**Rapport Final**

**Préambule**

Pour les questions traitées en Python, nous avons converti chaque fichier fourni en cvs, ordinateur paramétré sur séparateur points virgules et les nombres en format anglo saxo c’est-à-dire le séparateur des décimales correspond à un point. Il est important de prendre cela en considération pour que chaque programme fonctionne.

Nous avons ensuite fait appel à l’outils pandas de Python pour pouvoir stocker et parcourir aisément nos valeurs.

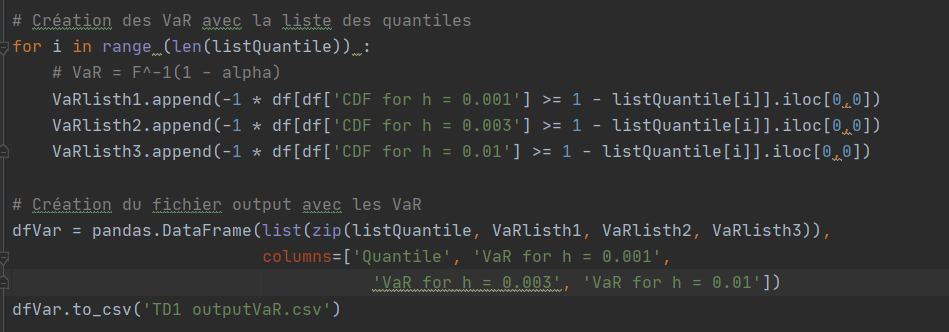
Nous avons également utilisé les modules de mathématiques pour pi, exponentielle, logarithme, etc.

**Question A**

**Fonctions utilisées :**

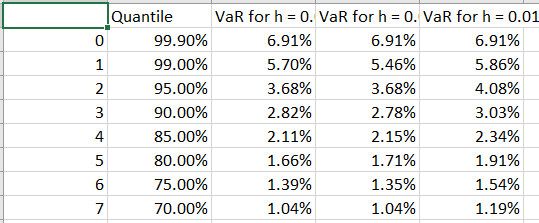
* **Kerneldensity :** fonction densité de la fonction kernel avec la formule du cours
* **KernelCDF :** Intégrale de la fonction kernel avec suffisamment de points pour avoir une intégrale précise
  + avec les x sélectionnés aléatoirement de manière uniforme dans l’intervalle considéré
* **FApproximation :** Calcul de la fontion de répartition avec

Nous avons considéré deux méthodes, premièrement de manière historique avec les prix passés mais après avoir vu que les prix ne suffisaient pas pour avoi suffisamment de valeurs, nous avons décidé de faire une simulation monte carlo.

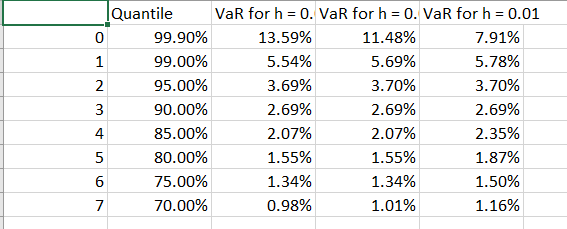
Nous avons ensuite tout enregistré dans des fichiers csv, par exemple :

Nous avons donc 4 fichiers output, deux pour les valeurs des fonctions de réparation en fonction des rendements et deux autre avec les VaR. Nous n’allons présenter que celui avec les VaR.

**Résultat obtenu 1 :**

Pour les prix déjà existants, nous avons pu générer ces VaR, nous pouvons voir que malgré les différences de h, les VaR restent très similaires.

**Résultat obtenu 2 :**

Pour les returns générés aléatoirement, nous avons les résultats suivant :

Les résultats restent assez similaires exceptés au niveau des valeurs « extrêmes » où ce modèle illustre des cas un peu plus « dramatiques » et prédit des pertes plus importantes.

Par conséquent, nous pouvons dire que nous pouvons espérer perdre au maximum 5 à 7% au jour prochain si nous suivons ces données passées, que nous nous basons sur un modèle historique ou un modèle monte carlo.

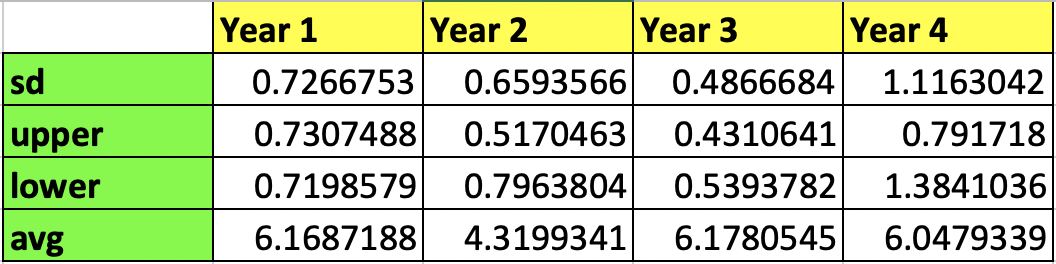
**Question B**

**Fonctions utilisées :**

* **Function average(prices() As Double) As Double :** Average permet de calculer la moyenne des prix
* **Function sd(prices() As Double) As Double :** Sd permet de calculer l’écart-type/volatilité des prix
* **Function uppersd(prices() As Double) As Double** **:** Uppersd permet de calculer la semi-deviation supérieure
* **Function lowersd(prices() As Double) As Double :** Lowersd permet de calculer la semi-deviation inférieure

**Résultat obtenu :**

Dans un premier temps, nous avons stocké les prix dans 4 tableaux correspondant aux 4 années de data. Grace à ces tableaux, nous avons réussi à déterminer les valeurs suivantes :

****

**Interprétation :**

La volatilité est un indicateur cohérant de risque. En année 1 la volatilité était de 72,7%, puis 66% en année 2, puis 48,6% et 111,6% en année 4. Nous pouvons conclure que le stock est plus risqué en année 4, puis de moins en moins risqué en année 1, 2 et 3 respectivement.

La semi-déviation montre la volatilité au-dessus et en-dessous de la moyenne. En année 4, la semi-déviation a atteint 138,4.

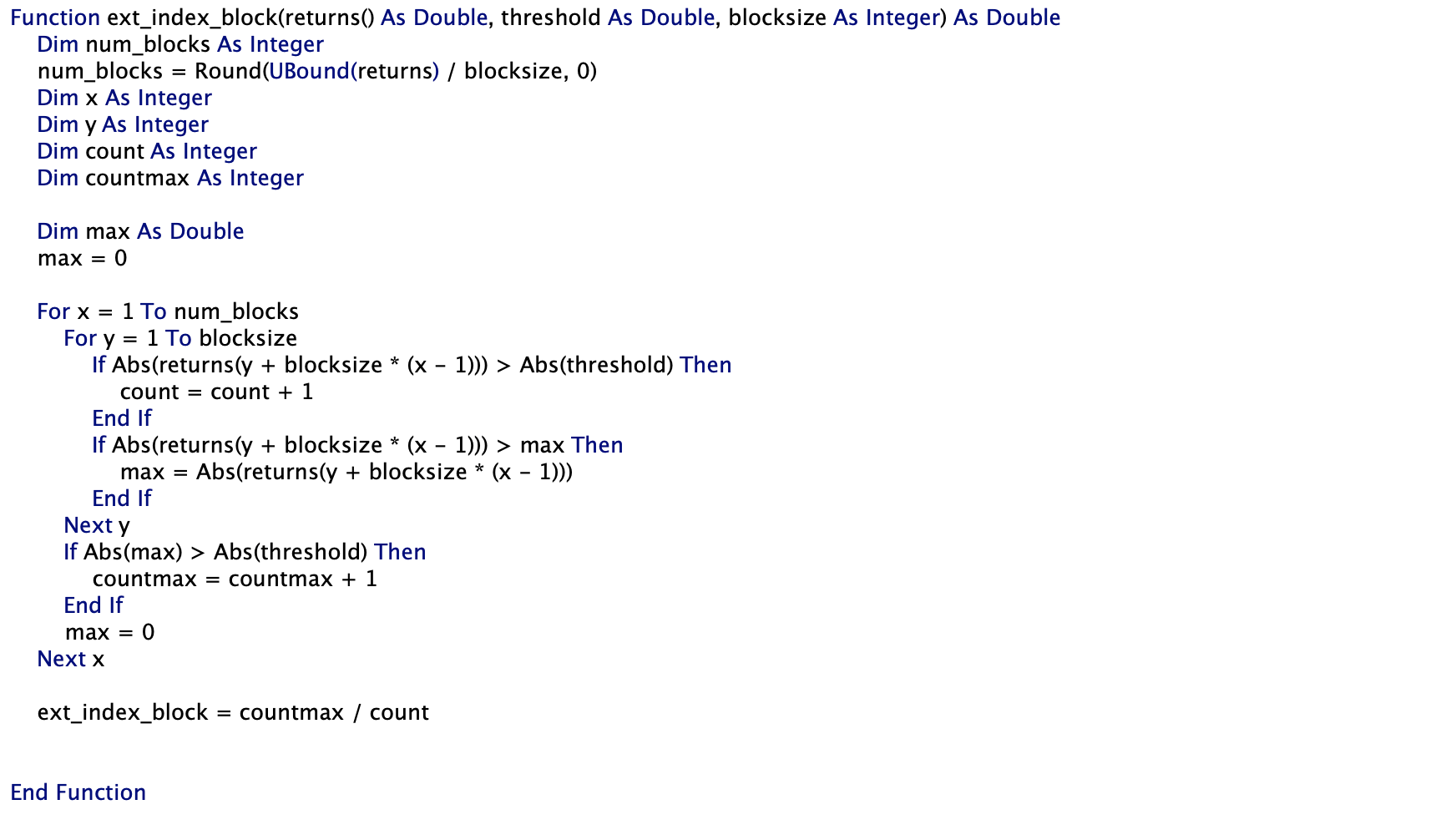
**Question C**

1. **Extremal index**

Nous avons utilisé la formule de block de-clustering pour déterminer le Extremal Index.

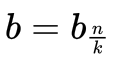
Pour cela, nous avons appliqué la formule dans le cours, et implémenté dans une fonction « **Function ext\_index\_block(returns() As Double, threshold As Double, blocksize As Integer) As Double**».

Cette fonction prend en paramètre le tableau des returns, la valeur seuil, et la taille des blocs. Deux compteurs sont ensuite initialisés, « count » et « countmax ». « Count » incrémente lorsque la valeur de la cellule est au-dessus du seuil. « Countmax » lui, nécessite de trouver le return max du bloc, et incrémente lorsque cette valeur est au-dessus du seuil. La fonction retourne le ratio entre la somme des « count » et la somme des « countmax ».

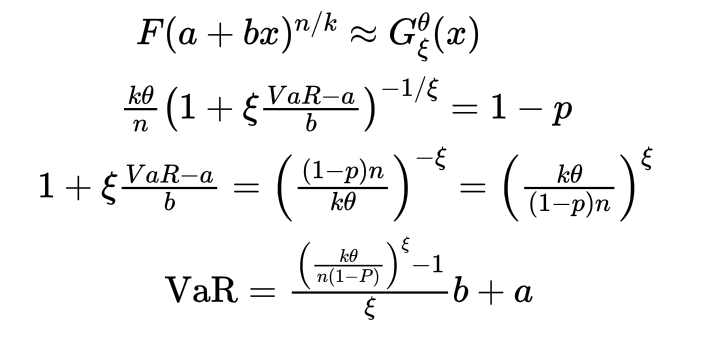


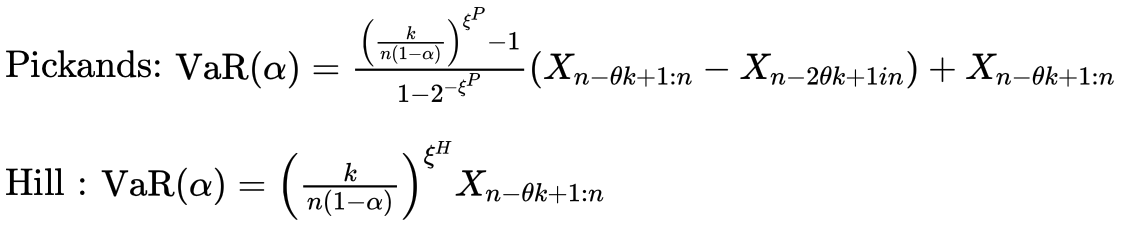
1. **EVT VaR dépendance des return**

/var/folders/m5/rrjlydgd0_v1_g4ffj854vmr0000gn/T/TemporaryItems/(A Document Being Saved By screencaptureui 18)/Screenshot 2020-12-27 at 21.41.30.png

La VaR avec hypothèse d'indépendance est calculée en corrigeant le

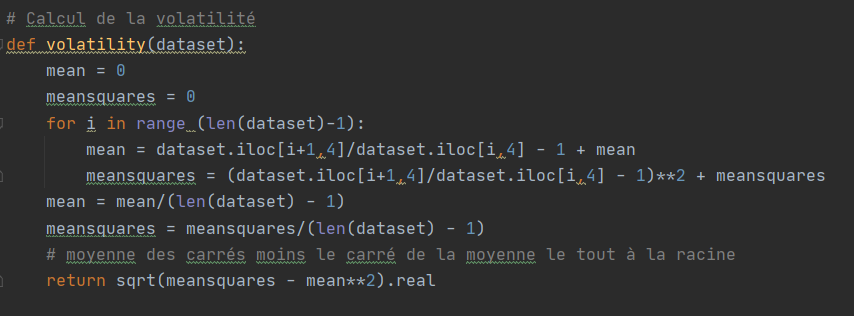
/var/folders/m5/rrjlydgd0_v1_g4ffj854vmr0000gn/T/TemporaryItems/(A Document Being Saved By screencaptureui 19)/Screenshot 2020-12-27 at 21.47.48.pngOn a donc, avec et :



On peut identifier la même formule qu'avec l'hypothèse d'indépendance, avec une correction de k qui devient . Ainsi par identification on peut dire qu'en utilisant les estimateurs du paramètre GEV, on a:

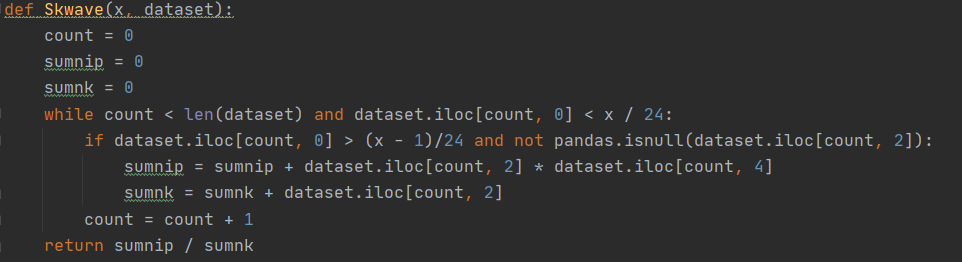
**Question D**

De ce que nous avons compris, cette question consistait à estimer l’espérance du modèle, la variance, mais également sigma, êta, gamma, lambda.

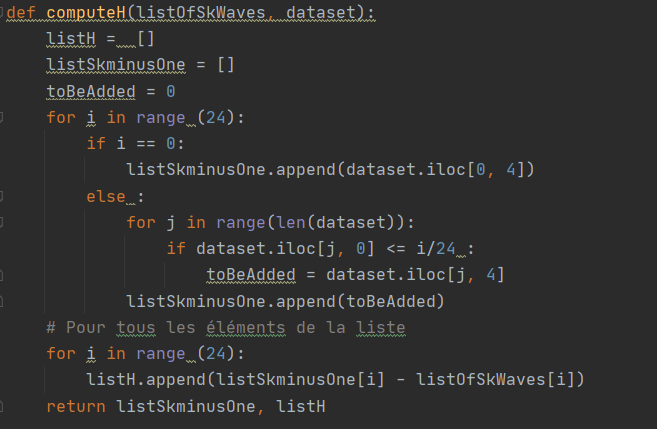
Tout d’abord nous avons commencé par estimer sigma qui est tout simplement la volatilité du modèle. Pour ce faire, nous avons pris en considération le volume échangé à chaque instant t et le signe de l’opération.

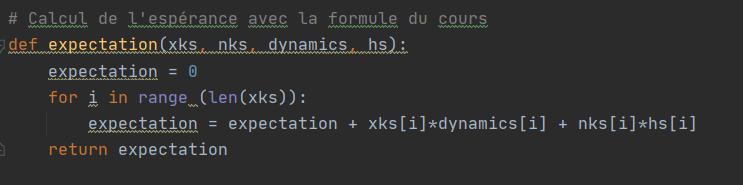
Nous avons également créé les xk qui représentent le volume qui doit être liquidé à t et le volume total nommé sumOfNk.

Pour estimer l’espérance, nous avions eu besoin de la liste des h, des g \* tau, des xk et des nk qui ont été cité ci-dessus. Pour estimer h, nous avons commencé par estimer sk vague qui a pour formule donc on peut calculer le terme tout à droite puis par égalité estimer les h étant donné que représente juste le prix à .

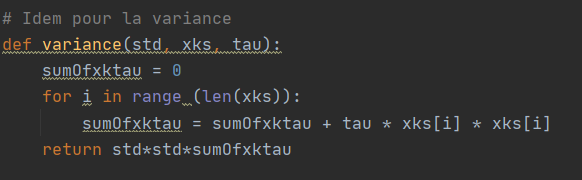


Et :

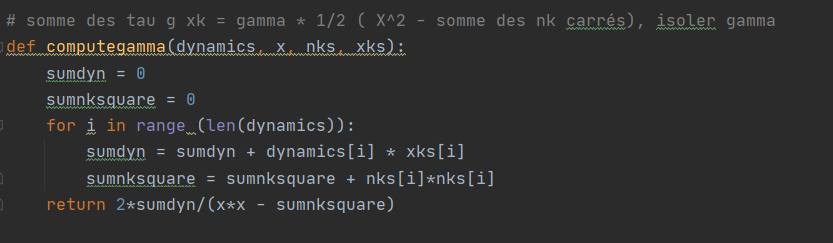


Pour les g, nous utilisons le fait que ayant les prix, nous pouvons estimer g et donc la moyenne aisément en supposant que les pertes ont une moyenne qui est nulle, il nous reste que

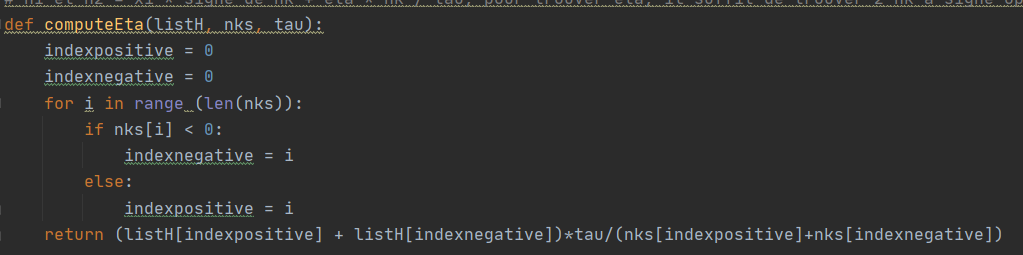
Et donc nous pouvons également estimer la variance :



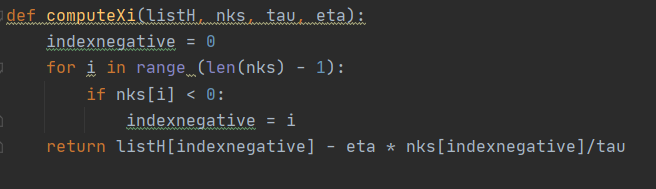
Après cela, par Almgren et Chriss, en supposant que g est linéaire on peut estimer gamma avec la fonction suivante :au



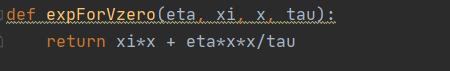
Puis Eta est assez trivial à estimer étant donné que , il nous suffit de prendre deux h tels que les n aient un signe opposés, on élimine alors de l’équation et on a donc :



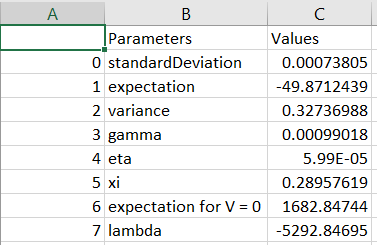
Pour estimer étant donné que nous avons déjà eta, il nous suffit de résoudre une équation du premier degré à une seule inconnue avec h, nk, eta et tau connus :



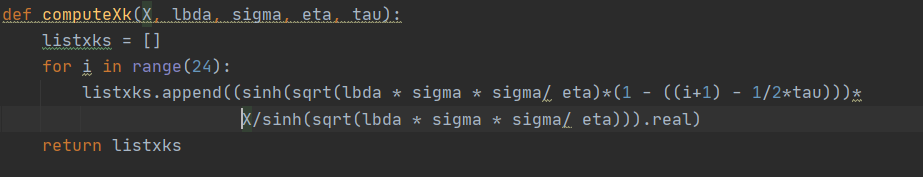
Pour trouver lambda, notre dernier paramètre, nous avons utilisé le fait que nous supposions linéaire, en appliquant la formule pour avec V équivalent à 0 on trouve une espérance aussi nommée « expectedForV0 ».

A partir de là, nous résolvons une équation pour une fonction affine avec et donc :

Et on a donc un fichier output avec tout cela :

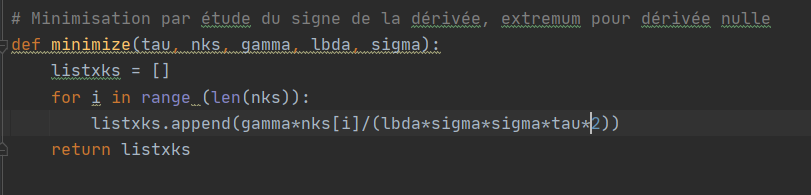
Les valeurs nous paraissent cohérentes car en regardant les opérations, nous avons acheté plus que nous n’avons vendu. Cependant xi étant supposé être la moitié du spread bid-ask, nous avons un doute sur l’exactitude du modèle.

A présent, devons simuler une stratégie. Pour ce faire, nous avons élaboré deux méthodes, la première partant des xk générés par sinus hyperboliques avec la formule du cours, qui représente le juste milieu avec l’aversion au risque.

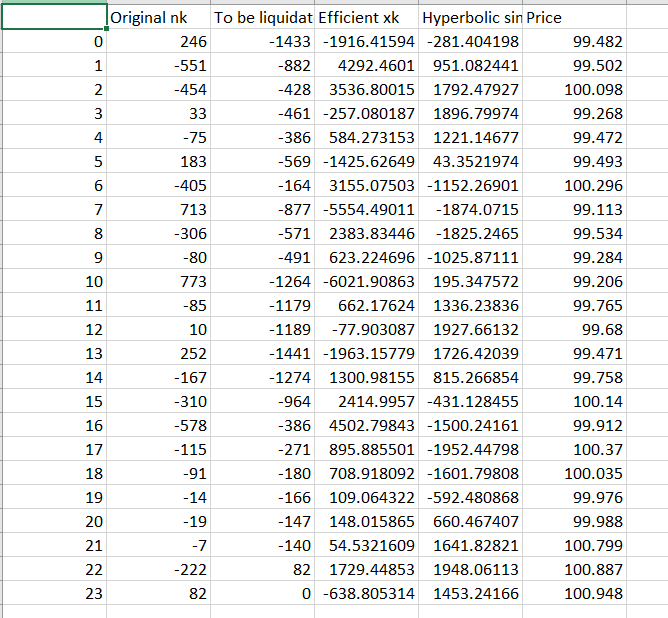


La deuxième méthode que nous avons utilisé est basé sur la minimisation de étant donné lambda connu et

Nous étudions les variations et voulons trouver un extremum, on fait donc la dérivée par rapport aux xk = 0 pour l’extrema.



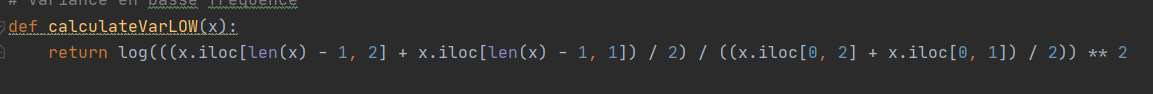
Et on se retrouve avec les output :



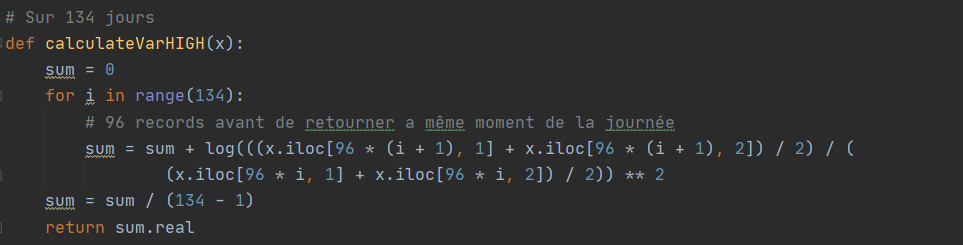
Nous observons que les valeurs estimées par la deuxième méthode, (colonne 3) sont « extrêmes » et difficiles à accomplir, cependant, en regardant la méthodes des xk par sinus hyperbolique, cela nous semble faisable. Donc nous utiliserions plutôt cette méthode.

**Question E**

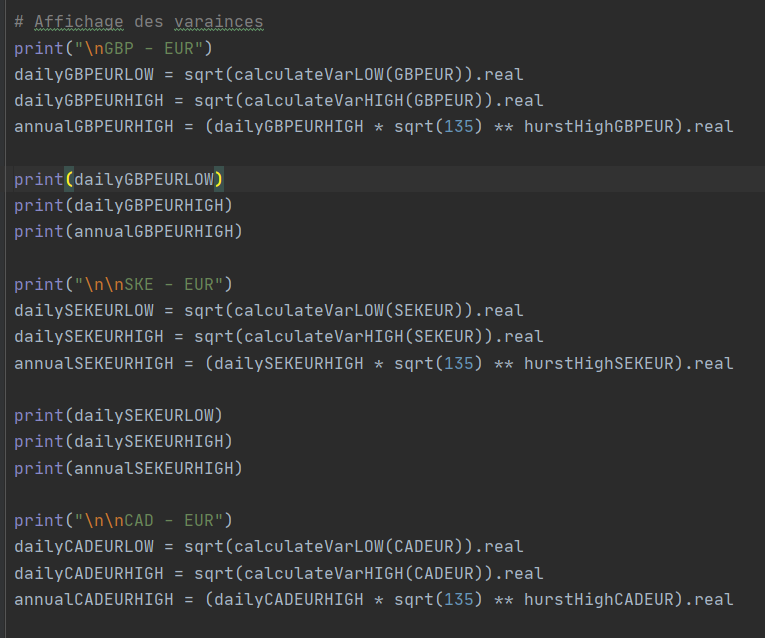
Pour cette question, nous avons utilisé les valeurs initiales et finales pour calculer les variances à basse fréquence :



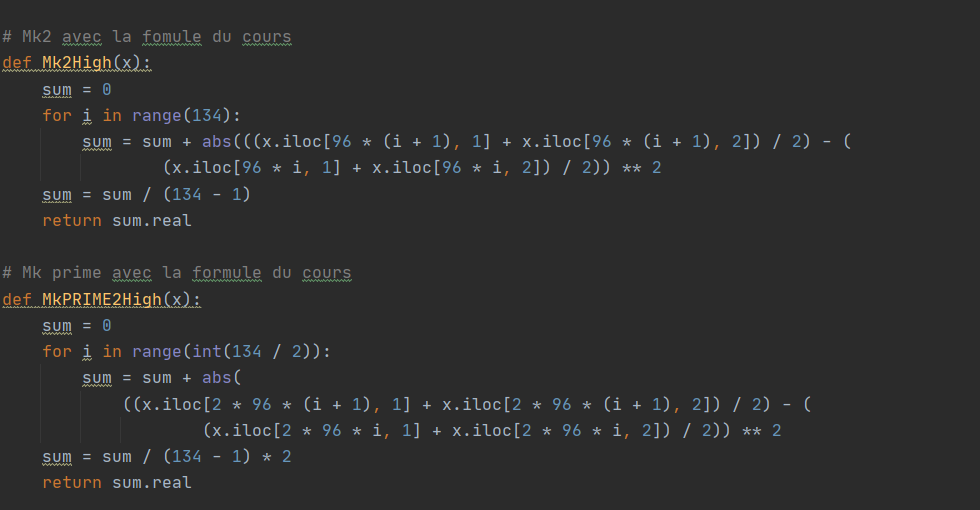
Pour les hautes fréquences, nous utilisons tous les jours (et non pas toutes les 15 min comme indiqué sur le fichier fourni) Nous avons 134 / 135 jours et des 96 fois 15 min par jour sur notre excel de fait on utilise :

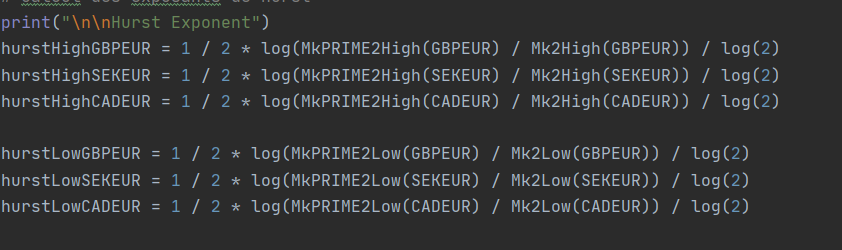


Pour la variance annuelle, on multiplie cette var par 134 étant donné que nous n’avons pas les données sur 252 jours comme l’année de trading le voudrait dans nos données.

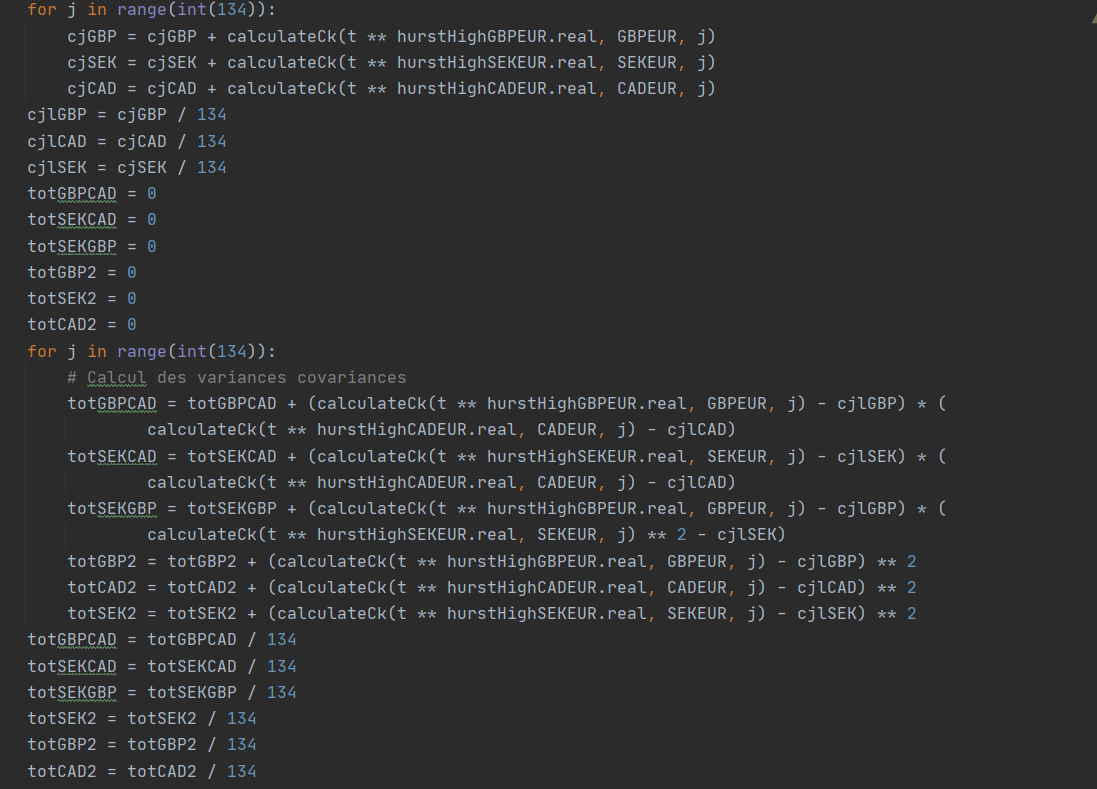


Et pour les exposants de hurst nous avons utilisé

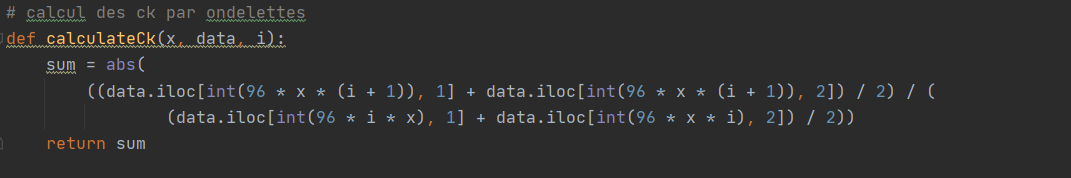




Et pour estimer les variances, nous utilisons les formules suivantes, avec des temps d’investissement générés au hasard 200 fois :



Où ck représente



Tout cela deux cent fois pour pouvoir sélectionner les valeurs pour lesquelles la variance ou covariance est minimale :

