TP 8 : Modèle de Black-Scholes discret, cours du Bitcoin

Ce TP cherche à mettre en application les notions de modèles, de simulation et de vraisemblance au service de données réelles. Ici, on introduit un modèle phare en mathématiques financières, puis on applique les résultats théoriques à l'analyse et la prédiction du cours du Bitcoin.

Première partie : Un modèle probabiliste pour le prix d'un actif. Le modèle de Black-Scholes permet d'expliquer la variation du cours du prix d'un actif dans le temps. Dans sa version en temps discret, le modèle suppose que le rendement de l'actif entre t et $t+1$ est égal à une tendance b (que nous prendrons ici constante dans le temps), à laquelle s'ajoute un bruit σZ, où Z est une gaussienne centrée réduite et $\sigma > 0$ la volatilité (là aussi supposée constante).

On initialise le prix au temps 0 à S_0. Puis, en notant S_t le prix de l'actif à l'instant t, on a donc dans ce modèle,

$$\forall t > 0,\ \frac{S_{t+1} - S_t}{S_t} = b + \sigma Z_t,$$

où les $(Z_t)_{t>0}$ sont des gaussiennes centrées réduites indépendantes.

1. Ce modèle est paramétrique. Quels en sont les paramètres ?

2. Ecrire une fonction `traj(T,S0,b,sigma)` qui simule une réalisation de ce modèle pour t variant de 0 à T. Ensuite, pour $T = 500$, $\sigma = 0.01$, $b = 0.005$ et $S_0 = 100$, représenter le prix S_t en fonction de t. Comment varie le prix de l'actif dans ce cas ? Commenter.

3. Refaites tourner le code précédent avec $\sigma = 0.03, \sigma = 0.07$, puis $\sigma = 0.12$. Comment évoluent les courbes ? Commenter.

4. Pour $T = 100$, simuler $N = 50000$ trajectoires pour les paramètres $\sigma = 0.05, b = 0.002$ et $S_0 = 100$. Représenter l'histogramme des valeurs de S_{100}, puis celui des valeurs de $\log S_{100}$. Que remarque-t-on ? Commenter.

5. A l'aide de simulations, répondez à la question suivante : si $b = 0$, $\sigma = 0.15$, quelle est la probabilité qu'un actif avec une valeur initiale de 100 ait diminué d'au moins 50% au temps $T = 100$? Augmenté d'au moins 50 % ? Doublé au moins ?

6. D'après le modèle précédemment décrit, il est facile de vérifier que pour une trajectoire $S_0, \ldots, S_T$, la log-vraisemblance s'écrit

$$\log L(b, \sigma, S_0, \ldots, S_T) = -\frac{T}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^{T} \left(\frac{S_t - S_{t-1}}{S_{t-1}} - b \right)^2$$

Ecrire une fonction `logV(b,sigma,S)` qui renvoie la log-vraisemblance pour une trajectoire S et les paramètres b, σ.

7. D'après le modèle et vos connaissances, quels sont les deux estimateurs naturels de b et de σ ? Pour cette question, aucun calcul n'est attendu.

Deuxième partie : application à l'analyse et à la prédiction du cours du Bitcoin. A présent, nous appliquons le travail précédent à des données réelles : le cours du prix du Bitcoin, leader des cryptomonnaies, créé en 2009. Le fichier 'bitcoin.csv' est mis à votre disposition sur Moodle. Vous le téléchargerez et vous le placerez dans le même dossier que votre fichier .Rmd du TP8. Il suffira ensuite d'exécuter le code suivant pour lire le fichier :

```
data = read.csv("bitcoin.csv")
head(data)
```

```
##         Date    Last Volume    Open    High     Low
## 1 03/22/2021 54577.0    N/A 57558.1 55287.5 53842.1
## 2 03/21/2021 57316.4    N/A 57809.8 57443.4 56357.5
## 3 03/20/2021 57754.5    N/A 58419.7 58583.8 57471.4
## 4 03/19/2021 58296.2    N/A 57773.0 58877.5 57863.0
## 5 03/18/2021 57859.3    N/A 58724.5 57876.5 56305.1
## 6 03/17/2021 58727.6    N/A 55489.5 59489.4 58621.9
```

Ce dataframe retrace l'évolution du prix d'un Bitcoin (en dollars) entre 2020 et 2021. Il a plusieurs colonnes : la date, le prix à la fin de la journée (c'est celui qui nous intéresse dans notre étude), le volume échangé au cours de la journée, le prix en début de journée, ainsi que les valeurs extrêmes sur 24h.

8. Extraire la colonne `Last` de la database (commande `data$Last`), puis représenter l'évolution du cours sur la dernière année. Attention, pour remettre les données dans le bon ordre, on pourra utiliser la fonction `rev`.

9. Calculer les valeurs des estimateurs de la question 7 sur ce jeu de données (on les notera `hatb` et `hatsigma`), puis les afficher. Commenter.

10. On voudrait à présent visualiser la vraisemblance de l'échantillon en fonction des paramètres b et σ. Compléter le code suivant qui crée un tableau `logVs` de taille `size` $\times$ `size` contenant les log-vraisemblances pour un intervalle discrétisé de valeurs de b allant de -0.03 à 0.03, et un intervalle discrétisé de valeurs de σ allant de 0.001 à 0.1. On prendra `size` égal à 500. *Indication : on utilisera le paramètre `length.out` de la fonction `seq`.*

```
###  A DECOMMENTER
# size = 500
# # bs = seq(...) <-- A compléter
# # sigmas = seq(...) <-- A compléter
#
# logVs = matrix(data = replicate(size*size,0),nrow=size,ncol=size) # on crée une
# # logmatrice de taille size*size ne contenant que des zéros
# for (i in 1:size){
#   b = bs[i]
#   for (j in 1:size){
#     sigma = sigmas[j]
#     # <-- A compléter, on remplit logVs[i,j]
#   }
# }
```

11. Le code suivant représente par un code couleur (heatmap) de l'évolution de la log-vraisemblance en fonction de b et de σ. Faites le tourner. Observe-t-on un dessin clair ? A quoi cela est-il du ?

```
###  A DECOMMENTER
# library(ggplot2)
# data = expand.grid(b=bs,sigma=sigmas) # on crée une matrice contenant les
# # valeurs de b croisées avec celles de sigma.
```

```
# data$logV = matrix(logVs,ncol = 1,nrow=size*size) # on ajoute à ce dataframe
# # un champ logV qui correspond aux logvraisemblances précedemment calculées
# ggplot(data, aes(b, sigma, fill= logV)) + geom_tile() # on dessine le heatmap
```

12. Pour y voir plus clair, on opère une renormalisation de la log-vraisemblance.

```
### A DECOMMENTER
# exposant = 30
# renorm.logV = pmax(0,logVs/max(logVs))^exposant
```

Quel est l'effet de cette renormalisation sur les données ? Comment cet effet dépend-il de l'exposant choisi ? Adapter le code de la question 10 afin qu'il affiche le heatmap de cette log-vraisemblance renormalisée. On superposera le point correspondant aux valeurs des estimateurs, en rouge, à l'aide de l'ajout de la forme "+ geom_point".

Commenter ce que vous observez.

13. Pour les valeurs de b et de $sigma$ inférées avec les deux estimateurs, compléter le code suivant qui lance et stocke $N = 5000$ trajectoires qui prolongent les données de 365 jours, en se basant sur la dernière valeur de l'actif, S_{365}.

```
### A DECOMMENTER
# # N = <- à compléter
# S365 = S[365]
# b = hatb
# sigma = hatsigma
# T=365+365
# trajectoires = matrix(replicate(N*T,0),nrow = N, ncol = T)
#
# for (i in 1:N){
#   # trajectoires[i,1:365] = S <- à compléter
#   # trajectoires[i,365:T] = S <- à compléter
# }
```

14. Sur un graphique, compléter le code suivant pour représenter la trajectoire moyenne de l'actif, ainsi que l'intervalle de confiance empirique de prédiction à 95%.

```
### A DECOMMENTER
# T=365+365
# x=1:T
# traj_inf = apply(trajectoires,2,quantile,0.025)
# traj_sup = apply(trajectoires,2,quantile,0.975)
# traj_mean = colMeans(trajectoires)
# # plot(...,lty=3) <- à compléter
# # lines(..., lty = 3) <- à compléter
# # lines(..., lty = 3) <- à compléter
```

Commentez les résultats obtenus.

15. Représenter l'histogramme de la dernière valeur de la trajectoire S_{730}. C'est le prix estimé du Bitcoin dans un an. Donner sa valeur moyenne prédite et l'écart-type prédit.

16. Certains prédisaient en 2020 que le cours du Bitcoin atteindrait probablement 500 000 $ en 2022. Etait-ce une prédiction réaliste ? Pourquoi ces résultats sont-ils à prendre avec grande précaution ?