

# Physique des marchés, TP1 : faits stylisés

Il est fortement conseillé d'utiliser des notebooks, jupyter pour Python, rmarkdown Rstudio pour R. Un notebook bien commenté est accepté comme rendu. Le notebook doit contenir le résultat des calculs.

## 1 Buts du TP :

- Manipuler des données financières et caractériser des faits stylisés.
- Utiliser des objets de séries temporelles
- fusionner des séries temporelles asynchrones
- voir cheat sheets [Python]

## 2 Obtenir des données

### 2.1 Données journalières

Procurez-vous des données journalières en utilisant `yfinance` pour Python. Choisissez des titres les plus anciens possibles afin d'avoir le plus de données possible.

### 2.2 Données intraday

1. Voir lien sur le cours. Pour charger et interpréter les fichiers `bbo`, utilisez le code disponible sur le site du cours (`loadbbo.py`)
2. Plus de données sont à votre disposition pour des projets de recherche si nécessaire

## 3 Calcul des rendements

Calculer des log-rendements  $r$  :

- Python : `numpy.log(p).diff()`

### 3.1 Données journalières :

Les log-rendements sont à calculer à partir de la colonne Adjusted Close (si les prix ne sont pas ajustés automatiquement)

### 3.2 Données intraday

- nettoyer les données de sorte à ne conserver que les données de marché lorsqu'ils sont ouverts, i.e., entre 9h00 et 17h30 pour les titres européens, et entre 9h30 et 16h pour les titres américains.
- Ajouter une colonne de prix moyen  $\text{mid}=(\text{bid}+\text{ask})/2$ .
- Les rendements sont à calculer à partir du mid.
- enlever les rendements NA
- calculer les log-rendements à 5 secondes (ou plus) (`resample`)

## 4 Analyse empirique

Les questions suivantes portent sur les données journalières ET intraday a priori, sauf en cas d'impossibilité manifeste.

### 4.1 Rendements : types de distribution

1. Gaussianité : déterminer graphiquement si les rendements des prix intraday  $\text{mid}=(\text{bid}+\text{ask})/2$  et de l'adjusted close des données journalières provenant de Yahoo sont gaussiens avec des qq-plots.
2. Queues lourdes déterminer graphiquement si la distribution de la valeur absolue rendements a une queue lourde :

- (a) tracer la ccdf avec de axes linéaires en X et log en Y. Une convexité est la trace de queues lourdes.
- `from statsmodels.distributions.empirical_distribution import ECDF`
- (b) tracer le qqplot de  $|r|/E(|r|)$  avec une distribution exponentielle de moyenne 1 comme distribution théorique. Commentez.
- (c) calculer la différence maximale entre les quantiles empiriques de  $|r|/E(|r|)$  et les quantiles d'une distribution exponentielle de moyenne 1 pour les rendements plus extrêmes que les rendements exponentiels, i.e.  $\max_q[q(\phi) - q_{exp}(\phi)]_+, \quad \phi \in ]0, 1[, \text{ où } [x]_+ = \max(0, x).$
- (d) générez des rendements iid avec une loi de Laplace (double exponentielle) et répétez le point précédents. Commentez la différence.
3. Puisque vous trouvez des queues lourdes, déterminer l'exposant de la queue de distribution avec le paquet `powerlaw` pour Python . Il est conseillé de se référer au [tutoriel] et d'utiliser la fonction qui correspond à une distribution avec valeurs continues. Notez que ce paquet fournit à la fois abscisse de début de la loi de puissance, l'exposant de la loi de puissance et permet de produire des figures qui superposent les données et la calibration.
4. Utilisez le paquet `powerlaw` pour tester la présence d'une queue de type loi de puissance contre les alternatives suivantes : loi exponentielle (heavy tails ?) et loi log-normale (autre type de queue lourde).

#### Notes techniques

- Vous pouvez supposer que  $P(r) = P(-r)$ , et donc ne calculer que  $P(|r| > x)$
- en Python, `myecdf=ECDF(|r|)` est un object qui peut s'appliquer comme fonction ; il contient également `myecdf.x` et `myecdf.y` utilisables directement dans une fonction `plot`.

## 4.2 Distribution des rendements dans les marchés en périodes de hausse et de baisse

Pour les titres suivants (au moins), ^GSPC, AMD, MSFT, INTC, et C, depuis 1900-01-01, en données journalières

1. Trouvez les maxima et minima locaux (par exemple avec `scipy.signal.find_peaks` ; prêtez attention aux paramètres de cette méthode). Ajustez les paramètres afin que le nombre de pics ne soit pas trop grand (typiquement quelque-uns par an)
2. Définissez une période haussière comme celle entre un minimum et un maximum local, et réciproquement pour une période baissière.
3. Tracez les CCDFs de chaque période et déterminer si les rendements ont une distribution à queue plus grasse durant les périodes baissières avec des méthodes statistiques.
4. Commentez les différences entre les titres et tirez-en une conclusion générale.

## 4.3 Disparition de la prévisibilité

1. Pour 3 titres suffisamment différents, dont une cryptomonnaie, appliquer la stratégie qui consiste à prendre une position du même signe que celui du rendement précédent. Tracer sa performance au cours du temps.
2. Séparer la performance long et short. Est-elle significativement différente ? Commentez.