



Fiche n° 7 (sur ordinateur)

Lors de chaque TD sur ordinateur, il est indispensable de venir avec votre login et votre mot de passe!

Ex 1. *Pour apprivoiser SCILAB*

Les étudiants ayant déjà utilisé SCILAB, par exemple dans le module M315, pourront passer directement à la troisième question de cet exercice.

1) Démarrer SCILAB en cliquant sur l'icône correspondante ou en tapant `Scilab&` dans une fenêtre de commande.

2) Cliquer sur `Help` et `Apropos` puis taper le mot-clé `bool2s`. Qu'obtient-on? Copier¹ les exemples fournis dans la fenêtre principale, puis essayer avec vos propres exemples. Vous pourrez répéter cette opération avec tous les mot-clés donnés dans cette fiche.

3) Effectuer quelques tirages de la loi $\mathcal{U}nif(]0, 1[)$ et de la loi $\mathcal{N}(0, 1)$ dans la fenêtre principale. On utilisera le mot-clé `rand`. La touche \uparrow du clavier sera également utile.

4) Choisir un n et un p et faire quelques tirages de la loi $\mathcal{B}in(n, p)$ (mots-clés : `bool2s` et `sum`). Comme on fera souvent des tirages de loi binomiale dans la suite, cliquer sur `Editor` et taper les commandes correspondantes dans la fenêtre de l'éditeur de programme. Sauver ce programme (en cliquant sur `File` et `Save`) sous le nom `binom.sce` puis cliquer sur `Execute` et `Load into scilab` pour l'exécuter.

Ex 2. 1) Transformer `binom.sce` en un programme qui affiche 10 tirages de la loi $\mathcal{B}in(3, \frac{1}{4})$. Le programme devra commencer par la ligne

```
n=3; p=0.25; nbtir=10;
```

et dans les lignes suivantes on écrira `n`, `p` et `nbtir` plutôt que 3, 0.25 et 10. Ceci permet de changer facilement les paramètres de la binomiale et le nombre de tirages à faire.

Indication : utiliser le mot-clé `for` et le vecteur-ligne `(1:nbtir)` qui contient la suite des entiers de 1 à `nbtir`.

1. Rappelons à tout hasard que le texte est enregistré en le surlignant avec le bouton gauche de la souris, et copié à l'emplacement du curseur quand on clique sur le bouton du milieu.

2) Créer un vecteur-ligne nommé `binomiales` (mot-clé : `ones`) et mettre les `nbtir` résultats de tirages dedans. Ajouter à la fin du programme les commandes `clf; histplot((-0.5):(n+0.5),binomiales)` et exécuter plusieurs fois le programme. Vérifiez que vous obtenez chaque fois l'histogramme des tirages : la surface de chaque rectangle doit être proportionnelle au nombre de tirages tombés dans la zone correspondante, et la surface totale couverte doit être égale à 1.

3) Combien valent les $P(X = k)$ pour une variable aléatoire X qui suit la loi $\mathcal{B}in(3, \frac{1}{4})$?

Exécutez votre programme avec `n=3`, `p=0.25` et `nbtir=10000`. Qu'observe-t-on² ? A quoi est dû ce phénomène ?

Dans la suite de cet exercice, on gardera `nbtir=10000`.

4) Exécuter le programme avec `p=0.25` et `n=50`, `p=0.25` et `n=100`, `p=0.25` et `n=200`, `p=0.25` et `n=300`, etc... (pour les grandes valeurs de `n`, cliquer sur `Zoom` et tracer un rectangle autour de la zone qui vous intéresse). Que constate-t-on ? Donner le nom mathématique du phénomène que vous observez.

5) Exécuter le programme avec `n=50` et `p=0.2`, `n=100` et `p=0.1`, `n=200` et `p=0.05`, `n=400` et `p=0.025`, etc... Que constate-t-on et quel est le nom mathématique du phénomène que vous observez ?

On pourra rajouter en fin de programme la ligne

```
plot2d3([0:40]', exp(-10)*10^(0:40)'./[1 cumprod(1:40)]', 2)
```

qui surimprime à notre histogramme celui d'une loi de Poisson (de quel paramètre ?).

Ex 3. 1) Calculer $\int_0^1 u^2 (\ln(u))^8 du$

2. Si le serveur est surchargé au moment du TP, il est possible qu'on n'observe rien du tout ! Il faudra alors remplacer 10000 par 1000, mais les résultats observés seront moins précis.

Comparer avec la moyenne empirique d'un grand nombre de tirages de $X = U^2(\ln(U))^8$ où U est une variable aléatoire de loi uniforme sur $]0, 1[$ (utiliser le mot-clé `log` et les symboles `*` et `^`). Comment s'appelle cette technique ?

2) A l'aide d'un programme, trouver une valeur approchée de

$$\frac{1}{\sqrt{2\pi}} \int_0^1 e^{-\frac{u^2}{2}} du$$

et comparer avec la valeur qui figure dans la table de la loi $\mathcal{N}(0, 1)$ (mots-clés : `exp sqrt %pi`). Comment les tables de valeurs numériques sont-elles construites ?

Ex 4. 1) Effectuer 1000 tirages X_1, X_2, X_3, \dots de la loi exponentielle de paramètre a , avec le a de votre choix. Le programme devra commencer par

```
nbtir=1000; a= ;
```

pour qu'on puisse changer facilement le nombre de tirages et le paramètre. Construire un vecteur-ligne à 1000 coordonnées contenant les 1000 moyennes empiriques successivement obtenues $\overline{X_1}, \overline{X_2}, \overline{X_3}, \dots$. Ce vecteur-ligne devra s'appeler `moyennes` (utiliser le mot-clé `cumsum` et la commande `./` qui effectue la division coefficient par coefficient entre deux vecteurs). Rajouter en fin de programme la ligne suivante qui affiche le graphe de la suite des $(\overline{X_n})_n$:

```
clf; plot2d2(1:nbtir,moyennes)
```

Que constate-t-on ? Cela se produit-il pour toutes les suites de tirages de la loi $\mathcal{Exp}(a)$? Pourquoi ? Observer ce phénomène en faisant varier le paramètre a et la longueur de la suite de tirages.

2) Faire la même chose en remplaçant la loi exponentielle par la loi de Cauchy centrée en 0 de paramètre d'échelle 1 (densité $x \mapsto \frac{1}{\pi(1+x^2)}$). On utilisera le mot-clé `tan`. Augmenter le nombre de tirages et zoomer si nécessaire. Que constate-t-on ? A quoi cela est-il dû ?

Ex 5. 1) Vérifier que la fonction $\frac{1}{2}(\mathbf{1}_{[0,1]} + \mathbf{1}_{[2,3]})$ est bien une densité de probabilité. Calculer l'espérance et la variance de la loi correspondante.

En utilisant la méthode du rejet, réaliser 1000 tirages suivant cette loi. Afficher les valeurs observées de la moyenne empirique et de la variance empirique (utiliser le symbole \wedge qui sert à élever à une puissance tous les coefficients d'un vecteur).

2) Même question avec la fonction $\frac{1}{2}(\mathbf{1}_{[0,1]} + \mathbf{1}_{[100,101]})$.

Pourquoi la même méthode (rejet) est-elle peu recommandée pour simuler cette loi? Trouver une méthode de simulation qui convienne, réaliser 1000 tirages et afficher les valeurs observées de la moyenne empirique et de la variance empirique.

Ex 6. Choisir $\mu \in \mathbb{R}$ et $\sigma \in \mathbb{R}^+$ et afficher 5 tirages de la loi $\mathcal{N}(\mu, \sigma^2)$. Le programme devra commencer comme d'habitude par
`mu= ; sigma= ; nbtir=5;`
et on appellera X le vecteur-ligne contenant les résultats des 5 tirages. Comment s'appelle la fonction dont SCILAB trace le graphe quand on rajoute en fin de programme les commandes suivantes?

```
clf; plot2d([mu-4*sigma,mu+4*sigma],[0,1],-1);  
Y=gsort(X,"g","i");  
xsegs([mu-10*sigma Y;Y mu+10*sigma],[0:nbtir;0:nbtir]/nbtir);
```

1) Essayer avec `nbtir=10`, `nbtir=20`, `nbtir=30`,... (il pourra être utile de supprimer l'affichage des valeurs de X en rajoutant un `;` à la ligne correspondante). Comment évolue le graphe quand on augmente le nombre de tirages ? Quel nom porte cette propriété et à quoi sert-elle ?

2) Les commandes suivantes ajoutent sur le même écran graphique la fonction de répartition de la loi $\mathcal{N}(\mu, \sigma^2)$:

```
xx=mu-4*sigma:0.1:mu+4*sigma;
plot2d(xx,cdfnor("PQ",xx,mu*ones(xx),sigma^2*ones(xx)),5);
```

Vous avez (ou devez avoir) en tête une idée de la façon dont varie le graphe de la fonction de répartition de la loi $\mathcal{N}(\mu, \sigma^2)$ quand on fait varier μ et σ . Vérifier-la en exécutant votre programme avec différentes valeurs de μ , puis en prenant des σ de plus en plus grands ou de plus en plus petits.

Ex 7. 1) On considère une variable aléatoire X de loi $\mathcal{B}in(n, p)$. Construire à partir de X un intervalle de confiance à 95% pour l'estimation de p , en utilisant le théorème central limite avec autonormalisation :

Ecrire un programme qui fait un tirage selon la loi $\mathcal{B}in(n, p)$ (avec $n = 200$ et $p = \frac{1}{2}$) et qui affiche l'intervalle de confiance correspondant. Le programme devra commencer par `n=200; p=0.5;` afin de faire varier facilement n et p . Utiliser plusieurs fois le programme et vérifier qu'il fonctionne correctement.

2) Rajouter `nbtir=100;` sur la première ligne du programme. Créer deux vecteurs-lignes ayant chacun `nbtir` coordonnées, nommés `intinf` et `intsup`. Faire `nbtir` tirages selon la loi $\mathcal{B}in(n, p)$ en stockant les `nbtir` bornes inférieures des intervalles de confiance dans le vecteur `intinf` et les bornes supérieures correspondantes dans le vecteur `intsup`. Rajouter en fin de programme les commandes

```
intcouleur=5*ones(1:nbtir)+8*bool2s( p>intinf & p<intsup );
clf; plot2d([0,nbtir+1],[0,1],-1);
xsegs([1:nbtir;1:nbtir],[intinf;intsup],intcouleur);
xsegs([0;nbtir+1],[p;p],2);
```

Ceci dessine tous les intervalles de confiance obtenus, avec des couleurs différentes selon qu'ils contiennent ou non le paramètre p . Combien obtenez-vous d'intervalles ne contenant pas p ? Est-ce normal? Quel loi suit le nombre d'intervalles ne contenant pas p ?

Programmes SCILAB utilisés dans la fiche 7 (IS 2006-2007)

Remarque :

Lorsqu'on écrit un programme, on veille généralement à optimiser le code pour que l'exécution soit la plus rapide possible (beaucoup de fonctions vectorielles, peu de boucles) ou que la place utilisée en mémoire soit la plus limitée possible (structure très séquentielle avec beaucoup de boucles). Les programmes suivants ne sont optimaux en aucun de ces deux sens. Leur structure a été choisie pour être la plus simple et la plus lisible possibles (sauf en ce qui concerne les commandes graphiques, pour lesquelles on a privilégié la brièveté du code).

Ex 2.

```
n=800; p=0.0125; nbtir=10000;
binomiales=ones(1:nbtir);
for i=(1:nbtir) do
    binomiales(i)=sum(bool2s(rand(1:n,"uniform")<p));
end;
clf; histplot((-0.5):(n+0.5),binomiales)
plot2d3([0:n]',exp(-10)*10^(0:n)'./[1 cumprod(1:n)]',2)
```

Ex 3.

```
nbtir=100000; a=10;
//tirages=-log(rand(1:nbtir,"uniform"))/a; // loi exponentielle
tirages=tan(%pi*(rand(1:nbtir,"uniform")-0.5*ones(1:nbtir))); // loi de Cauchy
moyennes=cumsum(tirages)./(1:nbtir);
clf; plot2d2(1:nbtir,moyennes)
```

Ex 4. 1)

```
nbtir=500000;
U=rand(1:nbtir,"uniform");
integrale=sum((U.^2).*(log(U)).^8)/nbtir
```

2)

```
nbtir=500000;
U=rand(1:nbtir,"uniform");
intnormale=sum(exp(-(U.^2)./2))/nbtir/sqrt(2*%pi)
```

Ex 5. 1)

```
nbtir=1000;
tirages=ones(1:nbtir);
for i=(1:nbtir) do
    X=3*rand(1,"uniform");
```

```

    while ((X>1)&(X<2)) do X=3*rand(1,"uniform"); end;
    tirages(i)=X;
end;
moyenneempirique=sum(tirages)/nbtir
varianceempirique=sum((tirages-moyenneempirique*ones(tirages)).^2)/nbtir

2)

nbtir=1000;
tirages=rand(1:nbtir,"uniform");
tirages=2*tirages+99*bool2s(tirages>0.5);
moyenneempirique=sum(tirages)/nbtir
varianceempirique=sum((tirages-moyenneempirique*ones(tirages)).^2)/nbtir

```

Ex 6.

```

mu=5; sigma=2; nbtir=1000;
X=sigma*rand(1:nbtir,"normal")+mu*ones(1:nbtir);
clf; plot2d([mu-4*sigma,mu+4*sigma],[0,1],-1);
Y=gsort(X,"g","i");
xsegs([mu-10*sigma Y;Y mu+10*sigma],[0:nbtir;0:nbtir]/nbtir);
xx=mu-4*sigma:0.1:mu+4*sigma;
plot2d(xx,cdfnorf("PQ",xx,mu*ones(xx),sigma*ones(xx)),5);

```

Ex 7.

```

n=200; p=0.5; nbtir=100;
intinf=ones(1:nbtir); intsup=ones(1:nbtir);
for i=(1:nbtir) do
    proportionempirique=sum(bool2s(rand(1:n,"uniform")<p))/n;
    ecart=1.96*sqrt(proportionempirique*(1-proportionempirique)/n);
    intinf(i)=proportionempirique-ecart;
    intsup(i)=proportionempirique+ecart;
end;
intcouleur=5*ones(1:nbtir)+8*bool2s( p>intinf & p<intsup );
clf; plot2d([0,nbtir+1],[0,1],-1);
xsegs([1:nbtir;1:nbtir],[intinf;intsup],intcouleur);
xsegs([0;nbtir+1],[p;p],2);

```