

Projet 2L : Trading à backtester

Vous avez à disposition un fichier csv nommé BTC.csv donnant :

- la date (ici des mesures faites heure par heure)
- la valeur du bitcoin au début de l'heure
- la valeur du bitcoin au plus haut dans l'heure
- la valeur du bitcoin au plus bas dans l'heure
- la valeur du bitcoin à la fin de l'heure
- le volume (caractérisant le nombre de pièces cryptographiques vendues)

1. Ecrire une fonction permettant de récupérer :

- une liste des dates
- une liste des valeurs du bitcoin en fin d'heure

2. La date est donnée en timestamp et nécessite une conversion (le timestamp se définit comme le système d'horodatage qui donne le nombre de temps en ms qui s'écoule depuis le 1er janvier 1970 à 00 h). Pour effectuer cette conversion, nous prendrons une année équivalente à 365.25 jours et un mois à 30,4375 jours. Ainsi 1502942400000 devient le 15/7/2017 à 4 H. Ecrire une fonction conversion qui prend en entier en timestamp et renvoie la date associée.

Une moyenne mobile représente simplement la moyenne du prix sur les n dernières périodes.

La stratégie du backtester consiste à acheter lorsqu'une moyenne mobile à court terme (par exemple 50 périodes) croise à la hausse une moyenne mobile à long terme (par exemple 200 périodes). Inversement, un signal de vente est déclenché lorsque la moyenne mobile à court terme croise à la baisse la moyenne mobile à long terme

3. Ecrire une fonction moyenne qui prend en argument une liste L et une entier N. Cette fonction renvoie une nouvelle liste L_new dont chaque élément est la moyenne de N éléments successif de L. L_new contient donc N fois moins d'éléments que L

4. Représenter sur le même graphe :

- l'évolution de la valeur du bitcoin en fin d'heure
- l'évolution la valeur moyenne du bitcoin en fin d'heures pour N=50
- l'évolution la valeur moyenne du bitcoin en fin d'heures pour N=200 En abscisse, le temps sera représenté par le numéro de l'échantillon (indice de la liste des dates)

Les premières valeurs des moyennes glissantes n'existant pas, on pourra les remplacer par des valeurs None

5. Ecrire une fonction croissante qui prend en argument une liste L et un indice i et qui renvoie True si cette liste a tendance à croître autour de l'indice i

6. Ecrire une fonction décroissante qui prend en argument une liste L et un indice i et qui renvoie `True` si cette liste a tendance à décroître autour de l'indice i

7. Ecrire une fonction nommée `achat` qui prend en argument deux listes $L1$ et $L2$ ainsi qu'un scalaire e . Cette fonction renvoie la liste des indices i pour lesquels:
 - $L1[i] \approx L2[i]$ avec une tolérance de e
 - Et $L1$ et $L2$ croissante

8. Ecrire une fonction nommée `vente` qui prend en argument deux listes $L1$ et $L2$ ainsi qu'un scalaire e . Cette fonction renvoie la liste des indices i pour lesquels:
 - $L1[i] \approx L2[i]$ avec une tolérance de e
 - Et $L1$ et $L2$ décroissante

9. En déduire les temps pour lesquels il fallait acheter si on prend une tolérance de 0.02 une moyenne sur 50 et 200 valeurs. Placer des points sur le graphe précédent et commenter la pertinence de cette méthode de trading