‘’All models are wrong BUT SOME ARE USEFUL’’

Georges BOX

‘’LA MODELISATION EST BIEN PLUS DE L’ART QUE DE LA SCIENCE’’

JAOUAD MADKOUR

Table des matières

**Liste des figures4**

Remerciements5

**Introduction6**

Chapitre I : PRELIMINAIRES 7

I.1 Présentation des outils7

I.1.1 Descriptif du logiciel Python………………………………………………………………………………………….7

I.2.2 Google Colaboratory…………………………………………………………………………………………………….7

I.2 Présentation des données……………………………………………………………………………………………………...7

CHAPITRE II : Analyse de la série temporelle de cours ajustés de MICROSOFT……………………………………9

II.1 Analyse graphique…………………………………………………………………………………………………………………9

II.2 Statistiques descriptives et tests statistiques……………………………………………………………………….15

CHAPITRE III : Analyse des rentabilités de MICROSOFT……………………………………………………………………..18

III.1 Analyse graphique……………………………………………………………………………………………………………...18

III.2 Statistiques descriptives et tests statistiques………………………………………………………………………21

CHAPITRE IV : Modélisation des log-rentabilités à l’aide de la famille ARMA…………………………………….24

IV.1 Présentation de la famille ARMA………………………………………………………………………………………..24

IV.2 Modélisation des log-rentabilités de MICROSOFT……………………………………………………………….24

IV.2.1 Recherche du modèle……………………………………………………………………………………………..25

IV.2.2 Construction du modèle à l’aide du ARMA(1,2)………………………………………………………26

CHAPITRE V-Contraintes de travail……………………………………………………………………………………………………27

CONCLUSION……………………………………………………………………………………………………………………………………28

WEBOGRAPHIE…………………………………………………………………………………………………………………………………29

LISTE DES TABLEAUX ET FIGURES

Tableau 1 : Liste des packages Python exploités

Tableau 2 : Visualisation des données

Tableau 3 : Tableau récapitulatif ressortant le modèle

Figure 1 : Box plot de la série de cours ajustés

Figure 2 : QQ plot de la série de cours ajustés

Figure 3 : Histogramme présentant la distribution des cours ajustés

Figure 4 : Trajectoire de la série temporelle de cours ajustés

Figure 5 : Fonction d’autocorrélation de la série de cours

Figure 6 : Box plot des rentabilités logarithmiques

Figure 7 : Kernel density des rentabilités logarithmique

Figure 8 : Trajectoire des rentabilités logarithmiques

Figure 9 : Fonction d’auto corrélation des rentabilités logarithmiques

Figure 10: Train data vs test data

REMERCIEMENTS

Parce qu’un travail ne peut se parfaire seul, je tiens à remercier dans un premier temps celui qui a pourvu à ma force, ma persévérance et mon intelligence pour l’accomplissement de travail, à savoir le Créateur.

Dans un second temps, je remercie profondément mon professeur d’économétrie, Mr Jaouad MADKOUR qui durant toutes les séances de cours nous a fait promener dans le monde de la statistique et de l’économétrie, avec patience, persévérance et attention et nous a fait découvrir ce qu’il a appelé : ’’l’âme de la statistique’’.

INTRODUCTION

L’analyse des séries temporelles est au cœur de l’étude économétrique. Les analyses portant sur les agrégats économiques tels que le PIB ou le taux d’inflation servent grandement à conduire les politiques économiques à l’échelle macroéconomique ou à orienter les prises de décision des entreprises à l’échelle microéconomique.

Afin de parvenir à faire des prévisions de ces séries temporelles, il est nécessaire de déceler leur ’’comportement’’. C’est dans ce cadre qu’est introduite la notion de modélisation. En économétrie, plusieurs modèles peuvent servir à reproduire la marche d’une série chronologique.

Partant de cela, nous introduirons de ce fait l’objet de notre travail dans la suite de ce propos.

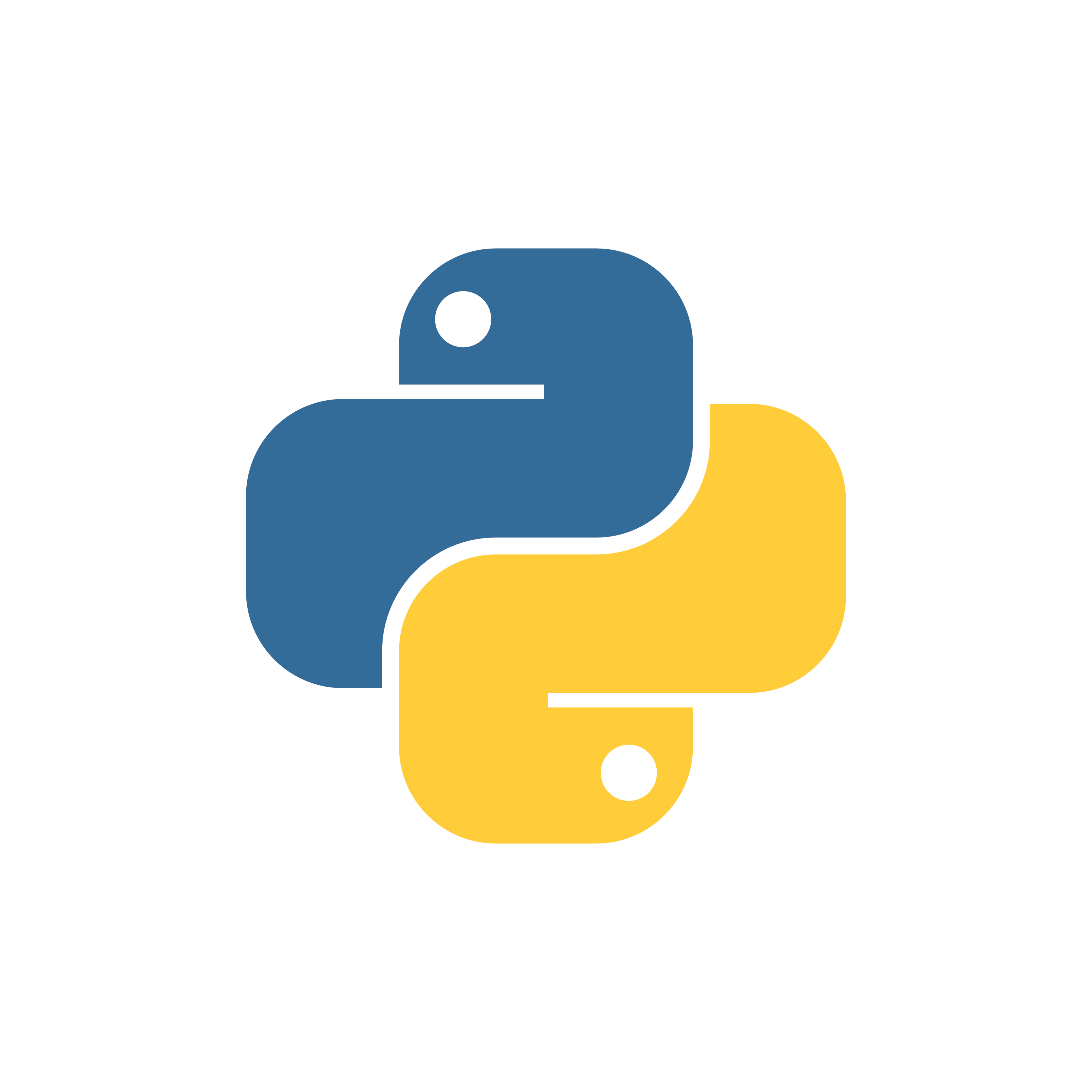
Le cœur de ce projet concerne l’usage de la famille ARMA pour modéliser une série temporelle.

Pour ce faire, nous mènerons une analyse de cours ajustés à la clôture d’une action, nous avons choisi celle de MICROSOFT, sur une période determinée.Puis nous conclurons sur sa non stationnarité, ce qui nous conduira à une transformation de cette série de cours en série de rentabilités, afin de la rendre stationnaire. Après avoir conclu, sur la base de cette seconde analyse, la stationnarité de la série de rentabilités  nous finirons avec la modélisation de cette série en utilisant un modèle approprié de la famille ARMA.

CHAPITRE I : PRELIMINAIRES

Dans cette section, nous introduirons dans un premier temps, les outils qui nous ont servi pour la mise en œuvre de ce projet, puis dans un second temps les données que nous avons utilisées pour parfaire l’analyse qui suivra.

I.1 Présentation des outils

Dans le cadre de ce projet, nous avons principalement utilisé le logiciel de programmation dénommé Python ainsi qu’une extension de Google drive appelée Google Colaboratory.

I.1.1 Descriptif du logiciel Python

Python est un langage de programmation qu’on pourrait qualifier de multitâches, tant utilisé pour le développement d’applications web que pour l’apprentissage automatique (ou Machine Learning), en passant par la science des données ainsi que par l’analyse et la modélisation des séries chronologiques  qui constituent justement l’objet de la présente étude.

Python disposant de plusieurs packages, nous présenterons ceux qui ont été employé tout au long de cette analyse :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Pandas** | **Numpy /Pylab** | **Matplotlib** | **Statsmodels** | **Scipy** | **ARCH** | **Pmdarima** |
| Pandas est une bibliothèque qui sert à l’analyse de données mais également en particulier aux opérations de manipulation de tableaux numériques. | Ces deux packages nous permettent aisément d’effectuer des calculs numériques. | Cette librairie est très utilisée pour visualiser des données en produisant des graphiques de haute qualité. | C’est une bibliothèque d’analyse et de modélisation statistiques. | C’est un package qui dispose de plusieurs fonctions mathématiques et scientifiques utiles pour des tests statistiques également. | Ce package est conçu particulièrement pour l’économétrie financière. | Cette librairie performe aisément dans le cadre de l’analyse de séries temporelles. |

Tableau 1 : Liste des packages Python exploités.

I.2.2 Google Colaboratory

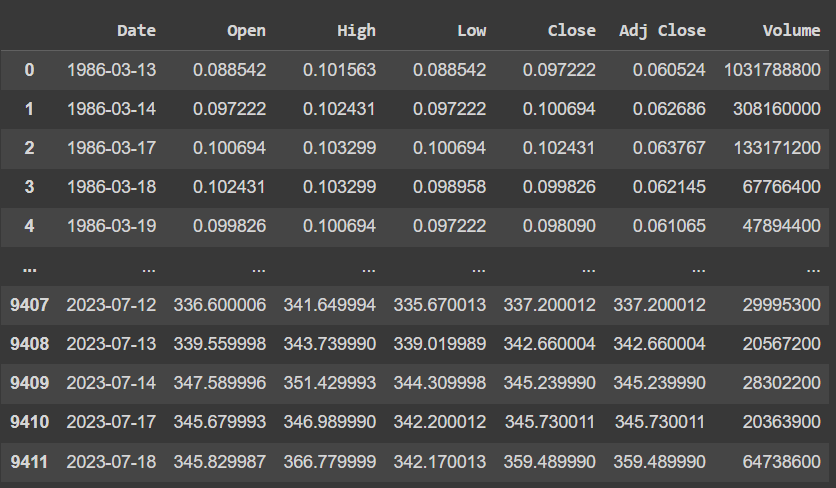
Google colab est une représentation de Jupiter Notebook en ligne, qui agit de ce fait comme un éditeur de texte servant à l’écriture de scripts ou de code source Python, dans notre cas.

I.2 Présentation des données

Les données que nous avons utilisées ont été extraites du site Yahoo Finance, qui dispose d’une large base de données historiques de cours de plusieurs entreprises telles que Tesla, Netflix ou Renault.

Dans notre cas, nous étudierons la série temporelle de cours ajustés à la clôture de l’entreprise Microsoft. Nous avons choisi la période maximale s’étalant du 13/03/1986 au 18/07/ 2023.

Visuellement, la série de données se présente comme suit :

Tableau 2 : Visualisation des données

Nous pouvons constater qu’il y a plusieurs colonnes sur le tableau précédent présentant les différentes informations relations relatives aux cours ; toutefois la colonne qui nous sera fortement utile est celle qui correspond aux cours ajustes à la clôture, se référant dans le tableau à la colonne nommée ‘’Adj Close’’.

Il est nécessaire de préciser que les données étant nombreuses , nous avons tenu simplement à avoir un bref aperçu de ces dernières par le tableau présenté, les premières lignes ainsi que les dernières.

Toutes les données ne sont par conséquent pas présentées dans le dit tableau, cela n’affectera aucunement les analyses que nous opèrerons.

Après cette section de préliminaires, qui a amorcé les différents outils de travail et les données utilisées, nous entamerons dans la suite l’analyse des cours ajustés à la clôture de la série MICROSOFT.

CHAPITRE II : Analyse de la série temporelle de cours ajustés de MICROSOFT

Dans cette section nous mènerons une analyse exploratoire de la série temporelle de cours ajustés à la clôture de MICROSOFT. Cette dernière sera scindée en deux parties distinctes ; à savoir; d’une part une série d’observations graphiques sur la base desquelles nous émettrons des hypothèses ; et d’autre part des statistiques descriptives et des tests permettant d’appuyer, confirmer ou d’infirmer les hypothèses énoncées.

La finalité de cette première analyse est de parvenir à ressortir la présence ou l’absence de valeurs manquantes[[1]](#footnote-1) et/ou aberrantes[[2]](#footnote-2) mais également de savoir si la distribution des données est normale[[3]](#footnote-3) et stationnaire.

II.1 Analyse graphique

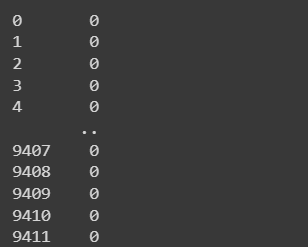
Dans le cadre de cette analyse, nous userons particulièrement des graphiques tels que des box-plots, des histogrammes, des QQ[[4]](#footnote-4) plot et des fonctions d’autocorrélation (AutoCorrelogramFunction ou ACF).

Dans un premier temps, il nous est nécessaire de performer une étape cruciale préalable à toute analyse de données, celle du prétraitement. Nous vérifierons de ce fait, dans un premier temps, s’il y a des valeurs manquantes dans notre série de données.

Pour ce faire, le logiciel Python nous permet cette action par le biais de la fonction ‘’isnull’’. Cette dernière nous retourne ‘1’ si la donnée manque ou ‘0’ dans le cas contraire.

Nous pouvons le voir au travers du code suivant :



Nous avons observé l’absence de valeurs manquantes dans les données par le résultat

Dans un second temps, nous devons vérifier la présence de données aberrantes qui peuvent constituer des anomalies entravant la poursuite de notre analyse. Si elles s’avèrent présentes, il faudrait les traiter avec des méthodes propres aux séries temporelles.

Pour ce faire ; nous avons utilisé une figure nommée box plot ou encore boite à moustaches.

Visuellement, dans le contexte de notre série temporelle, elle se présente ainsi :

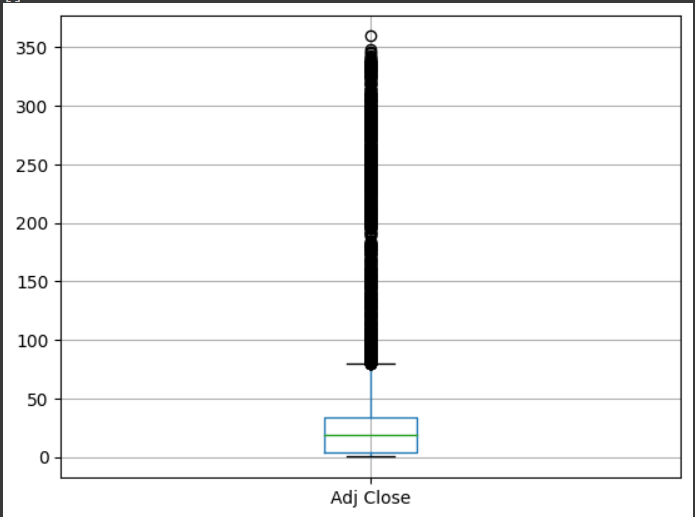
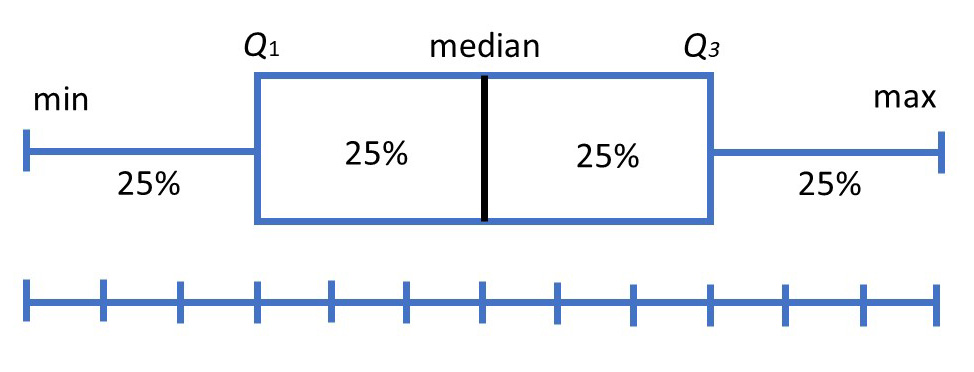


Figure 1 : Box plot de la série de cours ajustés.

Pour pouvoir au mieux l’expliquer  nous allons introduire un second graphique :

Sur le graphique qui précède, nous avons la médiane représentée comme une ligne au centre du rectangle ; elle divise la distribution en deux parties égales, des valeurs autant inferieures que supérieures. Sur les extrémités, nous avons des quartiles, précisément le Q1 (valeur au-dessous de laquelle se trouvent 25 % des données de la distribution) et le Q3 (valeur au-dessous de laquelle se trouvent 75% des données de la distribution).Dans les extrémités ; nous avons les valeurs maximale et minimale de notre distribution. Les valeurs se situant au-delà du min et du max de la distribution peuvent être considérées ; dans certains cas comme des valeurs aberrantes.

Ainsi, dans notre cas, nous pouvons supposer, sur la base de cette boite à moustaches et du résultat produit par le logiciel, qu’il y a une forte présence de valeurs extrêmes positives dans notre série de cours. Autrement dit, il y a de nombreux cours dont les valeurs s’éloignent significativement mais positivement des valeurs des autres cours.

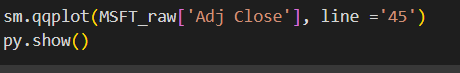
Pour pouvoir avoir la figure de la boite à moustaches sur Python, nous avons utilisé la librairie matplotlib, en entrant le code suivant :



MSFT\_raw étant le nom de notre ensemble de données ou dataset.

Par la suite ; nous allons vérifier la normalité de la distribution des données en utilisant un graphique appelé diagramme quantile-quantile.

Afin de pouvoir expliquer le principe  nous allons introduire le code source générant le graphique et le graphique lui-même.



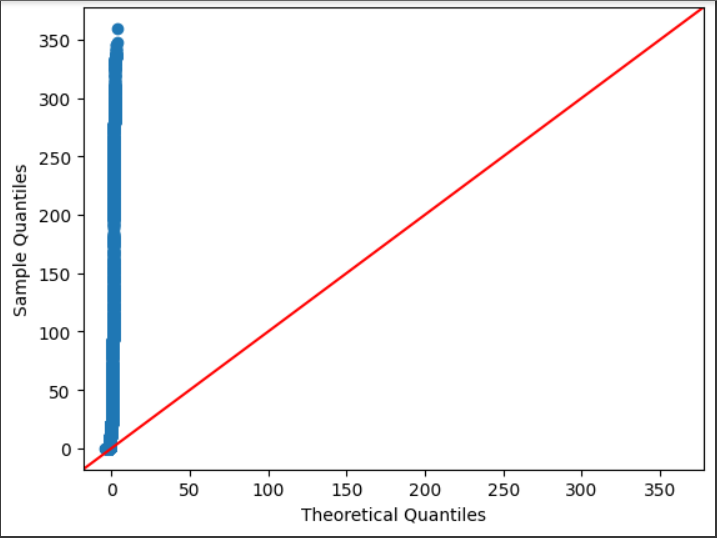
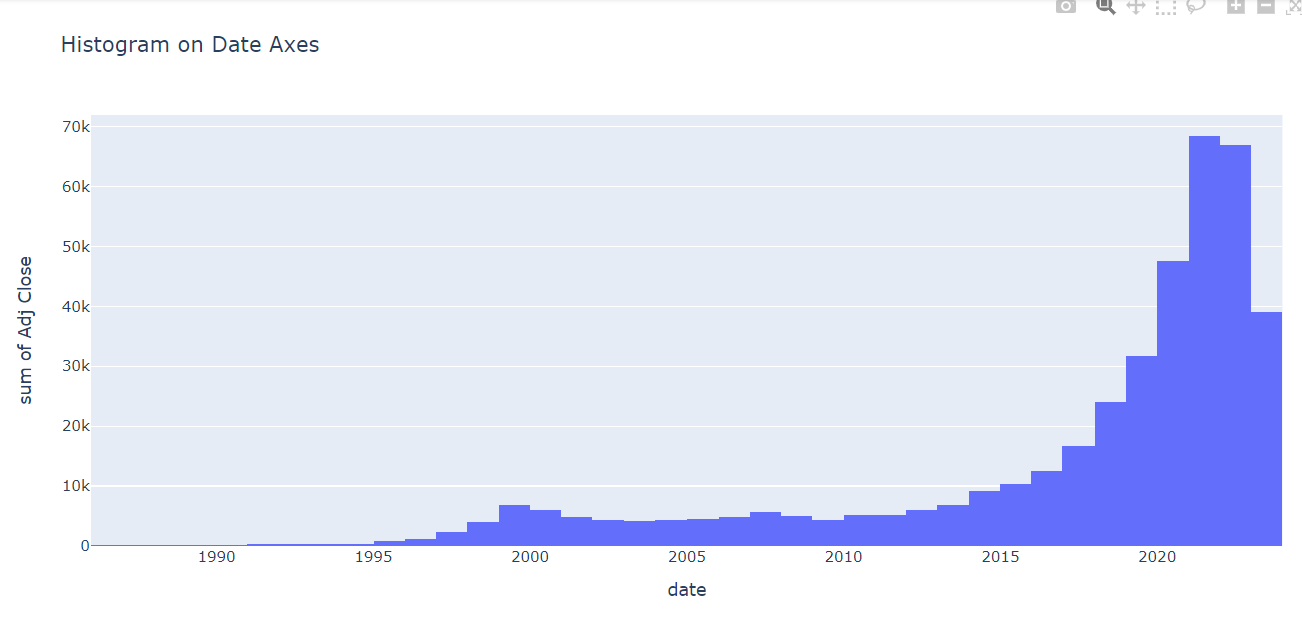
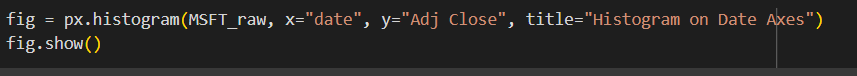


Figure 2 : QQ plot de la série de cours ajustés.

Le principe de ce graphique est assez simple. En effet, il présente deux distributions distinctes : une distribution de notre série de données en ordonnée et une distribution théorique en abscisse. Cette dernière est en général une distribution normale. De ce fait, si notre distribution est également gaussienne, les points représentant notre distribution(en bleu) devraient s’aligner sur la droite présentée dans le plan (en rouge).

Cependant, au regard de notre graphique, les points ne s’ajustent pas à la droite rouge, autrement dit notre distribution n’est pas la même que la distribution théorique qui elle est supposée normale. Par conséquent, nous pouvons donc émettre, sur la base du principe du digramme quantile-quantile que la distribution de notre série de cours ajustés n’est pas normale.

En outre, nous pouvons également nous servir d’un histogramme pour voir la distribution des données et en déduire sous observation sur la normalité des données.

Figure 3 : Histogramme présentant la distribution des cours ajustés.

Une distribution normale est très souvent représentée par une courbe en cloche, pourtant ce n’est point le cas ici. On remarque plutôt une queue plus longue sur l’extrême gauche, se rapportant à une asymétrie à gauche, ce qui nous permet d’appuyer la supposition énoncée précédemment ; notamment celle de la présence de valeurs extrêmes dans la série de données. Cette asymétrie pourrait nous permettre de supposer que les données ne suivent pas une loi normale.

Par la suite et dernièrement, nous allons vérifier la stationnarité de la série temporelle de cours ajustés au travers dans un premier temps d’un simple graphique en ligne, puis dans un second temps du tracé de sa fonction d’autocorrélation[[5]](#footnote-5).

Le graphique en ligne nous permet d’avoir une idée simple de la ‘’marche’’ des données. Ceci nous permettra de découler sur la nature de notre distribution (stationnaire ou pas).Il est important de rappeler qu’une série temporelle est dite stationnaire si, visuellement ses données fluctuent autour d’une même moyenne et que sa variance est stable. On suppose donc que la série ne suit pas une marche aléatoire.

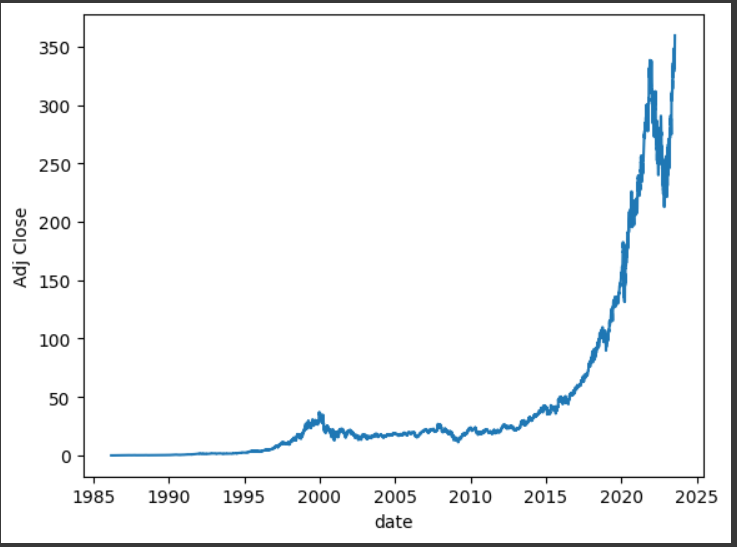
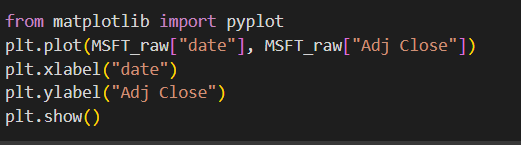
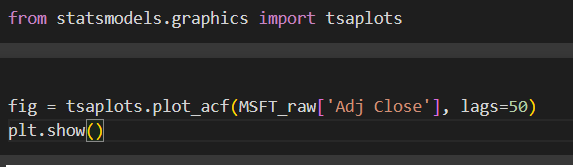


Figure 4 : Trajectoire de la série temporelle de cours ajustés.

Nous pouvons bel et bien observer que la série suit une marche aléatoire, elle ne peut par conséquent pas être considérée comme stationnaire.



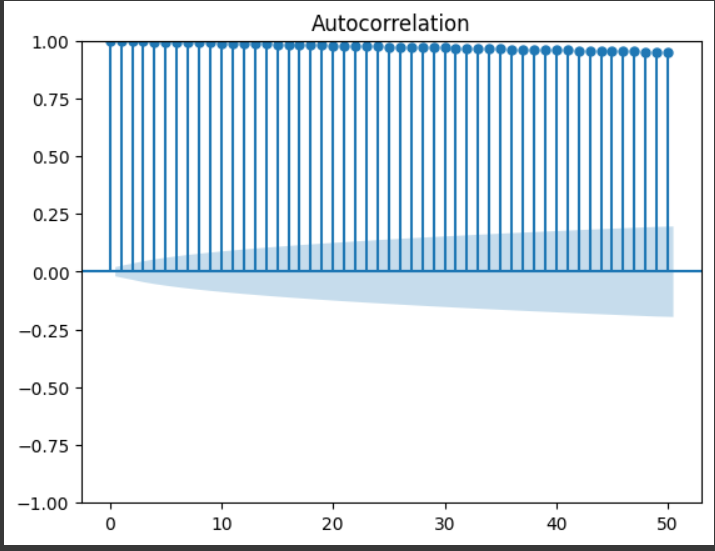
Le graphique qui suit présente la fonction d autocorrélation de la série ; cette dernière en cas de stationnarité de la série est censée chuter de manière exponentielle partant de sa première valeur pour se retrouver à zéro. Ceci s’explique par le fait que sur la base d’un théorème énoncé par l’économiste et statisticien Herman WOLD, selon lequel ‘’tout processus stochastique[[6]](#footnote-6) stationnaire peut s’écrire sous forme d’une somme pondérée de bruits blancs présents et passés ; plus un processus déterministe parfaitement prévisible’’, nous parvenions au fait que les bruits blancs de notre processus ont une importance décroissante à mesure qu’évolue le temps. La mémoire du processus représentée par la fonction d’autocorrélation baisse donc de manière significative jusqu’ à ce que les chocs n’aient plus d’importance (ou soient nulles dans le cadre de la figure). Ces chocs sont dits temporaires ou transitoires.

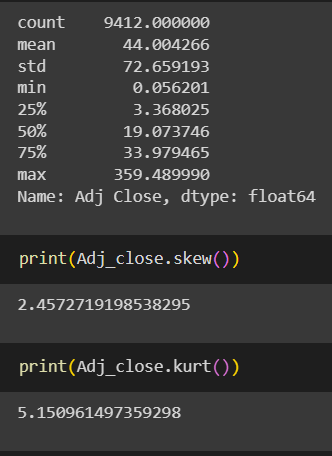
Figure 5 : Fonction d’autocorrélation de la série de cours.

Nous observons bel et bien que la fonction ne chute pas de manière exponentielle vers zéro, se ramenant à la supposition que les chocs gardent une importance significative bien qu’inferieurs à ceux qui les précèdent. Notre série n’est par conséquent pas stationnaire, sur la base de cette observation.

Pour parvenir à donner un caractère vérace à nos hypothèses, nous utiliserons des statistiques descriptives et des tests statistiques, ce qui fera l’objet de la deuxième partie de notre analyse exploratoire.

II.2 Statistiques descriptives et tests statistiques

Nous nous servirons dans ce cas des tests de jarque-bera et aussi de Kolmogorov-Smirnov pour la normalité et celui de Dickey-Fuller pour la stationnarité.



Les fonctions skew et kurt renvoient les valeurs des coefficients de skewness (coefficient d’asymétrie) et kurtosis (coefficient d’aplatissement).

Il est nécessaire de rappeler qu’une distribution normale a les propriétés suivantes :

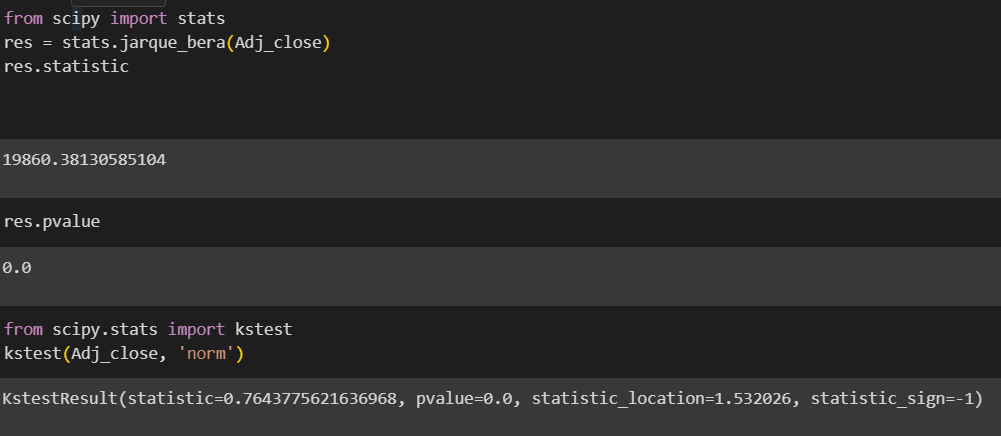
Skewness=0 se référant à une parfaite symétrie de la distribution(les valeurs présentent à gauche de la distribution sont autant que celles présentes à droite de la distribution)

Kurtosis=3, la distribution n’est ni très arrondie ni très pointue, les queues ne sont donc ni trop épaisses ni trop resserrées par rapport à la moyenne de la distribution.

Dans notre cas, le skewness=2,46 par arrondi, ce qui signifie que notre distribution est skewed sur la droite ; ce qui signifie que les cours supérieurs à la moyenne(44) sur toute la période d’étude  sont plus fréquents que ceux inferieurs a la moyenne.

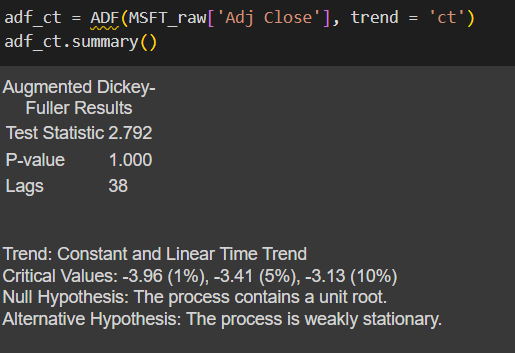
Parallement le kurtosis est positif et est égal à 5,15 ; ce qui signifie que la distribution n’admet pas de queues très épaisses mais elle demeure toutefois leptokurtique ; la courbe da la distribution est moins aplatie que la normale.

Nous pouvons également nous servir des tests susmentionnés, en observant la valeur de la p-value associée à la t-statistique, si cette dernière sous un seuil précis de 5% à la valeur de ce seuil, nous rejetterons l’hypothèse nulle qui est celle de la normalité et nous accepterons l’hypothèse alternative qui est celle de la non normalité de la distribution.



Les résultats des deux tests pris ensemble traduisent une non normalité de notre distribution de cours ajustés à la clôture due au fait que les p-value associées aux valeurs des t-statistic fournis par les deux tests sont toutes les deux inferieures à la valeur du seuil de significativité (5 %).

En dernier lieu ; nous allons confirmer l’hypothèse de non stationnarité émise plus haut par le test ADF (Augmented Dickey Fuller).En admettant le même seuil de significativité, l’hypothèse nulle de ce test est la présence d’une racine unitaire[[7]](#footnote-7) (correspondant à une non stationnarité) tandis que l’hypothèse alternative suppose l’option contraire.



Nous constatons que la p-value associée à la valeur de la t statistique est supérieure au seuil de significativités , nous ne pouvons donc pas rejeter l’hypothèse nulle ; le processus contient donc une racine unitaire et est par conséquent non stationnaire.

Afin de rendre la série stationnaire, nous allons employer une méthode fréquemment utilisée dans le cas des séries temporelles qui est celle de la différenciation.

Nous allons donc transformer notre série de cours en série de rentabilités et mener également la même analyse exploratoire.

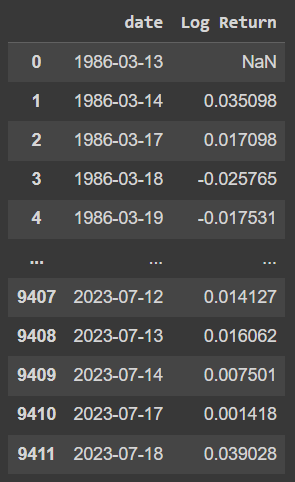
CHAPITRE III : Analyse des rentabilités de MICROSOFT

Il est primordial de préciser que le type de rentabilités que nous allons utiliser est les log-rentabilités.

Nous allons parfaire la même analyse que celle des cours et appuyer nos hypothèses par des statistiques descriptives et des tests d’hypothèses.

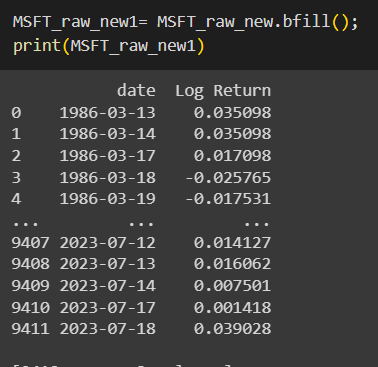
III.1 Analyse graphique

Nous allons dans un premier temps vérifier la présence de valeurs manquantes (à l’aide du même code source).



Au regard de ce résultat, nous avons une valeur manquante dans notre série de données. Pour traiter cette donnée manquante, nous allons utiliser une méthode appropriée pour une série temporelle dite ‘’backfill’’ qui remplace la donnée manquante par celle qui la précède.

Ainsi, nous obtenons le résultat suivant :



Dans un second temps, nous allons visualiser un box plot pour vérifier la présence de valeurs extrêmes.

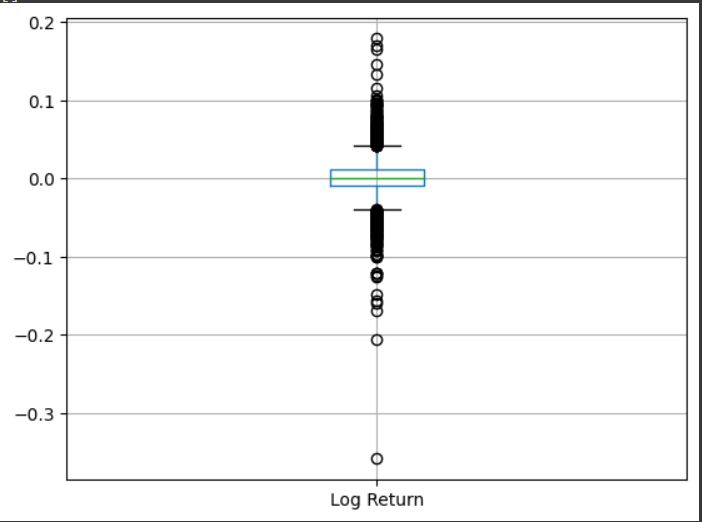
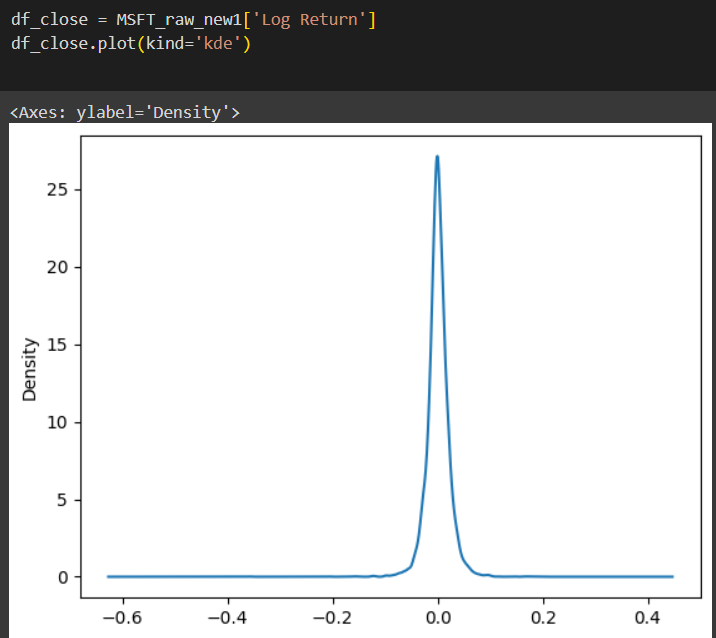


Figure 6 : Box plot des rentabilités logarithmiques.

Nous pouvons observer des valeurs extrêmes non seulement positives mais également négatives.

Apres cela, nous vérifierons la normalité de la distribution à l’aide d’un graphique nommé ’’kernel density ’’.L’objectif est de pouvoir faire une comparaison entre la représentation graphique d’une distribution normale et celle de notre distribution.

Figure 7 : Kernel density des rentabilités logarithmique

Nous pouvons remarquer que la forme de la forme de la courbe est plus pointue , ce qui pourrait supposer un kurtosis supérieure à 0(pour une distribution normale) et donc qualifier notre distribution de leptokutique.

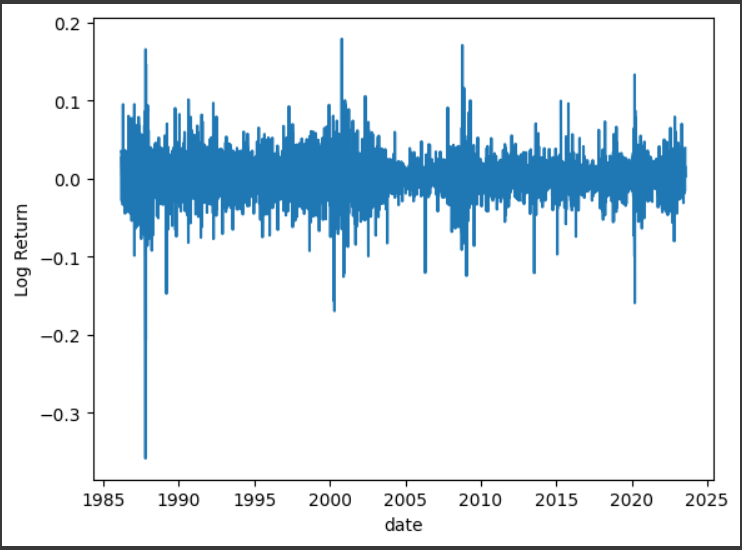
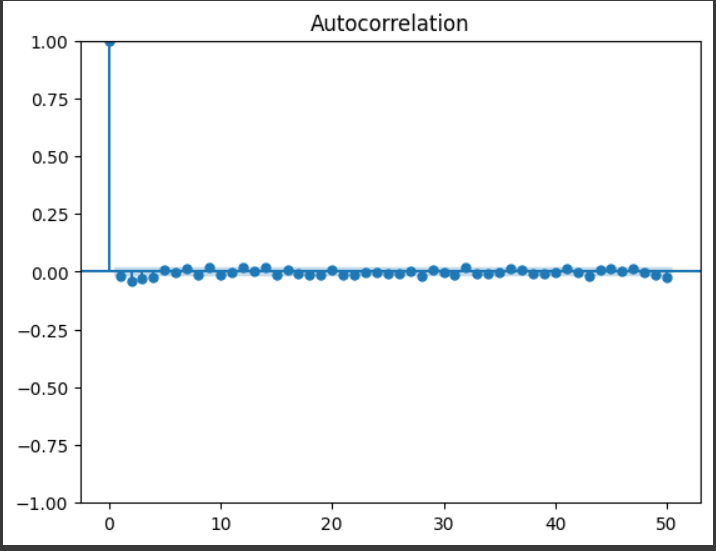
En dernier lieu, nous allons vérifier la stationnarité du processus par le tracé de sa trajectoire et sa fonction d’autocorrélation.

Figure 8 : Trajectoire des rentabilités logarithmiques

Nous observons que la trajectoire fluctue autour de sa moyenne (égale à 0) et dans un couloir assez précis, avec toutefois quelques clusters de volatilité[[8]](#footnote-8). Ce qui peut nous laisser supposer que la série est stationnaire.

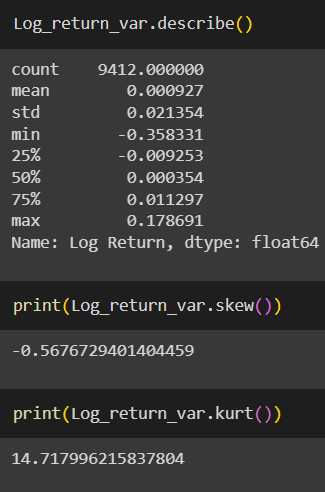
Figure 9 : Fonction d’auto corrélation des rentabilités logarithmiques

Au regard de cette chute exponentielle de la fonction d’auto corrélation ; nous pouvons déduire que la série est stationnaire.

Nous allons maintenant confirmer ou infirmer ces hypothèses par des statistiques descriptives et des tests d’hypothèses.

III.2 Statistiques descriptives et tests statistiques

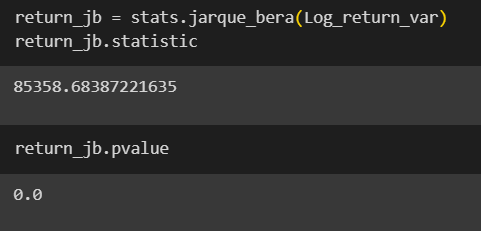
Nous nous servirons dans ce cas du test de jarque-bera pour la normalité et celui de Dickey-Fuller pour la stationnarité.



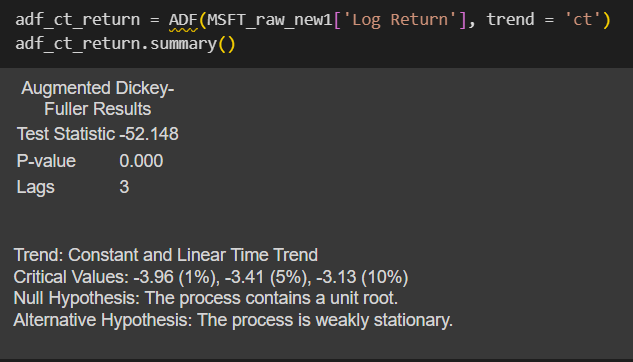
Avec un skewness négatif égal à – 0,56 ; ce qui traduit une asymétrie à gauche de notre distribution ; ce qui signifie que les rentabilités logarithmiques supérieures à la moyenne(0,000927) sont moins fréquents que ceux qui inferieures à cette dernière.

Nous constatons entre autres que la valeur du kurtosis est largement supérieure à 3 ; ce qui signifie que les queues de la distribution sont épaisses et la distribution est leptokurtique.

Nous pouvons également utiliser le test de jarque-bera.



Nous remarquons que la p-value associée à la t-statistique fourni par le processus est nulle ; on ne peut donc pas confirmer la normalité de la distribution.



Nous terminerons par le test de Dickey Fuller pour la stationnarité.

Nous constatons que la p-value associée à la t-statistique fourni par le processus est nulle ; ce qui nous conduit à rejeter l’hypothèse nulle de la présence d’une racine unitaire et d’accepter l’hypothèse alternative de la stationnarité de notre série.

La condition de stationnarité étant vérifiée , nous pourrons de ce fait user d’un des modèles de la famille ARMA .

CHAPITRE IV : Modélisation des log-rentabilités à l’aide de la famille ARMA[[9]](#footnote-9)

Cette section est le cœur de notre projet, elle fait ressortir l’importance de la stationnarité en ce sens que cette dernière constitue une condition indispensable à l’usage de la famille ARMA. Ainsi, nous présenterons dans un premier temps le modèle ARMA, puis nous déboucherons sur la modélisation des log-rentabilités de MICROSOFT.

IV.1 Présentation de la famille ARMA

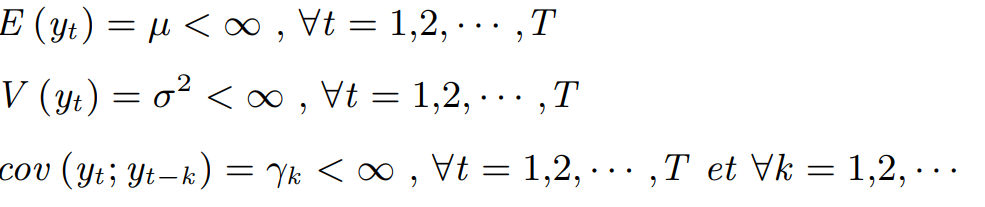
La famille ARMA est beaucoup utilisée en statistique pour la modélisation de séries chronologiques. Elle se décompose en deux modèles distincts : le modèle AR(AutoRegressive) qui permet de régresser le processus sur ce dernier  ainsi que le modèle MA (Mooving Average) qui lui permet de régresser sur les chocs du processus. Nous pouvons obtenir de ce fait le modèle suivant :

|  |
| --- |
| *yt* = *θ* + *α*1*yt-*1 + *β*1 ϵ*t-*1 + ϵ*t* |

Nous avons ici une mise en relations des valeurs retardées du processus ainsi que de ses chocs dans le cas d’un ARMA (1,1). En outre se rajoutent au modèle une constante (valeur à laquelle sera réduit le processus si les paramètres du modèle sont nuls) et un choc (qui constitue la partie non prévisible du modèle).

Seulement, l’utilisation de ces modèles joints ensemble suppose la stationnarité de la série temporelle. Ce qui nous conduit à mieux expliquer cette notion.

On peut qualifier un processus stochastique de stationnaire au sens faible, si ce dernier répond aux exigences suivantes :

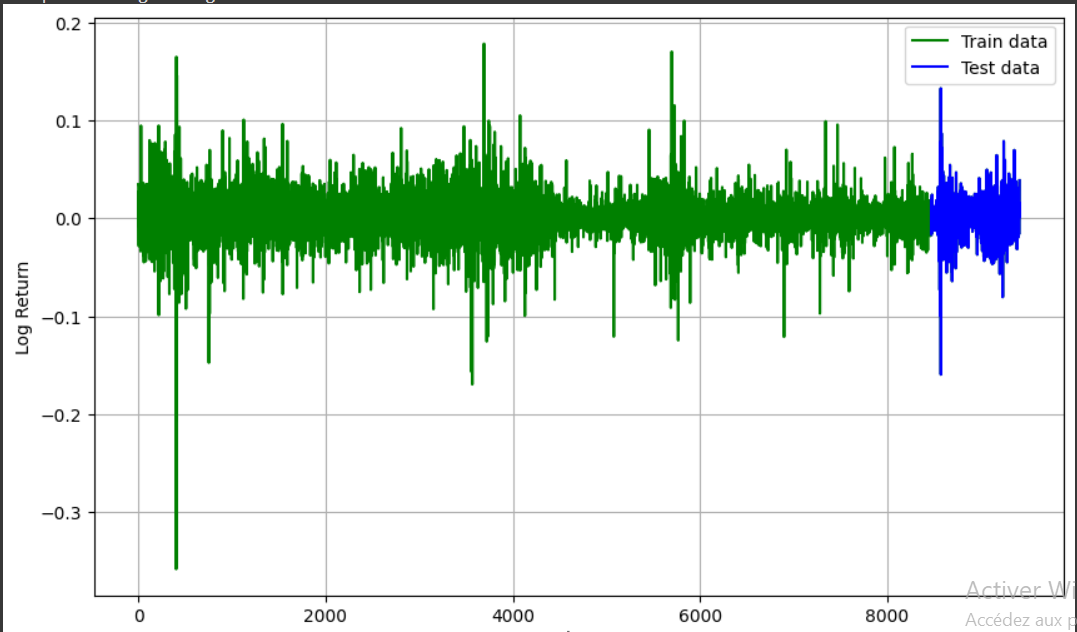


IV.2 Modélisation des log-rentabilités de MICROSOFT

Cette partie s’axe autour de deux actions principales : la première concerne la recherche du modèle adéquat de la famille ARMA permettant de modéliser notre distribution ; la seconde constitue l’application du modèle trouvé à nos données afin de ressortir le modèle pouvant servir à des fins de prévisions.

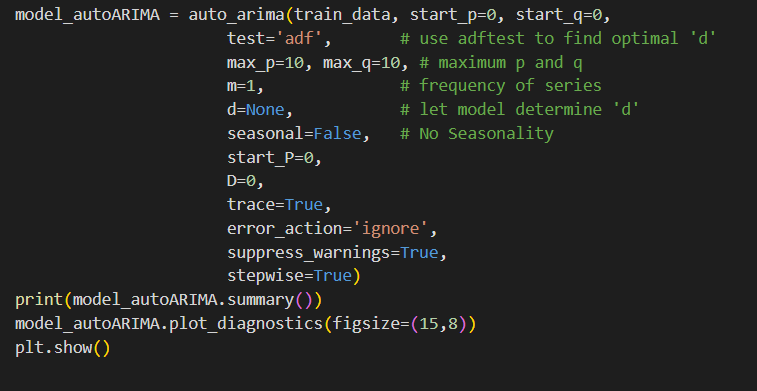
Pour parfaire les deux actions énoncées, nous partitionnerons nos données en deux : une partie qui se constituera de données d’entrainement (90%) et une autre dédiée aux données de test[[10]](#footnote-10) (10%).

Visuellement, nous avons ceci :

Figure 10: Train data vs test data

IV.2.1 Recherche du modèle

Afin de parfaire cette tâche, nous avons utilisé les packages statsmodels et pmdarima.

 Le code qui précède nous a permis de trouver le meilleur modèle de la famille ARMA permettant de modéliser nos données, sur la base d’un critère d’information, qui dans notre cas est le AIC[[11]](#footnote-11). Ayant dans un premier temps différencier la série, nous aurons un ARIMA (p, 0, q) qui correspond à ARMA (p,q).En outre ; nous avons indiqué le maximum d’ordre pour p(ordre de la partie AR) et q(ordre de la partie MA).Nous avons fait abstraction de la présence d’une quelconque saisonnalité dans notre série . Et, en finalité nous avons opté pour la méthode stepwise pour le choix du modèle. Cette dernière est dite progressive, car elle essaie un modèle et si ce dernier ne s’avère pas l’idéal, elle l’élimine et en essaie un autre jusqu’ à trouver le modèle optimal.

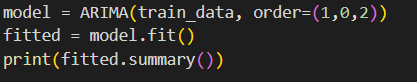
Le résultat est le suivant :



Au regard du résultat ; il s’avère que le modèle adéquat de la famille ARMA permettant de modéliser nos données est le ARMA(1,2).

IV.2.2 Construction du modèle à l’aide du ARMA(1,2)

Nous allons donc appliquer notre modèle aux données d’entrainement.



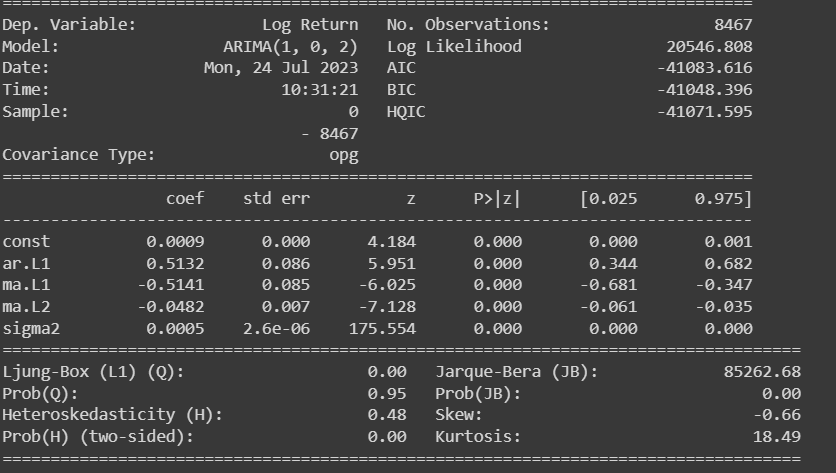


Tableau 3 : Tableau récapitulatif ressortant le modèle.

Ainsi, sur la base du AIC, le modèle présentant les valeurs des paramètres qui nous étaient jusqu’ ici inconnus est le suivant :

yt = 0,0009 + 0,5132Y(t-1) – 0,5141ϵ(t-1)-0,0482ϵ(t-2) +ϵ(t).

CHAPITRE V-Contraintes de travail

Dans le cadre de ce projet, nous avons été confrontés particulièrement à la difficulté pour interpréter certains graphiques tels que des box plots ou encore des plots se référant à nos analyses. En outre, nous avons découvert de profondes lacunes concernant l’interprétation de statistiques descriptives découlant de la distribution de nos données.

En dépit de ces difficultés, nous avons appris énormément concernant l’analyse de séries temporelles et ses implications.

CONCLUSION

En définitive, la finalité de notre travail étant de modéliser une série temporelle avec un modèle adéquat de la famille ARMA, il en va de rappeler les étapes par lesquelles nous sommes passées avant que d’aborder les conclusions relatives à cette modélisation.

Dans un premier temps, nous avons analysé une série temporelle de cours ajustés à la clôture de l’action MICROSOFT, obtenue via le site Yahoo finance. Nous avons mené une analyse exploratoire de cette dernière à l’issue de laquelle nous avons conclu qu’elle n’était pas stationnaire.

Compte tenu de cela, nous avons employé une méthode de différenciation pour la rendre stationnaire, nous avons donc transformé les cours en rentabilités, et plus précisément en log rentabilités. Ayant performé également une analyse exploratoire de ces log-rentabilités, nous avons pu déboucher sur la stationnarité de cette série et entamé la modélisation avec ARMA.

Cette dernière étape s’est effectuée en deux sections différentes : d’une part la recherche du modèle ARMA optimal, et d’autre par l’application de ce modèle ARMA à nos données afin de trouver les paramètres inconnus de notre modèle de log-rentabilités.

En définitive, le modèle adéquat permettant de modéliser notre série de données était le ARMA(1,2).

Au travers de cette étude, nous avons pu relever particulièrement l’importance de la stationnarité d’un processus nécessaire à l’usage de la famille ARMA, mais également mieux approcher plusieurs notions soulevées dans le cadre d’une analyse de série chronologiques, tels que le test de Dickey-Fuller ou encore la fonction d’auto corrélation.

Toutefois, au-delà de la famille ARMA, nous avons également des familles telles que SARIMAX (Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors) ou encore ARCH (AutoRegressive Conditional Heteroskedasticity), qui pourraient faire l’objet d’une prochaine analyse.

WEBOGRAPHIE :

<https://www.codecademy.com/article/visualizing-time-series-data-with-python>

<https://mlforanalytics.com/2019/09/25/financial-analytics-log-returns-using-python/>

<https://medium.com/codex/what-is-stationarity-in-time-series-how-it-can-be-detected-7e5dfa7b5f6b>

<https://note.nkmk.me/en/python-pandas-nan-judge-count/>

<https://plotly.com/python/time-series/>

<https://www.askpython.com/python/examples/calculate-summary-statistics>

<https://www.statology.org/kolmogorov-smirnov-test-python/>

<https://www.statology.org/dickey-fuller-test-python/>

<https://www.projectpro.io/recipes/deal-with-missing-values-in-timeseries-in-python>

<https://www.analyticsvidhya.com/blog/2021/07/stock-market-forecasting-using-time-series-analysis-with-arima-model/>

1. Principalement en statistiques, nous abordons la notion de valeurs ou données manquantes lorsque nous n’avons pas d’observations pour une variable donnée, pour un individu donné. [↑](#footnote-ref-1)
2. Les valeurs aberrantes sont également appelées valeurs extrêmes ou encore ‘’outlier’’ en anglais .Ce sont des valeurs qui s’éloignent fortement des valeurs des autres observations, anormalement faibles ou élevées. [↑](#footnote-ref-2)
3. Une distribution normale fait référence à la loi de probabilité que suivent les données, en l’occurrence la loi de Gauss, qui se caractérise par deux paramètres principaux : son espérance mathématique qui est un nombre réel noté ‘’µ’’ et son écart-type qui est un nombre réel positif noté σ. [↑](#footnote-ref-3)
4. Quantile-Quantile [↑](#footnote-ref-4)
5. La fonction d’autocorrélation se réfère à la mémoire du processus ou encore de la série chronologique. En d’autres termes ; elle mesure la corrélation entre des observations d’une série temporelle séparées par k unités de temps (yt et yt–k). Elle est une alternative nous permettant de confirmer une des conditions d’un processus stationnaire concernant sa fonction d’auto covariance. Ne pouvant la visualiser via un graphique, nous la transformons en fonction d’autocorrélation et traçons ‘’l’auto correlogramme function’’. [↑](#footnote-ref-5)
6. [↑](#footnote-ref-6)
7. Résultante de l’équation caractéristique associée au processus ; sa présence traduit une non stationnarité de ce dernier. [↑](#footnote-ref-7)
8. Ceci fait référence à une uniformité de volatilités pouvant laisser supposer la présence d’une heteroscedasticité qui n’a pas été vérifiée dans le cadre de ce projet. [↑](#footnote-ref-8)
9. Auto Regressive Mooving Average. [↑](#footnote-ref-9)
10. Dans le cadre de ce projet, nous n’avons pas poursuivi jusqu’au test du modèle. Nous avons simplement usé des données d’entrainement qui représentaient une large partie de la distribution. [↑](#footnote-ref-10)
11. Les critères d’information mesurent la qualité des modèles statistiques et facilitent la sélection du modèle qui représente au mieux la réalité. Nous avons énoncé deux en particulier, notamment : le AIC (Akaike Information Criterion) ainsi qu’une version plus amelioree le BIC (Bayesian Information Criterion). [↑](#footnote-ref-11)