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, au format html. Sinon, rapport en pdf.

1 Buts du TP :

- Manipuler des données financières et caractériser des faits stylisés.
- Utiliser des objets de séries temporelles (R : `xts`, Python : `pandas timeseries`) afin de pouvoir manipuler les séries temporelles plus facilement. Avec un objet de séries temporelles X, on peut
  - sélectionner des dates
    - `R: X['2015'], X['2015-01'], X['2015-01-01'], X['2015/2016']`
    - `Pandas: X['2014-07-04'], X['2014-07-04':'2015-07-04']`
  - sélectionner une plage horaire pour tous les jours
    - `R: X['T08:09:10/T12:34:45']`
    - `Pandas: X.between_time('08:09:10','12:34:45')`
  - effectuer des opérations arithmétiques habituelles (+,-,*,/, etc)
  - fusionner des séries temporelles asynchrones
  - voir cheat sheets [R], [Python]

2 Obtenir des données

2.1 Données journalières

Sauvez quelques fichiers de données obtenus ou

1. Automatiquement : quantmod pour R, yfinance pour Python

2. Manuellement (pour l'exemple)http://finance.yahoo.com : pour un titre donné, entrez son symbole (Apple ≡ AAPL), cliquer sur 'historical prices', puis 'download to spreadsheet'. Cela vous produit un fichier .csv lisible depuis R (read.csv). Notez que l'URL peut facilement être modifiée pour télécharger d'autres symboles (p.e. IBM, INTC, MSFT, A).

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.R et 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 :
- `R: diff(log(price))`
- Python : `numpy.log(p).diff()`

3.1 Données journalières :

Les log-rendements sont à calculer à partir de la colonne AdjustedClose.

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 mid=(bid+ask)/2.
— Les rendements sont à calculer à partir du mid.
— enlever les rendements NA (`na.omit`)
— calculer les log-rendements à 5 secondes (ou plus)
  — Python : `logprix.resample('5S').last().diff()`
  — R : `diff(to_seconds(logprix,5,OHLC=FALSE))`

4 Analyse empirique

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

4.1 Rendements : distribution

1. Déterminer graphiquement si les rendements des prix intraday mid=(bid+ask)/2 et de l'adjusted close des données journalières provenant de Yahoo sont gaussiens avec des qq-plots.
2. 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 concavité est la trace de queues lourdes.
     — R : `myecdf=ecdf(abs(r))` est une fonction, on doit l'appliquer e.g. à `sort(abs(r))`
     — Python: `from statsmodels.distributions.empirical_distribution import ECDF`
   (b) calculer la distance maximale entre les quantiles empiriques de $|r|/E(|r|)$ et les quantiles d'une distribution exponentielle de moyenne 1.
3. Si vous voyez des queues lourdes, déterminer l'exposant de la queue de distribution avec le paquet `poweRlaw` pour R, ou `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.

Notes techniques
— Vous pouvez supposer que $P(r) = P(-r)$, et donc ne calculer que $P(|return| > |r|)$
— Il arrive que des fonctions de R ne sachent pas quoi faire de la colonne temporelle des objets `xts`, par exemple `ecdf`. Il faut dont les utiliser sur les données débarrassées de leur attribut temporel : `r_c=coredata(r)` au lieu de `r`.
— en R, `ecdf(x)` retourne une fonction, qu'il faut donc appliquer à des abscisses, donc par exemple, avec `x=sort(abs(r_c))`
`plot(x,1-ecdf(x)(x),log='xy',t='s')` pour tracer $P(return > r)$, ou définir
`myecdf=ecdf(x)` et tracer
`plot(x,1-myecdf(x),log='xy',t='s')`
— en Python, myecdf=ECDF($|r|$) est également un object qui peut s'appliquer comme fonction ; cela dit, myecdf.x et myecdf.y sont utiles.

4.2 Autocorrélation

1. Tracer les courbes d'autocorrélation des rendements du mid à une certaine échelle (utiliser `to.seconds()`) en R et .resample() en Python
2. Tracer la courbe d'auto-corrélation de la valeur absolue des rendements avec des axes log-log avec temps de lag maximal suffisamment grand pour les rendements journaliers (par exemple 256). Commentez.

4.3 Mesures de volatilité

1. Estimateur de variation quadratique réalisée du prix S_t :

$$V = A \times \frac{1}{T} \sum_{i=1}^{T} \ln \left(\frac{S_{t-i+1}}{S_{t-i}} \right)^2,$$

où A est le facteur d'annualisation : en supposant que les prix sont diffusifs et que la résolution temporelle des rendements est de δt (en unité de temps), alors

$$A = \text{nombre d'unités de temps en une année de trading,}$$

par exemple 252 pour des données journalières (nombre moyens de jours d'ouverture des marchés américains).

2. Estimateur de Garman Klass (1980), à annualiser de la même manière (H : high, L : low, O : open, C : close)

$$V = A \times \frac{1}{T} \sum_{t=1}^{T} \frac{1}{2} \left[\ln \frac{H_t}{L_t} \right]^2 - (2 \ln 2 - 1) \left[\ln \frac{C_t}{O_t} \right]^2$$

3. Comparer les valeurs entre les deux estimateurs. Que constatez-vous ?

Note : pour les données intraday, aligner les données par exemple toutes les 5 minutes avec la fonction `to.minutes(5) de xts` qui retourne les valeurs O,H,L,C. Pour Python, la chaîne .resample('5min').ohlc() retourne également les valeurs OHLC.

4.4 Estimation du nombre de sauts

Téléchargez le fichier de données ES[...].parquet sur le dossier partagé du cours. Calculer les log-rendements du prix mid. du tableau qui contient des quotations à 1 minute du future ES mini sur le S&P500 entre 1997 et 2017.

— Ce fichier est en format parquet, qui est le meilleur format pour les tables à l'heure actuelle. Pour charger les données
  — Python : installer `fastparquet`, et utiliser `pandas.read_parquet('datafile.parquet')`
  — R : install arrow, et utiliser `as.xts(as.data.table(read_parquet('datafile.parquet')))`.

1. Calculer la moyenne mobile à 30 minutes de la valeur absolue des log-rendements. Cela donne une estimation de la volatilité récente.
   Python : `rolling('30min').mean()`.
   R : utiliser la fonction `roll_mean()` de la bibliothèque `roll`.
2. Comparer cette moyenne mobile avec la valeur absolue du rendement suivant. Il est plus facile de comparer la valeur courante de la valeur absolue du rendement avec la moyenne mobile décalée.
   Python : .shift()
   R : lag()

   Fixer un seuil $s = 1, \cdots, 20$, et pour chaque valeur de s, compter la fraction $\phi(s)$ de rendements en valeur absolue > moyenne mobile.
3. Tracer $\phi(s)$ en fonction de s. Est-ce que vous obtenez une loi de puissance ? Quel est l'exposant de queue ? Est-il comparable avec une des valeurs rapportée dans la littérature ?