Econometrics

Project assignment - High-Frequency Financial Data

Lucas RODRIGUEZ
lucas.rodriguez@ensiie.fr

Sunday, January 15nd 2023

Intraday volatility estimation from high frequency data

Master in Quantitative Finance (M2QF)

Table of contents

We have used Python as numerical tool for all the required computations and to perform model simulations. In order to follow our practical approach, please refer to the attached Jupyter Notebook file.

1 Introduction

In all what follows, $(\Omega, \mathcal{F}, \mathbb{F} := (\mathcal{F}_t)_{t\in\mathbb{R}^+}, \mathbb{P})$ is a filtered space fulfilling the usual conditions.

The main objective of this project is to estimate the daily volatility and assess the effect of the micro-structure noise lying within financial high-frequency data. As sample data, in what follows, we will only consider one security : the IVE index also known as **iShares S&P 500 Value Index (ETF)**.

1.1 Objectives

We have been given a research project based on the analysis of high-frequency financial time series. Our work will be decomposed in four different steps :

(1) Estimate and plot the values of the estimated realized volatility when using various observation frequencies ranging from 30 seconds to 15 minutes.

(2) Compare these estimations with the long range estimation of the volatility (based on 1 month of daily data)

(3) Provide some estimation of the market micro-structure noise size, by using the autocorrelation between the returns at different scales

(4) Plot the evolution of the estimated daily volatility of the IVE over the last year

These four subgoals will compose the outline of the present report.

1.2 Dataset description

We have been given a dataset containing high-frequency financial data representing history of trade operations over the IVE index.

Remark 1. *We denote* **T** *as the set of datetimes corresponding to each trade occurring between 2009-09-28 and 2022-10-28.*

The dataset is initially composed of 6 different columns :

Date	Time	Price	Bid	Ask	Size
...	...	...	...	...	...
Composed-index column		Relevant	Useless data *to be dropped*		

After some transformation by combining **Date** and **Time** together in order to get a *datatime* object as index column, we only have one column : the **Price**[1] acting as an univariate time-series ; the **Bid**, **Ask** and **Size** time-series are automatically dropped.

Date/Time	Price
...	...

Finally, after some research, the value of the IVE S&P index seems to be **only updated thanks to markets movements from** 9 : 30 **to** 16 : 00. However, for plotting reasons, we will keep our actual time range (9 : 00 to 17 : 30) in our implementation.

Remark 2. *A more accurate discussion will be presented in Conclusion on the use of OHLC data in order to refine our volatility estimators.*

1.3 Pre-processing & Preliminary steps

As we have automatically performed the pre-processing steps described above during the import phase, the remaining part is to implement function to automate the plotting of the high-frequency data.

1.4 Considered model specifications

We denote by $(S_t)_{t\geq 0}$ the price process of the considered index. In the following sections, we will also use the log-price process $(X_t)_{t\geq 0}$ in order to deal with the log-returns time series :

$$\forall t \in \mathbf{T},\ X_t := \log(S_t)$$

1. We have assumed during this research project that the price of the asset at time $t > 0$ is given by the value of the closest (in time) past transaction.

We assume the following expression driving the $(X_t)_t$ process :

$$X_t = \mu t + \sigma B_t$$

where $(B_t)_{t\geq 0}$ is a $(\mathbb{F}, \mathbb{Q})$ Brownian motion [2], μ & $\sigma \in \mathbb{R}$. Right here, the volatility σ is considered constant. We currently want to estimate σ^2 from the log price process $(X_t)_t$

1.5 Data visualization

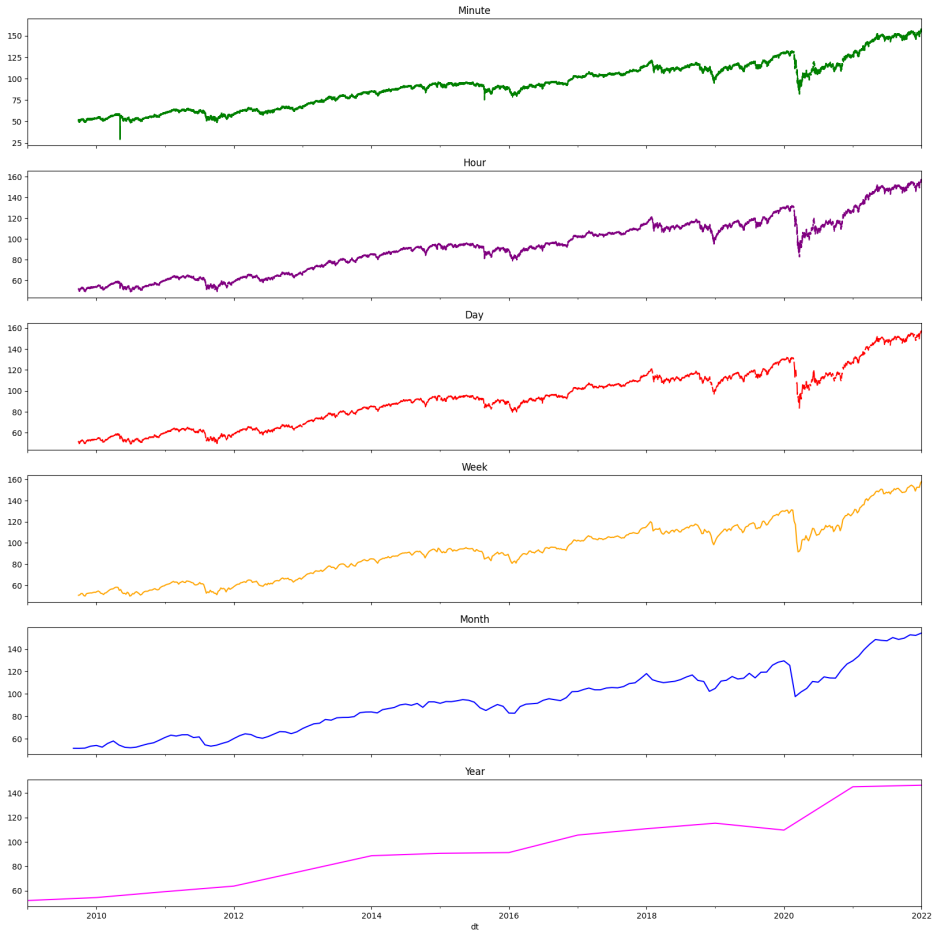

FIGURE 1 – Visualization of the price time-series for various resampled frequencies

2 Answer

Each sub section represents the complete answer to one given question.

2.1 Estimation of the realized volatility for various frequencies

We want to estimate the realized volatility for various observation frequencies. The current observation frequency, referred as f is the directly linked to n, the number of trade operations performed **within one day**. For the sake of simplicity, we will only use the n quantity from now on.

2. $\mathbb{Q}$ represents the risk-neutral probability measure.

Definition 1 (Estimated realized volatility)**.** *The realized volatility* Q_n [3] *assesses variation in log-returns for an investment product by analyzing its historical returns within a defined time period.*

The realized volatility is computed as follows :

$$Q_n := \sum_{i=0}^{n-1} \left(X_{\frac{i+1}{n}} - X_{\frac{i}{n}} \right)^2$$

A consistent estimator $\widehat{Q}_n$ of Q_n is defined as follows :

$$\widehat{Q}_n := \sum_{i=0}^{n-1} \left(\widehat{X}_{\frac{i+1}{n}} - \widehat{X}_{\frac{i}{n}} \right)^2$$

where :

- ▷ $\widehat{Q}_n$ is the **estimated realized volatility**.
- ▷ $\widehat{X}_{\frac{i+1}{n}}$ is the log-price in high-frequency, **observed on the market**.
- ▷ n is the number of data points at such frequency.

Proposition 1 (Asymptotical convergence)**.** We have :

$$\widehat{Q}_n \xrightarrow[\mathbb{P}]{n \longrightarrow +\infty} \sigma^2$$

The estimator $\widehat{Q}_n$ is **consistent** when $n \longrightarrow +\infty$ (specification of high-frequency financial data settings).

Remark 3 (More general model)**.** *For a more refined model whose dynamic is :* $\mathrm{d}X_t := \mu(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}W_t$*, we have :*

$$\widehat{Q}_n \xrightarrow[\mathbb{P}]{n \longrightarrow +\infty} < X >_1 = [X, X] = \int_0^1 \sigma^2(X_s)\mathrm{d}s$$

This is why $\widehat{Q}_n$ *helps estimating the integrated volatility.*

Thanks to this result, we will be able to compute a clear estimation of the realized volatility. Let's take a look at our strategy :

Simulation frequencies We have decided to conduct the study on various observation frequencies ranging **from 30 seconds to 15 minutes** [4].

30 sec, 32 sec, 35 sec, 40 sec, 45 sec, 50 sec, 1 min, 1 min 30, 2 min, 2 min 30, 3 min, 5 min, 8 min, 10 min, 12 min, 15 min

We then adopt and implement this protocol :

3. We can draw a parallel with the $< \cdot >= [\cdot, \cdot]$ (squared brackets) of a stochastic process.

4. If we take frequencies higher than 15 minutes, we are not anymore in the high-frequency domain. Likewise for frequencies lower than 30 seconds, the obtained results are assumed to be **too subject to noisy behaviors** (see question 3).

Strategy We perform the computation using the following steps :

(1) Get the financial data (trading prices for each operation) from the considered day d (log-price process $(X_t)_t$)

(2) Re-sample the obtained time-series with the given observation frequency

(3) Compute the log-returns series $(R_t)_t$, defined as follows :

$$R_i := R_i^{\log} = X_i - X_{i-1} = \log S_i - \log S_{i-1} = \log \frac{S_i}{S_{i-1}}$$

(4) Fill the NaN values with value 0 in order to avoid any computation error

(5) Apply the formula of $\widehat{Q}_n$ to estimate the realized volatility

Results For the day of 2022/01/03, we obtained the following results :

Frequency	Est. vol. $\widehat{\sigma}$
30S	0.0000285384
32S	0.0000293432
35S	0.0000280095
40S	0.0000311544
45S	0.0000319695
50S	0.0000324756
1M	0.0000363214
90S	0.0000364157
2M	0.0000368909
150S	0.0000364330
3M	0.0000319507
5M	0.0000303374
8M	0.0000292673
10M	0.0000191933
12M	0.0000232185
15M	0.0000220643

TABLE 1 – Obtained results from our experiment (check graph evolution on 2)

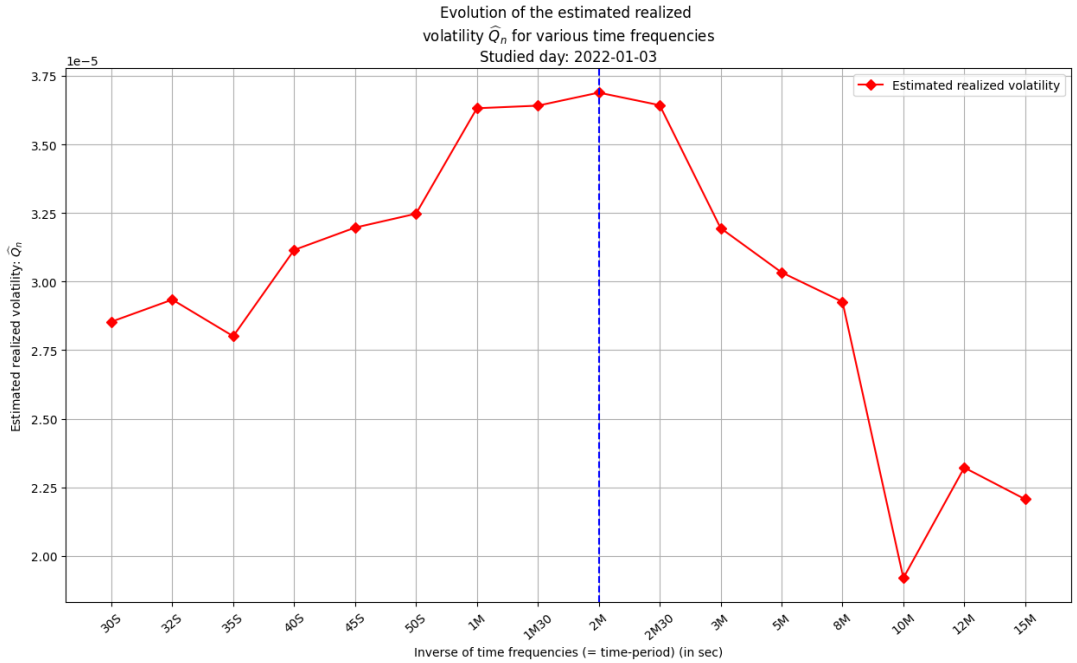

FIGURE 2 – Evolution of the estimated realized volatility for different observation frequencies (numerical data on 1)

Conclusion We have successfully computed some estimations of the realized volatility for various observation frequencies for a fixed time period (a day in our case).

We can see that the more the lower period is (the higher the observation frequency), the more convergent the estimated realized volatility is. We can clearly split the obtained graph (fig. 2) into two independent regions :

- ▷ the one (on the left) with the higher time frequencies corresponding to $n \longrightarrow +\infty$ where we can see a convergence of the $\widehat{Q}_n$ quantity
- ▷ the other one (on the right) with the higher time periods (lower frequencies) where we can clearly shed light on a divergence behavior with some oscillations for $T = 10M$.

This split can be explaiend by the shift between high and low frequencies domains within our sampling method.

As explained in [5], we can segment the trading observation frequencies in three distinct categories [6] :

Level	Specific time-frame
Low-frequency	4 hours to a few days
Medium-frequency	2 to 15 min / 15 min to 4 hours
High-frequency	Micro seconds to 60/90 seconds

TABLE 2 – Different categories of trading frequencies

As we can see in 2, the limit of 1-2 minutes represents the effective frontier from high to medium frequency trading ; this can explain the change on the curve 2.

5. https://www.bots.io/botspedia/the-differences-between-high-and-low-frequency-trading-bots
6. It depends on the source found in the literature.

Remark 4. *However it's important to point out that the literature does not have any common understanding on the exact time-frames for each kind of trading.*

Additionally, the observed order of magnitude for these intraday estimated realized volatility is of 10^{-5}.

We can observe a small convergence behavior from the 1-min to 30-sec frequencies, converging towards $\boxed{\sim 2.8 \times 10^{-5}}$.

However, as we said earlier, going outside the interval 30 sec - 15 min is not relevant in our case of high-frequency financial data study.

2.2 Comparison with long range estimation of the volatility

After having computed the estimation of realized volatility for various observation frequencies, we now want to compare it to long range estimation using monthly-based data.

To do so and as we have studied the estimated volatility properties for random day of January 2022 in the first question, we will now focus on the log price evolution during that month, which is resampled on a daily-basis.

$\longrightarrow$ We have, regardless NaN values due to holidays and week-ends, 31 data points.

As we are not anymore in a high-frequency setting, we have to apply the standard estimator for the long-range volatility, given by the **emprical variance estimator** as follows :

$$\widehat{\sigma}_{\text{day}} := \sqrt{\frac{1}{T-1}\sum_{i=1}^{T}\left(R_i - \overline{R}_T\right)^2}$$

where :

$$\overline{R}_T := T^{-1}\sum_{i=1}^{T} R_i$$

with $R_1, \ldots, R_T$ are the **observed log returns** and $T = 21$, the number of trading days within 1 month **in average**.

This is the standard deviation of the log-returns time-series, used to compute the long-range estimation of volatility during this month.

After numerical computation, we obtained :

$$\boxed{\sigma^{\textbf{monthly}}_{\text{long-range}} \simeq 9.9598 \times 10^{-2}}$$

We finally check the evolution of the *relative* difference between this long-range estimation $\sigma^{\textbf{monthly}}_{\text{long-range}}$ and estimated realized volatility for various values of n.

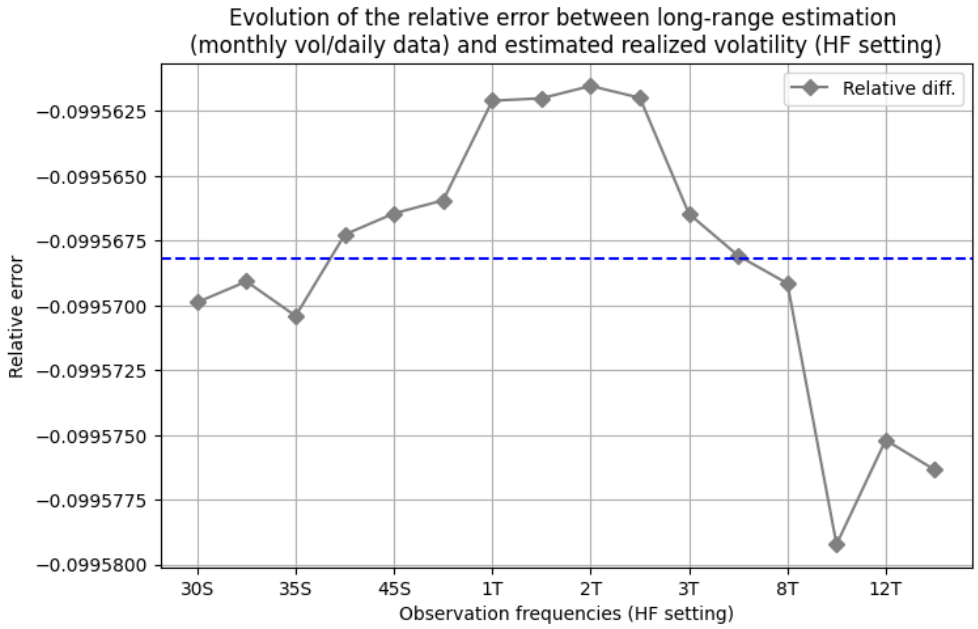

FIGURE 3 – Evolution of the relative error between estimated long-range volatility (monthly-based/daily data) and the estimated realized volatilities (HF setting)

Observations We can state that the two studies do not contain the same order of magnitude : 10^{-5} for high-frequency samples vs. 10^{-2} for long-range estimation of the volatility.

We can also observe, as a complement that the relative error between the obtained volatilities from the two settings (intra-day HF vs. long-range) seems small (~ -0.0995 in average).

After having implemented estimators of realized volatility for various sample frequencies, we have observed some convergence results in practice but also the limits of sampling with too heavy observation frequencies, driving us to irrelevant results. We now have decided to **estimate** the size of the market micro-structure noise ϑ.

2.3 Estimation of ϑ : micro-structure noise size

However, the measures of realized volatility over a high-frequency financial dataset present a major problem : it includes parasite information also known as *micro structure noise*.

The source of this noise can be explained by various factors [Yac09] [Gus20] : limit order books movements, trade operations, market participant behaviors, ... Market micro-structure noise captures a variety of frictions inherent in the trading process : bid–ask bounces, discreteness of price changes, differences in trade sizes or informational content of price changes, gradual response of prices to a block trade, strategic component of the order flow, inventory control effects, etc. [7]

We have been provided various theoretical results relatively to this micro structure noise, whose size is denoted as ϑ. The main objective of this section is to estimate its

7. https://www.princeton.edu/~yacine/liquidity.pdf

value thanks to the use of auto-correlation function. We are performing a very common operation in time series processing : the signal/noise decomposition [8]

$$\widehat{X}_{\frac{i}{n}} := X_{\frac{i}{n}} + \vartheta\varepsilon_{\frac{i}{n}}, \ \forall i \in [\![0, n-1]\!]$$

where :

- ▷ $\widehat{X}_{\frac{i}{n}}$: the observed price in high frequency settings
- ▷ $X_{\frac{i}{n}}$: the latent price (true value of the asset) **but not observable**
- ▷ $\varepsilon_{\frac{i}{n}}$: the noise process
- ▷ ϑ : the size of the noise

$\longrightarrow \vartheta\varepsilon_{\frac{i}{n}}$ is denoted as the **micro-structure noise of the market**.

General assumption : We assume that the noise process $(\varepsilon_{\frac{i}{n}})_{i\in[\![0,n]\!]}$ is an independent and identically distributed (iid) sequence following the $\mathcal{N}(0,1)$ law [9].

$$\begin{aligned}\widehat{X}_{\frac{i+1}{n}} - \widehat{X}_{\frac{i}{n}} &= X_{\frac{i+1}{n}} - X_{\frac{i}{n}} + \vartheta\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right) \\ &= \underbrace{\sigma\left(B_{\frac{i+1}{n}} - B_{\frac{i}{n}}\right)}_{\text{size } \frac{\sigma}{\sqrt{n}}} + \underbrace{\vartheta\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right)}_{\text{size } 2\vartheta}\end{aligned}$$

When n is large, $\frac{\sigma}{\sqrt{n}} \ll 2\vartheta$, which points out that the noise dominates.

Remark 5 (Noise composition)**.** *A relevant extension to this research project would to be consider a different probabilistic distribution for the noise process.*

Remark 6 (Magnitude of ϑ)**.** *The quantity $\vartheta \in \mathbb{R}_*^+$ has, in general, a **low order of magnitude**. This remark will help us to validate our obtained numerical results.*

To conduct the estimation of ϑ, we can adopt two different strategies :

(1) using an asymptotic result using the previously-computed realized volatilities

(2) using a result on 1-lag auto-correlation function from the log-returns time series

In order to compare our results, we have chosen to perform these two methods.

2.3.1 Estimation using asymptotic result

The first idea would be to use an asymptotic convergence property. As a reminder, we have this result :

$$\frac{\widehat{Q}_n}{n} = \frac{Q_n}{n} + \frac{o(1)}{n} + \frac{\vartheta^2}{n}\sum_{i=0}^{n-1}\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right)^2$$

Proposition 2 (Estimator of ϑ)**.**

$$\boxed{\frac{\widehat{Q}_n}{n} \xrightarrow[\mathbb{P}]{n\longrightarrow+\infty} 2\vartheta^2}$$

8. We decompose the observed signal as a sum or linear combination of two components : the real signal (not observed) and a noise process (which include randomness).

9. As an extension of this project, we could consider other probabilistic distribution such as : t-dist (Student) or other heavy-tailed distributions.

So, thanks to this theoretical result [10], we can recover a good estimator of ϑ.

$$\boxed{\vartheta = \lim_{n \to +\infty} \sqrt{(2n)^{-1} \widehat{Q}_n}}$$

In order to conduct a small comparison study on this estimation, we will estimate this quantity with the previous values of the estimated realized volatility computed in the section above. This will help us to study the impact of the observation frequency on the micro structure noise size estimation and verify the theoretical result regarding the convergence as $n \longrightarrow +\infty$.

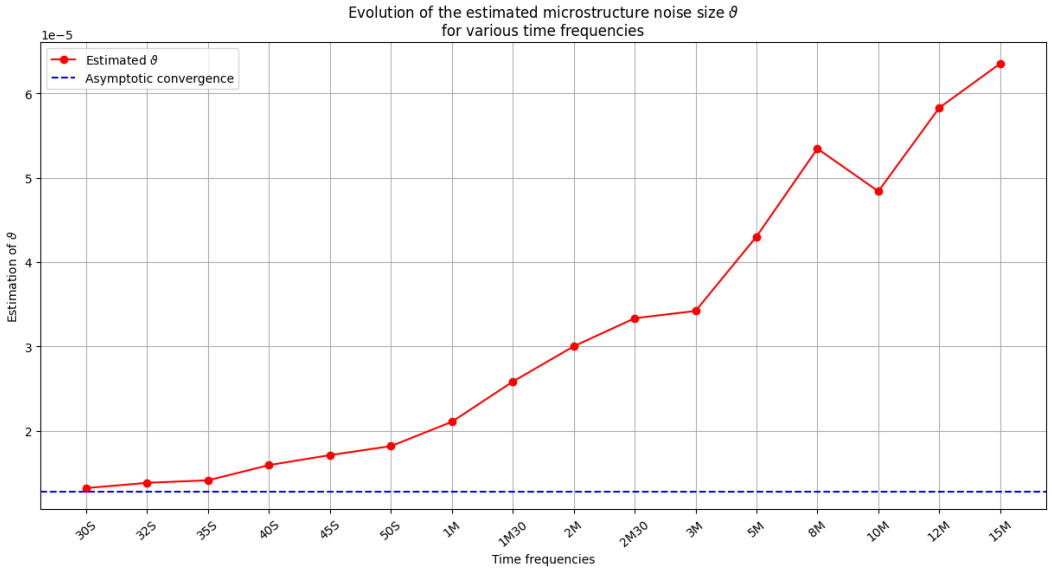

FIGURE 4 – Evolution of the estimated micro structure noise size ϑ for various time frequencies

Conclusion We observe a satisfying convergence result, where the estimation of ϑ seems to be about $\boxed{\vartheta \simeq 1.2 \times 10^{-5}}$.

Remark 7. *We cannot test for higher values of n (higher frequency or lower period) because of the noise effect on the computations of $\widehat{Q}_n$ which directly intervenes within the calculation of our estimator.*

Let's now take a look at our second approach.

2.3.2 Estimation using auto-correlation between the returns at different scales

A second approach is to use the 1-lag auto-correlation series for the log-returns series :

Definition 2 (Auto-covariance series)**.** *We define, $\forall h \in \mathbf{Z}$*

$$\gamma(h) := \mathrm{Cov}(X_t, X_{t+h})$$

where γ does not depend on t

10. We can prove that the realized volatility is contaminated by the noise. See (3)

Definition 3 (Estimator of 1-lag auto-covariance series)**.** *We now define the estimator of the 1-lag auto-covariance series for the log-price series*

$$\widehat{\gamma}_n(1) := \text{Cov}\Big(\widehat{X}_{\frac{i+1}{n}} - \widehat{X}_{\frac{i}{n}}, \widehat{X}_{\frac{i+2}{n}} - \widehat{X}_{\frac{i+1}{n}}\Big)$$

We will use $\widehat{\gamma}_n(1)$ to estimate ϑ. As a reminder, we have the following results :

Log-price series

$$\begin{aligned}\gamma_n(1) = \text{Cov}\Big(X_{\frac{i+1}{n}} - X_{\frac{i}{n}}, X_{\frac{i+2}{n}} - X_{\frac{i+1}{n}}\Big) &= \sigma^2\text{Cov}\Big(B_{\frac{i+1}{n}} - B_{\frac{i}{n}}, B_{\frac{i+2}{n}} - B_{\frac{i+1}{n}}\Big) \\ &= 0\end{aligned}$$

But, we are only interested by the noise-contaminated log-price series :

Contaminated log-price series

$$\begin{aligned}\widehat{\gamma}_n(1) = \text{Cov}\Big(\widehat{X}_{\frac{i+1}{n}} - \widehat{X}_{\frac{i}{n}}, \widehat{X}_{\frac{i+2}{n}} - \widehat{X}_{\frac{i+1}{n}}\Big) &= \sigma^2\text{Cov}\Big(B_{\frac{i+1}{n}} - B_{\frac{i}{n}}, B_{\frac{i+2}{n}} - B_{\frac{i+1}{n}}\Big) \\ &\quad + \vartheta^2\text{Cov}\Big(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}, \varepsilon_{\frac{i+2}{n}} - \varepsilon_{\frac{i+1}{n}}\Big) \\ &= 0 - \vartheta^2\text{Var}\Big(\varepsilon_{\frac{i+1}{n}}\Big) \\ &= -\rho^2 < 0\end{aligned}$$

However, since $(\varepsilon_{\frac{i}{n}})_{i\in\{0\dots n\}}$ is an iid sequence $\mathcal{N}(0,1)$, we have $\text{Var}(\varepsilon_{\frac{i}{n}}) = 1$.

$$\Longrightarrow \widehat{\gamma}_n(1) = -\vartheta^2 \Longrightarrow \boxed{\vartheta \simeq \sqrt{|\widehat{\gamma}_n(1)|}}$$

Some observations :

- ▷ The micro-structure noises induce negative correlation coefficient for high-frequency returns
- ▷ The estimation of ϑ can be performed using the estimation of the first auto-covariance coefficient $\widehat{\gamma}_n(1)$.
- ▷ It turns out that we can observe these auto-correlation on real data.

In order to deal with the $\rho(\cdot)$ and $\gamma(\cdot)$ functions (respectively auto-correlation and auto-covariance functions) and avoid any misunderstanding, we redefine them in our Python implementation.

General assumption : Since **the lag has to stay fixed**, we can only modify the sampling frequency in our study. However, as we want to study the impact of very high fobservation requency on the ϑ estimation, we will use additional frequencies as defined in the first question.

$\longrightarrow$ We will add : 1, 2, 3, 5, 8, 10, 12, 15, 20 and 25 seconds [11] as new high-frequencies, to study the impact of HF on $\widehat{\rho}_n(1)$, $\widehat{\gamma}_n(1)$ and the final estimation of ϑ.

Moreover, we will use the financial log-price for a short period of time : for instance, one fixed day (in our case : 2022/05/13).

11. It is not relevant to add higher frequency ($<$ 1 sec) because of the original dataset sampling.

Results We obtain the following graphs :

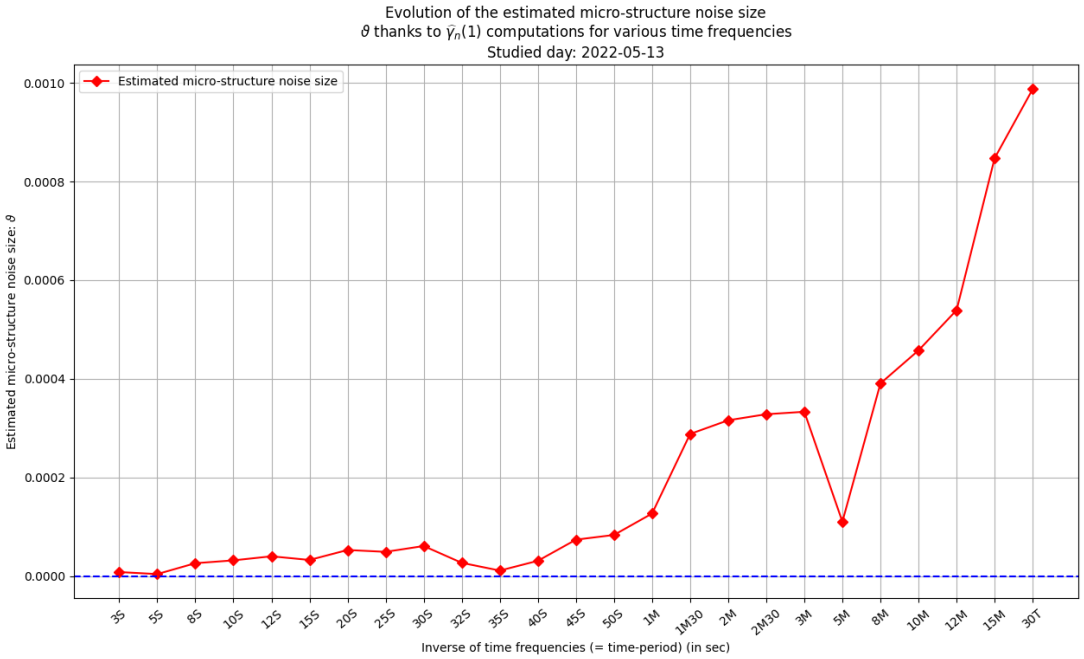

FIGURE 5 – Evolution of the estimated micro structure noise size ϑ for various time frequencies

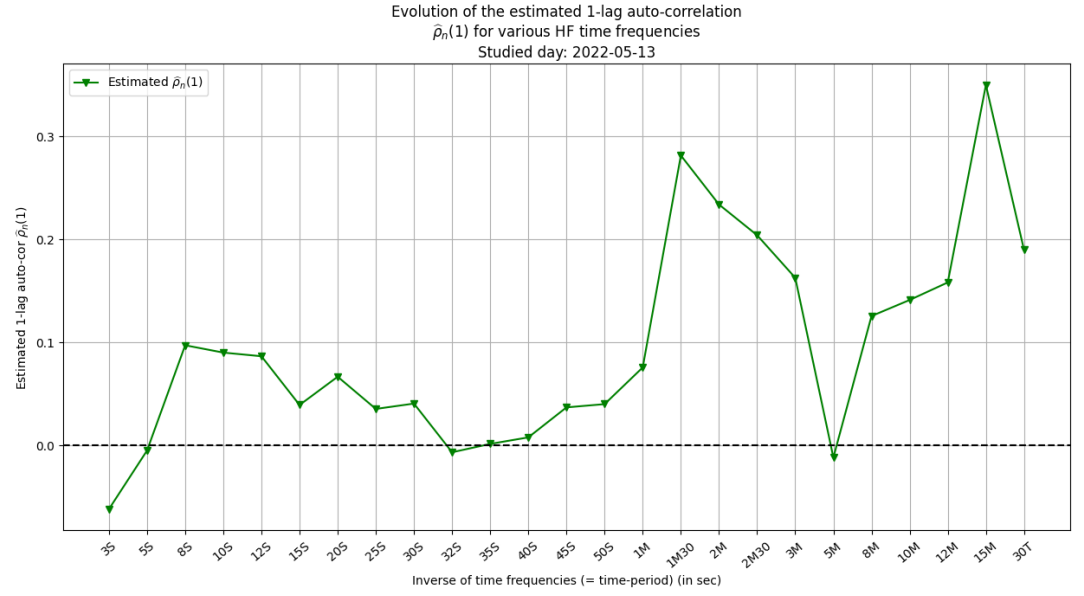

FIGURE 6 – Evolution of the estimated 1-lag auto-correlation $\widehat{\rho}_n(1)$ for various HF time frequencies

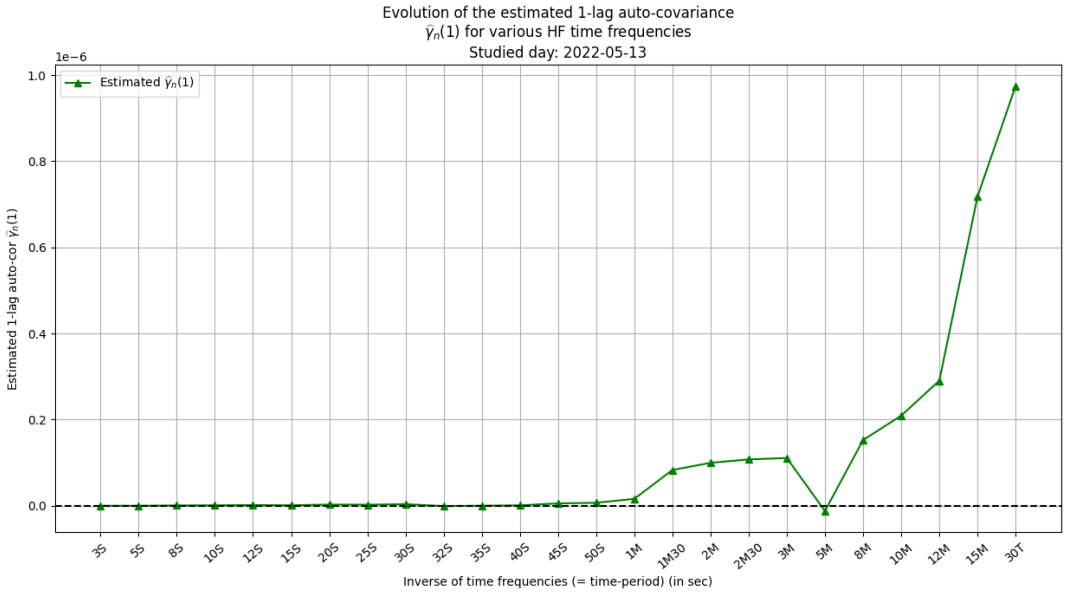

FIGURE 7 – Evolution of the estimated 1-lag auto-covariance $\widehat{\gamma}_n(1)$ for various HF time frequencies

Observations & Discussion First of all, we can see a clear convergence behavior on fig 5, going towards a really small value (near 0), which verifies the theoretical results which stays ϑ has clearly a small order of magnitude.

If we now look at the 1-lag auto-correlation and auto-covariance evolution with respect to the observation frequency, we can clearly recover a negative auto-correlation on 6, as proved in the course.

As for 7, the $\widehat{\gamma}_n(1)$ is converging very quickly to 0.

Frequency	Est. ϑ
1S	0.0000027799
2S	0.0000080611
3S	0.0000070197
5S	0.0000175452
8S	0.0000484120
10S	0.0000686855
12S	0.0000725810
15S	0.0000893070
20S	0.0001251127
25S	0.0001391273
30S	0.0001350808
...	...

TABLE 3 – Obtained results from our experiment (check graph evolution on 5) (Truncated to simplify)

If we look at the numerical results in details, we can clearly estimate that the order of magnitude, recovered thanks to this method, is around 10^{-5} or 10^{-6} depending on our point of view on high-frequency/ultra high-frequency threshold.

$\longrightarrow$ This magnitude **perfectly matches** the results from the first method using the consistent estimator of ϑ.

As a complement, we have plotted the ACF (autocorrelation function) on log-returns time-series over **T** on (14) and (15).

2.4 Evolution of the estimated daily volatility of the IVE over the last year

First, we wanted to answer the question by using financial data from 2021. However, we have assumed that taking the log-price process from 2020 would be **clever** and **more relevant** as we would be able to highlight market volatility peaks on the IVE. In fact, the two first global Covid surges can be directly seen on the resulting graph.

Introduction We are asked to estimate the daily volatility over 1 fiscal year. This means we want to compute the volatility for each of the $T = 252$ trading days within one year (here : 2020).

We will **sample our financial time-series with a 30-min frequency** in order to not be disturbed by the previously-studied micro-structure noise.

We will compute one Q_n for each day corresponding to this trading year, and we will get a $\mathbb{R}^+$-valued series $(Q_n^i)_{i\in[\![1,T]\!]}$ where Q_n^i denotes the estimated realized volatility for the i-th day of the considered year.

Remark 8. *However, we can already make a comment on the small number of samples for each day to compute our Q_n^i, since the intra-day data will be re-sampled with respect to a 30-minute scale. There will undoubtedly be **estimation error** for each day.*

$\longrightarrow$ This phenomena will certainly issue on the implementation of smoothing/other processing methods in order to "stabilize" the estimation.

Data visualization We first plot the evolution of the log-price and log-returns processes *(sampled and not sampled)* over the given year.

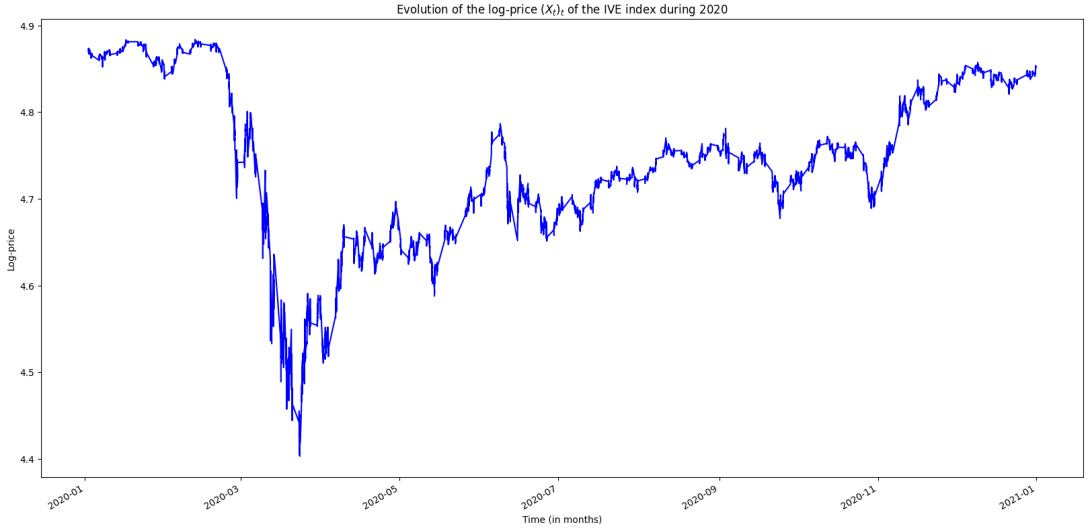

FIGURE 8 – Log-price evolution of the IVE index over the FY 2020

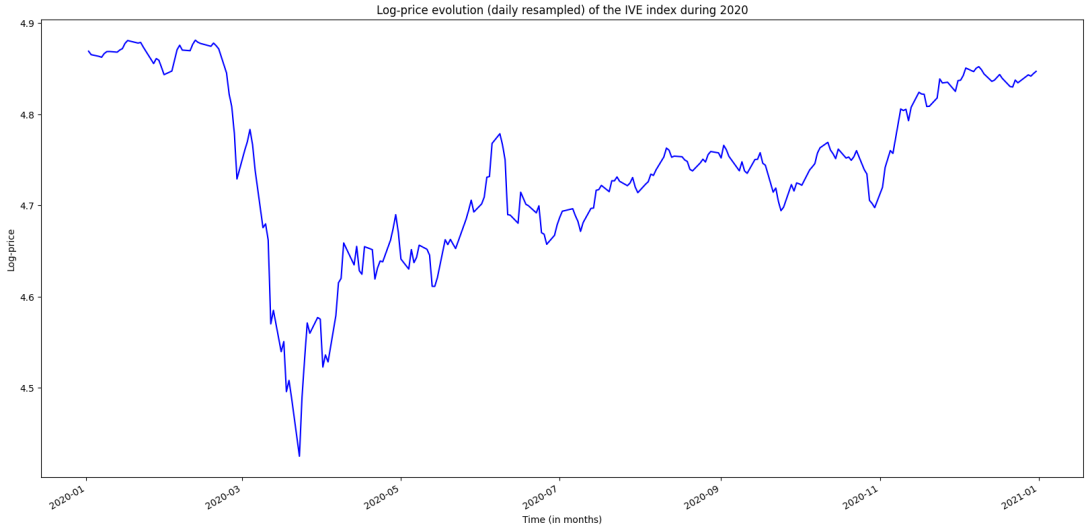

FIGURE 9 – Log-price evolution **(daily resampled)** of the IVE index over the FY 2020

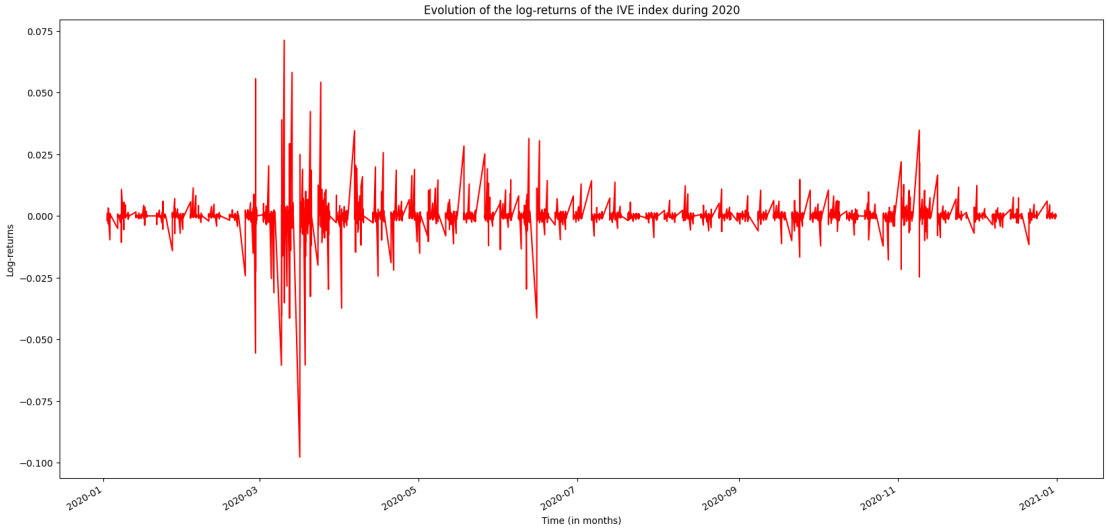

FIGURE 10 – Log-returns evolution of the IVE index over the FY 2020

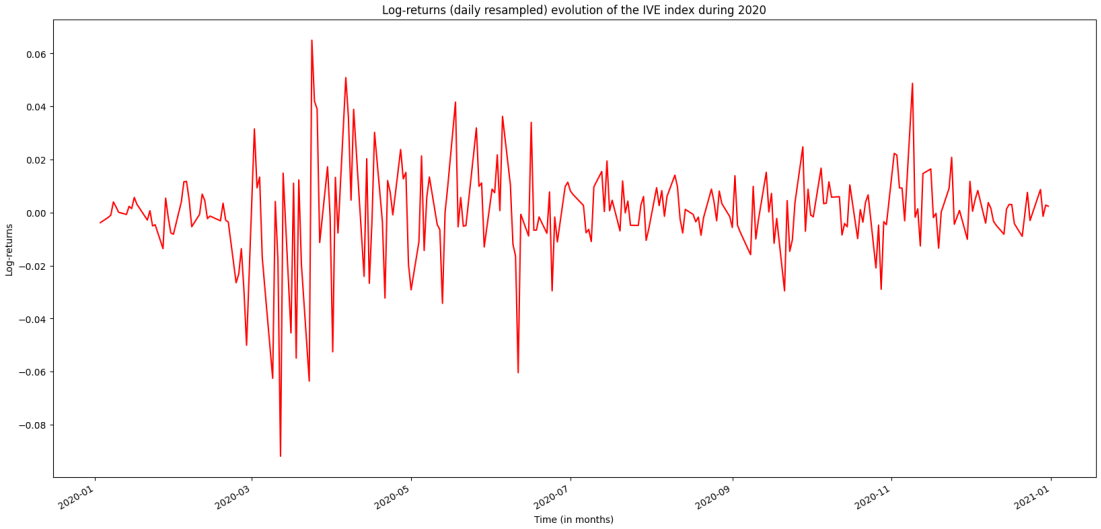

FIGURE 11 – Log-returns evolution **(daily resampled)** of the IVE index over the FY 2020

Results Finally, after computations, we obtain the following graph :

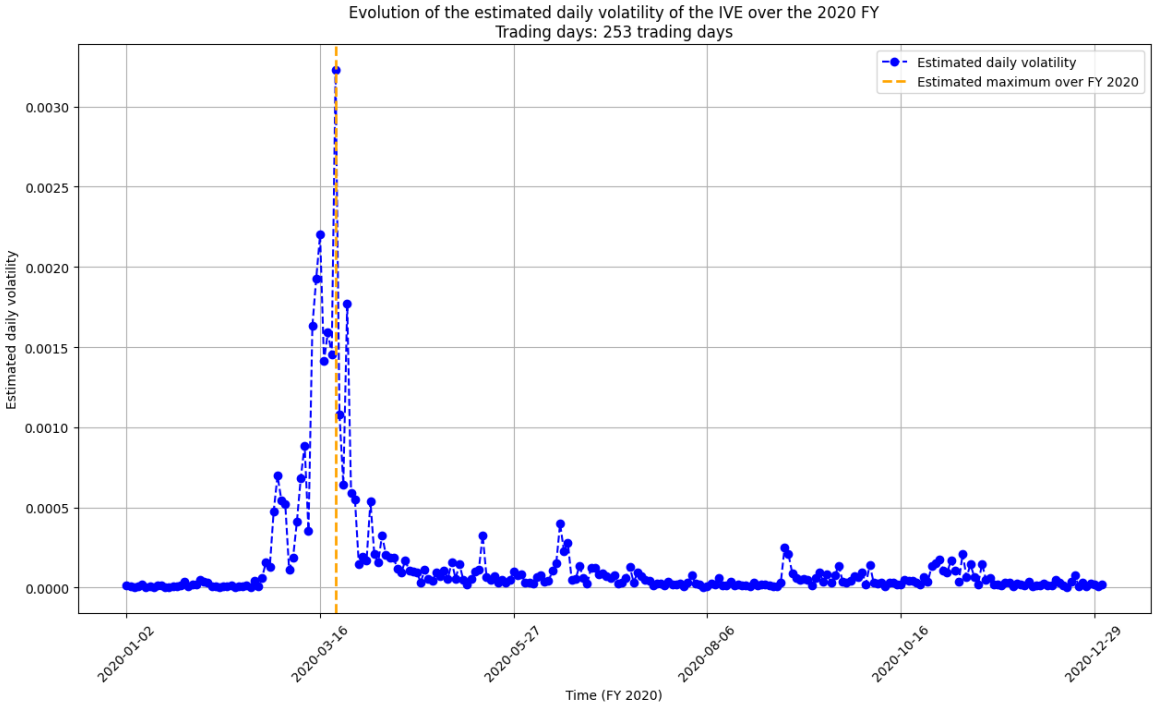

FIGURE 12 – Volatility evolution of the IVE index over the FY 2020

In **orange**, we have highlighted the maximum of estimated daily volatility over the year 2020.

Results interpretation We have successfully computed and drawn the evolution of the estimated daily volatility. We can easily see a high peak related to the first Covid

surge (March 2020), which had importantly impacted the global financial market.

Smoothing In order to "stabilize" the previous estimation, one can apply smoothing methods on our time-series [Eas22] :

Definition 4 (Smoothing)**.** ***Smoothing*** *is a time-series method usually done to help us better see patterns, trends for example, in time series. Generally smooth out the irregular roughness to see a clearer signal.*

The term ***filter*** *is sometimes used to describe a smoothing procedure.*

One may compute the "final" volatility using rolling windows for instance; there exists multiple strategies [Lef20] to do so but we will only consider the two following ones :

(1) **Simple Moving Average** (SMA)

(2) **Exponentially Weighted Moving Average** (EWMA)

Let's briefly defined their underlying mathematical definitions :

Definition 5 (Simple Moving Average)**.** *Let $(X_t)_t$ a $\mathbb{R}$-valued time-series. We define the SMA ψ with characteristics $(m_1, m_2, \Theta := (\theta_{-m1}, \dots, \theta_{m_2}))$ as follows :*

$$\psi X_t := \sum_{k=-m_1}^{m_2} \theta_k X_{t+k}$$

This filter has several characteristics; for sake of simplicity, we have decided to take it : **normalized**, **centered** and **symmetrical**. [Lef20]

Definition 6 (Exponentially Weighted Moving Average)**.** *Let $(X_t)_t$ a $\mathbb{R}$-valued time-series. We define the EWMA Γ with characteristics α as follows :*

$$\Gamma X_t := (1-\alpha) \sum_{s=0}^{t-1} \alpha^s X_{t-s}$$

Remark 9. *The observations have all the more weight because they are recent; the weight α^s increases when s decreases.*

After several empirical tests, we have decided to set the following hyper-parameters :

- ▷ for SMA $\rightsquigarrow$ windows size $\boxed{m = 10}$ days
- ▷ for EWMA $\rightsquigarrow$ $\boxed{\alpha = 9 \times 10^{-2}}$ [12]

because of the good smoothing effect they are providing to the firstly obtained volatility curve.

Remark 10. *In addition, their respective numerical determination were performed* ***to avoid any under/over-smoothing behaviors****.*

Here's the results :

12. In theory, $\alpha \in]0, 1[$ and its value is entirely determined using an optimization algorithm.

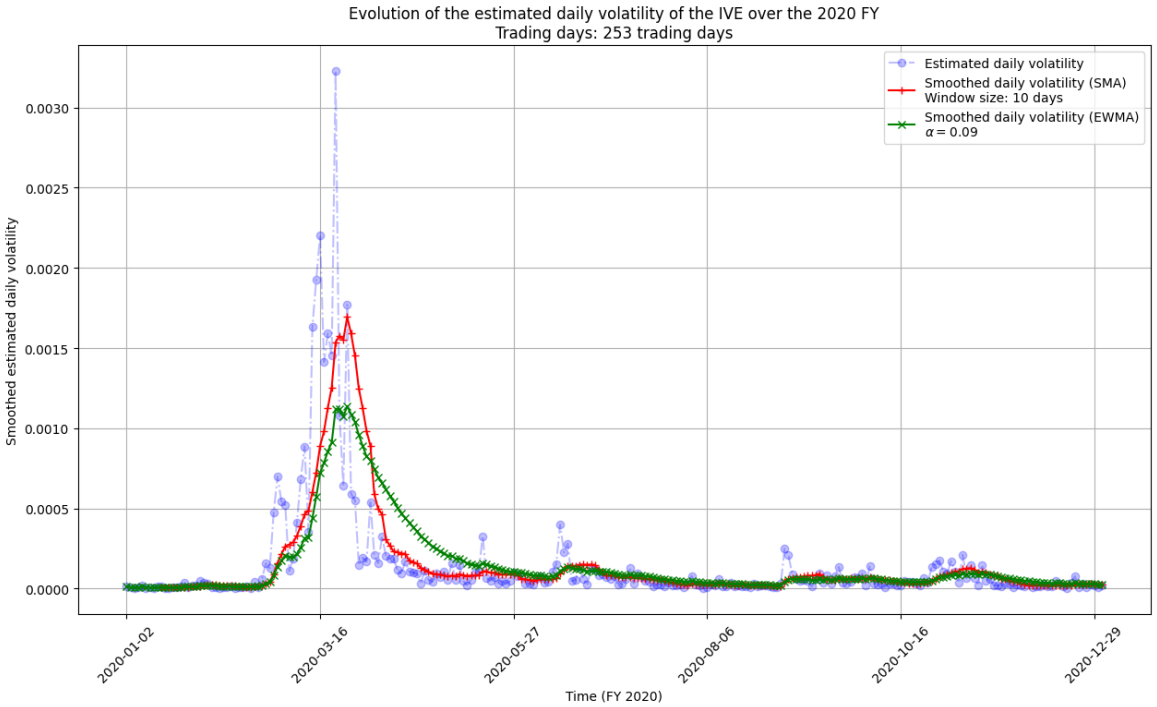

FIGURE 13 – Volatility evolution *(with smoothing)* of the IVE index over the FY 2020

The smoothed volatility curves tend to highlight in a more efficient way the general trend of the market moves over 2020. Even if they perfectly catch the volatility peaks due to COVID-19 pandemic, they are less sensitive to them.

Remark 11. *Other smoothing methods, as Double Exponential Moving Average (DEMA) for instance, can also be tested, as extension of this research project.*

3 Conclusion

This research project has allowed us to deal with high-frequency financial datasets and process them for insight-delivery purposes, through the computation of an estimated metric : the *market volatility*.

We have :

- ▷ computed and plotted the estimated realized volatility by varying the observation frequency, in order to study its impact on the estimation
- ▷ compared these estimations with longer-range ones (for instance monthly volatility)
- ▷ provided multiple estimations of the micro-structure noise size using different techniques
- ▷ computed and plotted the evolution of the estimated daily volatility of the IVE over a complete year.

These steps helped us to understand the main stakes and issues regarding the processing of high-frequency data.

Ouverture/Discussion However, we can shed light on some pitfalls or drawbacks of the used methodology ; in fact, the major part of this research project was about dealing with *estimated realized volatility*.

According to [Tsa05], advantages of realized volatility include simplicity and making use of intra-daily log returns. Intuitively, one would like to use as much information as possible by choosing a large n. However, when the time interval between $r_{t,i}$ tends to become smaller, the returns are subject to the effects of market micro-structure, for example, bid-ask bounce, which often result in a biased estimate of the volatility. The problem of choosing an optimal time interval for constructing realized volatility has attracted much research lately. For heavily traded assets on European/US financial exchanges, a time interval of 3-15 minutes is often used. Another problem of using realized volatility for stock returns is that the **overnight return**, which is the return from the closing price of day $t-1$ to the opening price of t, **tends to be substantial**. Ignoring overnight returns can seriously underestimate the volatility. On the other hand, our limited experience shows that overnight returns appear to be small for foreign exchange returns or index returns, as the considered asset (IVE).

Over the last two decades, many solutions have been developed, both in the industry and in the literature to solve these two main problems : market micro-structure noise & overnight-returns.

In order to reduce the impact of the micro-structure noise size, we have different alternative solutions :

(1) Use of the realized kernel in order to remove the noise effect the and obtain a robust estimator of the volatility

(2) Use of the pre-averaging methods : Pre-averaging is a popular strategy for mitigating microstructure in high frequency financial data.

(3) Use of daily OHLC prices : For many assets including financial indices, daily opening, high, low and closing prices are available. One can use such information to improve volatility estimation. [Alm09]

Bibliography

[Tsa05] Ruey S Tsay. *Analysis of financial time series*. Wiley series in probability and statistics. Wiley, 2005.

[Alm09] Robert Almgren. "High Frequency Volatility". In : (déc. 2009). https ://-web.archive.org/web/20151229220319/http ://cims.nyu.edu/ almgren/time-series/notes7.pdf.

[Yac09] J Ialin Yacine Aït-Sahalia. "High Frequency Market Microstructure Noise Estimates and Liquidity Measures". In : *The Annals of Applied Statistics* 3.1 (2009). https ://www.princeton.edu/ yacine/liquidity.pdf, p. 422-457.

[Gus20] Cristina Mabel Scherrer Gustavo Fruet Dias Marcelo Fernandes. "Price discovery and market microstructure noise". In : - (nov. 2020).

[Lef20] Vincent Lefieux. *Analysez et modélisez des séries temporelles*. Online course. 2020. url : https://openclassrooms.com/fr/courses/4525371-analysez-et-modelisez-des-series-temporelles.

[Eas22] Ed Easterling. "Volatility In Perspective". In : *Crestmont Research* (jan. 2022). https ://www.crestmontresearch.com/docs/Stock-Volatility-Perspective.pdf.

Appendices

Contamination of the realized volatility by the noise $(\vartheta\varepsilon_{\frac{i}{n}})_{i\in[\![0,n-1]\!]}$

We can easily check that the realized volatility is contaminated by the micro-structure noise :

$$\begin{aligned}
\widehat{Q}_n &= \sum_{i=0}^{n-1}\left(\widehat{X}_{\frac{i+1}{n}} - \widehat{X}_{\frac{i}{n}}\right)^2 \\
&= \sum_{i=0}^{n-1}\left[X_{\frac{i+1}{n}} - X_{\frac{i}{n}} + \vartheta\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right)\right]^2 \\
&= \sum_{i=0}^{n-1}\left(X_{\frac{i+1}{n}} - X_{\frac{i}{n}}\right)^2 + 2\vartheta\sum_{i=0}^{n-1}\left(X_{\frac{i+1}{n}} - X_{\frac{i}{n}}\right)\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right) + \vartheta^2\sum_{i=0}^{n-1}\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right)^2 \\
\widehat{Q}_n &= Q_n + \underbrace{2\vartheta\sum_{i=0}^{n-1}\left(X_{\frac{i+1}{n}} - X_{\frac{i}{n}}\right)\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right)}_{\text{bounded as } n\to+\infty} + \underbrace{\vartheta^2\sum_{i=0}^{n-1}\left(\varepsilon_{\frac{i+1}{n}} - \varepsilon_{\frac{i}{n}}\right)^2}_{\text{goes to } +\infty \text{ as } n\longrightarrow+\infty}
\end{aligned}$$

We have a large positive bias as n goes to $+\infty$.

Log-returns computation

The log-returns computation process is deeply involved in this project ; however, we can find many different methods in the literature dealing with this quantitative aspect.

Here are three different approaches in Python to compute the series $(R_t)_t$ of log-returns of a given market asset.

(1) `.pct_change()` [13] $\rightsquigarrow$ This function computes **the percentage change** from the immediately previous row. This is useful in comparing the percentage of change in a time series of elements.

(2) `.shift(1)` [14] $\rightsquigarrow$ `.shift(1)` can be used to shift index by desired number of periods (here 1) and directly apply the formula necessary to compute $(R_t)_t$.

(3) `.diff()` [15] $\rightsquigarrow$ Calculates the difference of a DataFrame element compared with another element in the DataFrame (default is element in previous row).

To adopt a fixed methodology, we have decided, since the beginning of our research, to **only deal with the solution** (3), involving the use of `.diff()`, over a Pandas series representing the log price $(X_t)_t$.

ACF plots

As a complement, we can compute and plot the auto-correlation function evolution for the log-returns series, for various values of lag.

13. Doc : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.pct_change.html
14. Doc : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.shift.html
15. Doc : https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.diff.html

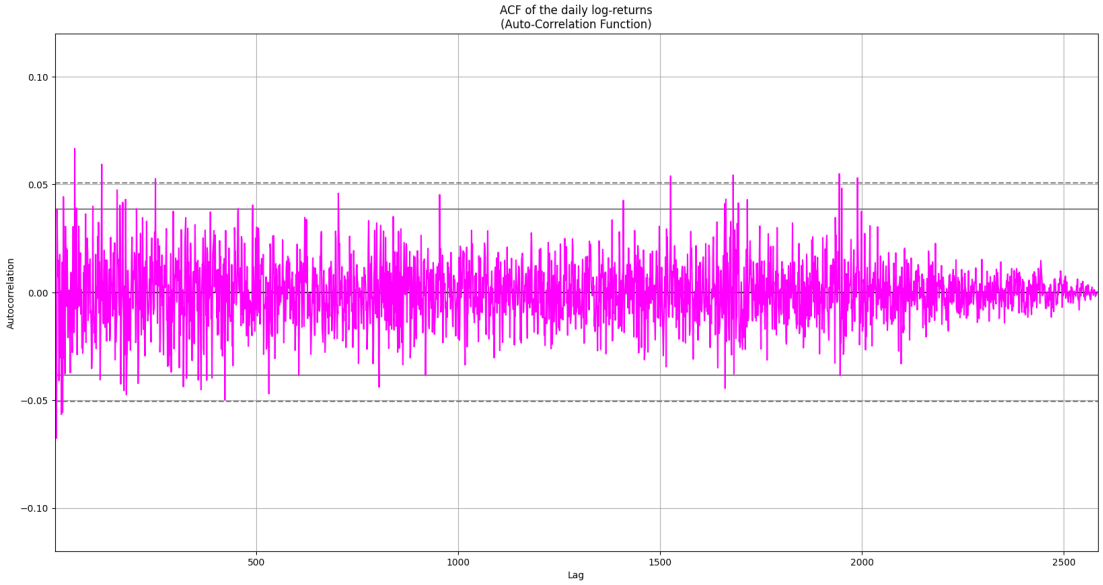

FIGURE 14 – Auto-correlation function plot for the log-returns on **T**

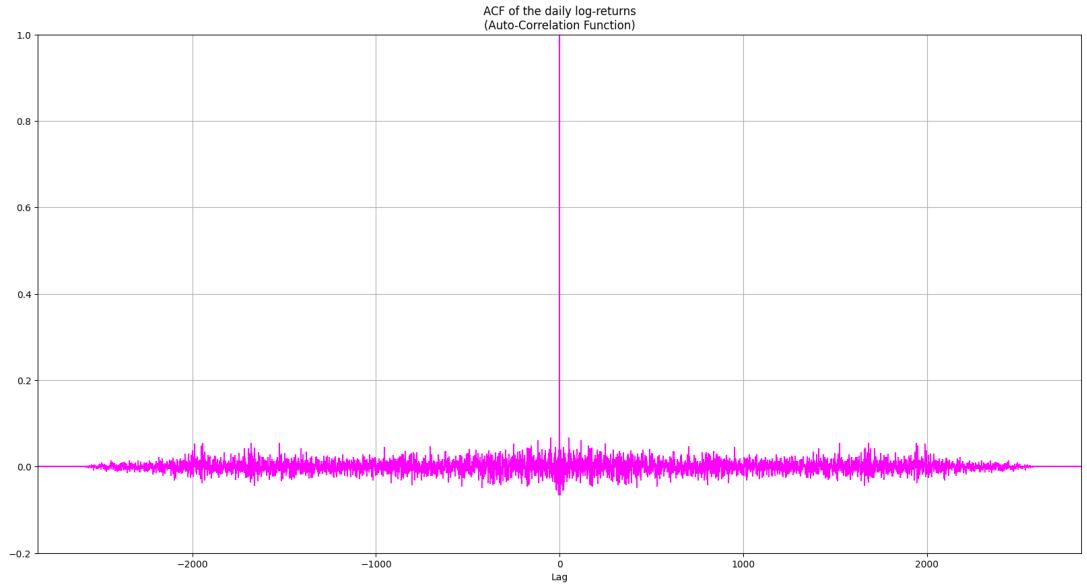

FIGURE 15 – Auto-correlation function plot for the log-returns on **T** (positive and negative lags)

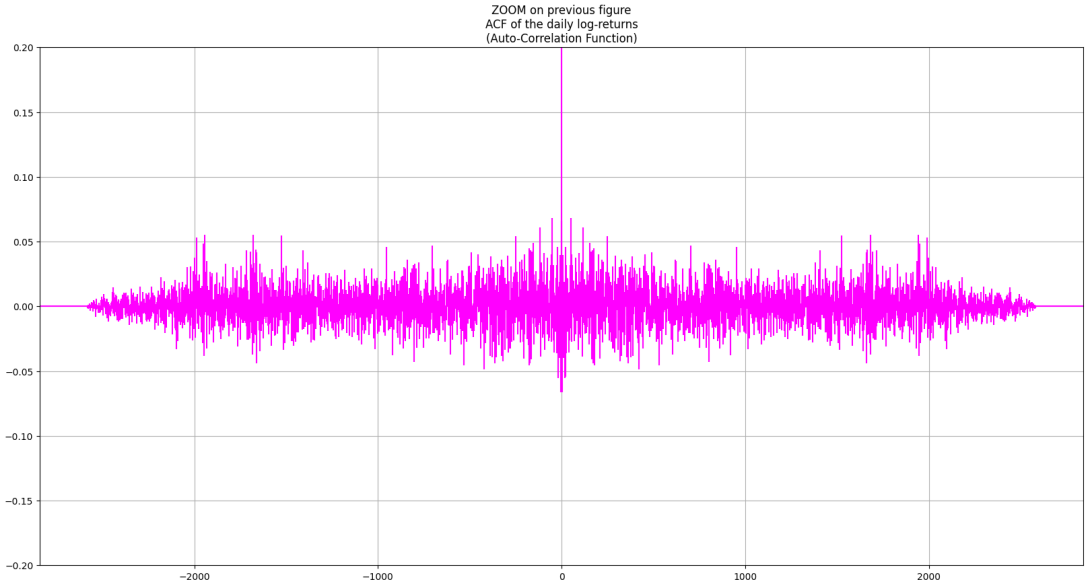

FIGURE 16 – Auto-correlation function plot for the log-returns on **T** (positive and negative lags)