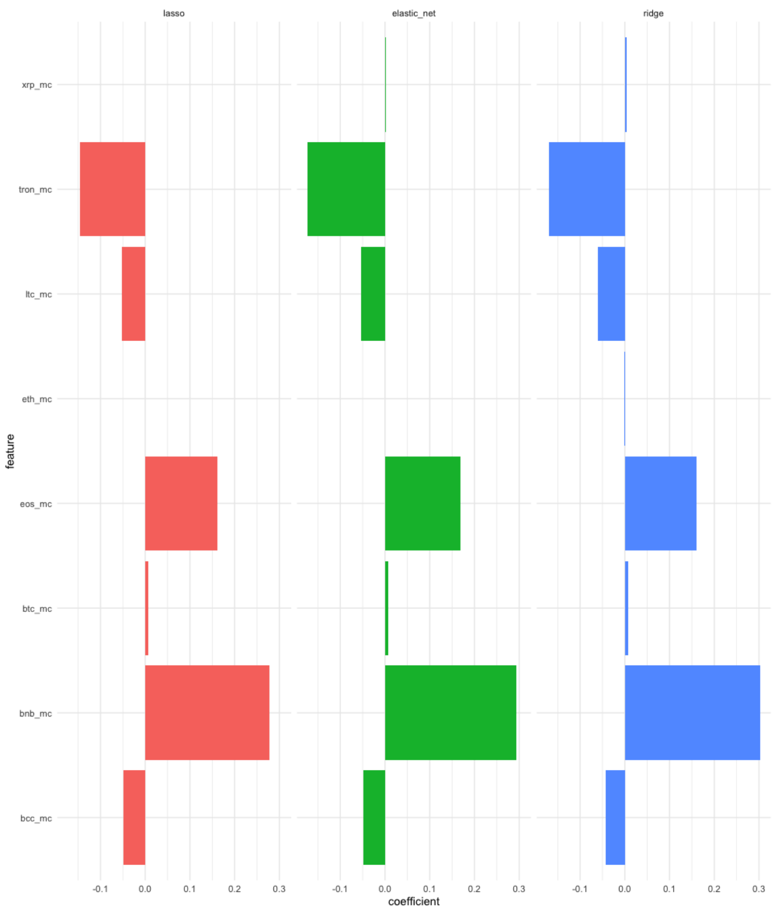
As a result, the data confirms our assumptions and thoughts.

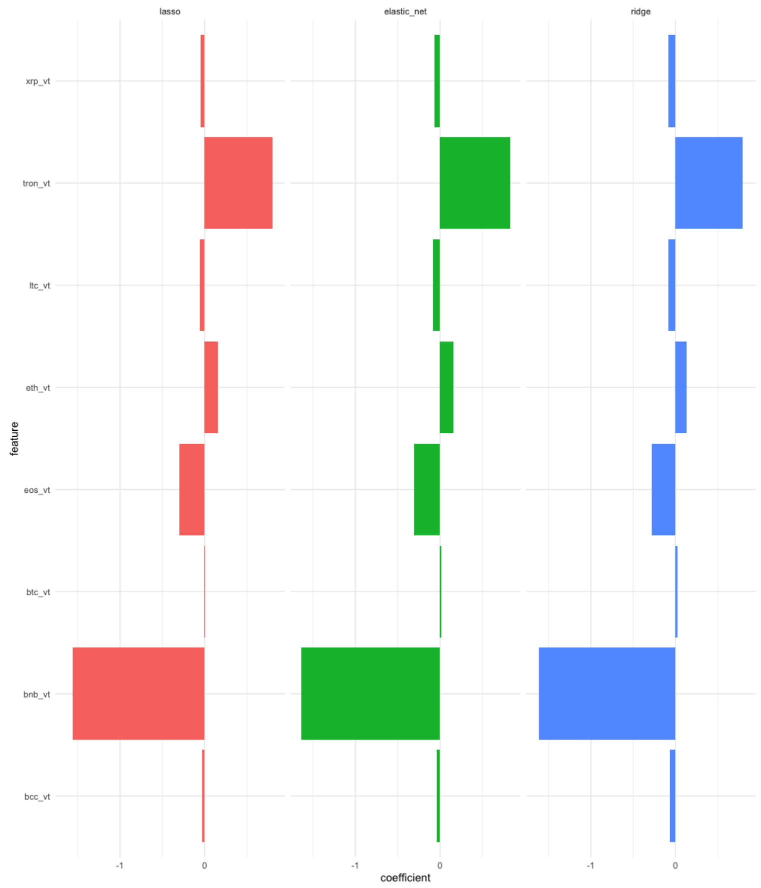
Hugo Joncour - 40139130
Zié Coulibaly - 40046155



Hugo Joncour - 40139130
Zié Coulibaly - 40046155

If we compare by market cap and volume traded we get:





When it comes to the comparison between Market Cap and Volume Traded, we can see something quite impressive regarding the three methods (lasso, elastic net, and ridge).

It would seem that within a single cryptocurrency, the three methods are in agreement: they move in the same direction.

However, it is nonetheless important that they vary very slightly. It would seem that in terms of coefficients, we observe that the Market Cap of Ripple, Tron, Litecoin and Bitcoin Cash considering the three methods, tend to yield negative coefficients, whereas the Market Caps of EOS, Bitcoin and Binance Coin tend towards a positive coefficient.

This shows that these coins either negatively or positively affect the Market Cap of Tether, depending on the coin.

On the other hand, when it comes to the Volume traded, we observe a different outcome: it seems that the Volume traded of Ripple, Litecoin, EOS, Binance coin and Bitcoin Cash have a negative tendency with respect to correlation Tether. Alternatively, it looks as though the Volume traded of Tron and Ethereum are of positive coefficients.

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

B) Prediction:

We decided that we would use the lasso regression as the best fit and remove the Tether's and the Ripple's Market Cap.

Having identified which variables would be ideal in this model we decided to use a Support Vector Regression to predict how will the Tether evolve in consequence.

Support Vector Regression uses the same idea as Support Vector Machine studied in class but applies it to real values rather than a class: It doesn't try to predict a class but a value.

We will first predict values for the parameters of the regression:

```
crypto <- read_csv("~/Desktop/ECON422_R_Project/crypto.csv")
crypto$day <- 1:869

#Forecasting for next 30 days of 1Bitcoin

svmodel <- svm(X1_Bitcoin$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_1BitcoinMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of TRON

svmodel <- svm(X15_TRON$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_15TRONMC <- predict(svmodel, newdata=data.frame(day=range_30days))
```

We will use these predicted values (above is just a sample, the full code is in annex) to predict the value of the Tether:

```
#Predicting 5_Tether using SVM regression and other Crypto-currencies for 30 days

svmodel <- svm(X5_Tether$MarketCap[1:869] ~ X4_Bitcoin_Cash$MarketCap[1:869] + X1_Bitcoin$MarketCap[1:869] + X7_Litecoin$MarketCap[1:869] +
               X8_EOS$MarketCap[1:869] + X9_Binance_Coin$MarketCap[1:869] +  X4_Bitcoin_Cash$VolumeTraded[1:869] +
               X1_Bitcoin$VolumeTraded[1:869] + X7_Litecoin$VolumeTraded[1:869] + X8_EOS$VolumeTraded[1:869] +
               X9_Binance_Coin$VolumeTraded[1:869]  + X2_Ethereum$VolumeTraded[1:869] + X3_XRP$VolumeTraded[1:869],
               data=day, type="eps-regression",kernel="radial",cost=10000, gamma=10)

predict30_X5Tether <- predict(svmodel, newdata = data.frame(X4BitcoinCashMC = predict30_4BitcoinCashMC, X4BitcoinCashVT = predict30_4BitcoinCashVT,
                                                  X3XRPVT = predict30_3XRPVT,X2EthereumVT = predict30_2EthereumVT,
                                                  X1BitcoinMC = predict30_1BitcoinMC, X1BitcoinVT = predict30_1BitcoinVT,
                                                  X7LitecoinMC = predict30_7LitecoinMC, X7LitecoinVT = predict30_7LitecoinVT,
                                                  X8EOSMC = predict30_8EOSMC, X8EOSMC = predict30_8EOSVT,
                                                  X9BinanceCoinMC = predict30_9BinanceCoinMC, X9BinanceCoinMC = predict30_9BinanceCoinMC))
```

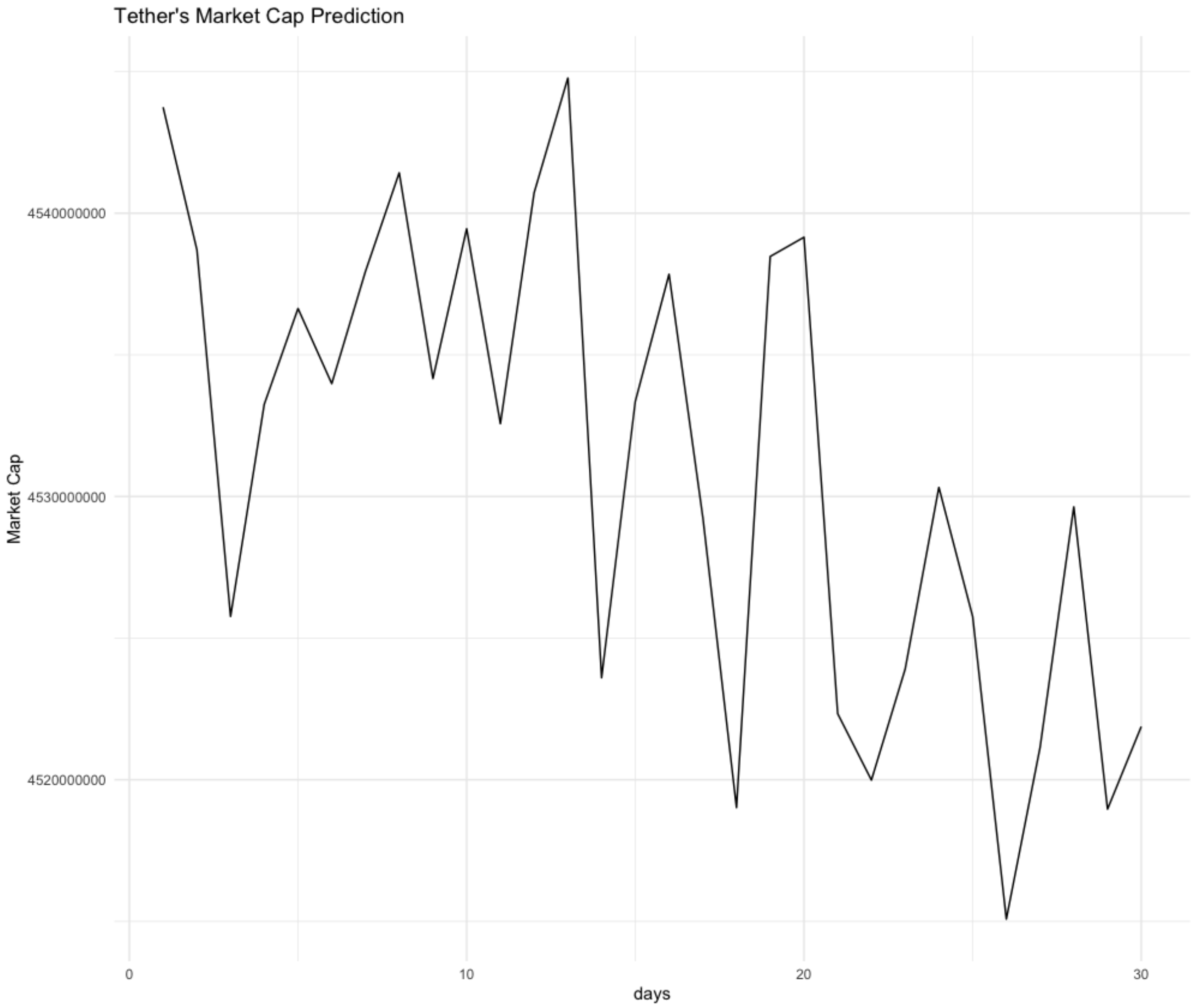
With the code above we make a regression model using the historical data of Tether and the other parameters we saved:

```
> svmodel

Call:
svm(formula = X5_Tether$MarketCap[1:869] ~ X4_Bitcoin_Cash$MarketCap[1:869] + X1_Bitcoin$MarketCap[1:869] + 
    X7_Litecoin$MarketCap[1:869] + X8_EOS$MarketCap[1:869] + X9_Binance_Coin$MarketCap[1:869] + X4_Bitcoin_Cash$
lumeTraded[1:869] + 
    X1_Bitcoin$VolumeTraded[1:869] + X7_Litecoin$VolumeTraded[1:869] + X8_EOS$VolumeTraded[1:869] + 
    X9_Binance_Coin$VolumeTraded[1:869] + X2_Ethereum$VolumeTraded[1:869] + X3_XRP$VolumeTraded[1:869], 
    data = crypto, type = "eps-regression", kernel = "radial", cost = 10000, gamma = 10)


Parameters:
   SVM-Type:  eps-regression 
 SVM-Kernel:  radial 
       cost:  10000 
      gamma:  10 
    epsilon:  0.1 


Number of Support Vectors:  676
```

With this model, we will predict the value of the Tether using the values we predicted for the other cryptocurrencies for the next 30 days:

```
> View(X5_Tether)
> predict30_X5Tether <- predict(svmodel, newdata = data.frame(X4BitcoinCashMC = predict30_4BitcoinCashMC, X4BitcoinCashVT = predict30_4BitcoinCashVT,
+                                                             X3XRPVT = predict30_3XRPVT,X2EthereumVT = predict30_2EthereumVT,
+                                                             X1BitcoinMC = predict30_1BitcoinMC, X1BitcoinVT = predict30_1BitcoinVT,
+                                                             X7LitecoinMC = predict30_7LitecoinMC, X7LitecoinVT = predict30_7LitecoinVT,
+                                                             X8EOSMC = predict30_8EOSMC,        X8EOSMC = predict30_8EOSVT,
+                                                             X9BinanceCoinMC = predict30_9BinanceCoinMC, X9BinanceCoinMC = predict30_9BinanceCoinMC))
> 
> predict30_X5Tether
         1          2          3          4          5          6          7          8          9 
4543740621 4538718492 4525765025 4533257474 4536633599 4533981371 4537945958 4541425636 4534162019 
        10         11         12         13         14         15         16         17         18 
4539449242 4532568152 4540721481 4544768570 4523599444 4533356387 4537842941 4529274131 4519015866 
        19         20         21         22         23         24         25         26         27 
4538471862 4539150768 4522330827 4519987420 4523902247 4530317237 4525767521 4515080250 4521140345 
        28         29         30 
4529631962 4518962189 4521875576 
```

And if we plot these predicted values we get:



Hugo Joncour - 40139130
Zié Coulibaly - 40046155

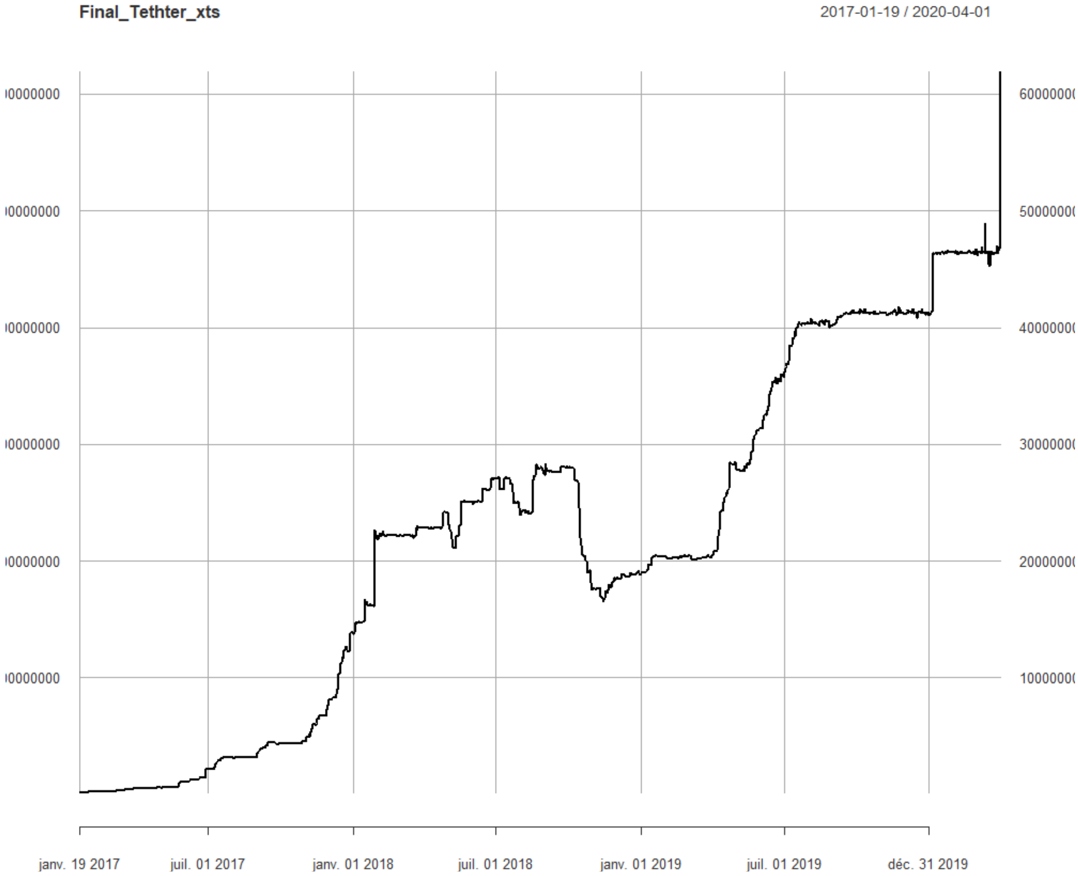
5. Conclusion

In this project we tried to predict the evolution of the Tether's market Cap given other cryptocurrencies's attributes. We could observe that a linear regression was unfit for such task because they would either be inaccurate or simply overfit.

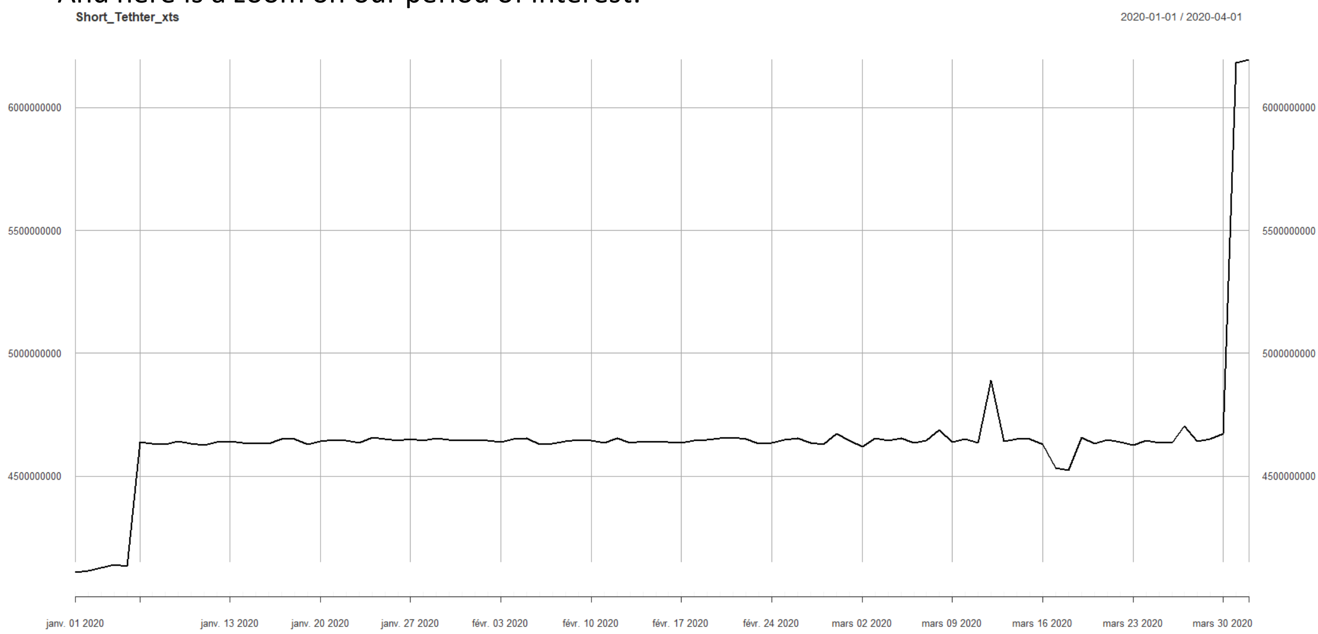
We used regularization with lasso, ridge and elastic net to identify with more accuracy the impact of each variable and compared them.

We sticked to the lasso coefficients and predicted the value with a Support Vector Regression on the next 30 days

Here is the historical data including the months we haven't used in our models (January, February, March, April of 2020):



And here is a zoom on our period of interest:

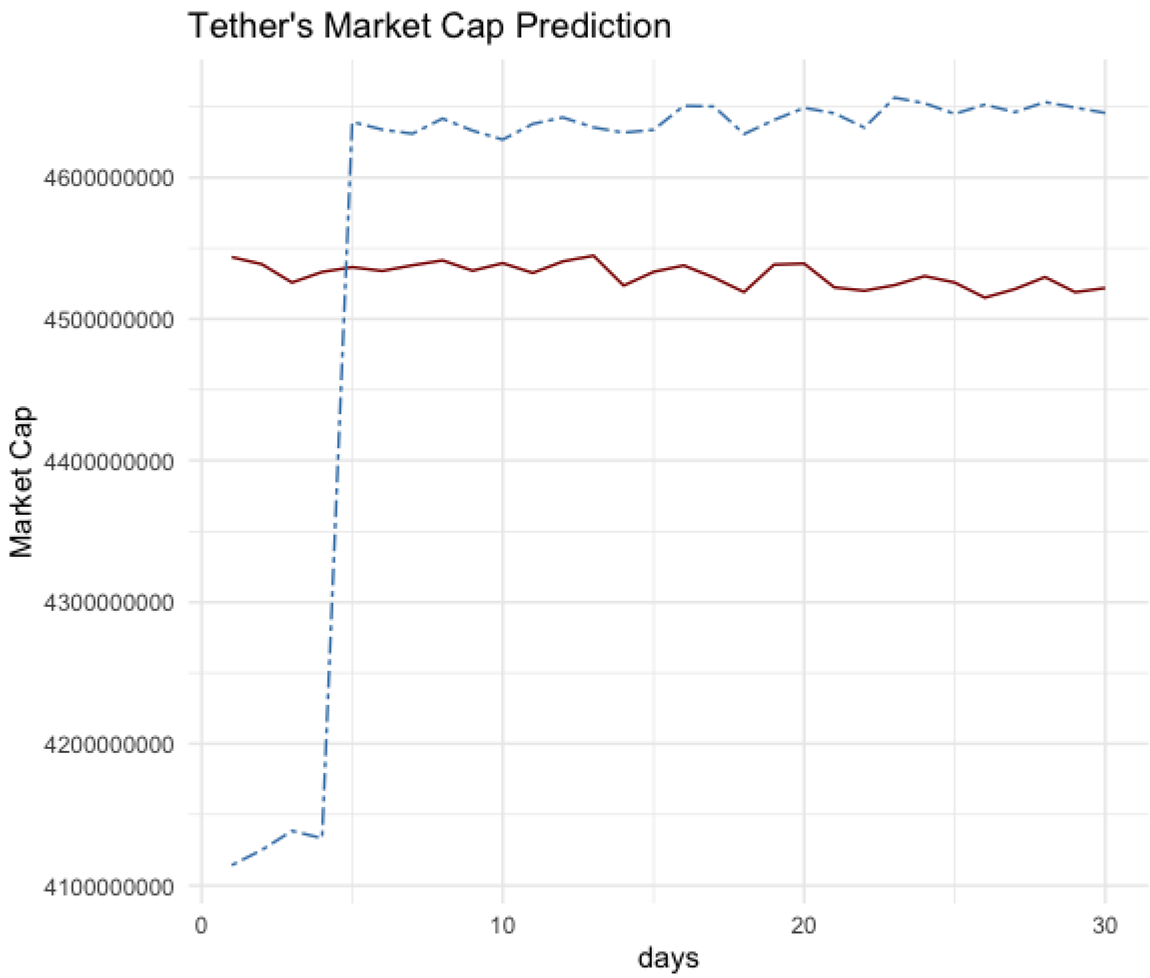


Hugo Joncour - 40139130
Zié Coulibaly - 40046155

If we plot together our predictions and our period of interest:

```
data <- data.frame(cbind(predict30_X5Tether,Final_Tether30Days$MarketCap[1:30] ))
colnames(data) <- c("predicted", "real")

g <- ggplot(data, aes(x=time30)) +
      geom_line(aes(y=data$predicted), color = "darkred") +
      geom_line(aes(y=data$real), color="steelblue", linetype = "twodash")+
      labs(title = "Tether's Market Cap Prediction", x = "days", y ="Market Cap")

g
```

We get:



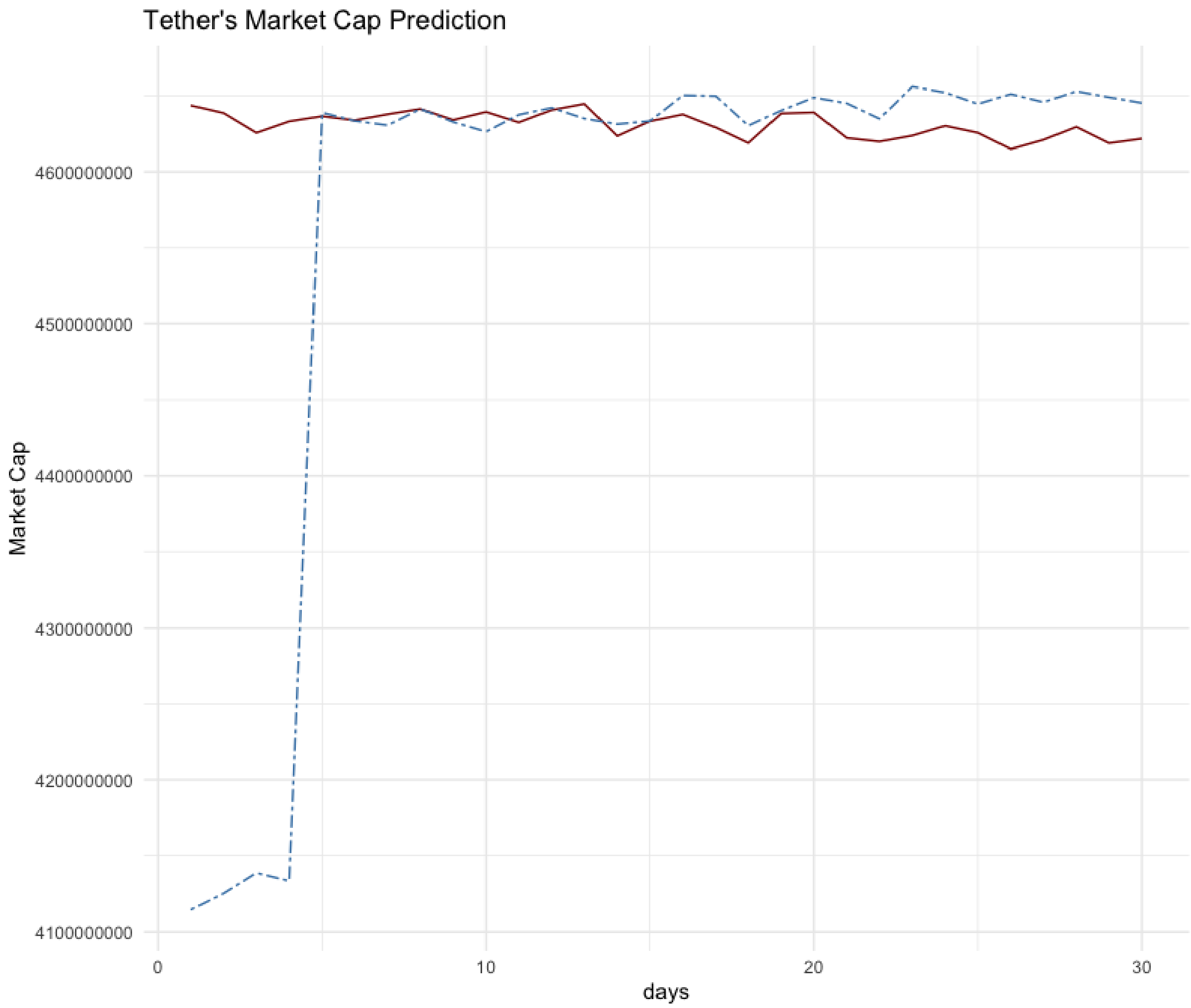
However, the raise in the 'real' Tether's Market Cap (in blue) is not caused a market activity (which would be visible on the market cap, price, and volume traded of other cryptocurrencies) but by Bitfinex who decided to create ex nihilo 500 000 000$ of USDT, that day, the 5th of January 2020, and with it to buy assets to other people.

If we look at the historical data of the Tether on the previous page, we can see that it sometimes happens with a sharp rise in the market cap.

As we saw in the part 4, the regression model is based on historical data, and the prediction model, on predicted data, thus it is not an event that could be anticipated or considered by any of these models.

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

If we decide to change the intercept in our predicted data, we get:



In the end, the best we can say for now is that:

“this model may be moderately accurate on a short period of time because he seems able to approximate the movements of the market cap in a near future”.

However, we have to keep in mind:

1) That it doesn’t predict arbitrary changes such as the printing of 100 000 000 USDT that occurred the 5th of January
2) This model, if it indeed works, has a very limited window prediction: the more the time passes, the bigger the gap between the predicted and the real value
3) It may be pure coincidence: these predicted values of the other cryptocurrency used to predict the Tether might just fit at the right time and would give a bad prediction at another time.

To conclude, it might be possible to predict the Tether’s Market Cap using the market data and other assets, however, to be accurate and reliable it would require much more data, much more training and would still have some limits.

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

6. Annex

Here is most of the code we used for this project:

a) Java

This method was used to identify data on a website's page and make a java object out of it through the Currency Class.

```java
public static void scrapeHistoricalCurrency(Currency currency){
    try {
        int iterator = 1;
        final Document coinMarketCapCurrencies = Jsoup.connect(currency.getHistoricsURL()).get();
        for(Element row: coinMarketCapCurrencies.select(".cmc-tab-historical-data tr")){
            if(iterator > 4) {
                String date = row.select(".cmc-table__cell--sticky.cmc-table__cell").text();
                String priceOpen = row.select(".cmc-table__cell:nth-of-type(2)").text();
                String priceClose = row.select(".cmc-table__cell:nth-of-type(3)").text();
                String dayHigh = row.select(".cmc-table__cell:nth-of-type(4)").text();
                String dayLow = row.select(".cmc-table__cell:nth-of-type(5)").text();
                String volumeTradedUSD = row.select(".cmc-table__cell:nth-of-type(6)").text();
                String marketCapUSD =  row.select(".cmc-table__cell:nth-of-type(7)").text();
            currency.addHistorics(currency.getName(),
                    date, priceOpen, priceClose, dayHigh, dayLow, volumeTradedUSD, marketCapUSD);
            } iterator++;
            }
        } catch(Exception exception) {exception.printStackTrace(); }
    }
}
```

This method was used to make a CSV for every currency we had data on. For parsing reasons, we used semicolons because commas were used for decimals on the site.

```java
package prog.output.csv;

import java.io.File;
import java.io.PrintWriter;
import prog.algo.types.*;

public class Printer {
    public static void createFullCSV(Currency currency){
        try {
            PrintWriter pw = new PrintWriter(new File("D:\\Desktop\\Projets\\TestDefaultOutput\\"
                    +currency.getRank()+"_"+currency.getName()+".csv"));
            StringBuilder sb = new StringBuilder();
            sb.append("Date");
            sb.append(";");
            sb.append("Opens at:");
            sb.append(";");
            sb.append("Closes at:");
            sb.append(";");
            sb.append("Highest at:");
            sb.append(";");
            sb.append("Lowest at:");
            sb.append(";");
            sb.append("Volume Traded:");
            sb.append(";");
            sb.append("Market Cap:");
            sb.append("\r\n");
            pw.write(sb.toString());
            for(int i = 0; i < currency.getHistory().size(); i++) {
                StringBuilder line = new StringBuilder();
                line.append(currency.getHistory().get(i).getDay()+"/"+
                        currency.getHistory().get(i).getMonth()+"/"+
                        currency.getHistory().get(i).getYear()+";");
                line.append(currency.getHistory().get(i).getPriceOpen()+";");
                line.append(currency.getHistory().get(i).getPriceClose()+";");
                line.append(currency.getHistory().get(i).getDayHigh()+";");
                line.append(currency.getHistory().get(i).getDayLow()+";");
                line.append(currency.getHistory().get(i).getVolumeTraded()+";");
                line.append(currency.getHistory().get(i).getMarketCap()+";\n");
                pw.write(line.toString());
        }
            pw.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

b) R

Code we used for the covariance matrices of the market cap and volume traded

```
X1_Bitcoin <- read_csv("10 first/1_Bitcoin.csv")
X2_Ethereum <- read_csv("10 first/2_Ethereum.csv")
X3_XRP <- read_csv("10 first/3_XRP.csv")
X4_Bitcoin_Cash <- read_csv("Data/4_Bitcoin_Cash.csv")
X5_Tether <- read_csv("10 first/5_Tether.csv")
X6_BitcoinSV <- read_csv("10 first/6_Bitcoin SV.csv")
X7_Litecoin <- read_csv("10 first/7_Litecoin.csv")
X8_EOS <- read_csv("10 first/8_EOS.csv")
X9_BinanceCoin <- read_csv("10 first/9_Binance Coin.csv")
X10_Tezos <- read_csv("10 first/10_Tezos.csv")
X11_Chainlink <- read_csv("10 first/11_Chainlink.csv")
X12_Cardano <- read_csv("Data/12_Cardano.csv")
X13_Monero <- read_csv("Data/13_Monero.csv")
X14_Stellar <- read_csv("Data/14_Stellar.csv")
X15_TRON <- read_csv("Data/15_TRON.csv")
X19_Neo <- read_csv("Data/19_Neo.csv")
X20_Dash <- read_csv("Data/20_Dash.csv")

btc_mc <- X1_Bitcoin$MarketCap[1:857]
eth_mc <- X2_Ethereum$MarketCap[1:857]
xrp_mc <- X3_XRP$MarketCap[1:857]
bcc_mc <- X4_Bitcoin_Cash$MarketCap[1:857]
usdt_mc <- X5_Tether$MarketCap[1:857]
bsv_mc <- X6_BitcoinSV$MarketCap[1:857]
ltc_mc <- X7_Litecoin$MarketCap[1:857]
eos_mc <- X8_EOS$MarketCap[1:857]
bcb_mc <- X9_BinanceCoin$MarketCap[1:857]
xtz_mc <- X10_Tezos$MarketCap[1:857]
xlm_mc <- X14_Stellar$MarketCap[1:857]
trx_mc <- X15_TRON$MarketCap[1:857]
neo_mc <- X19_Neo$MarketCap[1:857]
dash_mc <- X20_Dash$MarketCap[1:857]

market_cap_comatrix <- data.frame(btc_mc, eth_mc, xrp_mc, bcc_mc, usdt_mc, ltc_mc, eos_mc, bcb_mc, xlm_mc, trx_mc, neo_mc, dash_mc)
View(market_cap_comatrix)
mc_covmatrix = matrix(c(cov(market_cap_comatrix)), nrow=12, ncol=12)

mc_covmatrix
btc_vt <- X1_Bitcoin$VolumeTraded[1:857]
eth_vt <- X2_Ethereum$VolumeTraded[1:857]
xrp_vt <- X3_XRP$VolumeTraded[1:857]
bcc_vt <- X4_Bitcoin_Cash$VolumeTraded[1:857]
usdt_vt <- X5_Tether$VolumeTraded[1:857]
ltc_vt <- X7_Litecoin$VolumeTraded[1:857]
eos_vt <- X8_EOS$VolumeTraded[1:857]
bcb_vt <- X9_BinanceCoin$VolumeTraded[1:857]
xtz_vt <- X10_Tezos$VolumeTraded[1:857]
xlm_vt <- X14_Stellar$VolumeTraded[1:857]
trx_vt <- X15_TRON$VolumeTraded[1:857]
neo_vt <- X19_Neo$VolumeTraded[1:857]
dash_vt <- X20_Dash$VolumeTraded[1:857]

volume_traded_comatrix <- data.frame(btc_vt, eth_vt, xrp_vt, bcc_vt, usdt_vt, ltc_vt, eos_vt, bcb_vt, xlm_vt, trx_vt, neo_vt, dash_vt)
View(volume_traded_comatrix)
vt_covmatrix = matrix(c(cov(volume_traded_comatrix)), nrow=12, ncol=12)
vt_covmatrix
```

Code for the cryptocurrencies plots 'First Results':

```
library(tidyverse)
library(ggplot2)
library(reshape2)
library(scales)
library(lubridate)

#Scientific to numerical notation
options(scipen = 999)

Bitcoin <- read_csv("Data/1_Bitcoin.csv")

data <- Bitcoin #Coin file
data$Date <- dmy(data$Date)

#Normal portion
meltdata <- melt(data,id="Date")
ggplot(meltdata,aes(x=Date,y=value,colour=variable,group=variable)) + geom_line()

#Log portion
lg <- ggplot(meltdata,aes(x=Date,y=value,colour=variable,group=variable)) + geom_line()

lg + scale_y_continuous(trans = log10_trans())
```

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

Code for the linear models and matrices (for Volume Traded, Market Cap is the same):

```
#VOLT
vol_t <- cbind(X1_Bitcoin$VolumeTraded[1:919], X2_Ethereum$VolumeTraded[1:919], X3_XRP$VolumeTraded[1:919],
               X4_Bitcoin_Cash$VolumeTraded[1:919], X7_Litecoin$VolumeTraded[1:919], X8_EOS$VolumeTraded[1:919],
               X9_Binance_Coin$VolumeTraded[1:919])

volt <- (cbind(vol_t, X5_Tether$MarketCap[1:919]))

colnames(volt) <- c("btc_vt", "eth_vt", "xrp_vt", "bcc_vt", "ltc_vt", "eos_vt","bnb_vt", "tether_mc")

# Color
my_cols <- c("#FC4E07", "#DAF7A6", "#FFC300", "#FF5733", "#C70039", "#900C3F", "#581845", "#000000")
pairs(volt, pch = 1, col = my_cols, upper.panel = NULL)

lowerpannel<-function(x, y){
  points(x,y, pch = 1, col = my_cols)
}
panelcor <- function(x, y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), digits=2)
  txt <- paste0("R = ", r)
  cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(volt,upper.panel = panelcor, lower.panel = lowerpannel)
```

Lasso Regression:

```
volt <- cbind(X1_Bitcoin$VolumeTraded[1:860], X2_Ethereum$VolumeTraded[1:860], X3_XRP$VolumeTraded[1:860],
              X4_Bitcoin_Cash$VolumeTraded[1:860], X7_Litecoin$VolumeTraded[1:860], X8_EOS$VolumeTraded[1:860],
              X9_Binance_Coin$VolumeTraded[1:860], X15_TRON$VolumeTraded[1:860])
marketc <- cbind(X1_Bitcoin$MarketCap[1:860], X2_Ethereum$MarketCap[1:860], X3_XRP$MarketCap[1:860],
                 X4_Bitcoin_Cash$MarketCap[1:860], X7_Litecoin$MarketCap[1:860], X8_EOS$MarketCap[1:860],
                 X9_Binance_Coin$MarketCap[1:860], X15_TRON$MarketCap[1:860])

predictor_matrix <- cbind(volt, marketc)

set.seed(10000)
lasso_model = cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]),
                        lambda = 10^seq(9, 6, length = 80), alpha = 1, )
lasso_model $lambda.1se
plot(lasso_model)
```

Ridge Regression:

```
# FIT ON A RIDGE MODEL
ridge_model <- cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]), lambda = 10^seq(12, 7, length = 80), alpha = 0, )
plot(ridge_model)

best_lambda <- ridge_model$lambda.1se
ridge_coef <- ridge_model$glmnet.fit$beta[, ridge_model$glmnet.fit$lambda == best_lambda]
ridge_coef
```

Elastic net Regression:

```
# FIT ON ELASTICNET MODEL
elastic_net_model <- cv.glmnet(as.matrix(predictor_matrix), as.matrix(X5_Tether$MarketCap[1:860]), lambda = 10^seq(10, 6, length = 80), alpha = 0.5, )
plot(elastic_net_model)

best_lambda <- elastic_net_model$lambda.1se
elastic_net_coef <- elastic_net_model$glmnet.fit$beta[, elastic_net_model$glmnet.fit$lambda == best_lambda]
elastic_net_coef
```

Coefficient Comparison & Plots :

```
# COMPARE COEFFICIENTS

coef = data.table(lasso = lasso_coef,
                  elastic_net = elastic_net_coef,
                  ridge = ridge_coef)
coef

to_plot = melt(coef, id.vars = "feature", variable.name = "model", value.name = "coefficient")


ggplot(to_plot, aes(x=feature, y = coefficient, fill = model)) + coord_flip() + geom_bar(stat = "identity")
  facet_wrap( ~ model) + guides(fill = FALSE)


ggplot(to_plot[grepl('mc', feature), ], aes(x = feature, y = coefficient, fill = model)) + coord_flip() +
  geom_bar(stat="identity") + facet_wrap( ~ model) + guides(fill = FALSE)

ggplot(to_plot[grepl('vt', feature), ], aes(x = feature, y = coefficient, fill = model)) + coord_flip() +
  geom_bar(stat="identity") + facet_wrap( ~ model) + guides(fill = FALSE)
```

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

Support Vector Regression:

```
crypto <- read_csv("~/Desktop/ECON422_R_Project/crypto.csv")
crypto$day <- 1:869

#Forecasting for next 30 days of 1Bitcoin

svmodel <- svm(X1_Bitcoin$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_1BitcoinMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of TRON

svmodel <- svm(X15_TRON$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_15TRONMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 4BitcoinCash

svmodel <- svm(X4_Bitcoin_Cash ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_4BitcoinCashMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 7Litecoin

svmodel <- svm(X7_Litecoin$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_7LitecoinMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 8EOS

svmodel <- svm(X8_EOS$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_8EOSMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 9BinanceCoin

svmodel <- svm(X9_Binance_Coin$MarketCap[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_9BinanceCoinMC <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of TRON

svmodel <- svm(X15_TRON$VolumeTraded[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_15TRONVT <- predict(svmodel, newdata=data.frame(day=range_30days))

#Forecasting for next 30 days of 2Ethereum

svmodel <- svm(X2_Ethereum$VolumeTraded[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_2EthereumVT <- predict(svmodel, newdata=data.frame(day=range_30days))

#Forecasting for next 60 days of 2Ethereum

range_60days <- 870:929
predict60_2EthereumVT <- predict(svmodel, newdata=data.frame(day=range_60days))


#Forecasting for next 30 days of 3XRP

svmodel <- svm(X3_XRP$VolumeTraded[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899

predict30_3XRPVT <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 4BitcoinCash

svmodel <- svm(X4_Bitcoin_Cash ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_4BitcoinCashVT <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 7Litecoin

svmodel <- svm(X7_Litecoin$VolumeTraded[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_7LitecoinVT <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 8EOS

svmodel <- svm(X8_EOS$VolumeTraded[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_8EOSVT <- predict(svmodel, newdata=data.frame(day=range_30days))


#Forecasting for next 30 days of 9BinanceCoin

svmodel <- svm(X9_Binance_Coin$VolumeTraded[1:869] ~ day,data=predictor_matrix, type="eps-regression",kernel="radial",cost=10000, gamma=10)
range_30days <- 870:899
predict30_9BinanceCoinVT <- predict(svmodel, newdata=data.frame(day=range_30days))


#Predicting 5_Tether using SVM regression and other Crypto-currencies for 30 days

svmodel <- svm(X5_Tether$MarketCap[1:869] ~ X4_Bitcoin_Cash$MarketCap[1:869] + X1_Bitcoin$MarketCap[1:869] + X7_Litecoin$MarketCap[1:869] +
               X8_EOS$MarketCap[1:869] + X9_Binance_Coin$MarketCap[1:869] +  X4_Bitcoin_Cash$VolumeTraded[1:869] +
               X1_Bitcoin$VolumeTraded[1:869] + X7_Litecoin$VolumeTraded[1:869] + X8_EOS$VolumeTraded[1:869] +
               X9_Binance_Coin$VolumeTraded[1:869]  + X2_Ethereum$VolumeTraded[1:869] + X3_XRP$VolumeTraded[1:869],
               data=day, type="eps-regression",kernel="radial",cost=10000, gamma=10)

predict30_X5Tether <- predict(svmodel, newdata = data.frame(X4BitcoinCashMC = predict30_4BitcoinCashMC, X4BitcoinCashVT = predict30_4BitcoinCashVT,
                                                            X3XRPVT = predict30_3XRPVT,X2EthereumVT = predict30_2EthereumVT,
                                                            X1BitcoinMC = predict30_1BitcoinMC, X1BitcoinVT = predict30_1BitcoinVT,
                                                            X7LitecoinMC = predict30_7LitecoinMC, X7LitecoinVT = predict30_7LitecoinVT,
                                                            X8EOSMC = predict30_8EOSMC, X8EOSMC = predict30_8EOSVT,
                                                            X9BinanceCoinMC = predict30_9BinanceCoinMC, X9BinanceCoinMC = predict30_9BinanceCoinMC))
```

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

c) Python

Python code used to arrange the .csv so he can be correctly used by R (comma is used to separate thousands) in the scraped data.

```
import os

files = [f for f in os.listdir('.') if os.path.isfile(f) if f.endswith(".csv")]

for i in range(0, len(files)):
    w = open(files[i], "r+")
    list = []
    for line in w:
        tx = line.replace(",", "")
        tx = tx.replace(";", ",")
        list.append(tx)
    w.close()
    w = open(files[i], "w")
    for line in list:
        w.write(line)
```

Hugo Joncour - 40139130
Zié Coulibaly - 40046155

7. Sources

a) Data:

- https://coinmarketcap.com/currencies/
- https://medium.com/coinmonks/how-to-get-historical-crypto-currency-data-954062d40d2d

b) R:

- https://github.com/TaddyLab/BDS
- http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs
- http://www.sthda.com/english/articles/32-r-graphics-essentials/128-plot-time-series-data-using-ggplot/
- https://www.r-bloggers.com/plotting-time-series-data-using-ggplot2/
- http://www.sthda.com/english/articles/32-r-graphics-essentials/128-plot-time-series-data-using-ggplot/
- https://www.svm-tutorial.com/2014/10/support-vector-regression-r/
- https://www.kdnuggets.com/2017/03/building-regression-models-support-vector-regression.html
- http://www.semspirit.com/artificial-intelligence/machine-learning/regression/support-vector-regression/support-vector-regression-in-r/
- https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ffv
- https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html
- https://www.rstatisticsblog.com/data-science-in-action/lasso-regression/
- https://www.r-bloggers.com/ridge-regression-and-the-lasso/
- https://www.youtube.com/watch?v=FWCPFUwZkn0
- https://www.youtube.com/watch?v=NGf0voTMlcs
- Business Data Science – Matt Taddy

c) Diverse:

- https://www.bloomberg.com/news/articles/2018-11-20/bitcoin-rigging-criminal-probe-is-said-to-focus-on-tie-to-tether
- https://arstechnica.com/tech-policy/2018/02/tether-says-its-cryptocurrency-is-worth-2-billion-but-its-audit-failed/

The code is on GitHub: https://github.com/SanteauX/ECON422_R_Project