RAPPORT DU PROJET : ANALYSE DES PROPRIÉTÉS STATISTIQUES DES SÉRIES FINANCIÈRES
CAS D'APPLICATION : S&P 500

AUTEURS : YOUSSEF BENALLAL - MOHAMED AMINE CHAFIK
MOHAMED EL MEHDI CHBIHI - YOUSSEF LAZREK

ENCADRANT : KHALIL SAID
EXAMINATEUR : SAAD BENJELLOUN

Abstract

Nous (auteurs de ce travail) essayons d'analyser quelques propriétés statistiques phares de l'indice boursier américain du S&P 500. D'autre part, nous étudions dans une seconde partie la modélisation et la prédiction de ladite série étudiée. Ce document représente le rendu d'un projet académique s'inscrivant dans le cadre du module statistique de la 2ème année ingénieur de l'école EMINES - School of Industrial Management.

Table des matières

Table des figures

Chapitre 1

Introduction aux séries financières

1.1 Introduction

Les statistiques sont une science consistant en l'application d'outils mathématiques afin d'évaluer divers éléments du monde réel. Cette évaluation se fait à travers l'analyse, la compréhension, l'explication des évènements passés et même la prédiction des évènements futurs avec plus ou moins de précision. Appliquées dans divers domaines (Sports, démographie, politique...) Nous allons, à travers cette étude, nous consacrer à un pan essentiel des applications des statistiques : l'étude de séries financières. En effet, comprendre l'évolution et les caractéristiques des cours de bourses ouvre plusieurs possibilités aux financiers afin de pouvoir prédire l'évolution de ceux-ci et donc à la fois maximiser les profits et anticiper d'éventuelles crises.

1.2 Les séries temporelles

Les séries temporelles constituent une branche de l'économétrie dont l'objet est l'étude des variables au cours du temps. Parmi ses principaux objectifs figurent la détermination de tendances au sein de ces séries ainsi que la stabilité des valeurs (et de leur variation) au cours du temps. On distingue notamment les modèles linéaires (principalement AR et MA, pour Auto-Regressive et Moving Average) des modèles

conditionnels (notamment ARCH, pour Auto-Regressive Conditional Heteroskedasticity). L'analyse de ces séries touche énormément de domaines de la vie professionnelle, et plus précisément celui de l'informatique décisionnelle. Une analogie intéressante pourraît-être faite est qu'une série temporelle analyse ressemblerait à un homme très âgé avec beaucoup d'expérience et une sagesse assez grande pour tirer des événements passés des indications sur le futur, une sorte d'oracle. En informatique, ce serait plutôt une structure fondée sur les bases de données, fournissant ainsi le volume nécessaire d'information permettant de dresser une chronique historique des événements passés. Dessus viendrait se greffer un protocole d'extraction des données, intégré suivant un modèle judicieusement adapté à l'analyse que l'on voudrait faire. Enfin, au sommet de cette pyramide, la réponse à la question posée au départ, qui sera la prévision.

Une série temporelle est donc toute suite d'observations correspondant à la même variable : il peut s'agir de données macroéconomiques (le PIB d'un pays, l'inflation, les exportations…), microéconomiques (les ventes d'une entreprise donnée, son nombre d'employés, le revenu d'un individu, le nombre d'enfants d'une femme…), financières (le CAC40, le prix d'une option d'achat ou de vente, le cours d'une action), météorologiques (la pluviosité, le nombre de jours de soleil par an…), politiques (le nombre de votants, de voix reçues par un candidat…), démographiques (la taille moyenne des habitants, leur âge…). En pratique, tout ce qui est chiffrable et varie en fonction du temps. La dimension temporelle est ici importante car il s'agit de l'analyse d'une chronique historique : des variations d'une même variable au cours du temps, afin de pouvoir comprendre la dynamique. La périodicité de la série n'importe en revanche pas : il peut s'agir de mesures quotidiennes, mensuelles, trimestrielles, annuelles… voire même sans périodicité.

1.3 La série financière étudiée : S&P 500

Géré par l'agence de notation financière Standard & Poor's, le S&P 500 est un indice basé sur 500 grandes sociétés cotées sur les bourses américaines.

FIGURE 1.1: Top 10 des entreprises (et leur pondération) du S&P500

S&P 500 TOP TEN	
Name	**Weight %**
Apple	4.6
Microsoft	4.1
Alphabet	3.3
Amazon.com	3.1
Facebook	2.1
Berkshire Hathaway	1.9
JPMorgan Chase & Co	1.4
Visa	1.3
Johnson & Johnson	1.3
Walmart	1.1

Source: Bloomberg

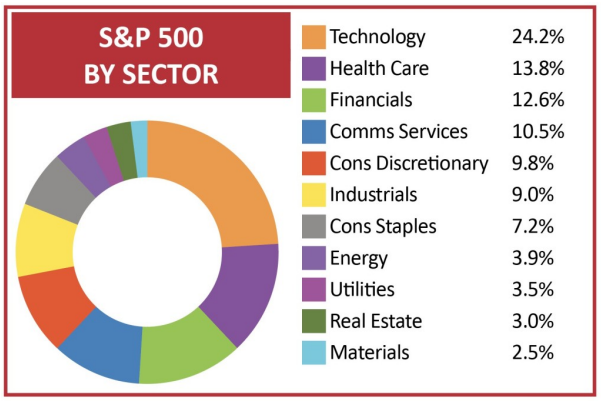
C'est l'indice le plus suivi par les gérants de fonds et les différents acteurs de la finance en général, sa représentativité atteint près de 80% de la capitalisation totale des marchés américains. On notera que 84.4% de la capitalisation totale du S&P500 provient du NYSE, 15,5% du NASDAQ et 0.1% de l'AMEX (troisième marchés américain).

Représentatif du marché boursier américain, il a peu à peu détrôné le Dow Jones Industrial Average (qui comprend les 30 plus grosses entreprises). En effet, il comprend davantage de titres (505 valeurs S&P500 contre 30 valeurs DJIA, Les actions du SP 500 sont au nombre de 505 car l'indice comprend 2 catégories d'actions pour 5 de ses entreprises.). De plus, sa valeur tient compte de la capitalisation boursière des entreprises qui le composent alors que celle du Dow Jones ne se base que sur les cours de la bourse. C'est-à-dire qu'une variation d'un dollar dans l'action SP500 d'une grosse entreprise aura donc plus d'impact sur l'indice que celle d'une entreprise plus petite. C'est pour cela que le S&P 500 est un indice plus représentatif.

Même au niveau mondial, le S&P500 est considéré comme un indice de référence

étant donné qu'il regroupe un grand nombre de sociétés, pas forcément américaines, mais côté sur les marchés américains qui font office de "baromètre" de l'économie mondiale. Au delà de cela, celui-ci englobe les divers secteurs de l'économie mondiale et de ce fait est grandement représentatif de la situation économique mondiale.

FIGURE 1.2: Secteurs couverts par le S&P500



Source : Shares Magazine

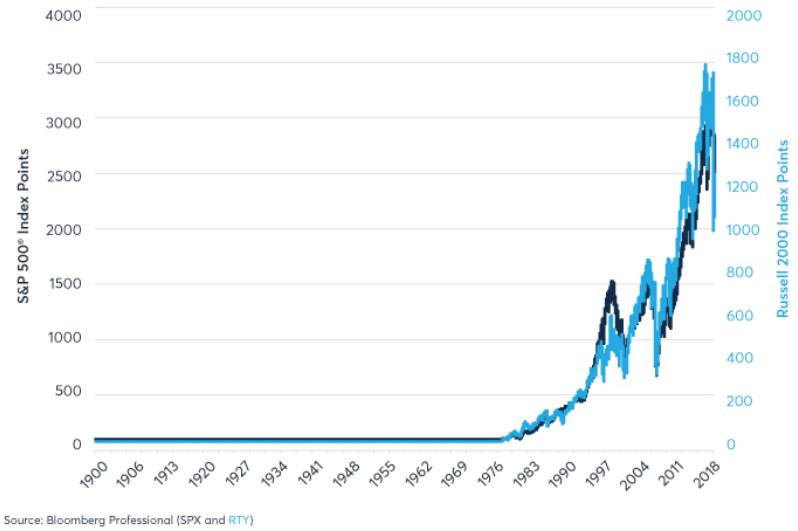
Compte tenu de tout cela, nous avons décidé d'orienter notre étude vers cet indice, estimant qu'une compréhension des rouages et des mécaniques de celui-ci reste applicable et permettrait de comprendre les diverses problématiques financières auxquelles nous devrons faire face.

1.4 Présentation des données

Lorsque l'on observe la variation du S%P500, depuis le jour de sa création jusqu'à aujourd'hui, on remarque qu'avant le début des années 90, celui-ci connaissait une croissance plutot stable et dont les fluctutions étaient nettement négligeables par rapport aux fluctuations notées à partir des années 90. De ce fait, à des fins d'efficience et de rapidité d'éxécution, nous estimons qu'une approche concentrée sur la phase post-1990.

Ici, nous avons réussi à obtenir un dataset conséquent à travers la plateforme Kaggle.Celui-ci regroupe les données d'ouverture, fermeture, la valeur la plus élevée, la plus basse, ainsi que la valeur de fermeture ajustée au dollar de 2020 (afin que les

FIGURE 1.3: Évolution du S&P500



diverses comparaisons et études soient représentatives, étant donnée l'inflation), pour chaque jour entre le 29 janvier 1993 et 24 décembre 2020. À cela vient s'ajouter le volume de transactions effectuées au cours de ladite journée.

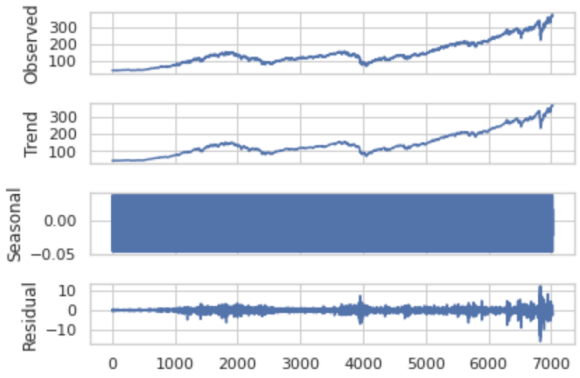
FIGURE 1.4: Aperçu du dataset utilisé

	Date	Open	High	Low	Close	Adj Close	Volume
0	1993-01-29	43.968750	43.968750	43.750000	43.937500	25.968958	1003200
1	1993-02-01	43.968750	44.250000	43.968750	44.250000	26.153660	480500
2	1993-02-02	44.218750	44.375000	44.125000	44.343750	26.209057	201300
3	1993-02-03	44.406250	44.843750	44.375000	44.812500	26.486113	529400
4	1993-02-04	44.968750	45.093750	44.468750	45.000000	26.596937	531500
...	...	...	...	...	...	...	...
7024	2020-12-18	370.970001	371.149994	367.019989	369.179993	369.179993	136542300
7025	2020-12-21	364.970001	378.459991	362.029999	367.859985	367.859985	96386700
7026	2020-12-22	368.209991	368.329987	366.029999	367.239990	367.239990	47949000
7027	2020-12-23	368.279999	369.619995	367.220001	367.570007	367.570007	46201400
7028	2020-12-24	368.079987	369.029999	367.450012	369.000000	369.000000	26457900

1.4.1 Cours du S&P500 : Valeur de clôture

Nous allons étudier la colonne "Close" désignant le prix à la clôture de l'indice S%P500, comme représentation de l'évolution globale de celui-ci. En effet, ce choix se base sur le fait qu'en premier lieu nous disposons d'une colonne où les valeurs sont ajustées et donc dont le traitement a réellement du sens. Au delà de cela, le prix à la

FIGURE 1.5: Decomposition du cours de clôture



fermeture est plus représentatif de l'évolution, vu que celui-ci prend en compte tous les evenements de la journée. Un autre facteur non negligeable est que comme pour "l'effet lundi", il y a un effet ouverture en bourse et donc prendre la clôture permet d'avoir une representation plus fidèle de l'évolution du S&P500.

1.4.2 Décomposition du cours de clôture

Nous pouvons donc procéder à une analyse plus poussée concernant lesdites données. En premier lieu, il nous incombe de réaliser une décomposition de notre série financière (en l'occurence le S&P500) afin d'avoir une meilleure compréhension des tendances qu'elle affiche, d'éventuelles saisonnalités etc.

Ici, on peut clairement observer qu'il y a une nette tendance croissante de notre série. Évidemment, il y a aussi eu quelques baisses principalement liées à 2 évènements majeurs : les attentas du 11 septembre 2001 et la crise des subprimes en 2008, mais globalement la valeur de l'indice étudié a continué de croître au fil du temps. Nous voyons aussi que la composante de saisonnalité est négligeable, et que le bruit (residual) est lui aussi d'amplitude négligeable par rapport à l'amplitude de la série traitée.

Cependant, pour une meilleure approche, où le facteur "tendance" ne rentre pas en jeu, nous allons introduire une nouvelle notion que nous allons étudier au vu de ses propriétés statistiques notamment la stationnarité : **le log-rendement**.

1.4.3 Log-Rendement

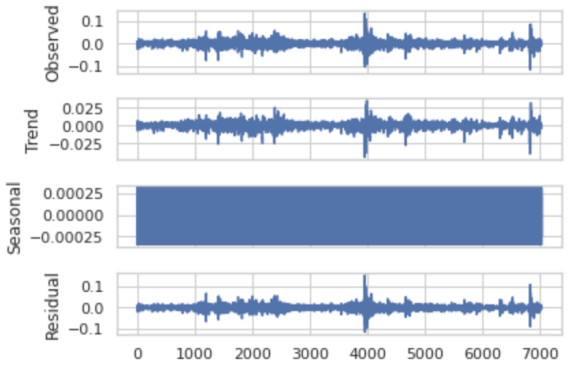
Soit x_t le cours du S&P500 au jour t, le rendement simple n'est autre que $r*_t = \frac{x_t - x_{t-1}}{x_{t-1}}$, on notera aussi le log-rendement comme suit : $r_t = log(\frac{x_t - x_{t-1}}{x_{t-1}})$.

On suppose dans divers cas que les prix suivent une distribution lognormale étant donné le comportement au niveau des queues de distribution (que nous traiterons plus amplement plus tard).

1.4.4 Décomposition de la valeur du log-rendement

Nous pouvons évaluer un peu plus en détail le log-rendement. Similairement, il nous faisons une décomposition dudit log-rendement afin d'observer les différences au niveau des éléments de la décomposition du log-rendement par rapport à ceux de la décomposition du cours du S&P500. D'après cette décomposition, il est clair que la

FIGURE 1.6: Decomposition du log-rendement



stationnarité dont nous faisions état aupravant est caractéristique du log-rendement par rapport au cours de clôture. On remarque aussi que la saisonnalité est elle aussi très faible. Cependant,le bruit (residual) est très présent et constitue (logiquement) la quasi totalité du log-rendement, ce qui est compatible avec la nature dudit log-rendement.

Chapitre 2

Analyse statistique des données

2.1 Statistiques descriptives

Les statistiques descriptives représentent le premier pas d'appréhension de notre série log-rendement et est également un état des lieux des particularités statistiques qu'elle présente.

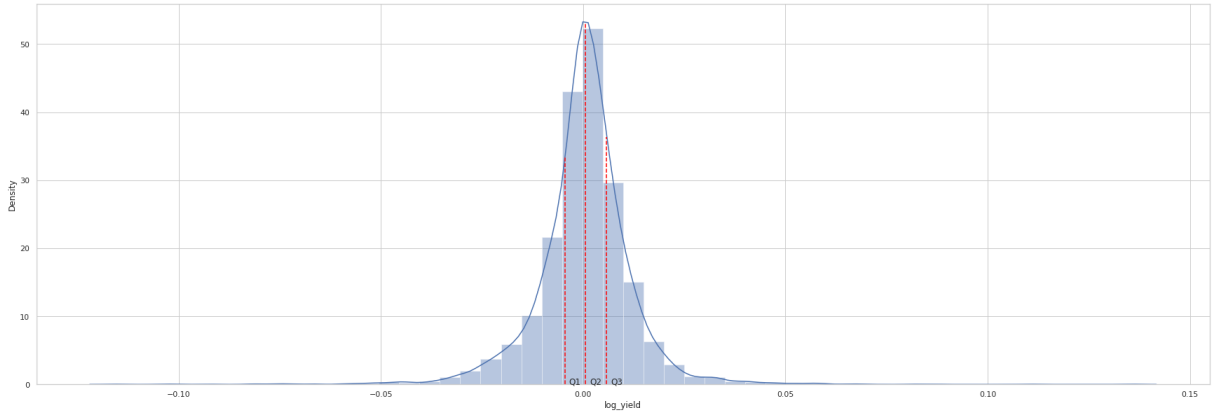


FIGURE 2.1: Histogramme du log-rendement de la série S&P 500 [**?**]

Moyenne	Variance	Minimum	Maximum	Asymétrie	Aplatissement
$0.30x10^{-3}$	0.01	-0.11	0.13	-0.29	11.62

Nous définissons les formules des coefficients :

$$Coeff d'asymetrie = E(\frac{X-\mu}{\sigma})^3 \quad (1)$$

$$Coeff d'aplatissement = E(\frac{X-\mu}{\sigma})^4 \quad (2)$$

Ces derniers coefficients devant être nuls pour une distribution normale nous rappellent la réalité des données réelles. Même si, le coefficient d'asymétrie est quasi-nul nous devrions quand même noter l'aplatissement positif **(d'une loi exponentielle)** de notre série log-rendement.

2.2 Stationnarité

La stationnarité est une caractéristique très importante des séries temporelles permettant d'évaluer si les propriétés statistiques telles : l'espérance, la variance ou l'auto-corrélation restent inchangées par rapport au temps. À cet effet, nous procédons au **Test de Dickey-Fuller augmenté** (il s'agit d'un type de test statistique appelé test de racine unitaire) où l'intuition derrière le test de racine unitaire est qu'il détermine dans quelle mesure une série chronologique est définie par une tendance. Donc, nous aurons à examiner **l'hypothèse nulle (H0) de ce test pour les deux colonnes Close et Log.yield** pour valider notre intérêt à cette transformation en logarithme et qui sera plus explicité ultérieurement dans ce rapport : Si (H0) n'est pas rejetée, cela suggère que la série temporelle a une racine unitaire, ce qui signifie qu'elle est non stationnaire. Elle a une certaine structure dépendante du temps.

Nous interprétons ce résultat en utilisant la valeur p du test :

p-value > 0,05 : échec du rejet de l'hypothèse nulle (H0), les données ont une racine unitaire et sont non stationnaires.

p-value $\leq$ 0.05 : rejeter l'hypothèse nulle (H0), les données n'ont pas de racine unitaire et sont stationnaires.

```
ADF Statistic for the -Close- Column:  1.54
p-value :  0.998   p-value > 0.05 : the data is non-stationary
```

FIGURE 2.2: Résultat du test Dickey-Fuller pour la colonne Close

```
ADF Statistic for the -log_yield- Column:  -15.466
p-value :  0.0   p-value < 0.05 : the data is stationary
```

FIGURE 2.3: Résultat du test Dickey-Fuller pour la colonne $\log_y ield$

Résultat important : La stationnarité de notre série est validée si on manipule la colonne du $\log_y ield$.

2.3 Normalité

La normalité est un test statistique très important dans l'étude des séries temporelles. Nous visons à tester cette hypothèse via le test de Shapiro ayant comme hypothèse nulle la normalité de la distribution, le résultat représente une p-value largement inférieure à 5%, ce qui nous mène à rejeter la normalité des log-rendements.

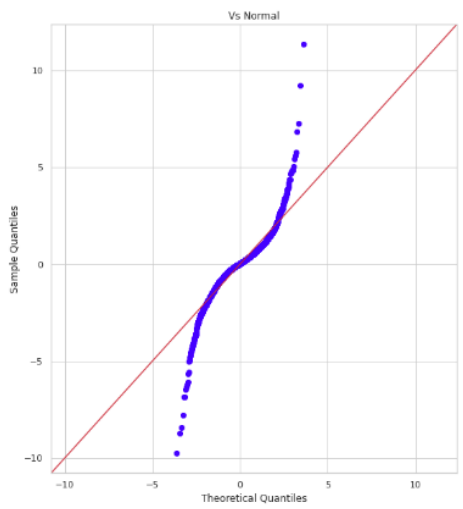
```
Shapiro-Wilk Statistic :  0.895
p-value :  0.0   p-value < 0.05 : the data is not normal
```

FIGURE 2.4: Résultat du test Shapiro

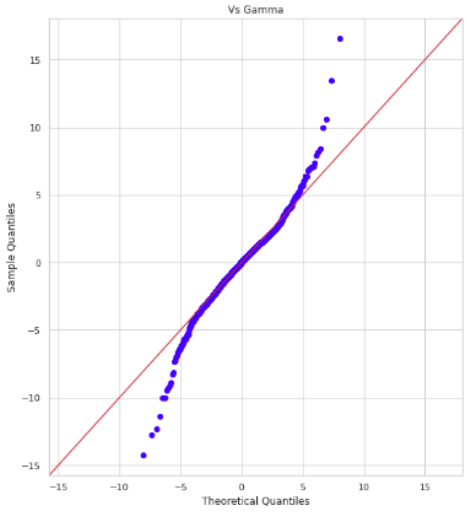
Cette étude va s'étendre dans la partie suivante pour comprendre les autres lois classique.

2.4 Diagrammes Q-Q

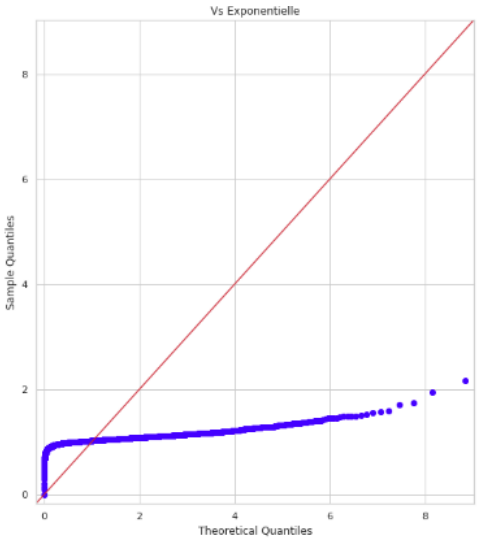
Le diagramme Quantile-Quantile couramment appelé le Q-Q plot est un graphique qui permet d'évaluer les quantiles d'une distribution donnée (en l'occurrence notre série log-rendement) avec ceux d'une distribution théorique.



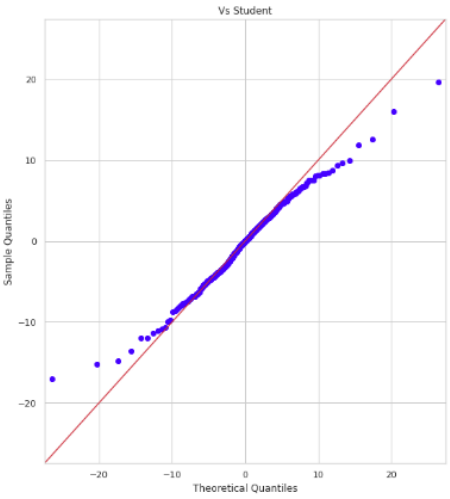
(a) Loi normale



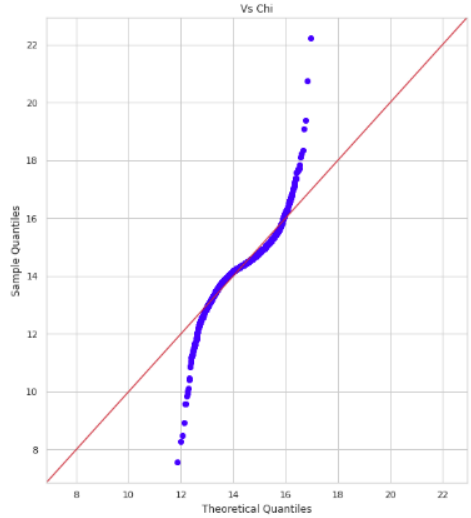
(b) Loi Gamma



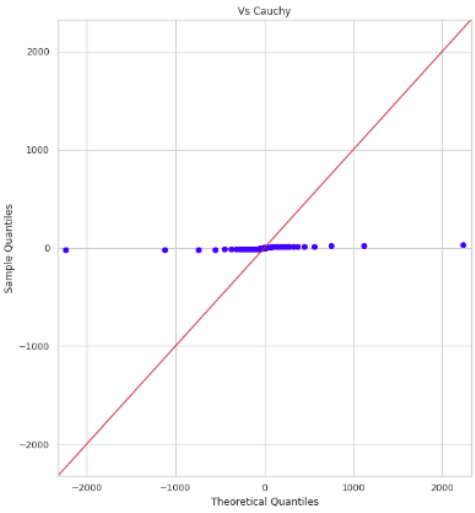
(c) Loi Exponentielle



(d) Loi Student



(e) Loi Chi



(f) Loi Cauchy

FIGURE 2.5: Q-Q plots de la série log-rendement avec des lois classiques

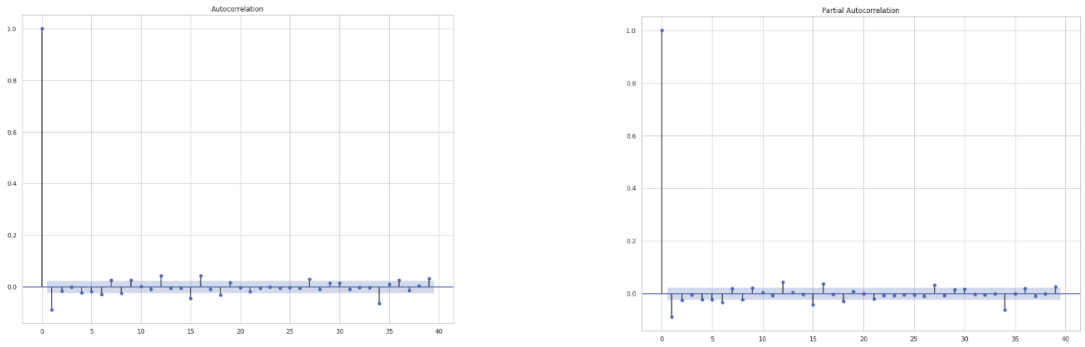
Nous remarquons clairement qu'il n'y pas égalité (suivant la ligne des 45 degrés) entre la distribution théorique et la série du log-rendement. Notamment dans le quatrième quartile des lois normale, Gamma et Student qui étaient de bons candidats pour reproduire le comportement des log-rendement. Un résultat attendu pour la loi normale compte tenu du test de normalité effectué dans la partie 2.3 .

2.5 Auto-corrélation

L'autocorrélation (ou l'autocovariance) d'une série fait référence au fait que dans une série temporelle ou spatiale, la mesure d'un phénomène à un instant t peut être corrélée aux mesures précédentes (au temps $t-1$, $t-2$, $t-3$, etc.) ou aux mesures suivantes (à $t+1$, $t+2$, $t+3$, ...). Une série autocorrélée est ainsi corrélée à elle-même, avec un décalage (lag) donné. Ainsi, l'autocorrélation depend de l (lag) et son expression empirique d'ordre l est :

$$\gamma_l = \frac{E(X_t - \mu)(X_{t+l} - \mu)}{\sigma^2} \quad (3)$$

Nous représentons cela sous forme des corrélogrammes affichés ci-dessus. La bande bleue représente le seuil au dessous duquel la corrélation est négligée. Notons que la différence entre l'ACF et le PACF réside dans le fait que la corrélation calculée du deuxième prend en considération les valeurs des intervalles intermédiaires.



(a) ACF (b) PACF

FIGURE 2.6: ACF/PACF de la série log-rendement

Conclusion : Il n'y a pas d'auto corrélation du log_yield. Le log_yield est alors un bruit blanc qu'on va étudier plus rigoureusement dans la partie modélisation et prévision.

2.6 Interprétation financière des propriétés

L'analyse statistique de la série des log-rendement nous a permis de dégager des points pertinents suivants :

1. La moyenne de la série du log-rendement certes est positif mais est **de l'ordre de** 10^{-3} **et son maximum (jour en jour) de 0,13**. En d'autres termes, un acteur présent assez longuement (années) assure en probabilité un rendu positif qui est très faible et le maximum que peut gagner un investisseur dans le S&P 500 qui trade dans la journée (sans omettre les commissions et les frais qui lui sont dus) est de **10% de son capital** en faisant face à beaucoup de risque.
2. La non-normalité de la série est un résultat cohérent. En effet, les séries financières ont en général un allure normale mais présentent des queues plus épaisses. **Les rentabilités inattendues (positives ou négatives) - outliers - ont plus de chances de se produire par rapport à la normale.**
3. L'absence de corrélation est aussi un point à souligner. Concrètement, **la corrélation d'une série financière ne peut pas être présente car elle résumerait l'investissement en bourse à la connaissance du résultat d'un instant pour prédire rigoureusement l'instant additionné au lag.** Ceci va contredire la nature même de la bourse et des actifs de type actions, indices et leur volatilité associée.

Chapitre 3

Modélisation et Prédiction I

3.1 Lissage exponentiel

3.1.1 Définition

Le lissage exponentiel utilise une logique similaire à celle de la moyenne mobile, mais cette fois, un poids décroissant différent est attribué à chaque observation. En d'autres termes, une importance moindre est accordée aux observations à mesure que l'on s'éloigne de la valeur à prédire.

3.1.2 Types

Lissage exponentiel simple

Ce modèle est aussi parfois appelé le modèle à moyenne mobile exponentiellement pondérée. Le lissage exponentiel simple permet de prédire une valeur en fonction des données passées, en donnant aux données un poids d'autant plus faible qu'elles correspondent à un passé éloigné. La pondération évolue de façon exponentielle, d'où le nom du modèle. En matière de prévision, ce modèle est assez limité, puisque les prévisions sont constantes au-delà de n+1. La formule du

lissage exponentiel simple s'écrit comme suit :

$$F_t = \alpha \cdot y_{t-1} + (1 - \alpha) \cdot F_{t-1} \tag{1}$$

avec :

F_t : la prévision en t.

α : Facteur de lissage qui prend des valeurs entre 0 et 1.

Lissage exponentiel double

Dans ce modèle les prévisions tiennent ici compte d'une tendance observée sur les données précédentes. Cette méthode implique une équation de prévision et deux équations de lissage, une pour l'état de la série (level l_t) et une pour la tendance (b_t) :
Équation de prévision :

$$\hat{y}_{t+h,h} = l_t + h \cdot b_t \tag{2}$$

Équation de l'état :

$$l_t = \alpha \cdot y_t + (1 - \alpha) \cdot (l_{t-1} + b_{t-1}) \tag{3}$$

Équation de tendance :

$$b_t = \beta \cdot (l_t + l_{t-1}) + (1 - \beta) \cdot b_{t-1} \tag{4}$$

α : Facteur de lissage qui prend des valeurs entre 0 et 1.

β : Facteur de lissage pour la tendance qui prend des valeurs entre 0 et 1.

$\hat{y}_{t+h,h}$: la prévision en $t + h$ (h pas)

La fonction de prévision n'est plus plate mais tendancielle. La prévision avec h pas est égale au dernier niveau estimé plus h fois la dernière valeur de tendance estimée. Les prévisions sont donc une fonction linéaire de h. (Toutes les visualisations qui vont suivre se situent entre juin 2019 et juillet 2020.)

3.1.3 Résultats du lissage

Représentation graphique du lissage exponentiel simple :

FIGURE 3.1: Lissage avec différentes valeurs de α [**?**]

Représentation graphique du lissage exponentiel double :

FIGURE 3.2: Lissage avec différentes valeurs de (α, β) [**?**]

3.1.4 Résultats de la prévision

Lissage exponentiel simple :

La prévision du lissage exponentiel simple est une constante. En optimisant la valeur l'erreur quadratique moyenne, nous avons trouvé les résultats suivants :

$\boldsymbol{\alpha = 0.27, RMSE = 2.56}$ (pour la série initiale)

$\boldsymbol{\alpha = 0.27, RMSE = 0.006}$ (pour la série log rendement)

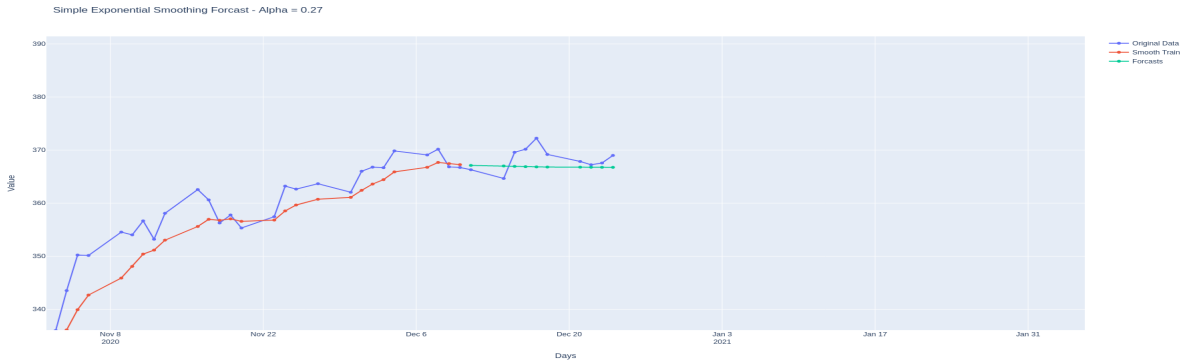


FIGURE 3.3: Série initiale [**?**]

FIGURE 3.4: Série log rendement [**?**]

Lissage exponentiel double

FIGURE 3.5: Série initiale [**?**]

On trouve une valeur optimale de l'erreur quadratique pour les facteurs α, β qui suivent :

$(\boldsymbol{\alpha, \beta}) = (\mathbf{0.6, 0.21}), \boldsymbol{RMSE} = \mathbf{1.98}$ (pour la série initiale)

3.2 Lissage par moyenne mobile

3.2.1 Définition

Le lissage par moyenne mobile permet de prendre en compte de manière simple et contrôlée des observations passées pour prédire le futur. L'utilité de la méthode réside plus dans sa nature de filtre, permettant de retirer à une série son bruit de fond, et de faire alors ressortir les grandes tendances.

3.2.2 Formule

Le lissage par la moyenne mobile exprime le forecast en t à travers la formule qui suit :

$$F_t = \frac{y_t + y_{t-1} + y_{t-2} + ... + y_{t-L}}{L+1} \tag{5}$$

3.2.3 Résultats du lissage

FIGURE 3.6: Moyenne mobile , size = 5 days [**?**]

Commentaire : En bleu on visualise la courbe du cours S-P 500, en rouge on visualise la tendance à la moyenne mobile. Le modèle de la moyenne mobile nous donne un bon aperçu de la tendance générale de nos séries. (MA) indique que le cours de l'action va probablement continuer à augmenter dans les jours à venir.

FIGURE 3.7: Moyenne mobile , size = 1 month [**?**]

3.2.4 Résultats du forecasting

Le forecasting à l'aide de la méthode moyenne mobile nous donne le résultat suivant pour un window size de 5 jours. En appliquant cette méthode de lissage sur

FIGURE 3.8: Forecasting , size = 5 days [**?**]

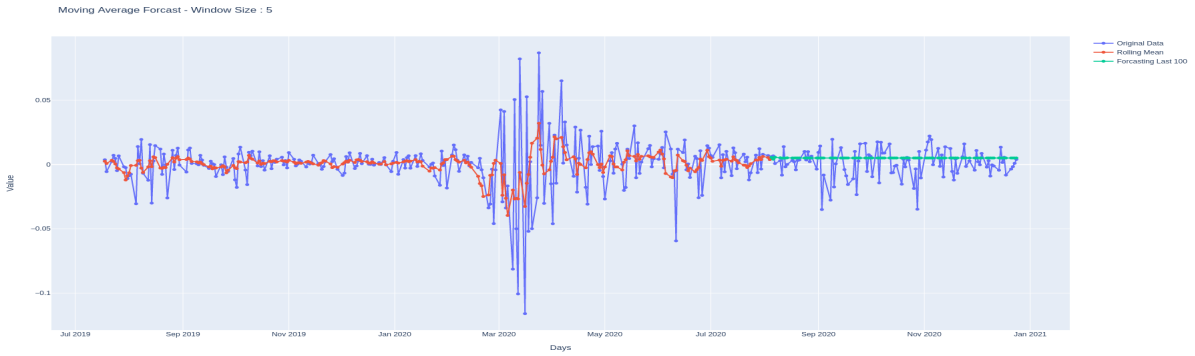


FIGURE 3.9: Forecasting log rendement , size = 5 days [**?**]

la série de données initiales :

$$RMSE = \sqrt{\sum_{1}^{n} \frac{(\check{y}_i - y_i)^2}{2}} = 23.633 \tag{6}$$

La méthode (MA) suppose que les composantes de tendance et de saisonnalité des séries temporelles ont déjà été éliminées. Donc nous allons appliquer cette méthode sur la série log-rendement qui est non stationnaire, on trouve les résultats suivants :

$$RMSE = 0.012 \tag{7}$$

3.2.5 Explication

La forecasting de la méthode lissage par moyenne mobile est une constante à travers le temps car si nous considérons un échantillon de 1 à n, la valeur prédite en n+1 sera :

$$y_{n+1} = \frac{y_1 + y_2 + ... + y_n}{n} \tag{8}$$

la valeur prédite en n+2 sera :

$$y_{n+2} = \frac{y_1 + y_2 + ... + y_{n+1}}{n+1} \tag{9}$$

$$\Rightarrow \frac{n+1}{n}.(\frac{y_1 + y_2 + ... + y_n}{n+1} + \frac{y_{n+1}}{n+1}) = \frac{n+1}{n}.y_{n+2} \tag{10}$$

$$\Rightarrow \frac{n+1}{n}.y_{n+1} = \frac{n+1}{n}.y_{n+2} \tag{11}$$

$$\Rightarrow y_{n+1} = y_{n+2} \tag{12}$$

D'où la prévision de la méthode lissage par moyenne mobile est une constante.

Chapitre 4

Modélisation et Prédiction II

4.1 Modèle ARIMA

4.1.1 Définition :

ARIMA est un algorithme de prévision basé sur l'idée que les informations contenues dans les valeurs passées de la série temporelle peuvent être utilisées pour prédire les valeurs futures. Il s'agit en fait de la combinaison de modèles plus simples pour obtenir un modèle complexe capable de modéliser des séries temporelles présentant des propriétés non stationnaires. Ces modèles sont :

— **Le modèle AutoregRessive AR(p)**) :

Dans ce modèle, nous prévoyons la variable d'intérêt en utilisant une combinaison linéaire de ses valeurs passées. Le terme autorégression indique qu'il s'agit d'une régression de la variable sur elle-même. Un modèle Autoregressif d'ordre p peut s'écrire comme suit :

$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + ... + \Phi_p y_{t-p} + \epsilon_t \quad (1)$$

y_t : l'observation à l'instant t

Φ_n : suite réelle

ϵ_t : bruit blanc

— **Le modèle de moyenne mobile MA(q)**) :

Plutôt que d'utiliser les valeurs passées de la variable à prévoir dans une régression, un modèle de moyenne mobile utilise les erreurs de prévision passées dans un modèle de type régression. Un modèle de moyenne mobile d'ordre q peut s'écrire comme suit :

$$y_t = b + \epsilon_t + \Theta_1\epsilon_{t-1} + \Theta_2\epsilon_{t-2} + ... + \Theta_q\epsilon_{t-q} \quad (2)$$

ϵ_{t-n} : l'erreur de la prévision à l'instant t-n

Θ_n : suite réelle

ϵ_t : bruit blanc

— **Ordre d'intégration I(d)**) : :

Le paramètre d représente le nombre de différences nécessaires pour rendre la série stationnaire. Si nous combinons la différenciation avec l'autorégression et le modèle de moyenne mobile, nous obtenons un modèle ARIMA(p, d, q). Le modèle complet peut être écrit comme suit avec y'_t : la série différentiée :

$$y'_t = c + \Phi_1 y'_{t-1} + \Phi_2 y'_{t-2} + ... + \Phi_p y'_{t-p} + \Theta_1\epsilon_{t-1} + \Theta_2\epsilon_{t-2} + ... + \Theta_q\epsilon_{t-q} + \epsilon_t \quad (3)$$

On peut utiliser l'ACF (AutoCorrelation Function) et le PACF (Partial AutoCorrelation Function) pour identifier des valeurs de p et de q. On procède comme suit :

→ On choisit p tel quel soit le plus grand décalage après lequel les autres décalages ne sont pas significatifs sur le graphique d'autocorrélation partielle.

→ On choisit q tel quel soit le plus grand décalage après lequel les autres décalages ne sont pas significatifs sur le graphique d'autocorrélation.

4.1.2 Différenciation et estimation des paramètres

Dans cette section nous allons essayer de trouver un ordre d de différentiation adéquat et extraire des valeurs appropriées des paramètres p et q afin de pouvoir calculer les prévisions correspondantes. Nous avons établi la fonction Diff qui reçoit

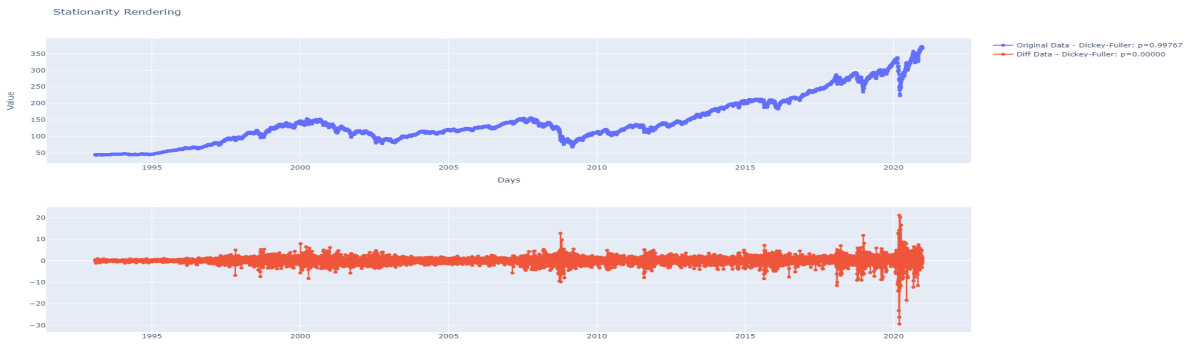


FIGURE 4.1: série initiale en bleu, série différentiée d'odre 1 en rouge [**?**]

comme arguments la série temporelle, le paramètre de différentiation d et le nombre de décalages à prendre en considération lors du dessin des plots ACF et PACF. Après avoir appliqué cette fonction pour d = 1, on obtient les résultats suivants :

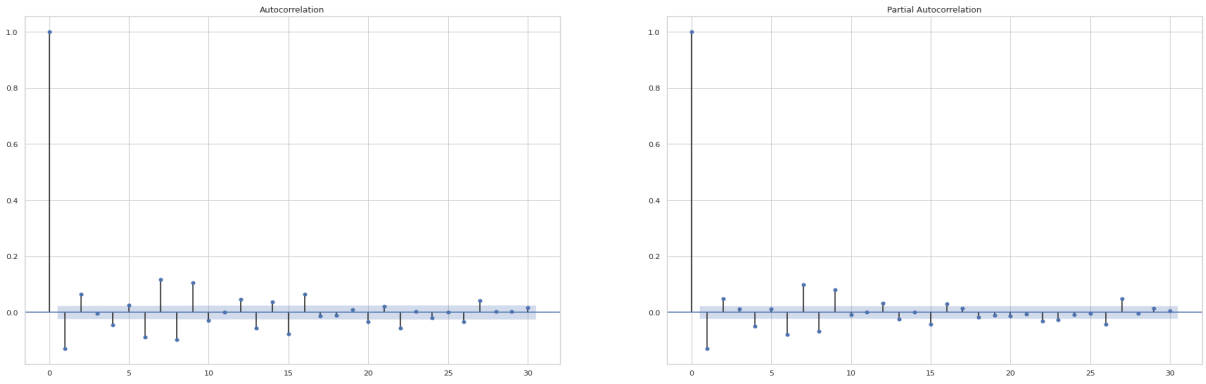


FIGURE 4.2: ACF et PACF de la série différentiée [**?**]

Comme nous pouvons le voir dans le graphique ci-dessus, pour d = 1, la p-value du test de stationnarité **"Augmented Dickey Fuller test"** est égale à zéro, ce qui signifie que nous pouvons rejeter l'hypothèse nulle, qui suggère que la série a une structure dépendante du temps. La série différenciée résultante est donc stationnaire. En plus, en observant les plots **ACF** et **PACF**, on constate qu'il n'y a pas

d'autocorrélation de la série différenciée. On peut donc la caractérisé par un bruit blanc. De ce fait, nous allons fixer la valeur du paramètre d à 1 tout au long de notre étude.

Notre série ressemble maintenant à quelque chose indescriptible, oscillant autour de zéro, Augmented Dickey-Fuller test indique qu'elle est stationnaire. Nous pouvons alors commencer la modélisation.

4.1.3 Prévisions

Commençons par déterminer les paramètres p et q. Une première vue des plots ACF et PACF indique qu'on peut prendre p = q = 1 vu que cette valeur représente le décalage après lequel les autres décalages ne sont pas significatifs. Notre modèle actuel est alors ARIMA(1, 1, 1)

Afin de réaliser des prévisions, nous avons établi la fonction ARIMAmodel qui prend pour arguments : la série initiale, les paramètres (p, d, q), le nombre de prévisions à calculer que nous avons choisi de le fixé à 10 tout au long de notre étude. Le nombre d'instances à prendre en considération lors du dessin des courbes de lissages et de prévisions. L'argument applylog qui, si égale à True, applique la transformation du log-rendement à la série avant de procéder à la modélisation (il est égal à False par défaut). L'argument ifplot qui, si égal à False, retourne une liste contenant le RMSE des prévisions ainsi qu'une liste des prévisions (il est égal à True par défaut).

Afin de réaliser les prévisions, nous avons d'abord divisé notre série en train set et test set. Ensuite, nous avons appliqué le modèle ARIMA au train set afin d'extraire la première prévision correspondante. Puis, on ajoute cette prévision au train set et on tire la prévision suivante. On répète ce processus jusqu'à ce que nous atteignions la dernière valeur de test set.

Appliquons maintenant cette fonction (ARIMA model) pour p = d = q = 1, avec une prévision sur 10 jours :

```
ARIMA_model(series = spy.Close, p = 1, d = 1, q = 1, n_test = 10, x_axis = spy.Date, plot_last = 365)
```

FIGURE 4.3: paramètres de la fonction [**?**]

FIGURE 4.4: prévision ARIMA sur la série initiale [**?**]

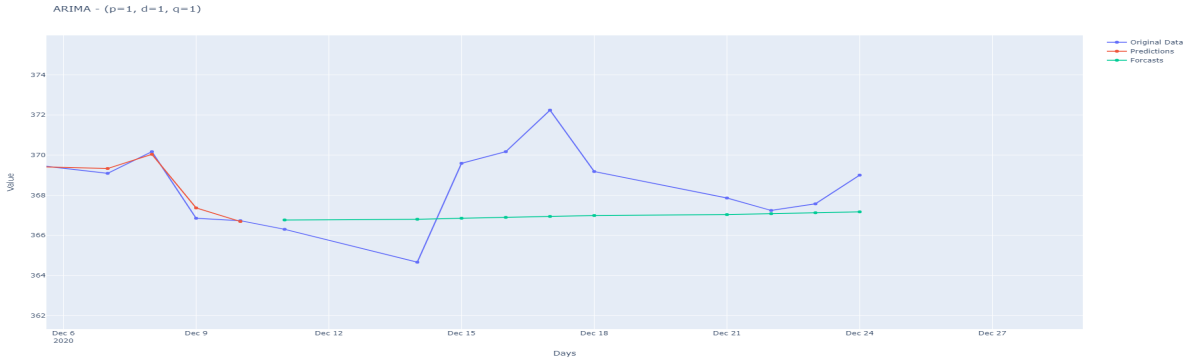


FIGURE 4.5: Zoom sur la prévision [**?**]

La prévision apparaît en vert dans la représentation graphique

$RMSE(prediction) = 0.255$

$RMSE(prevision) = 2.453$

Même si ce modèle présente de bons résultats, il peut être optimisé comme nous le verrons dans les prochaines étapes.

4.1.4 Optimisation du modèle

Optimisation du modèle à l'aide des corrélogrammes

Comme nous l'avons dit précédemment, les paramètres d'ARIMA peuvent être extraits des graphiques ACF et PACF. Dans cette section, nous allons tirer plusieurs valeurs de p et q en utilisant la même méthode d'extraction défini dans la partie dernière. Nous avons construit une fonction qui reçoit les valeurs de p et q et qui retourne le meilleur modèle sur la base de la métrique RMSE. L'implémentation de cette méthode d'optimisation est la suivante :

```
# Optimisons p et q en utilisant les corrélogrammes :
ps = [0, 1, 6, 7, 8, 9]
qs = [0, 1, 6, 7, 8, 9]

RMSE_OPT, failed_for = Manual_Opt_ARIMA(series = spy.Close, ps = ps, d = 1, qs = qs, n_test = 10)
print('Failed Conversion For : ', failed_for, '\n')

RMSE_OPT[RMSE_OPT.RMSE_Forecast == min(RMSE_OPT.RMSE_Forecast)]
```

FIGURE 4.6: Implémentation dans le code [**?**]

Nous obtenons le résultat suivant : **(ps,qs,RMSE)=(6,6,1.837)**

Optimisation du modèle à l'aide de l'AIC (Akaike's information criterion) :

Le Akaike information criterion (AIC) est une méthode mathématique permettant d'évaluer dans quelle mesure un modèle s'adapte aux données à partir desquelles il a été généré. En statistique, l'AIC est utilisé pour comparer différents modèles possibles et déterminer celui qui correspond le mieux aux données. L'AIC s'écrit comme suit :

$$AIC = -2\log(L) + 2K$$

L : vraisemblance des données

K : le nombre de paramètres En pratique, on sélectionne le modèle présentant l'AIC le plus petit.

Puisqu'on sélectionne le modèle ayant l'AIC le plus petit, un grand nombre de paramètres augmentera le score AIC et pénalisera donc le modèle. Or un modèle avec un grand nombre de paramètres pourrait être plus performant (risque de l'overfitting), l'AIC est alors utilisé pour trouver le modèle avec le plus petit nombre de paramètres qui donne de bons résultats. Le modèle le mieux ajusté selon l'AIC est celui qui explique la plus grande quantité de variation en utilisant le moins de paramètres possible.

Implémentation :

```
# Optimisation à l'aide de l'AIC :
ps = list(range(0, 9))
d = 1
qs = list(range(0, 9))

AIC_OPT = AIC_Opt_ARIMA(series = spy.Close, ps = ps, d = d, qs = qs, n_test = 10)

AIC_OPT[AIC_OPT.AIC == min(AIC_OPT.AIC)]
```

FIGURE 4.7: Implémentation dans le code [**?**]

Les paramètres optimales sont : $(p, d, q) = (6, 1, 6)$

4.1.5 Optimisation des prévisions

Modèle optimal

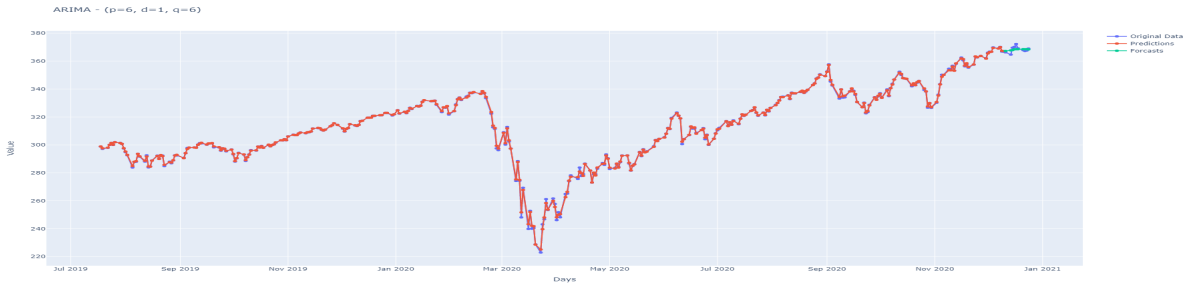


FIGURE 4.8: prévision sur 10 jours avec les paramètres optimales(6,1,6) ([**?**]

FIGURE 4.9: Zoom sur les prévisions [**?**]

Nous trouvons le résultat suivant :

(Params,AIC,RMSEpred,RMSE prevision)=((6,1,6),28211.10,0.42,1.83)

Les deux méthodes d'optimisation ont donné le même résultat. Le meilleur modèle jusqu'à présent est donc ARIMA(6, 1, 6).

Prévision avec log rendement

Afin d'appliquer la transformation du log−rendement, il suffit de rendre l'argument apply log égale à True. On obtient les résultats suivants :

(RMSEpred,RMSE prevision)=((0.23,1.82)

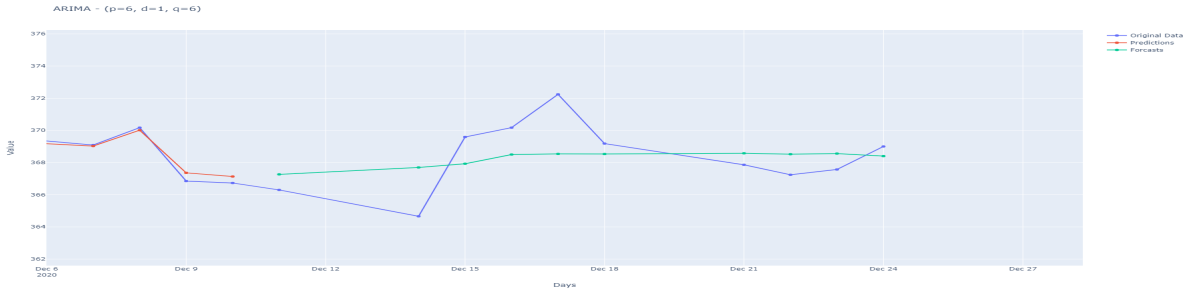


FIGURE 4.10: Zoom sur les prévisions après log rendement [**?**]

4.2 Étude du résidu du meilleur modèle - ARIMA(6, 1, 6)

4.2.1 Série, distribution et stationnarité des résiduels

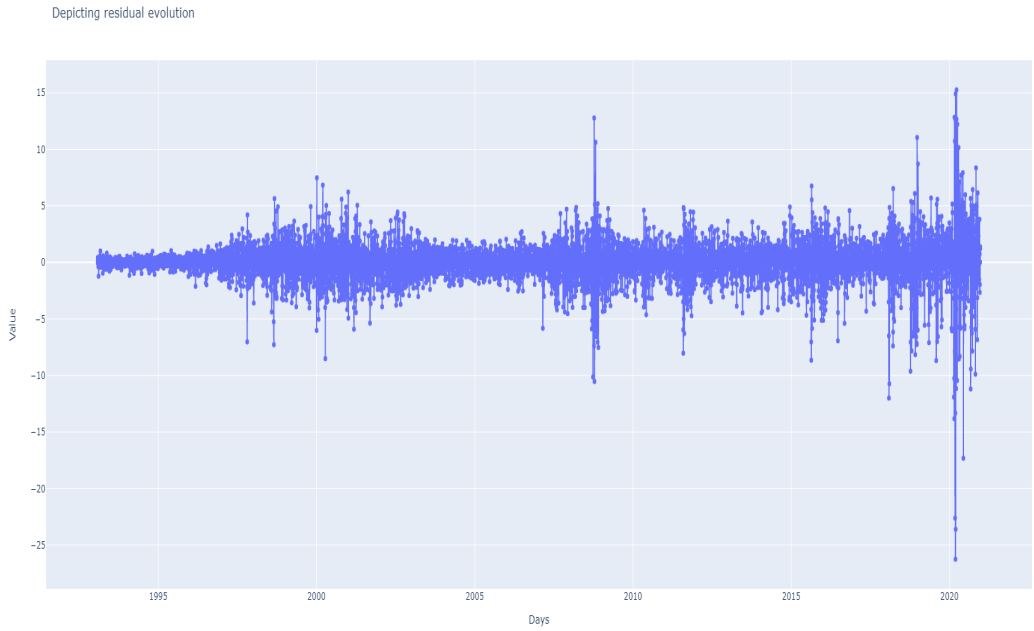


FIGURE 4.11: Série des résiduels

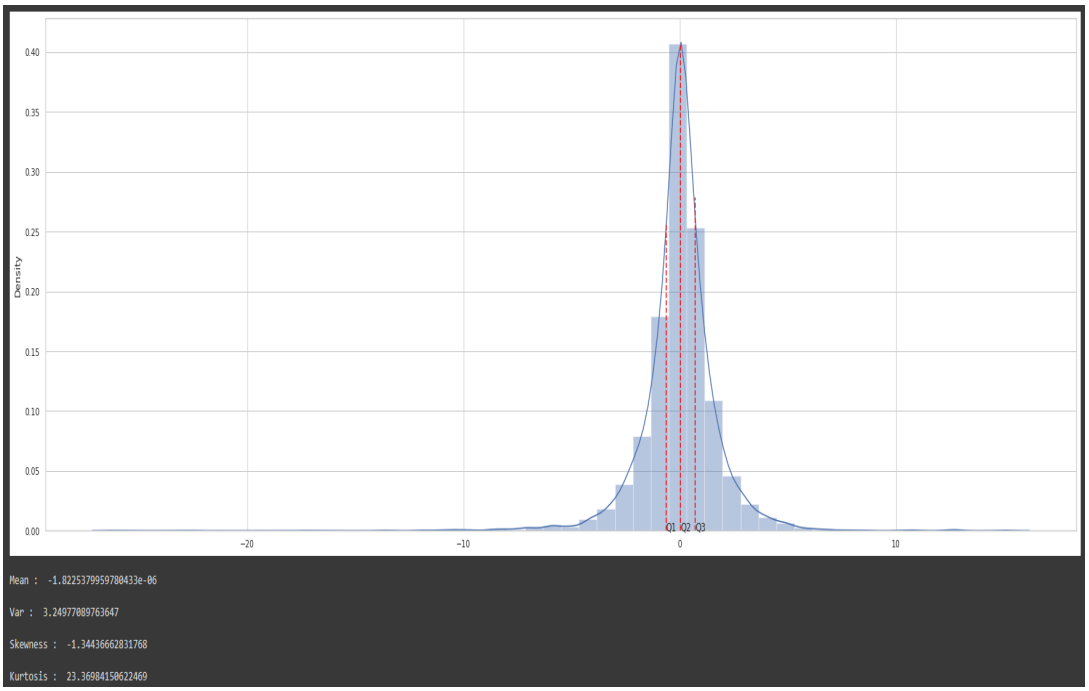


FIGURE 4.12: Distribution du Résiduel

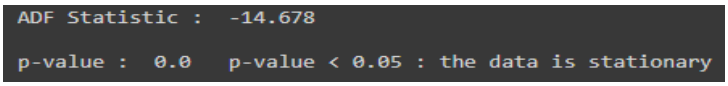


FIGURE 4.13: Test de Stationnarité ADF

4.2.2 Ljung-Box test

Le test de Ljung Box (parfois appelé test de Box-Pierce modifié, ou simplement test de Box) est un moyen de tester l'absence d'autocorrélation d'une série, jusqu'à un décalage m spécifié. Pour une série temporelle y de longueur n, la statistique du test est définie comme suit :

$$Q(m) = n(n+2)\sum_{j=1}^{m}\frac{r_j^2}{n-j},$$

Où r_j est l'autocorrélation estimée de la série au décalage j, et m est le nombre de décalages testés.

Le test détermine si les résidus sont un bruit blanc ou s'ils sont dépendents du temps ; c'est à dire si les autocorrélations des résidus sont non nulles. Il s'agit essentiellement d'un test de manque d'ajustement ("lack of fit") : si les autocorrélations des résidus sont très faibles, nous disons que le modèle ne présente pas de "manque d'ajustement significatif".

L'hypothèse nulle du test de Ljung Box, H0, suggère que le modèle ne présente pas de manque d'ajustement (Les données sont distribuées indépendamment (pas d'autocorrélation)). L'hypothèse alternative, H1, suggère que le modèle présente un manque d'ajustement (Les données sont autocorrélés).

Une p-value >= 0.05 suggère que l'hypothèse nulle selon laquelle la série temporelle n'est pas autocorrélée est rejetée par ce test.

Après avoir appliqué le test de Ljung-Box. On obtient les résultats suivants :

On conclu que l'hypothèse nulle qui stipule "Les données sont distribuées indépendamment (pas d'autocorrélation)" ne peut être rejetée avant 16ème lag. On

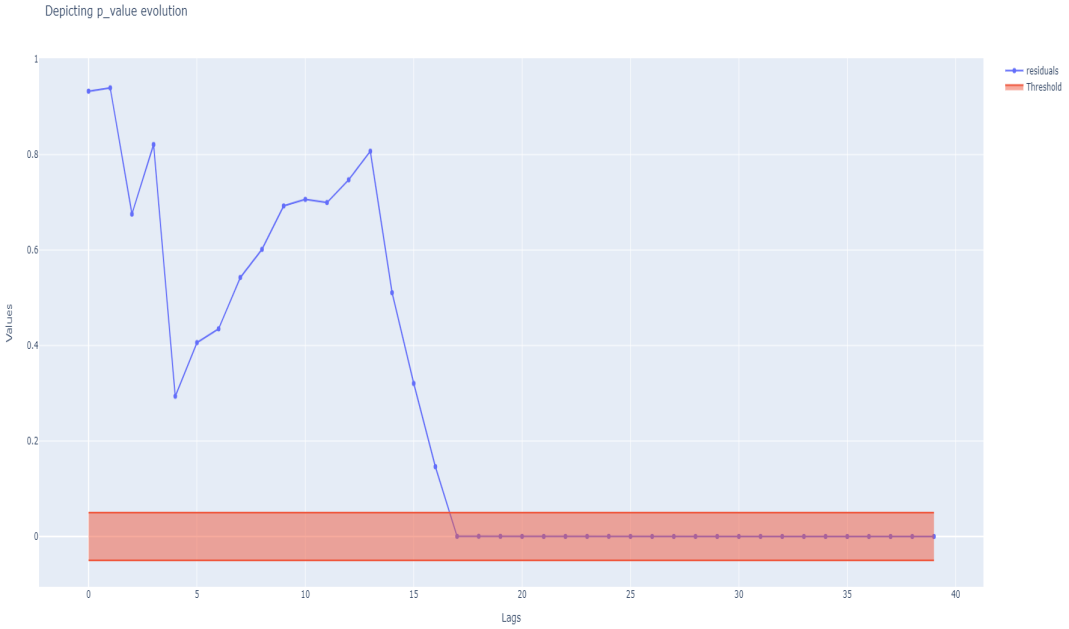
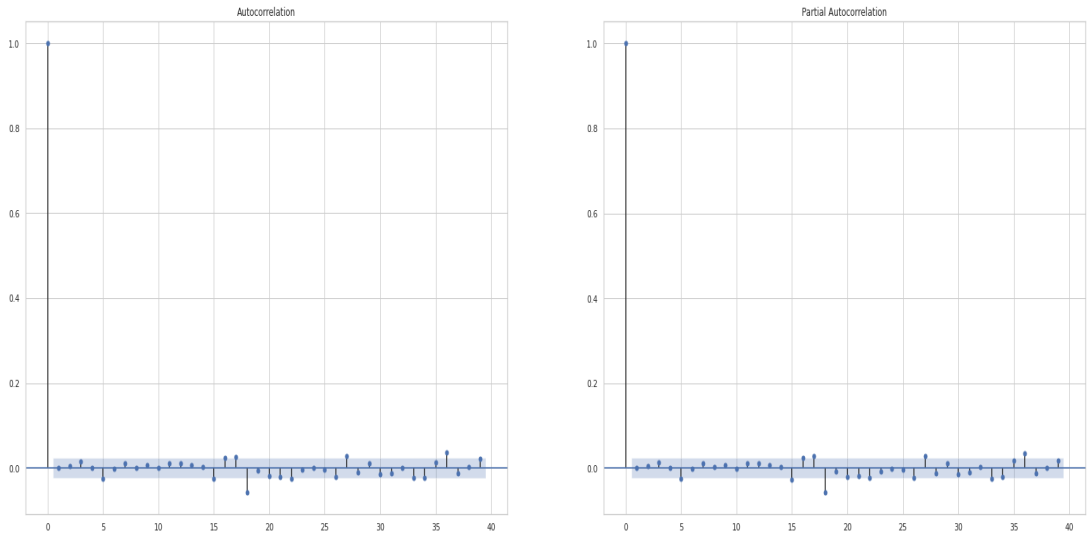


FIGURE 4.14: Evolution du p-value du Ljung-Box test (Threshold = 0.05)

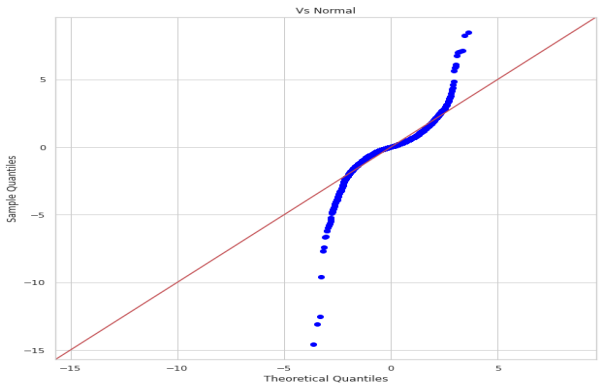
peut confirmer ce résultats par l'observation des plots ACF et PACF.

4.2.3 Autocorrelatioon du Résiduel



Il n'y a pratiquement pas d'autocorrélation du résidu. Il s'agit alors d'un processus de bruit blanc.

4.2.4 Q-Q Plot du Résidu et Test de Normalité de Shapiro-Wilk




```
Shapiro-Wilk Statistic :  0.839

p-value :  0.0   p-value < 0.05 : the data is not normal
```

4.2.5 Réalisation du test de l'effet ARCH A.K.A Test d'Engle pour l'hétéroscédasticité conditionnelle autorégressive

Le test ARCH/Engle est construit suivant le fait que si les résidus sont hétéroscédastiques ; c'est à dire qu'ils ont une variance variable ou irrégulière tout au long de la série. Alors, les résidus au carré sont autocorrélés. Il s'agit du test du multiplicateur de Lagrange qui consiste à ajuster un modèle de régression linéaire pour les résidus quadratiques et à examiner si le modèle ajusté est significatif. L'hypothèse nulle est donc que les résidus quadratiques sont une séquence de bruit blanc, c'est-à-dire que les résidus sont homoscédastiques.

L'effet ARCH est lié à la volatilité conditionnelle variable dans le temps, ainsi :

$$\sigma_t^2 = E\left[(x_t - \bar{x}_t)^2\right] = E\left[x_t^2\right] - \bar{x}_t^2$$

Où :

σ_t^2 : variance conditionnelle

$\bar{x}_t$: moyenne conditionnelle

En supposant que la série temporelle n'a pas de moyenne significative alors la variance conditionnelle s'exprime comme suit :

$$\sigma_t^2 = E\left[(x_t - \bar{x}_t)^2\right] = E\left[x_t^2\right] = E\left[y_t\right] \approx x_t^2$$

Si l'on suppose que la série temporelle quadratique (yt) est autocorrélée, alors la volatilité conditionnelle (t) varie dans le temps et présente un phénomène de regroupement (par exemple, des périodes d'oscillations suivies de périodes de calme relatif).

Après avoir appliqué ce test, on obtient les résultats suivants :

```
Lagrange multiplier test statistic :  2620.31

p-value :  0.0   p-value < 0.05 : Les résidus quadratiques présentent une hétéroscédasticité.
```

Conclusion : le résidu de modèle ARIMA(6, 1, 6) est une séquence de bruit blanc stationnaire qui ne suit pas une distribution normale et qui présente une hétéroscédasticité. On peut alors commencer la modélisation de la volatilité à l'aide du modèle ARCH et GARCH.

4.3 Modélisation de la volatilité (ARCH & GARCH)

4.3.1 Corrélation des résidus quadratiques

La volatilité peut être détectée en regardant le corrélogramme des valeurs quadratiques puisqu'elles sont équivalentes à la variance, à condition que la série soit ajustée pour avoir une moyenne de zéro.

On constate d'après les plots ACF et PACF que la variance est en fait assez autocorrélée, ce qui signifie qu'elle est dépendante du temps.

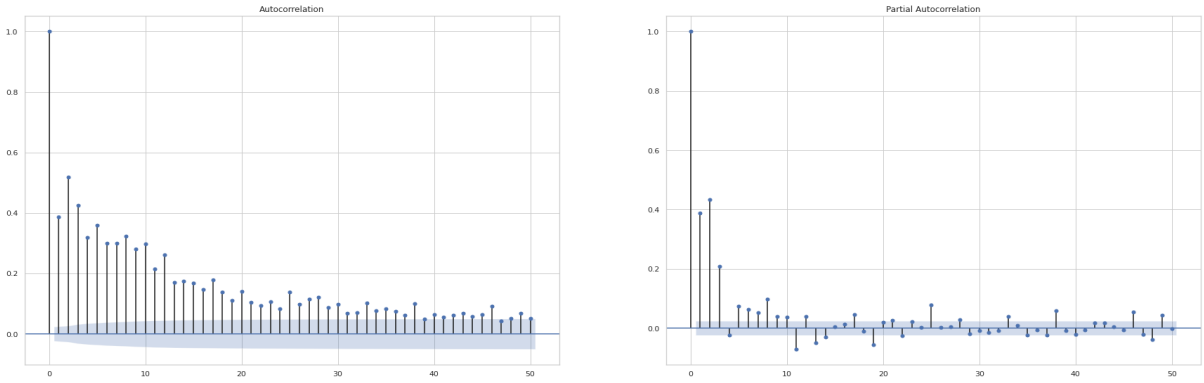


FIGURE 4.15: ACF/PACF des résidus quadratiques

4.3.2 Modèle Autoregressive Conditional Heteroskedasticity "ARCH"

ARCH est une méthode qui modélise explicitement la variation de la variance au cours du temps dans une série temporelle. Les modèles ARCH sont utilisés pour décrire une variance volatile. Bien qu'un modèle ARCH puisse éventuellement être utilisé pour décrire une variance qui augmente progressivement dans le temps, il est le plus souvent utilisé dans des situations dans lesquelles on a de courtes périodes de variation accrue.

Les modèles ARCH ont été créés dans le contexte de problèmes économétriques et financiers liés à la quantité d'investissements ou d'actions qui augmentent (ou diminuent) périodiquement. On a donc tendance à les décrire comme des modèles pour ce type de variable.

Supposons que nous modélisions la variance d'une série yt . Sous le modèle ARCH(1), la variance de la série yt qui est conditionnelle à yt-1 s'écrit comme suit :

$$\mathrm{Var}(y_t|y_{t-1}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2$$

La variance à l'instant t est liée à la valeur de la série à l'instant t-1. Une valeur relativement grande de $yt - 1^2$ donne une valeur relativement grande de la variance au temps t.

Si nous supposons que la série a une moyenne nulle, le modèle ARCH(1) pourrait

s'écrire comme suit :

$$(2)\ y_t = \sigma_t \epsilon_t,$$

$$\text{with } \sigma_t = \sqrt{\alpha_0 + \alpha_1 y_{t-1}^2},$$

$$\text{and } \epsilon_t \overset{iid}{\sim} (\mu = 0, \sigma^2 = 1)$$

FIGURE 4.16: ACF/PACF des résidus quadratiques

epsilon t suit une loi normale

Un processus ARCH(m) est un processus pour lequel la variance à l'instant t est conditionnelle aux observations des m instants précédents :

$$\mathrm{Var}(y_t | y_{t-1}, \ldots, y_{t-m}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \cdots + \alpha_m y_{t-m}^2.$$

Son paramètre - m : Le nombre de décalages des observations quadratique à inclure dans le modèle ARCH.

```
                      Zero Mean - ARCH Model Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                       ARCH   Log-Likelihood:               -12114.9
Distribution:                  Normal   AIC:                           24237.8
Method:            Maximum Likelihood   BIC:                           24265.2
                                        No. Observations:                 7017
Date:                Tue, Apr 27 2021   Df Residuals:                     7017
Time:                        23:31:44   Df Model:                            0
                          Volatility Model
==============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
omega          0.5564  4.951e-02     11.237  2.681e-29 [  0.459,  0.653]
alpha[1]       0.2753  3.531e-02      7.797  6.317e-15 [  0.206,  0.345]
alpha[2]       0.3835  3.643e-02     10.526  6.554e-26 [  0.312,  0.455]
alpha[3]       0.3280  3.988e-02      8.223  1.987e-16 [  0.250,  0.406]
==============================================================================
```

FIGURE 4.17: Résumé du modèle ARCH(3)

Après avoir appliqué le modèle pour plusieurs valeurs de p, il s'est avéré que p = 3 est le choix optimal.

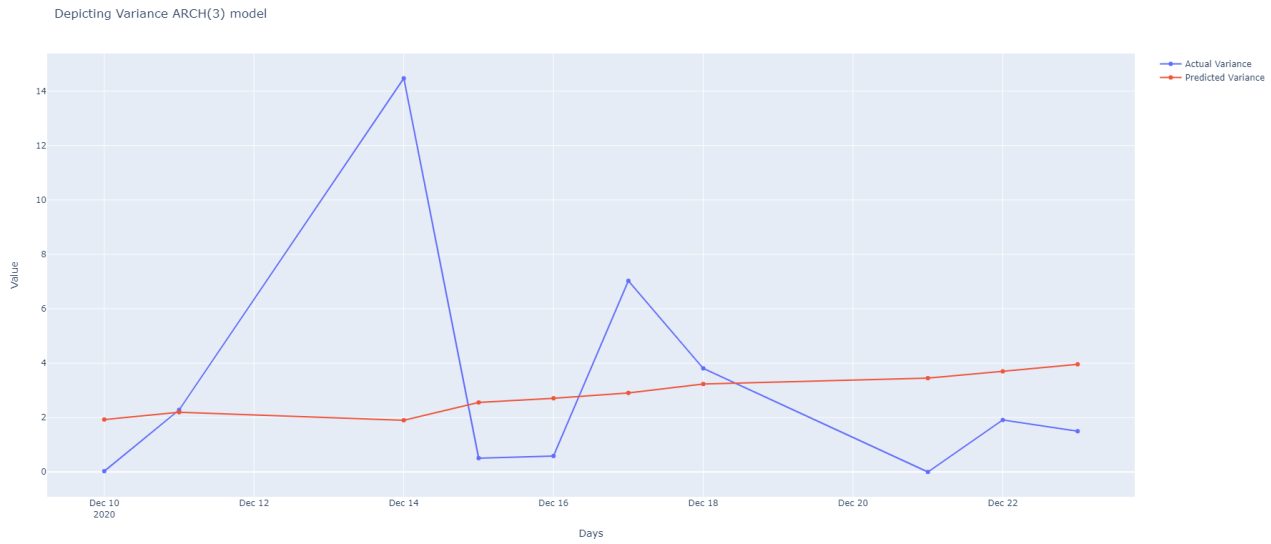


FIGURE 4.18: Prédiction de la variance avec ARCH(3)

4.3.3 Modèle Generalized Autoregressive Conditional Heteroskedasticity "GARCH"

GARCH est une extension du modèle ARCH qui intègre une composante moyenne mobile en plus de la composante autorégressive. Dans le processus ARCH(p), la variance conditionnelle est spécifiée comme une fonction linéaire des variances passées de l'échantillon uniquement, alors que le processus GARCH(p, q) permet aux variances conditionnelles retardées d'entrer également dans le modèle. Les modèles GARCH englobe les modèles ARCH, où un GARCH(p, 0) est équivalent à un modèle ARCH(p). Supposons que nous modélisions la variance d'une série yt . Sous le modèle GARCH(1, 1), la variance de la série yt qui est conditionnelle à yt-1 et sigma(t-1) s'écrit comme suit :

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

Si nous supposons que la série a une moyenne nulle, le modèle ARCH(1) pourrait

s'écrire comme suit :

$$y_t = \sigma_t.\epsilon_t \tag{4}$$

avec $\sigma_t = \sqrt{\alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1.\sigma_{t-1}^2}$

et ϵ_t *suit une loi normale* $(\mu = 0, \sigma^2 = 1)$

Un processus GARCH(m, n) est un processus pour lequel la variance à l'instant t est conditionnelle aux observations des m instants précédents :

$Var(y_t \mid y_{t-1}, ..., y_{t-m}, \sigma_{t-1}, \sigma_{t-m}) = \sigma_t^2$

$\sigma_t^2 = \alpha_0 + \alpha_1.y_{t-1} + .. + \alpha_m.y_{t-m} + \beta_1.\sigma_{t-1} + .. + \beta_n.\sigma_{t-n}$

Paramètres :

m : Le nombre de décalages des observations quadratique à inclure dans le modèle GARCH.

n : Le nombre de décalages des variances à inclure dans le modèle GARCH.

```
                      Zero Mean - GARCH Model Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Mean Model:                 Zero Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:               -11304.8
Distribution:                  Normal   AIC:                           22619.5
Method:            Maximum Likelihood   BIC:                           22653.8
                                        No. Observations:                 7017
Date:                Wed, Apr 28 2021   Df Residuals:                     7017
Time:                        00:09:17   Df Model:                            0
                            Volatility Model
==============================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
omega      3.4183e-03  1.658e-03      2.062  3.925e-02 [1.684e-04,6.668e-03]
alpha[1]       0.0539  1.226e-02      4.395  1.108e-05 [2.986e-02,7.792e-02]
alpha[2]       0.0700  1.517e-02      4.616  3.909e-06 [4.030e-02,9.978e-02]
beta[1]        0.1859      0.141      1.322      0.186  [-8.976e-02,  0.462]
beta[2]        0.6902      0.130      5.319  1.042e-07     [  0.436,  0.944]
==============================================================================
```

FIGURE 4.19: Résumé du modèle GARCH(2, 2)

Après avoir appliqué le modèle pour plusieurs valeurs de m et n, il s'est avéré

que m = n = 2 est le choix optimal.

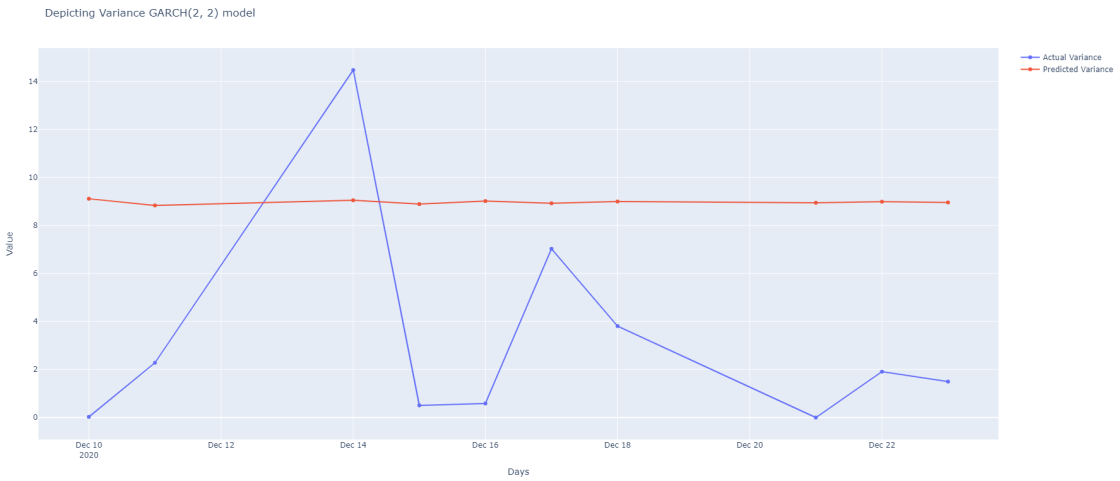
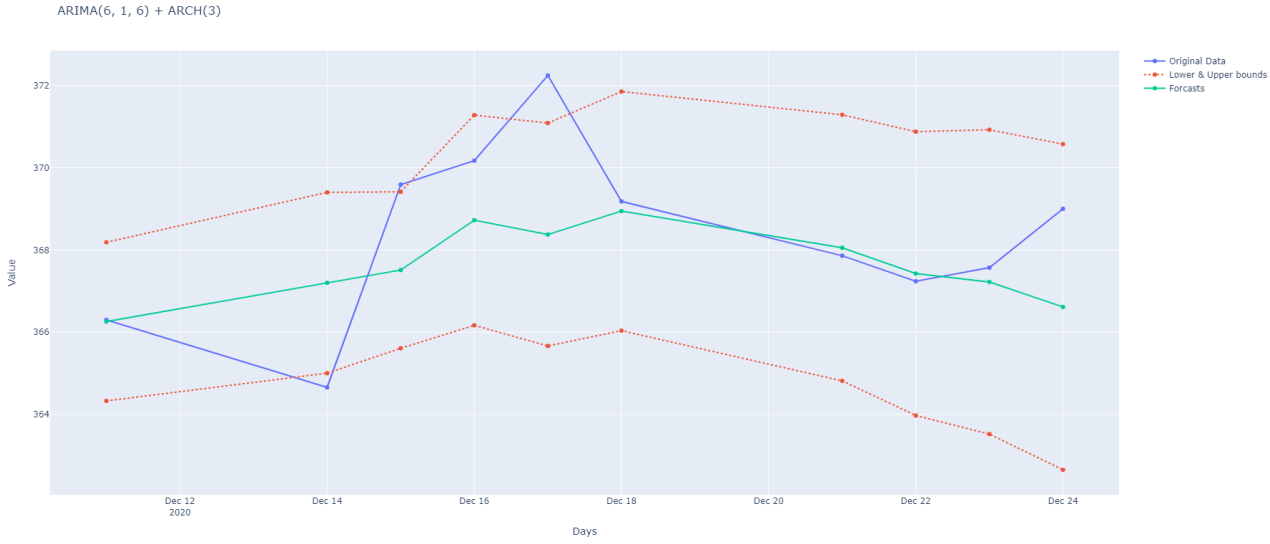


FIGURE 4.20: Prédiction de la variance avec GARCH(2, 2)

Dans la section suivante on combinera le modèle ARIMA(6, 1, 6) et ARCH(3) vu que ce dernier présente le plus petit RMSE.

4.4 Combinaison des modèles ARIMA(6, 1, 6) et ARCH(3)



Nous avons réalisé la combinaison des modèles ARIMA(6, 1, 6) et ARCH(3) en ajoutant les intervalles de confiances de la variance résiduelle extraite du modèle ARCH(3).

Chapitre 5

Conclusion

Il est évident que l'indice S&SP500 présente énormément d'éléments à traiter. Celui-ci, comme sus-mentionné, présente des caractéristiques qui représentent parfaitement le comportement d'un indice financier, i.e. d'une série financière. Nous avons essayé de couvrir divers aspects que ce soit au niveau de l'analyse ainsi que de la prédiction, avec divers éléments d'ouverture que nous pourrons explorer ultérieurement. Nous avons enormément appris à lier les reéultats matheéatiques et statistiques à des interprétations financières, ce que nous estimons va nous être d'une extrême utilité étant donné nos aspirations de carrière en finance. Nous tenons aussi à remercier messieurs Khalil Said et Saad Benjelloun pour leur accompagnement et leur partage.

Bibliographie

[1] - towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775

[2] - medium.com/open-machine-learning-course/open-machine-learning-course-topic-9-time-series-analysis-in-python-a270cb05e0b3

[3] - www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm

[4] - www.sciencedirect.com/topics/agricultural-and-biological-sciences/time-series-analysis

[5] - www.aptech.com/blog/introduction-to-the-fundamentals-of-time-series-data-and-analysis/

[6] - www.espon.eu/sites/default/files/attachments/TR$_{T}ime_{S}eries_{j}une$2012.pdf

[7] - analyticsindiamag.com/hands-on-guide-to-time-series-analysis-using-simple-exponential-smoothing-in-python/

[8] - machinelearningmastery.com/exponential-smoothing-for-time-series-forecasting-in-python/

[9] - towardsdatascience.com/time-series-in-python-exponential-smoothing-and-arima-processes-2c67f2a52788

[10] - www.statsmodels.org/stable/examples/notebooks/generated/exponential$_{s}moothing.html$

[11] - www.analyticsvidhya.com/blog/2020/10/how-to-create-an-arima-model-for-time-series-forecasting-in-python/

[12] - www.statsmodels.org/stable/examples/notebooks/generated/tsa$_{a}rma_{0}.html$

[13] - towardsdatascience.com/advanced-time-series-analysis-with-arma-and-

arima-a7d9b589ed6d

[14] - machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

[15] - towardsdatascience.com/time-series-forecasting-with-sarima-in-python-cda5b793977b

Annexe A1

Appendix 1 - Modèle ARIMA, Son Optimisation et ARIMA(6,1,6)-ARCH(3)

```
# -*- coding: utf-8 -*-
"""Highlights.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1
    li90VVaAJgDKTPjLJI0nKMtW3xlp79t1
"""

from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.arima_process import ArmaProcess
from statsmodels.tsa.arima_model import ARIMA

def ARIMA_model(series, p, d, q, n_test, x_axis, plot_last, summary =
    False, if_plot = True, apply_log = False) :
  if apply_log :
    train_set = [obs for obs in np.log(series[:len(series) - n_test])]
    test_set = [obs for obs in np.log(series[len(series) - n_test:])]

```

```
else :
  train_set = [obs for obs in series[:len(series) - n_test]]
  test_set = [obs for obs in series[len(series) - n_test:]]

model = ARIMA(endog = train_set, order = (p, d, q))
model_fit = model.fit()

train_prediction = model_fit.predict(start = 1, end = len(series) -
 n_test, typ = 'levels')


train = train_set
train.append(float(model_fit.forecast()[0]))
test_forcast = []

for i in range(len(test_set)) :
  forcast_model = ARIMA(endog = train, order = (p, d, q))
  forcast_model_fit = forcast_model.fit()

  test_forcast.append(float(forcast_model_fit.forecast()[0]))
  train.append(float(forcast_model_fit.forecast()[0]))

train_set = [obs for obs in series[:len(series) - n_test]]
test_set = [obs for obs in series[len(series) - n_test:]]

if apply_log :
  train_prediction = np.exp(train_prediction)
  test_forcast = np.exp(test_forcast)

rmse_prediction = np.sqrt(MSE(y_true = train_set, y_pred =
 train_prediction))
rmse_forcast = np.sqrt(MSE(y_true = test_set, y_pred = test_forcast))

if if_plot :
  plot_original = go.Scatter(x = x_axis[len(series) - plot_last:], y
```

```
    = series[len(series) - plot_last:], mode = "markers + lines", name =
     'Original Data')
52   plot_prediction = go.Scatter(x = x_axis[len(series) - plot_last:],
    y = train_prediction[len(series) - plot_last:], mode = "markers +
    lines", name = 'Predictions')
53   plot_forcast = go.Scatter(x = x_axis[len(series) - n_test:], y =
    test_forcast, mode = "markers + lines", name = 'Forcasts')
54
55   fig = go.Figure(data = [plot_original, plot_prediction,
    plot_forcast])
56   fig.update_layout(height = 800, width = 1800, title_text = "ARIMA -
     (p=%s, d=%s, q=%s)" % (p, d, q), xaxis_title = 'Days', yaxis_title
    = 'Value')
57   fig.show()
58
59   print('\n''RMSE Prediction :', round(rmse_prediction, 3))
60   print('\n''RMSE Forcast :', round(rmse_forcast, 3))
61
62  else :
63    return rmse_forcast, test_forcast
64
65  if summary :
66    model_fit.summary()
67
68 # Optimisons p et q en utilisant les corr[U+FFFD]logrammes :
69 ps = [0, 1, 6, 7, 8, 9]
70 qs = [0, 1, 6, 7, 8, 9]
71
72 RMSE_OPT, failed_for = Manual_Opt_ARIMA(series = spy.Close, ps = ps, d
    = 1, qs = qs, n_test = 10)
73 print('Failed Conversion For : ', failed_for, '\n')
74
75 RMSE_OPT
76
77 RMSE_OPT[RMSE_OPT.RMSE_Forecast == min(RMSE_OPT.RMSE_Forecast)]
```

```
78
79 # Optimisation [U+FFFD] l'aide de l'AIC :
80 ps = list(range(0, 9))
81 d = 1
82 qs = list(range(0, 9))
83
84 AIC_OPT = AIC_Opt_ARIMA(series = spy.Close, ps = ps, d = d, qs = qs,
       n_test = 10)
85
86 AIC_OPT
87
88 AIC_OPT[AIC_OPT.AIC == min(AIC_OPT.AIC)]
89
90 """**Best Model : ARIMA(6, 1, 6) + ARCH(3)**"""
91
92 ARIMA_forecasts = ARIMA_model(series = spy.Close, p = 6, d = 1, q = 6,
       n_test = 10, x_axis = spy.Date, plot_last = 365, if_plot = False)[1]
93
94 series = spy.Close
95 x_axis = spy.Date
96 plot_last = 365
97
98 n_test = 10
99
100 pred_arch = arch_prediction.residual_variance.values[-1, :]
101
102 upper_bound = ARIMA_forecasts + pred_arch
103 lower_bound = ARIMA_forecasts - pred_arch
104
105 plot_original = go.Scatter(x = x_axis[len(series) - n_test:], y =
       series[len(series) - n_test:], mode = "markers + lines", name = '
       Original Data')
106 plot_forcast = go.Scatter(x = x_axis[len(series) - n_test:], y =
       ARIMA_forecasts, mode = "markers + lines", name = 'Forcasts')
107
```

```
plot_lower_bound = go.Scatter(x = x_axis[len(series) - n_test:], y = lower_bound, line = dict(dash = 'dot', color = "#EF553B"), name = 'Lower & Upper bounds')
plot_upper_bound = go.Scatter(x = x_axis[len(series) - n_test:], y = upper_bound, line = dict(dash = 'dot', color = "#EF553B"), showlegend = False)


fig = go.Figure(data = [plot_original , plot_lower_bound, plot_forcast, plot_upper_bound])
fig.update_layout(height = 800, width = 1800, title_text = "ARIMA(6, 1, 6) + ARCH(3)", xaxis_title = 'Days', yaxis_title = 'Value')
fig.show()
```