Market Risk Project

Nathan Fleury, Julien Gilles

30/12/2024

Introduction

This document outlines the key stages undertaken in the **Market Risk** project. The project encompasses a variety of exercises, including:

- Calculating Value at Risk (VaR) using historical data and kernel methods,
- Estimating VaR for a call option via the Monte Carlo method,
- Analyzing price returns through Extreme Value Theory (EVT),
- Identifying the optimal liquidation framework using the Almgren and Chriss model,
- Determining multi-resolution correlations across various FX rates with Haar wavelets,
- Computing the Hurst Exponent.

Question A: Estimation of the Historical Value at Risk using Kernel Approach

Question A.a: Non-Parametric Historical Estimation of Value at Risk (VaR)

From the time series of daily prices of the stock **Natixis** between January 2015 and December 2016 (provided with TD1), we estimate the **historical value at risk (VaR)** for price returns on a one-day horizon, for a given probability level. This probability must be easily adjustable. The VaR is based on a nonparametric distribution using the **logistic kernel**, where the kernel $K(x)$ is the derivative of the logistic function:

Importing Libraries and Data:

- Import of necessary libraries: `pandas` for data manipulation, `numpy` for numerical computations, and `matplotlib` for data visualization.
- The *Natixis* data was directly embedded in the notebook, eliminating the need for an external CSV or text file. The data set includes daily prices and dates.

Computation of the Kernel Density:

The logistic function's derivative is given by:

$$K(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

Using this kernel function, we estimate kernel density with the following formula:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

where X_i represents the i^{th} daily returns of the Natixis stock between 2015 and 2016.

Estimation of the Bandwidth (h):

Through research, we found several ways to estimate the optimal bandwidth h for our returns. A simple way is to use Scott's rule of thumb, defined as:

$$h = 1.06 \cdot \sigma \cdot n^{-1/5}$$

Alternatively, we can use Silverman's rule of thumb, defined as:

$$h = 0.9 \cdot \min\left(\sigma, \frac{IQR}{1.34}\right) \cdot n^{-1/5}$$

- h: The bandwidth, which controls the degree of smoothing in the kernel density estimation.
- σ: The standard deviation of the returns.
- IQR: The interquartile range, defined as the difference between the 75th percentile and the 25th percentile of the data.
- n: The number of observations in the dataset.

These estimators are well-suited for unimodal return distributions. However, Scott's Rule of thumb relies on the assumption that the returns follow a normal distribution, while Silverman's Rule of Thumb is more robust but still limited to unimodal distributions.
We calculate both estimators and determine which one is more suitable for our problem.

Research sources:

- `https://aakinshin.net/posts/kde-bw/`
- `https://en.wikipedia.org/wiki/Kernel_density_estimation`

Kernel Density Plot

After calculating the estimator for h, we plot the kernel density using different values of h to validate our estimation and determine the most appropriate value for our problem.

To identify the optimal h for estimating the distribution, the kernel density must smoothly approximate the historical distribution without being too erratic. This means that the kernel should not overfit the data, ensuring that it captures the distribution's general trend rather than individual fluctuations.

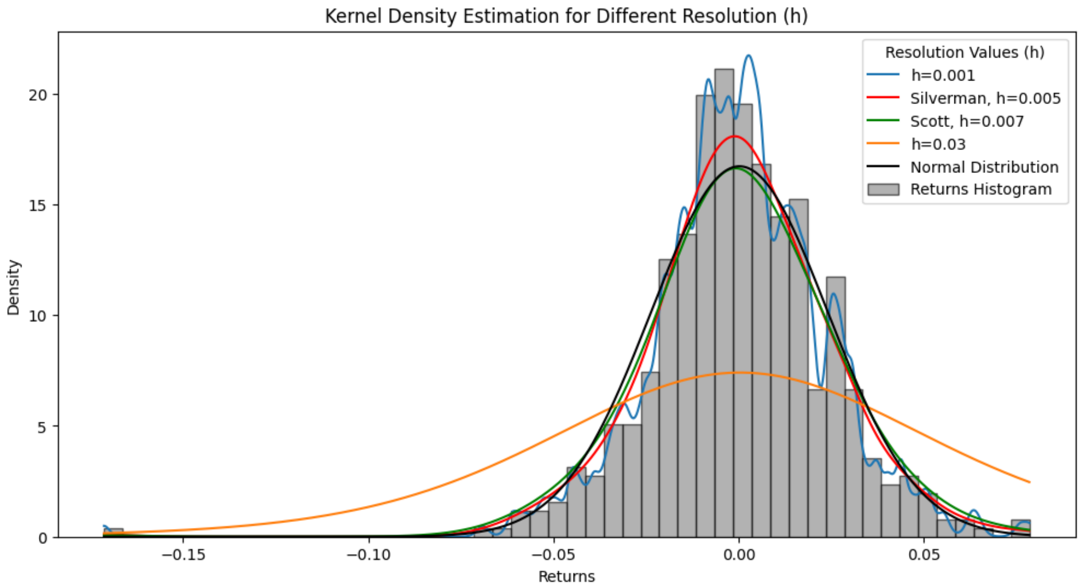

Figure 1: Kernel Density Estimation for Natixis stock returns with varying values of h

We need a function that accurately represents our returns without being too erratic. With $h = 0.001$, the function becomes overly volatile and overfits the return distribution.
Since our return distribution is unimodal, both Silverman's and Scott's Rules of Thumb fit the data well. Moreover, the kernel using Scott's estimator closely approximates a normal distribution.

Compared with the normal distribution, it is evident that kernel density estimation offers a more precise fit for our problem.
After examining the graph, we conclude that $h = 0.005$ using Silverman's estimator provides the best balance for our project, avoiding overfitting and excessive smoothing.

Cumulative Kernel density function:

The Cumulative Kernel Density (CDF), which represents the cumulative sum of the kernel density, is given by the integral of the kernel density function:

$$\hat{F}(x) = \int_{-\infty}^{x} \hat{f}(u)\, du$$

This integral sums the contributions of all data points up to x. In practice, we approximate this integral by computing the cumulative sum of the kernel density values for all data points. In our case, we choose 1000 points and normalize the result by dividing by the total sum to ensure that the cumulative distribution function trends toward 1:

$$\hat{F}(x) = \frac{\sum_{i=1}^{m} \hat{f}(x_i)}{\sum_{i=1}^{n} \hat{f}(x_i)}$$

Where:

- $\hat{f}(x_i)$ is the kernel density estimate for the data point x_i,
- m is the number of data points less than or equal to x,
- n is the total number of data points (in our case, 1000).

This expression normalizes the cumulative sum of the kernel densities to compute the empirical cumulative distribution function, ensuring it trends toward 1 as expected.

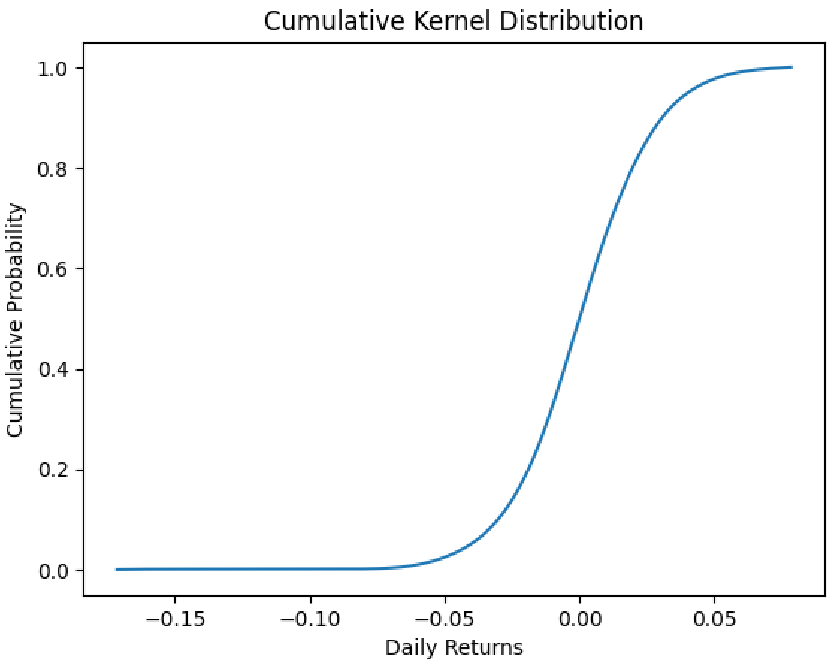

Figure 2: Cumulative kernel density distribution

Computation of the Value at Risk (VaR):

Then, we can compute the value at risk with a risk level of α using the following formula (where X represents the gain and α is the probability of experiencing a return worse than the VaR):

$$\mathrm{VaR}_\alpha = \mathrm{F}_\alpha^{-1}(X)$$

This means that we have a probability of α that the daily return is below or equal to the value at risk.

Once we have the cumulative distribution, the next step is to determine the value corresponding to a given probability level, α. The objective is to find the return threshold below which there is a probability of α (such as 5%) that the actual return will fall.

After implementing the function in Python, we calculated a Value at Risk (VaR) of -4.03% for a 5% risk level, indicating a 95% confidence that daily returns will not exceed this threshold.

Question A.b: VaR Validation

Which proportion of price returns between January 2017 and December 2018 exceed the VaR threshold defined in the previous question? Do you validate the choice of this non-parametric VaR?

To assess the accuracy of our previously calculated non-parametric Value at Risk (VaR), we perform a backtesting exercise using data from the period January 2017 to December 2018. The goal of this analysis is to determine the proportion of price returns that exceed the VaR threshold.

Here is the procedure:

- Import the Natixis returns for the period between January 2017 and December 2018.
- Calculate the number of returns that exceed the VaR threshold of -4.03%.
- Compute the proportion of returns exceeding the VaR.
- Assess and conclude whether the VaR is valid based on the proportion of exceeded returns.

After computing the proportion, we found that 1.57% of the returns exceed the Value at Risk (VaR).
If the VaR is accurate, we would expect the proportion of returns exceeding the VaR to be lower than the chosen confidence level, α (5% in this case), for the period between 2017 and 2018.
In our case, only 1.57% of the returns exceed the VaR threshold, indicating that the computed VaR is both effective and conservative.

However, this result raises questions, as the proportion deviates noticeably from the expected 5%. This discrepancy likely arises from the difference in the distribution of price returns between 2015–2016 and 2017–2018, as illustrated in the next plot.

Although the VaR estimate is **validated**, it can be considered overly conservative for this specific period.

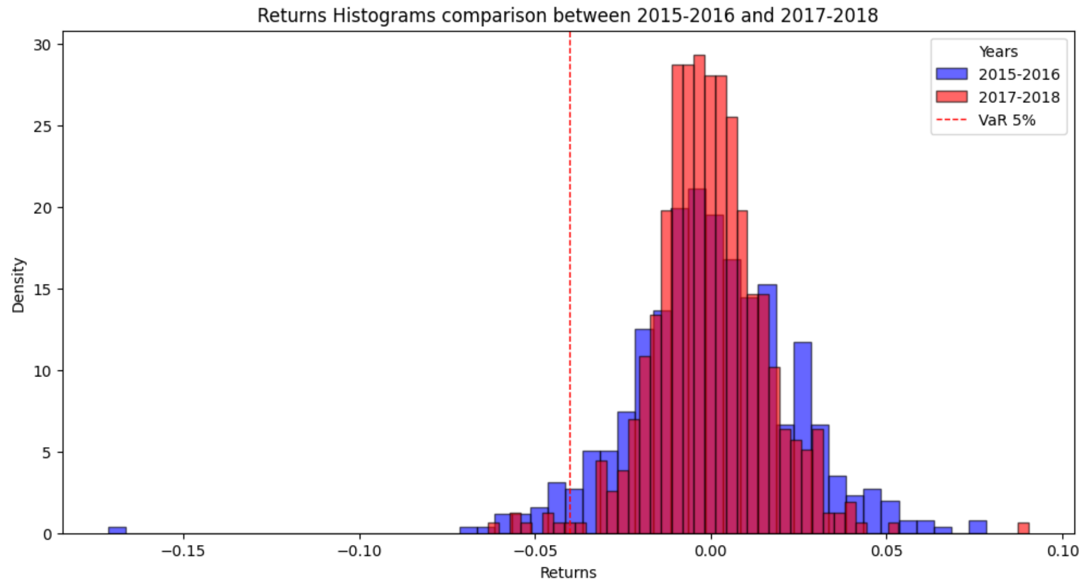

Figure 3: Returns Histograms comparison between 2015-2016 and 2017-2018

The return distribution between the 2015-2016 and 2017-2018 periods is clearly different. This explains why our 5% Value at Risk for 2015-2016 is exceeded in only 1.57% of the returns during the 2017-2018 period.

Conclusion of Question A:

This underscores the well-known adage "Past performance is not indicative of future results," which serves as a reminder that market conditions and risk factors can change over time. The differences in return distributions between the 2015-2016 and 2017-2018 periods demonstrate that relying solely on historical data may not always provide an accurate picture of future risk.

In light of this, it is crucial to use a range of approaches when calculating Value at Risk (VaR), rather than relying exclusively on historical methods. While historical simulations can offer valuable insights, they are based on the assumption that future returns will follow similar patterns to those observed in the past. However, financial markets can experience structural shifts, changes in volatility, or unexpected events that can cause the return distribution to deviate from past trends.

To enhance the accuracy and robustness of VaR estimates, it is important to incorporate other methods, such as Monte Carlo simulations, parametric approaches, which can better account for the uncertainty and dynamics of future market conditions. By combining multiple techniques, we can create a more comprehensive risk assessment that reflects the potential for various scenarios and better captures the true level of risk.

Question B: Calculation of the VaR for a call option on the Natixis stock

The objective in this question is to calculate the VaR (on the arithmetic variation of price, at a one-day horizon) for a call option on the Natixis stock. We are going to implement a Monte-Carlo VaR as the call price is a non-linear function of the underlying price, that we are able to model thanks to historical data.

We started by calculating the returns and the squared ones on the considered period in order to simplify our future calculations.

Question B.a: Parameters Estimation

The first step of this exercise consists in estimating the parameters of a standard Brownian motion on our stock between 2015 and 2018, using an exponential weighting of the data.

The formula that characterizes a brownian geometric motion can be summarized as follows :

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t$$

where :

- S_t is the value of the asset at time t.
- μ is the expected growth rate.
- σ is the volatility.
- W_t is a standard Brownian motion (following a standard normal distribution).

The objective was therefore to estimate both the parameters μ and σ in our context.

Estimation of μ

This estimation was quite easy to compute as it corresponds to the mean of every return on our period.

We obtained the following result :

$$\mu = -9.696604801149598 \times 10^{-5}$$

Estimation of σ

An estimation of the volatility is given by the following formula :

$$\sigma_t^2 = (1 - \lambda)(r_{t-1}^2 + \lambda \cdot r_{t-2}^2 + \lambda^2 \cdot r_{t-3}^2 + \dots)$$

The idea through this process was to give lower weights to the oldest returns in order to focus more on the most recent (to whom we gave a higher weight thanks to the lower power of lambda) . As seen in class, we decided to choose the smoothing factor lambda as 0,94. The implementation of the formula is made in the calcul_variance function.

After annualizing the volatility considering 252 days of trading we obtained the following results:

- Estimated volatility (1 day): 0.02424224156705614
- Annualized volatility: 0.3848336544550998

Question B.b: Prices simulation

The formula to simulate a price that we used is the following :

$$S_t = S_0 \exp\left((\mu - \frac{1}{2}\sigma^2)t + \sigma W_t\right)$$

where :

- S_t is the value of the asset at time t.
- S_0 is the initial price of the asset at the end of 2018.
- μ is the expected growth rate.
- σ is the volatility.
- W_t is a standard Brownian motion (following a standard normal distribution).

The calculation for 1000 simulations and t=1 (as we consider a one-day time horizon) was computed and the results were stored in the simulated_prices tab.
1000 simulations were necessary in order to get more stable and reliable results. With such a number of simulations, it is easier to reduce the variability due to randomness and obtain a better estimate of the future price distributions. It is also interesting to perform such numerous simulations as the mean, variance, and quantiles of the simulated prices converge to their theoretical values as the number of simulations increases.

Question B.c: Transformation in Prices of the corresponding call

We then used the Black and Scholes formula in order to price the call options for our dataset. It is defined as follows :

$$C = S_0 \cdot N(d_1) - K \cdot e^{-rT} \cdot N(d_2)$$

where:

- C is the price of the call option.
- S_0 is the current stock price.
- K is the strike price.
- r is the risk-free interest rate.
- T is the time to maturity.
- $N(d)$ is the cumulative distribution function (CDF) of the standard normal distribution.
- d_1 and d_2 are given by:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}},$$
$$d_2 = d_1 - \sigma\sqrt{T}.$$

where:

- σ is the volatility of the asset.

As given in the statement and as discussed in class, the strike price K was set to 4 while the risk free rate was set to 0 and the time to maturity was 21 (1 month corresponding to 21 days of trading).
We implemented the Black and Scholes formula through the BS_call function and performed the pricing of the call options corresponding to each simulated price using the parameters we previsouly defined.

Question B.d: Empirical quantile of the N call prices to build the VaR of the call.

In order to find the VaR associated to our problem, we first started by calculating the return for each call price compared to the original call price using the following formula :

$$\frac{C_i - C_0}{C_0}$$

We then sorted the returns by ascending order to prepare for the use of the index in the VaR calculation.
We then computed the calculate_VaR function taking as arguments the sorted returns and a confidence level and returning the VaR for different p.
We had finally been able to obtain the VaR for different levels using the quantile index defined as the difference between the number of returns and the multiplication of the number of returns by the confidence level p.

Final results

We obtained the following final results :

- VaR at 99%: 0.6868491834
- VaR at 1%: -0.4535036517

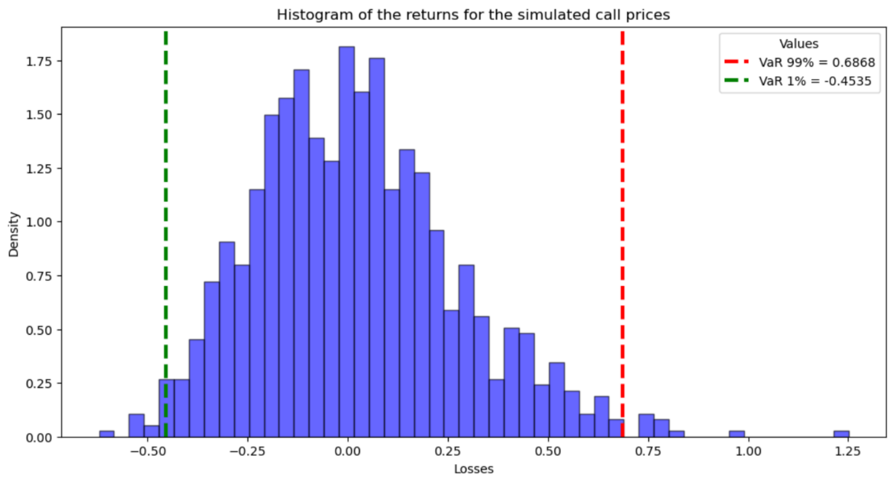

Figure 4: Histogram of the call returns

It suggests that :
- there is 1% probability that the option's return exceeds a gain of 68.69%.
- there is 1% probability that the option's return exceeds a loss of 45,33%.

Question C: Analyze of price returns using Extreme Value Theory

Question C.a: Estimate the GEV parameters for the two tails of the distribution of returns, using Pickands' estimator

In this question, we will apply Pickands' estimator, so let's begin by defining it:

Let (X_n) be a sequence of i.i.d. random variables, whose cdf F belongs to the max-domain of attraction of a GEV distribution with parameter $\xi \in \mathbb{R}$. Let k be a function $k : \mathbb{N} \to \mathbb{N}$. If

$$\lim_{n\to\infty} k(n) = \infty \quad \text{and} \quad \lim_{n\to\infty} \frac{k(n)}{n} = 0,$$

then the Pickands estimator, defined by:

$$\xi_P^{k(n),n} = \frac{1}{\log(2)} \log\left(\frac{X_{n-k(n)+1:n} - X_{n-2k(n)+1:n}}{X_{n-2k(n)+1:n} - X_{n-4k(n)+1:n}}\right),$$

converges in probability to ξ.
Moreover, if

$$\lim_{n\to\infty} \frac{k(n)}{\log(\log(n))} = \infty,$$

then the convergence of $\xi_P^{k(n),n}$ towards ξ is almost sure and not only in probability.

In this exercise, $X_{i:n}$ is the i^{th} value of daily price returns when arranged in ascending order.

Split Returns between gains and losses:

We aim to estimate the GEV parameters for both tails of the return distribution. To achieve this, we first separate the positive and negative daily returns.

For the negative returns, we compute the absolute values, allowing us to apply the Pickands estimator in the same manner as for positive returns.
Additionally, both datasets are sorted in ascending order to apply the Pickands estimator formula correctly.

Determining the function $k(n)$:

To compute Pickands estimator, we first need to find the function k present in the previous definition
For our project, we will use the following function:
$f(x) = \lfloor ln(x) \rfloor$
Indeed, it follows all the condition needed :
$\lim_{n\to\infty} \lfloor ln(n) \rfloor = \infty$ and $\lim_{n\to\infty} \frac{\lfloor ln(n) \rfloor}{n} = 0$ and f is a function $\mathbb{N} \to \mathbb{N}$.

Computation and Results:

After applying the formula above, we compute the pickands estimator for gains and losses :

- For the positive returns, we calculated a Pickands estimator of $\xi = 0.577$. Since $\xi > 0$, the cumulative distribution function of the extreme gains belongs to the max-domain of attraction of the Fréchet distribution, which encompasses thick-tailed distributions.
- For the negative returns, we calculated a Pickands estimator of $\xi = -0.509$. Since $\xi < 0$, the cumulative distribution function of the extreme losses belongs to the max-domain of attraction of the Weibull distribution, which encompasses distributions with finite support.

Question C.b: Determine the extremal index using the block or run declustering

For our project, we decided to determine the extremal index using the run de-clustering

Run Declustering:

Run declustering is a method used to estimate the **extremal index** θ, which measures the dependence structure of extreme values in a time series. Specifically, θ quantifies the rate at which extreme events (such as large values or outliers) occur independently within the series. If the series is independent, $\theta = 1$, meaning that each extreme event is isolated, and there is no dependence between them. As the dependence between extreme values increases, θ decreases.

In run declustering, the time series is divided into "runs" based on a sliding window of arbitrary size r. The window slides across the series, and for each window, the maximum value of the data within that window is computed. The aim is to identify clusters of extreme values that are close to each other in time, using a threshold u.

The extremal index for θ, denoted by $\hat{\theta}_R^n(u;r)$, is computed as follows:

$$\hat{\theta}_R^n(u;r) = \frac{\sum_{i=1}^{n-r} 1(X_i > u, M_{i,i+r} \leq u)}{\sum_{i=1}^{n-r} 1(X_i > u)}$$

Where:
- X_i is the observed price return at time i,
- $M_{i,i+r} = \max(X_i, X_{i+1}, \ldots, X_{i+r})$ is the maximum value in the sliding window of size r
- u is the threshold for the extreme value
- R is the window size

This method contrasts with block declustering, which uses fixed-sized blocks of data. Instead, run declustering uses a sliding window to account for the dependence between consecutive extremes, allowing for a more dynamic estimation of the extremal index.

Data Preparation:

Since we are examining both tails of the distribution, we will compute two extremal indices: one for extreme gains and one for extreme losses.

The first step is to compute the returns without sorting the data, ensuring that the series remains in chronological order.

For the extreme loss, we can't transform with the absolute value so we adapt the formula for losses:

$$\hat{\theta}_R^n(u;r) = \frac{\sum_{i=1}^{n-r} 1(X_i < u, M_{i,i+r} \geq u)}{\sum_{i=1}^{n-r} 1(X_i < u)}$$

Choice of the window size:

In our analysis, we need to define the parameter r, which represents the window size. To ensure meaningful results, we will set the window size to one trading month, or 21 days.

Computation and Results:

Then, we can implement these functions in python and we obtained:

- Extremal index for extreme gains: 0.583
- Extremal index for extreme losses: 0.462

Question D: Almgren and Chriss Model: Framework for Asset Liquidation

The objective of this question is to analyze stock transaction data using the Almgren and Chriss model to determine the optimal strategy for liquidating a portfolio.
Achieving an effective liquidation framework requires balancing the trade-off between the expected value of the liquidation and its variance. Selling all shares at once may result in a lower price but offers certainty about the proceeds. On the other hand, spreading the sale over multiple time steps exposes us to market risk, introducing uncertainty about the final value, but potentially yielding a higher expected value. This trade-off is influenced by the level of risk aversion to market fluctuations—the lower the risk aversion, the faster the portfolio will be liquidated.

Question D.a : Estimation of all the parameters of the model of Almgren and Chriss. Is this model well specified?

Data Import and Cleaning

To begin, we need to import the data for this question. To simplify its use, we directly load the file into our notebook and perform data cleaning in Python.
After importing the Excel files, we compute the returns between consecutive time steps and calculate the cost, defined as the price difference between two time steps.

Estimation of the volatility, σ

To estimate the volatility, we calculate the historical standard deviation of the returns and annualize it by multiplying by $\sqrt{252 \times 8}$, as the returns are expressed on an hourly basis (8 trading hours per day).
After performing the computation, we find:

$$\sigma_{annualized} = 0.0332$$

Estimation of the permanent impact, γ

After calculating the asset's volatility, we remove transactions with unknown volumes, as they cannot be used in the calculation of the other parameters.

Next, we apply the model formula from the Almgren and Chriss model:

$$S_k = S_{k-1} + \sigma\sqrt{\tau}\epsilon_k - \tau g\left(\frac{n_k}{\tau}\right) \tag{1}$$

- S_k represents the current value of the asset.
- S_{k-1} is the previous value of the asset.
- σ is the asset's volatility.
- τ is the time parameter, which represents the time step of the transaction.
- ϵ_k is a random noise term, reflecting market uncertainty.
- $g\left(\frac{n_k}{\tau}\right)$ is a function of $\frac{n_k}{\tau}$, where n_k is the number of assets to be sold at time k.

The function g is defined as:

$$g(v) = \gamma v \tag{2}$$

Substituting Equation (2) into Equation (1), we get:

$$S_k - S_{k-1} = \sigma\sqrt{\tau}\epsilon_k - \tau\gamma\frac{n_k}{\tau} \tag{3}$$

Simplifying the expression, we obtain:

$$S_k - S_{k-1} = \sigma\sqrt{\tau}\epsilon_k - \gamma n_k \tag{4}$$

In our files, transactions occur in both directions (buy and sell), so we transform n_k by the signed volume. Therefore, the final equation is:

$$\Delta S_k = \gamma n_k + \sigma\sqrt{\tau}\epsilon_k \tag{5}$$

Here, we can estimate the price difference using a linear regression, where $\epsilon = \sigma\sqrt{\tau}\epsilon_k$ represents the residuals of the regression.

Linear Regression to estimate γ

The linear regression is defined by:

$$\Delta S_k = \alpha + \beta n_k + \epsilon_k \tag{6}$$

In a linear regression model, we aim to estimate the parameter vector β that minimizes the residual sum of squares, which is the difference between the observed values y and the predicted values $\hat{y} = X\beta$.

Given:
- X is the design matrix, where each row corresponds to a feature vector for a specific observation.
- y is the vector of observed outcomes.

The formula for estimating the vector β using least squares is derived from the normal equation:

$$\beta = (X^T X)^{-1} X^T y$$

The solution β gives the estimated coefficients of the linear regression model, which can then be used for predictions on new data.

In this case, $X = \begin{pmatrix} 1 & n_1 \\ 1 & n_2 \\ 1 & n_3 \\ \vdots & \vdots \\ 1 & n_n \end{pmatrix}$ and $y = \begin{pmatrix} \Delta S_1 \\ \Delta S_2 \\ \Delta S_3 \\ \vdots \\ \Delta S_n \end{pmatrix}$

The first column is used to predict the intercept, which, according to the Almgren and Chriss model, should be zero ($\alpha = 0$).

After performing the linear regression, we obtain the following estimates for the coefficients:

$$\hat{\beta} = \begin{pmatrix} -0.00075 \\ 0.000502 \end{pmatrix}$$

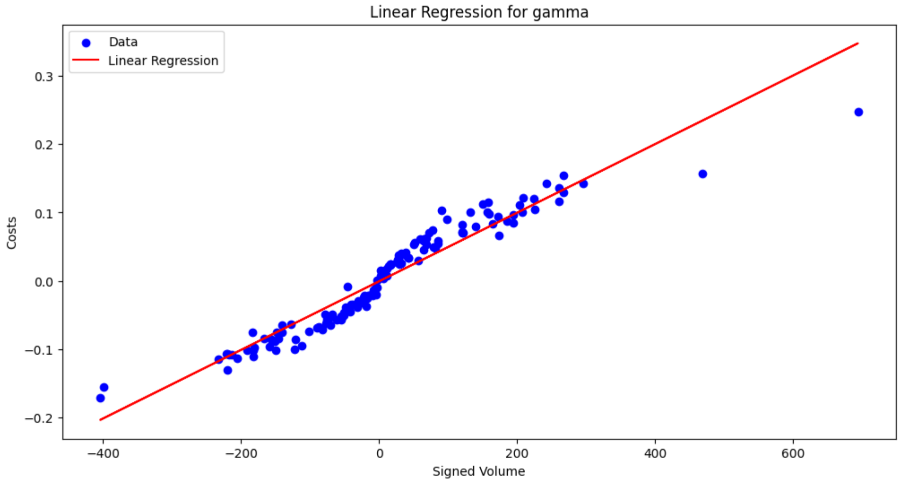

Figure 5: Cost and prediction of the linear regression for γ

This indicates that the intercept $\alpha = -0.00075$, which is close to zero as expected in the model, and the slope $\gamma = 0.000502$.

Validation of the parameter γ

To validate the parameter γ, we need to prove that its value is statistically significant. To achieve this, we employ three different mathematical tools:

- A t-test (Student's test with $H_0 : \beta = 0$),
- Mean Square Error (MSE),
- The R^2 score.

After computing all the statistical metrics, we obtained a t-value of 39.19, an MSE of 0.000491, and an R^2 of 0.919. The t-value of 39 provides strong evidence that γ is significant ($H_0 : \beta = 0$), indicating that γ effectively explains the cost of the transaction.
Additionally, the computed MSE of 0.00049 and R^2 score of 0.91 further reinforce that γ is a significant parameter.
In conclusion, γ is statistically significant for our model, and the parameter is well-defined.

Estimation of the temporary impact, η

To estimate the temporary impact, η, we can use this given formula:

$$\bar{S_k} = S_k - h\left(\frac{n_k}{\tau}\right) \tag{7}$$

and we know that:

$$h\left(\frac{n_k}{\tau}\right) = \xi \operatorname{sgn}(n_k) + \eta \frac{n_k}{\tau} \tag{8}$$

so we have :

$$\bar{S_k} = S_k - \xi \operatorname{sgn}(n_k) - \eta \frac{n_k}{\tau} \tag{9}$$

$$\Delta P = \xi \operatorname{sgn}(n_k) - \eta \frac{n_k}{\tau} \tag{10}$$

by multiplying by the volume:

$$n_k \Delta P = \xi \, |n_k| - \eta \, \frac{n_k^2}{\tau} \tag{11}$$

Then we can compute a linear regression as before and we get:

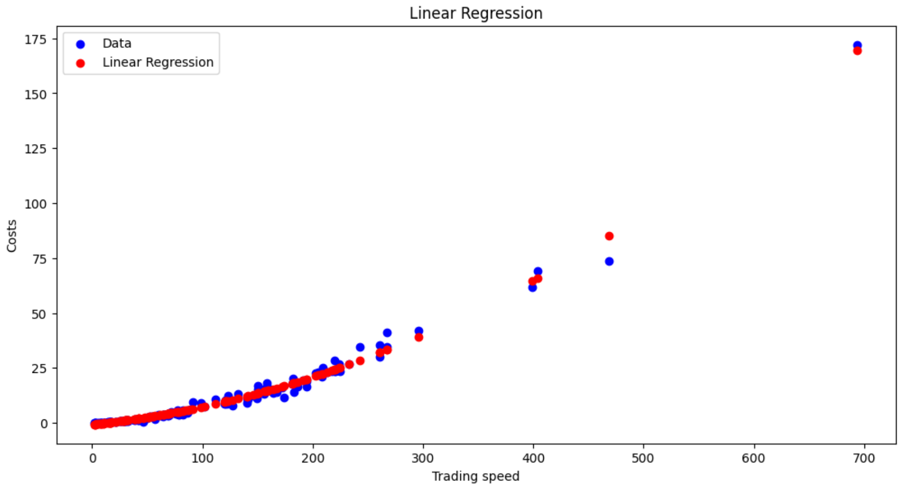

Figure 6: Linear Regression for η

To validate this, we compute the same test as before and we get :

- T-value = 0.2607
- Mean Square Error (MSE) = 0.00584
- The R^2 = 0.0397

We observe that, despite an MSE of 0.005, the R2 value is low and the t-value is also small, indicating that η is not statistically significant. Therefore, our model is not well-defined.

Question D.b : Liquidation strategy in the framework of Almgren and Chriss (we can only make transactions once every hour)

The goal of the liquidation framework is to minimize the loss, which is defined by the following utility function:

$$E(X) + \lambda V(X)$$

where X represents the total loss from the transactions, $E(X)$ is the expected value of the loss, and $V(X)$ is the variance of the loss. The parameter λ is the risk aversion factor. The lower the value of λ, the less the trader is concerned with risk (i.e., the variance $V(X)$), and the more weight is placed on minimizing the expected value of the loss $E(X)$. Conversely, as λ increases, the trader becomes more risk-averse, placing greater emphasis on minimizing the variance of the liquidation process.

Then, there exist an efficient frontier for several levels of risk aversion corresponding to the lagrangian λ solved by Euler-Lagrange:

$$x_k = \frac{\sinh\left(K\left(T - \left(k - \frac{1}{2}\tau\right)\right)\right)}{\sinh(KT)} \cdot X$$

Where:
- x_k : quantity to sell at time k
- T : total execution time,
- τ : time step,
- X : total quantity to sell.

Thus, after performing the computation for different risk aversion with the previously estimated parameters, we obtain the following:

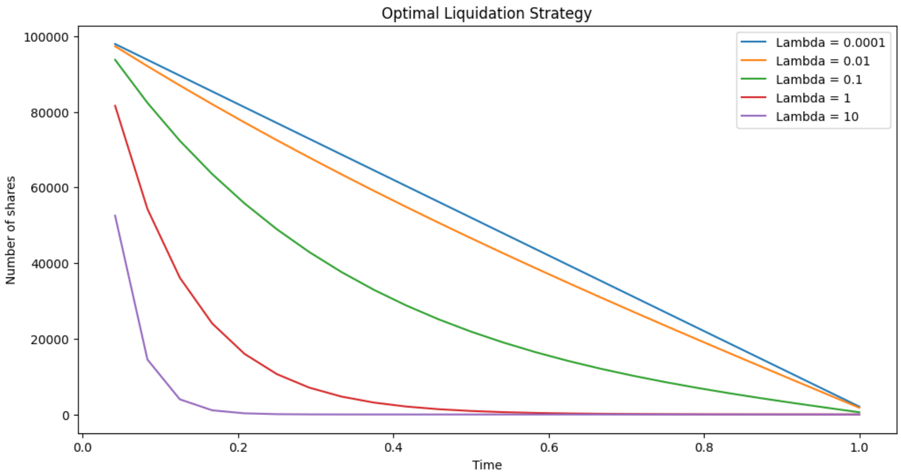

Figure 7: Optimal Liquidation Strategy for Varying Levels of Risk Aversion with a quantity of 100000 shares

Even if the model is not well-defined, the resulting liquidation strategies demonstrate how the choice of λ influences the liquidation speed. A higher value of λ leads to a more aggressive liquidation early in the process, as the trader is willing to take on greater liquidity risk to reduce market risk exposure. On the other hand, a lower λ results in a more gradual liquidation, prioritizing the minimization of liquidity risk, even at the cost of increased market risk exposure.

Question E: Haar wavelets and Hurst exponent

The idea in this part was to focus on the implementation of the multiresolution covariance by using Haar wavelets and to approximate the Hurst exponent in order to determine the annualized volatility.

The first step was therefore to import the data given for each FX rates: GBPEUR, SEKEUR and CADEUR and to compute the mean price for each date using both the low and high prices.

Question E.a : Haar Wavelets and multiresolution covariance

The objective in this question is to calcuate the multiresolution covariance by using Haar wavelets and to determine wether or not we can observe Epps effect.

The first step was to calculate the returns for each FX rate.
Then we started defining the formulas from the course that we will by using later in this part.

Scaling function of the Haar wavelet :

$$\phi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

This "door" function is the basis in order to construct Haar wavelets.

Scaling function for the Haar wavelets for different scales j and translations k :

$$\phi_{j,k}(t) = \phi(2^j t - k)$$

This function is obtained by scaling the previous function $\phi(t)$ by 2^j and by translating it by k. It made it possible for us to work and perform our analysis with different resolutions thanks to it.

Mother wavelet for Haar wavelets :

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

By the same logic, this function represents the basic mother wavelet which defines the Haar wavelet.

Mother wavelet for Haar wavelets for different scales j and translations k :

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \cdot \psi(2^j t - k)$$

Here is the previous function after being scaled by 2^j and translated by k.

Scaling coefficients :

The **scaling coefficients** for a data series z at scale j and translation k are given by:

$$S_{j,k}(z) = \sum_{i=0}^{N-1} z[i] \cdot \phi_{j,k}(i)$$

That is what we performed using the scaling_coeff function.

Multiresolution correlation :

We are now able to perform the multiresolution correlation.
The formula (given in the course) for the covariance is the following :

$$\mathrm{Cov}(j) = \frac{1}{T}\sum_{k=1}^{T}\left(c_{1,j,k} - \frac{1}{T}\sum_{l=1}^{T} c_{1,j,l}\right)\left(c_{2,j,k} - \frac{1}{T}\sum_{l=1}^{T} c_{2,j,l}\right)$$

But we are actually interested in the calculation of the multiresolution correlation which necessitated a small change in our final formula.
The idea behind the `wavelet_corr_multiresolution` function is to first calculate the multiresolution covariance for our two series of returns, and then, using the `scaling_std` function (to obtain the standard deviation of each series at the specific scale j), determine the correlation at different scales (thanks to the loop).

We obtained the following results for each pair :

- GBPEUR - CADEUR
  - scale 0 : 0.2451350036283479
  - scale 1 : 0.47562970366909646
  - scale 2 : 0.5218026817184
  - scale 3 : 0.5772770992369888
- GBPEUR - SEKEUR
  - scale 0 : 0.19143244817378186
  - scale 1 : 0.17159440765410128
  - scale 2 : 0.21931350853400217
  - scale 3 : 0.2507290008933311
- SEKEUR - CADEUR
  - scale 0 : 0.19276505385377804
  - scale 1 : 0.27621944473102916
  - scale 2 : 0.32456740480498186
  - scale 3 : 0.3715728584384291

Epps effect :

As this concept wasn't mentioned in the course, we needed to perform extensive research in order to understand it.
The Epps effect refers to the observed decrease in correlation between two financial time series as the sampling frequency increases. It arises from the discrete nature of high-frequency financial data, where prices are observed at fixed intervals rather than continuously. This leads to a reduction in the precision of correlation at higher frequencies.

In our context, to detect the Epps effect, we decided to calculate the multi-resolution correlation between series at different scales (j=0 until j=3) and observed if the correlation decreased at finer time scales. If this would have occurred, it would suggest the presence of the Epps effect.
According to the multiresolution correlation results previously seen, we observe that for each FX rate pair, when the scale increases (from 0 to 3), the correlation tend to globally increases too. It therefore appears that there is no significant Epps effect to observe here, suggesting that the correlation is reinforcing in the long term. To observe Epps effect, we would have probably needed higher frequency data.

Question E.b : Hurst exponent and annualized volatility

The objective in this question is to calculate the Hurst exponent of GBPEUR, SEKEUR, and CADEUR and to determine their annualized volatility using the daily volatility and Hurst exponents we found.

Hurst exponent

First of all, we know that we can approximate the Hurst exponent using the following formula :

$$\hat{H} = \frac{1}{2} \log_2 \left(\frac{M_2'}{M_2} \right)$$

where :

$$M_2' = \frac{2}{NT} \sum_{i=1}^{NT/2} \left| X\left(\frac{2i}{N}\right) - X\left(\frac{2(i-1)}{N}\right) \right|^2$$

and :

$$M_2 = \frac{1}{NT} \sum_{i=1}^{NT} \left| X\left(\frac{i}{N}\right) - X\left(\frac{i-1}{N}\right) \right|^2$$

As seen in class, we computed the function momentum_hurst using T = 1, simplifying our work here and translated the previous formulas in a sole function.

Thanks to it, we have been able to obtain the following results for each FX rate :

- Hurst exponent for GBPEUR: 0.6714143303551515
- Hurst exponent for SEKEUR: 0.6545913434210177
- Hurst exponent for CADEUR: 0.6552439913405976

Annualized volatility

After being able to compute the Hurst exponent for every FX rate, we could finally take a look at how to compute the annualized volatility thanks to them.

We started by performing the returns for each FX rate using the mean prices already computed.

We then coded a basic function (`calculate_vol`) to get the volatility of the returns of each FX rate on 15 minutes using the following formula :

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2}$$

We obtained the following results :

- Volatility 15 minutes (GBPEUR) : 0.0006232488782175001
- Volatility 15 minutes (SEKEUR) : 0.00032711834135326303
- Volatility 15 minutes (CADEUR) : 0.0005062593239049995

Finally, we implemented the annualized volatility using the following formula :

$$\text{volatility\_annualized} = \text{volatility\_15\_minutes} \cdot (252 \cdot 24 \cdot 4)^H$$

where :

- H : Hurst exponent (that we already computed)
- 252 : number of trading days in a year
- 24 : hours in a day
- 4 : periods of 15 minutes in one hour

It allowed us to obtain the following final results for the annualized volatility of each FX rates :

- Volatility annualized GBPEUR: 0.5469171651061766
- Volatility annualized SEKEUR: 0.2422247358722975
- Volatility annualized CADEUR: 0.3773528414975794