UNIVERSITÉ PARIS II
PANTHÉON-ASSAS

What determines the Bitcoin price ?

Florentin LAVAUD

Direction : S. Cavaco (LEMMA)

Master 1 – Analyse et politique économique parcours recherche en sciences économiques

Term 1

Software: R Studio

2020-2021 academic year

Table of contents

Introduction

What determines the Bitcoin price ?

Stefan Thomas, a German programmer in San Francisco, is now famous for a gloomy story. Indeed, he had just lost his bitcoin password. Due to a speculative bubble, his 7 000 locked Bitcoins worth now more than $240 million dollar. This funny story questions the recent speculative bubble : why the market price of bitcoin explodes ? Our objective is to study the main determinant of the bitcoin market price ie what driven the bitcoin prices. Especially, we want to know if there is a speculative bubble over the bitcoin price. To concern this financial economics issue, we are going to study time-series data.
Bitcoin is the most employed cryptocurrencies which is decentralized (ie without a central bank) and was created by Satoshi Nakamoto. It's mostly famous for two facts : the first is that this currency is extremely volatile and the second is that bitcoin is mainly use as money for the underground economy.

Bitcoin was a new form of money and therefore is challenging the traditional design of the currency as a unitary, territorialized, and centralized system. Thus, it will be interesting to compare the relation between bitcoin and cryptocurrencies (such as Ripple or Litecoin) with the relation between bitcoin and historical currencies (such as the Euro, the Dollar and so on).

Presentation of the data base

Our data are mainly coming from inverting.com and Boursorama bank. Investing.com is quite famous and sometime described as the "robin hood of market data". Indeed, all information are free-to-access, in multiple language.

Data are download in csv format so that the importation is easy. We import weekly data because with monthly data we don't have enough data so that the degree of freedom is too low. If we had use daily data, we will have a problem of date. Indeed, bitcoin market is open all day but historical money, such as euro, pounds, and yen, are only open from Monday to Friday. This gap would complicate a lot our data treatment. Weekly data enables us to avoid getting into trouble.

Firstly, we take all currency from 2016 to 2021 because data were available in English for this period. Note that we import data in English because our software is in English. Importing french data causes an issue: a thousand is written 1,000.00 in English and 1 000,00 in French. Despite changing delimiter parameter while importing data, the problem still exists so we import English data.

Our data are time series data. Indeed, we have the evolution of data over the time. We have a series of data points indexed in time order. Note that all data are expressed in Dollar (ie currency's bilateral) in such a way that all data are comparable. Needless to say, the role of the dollar in explaining the bitcoin price is presumed important. But, by expressing all currencies face to dollar, they are comparable which is far more important.

Our observations must be reduced from 264 weeks for Bitcoin to 157 weeks. Indeed, Litecoin is a very recent cryptocurrency so that data can't be found before. Therefore, we cut all data to have our data base.

from the most recent week January 21st, 2021 to January 21st, 2018. Moreover, for creating our database we must merge same size vectors.

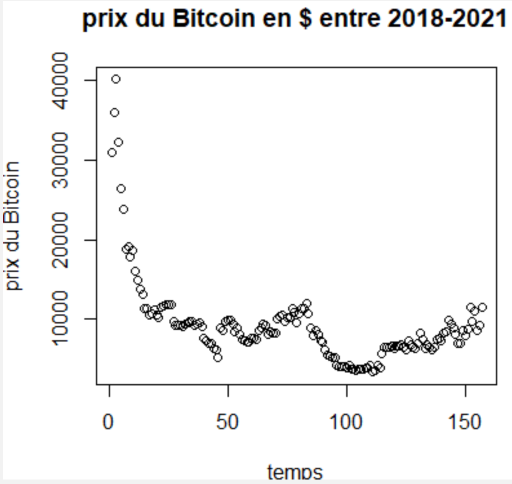
We have 157 weeks for 6 different currencies. Bitcoin, Litecoin and Ripple which are cryptocurrencies and Euro, Pounds and Yen which are historical currencies. It could be possible that cryptocurrencies have a different impact on Bitcoin than historical currency. We'll find out about this relation later.

At first, our data are too rich in information. Indeed, we have the date, the price, the open price, the highest and lowest price, the variation, and the volume. Even though all information is interesting, we choose to focus on the market price during the past 157th week (from January 21st, 2021 to January 21st, 2018).

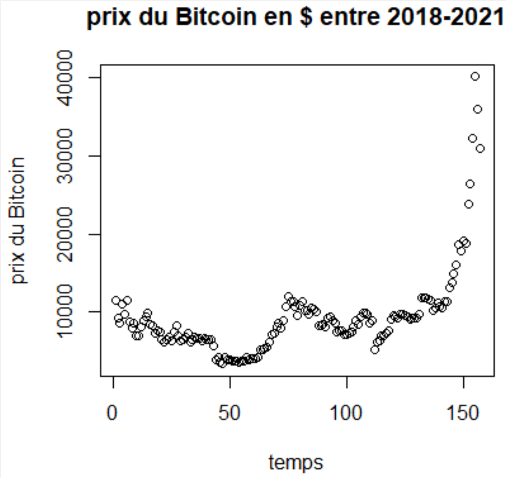
While creating our data base we encounter a problem: our data start with the most recent date. But the interpretation of graphs isn't intuitive (see graph 1). We found a solution on the internet by computing:

BDD<-BDD1[dim(BDD1)[1]:1,]

Then, the interpretation becomes far more intuitive (see graph 2) – from the older to the newly



Graph 1



Graph 2

To sum up :

Data	Variable name
Bitcoin (Y)	Weekly Bitcoin market price /$
Litecoin	Weekly Litecoin market price /$
Ripple	Weekly Ripple market price /$
Euro	Weekly Euro market price /$
Sterling	Weekly Pounds market price /$
Yen	Weekly Yen market price /$

This was our initial data base. But, because of the presence of either speculative bubble or nonlinear regression, we create different data base that we'll explain in good time. But we can provide a recap chart:

Presentation of the data base

Data base name	Information (number of observations)
BDD	All data (157 for 6 variables)
BDD_SBS	All data without the last episode of speculative bubble (145 for 6 variables)
LBDD	All data in log (149 for 6 variables)
BDD_D	All data but differentiate (156 for 6 variables)
BDD_2	All log data without the 2 speculative bubbles (140 for 6 variables).

Why do we create BDD_SBS?

Thanks to a SADF test that we develop later, we see that in the last weeks there is a speculative bubble over the bitcoin market price. Thus, we create a new data base without the last bubble to see if our estimation differs.

Why do we create LBDD?

A raintest over the regression "RegTotSBS" concludes that our model is not linear. To make it linear we log-linearize our model. That's why we create LBDD in which all variable is express in log.

Why do we create BDD_D?

After a KPSS test and a diagnostic graph we conclude that our variables are not stationary. To make it stationary, we differentiate them. That's why we create BDD_D in which all variables are differentiate.

Note that we have made a mistake (which is now fix): we've created a data base that was differentiate and then log-linearize. But the fact is that a variable differentiate could be negative so that we apply log on a negative variable. That's mathematically impossible.

Descriptive statistics

In this part, we're going to present and describe descriptive statistics of our data base.

Univariate analysis

- Statistical moments

For Bitcoin

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3285	6514	8354	9235	10152	40171

For Litecoin

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
23.61	47.21	59.84	76.17	88.07	227.89

For Ripple

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.1460	0.2435	0.3075	0.3723	0.4547	1.2200

For Euro

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.069	1.115	1.137	1.147	1.177	1.246

For Livre

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.164	1.271	1.298	1.299	1.321	1.424

For Yen

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.8785	0.9053	0.9214	0.9219	0.9364	0.9689

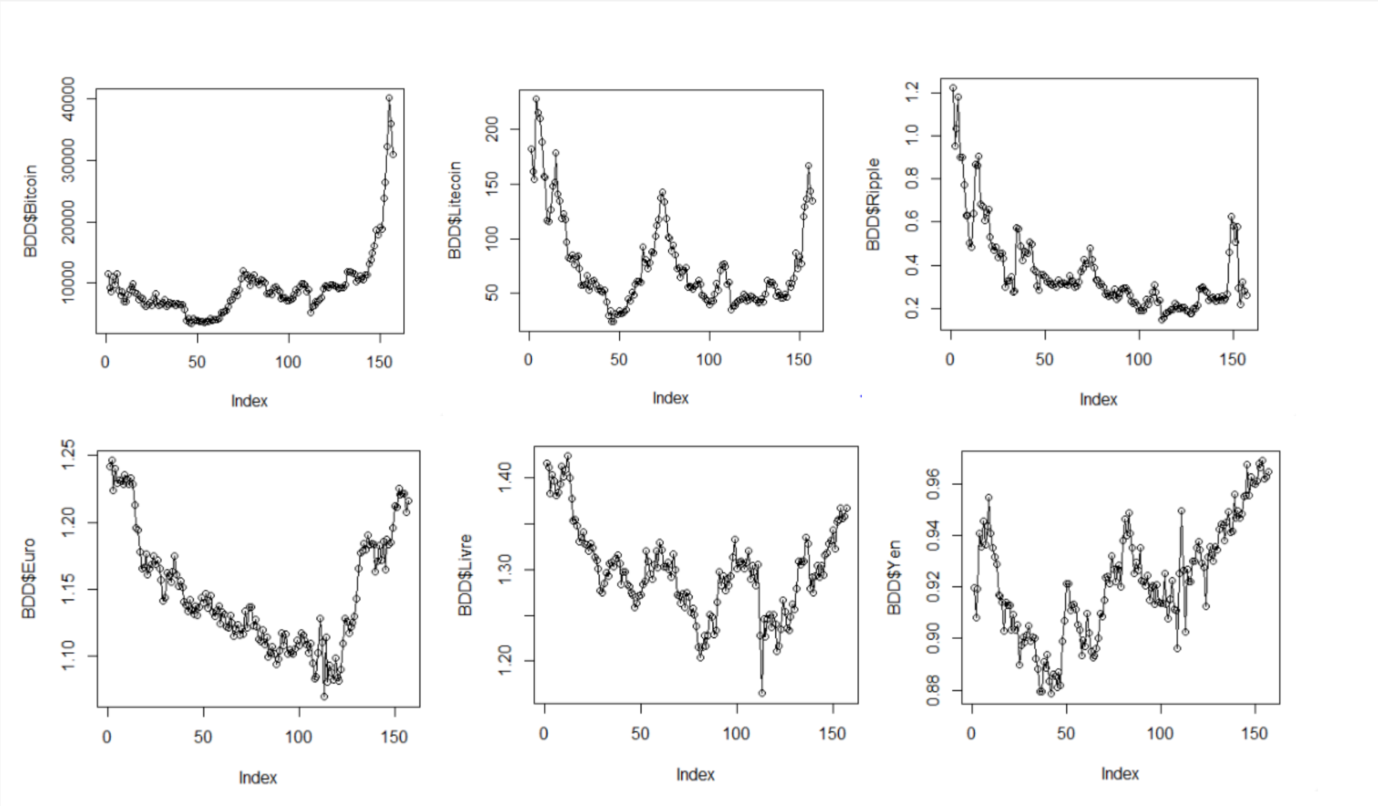
We notice that the Bitcoin is the only variable with a very high median. We also note that Bitcoin and Litecoin have a huge gap between the 3rd quarter and the maximum. It's possible that we have extreme values. We also see that for Ripple, Euro, Livre and Yen the first and third quartile are close which means that the price is stable.

Variable	Mean	Var
Bitcoin	9234.571	29861380
Ripple	0.372265	0.03971341
Litecoin	76.16578	1747.481
Euro	1.147443	0.001837046
Livre	1.298536	0.002324577
Yen	0.9219424	0.0004815111

We note that the variance of Bitcoin and Litecoin are very high ie variables are extremely volatile. We'll differentiate our data so that our relation will be stationary.

Descriptive statistics

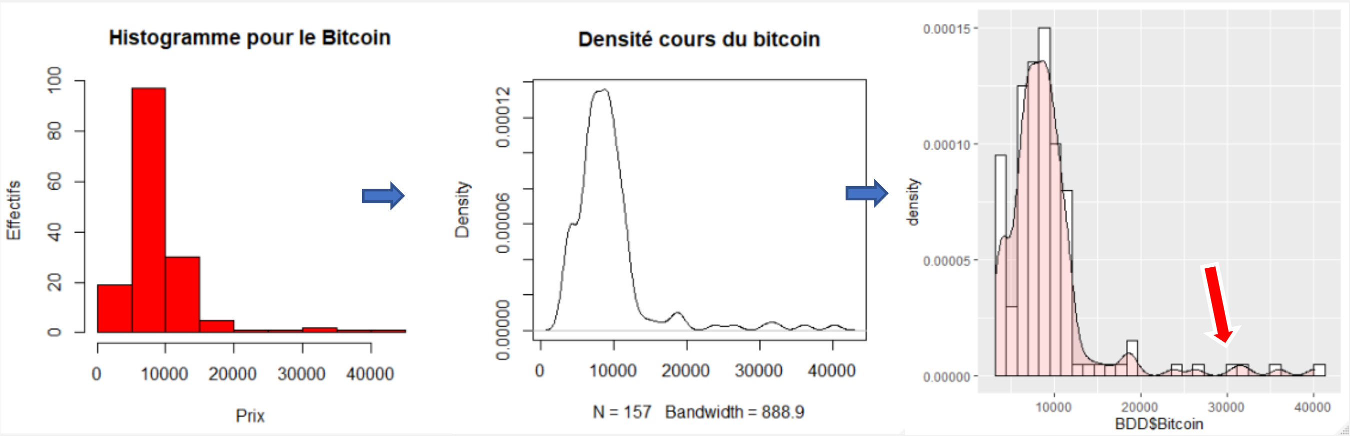
- Plot



Note: graphs above represent the market price of all currencies. We note that all series have a trend increase (especially for Bitcoin or Yen) or a trend decrease (especially for Ripple or Euro at the beginning). Those series don't seem to be stationary neither at the first order nor at the second order.

- Histogram

The main objective of histograms is to know what is the behavioral of the density of a variable. We summarize all both histogram and density in the last graph. From this graph, we deduce that the bitcoin market price tends to be around 10, 000 but sometimes it's far higher (see red arrow).

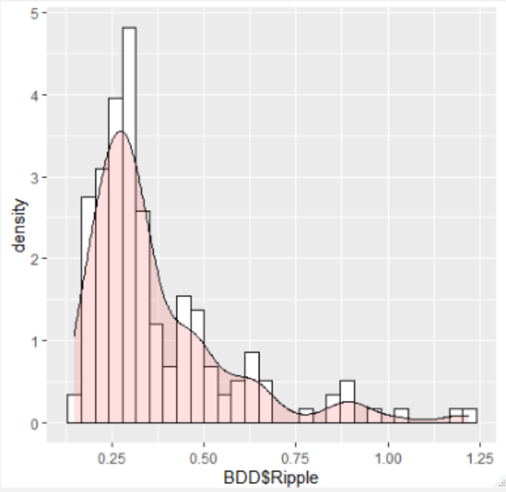


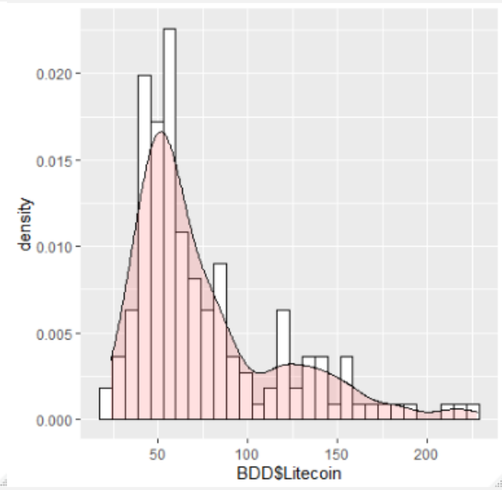
Descriptive statistics

We see that the distribution of the bitcoin price is not symmetric, mesokurtic and normal. On the contrary, the distribution is asymmetric, leptokurtic and has a distribution tail. Those assumptions are confirmed by the following test:

Skewness (BDD$Bitcoin) # Sk = 3,13 > 1,96 => asymmetric distribution
Kurtosis (BDD$Bitcoin) # Kr = 15,33 > 1,96 => leptokurtic distribution (Kr>3)
Jarque.test(BDD$Bitcoin) # residuals are non-normal distributed

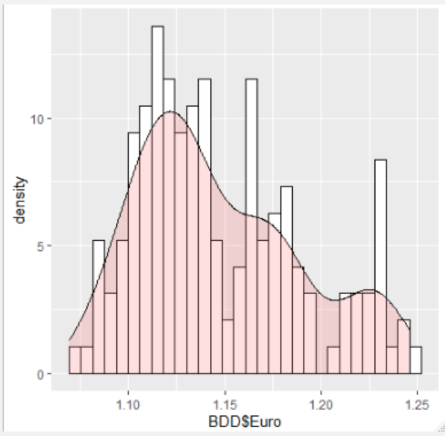
Same conclusion can be made with other cryptocurrencies:

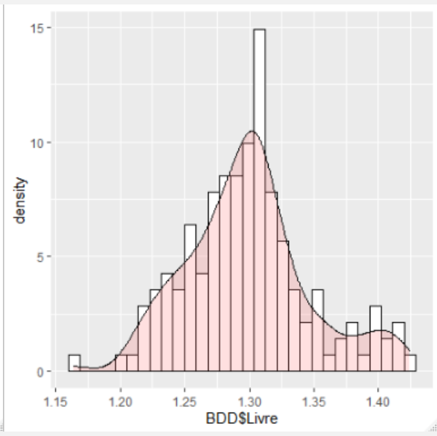


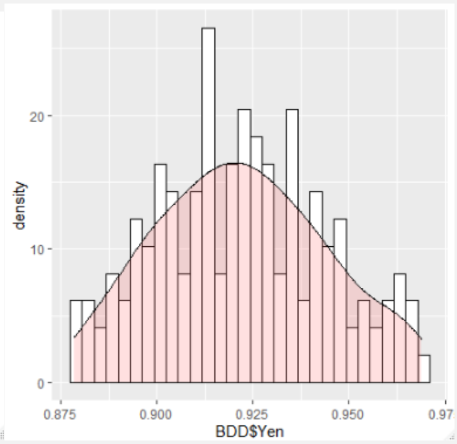


This imply that there is some extreme value. But we'll cope with this issue later.

Now what about historical currencies ? We see on the following graph that their price is more stable especially for the pound (around 1,3/$). The Euro tends more to appreciate than depreciate.







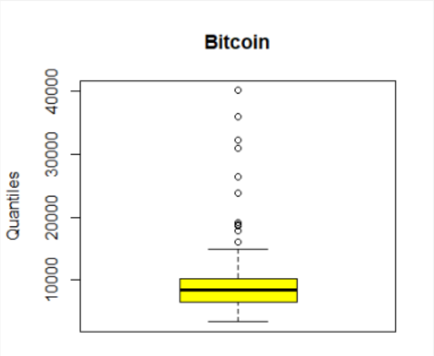
Nonetheless, we suspect the presence of extreme value. Because the bitcoin could be a store value in tension time, it's interesting to keep those extreme value during the modelling only if those values don't skew our estimation.

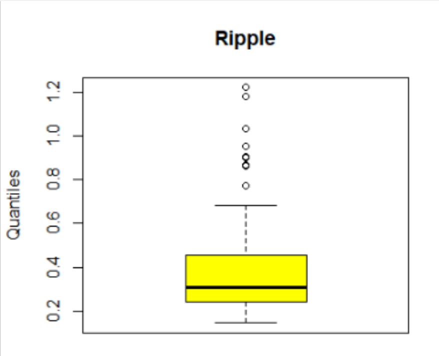
Descriptive statistics

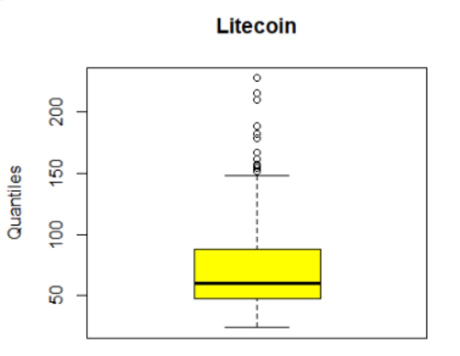
- Boxplot

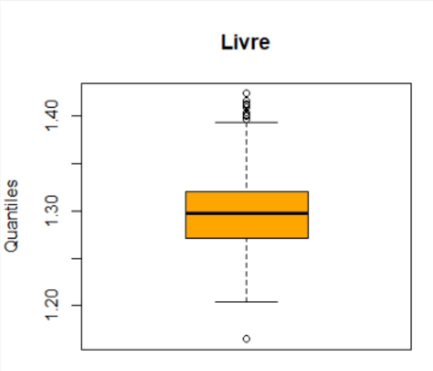
The main idea of boxplot is to detect presumed extreme values and, by the same time, to represent median, quartile and others. Then we divide our results in two categories:

1st category: suspicion of extreme values





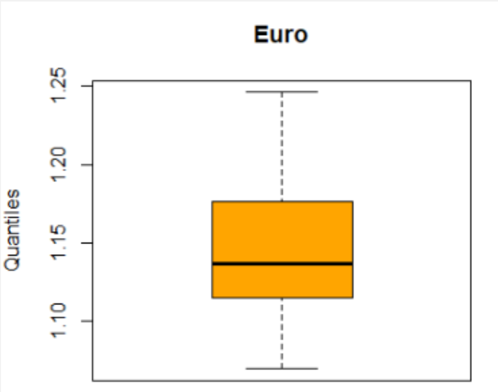


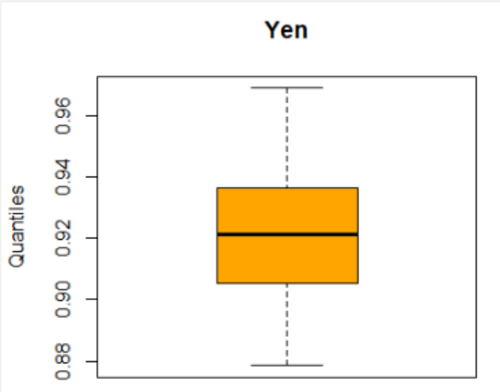


We remark that there is a lot of extreme value. Indeed, numerous values are above the extremity of the boxplot.

We are going to study those points in deeper detail with scatterplot.

2nd category: no suspicion of extreme value





We see that none of our variables seemed to have extreme values. Indeed, no point is above or below the extremity of the boxplot.

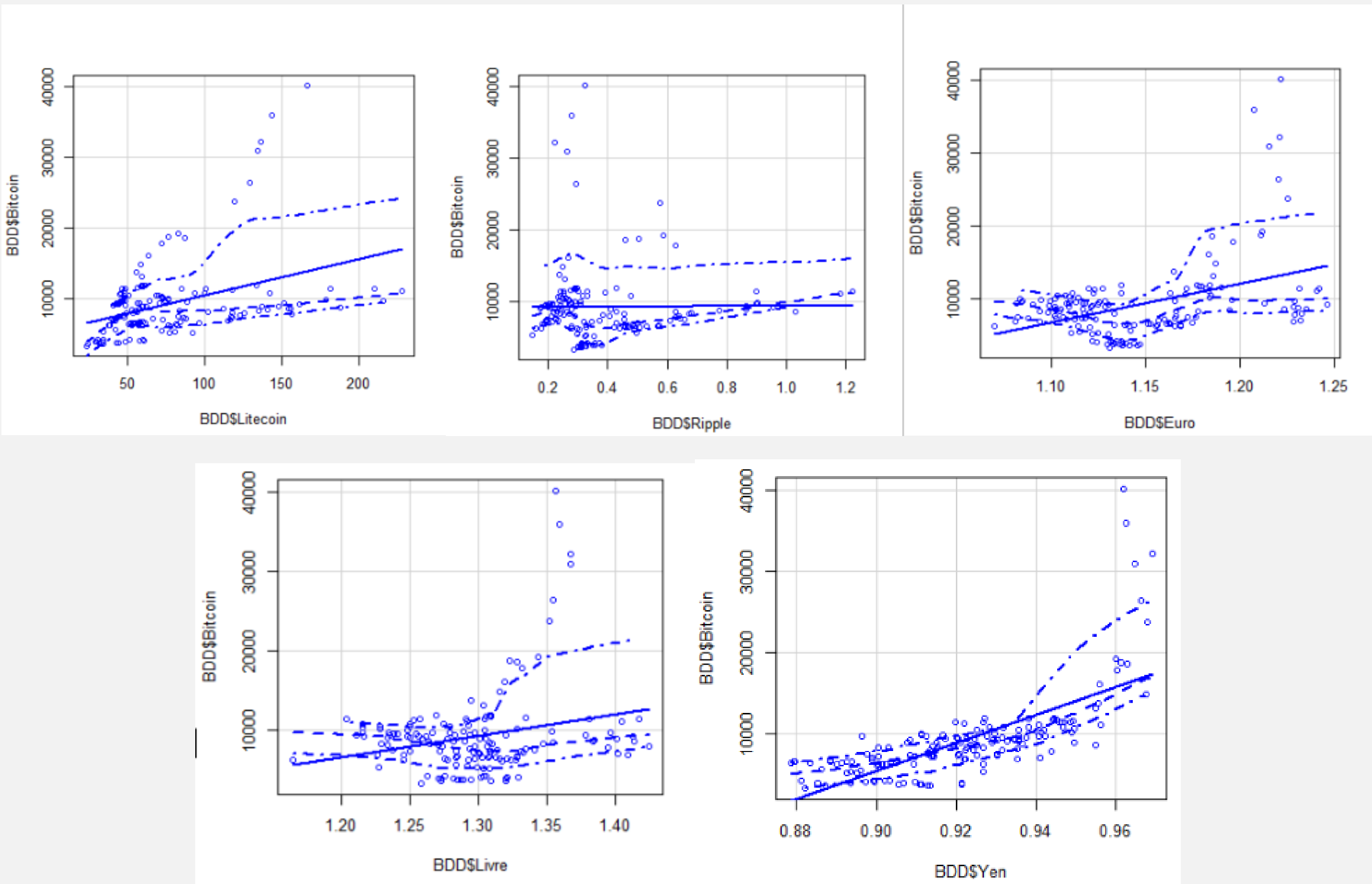
The boxplot is large, which means that the price is stable.

Descriptive statistics

Bivariate analysis

- Scatterplot

We use scatterplot to see if our extreme values skew (or bias) our regression. Here, we compute bivariate statistics.



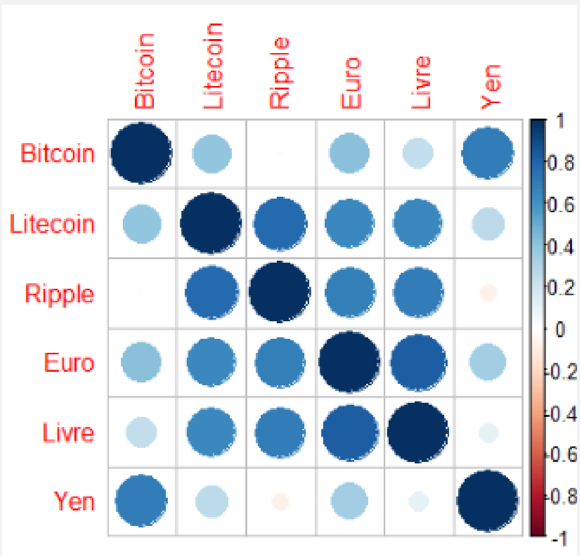
If the solid ligne in the figure is outside the corridor delimited by the two stippled lines, the relation is skew. Oddly it appears that extreme values don't skew the bivariate regression. Maybe, this is due to the number of data which is high and thus restore the regression.

- Correlation

We note that Bitcoin and Ripple seem to be few correlated.

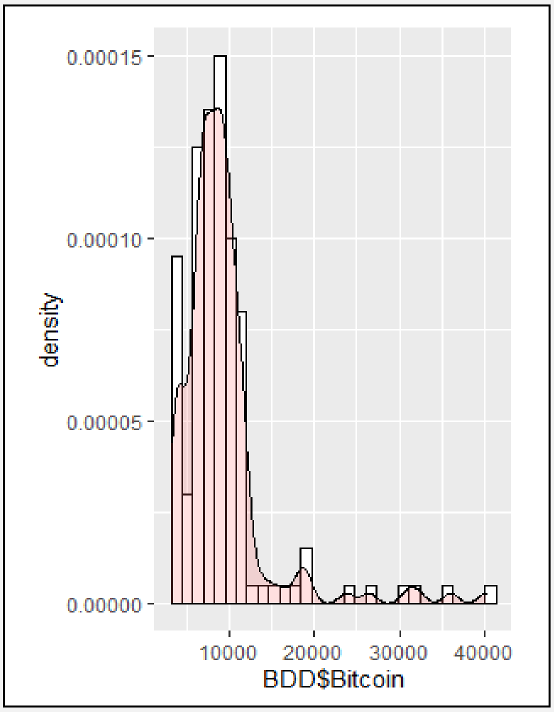
The Yen and Bitcoin are very correlated (around 0,6). We can explain that by the fact that the Bitcoin was developed by Satoshi Nakamoto in Japan. The Japan Central Bank recognize cryptocurrencies as a means of payment since 2017. The Japanese transaction platform Mt box gathered more that 75% of the exchange.

All currencies are correlated one another. We may have multicollinearity problem.



Speculative bubbles in Bitcoin markets?

- In this part, we'll test the presence of exploding models. This issue is current with models that are leptokurtic and asymmetric with a right distribution tail. We can remark this on the following graph:



How do we proceed?

We use the Philipps, Wu, and Yu test with the following hypothesis: H0: non-exploding model vs H1: exploding model.

Firstly, the software must detect that we have time series data. Therefore, we compute:

ADFBtc = ts(data = BDD[,1])

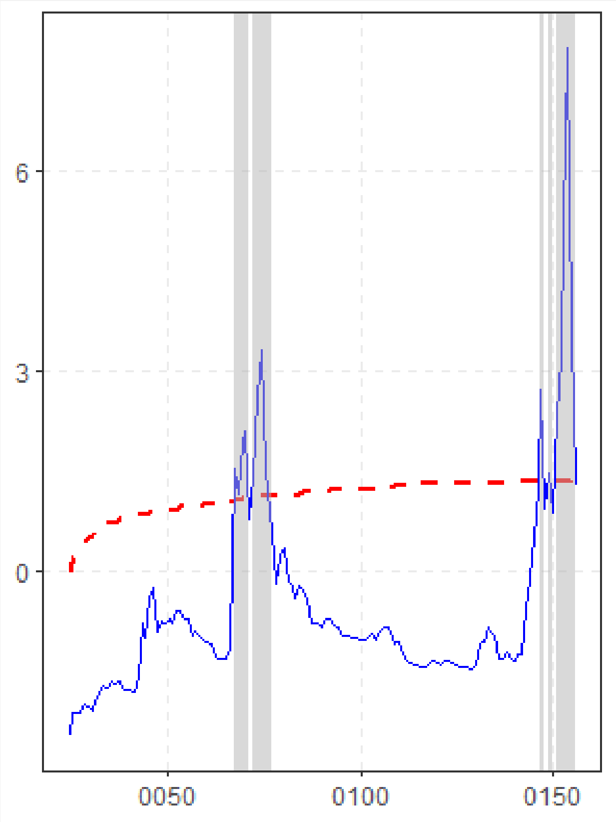
plot(ADFBtc)

The last command gives us the price of Bitcoin over the past two years but in time series (ts).

Then we compute the test:

sadf_gsadf(ADFBtc, adflag = 1,mflag = 1, IC=1, parallel = FALSE)

Results



When the empirical SADF statistic (in blue) is over the theorical SADF (in red), there is a speculative bubble. Those periods are characterized by green area.

Over the past 2 years, there've been 2 major speculative bubbles. The first around the 70th weeks and the second recently, during the past 10 weeks.

Note that the first bubble didn't have a strong impact on Bitcoin prices (regarding bitcoin price's graph). Conversely, the -recent- second bubble explodes the bitcoin market price. The second bubble has been more powerful because SADF coefficient is higher.

This result is interesting. Indeed, now we can create two different data bases. One that includes the speculative bubbles and another without them. Because a speculative bubble is defined as situation when the market value is higher than the real value of the asset, create those 2 databases allows us to estimate the real value of Bitcoin and then know when the bitcoin is overvalued.

First model, explaining the market price of bitcoin by both cryptocurrencies and historical currencies.

Firstly, we're going to make some naive regression with our first data base. We do that to better understand what type of relation between our variables is.

Results of a "naïve" regression:

Bitcoin ~ cryptocurrencies

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    7152.52     765.59   9.343  < 2e-16 ***
BDD$Litecoin    127.10      13.25   9.590  < 2e-16 ***
BDD$Ripple   -20411.70    2780.07  -7.342 1.14e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4352 on 154 degrees of freedom
Multiple R-squared:  0.374,     Adjusted R-squared:  0.3659
F-statistic:    46 on 2 and 154 DF,  p-value: < 2.2e-16
```

First and foremost, the model is significant according to Fisher test. Cryptocurrencies explain 37% of the variation of the bitcoin price.
All variables are significant.
An increase in 1 unit of Litecoin (which is enormous) implies an increase of 127 units of Bitcoin.

bitcoin ~ historical currencies

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -165849      15015 -11.045   <2e-16 ***
BDD$Euro       21456      14429   1.487    0.139
BDD$Livre       3606      12092   0.298    0.766
BDD$Yen       158125      15849   9.977   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3821 on 153 degrees of freedom
Multiple R-squared:  0.5204,    Adjusted R-squared:  0.511
F-statistic: 55.34 on 3 and 153 DF,  p-value: < 2.2e-16
```

First and foremost, the model is significant according to Fisher test. Historical currencies explain 51% of the variation of the bitcoin price, so more than cryptocurrencies.
Only intercept and Yen are significant.
An increase in 1 unit of Yen (which is enormous) implies an increase of 15 8125 units of Bitcoin, result which is odd.

Bitcoin ~ all currencies

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -122612.98   15955.99  -7.684 1.81e-12 ***
BDD$Litecoin      76.13      12.40   6.141 6.92e-09 ***
BDD$Ripple    -17847.72    2969.84  -6.010 1.34e-08 ***
BDD$Euro       53643.65   14414.63   3.721 0.000279 ***
BDD$Livre      -8120.36   11228.08  -0.723 0.470664
BDD$Yen        88600.82   17725.68   4.998 1.58e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3389 on 151 degrees of freedom
Multiple R-squared:  0.6276,    Adjusted R-squared:  0.6153
F-statistic:  50.9 on 5 and 151 DF,  p-value: < 2.2e-16
```

First and foremost, the model is significant according to Fisher test. Currencies explain 61% of the variation of the bitcoin price.
All variables are significant except pounds.

An increase in 1 unit of Euro (which is enormous) implies an increase of 53 643 units of Bitcoin, result which is odd.

- Then we made the same regression but on BDD_SBS (without the last speculative bubble)

First, the model is significant according to Fisher test. Currencies explain now 60% of the variation of the bitcoin price.
Ripple is no longer significant. It's possible that the Ripple helps to understand speculative bubble only.
An increase in 1 unit of Yen (which is enormous) implies a decrease of 100 328 units of Bitcoin, result which is odd.

```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -75249.306   8962.016  -8.396  4.1e-14 ***
BDD_SBS$Litecoin      8.972      7.877   1.139    0.257
BDD_SBS$Ripple     1830.618   1991.742   0.919    0.360
BDD_SBS$Euro       2467.133   8014.561   0.308    0.759
BDD_SBS$Livre    -10018.712   5887.167  -1.702    0.091 .
BDD_SBS$Yen      100328.988   9801.753  10.236  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1770 on 143 degrees of freedom
Multiple R-squared:  0.6026,    Adjusted R-squared:  0.5887
F-statistic: 43.36 on 5 and 143 DF,  p-value: < 2.2e-16
```

First model

Because values are strange, we have tested our last model.

- *raintest (RegTotSBS) => reject H0: the model is not linear as it was the case with regression on the initial BDD*
- *dwtest(RegTotSBS) => reject H0: residuals are correlated*
- *bptest(RegTotSBS) => reject H0: residuals are heteroscedastic*
- shapiro.test(RegTotSBS$residuals) => reject H0 : residuals are not normal
- vif(RegSBS) => no multicollinearity

So, this model is not satisfactory. The first next step is to linearize the model. To do so, we log-linearize our data base. Thus, we have LBDD_SBS which is log (BDD) without the last speculative bubble.

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     5.40727    0.38521  14.037  < 2e-16 ***
LBDD$LLitecoin  0.70251    0.07504   9.362  < 2e-16 ***
LBDD$LRipple   -0.55617    0.08024  -6.931 1.24e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2831 on 146 degrees of freedom
Multiple R-squared:  0.3753,	Adjusted R-squared:  0.3667
F-statistic: 43.85 on 2 and 146 DF,  p-value: 1.216e-15
```

First, the model is significant. All variables are significant too. This model explains 36% of the bitcoin price variation. An increase in 1% of the Ripple Price implies a decrease of 0,55% of the Bitcoin price, which seem more reliable.

First, the model is significant. Only the intercept and the Yen price are significant. This model explains 50% of the bitcoin variation.
An increase in 1% of the Yen Price implies an increase of 11 % of the Bitcoin price. The Yen has a huge role in the establishment of the Bitcoin price

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.8785     0.1712  57.712   <2e-16 ***
LBDD$LEuro    0.7071     1.0853   0.651    0.516
LBDD$LLivre  -0.2584     1.0231  -0.253    0.801
LBDD$LYen    11.3531     1.0202  11.129   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.251 on 145 degrees of freedom
Multiple R-squared:  0.5125,	Adjusted R-squared:  0.5024
F-statistic: 50.81 on 3 and 145 DF,  p-value: < 2.2e-16
```

First, the model is significant. Only the euro is not significant. This model explains 62% of the bitcoin variation, which is satisfactory.
All variables, except the Euro, are significant at 5%.
An increase of 1% of the pounds price decreases by 2,16% the Bitcoin price while an increase of 1% of the Litecoin price increases the bitcoin price by 0,42%.

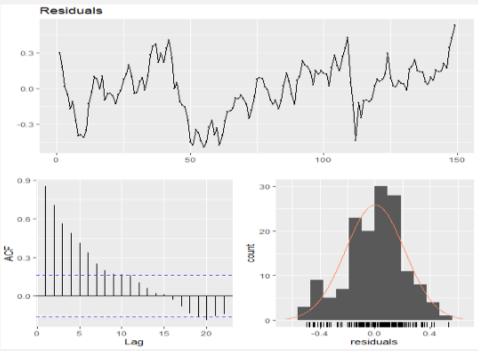
```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     8.03823    0.53657  14.981  < 2e-16 ***
LBDD$LLitecoin  0.42341    0.07216   5.868 2.95e-08 ***
LBDD$LRipple   -0.21137    0.10608  -1.993   0.0482 *
LBDD$LEuro      1.50835    1.19072   1.267   0.2073
LBDD$LLivre    -2.16272    0.93227  -2.320   0.0218 *
LBDD$LYen       8.68325    1.22700   7.077 6.09e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2184 on 143 degrees of freedom
Multiple R-squared:  0.6358,	Adjusted R-squared:  0.6231
F-statistic: 49.93 on 5 and 143 DF,  p-value: < 2.2e-16
```

RegLtot

Is this model satisfactory?

First note that this model is linear according to rain.test. Indeed, the p-value is above 5% so we accept the null hypothesis. But here the following result:

- dwtest(RegLTot) => reject H0 : residuals are correlated
- bptest(RegTotSBS) => reject H0 : residuals are heteroscedastic
- shapiro.test(RegLTot$residuals) => reject H0 : residuals are normal distributed at 1%
- vif(RegTotSBS) # no value > 10 => no multicollinearity

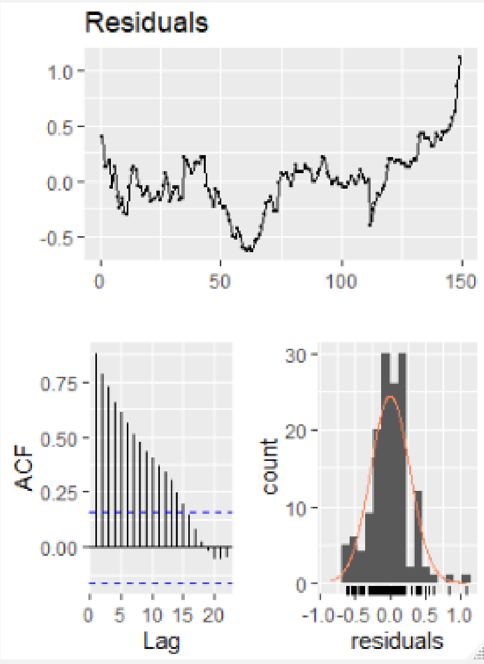


- Residuals don’t have a mean equal to 0 and their levels vary.
- The ACF (autocorrelation function) has residuals significant up to the 12th lag.
- Residuals don’t have a normal distribution.

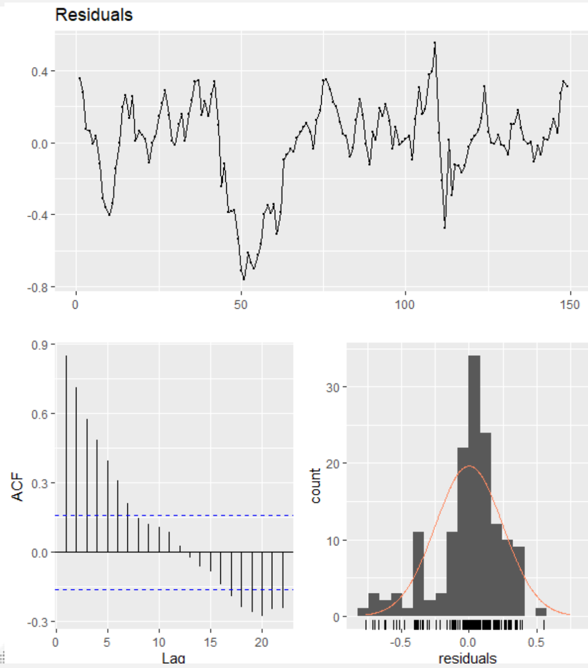
First model

We can make the same conclusion:

For Log(bitcoin) ~ Log(cryptocurrencies)

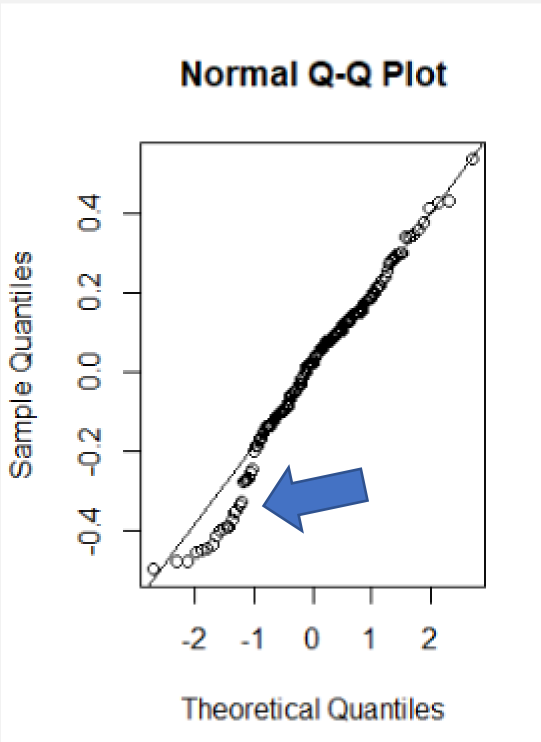


For Log(bitcoin) ~ Log(historical currencies)

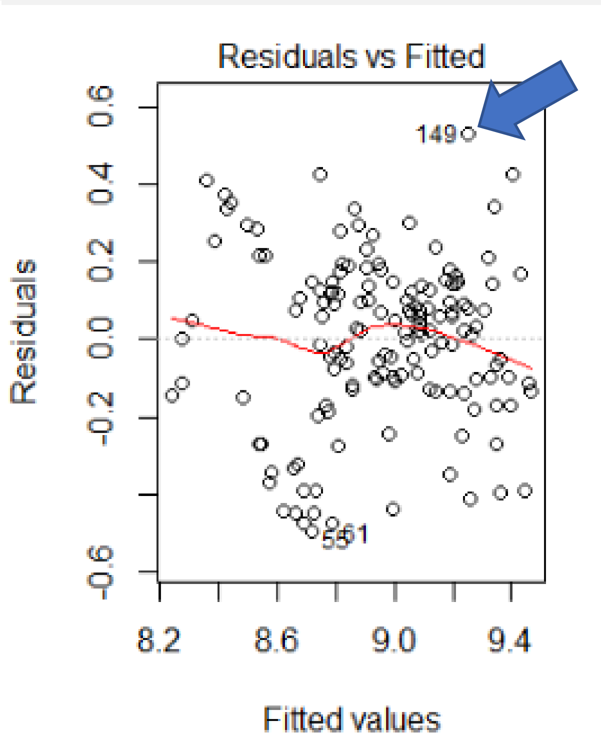


- Let's see residuals of our main regression (RegLtot) into deeper details:

(1)



(2)



The first graph (1) allows us to compare the residuals of RegLtot (the regression of all currencies with log data base). We can see that the major part of our residuals is linear. We also see that residuals of the lower quantiles don't follow a normal distribution.

First model

The second graph allows us to compare the real residuals and those predict by the model. We can see that the regression is not so good and their still some extreme value (see the arrow). Let's keep in mind that we have cut the bubble according to the SADF test, but It could have been better to cut the bubble sooner than what the test suggests. Indeed, we can see on the bitcoin market price graph that the increase starts earlier.

If we do that (creating LBDDS for log (BDD) without speculative bubble earlier) then we obtain the following results:

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        7.31885    0.54730  13.373  < 2e-16 ***
LBDDS$LLitecoin    0.52375    0.07275   7.200 4.00e-11 ***
LBDDS$LRipple     -0.30227    0.10860  -2.783  0.00616 **
LBDDS$LEuro        1.41526    1.19169   1.188  0.23710
LBDDS$LLivre      -2.30381    0.89304  -2.580  0.01097 *
LBDDS$LYen         6.00183    1.29977   4.618 9.05e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.208 on 133 degrees of freedom
Multiple R-squared:  0.6154,    Adjusted R-squared:  0.6009
F-statistic: 42.56 on 5 and 133 DF,  p-value: < 2.2e-16
```

RegLtotS

This model is satisfactory according to the F-statistic and this model is linear. But :

dwtest(RegLTotS) # residuals are correlated

bptest(RegLTotS)# residuals are heteroscedastic

shapiro.test(RegLTotS$residuals)# residuals are not normal distributed

vif(RegLTotS) # no value > 10 => no multicollinearity

Which model do we prefer?

```
> AIC(RegLTot)
[1] -22.61434
> AIC(RegLTotS)
[1] -34.13266
```

Thus, we prefer RegLTotS ie this equation:

Bitcoin = 7,31*+ 0,52*Litecoin - 0,3*Ripple -2,3Pounds*+6*Yen+ resid

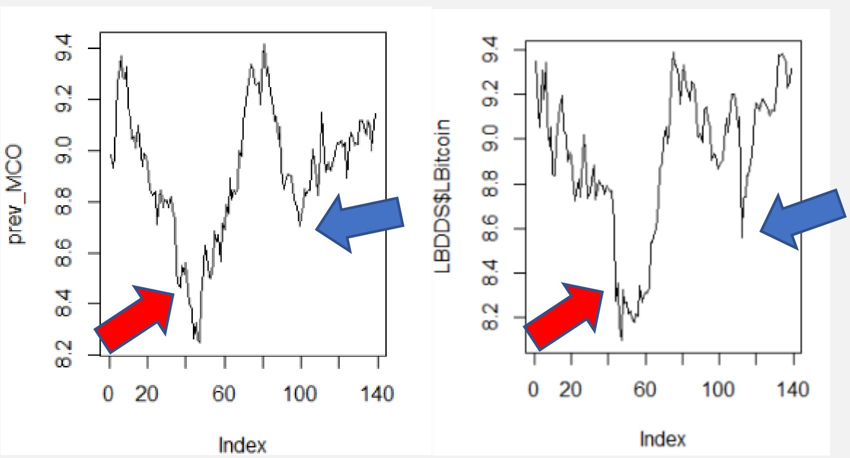
Sign. at 5% = * and 10% = **

Data Base: Log-linearize variables between January 21st, 2018 and November 24th, 2020.

- Interpretation

All interpretations are in percentage because of the log-linearization. All things being equal, if the pounds price increases by 1% then the Bitcoin price is reduced by 2,3%. All things being equal, if the Litecoin price increases by 1% then the Bitcoin increases by 0,52%. We note that the Yen price has a huge role in explaining the variation of the Bitcoin price. Indeed, all things being equal, if the Yen price increases by 1% then the Bitcoin price increases by 6%

- Forecasting



MAPE (Mean absolute percent error): 0.01894416% / MAE (Mean Absolute Error): 0,16

Firstly, our model understands the main market trend before the 60th week.

The model estimates the bitcoin price with a Mean Absolute Error (MAE) of 0,16%. But MAE depends on the unit measure. Thus, we us the MAPE which is good: a standard error of 0,01%.

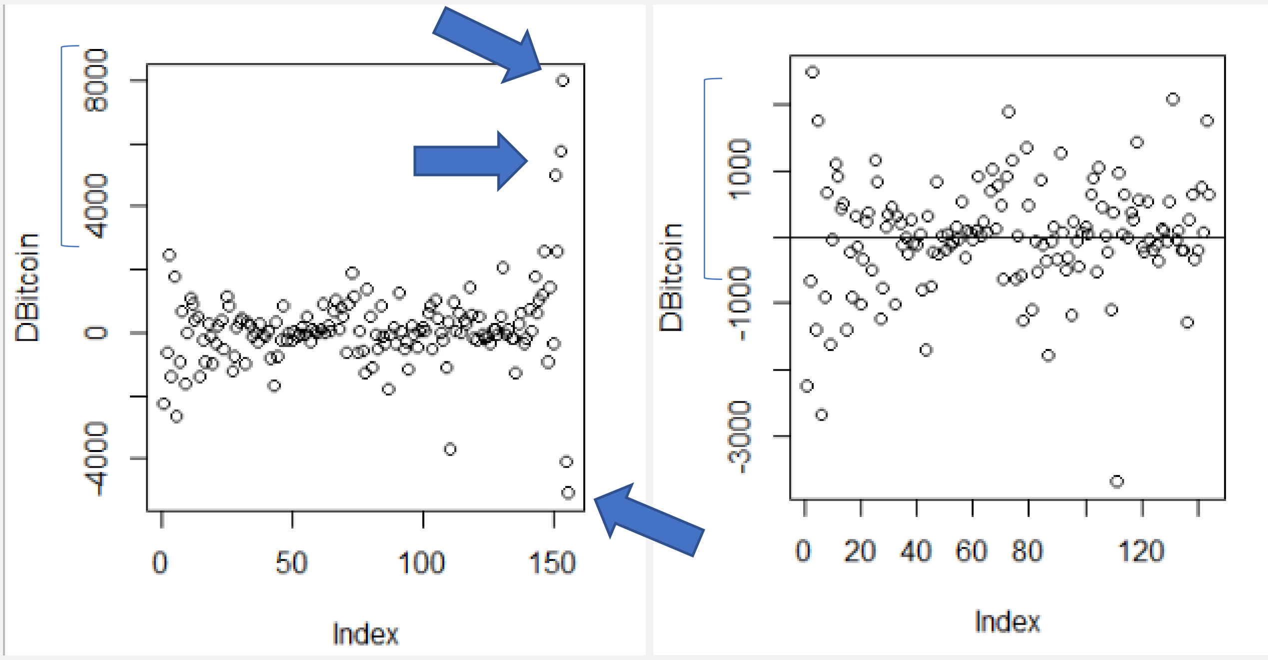
But it's also seemed to be strange. Indeed, the model doesn't remark two drops (in red and blue). The model tends to minimize the range of variation.

What causes what? Granger causality test

Tests on our model's residuals are not satisfactory. This means that there still important information in our residuals. Thus, we can wonder if we had taken the relation in the right causality direction.

Setting up the test

At the begging, we diffrentiate over BDD but this data base include extreme values (see blue arrows). So we cut the bubble just as we've did before.



Before all, we need to be sure that all variables are stationary. To do so, we use both KPPS and Philipp-Peron (PP) tests. Those tests confirmed our graph analysis: variables are not stationary. For instance, Bitcoin is a I(1) process that needs to be differentiated to be stationary. Stationarity is important in time series so that our test will not be skew. Hence, we create a new data base with differentiate variable BDDDiff without the last speculative bubble.

```
> causality(VAR_DL, cause = "DBitcoin")
$Granger

        Granger causality H0: DBitcoin do not Granger-cause DLitecoin DRipple DEuro DLivre DYen

data:  VAR object VAR_DL
F-Test = 1.9413, df1 = 5, df2 = 888, p-value = 0.08514


$Instant

        H0: No instantaneous causality between: DBitcoin and DLitecoin DRipple DEuro DLivre DYen

data:  VAR object VAR_DL
Chi-squared = 59.589, df = 5, p-value = 1.478e-11
```

Results

First, we test H0: differentiate bitcoin do not Granger-Cause cryptocurrencies and historical currencies. Because the p-value is above 5%, we accept H0.

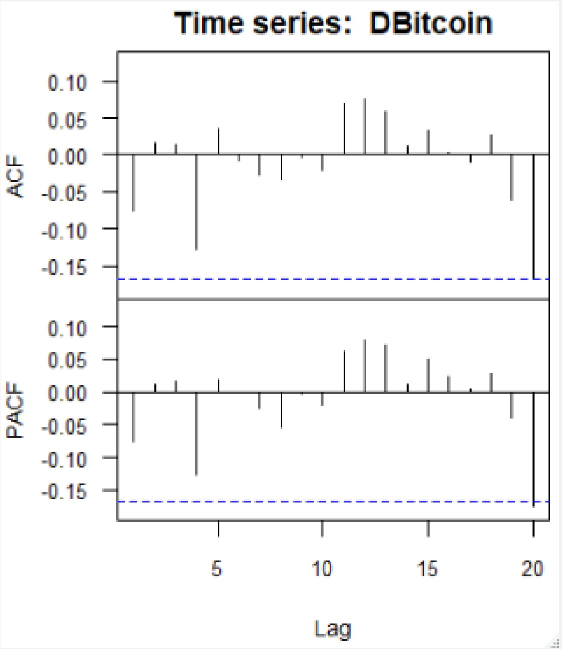
Secondly, we test H0: there is no instantaneous causality between DBitcoin and both cryptocurrencies and historical currencies. Because the p-value is below 5%, we reject H0, so there is an instantaneous causality between DBitcoin and both cryptocurrencies and historical.

The Granger causality test gives us the following result: Bitcoin price doesn't influence other currencies, but other currencies influence instantaneously the Bitcoin price.

Explaining bitcoin price by itself: ARIMA model

Thanks to the granger causality test, we know that we've taken the regression in the right order. But there is a variation of mean and variance over time, we can suspect the presence of an ARIMA model. Before all, we need to be sure that our variables are stationary. We see on (1) graph that for any lags, the ACF and PACF is equal to 0. This means that our model is not a random walk. Thus, there exists an ARIMA model to represent our model. Then we use an ARMA function (see 2) to shortlist our ARIMA models. The result gives us 3 ARIMA models' candidate: ARIMA (0,1,1); ARIMA (1,1,2); ARIMA (1,1,1).

Graph (1)



Graph 2

```
> armaselect(DBitcoin,max.p=20,max.q=20,nbmod=10)
```

```
      p q      sbc
 [1,] 0 1 1911.763
 [2,] 1 1 1914.603
 [3,] 0 2 1917.631
 [4,] 2 1 1919.386
 [5,] 1 2 1919.990
 [6,] 0 3 1923.081
 [7,] 3 1 1924.350
 [8,] 2 2 1924.444
 [9,] 1 3 1925.560
[10,] 0 4 1926.194
```

Graph 3

SBC : Schwartz criterium

Then we apply a trial and error methods. We use AIC determining the best ARIMA model. None of these models are significant: ARIMA(0,1,1), ARIMA(1,1,1), ARIMA(0,1,2). But results are better for ARIMA(2,1,1) and ARIMA(1,1,2).

```
Coefficient(s):
          Estimate  Std. Error  t value Pr(>|t|)
ar1       -0.93603     0.09422   -9.935   <2e-16 ***
ar2       -0.14056     0.07580   -1.854   0.0637 .
ma1        0.83565     0.08321   10.043   <2e-16 ***
intercept 91.99756   120.97246    0.760   0.4470
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Fit:
sigma^2 estimated as 641620,  Conditional Sum-of-Squares = 90495204,  AIC = 2342.19
```

ARIMA(2,1,1)

```
Coefficient(s):
          Estimate  Std. Error  t value Pr(>|t|)
ar1       -0.83554     0.08961   -9.324  < 2e-16 ***
ma1        0.79846     0.11937    6.689 2.25e-11 ***
ma2       -0.05567     0.08291   -0.671    0.502
intercept 90.36913   116.49721    0.776    0.438
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Fit:
sigma^2 estimated as 654823,  Conditional Sum-of-Squares = 92366337,  AIC = 2345.12
```

ARIMA(1,1,2).

Explaining bitcoin price by itself: ARIMA model

The chosen ARIMA model

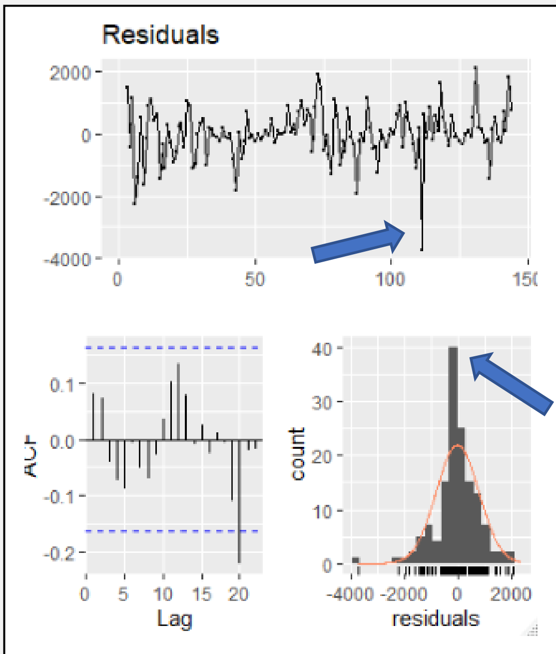
We choose ARIMA(2,1,1). The model is :

$$\Delta Bitcoin_t = \mu + \alpha \Delta Bitcoin_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}$$

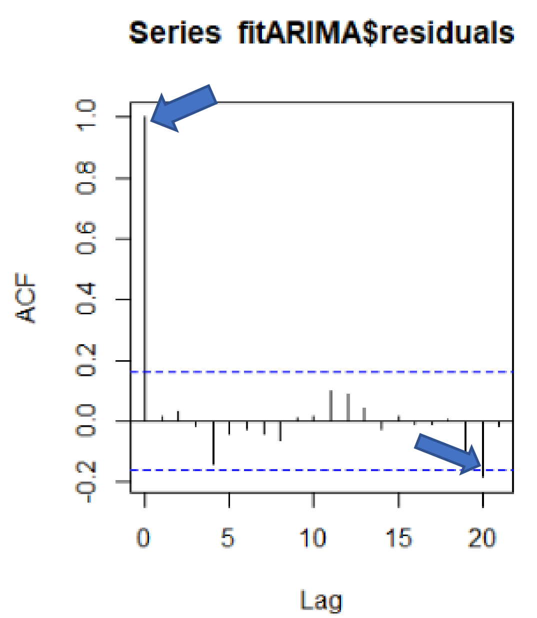
$$\Delta Bitcoin_t = 0{,}83^{*} \Delta Bitcoin_{t-1} + \varepsilon_t - 0{,}93^{*} \varepsilon_{t-1} - 0{,}14^{**} \varepsilon_{t-2}$$

Sign. at 5% = * and 10% = **

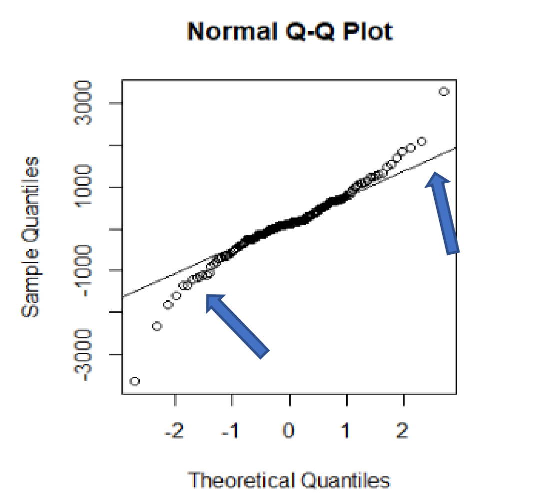
But the model has some limit. First one : residuals are volatile (see graph 1). This accordion configure is typical of self-correlated residuals. We see on graph 2 that residuals are correlated for the first and 20th lags. This is confirmed with a Ljung-Box test (see graph 4) : some lags (4th ,6th and 7th) have a p-value below 5% so we reject H0, residuals are self-correlated. Moreover, we can see on the 3rd graph that residuals have a non-normal distribution.



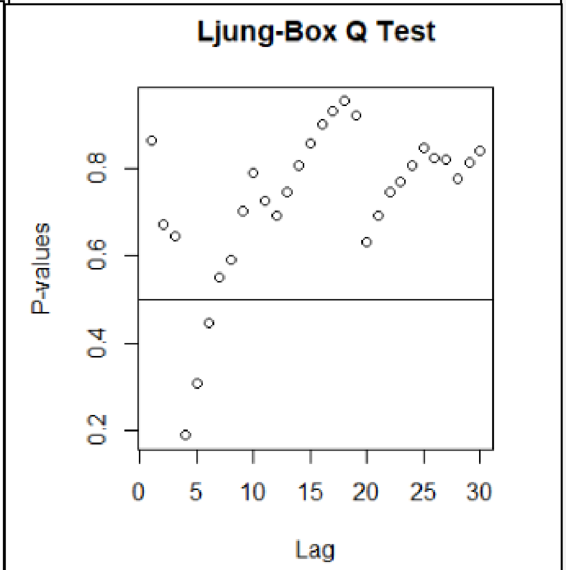
1



2



3

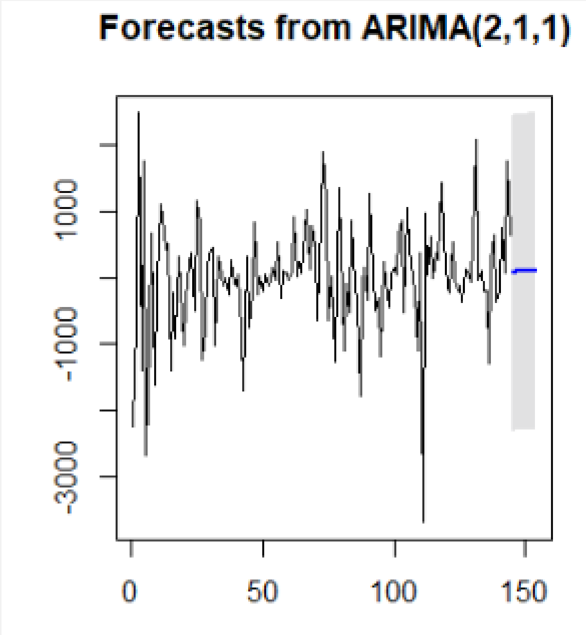


4

Explaining bitcoin price by itself: ARIMA model

Forecast

We can see on the following graph the forecast of our model. This is clearly not satisfactory : the forecast of the model is simply an extrapolation of the mean. Plus : the confidence interval is too large.



Which model is better?

To summarize, we have two models :

"RegLtotS"	Bitcoin = 7,31*+ 0,52*Litecoin - 0,3*Ripple -2,3Pounds+6*Yen + ε_t
ARIMA211	$\Delta Bitcoin_t$= $0{,}83^{*}\Delta Bitcoin_{t-1}$ + ε_t - $0{,}93^{*}\varepsilon_{t-1}$- $0{,}14^{**}\varepsilon_{t-2}$

```
> AIC(RegLTotS)
[1] -34.13266
```

```
> summary(ARIMA211)
AIC = 2342.19
```

Improving our model

We must choose the model with the minimum AIC. Thus, we choose RegLTot even though this model is not totally satisfactory. Nevertheless, the first choosing model include the first speculative bubble so that this model doesn't estimate the real Bitcoin price (ie without all speculative bubble). Then, we decide to cut both first and second bubble(BDD_2) and make a final regression :

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       8.02540    0.55071  14.573  < 2e-16 ***
BDD_2$LLitecoin   0.44614    0.07859   5.677 8.15e-08 ***
BDD_2$LRipple    -0.20423    0.10880  -1.877   0.0627 .  
BDD_2$LEuro       1.37744    1.22168   1.127   0.2615    
BDD_2$LLivre     -2.36314    0.99122  -2.384   0.0185 *  
BDD_2$LYen        8.68159    1.25820   6.900 1.88e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2229 on 134 degrees of freedom
Multiple R-squared:  0.632,	Adjusted R-squared:  0.6183 
F-statistic: 46.03 on 5 and 134 DF,  p-value: < 2.2e-16
```

AIC = -15.11

This model is linear, but residuals still correlated. Despite the change, this model is not better than RegLTot (-22 < -11).

Our final model of the bitcoin price which estimate the real (ie without the 1st speculative bubble) is :

$$\text{Bitcoin} = 7{,}31^{*} + 0{,}52^{*}\text{Litecoin} - 0{,}3^{*}\text{Ripple} - 2{,}3\text{Pounds} + 6^{*}\text{Yen} + \varepsilon_t$$

Sign. at 5% = * and 10% = **

Interpretation :

Both Ripple and Pounds increase have a negative impact on the Bitcoin market prices. Indeed, an increase by 1% of the Pounds price leads to a decrease by 2,3 of the Bitcoin prices. Litecoin price (resp. Yen) has the same impact on bitcoin price: an increase by 0,52% (resp. 6%). We also remark that the bitcoin is more sensitive to a variation of the Yen prices. We can explain that by the fact that Bitcoin's transaction platform are manly Japanese.

Our final model explains around 60% of the Bitcoin prices variation. This number may be due to the absence of the dollar. Note that this model is not satisfactory too (heteroskedasticity, residuals' correlation).

Conclusion

We have tried to explain what driven the bitcoin market price. At first, according to scatterplot visual test, our bivariate regressions are not skew by the actual extreme value. Thus, we've made some naïve regressions. Those first models were not satisfactory (non-linear).

Firstly, we decide to cut the last speculative bubble that we have determined with an SADF test. Indeed, our objective is to understand what make the real bitcoin price. That's the reason why we exclude speculative bubble; they are time when the asset's price is different than its real price. Despite this change, our regressions still skew and non-linear.

Then we decide to log-linearize our data base. Now all interpretations are in percentage and estimate coefficients seemed to be more reliable. Admittedly residuals of our first model are not satisfactory, but there is a slight improvement : at 1% residuals are normal according to shapiro.test. Then, we've decided to cut sooner the speculative bubble because we can see that the bitcoin price starts increasing before what suggest the SADF test. We were right to do that because the regression over this new data base is better. Nonetheless, tests on our last model's residuals are not satisfactory. This means that there still important information in our residuals. Thus, we can wonder if we had taken the relation in the right causality direction.

The Granger causality test confirms that we have taken the problem in the right order. So, we're now facing a problem : OLS regression isn't the right manner to model the bitcoin price. Then we develop a second model : ARIMA (2,1,1). The main idea is to explain the bitcoin price by itself. This model is not satisfactory too. Finally, we compare our two main models. The result is that the OLS regression is better.

Clearly, our chosen model is not satisfactory. Nonetheless, the main result is that the bitcoin price depends mainly of the Yen price. There are several ways to improve our model. When we were defining our ARIMA model we don't include the fact that residuals of our differentiate data remain volatile. This is typical of an ARCH/GARCH model. Because the constant is equal to zero, we can improve our model by applying an EGARCH model on our ARIMA residuals. Moreover, we have only tested the presence of speculative bubble on the bitcoin price. It will be interesting to test the presence of speculative bubble on other currencies, especially cryptocurrencies.