Méthodes numériques pour le calcul de la VaR

Mohamed Ben Alaya

7 février 2013

1 Introduction élémentaire à Scilab

Scilab (contraction de Scientific Laboratory) est un logiciel libre, développé conjointement par l'INRIA et l'ENPC. Il est téléchargeable gratuitement à partir de

http ://scilabsoft.inria.fr/

C'est un environnement de calcul numérique qui permet d'effectuer rapidement toutes les résolutions et représentations graphiques couramment rencontrées en mathématiques appliquées. Par contre, Scilab n'est pas un logiciel de calcul formel, comme Maple, il peut calculer

$$\det \begin{pmatrix} 1 & 3 \\ -4 & 2 \end{pmatrix}$$

mais pas

$$\det \begin{pmatrix} a & 3 \\ -4 & 2 \end{pmatrix}$$

Scilab ressemble beaucoup à Matlab et il est basé sur le principe que tout calcul, programmation ou tracé graphique peut se faire à partir de matrices rectangulaires. En Scilab, tout est matrice : les scalaires sont des matrices, les vecteurs lignes des matrices, les vecteurs colonnes des matrices.

Les lignes de commande peuvent être directement tapées sous Scilab pour être exécutées immédiatement. Elles peuvent également être écrites dans un fichier `nom_du_fichier.sce`, puis exécutées en tapant sous Scilab `exec("nom_du_fichier.sce","c")`. Dans une ligne de commande, tout ce qui suit // est ignoré, ce qui est utile pour les commentaires. Les commandes que nous proposons sur des lignes successives sont supposées être séparées par des retours-chariots.

1.1 Vecteurs et matrices

```
A=[1,2,3;4,5,6;7,8,9]  // définit une matrice 3X3
```

La variable A est une matrice 3×3. La matrice avec une seule ligne ou une seule colonne est un vecteur et une une matrice 1×1 est un scalaire. Les éléments d'une matrices peuvent être

référencés par leurs indices. Ajouter un point virgule en fin de ligne supprime l'affichage du résultat (le calcul est quand même effectué). Ceci évite les longs défilements à l'écran, et s'avère vite indispensable.

```
x=[-1.3 sqrt(3) 3*4/5];
x(5)=abs(x(-1));
x
```

Noter que la dimension de x a été automatiquement ajustée. Des nouvelles lignes ou des nouvelles colonnes peuvent être ajoutées très facilement. Les dimensions doivent coincider dans l'instruction.

```
 r=[10,11,12];
 B=[A;r]
y=[x,r]
```

On peut extraire une sous matrice d'une matrice.

```
A(:,3) // extrait la 3ème colonne de A
A(2,:) // extrait la 2ème ligne   de A
A(:,2:3) // extrait la sous matrice formée des colonnes de 2 à 3.
```

Scilab Contient de nombreuses expressions prédéfinies pour la manipulation de matrices. Les variables matricielles peuvent être ajoutées $A + B$, soustraites $A - B$ et multipliées $A * B$, à condition de respecter les dimensions. A' désigne la transposée de la matrice A. En plus de ces opérations habituelles, on peut ajouter un scalaire à tous les éléments d'une matrice, les multiplier par un scalaire, faire la somme de tous les éléments d'une matrice.

```
x=[1 2 3 4];
x+1
2*x
sum(x)
```

On peut appliquer une fonction à tous les éléments d'une matrice

```
x=[1 4 9 16];
sqrt(x)
```

On a parfois besoin d'effectuer une opération élément par élément. Les opérations $A. * A$ et $A./A$ désignent la multiplication et la division terme à terme.

```
A
A./A
A.*A
sqrt(ans)
```

Les résultats sont affectés par défaut à la variable `ans` ("answer"), qui contient donc le résultat du dernier calcul non affecté.

1.2 Quelques commandes utiles

Toutes les variables d'une session sont globales et conservées en mémoire.

```
x=ones(1,100);          // rien n'apparaît
x                       // le vecteur x a bien été défini
```

Des erreurs proviennent souvent de confusions avec des noms de variables déjà affectés. Il faut penser à ne pas toujours utiliser les mêmes noms, ou à libérer les variables par clear. Les variables courantes sont accessibles par who et whos.

```
a=[1,2]; A=[1,2;3,4];  // affecte a et A
1+1                    // affecte ans
who                    // toutes les variables
whos()                 // les détails techniques
clear a
who                    // a disparaît
clear
who                    // a, A et ans disparaissent
xbasc()                // efface le contenu de la fenêtre  graphique active
```

La commande `stacksize` permet de connaître la taille de la pile utilisée par Scilab pour stocker les variables. Si cette taille est trop faible, on peut l'ajuster grâce `stacksize(n)` où n est un entier. Pour plus de détails, faire `help stacksize`. L'aide en ligne est appelée par `help`. La commande `apropos` permet de retrouver les rubriques d'aide quand on ignore le nom exact d'une fonction.

```
help help
help apropos
apropos matrix         // rubriques dont le titre contient "matrix"
help matrix            // aide de la fonction "matrix"
```

Pour effectuer une boucle on peut utiliser la commande `for`

```
for k=1:6 // Le sympble : sert à construire un vecteur incrémenté de 1
x(k)=2*k
x(k)=x(k)+1
end
```

On peut indiquer un autre incrément `x=2:0.5:4`. Il y a aussi la boucle `while`

```
y=1 ;
while y<14
y=2*y
end
```

On trace les graphiques grâce à la commande `plot2d`

```
// graphiques
r=rand(1,100)
plot2d(r)
xbasc()
plot2d(1:100,r,-5)
s=rand(1,100);
xbasc()
plot2d(r,s,-2)
```

Il est possible de choisir la couleur et le symbole utilisé. Pour les histogrammes, on utilise la commande `histplot`

```
// histogrammes
xbasc()
histplot(10,r)
xbasc()
histplot(0:0.2:1,r)
```

1.3 Fonctions Scilab

Une fonction est une suite d'instructions, éventuellement paramétrées, écrites dans un fichier `nom_du_fichier.sci`.

```
function [x] = s_normal(s,mu,sigma)
// simulation_normal réelle
// s = nombre de simulations
//--------------------
x=[];
y=rand(2,s);
for i=1:s do ;
x=[x,mu + sigma*sqrt(-2*log(y(1,i)))*cos(2*%pi*y(2,i))];
end;
x=x';
```

Une fonction est chargée en tapant sous Scilab `getf("nom_du_fichier.sce","c")`. La fonction est ensuite appelée sous Scilab par son nom.

```
// chargement de la fonction s_normal
getf("nom_du_fichier.sce","c")
// appel de la fonction s_normal
s_normal(100,-1,5)
```

2 Simulation

La fonction rand(m,n) permet la génération d'une matrice aléatoire $m \times n$. Les coefficients de la matrice générée sont des réalisations de variables aléatoires uniformes sur $[0, 1]$ indépendantes.

2.1 Simulation d'une loi discrète

On considère une variable aléatoire discrète à valeurs dans un ensemble fini E :

$$E = \{x_i, \quad i = 1 \cdots n\} \quad \text{et} \quad \mathbb{P}(X = x_i) = p_i, \quad i = 1 \cdots n.$$

Soit U une variable aléatoire de loi uniforme sur $[0, 1]$. Montrer qu'on obtient une variable aléatoire Y de même loi que X en posant :

$$Y = \sum_{i=1}^{n} x_i \mathbf{1}_{[m_{i-1}, m_i[}(U) \quad \text{avec} \quad m_i = \sum_{j=1}^{i} p_j \quad i = 1 \cdots n \quad \text{et} \quad m_0 = 0.$$

- Simuler un $n-$ échantillon de loi de Bernouilli pour diverses valeurs du paramètre.
- Simuler un $n-$ échantillon de loi uniforme sur $\{1, 2, 3, 4, 5, 6\}$
- Simuler une loi binomiale de taille n et de paramètre p.

Simulation de la loi binomiale $B(n, p)$ Il est parfois plus commode d'utiliser les propriétés particulières de la loi de X lorsqu'elles sont remarquables. On considère une variable aléatoire de loi binomiale de taille n et de paramètre p

$$\mathbb{P}(X = k) = C_n^k p^k (1 - p)^{n-k}.$$

Soient $U_i \;\; i = 1, \cdots, n$ des variables aléatoires uniformes sur $[0, 1]$ indépendantes. Soit

$$Y = \sum_{i=1}^{n} \mathbf{1}_{\{U_i < p\}}.$$

Montrer qu'on obtient une variable aléatoire Y de même loi que X.

2.2 Simulation d'une loi à densité

Soit X une variable aléatoire de fonction de répartition F supposée continue et strictement croissante. A partir de la loi uniforme U on pose $Y = F^{-1}(U)$. Montrer qu'on obtient une variable aléatoire Y de même loi que X.

- Simuler un $n-$échantillon de loi de Cauchy.
- Simuler un $n-$échantillon de loi exponentielle.
- Simuler un $n-$échantillon de densité $3x^2 \mathbf{1}_{[0,1]}$.

Simulation de la loi exponentielle de paramètre λ C'est la loi à densité continue $\lambda e^{-\lambda x} \mathbf{1}_{x \geq 0}$. Montrer que si U est une variable aléatoire de loi uniforme sur $[0, 1]$ alors $X = -\dfrac{1}{\lambda} \log(U)$ suit une loi exponentielle de paramètre λ.

2.3 simulation d'un vecteur gaussien

On veut simuler des variables aléatoires indépendantes suivant la loi gaussienne centrée réduite $\mathcal{N}(0,1)$.

1. Soient R et θ deux variables aléatoires indépendantes. On pose $X = R\cos(\theta)$ et $Y = R\sin(\theta)$. Montrer que si R^2 suit une loi exponentielle de paramètre $\frac{1}{2}$ et θ est uniformément répartie sur $[0, 2\pi]$, alors (X, Y) a pour densité $\frac{1}{2\pi}e^{-\frac{x^2+y^2}{2}}$.
2. En déduire que si (U_1, U_2) sont deux variables aléatoires indépendantes de loi uniforme sur $[0, 1]$ alors $(\sqrt{-2\log(U_1)}\cos(2\pi U_2), \sqrt{-2\log(U_1)}\sin(2\pi U_2))$ est un couple de variables aléatoires indépendantes de loi $\mathcal{N}(0, 1)$.

On veut maintenant simuler un vecteur gaussien centré de matrice de covariance $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. Montrer que si $\begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$ suit la loi $\mathcal{N}\left(0, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)$ alors $\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix}\begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$ suit la loi $\mathcal{N}\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$. Utilisez le programme suivant pour simuler un vecteur gaussien centré et faites varier ρ.

vecteur_gaussien.sce

```
function [] = gaussian_vector(rho)
  if abs(rho) >1
disp('la corrélation doit être comprise entre -1 et 1!')
disp('aborting...')
return
  end
  xbasc();
  n=1000;
  u1=rand(1,n);
  u2=rand(1,n);
  g1= ... // à compléter
  g2= ... // à compléter
  A=[1,0;rho,sqrt(1-rho^2)];
  z=A*[g1;g2];
  x=z(1,:);
  y=z(2,:);
  plot2d(x,y,-1)
endfunction
```

Remarque La matrice $\mathcal{L} = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix}$ est en fait la décomposition de Cholesky de $M = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ (i.e. $\mathcal{L}\mathcal{L}^T = M$ et $\mathcal{L}$ triangulaire inférieure). Cette méthode permet de simuler un vecteur gaussien dès que l'on a calculé la décomposition de Cholesky de sa matrice de

covariance. Sous Scilab la fonction `chol` permet de calculer la décomposition de Cholesky d'une matrice symétrique. Cette fonction renvoie $\mathcal{L}^T$. Une version de l'algorithme utilisé pour calculer la décomposition de Cholesky $\mathcal{L}$ d'une matrice symétrique positive A est donné par

$$\begin{array}{l} \text{pour } k = 1..size \text{ faire} \\ \quad \mathcal{L}_{k,k} = \sqrt{A_{k,k} - \sum_{j<k} \mathcal{L}_{k,j}^2} \\ \quad \text{pour } i = k+1..size \text{ faire} \\ \quad\quad \mathcal{L}_{i,k} = \dfrac{A_{i,k} - \sum_{j<k} \mathcal{L}_{k,j}\mathcal{L}_{i,j}}{\mathcal{L}_{k,k}} \end{array}$$

2.4 Loi Forte des Grands Nombres

Soit $(Y_1, \ldots, Y_n)$ un n-échantillon de variables aléatoires de loi uniforme sur $[0, 1]$. En regardant l'aide sur la fonction `cumsum` (tapez help cumsum), calculez le vecteur $\bar{Y} = (\bar{Y}_1, \ldots, \bar{Y}_n)$ des moyennes empiriques où $\bar{Y}_i = \frac{1}{i}\sum_{k=1}^{i} Y_k$ et tracez l'évolution de la moyenne empirique $i \longmapsto \bar{Y}_i$ à l'aide de la fonction `plot2d`.

2.5 Théorème de la limite centrale

On considère un n-échantillon $(Z_1, \ldots, Z_n)$ où les variables aléatoires Z_i sont i.i.d de même loi que $\sqrt{12p}\left(\frac{1}{p}\sum_{i=1}^{p} U_i - \frac{1}{2}\right)$, les variables aléatoires $(U_i, i \leq p)$ étant i.i.d de loi uniforme sur $[0, 1]$. Utiliser le programme suivant pour tracer l'histogramme à nc classes de $(Z_i, i \leq n)$. Faites varier n, p, nc. Qu'observez-vous pour $p = 1$, $p = 12$, nc grand et nc petit ? On choisira n de l'ordre de 1000.

TCL.sce

```
function [] = tcl(n, p, nc)
  xbasc();
  X=rand(n,p);
  Z=sqrt(12/p)*(sum(X,'c') - p/2); // somme des colonnes de X, centrage et
                                   // renormalisation
  histplot(nc,Z)
  C=[-5:1/1000:+5];
  plot2d(C,exp(-C.^2/2)/sqrt(2*%pi),3) // densité de la loi N(0,1)
endfunction
```

Exercice : Soit $(Y_1, \ldots, Y_n)$ un n-échantillon i.i.d., à l'aide du théorème de la limite centrale, construire un intervalle de confiance asymptotique pour la moyenne centré en $\bar{Y}_n = \frac{1}{n}\sum_{k=1}^{n} Y_k$ (i.e. de la forme $[\bar{Y}_n - r_n, \bar{Y}_n + r_n]$ où r_n est une fonction des observations). Dans le cas des variables aléatoires de loi uniforme sur $[0, 1]$ calculez le rayon de l'intervalle r_n. Pour $1 \leq i \leq n$, tracez l'évolution de la moyenne empirique $i \longmapsto \bar{Y}_i$, la borne supérieure $i \longmapsto \bar{Y}_i + r_i$ et la borne inférieure $i \longmapsto \bar{Y}_i - r_i$ avec des couleurs différentes dans le même graphe.

2.6 Estimateurs à noyaux

On cherche à estimer la densité de la loi d'un échantillon. La première approche est de tracer l'histogramme renormalisé des valeurs obtenues.

Une autre est d'estimer la densité de l'échantillon simulé par la méthode des noyaux décrite ci-après.

Soit f la densité de probabilité à estimer. Soit $(X_1,...,X_n)$ un échantillon de la v.a. i.i.d de loi de densité f. La mesure empirique

$$\Pi_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$$

de l'échantillon, où δ_x désigne la mesure de Dirac au point x, «converge» vers la mesure μ. Mais cette mesure empirique n'admet pas de densité par rapport à la mesure de Lebesgue. C'est pourquoi, on «régularise par convolution» Π_n avec une suite de noyaux $(K_h)_{h>0}$ qui vérifie :

$$\begin{cases} K_h(x) \geq 0 & \text{pour tout } h>0 \text{ et } x \in \mathbb{R}..... \\ \int_{\mathbb{R}} K_h(x)dx = 1 & \text{pour tout } h>0 \\ K_h \xrightarrow[h\to 0]{} \delta_0. & \end{cases}$$

On peut ainsi considérer la suite de noyaux $K_h(x) = K(x/h)$ où K peut par exemple désigner le noyau gaussien

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

ou le noyau d'Epanechnikov

$$K(x) = \frac{3}{4}(1-x^2)\mathrm{I}_{]-1,1[}(x).$$

On estime alors la densité f par la fonction

$$\hat{f}_n(x) = \frac{1}{nh_n}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h_n}\right).$$

La suite $\hat{f}_n$ converge vers f. On admettra qu'un choix judicieux pour la suite $(h_n)_n$ est de prendre $(h_n)_n$ de l'ordre de $= n^{-1/5}$.

On considère X et Y deux variables aléatoires gaussiennes centrées, réduites et indépendantes. On définit une nouvelle variable aléatoire Z telle que Z vaut X avec probabilité $\frac{1}{3}$ et $aY+b$ avec probabilité $\frac{2}{3}$ où $a>0$ et $b \in \mathbb{R}$. Vérifiez que la densité f de Z s'écrit

$$f(x) = \frac{2}{3a\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-b}{a}\right)^2} + \frac{1}{3\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

Utilisez le programme suivant pour voir comment évolue l'estimation de la densité en fonction de h. La vraie densité apparaît en rouge sur le graphique.

`noyau.sce`

```
clear;
// nc est le nombre de classes dans l'histogramme
// n la taille de l'échantillon
// h le pas de l'ordre de n^(-1/5)
function [] = estim_noyau(n,nc,hn)
  xbasc();
  a=1;
  b=3;
// 2 variables alétoires uniformes
  U=rand(n,1);
  V=rand(n,1);
// 2 gaussiennes centrées réduites indépendantes
  X = sqrt(-2*log(U)).*cos(2*%pi*V);
  Y = sqrt(-2*log(U)).*sin(2*%pi*V);
// Z = X avec proba 1/3 et aY+b avec proba 2/3
  epsilon = rand(n,1);
  Z = (a*Y+b).*(epsilon > 1/3) + X .* (epsilon <= 1/3);
// histogramme de la variable simulée E, nc=nombre de classes
  histplot(nc,Z)
// estimation de la densité de la loi de E par la méthode des noyaux,
// noyau Epanechnikov :
  C=[min(Z)-1:1/n:max(Z)+1];
  for i=1:length(C)
    B(i)=1/(n*hn)*3/4*sum((1-((C(i)-Z)/hn).^2).*(-1<(C(i)-Z)/hn).*((C(i)-Z)/hn<1));
  end
plot2d(C,B,2)
//tracé de la vraie densité
  f = (2/(3*a) * exp(-((C-b)/a).^2/2) + 1/3 * exp(-C.^2/2))/sqrt(2*%pi);
  plot2d(C,f,5);
endfunction
```

3 Value at Risk

La Value at Risk d'une fonction loss L à un niveau $\alpha \in]0,1[$ est

$$VaR_\alpha(L) \stackrel{\text{def}}{=} \inf\{m \in \mathbb{R} \mid \mathbb{P}(L \leq m) \geq \alpha\}$$

Intuitivement, dire que la $VaR_{95\%}$ d'un porefeuil est 100 veut dire que la fonction L dépassera la valeur 100 avec une probabilité inférieur à 5%. La $VaR_{95\%}$ nous informe sur le risque à un niveau donné, mais elle nous donne pas la répartition de la variable aléatoire loss L.

3.1 Value at Risk d'une loi log-normale

Calculer la Value at Risk aux différent nivaux (1%, 5%, 10%) d'une variable aléatoire log-normal. Tracer la fonction de répartition empirique d'une variable aléatoire de loi log-normal. Une variable aléatoire X est dite de loi log-normal si $X > 0$ and $\log X \sim \mathcal{N}(\mu, \sigma^2)$. Dans la simulation, on prendra $\mu = -1$ and $\sigma^2 = 0.09$.

```
stacksize(2*10^8);
N=1000; // nombre de simulation
moyenne=-1;
sigma2=0.09;
Y=grand(1,N,'nor',moyenne,sigma2) ;
// simulation d'une variable aléatoire de loi normale
Z=sort(exp(Y));
// decreasing sort of the simulated realizations of a Log-Normal r.v.
values=  Z($:-1:1);
// increasing sorting

 // the VaR is -quantile
alpha=0.99;
quantile=values(int(alpha*N));
disp('VaR_('+string(alpha) +')(Z)='+string(quantile))

alpha=0.95;
quantile=values(int(alpha*N));
disp('VaR_('+string(alpha) +')(Z)='+string(quantile))

alpha=0.90; // level
quantile=values(int(alpha*N));
disp('VaR_('+string(alpha) +')(Z)='+string(quantile))

xset('window',10) ; xbasc(); plot2d(values,linspace(0,1,N));
plot2d(values,alpha*ones(values),style=5);
xtitle('Empirical cumulated distribution of a Log-Normal random variable L:...log(L) ~ N

xset('window',11) ; xbasc(); histplot(10,Z);
xtitle('Empirical histogram of a Log-Normal random variable L:: ...
log(L) ~ N(' +string(moyenne) +',' +string(sigma2)+ ')')
```

3.2 Comparaison de la VaR suivant les lois

Comparer les quantiles de la loi normales et celle de la loi de Laplace et noter que la méthode paramétrique est très sensible au choix à priori de la famille paramétrique de lois.

```
stacksize(2*10^8);
```

```
N=10000; // nombre de simulation
moyenne=1;
sigma2=9;
Y1=grand(1,N,'nor',0,1) ;
Y2=grand(1,N,'exp',1);
epsilon=2*grand(1,N,'bin',1,0.5)-1 ;
Y2=epsilon.*Y2;
Y2=Y2/sqrt(2);
Y1=sqrt(sigma2)*Y1+moyenne;
Y2=sqrt(sigma2)*Y2+moyenne;
// simulation d'une variable aléatoire de loi normale
Z1=sort(Y1);
Z2=sort(Y2);
// decreasing sort of the simulated realizations of Normal r.v.
values1=  Z1($:-1:1);
values2=  Z2($:-1:1);
// increasing sorting
// the VaR is a quantile
alpha=0.99;
quantile1=values1(int(alpha*N));
quantile2=values2(int(alpha*N));
disp('VaR_('+string(alpha) +')(Z1)='+string(quantile1))
disp('VaR_('+string(alpha) +')(Z2)='+string(quantile2))

alpha=0.95;
quantile1=values1(int(alpha*N));
quantile2=values2(int(alpha*N));
disp('VaR_('+string(alpha) +')(Z1)='+string(quantile1))
disp('VaR_('+string(alpha) +')(Z2)='+string(quantile2))

alpha=0.90; // level
quantile1=values1(int(alpha*N));
quantile2=values2(int(alpha*N));
disp('VaR_('+string(alpha) +')(Z1)='+string(quantile1))
disp('VaR_('+string(alpha) +')(Z2)='+string(quantile2))

xset('window',10) ; xbasc();
plot2d(values1,linspace(0,1,N));
plot2d(values2,linspace(0,1,N),5);
plot2d(values1,alpha*ones(values1),style=5);
plot2d(values2,alpha*ones(values2),style=5);
xtitle('Empirical cumulated distribution of a Normal random variable and Laplace')
```

```
xset('window',10) ; xbasc();
alpha=[0.01:0.01:0.99]
plot2d(alpha,values1(int(alpha*N)), 5);
```

3.3 Méthode paramétrique et intervalle de confiance

Exercice : rappeler la méthode paramétrique pour construire un estimateur $\hat{\xi}_\alpha$ du quantile ξ_α, $\alpha \in]0,1[$, basé sur l'estimateur $\hat{\mu}$ de la moyenne μ et l'estimateur $\hat{\sigma}$ de l'écart-type σ de la variable qui nous intéresse L. Expliquer, en utilisant l'approche Monte Carlo, qu'on peut construire un intervalle de confiance empirique de ξ_α, à partir d'un échantillon i.i.d., de taille $M \in \mathbb{N}$, $(\hat{\xi}_\alpha^1, \cdots, \hat{\xi}_\alpha^M)$ suivant la loi de $\hat{\xi}_\alpha$, obtenu en simulant des couples de variables $((\hat{\mu}_1, \hat{\sigma}_1), \cdots, (\hat{\mu}_M, \hat{\sigma}_M))$ i.i.d. suivant (μ, σ). Illustrer votre construction par des simulations numériques.

3.4 Méthode nonparamétrique

Cette méthode est basée sur la statistique d'ordre et elle consiste à estimer le quantile d'ordre α par le quantile empirique du même ordre.

3.4.1 Intervalle de confiance exacte

On rappelle le résultat suivant.

Theorem 1 *Soit L une variable aléatoire de fonction de répartition F qu'on suppose continue et de fonction inverse généralisé F^{-1}. Soit $(L_k)_{1\leq k\leq n}$ un échantillon i.i.d. de même loi que L de taille $n \in \mathbb{N}$. Soit $(L_{(k)})_{1\leq k\leq n}$ sa statistique d'ordre. On pose $L_{(0)} = -\infty$ et $L_{(n+1)} = \infty$. On a pour tout $0 \leq k^- \leq k^+ \leq n$ et pour tout $p \in]0,1[$,*

$$\mathbb{P}(L_{(k^-)} \leq F^{-1}(p) \leq L_{(k^++1)}) = \mathbb{P}(k^- \leq \mathcal{B}(n,p) \leq k^+).$$

Exercice : Pour déterminer k^+ et k^- à un niveau de confiance x donnée on cherche le plus petit entier k tel que $\mathbb{P}(\mathcal{B}(n,p) > k) \leq \bar{x}/2$, $\bar{x} = 1 - x$, c'est aussi le premier indice k tel que la fonction de répartition de la loi binomiale de taille n et de paramètre p dépasse $1 - \bar{x}/2$. On déterminera k^- de la même façon. Ecrire un programme qui nous permet de construire un intervalle de confiance pour VaR_α pour un niveau donné. Illustrer ce résultat par des simulations numériques.

3.4.2 Intervalle de confiance asymptotique

Soit L une variable aléatoire et $(L_1, \cdots, L_n)$ un échantillon i.i.d. de même loi que L et de taille $n \in \mathbb{N}$. On veut construire un intervalle de confiance pour le quantile ξ_α, $\alpha \in]0,1[$, à l'aide du théorème de la limite centrale pour l'estimateur empirique $\hat{\xi}_\alpha$.

Theorem 2 *On suppose que L est une variable aléatoire de densité f qu'on suppose continue en ξ_α avec $f(\xi_\alpha) > 0$. L'estimateur empirique $\hat{\xi}_\alpha$ satisfait*

$$\frac{\sqrt{n}}{\sigma_\alpha}(\hat{\xi}_\alpha - \xi_\alpha) \xrightarrow[n\to\infty]{loi} \mathcal{N}(0,1) \quad avec \quad \sigma_\alpha = \frac{\sqrt{\alpha(1-\alpha)}}{f(\xi_\alpha)}.$$

Exercice :

1. Ecrire un programme qui nous permet de construire un intervalle de confiance pour VaR_α pour un niveau donné. On suppose que n est assez grand, $n \geq 30$, pour pouvoir utiliser l'approximation du théorème ci-dessus et qu'on approxime f par un estimateur de la méthode du noyau. Illustrer ce résultat par des simulations numériques.
2. Tracer la moyenne empirique ainsi que la médiane empirique de la loi normale centrée réduite en fonction de la taille de l'échantillon. Construire des intervalles de confiance asymptotique pour ces deux estimateurs et représenter les avec des couleurs différentes dans le même graphe.

3.5 Value at Risk sur un modèle financier

On s'intéresse à un modèle de panier constitué à partir de d actifs. On suppose que chacun de ces d actifs de prix S_t^i suit un modèle de black et Scholes guidé par un mouvement W_t^i :
$\frac{dS_t^i}{S_t^i} = rdt + \sigma dW_t^i, S_0^i = x_i.$

On prendra dans les applications numériques $x_i = 100 + (-1)^i * i$, $r = 0.05$ (taux d'intérêt exponentiel annuel) $\sigma = 0.3$ (volatilité annuelle) et $d = 10$.

Pour déterminer complètement le modèle on doit spécifier les corrélation entre les mouvements browniens. Pour simplifier le problème on suppose qu'ils sont indépendants.

1. Proposer une méthode de simulation pour le vecteur $(S_t^1, \ldots, S_t^d)$ pour $t \in [0, T]$. On prendra $\delta t = 1/250$, $T = 4$ et on simulera le vecteur au points $t_k = k\delta t$.
2. Pour chaque jour $t_k \in]0, T]$ on a le rendement de l'actif i est égale à

$$r_t^i = \frac{S_{t_{k-1}}^i - S_{t_k}^i}{S_{t_{k-1}}^i}$$

et le rendement du portefeuille

$$r_t = \sum_{i=1}^{d} p_i r_t^i.$$

avec $\sum_{i=1}^n p_i = 1$. On prendra dans les applications numériques $p_1 = \cdots = p_d = 1/d$, simuler r au points $t_k = k\delta t$.

3. La VaR à 99% historique du portefeuille est alors le quantille des rendements r_t obtenus au points $t_k = k\delta t$. Par exemple, si $k \in \{1, \cdots, 1000\}$, on classe les rendements par ordre croissant $r_{(1)}, \cdots, r_{(1000)}$ et $VaR_{0.99}^{hist} = r_{(10)}$. Calculer de même $VaR_{0.95}^{hist}$ et $VaR_{0.95}^{hist}$