Financial Econometrics

FIN - 407

Homework 1: GARCH and portfolio management

Authors:
Florin Laurent
Canoni–Meynet Jeremie
Bellahsene Allan
Kuber Malhotra
Faezeh Nassajian

Supervisor:
Prof. Pelgrin Florian

Teaching Assistant:
Maxime Couvert

10th May 2019

Contents

List of Figures

Introduction

Crypto currencies have become very popular this past few years. We all notably remember the bitcoin rush in the end of 2017 which have attracted attention on these new type of assets. These digital assets notably differ from classical currency as they are not centralized. Without going into the technical details, these decentralized system are based on a distributed ledger, usually blockchain, which enable to recorded on this blockchain every movement of the cryptocurrency. The cryptocurrencies have also notably led to many legal issues, indeed, nowadays they are not considered as financial securities like a typical stock. Thus, they are not subject to all the regulatory compliance of typical assets.

In this report, we will use two crypto-currencies as long with a risk free asset to create a portfolio and implement asset allocation methods. We will notably compare two asset allocation methods: static versus dynamic. We will model the time series of the returns of these assets with GARCH models. Finally, we will also assess whether we can make some gains by forecasting correctly the volatility dynamics (”timing-volatility”).

Choice of the 2 cryptocurrencies

For doing this, we have been provided with the daily closing prices of 20 cryptocurrencies and a daily risk free rate between 01/01/2019 and 04/07/2019. We are now going to choose two currencies almost these 20 assets. First, we computed the daily log-returns of all currencies (except Tether). We then decided to consider a mean-variance approach in order to chose the two currencies to invest in. In other words, considering an exponential utility function, we solved the following maximization problem:

$$\max \mu_p - \frac{1}{2}\sigma_p^2$$

where μ_p denotes the return of the portfolio and σ_p^2 the variance of the portfolio.

We then solved this maximization problem for all possible combination of 2 crypto-currencies among the 19 at our disposal, and chose to invest in the portfolio that yielded the highest utility.

Finally, we obtained that the best possible portfolio considering the mean-variance approach was a portfolio composed by Bitcoin and Ethereum which yields a utility of 0.002475. Hence, our study will be based on a portfolio composed by those two currencies, and the risk-free asset.

We are now going to focus on the volatility of the two crpytocurrencies.

Preliminary step

1. Estimate of the implied volatility

We are now going to estimate the weekly volatility of the Bitcoin and Etherum. We will start with an EWMA model of the weekly volatility.

EWMA-based estimate of the (weekly) volatility

The EWMA-based variance at period t is given by the following formula:

$$\sigma_{i,t}^2 = (1-\lambda)(r_{i,t} - m_{i,t})^2 + \lambda\sigma_{i,t-1}^2$$

where $r_{i,t}$ denotes the daily log-return of the crypto-currency at day t, $m_{i,t}$ is the average log-return of the currency, and $\lambda \in (0,1)$ is the "smoothing parameter". The EWMA variance approach differs from the simple variance approach precisely because of this parameter, as it assumes that not all observations must have the same weight (as opposed to the simple variance approach that gives the same weight to all observations). More precisely, the EWMA variance approach gives more weight to the most recent observations. The most common value used for λ is 0.94, as given by the risk management company RiskMetrics.

Hence, setting $\lambda = 0.94$, we first computed the EWMA-based daily variance using the formula above for both currencies (setting the first value σ_0^2 equal to the simple variance). The EWMA-based weekly volatility was then computed as follows:

$$\sigma_{EWMA} = \sqrt{7\sigma_t^2}$$

Where the factor seven comes from the fact that we want weekly volatility. We thus obtained the following plots for Bitcoin and Ethereum:

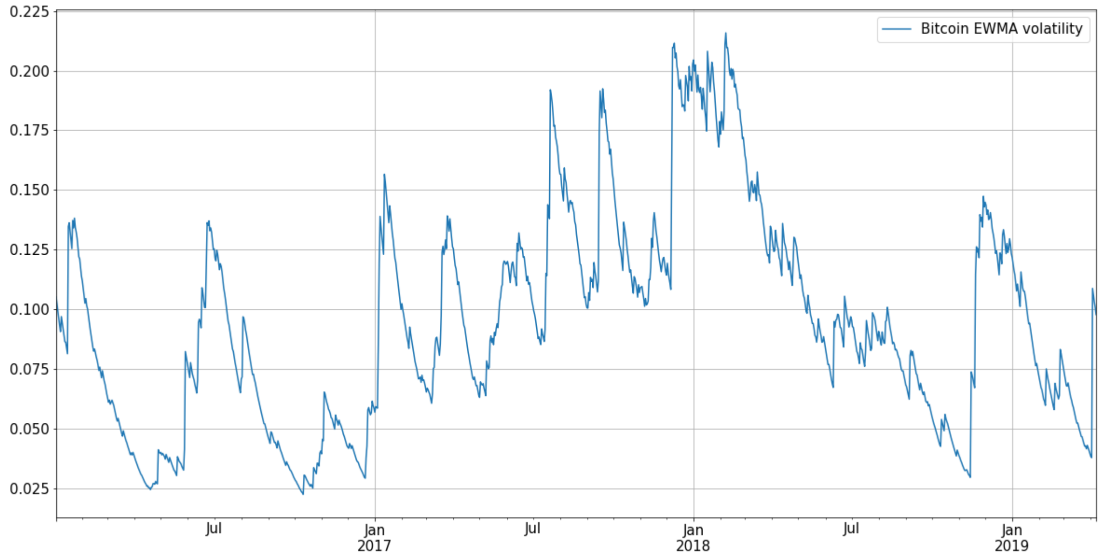

Figure 1: EWMA-based estimate of the volatility for the Bitcoin

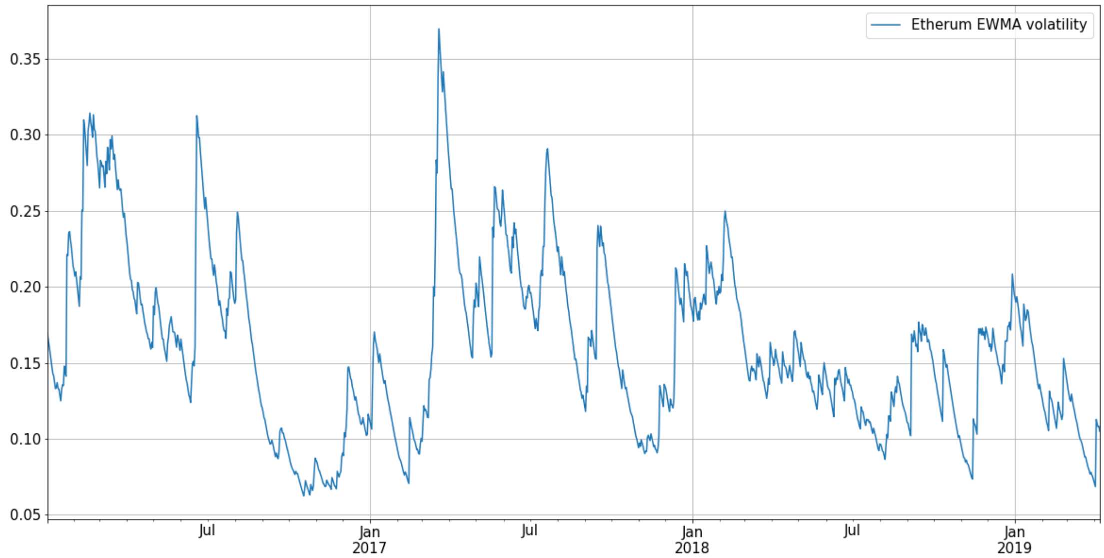

Figure 2: EWMA-based estimate of the volatility for the Etherum

Now that we have an EWMA-based estimate of the weekly volatility, we are going to model this volatility using another model with GARCH-based error terms.

GARCH-based estimate of the (weekly) volatility

The GARCH model states that the (conditional) variance of the (log-)returns at day t is given by:

$$\sigma_{i,t}^2 = \omega + \alpha\epsilon_{i,t-1}^2 + \beta\sigma_{i,t-1}^2$$

where $\epsilon_{i,t}^2 = (r_{i,t} - m_{i,t})^2$.

The parameters ω, α and β are obtained through the maximization of the Log-Likelihood function (under normality assumptions).

Hence, importing the *arch model* from the *arch* package on Python, we obtained the following values for the parameters:

$$(\omega_{bit}, \alpha_{bit}, \beta_{bit}) = (4.5422.10^{-5}, 0.1226, 0.8560)$$
$$(\omega_{eth}, \alpha_{eth}, \beta_{eth}) = (0.0004, 0.1907, 0.7229)$$

We obtained the following plots for the GARCH-based weekly volatility of both currencies:

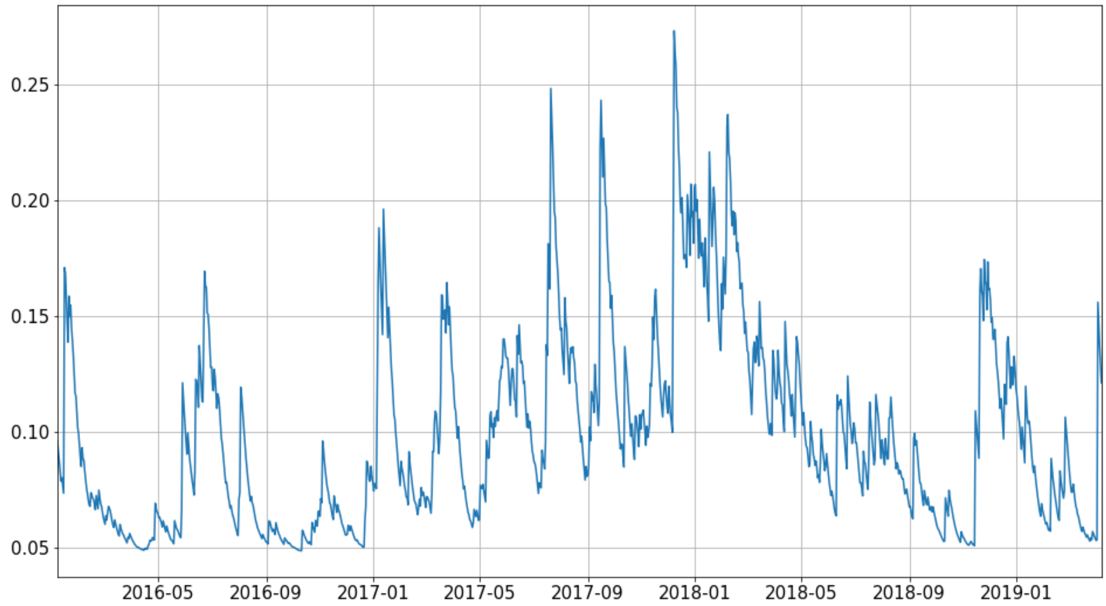

Figure 3: GARCH-based weekly volatility of Bitcoin

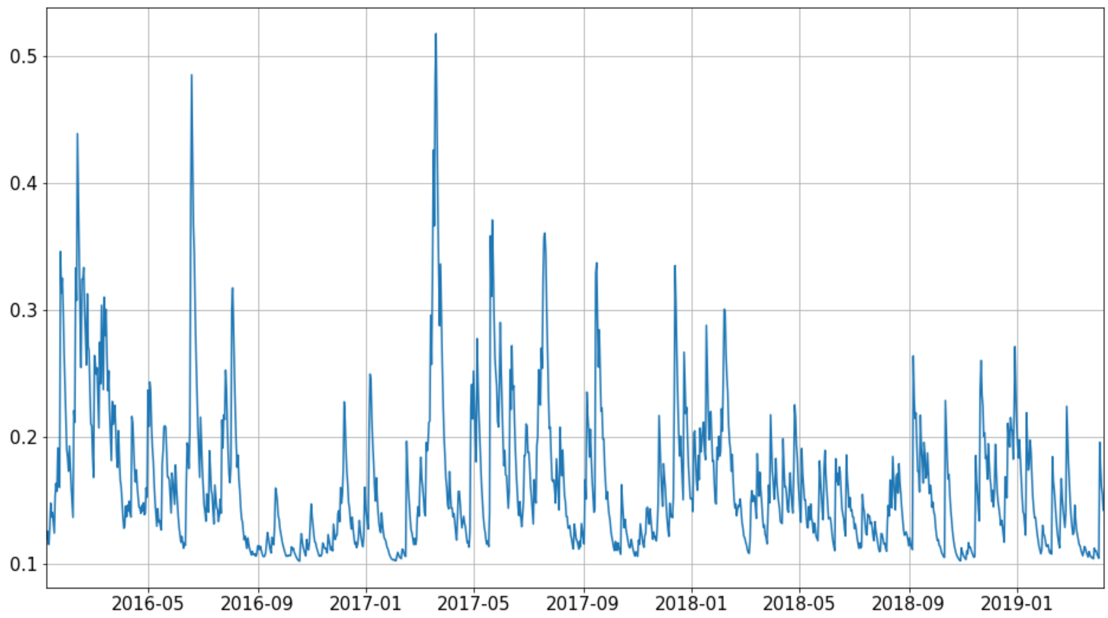

Figure 4: GARCH-based weekly volatility of Ethereum

We have now our two estimates of the volatility, an EWMA based and a GARCH based volatility. First thing that we can note from these estimations of the volatility is that the volatility obtained is relatively high, but we will go through detail later by taking standard stocks' volatility as measure of comparison.

Moreover, we can also note that the two models gave us plots with similar shape, however the scale is not exactly the same. Indeed, the EWMA tends to estimate lower value of volatility than the GARCH model.

Now, we are going to compute the term structure of volatility over the next 21 days.

Term structure of volatility over the next 21 days

The volatility term structure depicts how the implied volatility of the return of an asset is expected to evolve over a defined horizon h.

The objective here is to compute the volatility term structure of Bitcoin and Ethereum over the next 21 days. In order to do that, one first needs to compute forward variances for both currencies, which is nothing but the forecast of conditional variances for the next 21 days.

The GARCH-based forward variance is computed recursively, such that:

$$\sigma_{i,t}^2 = \omega_i + (\alpha_i + \beta_i)\sigma_{i,t-1}^2 \text{ for } i = \{\text{bit, eth}\}$$

Then, after computing the cumulative forward variance, the volatility term structure at day t is obtained by taking the square root of the cumulative forward variance at day t.

The resulting plots of the volatility term structures for Bitcoin and Ethereum are:

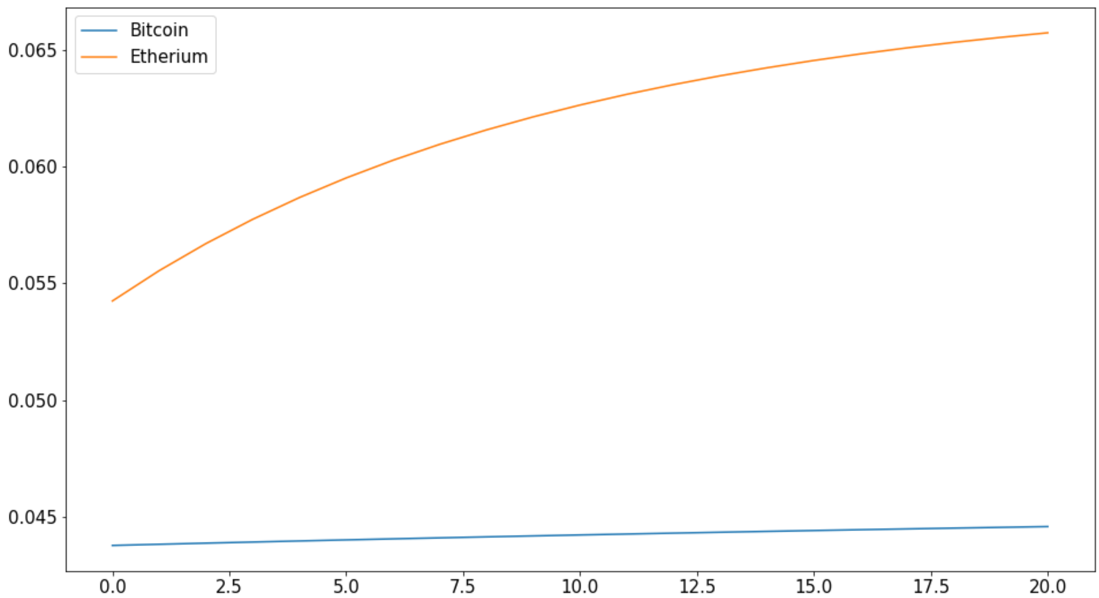

Figure 5: Conditional correlations

We can see that the volatility term structures for both currencies are upward sloping, meaning the model expects to see the volatility of both currencies go up within the next 21 days.

Now we are going to compare the volatility of these 2 cryptocurrencies with the volatility of a standard stock. We already had the intuition that the volatility of the Bitcoin and Etherum was high in absolute term, but we need a benchmark to confirm this intuition. For doing this, we downloaded the price of the S&P500 and fitted a GARCH(1,1) volatility model as we have done for the cryptocurrencies.

Comparison of the volatility of both currencies with S&P500

We fitted a similar GARCH(1,1) model to the SP500 during the same time frame in order to get a sense of whether the volatility of Bitcoin and Ethereum are "high" and plotted a GARCH-based weekly volatility of S&P500. The results can be seen on the plot bellow:

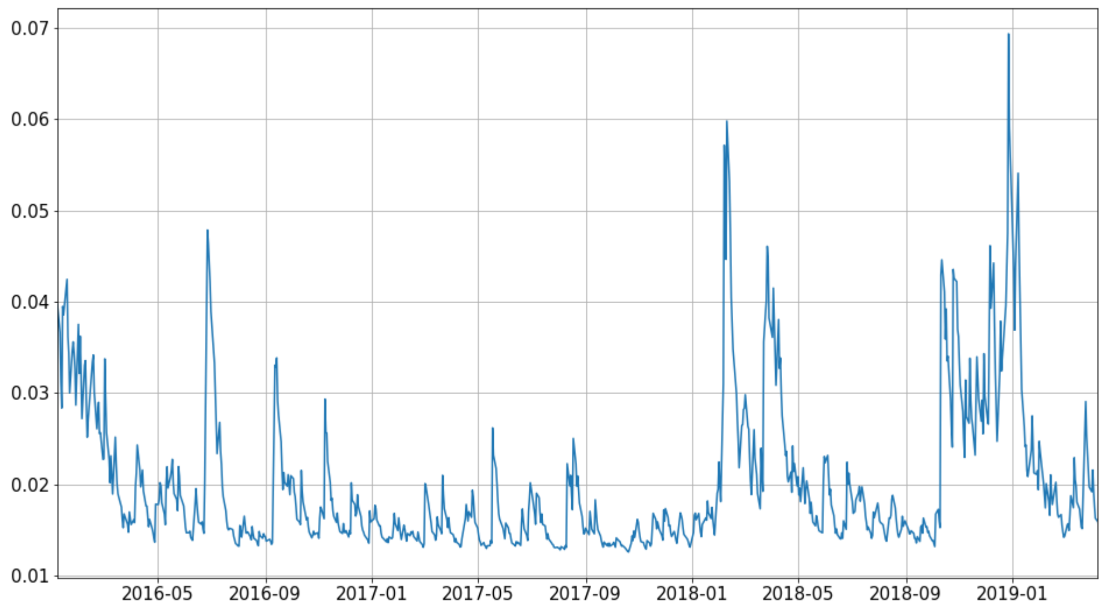

Figure 6: GARCH-based weekly volatility of S&P500 between 01-01-2016 and 04-07-2019

Clearly, we can establish that the volatility of a "standard" stock is way less significant than the volatility of a crypto-currency, as the GARCH-based volatility of S&P500 fluctuates between 1% and 7% while the one of Bitcoin reaches 25% or even 50% for Ethereum (keeping in mind that those two currencies are supposed to form the best portfolio in term of a mean-variance approach).

This observation is not surprising at all as crypto-currencies represent a relatively new market with a lot of uncertainty and speculation. There is a significant debate on whether 'cryptos' should be treated as regular securities or no, which tends to increase volatility. Therefore, any information can have a significant impact on their prices, which explains the relatively high volatility compared to traditional securities.

Finally, let us finish this preliminary step by studying correlations between the two log-returns. To this extent, we will display some measures of the conditional and unconditional correlation.

2. Unconditional and conditional correlation

Firstly, the unconditional correlation correlation has been computed using the already built Numpy function, thus the so-called Pearson correlation coefficient of the two log-returns is 0.4505. From this number, we can note that the two cryptocurrencies are quite strongly positively correlated. This is not surprising are both assets are part of the cryptocurrency industry and their returns are then strongly correlated.

Now, let us see how the conditional correlation evolves. To this extent, we grouped log-returns by week, then firstly we computed the empirical correlation between the first 7 days, then we computed the correlation between the first 14 days, then between the first 21 days,... and so on until we used the all sample. The results can be seen on the plot below:

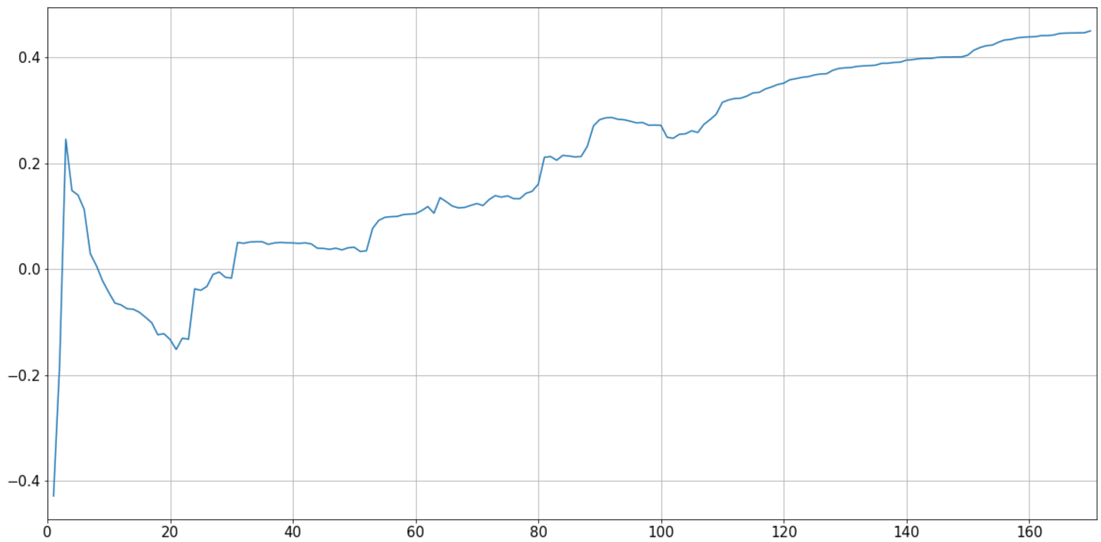

Figure 7: Conditional correlations

From the plot above, one can see that the conditional correlation is moving as the sample is increasing. On can see and as it should be the case, it tends to the Pearson correlation coefficient of approximately 0.45.

Static allocation

We are now going to focus on the static allocation portfolio. By static, we mean that we choose the weights at time $t = 0$ and won't change it at any time after. First, we will start by computing the first two 2-period moments. However, already note that to compute these moments we will use fully forward looking data and we can already said that this strategy won't be implementable in real life, we will come back to this point later in this report.

3. First and second 2-period moments

Let's start by computing the first 2-period moment $\mu[2]$. As we believe that this moment is constant over time and is i.i.d. Using these assumptions, we can notably used the law of large number which told us that in the limit $T \to \infty$:

$$2\frac{\mu[2]_1 + \mu[2]_3 + \mu[2]_T}{T} \to \mu[2]$$

Where $\mu[2]_t$ represents the 2 days return at time t. Note that the factor of 2 comes from the fact that we only computed this 2 days returns every 2 days, and thus the total number of observation if $\frac{T}{2}$. So, to compute this moment, we had at every time $t = 1, 3, 5...T$ compute the 2 days return (which is just the sum of the the two 1 day return, as we are using log-returns), and we just approximate $\mu[2]$ by making the average of this 2 days returns. We obtained the following numerical results:

$$\mu[2] = \begin{pmatrix} 0.0041 \\ 0.0087 \end{pmatrix}$$

Now, let's determine the second 2-period moment $\Sigma[2]$. For determining this moment, we just computed a table containing all the 2 periods returns for all the date $t = 1, 3, 5, ...T$ and we computed the empirical covariance of this matrix using the already built Numpy function. We obtained the following numerical results:

$$\Sigma[2] = \begin{pmatrix} 0.0034 & 0.0023 \\ 0.0023 & 0.0086 \end{pmatrix}$$

Now that we have the first and second 2-period moments, we can compute the optimal weights.

4. Optimal weights

To compute the optimal weights, we will will first solve the problem analytically and then we will implement it in our code. The mean-variance problem is:

$$\max \omega^T \mu[2] + r_f[2] - \frac{\lambda}{2}\omega^T \Sigma[2]\omega$$

Where ω is a dimension 2 vectors containing the weights invested in the risky assets. Let define the Lagrangian L as $L = \omega^T \mu[2] + r_f[2] - \frac{\lambda}{2}\omega^T \Sigma[2]\omega$. The first order condition is:

$$\frac{\partial L}{\partial \omega} = \mu[2] - \lambda \Sigma_t[2]\omega_t = 0$$

From this equation, we can deduced the optimal static weights for $\lambda = 2$ and $\lambda = 10$:

$$\omega = \frac{1}{\lambda}\Sigma[2]^{-1}\mu[2] \text{ and } \omega_f = 1 - \omega_1 - \omega_2$$

The numerical results obtained for $\lambda = 2$ are:

$$\omega_2 = \begin{pmatrix} 0.3093 \\ 0.4207 \\ 0.2700 \end{pmatrix}$$

So respectively 0.3093 invested in the Bitcoin, 0.4207 in the Etherum and 0.2700 in the risk free rate.
For $\lambda = 10$ we obtained:

$$\omega_{10} = \begin{pmatrix} 0.0619 \\ 0.0841 \\ 0.8540 \end{pmatrix}$$

So respectively 0.0619 invested in the Bitcoin, 0.0841 in the Etherum and 0.8540 in the risk free rate.

From these results, we can notably see that in the case of a high risk averse investor ($\lambda = 10$), we invest almost nothing in the crypto currencies and put almost everything in the risk free asset. It is logical as a risk aversion coefficient of 10 represent a highly risk averse investor, and as the crypto currencies are highly volatile assets, this type of investor won't invest in the crypto currencies and would prefer putting his wealth into the risk-free asset.

Dynamic allocation

5. Determination of $\Sigma_t[2]$

We want to determine the variance forecast for the 2-period log-returns $\Sigma_t[2]$. To this extent, we need to determine the 2-period variance forecast $\sigma_{i,t}^2$ for $t = 1, 3, 5...T$ and for $i = 1, 2$ corresponding to the Bitcoin and the Etherum.

First, let's determine the 1-step ahead forecast,$\sigma_{i,t}^2(1) = V[\epsilon_{i,t+1}|I_t]$, with I_t defining the information set at time t, which is all the information available up to time t, t included in this set. We have:

$$\begin{aligned}
\sigma_{i,t}^2(1) &= V[\epsilon_{i,t+1}|I_t] \\
&= V[\sigma_{i,t+1}z_{i,t+1}|I_t] \\
&= E[(\sigma_{i,t+1}z_{i,t+1})^2|I_t] - E[\sigma_{i,t+1}z_{i,t+1}|I_t]^2 \\
&= E[\sigma_{i,t+1}^2|I_t]E[z_{i,t+1}^2|I_t] - E[\sigma_{i,t+1}|I_t]^2 E[z_{i,t+1}|I_t]^2
\end{aligned}$$

Where we have notably used the fact that $z_{i,t+1}$ is independent from $\sigma_{i,t+1}$. Now, using the fact that $E[z_{i,t+1}|I_t] = E[z_{i,t+1}]$ which comes from the independently distribution of the $z_{i,t}$, we have $E[z_{i,t+1}|I_t] = 0$. Similarly we have $E[z_{i,t+1}^2|I_t] = E[z_{i,t+1}^2] = 1$. Thus, the above formula becomes:

$$\begin{aligned}
\sigma_{i,t}^2(1) &= E[\sigma_{i,t+1}^2|I_t] \\
&= E[\omega_i + \alpha_i\epsilon_{i,t}^2 + \beta_i\sigma_{i,t}^2|I_t] \\
&= \omega_i + \alpha_i\epsilon_{i,t}^2 + \beta_i\sigma_{i,t}^2
\end{aligned}$$

Now, we can express everything as a function of the residuals using the formula for the GARCH variance several times:

$$\begin{aligned}
\omega_i + \alpha_i\epsilon_{i,t}^2 + \beta_i\sigma_{i,t}^2 &= \omega_i + \alpha_i\epsilon_{i,t}^2 + \beta_i(\omega_i + \alpha_i\epsilon_{i,t-1}^2 + \beta_i\sigma_{i,t-1}^2) \\
&= \omega_i + \beta_i\omega_i + \alpha_i(\epsilon_{i,t}^2 + \beta_i\epsilon_{i,t-1}^2) + \beta_i^2\sigma_{i,t-1}^2 \\
&= \omega_i(1 + \beta_i + \beta_i^2) + \alpha_i(\epsilon_{i,t}^2 + \beta_i\epsilon_{i,t-1}^2 + \beta_i\epsilon_{i,t-2}^2) + \beta_i^3\sigma_{i,t-2}^2 \\
&= \omega_i \sum_{k=0}^{t} \beta_i^k + \alpha_i \sum_{k=0}^{t} \beta_i^k \epsilon_{i,t-k}^2 + \beta_i^{t+1}\sigma_{i,0}^2
\end{aligned}$$

So the 1-step ahead variance forecast as a function of the parameters and the squared residuals is given by:

$$\sigma_{i,t}^2(1) = \omega_i \sum_{k=0}^{t} \beta_i^k + \alpha_i \sum_{k=0}^{t} \beta_i^k \epsilon_{i,t-k}^2 + \beta_i^{t+1}\sigma_{i,0}^2 \tag{1}$$

Let us now determine the 2-step ahead variance forecast $\sigma^2_{i,t}(2)$. We have:

$$\begin{aligned}\sigma^2_{i,t}(2) &= V[\epsilon_{i,t+2}|I_t] \\ &= V[\sigma_{i,t+2}z_{i,t+2}|I_t] \\ &= E[(\sigma_{i,t+2}z_{i,t+2})^2|I_t] - E[\sigma_{i,t+2}z_{i,t+2}|I_t]^2 \\ &= E[\sigma^2_{i,t+2}|I_t]\end{aligned}$$

Where we have used as before the independence between $\sigma_{i,t+2}$ and $z_{i,t+2}$, and the fact that $E[\sigma_{i,t+2}|I_t] = 0$ and $E[\sigma^2_{i,t+2}|I_t] = 1$. Now, using the formula for the GARCH variance gives us:

$$\begin{aligned}\sigma^2_{i,t}(2) &= E[\sigma^2_{i,t+2}|I_t] \\ &= E[\omega_i + \alpha_i\epsilon^2_{i,t+1} + \beta_i\sigma^2_{i,t+1}|I_t] \\ &= \omega_i + \alpha_i\sigma^2_{i,t}(1) + \beta_i\sigma^2_{i,t}(1) \\ &= \omega_i + (\alpha_i + \beta_i)\sigma^2_{i,t}(1) \\ &= \omega_i + (\alpha_i + \beta_i)(\omega_i\sum_{k=0}^{t}\beta_i^k + \alpha_i\sum_{k=0}^{t}\beta_i^k\epsilon^2_{i,t-k} + \beta_i^{t+1}\sigma^2_{i,0})\end{aligned}$$

Finally, let us determine the 2-period variance forecast $\sigma^2_{i,t}[2] = V[\epsilon_{i,t+1} + \epsilon_{i,t+2}|I_t]$. We have:

$$\begin{aligned}V[\epsilon_{i,t+1} + \epsilon_{i,t+2}|I_t] &= V[\epsilon_{i,t+1}|I_t] + V[\epsilon_{i,t+2}|I_t] + 2Cov[\epsilon_{i,t+1}, \epsilon_{i,t+2}|I_t] \\ &= \sigma^2_{i,t}(1) + \sigma^2_{i,t}(2) + 2Cov[\sigma_{i,t+1}z_{i,t+1}, \sigma_{i,t+2}z_{i,t+2}|I_t]\end{aligned}$$

Let us now demonstrate that $Cov[\epsilon_{i,t+1}, \epsilon_{i,t+2}|I_t] = 0$. To this extent, note that as $\epsilon_{i,t+1}$ is known at time t (it only depends on the information available at time t as we can see from the GARCH variance formula), we can take it out of the covariance operator.

$$Cov[\sigma_{i,t+1}z_{i,t+1}\,,\,\sigma_{i,t+2}z_{i,t+2}|I_t] = \sigma_{i,t+1}\,Cov[z_{i,t+1}\,,\,\sigma_{i,t+2}z_{i,t+2}|I_t]$$

Now, we show that $Cov[z_{i,t+1}\,,\,\sigma_{i,t+2}z_{i,t+2}|I_t] = 0$:

$$\begin{aligned}Cov[z_{i,t+1}\,,\,\sigma_{i,t+2}z_{i,t+2}|I_t] &= E[z_{i,t+1}\sigma_{i,t+2}z_{i,t+2}|I_t] - E[z_{i,t+1}|I_t]E[\sigma_{i,t+2}z_{i,t+2}|I_t] \\ &= E[z_{i,t+1}\sigma_{i,t+2}z_{i,t+2}|I_t]\end{aligned}$$

Where we have used the fact that $E[z_{i,t+1}|I_t] = 0$. Now, using the law of iterated expectations, we have:

$$\begin{aligned}E[z_{i,t+1}\sigma_{i,t+2}z_{i,t+2}|I_t] &= E[E[z_{i,t+1}\sigma_{i,t+2}z_{i,t+2}|I_{t+1}]|I_t] \\ &= E[z_{i,t+1}\sigma_{i,t+2}E[z_{i,t+2}|I_{t+1}]|I_t] \\ &= 0\end{aligned}$$

Where we have used the fact that $\sigma_{i,t+2}$ is known at time $t+1$ so we can take it out of the conditional expectation at time $t+1$. So the covariance between the two residuals is indeed 0. The 2-period variance forecast is thus just the summation of the 1-step ahead and 2-step ahead variance forecast:

$$\begin{aligned}\sigma_{i,t}^2[2] &= \sigma_{i,t}^2(1) + \sigma_{i,t}^2(2)\\ &= \omega_i + (1+\alpha_i+\beta_i)\sigma_{i,t}^2(1)\\ &= \omega_i + (1+\alpha_i+\beta_i)(\omega_i \sum_{k=0}^{t} \beta_i^k + \alpha_i \sum_{k=0}^{t} \beta_i^k \epsilon_{i,t-k}^2 + \beta_i^{t+1}\sigma_{i,0}^2)\end{aligned}$$

Using the formula derived above, we were able to estimate $\sigma_{i,t}^2[2]$ for the two crypto-currencies chosen. The plot of these two series can be seen on figure 8.

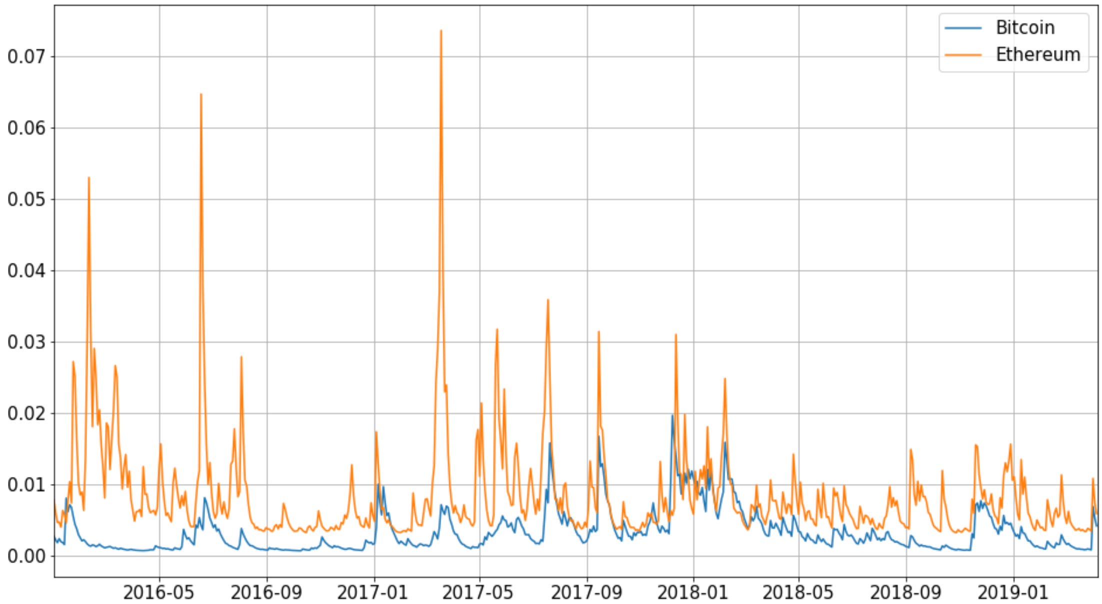

Figure 8: Two days variance forecast $\sigma_{i,t}^2[2]$ of Ethereum and Bitcoin

From Figure 8, we can see that for both Bitcoin and Ethereum, $\sigma_{i,t}^2[2]$ significantly changes over time. While $\sigma_{i,t}^2[2]$ of Ethereum is generally larger, spikes in $\sigma_{i,t}^2[2]$ often coincide. These spikes are however significantly larger for Ethereum than for Bitcoin, especially in the period before September 2017, when $\sigma_{i,t}^2[2]$ of Bitcoin starts to increase over all.

6. Optimal weights

Now that we have the variance-covariance matrix $\Sigma_t[2]$, we can easily compute the dynamics optimal weights. For doing this, we will just use the same formula that we used for the static allocation at each time $t = 1, 3, 5, ...T$,

$$\omega_t = \frac{1}{\lambda}\Sigma_t[2]^{-1}\mu[2]$$

where λ is the risk-aversion coefficient and $\mu[2]$ is the 2-period first moment as used in the static part.

The weights obtain for $\lambda = 2$ and $\lambda = 10$ can be respectively seen on figure 9 and 10. On these two plots, the static weights are represented as dashed lines, the dynamic weights as continuous lines.

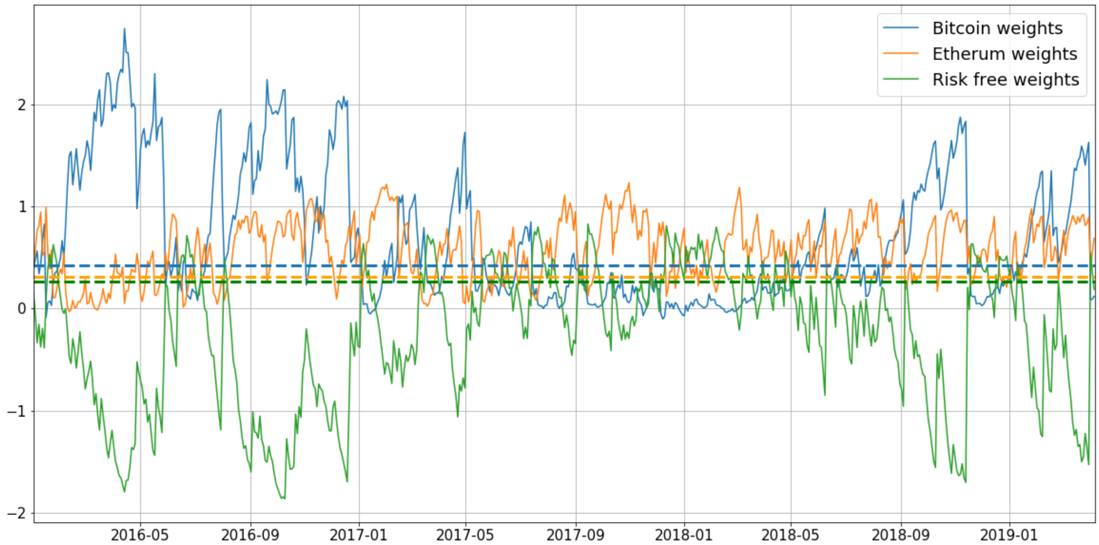

Figure 9: Portfolio weights for $\lambda = 2$

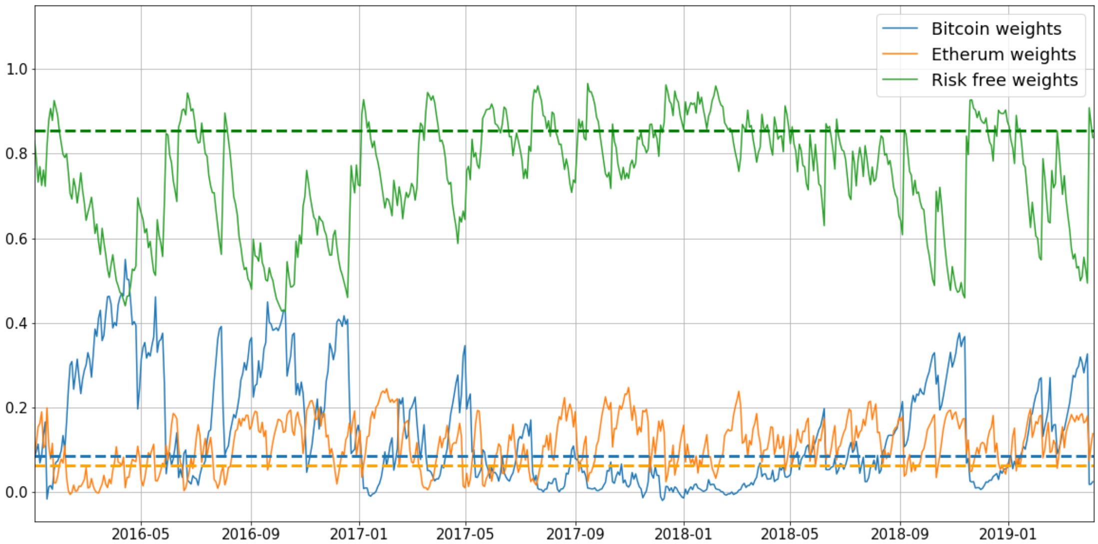

Figure 10: Portfolio weights for $\lambda = 10$

From figure 9 it is visible that in the dynamic strategy, with $\lambda = 2$, the characteristics of the weight allocation change over time. While the weight allocated to Ethereum stays between zero and one, in some periods a substantial amount of leverage is used to invest in Bitcoin. This is especially evident in the periods from February 2016 until September 2017 and June 2018 until April 2019. This means that in comparison to the static allocation, for which all weights are positive, in some periods a significantly larger amount of risk is taken.

Compared to the weight allocation, for $\lambda = 2$, for $\lambda = 10$, the weight allocated to the risk free asset is larger. For the dynamic strategy the risk free is never used to leverage the position in the cryptocurrencies. This is a stark difference to the allocation with $\lambda = 2$ where for most times the risk free is used to leverage the position in Bitcoin and to some extend Ether. The static allocation strategy allocates less than 20% into the cryptocurrencies and holds most of the wealth in the risk free rate.

A comparison of the dynamic allocation of, when $\lambda = 2$ and when $\lambda = 10$, shows that the general dynamic of the allocation to Bitcoin and Ethereum stays the same, just the magnitude of the investment changes. Hence, in times where an investor with risk aversion $\lambda = 2$ should invest more in to Bitcoin, an investor with risk aversion $\lambda = 10$ should invest more in Bitcoin as well and the proportion between the investments in Ethereum and Bitcoin should be identical. What is changing is the overall fraction of wealth invested in the currencies due to the risk aversion.

7. Comparison

The plots of the cumulative returns can be seen on figure 11. As we can see the static allocation performs slightly better than the dynamic allocation. This can be astonishing at the first look but it is logic. Indeed, this is mostly due to the fact that when we did the static allocation, we compute the co-variance matrix $\Sigma[2]$ using all of our data (first this is using all forward looking data and thus wouldn't make sense in real life but we will come back to this point during the conclusion). On the other hand, when we did the dynamic allocation, we computed at each time t the variance only using using data available at time t. So at the end, we have more information for the static allocation than for the dynamic one, which make the static performing better than the dynamic.

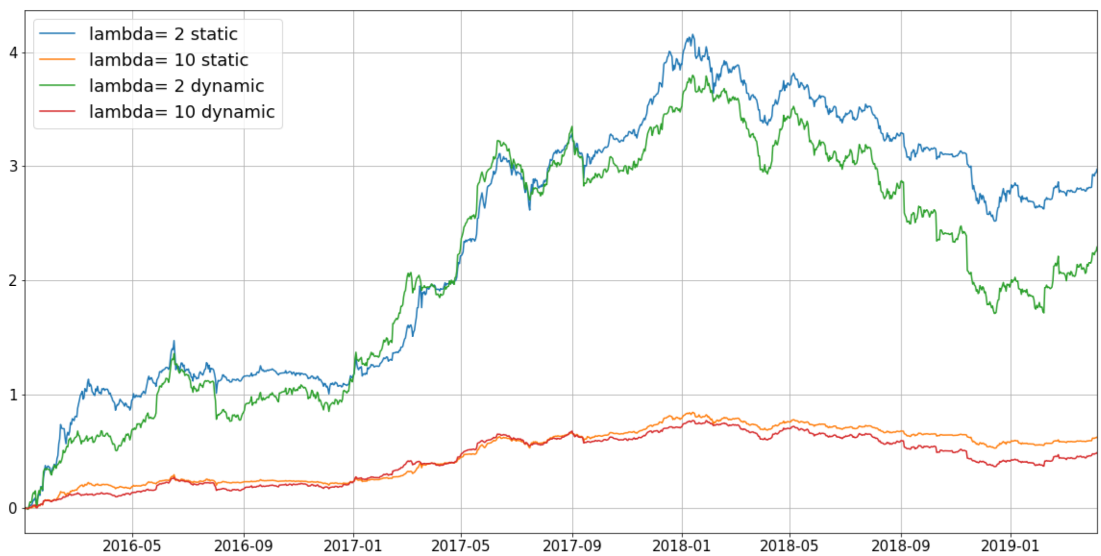

Figure 11: Cumulative returns of the dynamic and static strategy for $\lambda = 2$

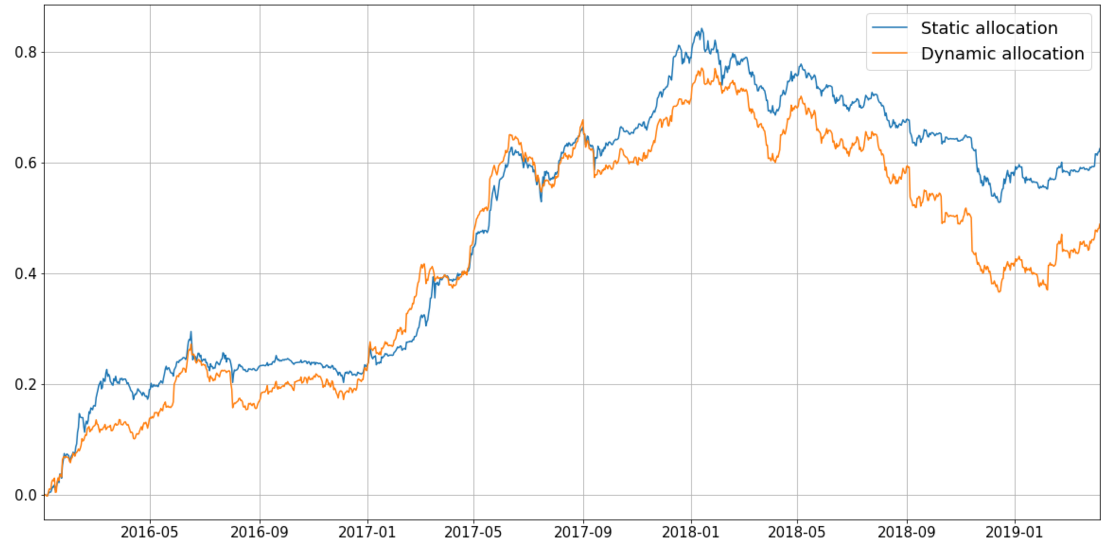

Figure 12: Cumulative returns of the dynamic and static strategy for $\lambda = 10$

We initially thought that the dynamic allocation method would perform better, but one of the most advantage of the dynamic allocation method is that at each time, you have new information and you use them to recompute your weights. But as for the static we already used all the information, this advantage disappear. And even more than that we finally having more information for compute the static weight than for computing dynamic one (except at the last time t where we had all the information for both).

To summarize why the static allocation performs better than the dynamic, let's keep in mind this simplification of the problem. On one hand, with the static allocation, we only computed one time the weights with all the information (fully forward looking) available in our dataset. on the other hand, with the dynamic allocation, we computed each 2 days the weights, with only information up to time t for the covariance matrix (we stilled had the full forward looking information for the vector of excess return μ and the Garch parameter as in the static allocation). Thus we computed some weights with less information than the one computed for the static part.

In reality and if we wanted to make a implementable strategy, this of course won't be true as we won't be able to use any forward looking information at all. We will explore more this point during the conclusion.

Pair trading

8. Pair trading using an error correction model

A pair trading strategy is a market-neutral strategy that matches a long position with a short position in a pair of highly correlated instruments.

In our situation, if we want to consider using a pair trading strategy with our two currencies, the first goal is hence to determine whether they are correlated. Running a simple regression of one currency over another could generate a misleading result over their possible correlation if they are non-stationary. So prior to everything, one first needs to test whether the two currencies are non-stationary. To do so, one might find useful to use a Dickey-Fuller test on both currencies.

Considering an AR(1) model:

$$P_{t,i} = \rho P_{t-1,i} + u_t$$

where $P_{t,i}$ is the price of crypto-currency i at date t.

Let $\Delta P_{t,i} = P_{t,i} - P_{t-1,i}$. We thus have:

$$\Delta P_{t,i} = (\rho - 1)P_{t-1,i} + u_{t,i} = \delta P_{t-1,i} + u_{t,i}$$

And Δ is also called the first-order difference operator.

Now, we run this regression and use the Dickey-Fuller table to interpret the result of the t-statistic (as we are running the regression over the residuals rather than the raw data, we cannot use a simple student table). With this test is we can conclude if the prices are stationary of order 1 or not, depending on whether we reject or not the hypothesis that $\delta = 0$ (which is equivalent to $\rho = 1$).

	t-stat	**p-value**
Bitcoin	-1.921	0.322
Ethereum	-1.988	0.292

Table 1: Results of the ADF-test for the price series

Table 1 shows the results of the ADF test applied to the price time series of Bitcoin and Ethereum. Clearly, the null hypothesis can not be rejected in both cases, hence, it can be concluded that the price processes of the two cryptocurrencies are non-stationary of order 1.We can now test for co-integration, which is fundamental if we want to consider a pairs trading strategy.

Co-integration allows to eliminate the effect of time on two variables that are non-stationary and thus to know if they are really correlated. In order to test the co-integration of the two currencies, let us first consider the following relationship between the two currencies:

$$P_{t,i} - \beta P_{t,j} = \epsilon_t$$

where $i, j = \{bit, eth\}$, $i \neq j$

The prices of the two crypto-currencies would be co-integrated if there exists a β such that the above relationship is stationary. In order to test that, one might run the following regression:

$$P_{t,i} = \hat{\alpha} + \hat{\beta} P_{t,j} + \hat{\epsilon}_t$$
$$\iff \hat{\epsilon}_t = P_{t,i} - \hat{\alpha} - \hat{\beta} P_{t,j}$$

Applying the first-order difference operator gives us:

$$\Delta \hat{\epsilon}_t = \hat{\delta_0} + \hat{\delta_1} \epsilon_{t-1} + ... + v_t$$

With this relation, we can apply the Dickey-Fuller test to know if the residuals are stationary, which would in turn imply that the two currencies are co-integrated. To implement the described method, first a ordinary least square regression of the two price series is conduction, the results of which are displayed in figure 13.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                bitcoin   R-squared:                       0.795
Model:                            OLS   Adj. R-squared:                  0.795
Method:                 Least Squares   F-statistic:                     4623.
Date:                Wed, 08 May 2019   Prob (F-statistic):               0.00
Time:                        21:13:49   Log-Likelihood:                -10578.
No. Observations:                1193   AIC:                         2.116e+04
Df Residuals:                    1191   BIC:                         2.117e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1157.7893     65.277     17.737      0.000    1029.718    1285.860
x1            12.4825      0.184     67.995      0.000      12.122      12.843
==============================================================================
Omnibus:                      463.126   Durbin-Watson:                   0.027
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2827.562
Skew:                           1.676   Prob(JB):                         0.00
Kurtosis:                       9.757   Cond. No.                         467.
==============================================================================
```

Figure 13: Summary of OLS regression Bitcoin price vs. Ethereum price

In a second step the residuals are extracted and the ADF test is performed on them. The test yield a p-value of 0.00044, which is clearly smaller than 1%, hence we can reject the null and accept the alternative hypothesis that the residuals are stationary at a conventional 95% threshold. This means that the two price series are co-integrated.

Having implemented all these steps, we can now introduce an Error Correction Model (ECM) and explain how it can be useful for pair trading. As both cryptos' prices are non-stationary of order 1, we can thus write:

$$\Delta P_{t,i} = \lambda_0 + \lambda_1 \Delta P_{t,j} + \nu_t \tag{2}$$

This equation depicts a short-term relationship between both currencies.

Now, using the fact that they are co-integrated to the same order, we can also write:

$$P_i^e = \alpha + \beta P_j^e \tag{3}$$

This relation characterizes a long-term relationship of equilibrium between the prices of the two currencies, which explicitly derives from their co-integration.

Now, recall the purpose of pair trading strategy: the objective is to seek for a temporary weakness of correlation between the two supposedly correlated stocks, and take advantage of this by going long in the under-performing stock and simultaneously short-selling the one that is over-performing, closing the position as the relationship returns to its statistical norms. Therefore, the main goal is to know *how long* will it take for the relationship to return to its statistical norms (i.e its long-run equilibrium relationship), and this is exactly what an Error Correction Model predicts:

$$A(L)\Delta P_{t,i} = \gamma + B(L)P_{t,j} + \mu(P_{t-1,i} - \beta_0 - \beta_1 P_{t-1,j}) + v_t \tag{4}$$

This equation formulates the ECM and is a combination of (2) and (3): it gives us a short term relationship between the two currencies and at what speed (the coefficient μ) the relationship returns to its long-run equilibrium. Therefore, it is clearly established that by estimating correctly μ, one can make profit by using a pair trading strategy on the two crypto-currencies.

To obtain the ECM the two-step method of Engle and Granger is used. Using the residuals of the OLS-regression summarized in figure 13, the lagged residuals $\hat{\epsilon}_{t-1}$ are calculated and the following regression is performed:

$$\Delta P_{t,i} = c + \lambda \hat{\epsilon}_{t-1} + \sum_{k}^{K_1} a_k \Delta P_{t-k,i} + \sum_{k}^{K_2} b_k \Delta P_{t-k,i} + v_t \tag{5}$$

To find the optimal lag for the prices, K1 and K2 the regression is performed with multiple lags, where $K1 \leq 3$ and $K2 \leq 3$ and the AIC and BIC criterion are used to find the best values. The results are displayed in table 2. As is clearly evident the regression with K1=1 and K2=3 has the lowest AIC and BIC and is therefore the most favorable model.

Model	**AIC**	**BIC**
(1,1)	17055.723465	17070.973628
(1,2)	17039.380107	17059.710302
(2,1)	17039.196778	17059.526972
(2,2)	17643.007762	17663.337956
(3,1)	17030.751756	17051.078590
(3,2)	17642.540721	17662.867555
(3,3)	18027.760987	18048.087822
(2,3)	17762.548128	17782.874963
(1,3)	17028.060140	17048.386975

Table 2: AIC and BIC values for different K1 and K2

Using these K1=1 and K2=3 renders the results displayed in figure 14. As can be seen, the model results in $\lambda = 0.0560 > 0$. This is unusual as we would expect $\lambda < 0$, the positive λ suggests, that the process does not converge in the long run. This can mean, that there are instabilities present, such as structural changes. Indeed, a closer look at the residuals reveals that their structure is not stable over time, as can be seen in figure 15. As a formal test of the structural break the Chow-test is used, the test has the null hypothesis that there is no structural break, and the alternative that there is a structural break. The test statistic is:

$$CHOW = \frac{(RSS_p - (RSS_1 + RSS_2))/k}{(RSS_1 + RSS_2)/(N_1 + N_2 - 2k)} \tag{6}$$

Using the proposed break point of 2017-05-01, we recive a p-value of $7.9 \times 10^{-49} \approx 0$, hence we reject the null hypothesis and conclude that there is indeed a structural break.

To implement a pairs trading strategy we need, that the speed at which the stationary process returns to its long term equilibrium is sufficient. For this to be the case λ has to be smaller than and significantly different from 0. As we have seen, this is not the case for Bitcoin and Ethereum. We can therefore conclude that this pair is not suitable for pairs trading.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                bitcoin   R-squared:                       0.993
Model:                            OLS   Adj. R-squared:                  0.993
Method:                 Least Squares   F-statistic:                 5.930e+04
Date:                Wed, 08 May 2019   Prob (F-statistic):               0.00
Time:                        22:35:48   Log-Likelihood:                -8510.0
No. Observations:                1190   AIC:                         1.703e+04
Df Residuals:                    1186   BIC:                         1.705e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         76.4501     28.475      2.685      0.007      20.582     132.318
x1             0.0560      0.023      2.437      0.015       0.011       0.101
x2             0.9549      0.022     43.586      0.000       0.912       0.998
x3             0.4766      0.273      1.744      0.081      -0.060       1.013
==============================================================================
Omnibus:                      551.470   Durbin-Watson:                   1.826
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            39368.844
Skew:                           1.270   Prob(JB):                         0.00
Kurtosis:                      31.063   Cond. No.                     1.77e+04
==============================================================================
```

Figure 14: Summary of ECM regression

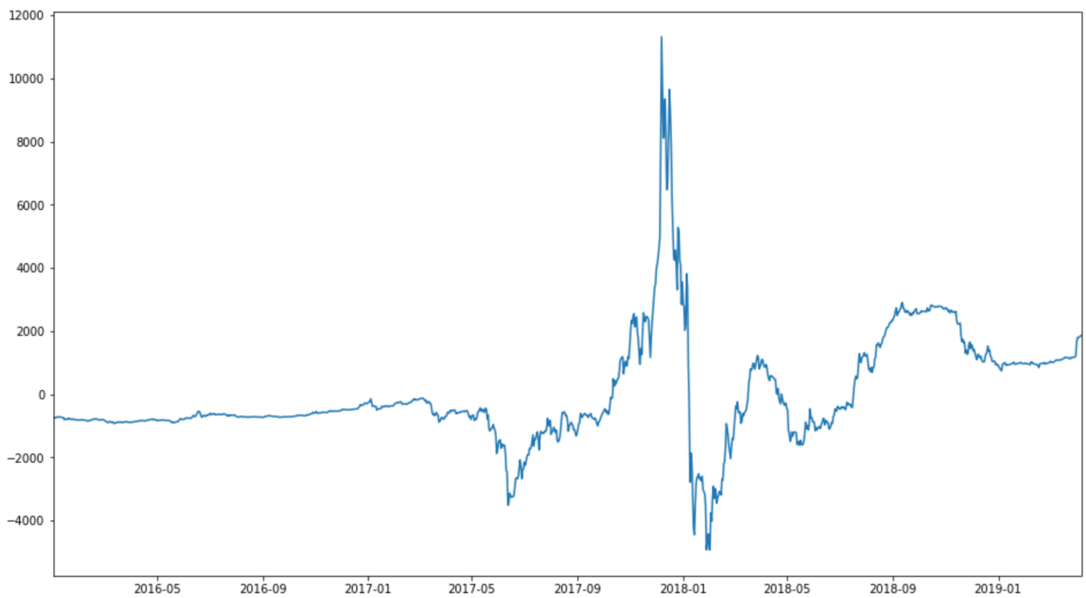

Figure 15: Residuals of OLS regression Bitcoin price vs. Ethereum price over time

To summarize, an ECM can help to implement a pairs trading strategy in several ways. It can be used to find an appropriate pair assets by finding a pair that is co-integrated, as well as a pair for which the return to the long term equilibrium is sufficiently fast. If two times series are co-integrated and a spread opens between the two series, we expect the prices of one or both of the series to change, in order to close the spread. As a_k and b_k give us the impact of the k-lagged prices of the two series, these values of the ECM can be used to create a pairs trading strategy.

Conclusion

9. Conclusion and final thoughts

Brief summary of our findings

GARCH specification and timing volatility

Now we come back to one of our main goal of this project, studying whether there is a so-called "timing-volatility" effect. If there is such an effect, then we should have been able to make some gain by correctly forecasting the volatility dynamics. Implicitly, one thing that we should have seen if this would have been true is notably a better performances of dynamic allocation. Indeed, we only used variance forecast for this strategy and not for the static one. From that we could conclude that GARCH specification was not useful in timing volatility, but this would be not exactly true.

Indeed, this is the case here that the GARCH specification forecast didn't help us beat the static strategy, but in the mean time we computed the static strategy using all the forward looking data. So this a bit like cheating, because on one side with the dynamic allocation we restrict ourselves to only used past data to forecast the volatility but on the other side we used all the data available to compute the static covariance matrix. So saying that the GARCH specification is not useful in timing volatility because this static allocation performs better cannot be generalized to all possible state.

It is also linked to what we just discussed before with forward looking information used to compute the GARCH coefficients, the mean excess return and most of the things used in the project.

Characteristics

Extension

10. Static and Dynamic allocation over two consecutive weeks

To calculate the allocation over two weeks, we first need to calculate the weekly log returns and log excess returns. As we consider log returns, the weekly log returns can be calculated as the sum of seven daily log returns. Having calculated the weekly log returns, we can then apply the same procedure as for the daily log returns. Figures 16 and 17 show the weights that are allocated to the assets for $\lambda = 2$ and $\lambda = 10$ respectively.

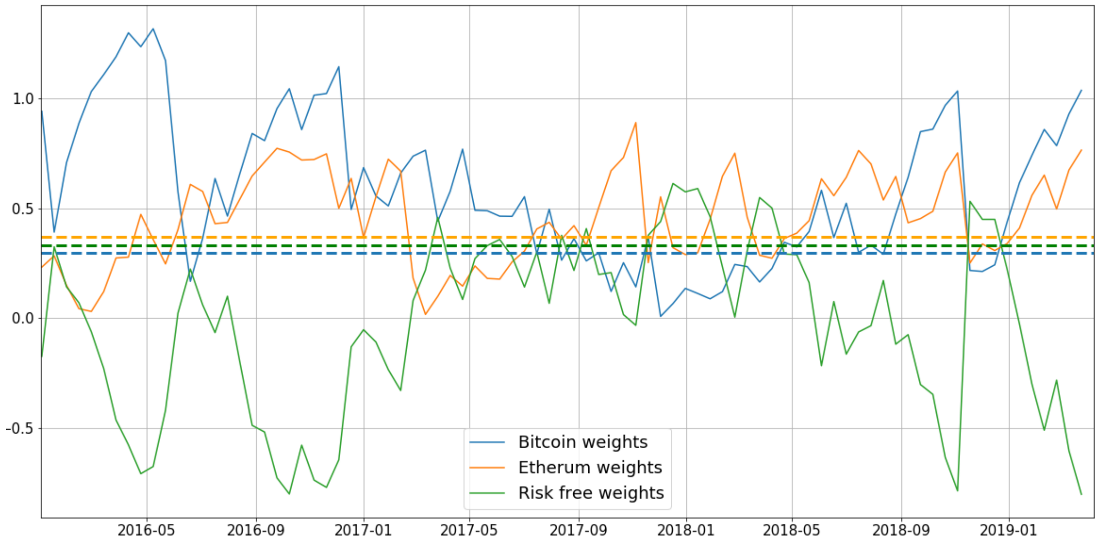

Figure 16: Weight allocation using two consecutive weeks with $\lambda = 2$

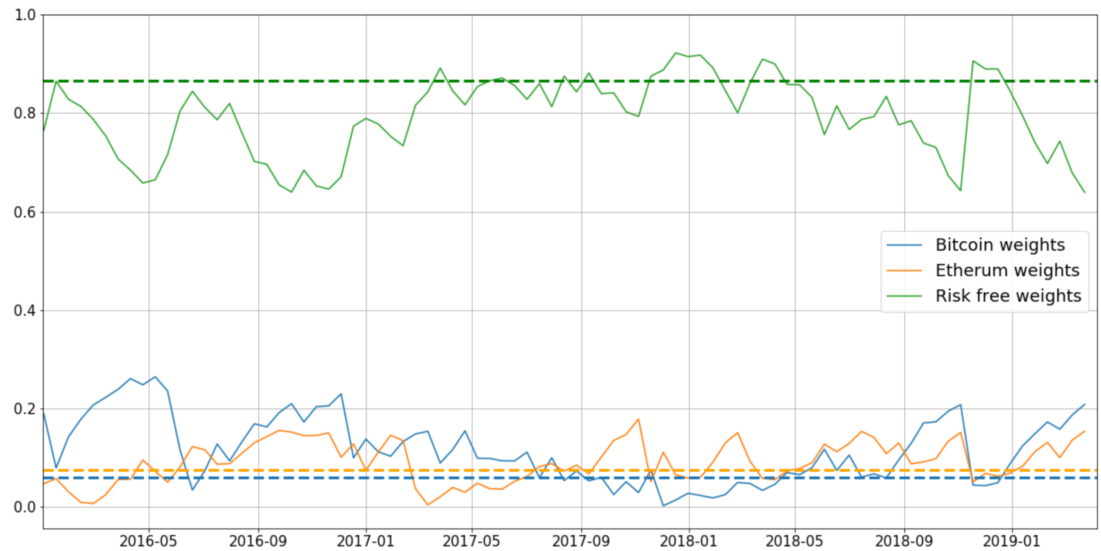

Figure 17: Weight allocation using two consecutive weeks with $\lambda = 10$

Using static allocation over two weeks instead of two days results in a higher weight for Ethereum and a lower weight for Bitcoin, while the allocation to the risk free stays more or less unchanged, this is the case for a risk aversion of $\lambda = 2$ and $\lambda = 10$. In the dynamic case, for $\lambda = 2$, the leverage is smaller in comparison to the two days consecutive allocation, consequently the weight allocation to the cryptocurrencies is smaller. On average the dynamic weight allocations for $\lambda = 10$ are similar for two consecutive weeks and two consecutive days, but the volatility of the weight allocations seems to be lower when using two consecutive weeks.

Figure 18 shows the cumulative returns of the dynamic and static allocation over two weeks. The general structure of the cumulative returns is identical, the dynamic strategy is still performing worse than the static. However, both strategies perform worse when using an allocation over two weeks.

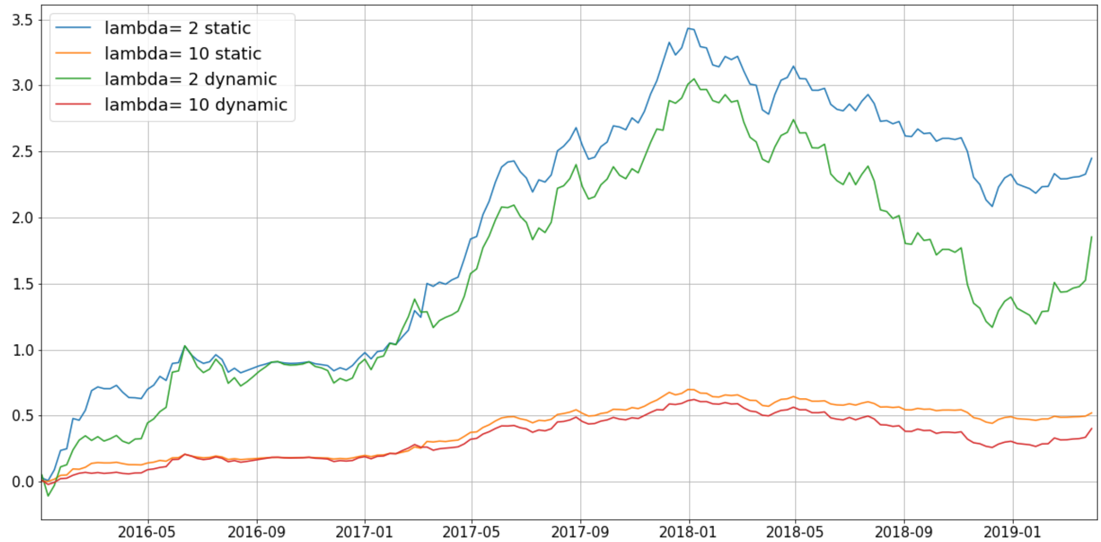

Figure 18: Cumulative returns of the dynamic and static allocation over two weeks

11. Improvement of the dynamic allocation

As we have seen before and quite surprisingly the static allocation performs slightly better than the dynamic allocation. One assumption which is very questionable and which might drag downs our performance for the dynamic strategy is the constant correlation ρ_{12} between the residuals. This assumption is clearly too strong and to confirm this, we computed a 21 days moving average of the correlation coefficient between the 2 log-returns. We plotted the results on the figure 19 bellow:

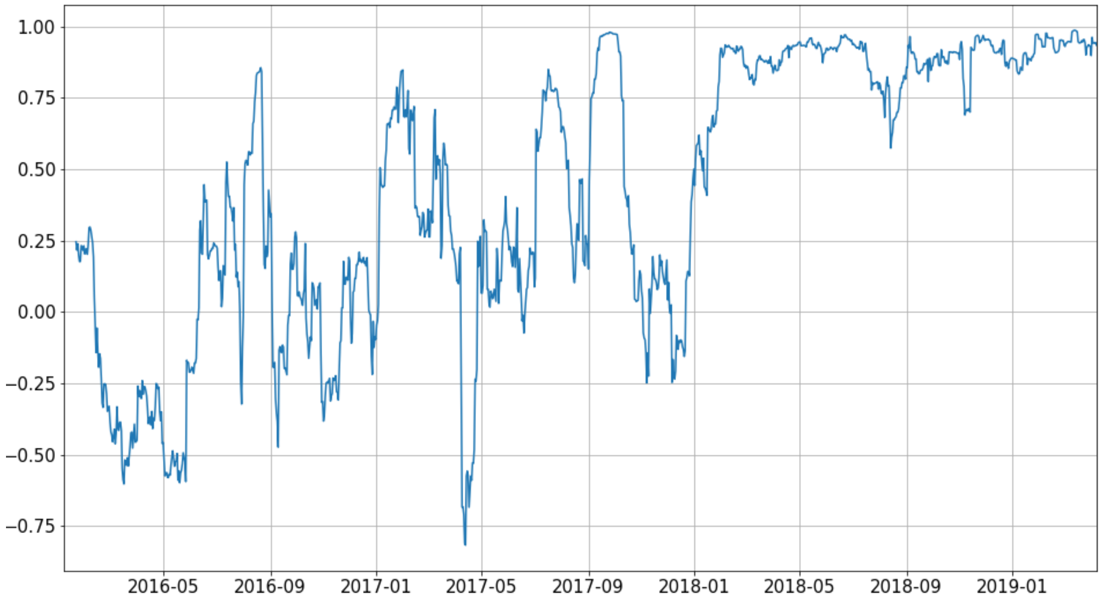

Figure 19: Moving average of the correlation between the two log-returns

The 21-days moving average correlation coefficient display above confirms our thought that this coefficient is clearly not constant over time and that we should take this into account. This leads to a "bad" estimation of the covariance between the Bitcoin and the Etherum. Then the performance of the dynamic allocation is affected by this inaccurate estimation of $\Sigma_t[2]$.

So a better formulation and easy improvement of the dynamic construction might be to recompute at each time the correlation coefficient, and compute the covariance-variance matrix $\Sigma_t[2]$ using this coefficient. Then, using this new matrix to determine the dynamic weights should allow us to obtain better performances.

12. Tether structure volatility

Tether has a different structure of volatility, mostly because Tether is a stable currency (or at least the company Tether claim it to be a stable currency) in the sense that 1 Tether is always valued at 1 USD. It makes the Tether to be a refuge cryptocurrency in a world where most of the cryptocurrencies are very volatile. But lot's of scandal have hit Tether this past few years, notably a case that is underway now about the link between a $850 Million loss of Bitfinex (cryptocurrency exchange platform) covered using some of the Tether Funds. It raises the question if the Tether will always really maintain is exchange rate and if it will have sufficient funds to do this at every time. The lack of transparency of this company make that some people don't believe in this "stable" currency.

References

[1] Investopedia. https://www.investopedia.com/university/guide-pairs-trading/

[2] Pelgrin F. (FIN 407). EPFL class of Financial Econometrics

[3] Zhe Huang, Franck Martin. Optimal pairs trading strategies in a cointegration framework., https://halshs.archives-ouvertes.fr/halshs-01566803/document