EXERCICES CORRIGES STATISTIQUES ET PROBABILITES

2DE

1RE

Scilab pour les Lycées

Ce livret a été co-écrit par le Consortium Scilab et Christine Gomez, professeur de mathématiques au lycée Descartes à Antony.

Introduction

1- Exercices de seconde

2- Exercices de première

3- Fonctions Scilab utiles

Introduction

Ce livret présenté sous la forme d'exercices corrigés de statistiques et probabilités donne des exemples d'utilisation de Scilab et de son module Lycée dans le cadre des nouveaux programmes 2011-2012 pour les classes de seconde et de première.

De nouvelles fonctions ont été ajoutées au module Lycée (disponibles à partir de la version 1.4-1) afin de faciliter l'utilisation des lois de probabilité introduites au programme de première :

- La loi binomiale,
- La loi géométrique tronquée.

Certains exercices se présentent sous forme de fiches guidées et peuvent être donnés tels quels aux élèves. D'autres, à la problématique volontairement plus ouverte, peuvent être adaptés et / ou personnalisés par l'enseignant.

Tous les fichiers Scilab correspondant aux exercices proposés sont téléchargeables sur le site : http://www.scilab.org/education/lycee/docs/livret_stat_proba

Exercices de seconde

STATISTIQUES

Exercice 1 (Fichier : Seconde-Stats-Ex 1.sce)

Prendre en main les différentes commandes, à montrer en classe et à faire refaire par l'élève à la maison.

On donne les températures sur une année à Mexico et à Barcelone.

Mois	J	F	M	A	M	J	J	A	S	O	N	D
Mexico	12,4	14,1	16,2	17,4	18,4	17,7	16,7	16,8	16,3	15,1	13,9	12
Barcelone	9,5	10,3	12,4	14,6	17,7	21,5	24,3	24,3	21,8	17,6	13,5	10,3

- Saisir les données de températures des deux villes dans Scilab.
- Tracer le diagramme en bâtons des températures de Mexico.
- Tracer le diagramme en bâtons des températures des deux villes côte à côte.
- Calculer et afficher l'étendue de la série, la moyenne annuelle, la médiane et les quartiles pour chaque ville.

Corrigé :

- Les donnés sont entrées dans Scilab sous la forme :

```
M=[12.4,14.1,16.2,17.4,18.4,17.7,16.7,16.8,16.3,15.1,13.9,12];
B=[9.5,10.3,12.4,14.6,17.7,21.5,24.3,24.3,21.8,17.6,13.5,10.3];
```

- Diagramme en bâtons des températures de Mexico :

```
clf; bar(M)
```

- Diagramme en bâtons des températures de Mexico et Barcelone côte à côte :

```
clf; bar([1:12],[M',B'])
```

- Étendue de la série :

```
eM=max(M)-min(M); eB=max(B)-min(B);
afficher("L''étendue des températures à Mexico est : "+string(eM))
afficher("L''étendue des températures à Barcelone est : "+string(eB))
```

◗ Moyenne annuelle :

```
mM=moyenne(M); mB=moyenne(B);
afficher("La température moyenne annuelle à Mexico est : "+string(mM))
afficher("La température moyenne annuelle à Barcelone est : "+string(mB))
```

◗ Médiane :

```
MM=mediane(M); MB=mediane(B);
afficher("La médiane des températures à Mexico est : "+string(MM))
afficher("La médiane des températures à Barcelone est : "+string(MB))
```

◗ Quartiles :

```
qM=quartiles(M); qB=quartiles(B);
afficher("Les quartiles à Mexico sont : "+string(qM(1))+" et ...
"+string(qM(2)))
afficher("Les quartiles à Barcelone sont : "+string(qB(1))+" et ...
"+string(qB(2)))
```

Exercice 2 **(Fichier : Seconde-Stats-Ex 2.sce)**

Comprendre et compléter un programme, séance TD évaluée en salle informatique.

On donne le programme suivant :
X représente les longueurs en centimètres des pantalons vendus dans un magasin en une semaine et **N** les effectifs correspondants :

```
X=[74,76,78,80,82,84,86];
N=[10,15,21,20,17,13,4];
ET=sum(N);
F=N/ET;
clf; bar(X,F,"r")
```

◗ Recopier ce programme dans l'éditeur et l'exécuter.
◗ Que calcule **ET** ? Que calcule **F** ? Les afficher.
◗ Que représente le diagramme en barres ? En allant dans Édition / Propriétés des axes, rendre le tracé plus lisible.
◗ À quoi sert **clf** ? À quoi sert **"r"** ?
◗ Ajouter à ce programme le calcul de la moyenne pondérée, l'afficher avec une phrase.

À noter
Le menu Propriétés des axes n'est pas disponible sous Mac OS X.

✐ Corrigé :

- **`ET`** calcule l'effectif total, **`F`** donne la liste des fréquences.

```
afficher(ET);afficher(F)
```

- Le diagramme en barres représente les fréquences. On le lira plus facilement en déplaçant l'axe des ordonnées :
Axe Y : Axis location / location / left.
- **`clf`** signifie « clear figure » et est utilisé pour effacer tout tracé précédemment affiché.
- **`"r"`** indique que le diagramme sera tracé en rouge (« red »).

```
m=moyenne_ponderee(X,N);
afficher("La moyenne pondérée est : "+string(m))
```

Exercice 3 **(Fichier : Seconde-Stats-Ex 3.sce)**

Travailler sur des classes, exercice peu guidé donc plus difficile, à personnaliser selon le niveau de la classe.

On a relevé les temps d'attente des skieurs à une remontée mécanique :

Temps d'attente en minutes	[0 ; 2[	[2 ; 6[	[6 ; 10[	[10 ; 30[
Nombre de skieurs	20	42	19	27

- Calculer l'effectif total.
- Tracer l'histogramme.
- Calculer le temps d'attente moyen et le transformer en minutes, secondes.

✐ Corrigé :

```
T=[0,2,6,10,30];
N=[20,42,19,27];
n=sum(N);
afficher("L''effectif total est : "+string(n))

clf; histogramme(T,N)

for i=1:4
    x(i)=(T(i)+T(i+1))/2;
end
m=(N*x)/n;
mn=floor(m);
s1=(m-mn)*60; s=round(s1);
afficher("Le temps d''attente moyen est :"+...
string(mn)+" minutes "+string(s)+" secondes")
```

> À noter
> **`N`** est un vecteur ligne et **`x`**, calculé dans la boucle, est un vecteur colonne. La moyenne doit donc se calculer en faisant **`N*x`** et non pas **`x*N`**. On pourrait aussi transformer **`x`** en vecteur ligne en le transposant : **`x'`** et utiliser **`moyenne_ponderee(x',N)`**

FLUCTUATIONS D'ÉCHANTILLONNAGES

Exercice 1 (Fichier : Seconde-Fluct-Ex 1.sce)

Illustrer le principe de l'intervalle de confiance. Exercice guidé, très progressif qui peut être utilisé tel quel en TD et évalué.

Simulons 100 lancers d'une pièce équilibrée et cherchons la fréquence d'apparition du côté pile. Pour cela, on tire aléatoirement les valeurs 0 ou 1, soit 0 pour le côté face et 1 pour le côté pile.

◗ Taper dans l'éditeur :

```
L= tirage_entier(100,0,1);
F=frequence(1,L);
afficher(F)
```

◗ Exécuter ce programme plusieurs fois. Que contient **L** ? Que compte **F** ? Que remarque-t-on pour les valeurs de **F** ?

Nous allons maintenant faire 1 000 séries de 100 lancers de pièce et observer comment fluctuent les fréquences.

◗ Taper dans l'éditeur :

```
for i = 1 : 1000
  L= tirage_entier(100,0,1);
  F(i)=frequence(1,L);
end
clf; quadrillage; plot(F,"*")
```

◗ Exécuter ce programme.
◗ Combien de fréquences sont en dehors de l'intervalle [0,4 ; 0,6] ?
◗ Peut-on dire qu'au moins 95 % des fréquences sont dans l'intervalle [0,5 – 1/10 ; 0,5 + 1/10] ?
◗ Écrire un programme simulant 1 000 séries de 400 lancers et tracer le nuage des fréquences.
◗ Peut-on dire qu'au moins 95 % des fréquences sont dans l'intervalle [0,5 – 1/20 ; 0,5 + 1/20] ?

✐ Corrigé :

Les élèves doivent compter moins de 5 % de fréquences hors de l'intervalle [0,4 ; 0,6]. Il peut être nécessaire de faire un zoom.
On prend $n = 400$, donc $\sqrt{n} = 20$. Changer le programme en :

```
for i = 1 : 1000
  L= tirage_entier(400,0,1);
  F(i)=frequence(1,L);
end
clf; quadrillage; plot(F,"*")
```

À noter
Si l'élève a simulé plus de 1 000 valeurs de **L**, les valeurs au-delà de l'indice 1 000 restent stockées dans **L**. Il peut être utile de vider **L** en écrivant **clear L**, voire de vider la mémoire en écrivant **clear** dans la console ou au début du programme.

Exercice 2 (Fichier : Seconde-Fluct-Ex 2.sce)

Reprendre des exercices classiques donnés dans les ressources du ministère de l'Éducation nationale et se familiariser avec les tirages aléatoires et les boucles.

Dans le village de Xicun, en Chine, 20 enfants dont 16 garçons sont nés en 2000. Dans la réserve d'Aamjiwnaang, au Canada, 132 enfants dont 46 garçons sont nés entre 1999 et 2003.
On supposera que la proportion habituelle de garçons à la naissance est de 50 % (elle est en réalité d'environ 51,2 %). Faire 100 simulations de chaque situation. Que peut-on en déduire ?

Corrigé :

```
//Xicun

for k=1:100
    T=tirage_entier(20,0,1);
    GX(k)=taille(find(T==1)); //ou bien GX(k)=frequence(1,T)*20;
end
clf; quadrillage; plot(GX,".")

//Aamjiwnaang

for k=1:100
    T=tirage_entier(132,0,1);
    GA(k)=taille(find(T==1)); //ou bien GA(k)=frequence(1,T)*132;
end
clf; quadrillage; plot(GA,".")
```

Les résultats prouvent que moins de 5 % des réponses concordent avec la réalité. Le hasard seul n'explique donc pas ces deux cas.
En Chine, la situation s'explique par l'acquisition d'une machine à ultra-sons permettant pour peu de frais de déterminer le sexe du fœtus et éventuellement de mettre un terme à une grossesse. Au Canada, la proximité d'industries chimiques est en cause, certains polluants déséquilibrant les sexe-ratios.

Exercice 3 (Fichier : Seconde-Fluct-Ex 3.sce)

Utiliser les tirages réels et la partie entière et permettre une analyse critique.

En Novembre 1976 dans un comté du sud du Texas, Rodrigo Partida était condamné à huit ans de prison. Il attaqua ce jugement au motif que la désignation des jurés de ce comté était discriminante à l'égard des Hispano-Américains. Alors que 79,1 % de la population de ce comté était d'origine hispanique, sur les 870 personnes convoquées pour être juré lors d'une certaine période de référence, il n'y eut que 339 personnes d'origine hispanique.

- Quelle est la fréquence des jurés d'origine hispanique observée dans ce comté du Texas ?
- Simuler avec Scilab, 100 échantillons aléatoires de taille $n = 870$ dans une population où la fréquence des habitants d'origine hispanique est $p = 0{,}791$.
  - Calculer les bornes de l'intervalle $\left[p-1/\sqrt{n}\,;p+1/\sqrt{n}\right]$
  - Quel est le pourcentage des simulations fournissant une fréquence en dehors de l'intervalle précédent ?
- Sur les simulations, est-il arrivé au hasard de fournir une fréquence d'habitants d'origine hispanique comparable à celle des jurés d'origine hispanique observée dans ce comté du Texas ?
- Comment expliquez-vous cette situation ?

✐ Corrigé :

- La fréquence observée des jurés d'origine hispanique est $f = \frac{339}{870}$ soit environ 0,39.
-
```
for k=1:100
    T=floor(tirage_reel(870,0.791,1.791))
    N(k)=frequence(1,T);
end
clf; quadrillage; plot(N,"+")
```

On a $\left[0{,}791 - \frac{1}{\sqrt{870}}\,;\,0{,}791 + \frac{1}{\sqrt{870}}\right]$ c'est-à-dire environ [0,76 ; 0,82].
Sur le graphique des 100 simulations, 4 points sont en dehors de l'intervalle précédent, soit 4 % des cas.

- Non. La fréquence observée 0,39 est très loin des valeurs obtenues sur les simulations.
- La constitution des jurys n'est sans doute pas totalement aléatoire.

> À noter
> Il faudra parler rapidement de `floor` (partie entière).
> Les données étudiées constituent une preuve statistique que la constitution de ces jurys n'est pas totalement aléatoire, c'est-à-dire que ceux-ci ne sont pas représentatifs : il n'est pas possible de considérer que les jurys résultent d'un tirage au sort où chaque élément de la population a les mêmes chances d'être choisi. Mais c'est tout ce que l'on peut dire et, en particulier, il n'est pas possible de se prononcer sur les causes avec cette simple étude statistique. D'autres critères doivent être intégrés à l'enquête pour conclure sur les conditions réelles de constitutions des jurys.

Approche des probabilités par les fréquences

Exercice 1 (Fichier : Seconde-Proba-Ex 1.sce)

Conjecturer un calcul de probabilités en utilisant le langage naturel et le langage Scilab. Exercice très guidé, à faire en TD.

On lance deux dés équilibrés numérotés de 1 à 6, on cherche la probabilité pour que la somme des deux numéros obtenus soit supérieure ou égale à 10.
On pose **`T=tirage_entier(2,1,6)`**. Que contient **`T`** ? Pourquoi permet-il de simuler l'expérience ?
On va faire 1 000 lancers et déterminer la fréquence des sommes supérieures ou égales à 10.

- Compléter :

Algorithme	Éditeur Scilab

Initialisons la fréquence f à 0
Début de boucle à répéter 1000 fois
  Simulons le lancer de 2 dés
  Calculons ……………………………. des 2 numéros
  Début de test : si S ≥ 10
    On augmente …. de 1/1000
  Fin du …
Fin de la …………
Affichons la fréquence
	`for n=1:1000`
	    `T=...............................;`
	    `S=sum(T);`
	    `if ............... then`
	        `f=...............;`
	    `end`
	`end`
	`....................(f)`

- Écrire le programme complété dans l'éditeur, puis l'exécuter.

- Modifier le programme pour exécuter 100 000 lancers, puis pour que le nombre **N** de lancers puisse être changé au début du programme.
- Exécuter plusieurs fois le programme. Deviner la probabilité cherchée et la justifier.

Corrigé :

Algorithme	Éditeur Scilab

Initialisons la fréquence f à 0
Début de boucle à répéter 1000 fois
Simulons le lancer de 2 dés
Calculons la somme des 2 numéros
Début de test : si S ≥ 10
On augmente f de 1/1000
Fin du test
Fin de la boucle.
Affichons la fréquence
	`f=0;`
	`for n=1:1000`
	`T=tirage_entier(2,1,6);`
	`S=sum(T);`
	`if S>=10 then`
	`f=f+1/1000;`
	`end`
	`end`
	`afficher(f)`

- **f** s'affiche.
- Remplacer 1 000 par 100 000 aux lignes 2 et 6, puis effectuer les modifications en gras, on peut alors changer la valeur de **N** :

```
f=0; N=1000;
for n=1:N
    T=tirage_entier(2,1,6);
    S=sum(T);
    if S>=10 then
        f=f+1/N;
    end
end
afficher(f)
```

- La probabilité cherchée est P(S=10) + P(S=11) + P(S=12), soit (3 + 2 + 1)*1/6*1/6, soit 1/6.

Exercice 2 **(Fichier : Seconde-Proba-Ex 2.sce)**

Simuler une expérience aléatoire, conjecturer puis démontrer les valeurs des probabilités. Exercice ouvert qui laisse place aux initiatives, à personnaliser selon l'usage souhaité.

Un lapin saute aléatoirement à droite ou à gauche, quatre fois.
En simulant 1 000 fois cette expérience, estimer la fréquence de chaque position finale. Retrouver les probabilités par calcul.

✐ Corrigé :

```
//Simulation de 1000 promenades
for n=1:1000
  t=2*tirage_entier(4,0,1)-1;
  s(n)=sum(t);
end

//Calcul des fréquences des 9 positions possibles
for i=1:9
  Y(i)=frequence(i-5,s);
end
afficher(Y)

//Diagramme en barres
clf; quadrillage; bar([-4:4],Y,"m")
```

À noter
Pour les probabilités, faire un arbre. La probabilité de chaque branche est 1/16. Il y a une branche pour les positions -4 et +4, donc $p = 0{,}625$, il y a quatre branches pour -2 et 2 donc $p = 0{,}25$, et 6 branches pour 0, donc $p = 0{,}375$

Exercice 3 **(Fichier : Seconde-Proba-Ex 3.sce)**

Exercice plus complexe faisant intervenir des notions un peu hors programme. Cet exercice permet de suivre les valeurs de la fréquence au fur et à mesure que le nombre de simulations augmente et de constater que la fréquence théorique n'est approchée qu'après un grand nombre de simulations.

Deux points A et B sont pris au hasard sur un segment de longueur 1.
Quelle est la probabilité de l'événement : « la longueur AB est supérieure à 0,5 » ?

- Simuler une expérience aléatoire.
- Faire ensuite une boucle pour simuler 1 000 fois l'expérience et construire le nuage des fréquences successives.

Corrigé :

```
// On prend aléatoirement 2 nombres entre 0 et 1
// On cherche la fréquence avec laquelle la distance entre eux
// est supérieure à 0.5

N=1000; f(1)=1;
for k=1:N
    T=tirage_reel(2,0,1);
    d=abs(T(1)-T(2));
    D=floor(d+0.5);
    f(k+1)=(k*f(k)+D)/(k+1);
end
clf;quadrillage;plot(f,".")
```

À noter

Il faudra parler rapidement de **`abs`** (valeur absolue), puis de **`floor`** (partie entière). On remarquera que si $0 \leq d < 0,5$, alors $D = 0$ et si $0,5 \leq d \leq 1$, alors $D = 1$.
On calcule le nombre de fois où la distance est supérieure à 0,5 en écrivant :
$(k+1) f(k+1) = k f(k) + D$, ce qui donne l'expression de $f(k+1)$.
On trouve que la fréquence se stabilise vers 0,25. On peut le prouver en admettant que la probabilité est égale à l'aire du domaine des points $M(x, y)$ avec x et y dans $[0 ; 1]$ et $|x - y| > 0,5$

Exercices de première

NOUVELLES LOIS DE PROBABILITÉ

Exercice 1

(Fichier : Première-Loi bin-Ex 1.sce)

Sur une situation simple, permettre une approche de la loi binomiale. Attention F(0) n'existe pas, il faut donc décaler les indices. Mise en œuvre de deux boucles.

On lance 10 fois un dé équilibré à 6 faces. On cherche combien de fois le 6 apparaît. Cette expérience est répétée 100 000 fois, on calcule les valeurs des fréquences avec lesquelles le 6 est apparu 0 fois, 1 fois, …, 10 fois.

```
for k=1:100000
    t=tirage_entier(10,1,6);
    B(k)=frequence(6,t)*10;
end
for i=0:10
    F(i+1)=frequence(i,B);
    afficher([i,F(i+1)])
end
clf; bar(F)
```

- Recopier ce programme dans l'éditeur. Que contient **`t`** ? Que compte **`B(k)`** ?
- Exécuter le programme. Pourquoi doit-on appeler **`F(i+1)`** la fréquence de **`i`** ?
- Retrouver les probabilités d'apparition du 6 par un calcul théorique. Comparer avec la simulation.

La loi binomiale

Soit X une variable aléatoire suivant une loi binomiale de paramètres n et p.
La fonction Scilab **`loi_binomiale(n,p)`** retourne le vecteur des $n+1$ valeurs $[p(X=0), \ldots, p(X=n)]$.
Si on précise un troisième argument, **`loi_binomiale(n,p,k)`** retourne la probabilité $p(X=k)$.
La fonction Scilab **`rep_binomiale(n,p)`** retourne le vecteur des $n+1$ valeurs $[p(X\leq 0), .., p(X\leq n)]$.
La fonction Scilab **`rep_binomiale(n,p,t)`** retourne la probabilité $p(X\leq t)$.

✎ Corrigé :

- **`t`** est une liste de 10 nombres entiers choisis aléatoirement entre 1 et 6, simulant les 10 lancers.
- **`B(k)`** compte le nombre de 6 apparus lors de ces 10 lancers, lors de la **`k`**ième expérience.

◗ L'exécution du programme demande quelques secondes d'attente pour simuler un million de lancers.
◗ **`F(0)`** n'existe pas, on décale donc les indices pour que **`F(1)`** représente la fréquence du 0 et par conséquent **`F(i+1)`** représente la fréquence de **`i`**.
◗ Le calcul théorique consiste à retrouver la formule de la loi binomiale et à comparer $P(X=i)$ avec **`F(i+1)`**.

Exercice 2 **(Fichier : Première-Loi bin-Ex 2.sce)**

Une approche de la loi binomiale plus ludique. On peut laisser les élèves programmer seuls sur le modèle de l'exercice précédent ou aménager des questions progressives. Prolongement intéressant avec un test. Faire ensuite les calculs théoriques.

Jean-Claude Dusse s'inscrit pour six jours de cours de ski. On lui annonce qu'il ne peut pas choisir son moniteur, mais que celui-ci sera tiré au hasard chaque matin parmi l'équipe qui comprend autant d'hommes que de femmes.
◗ Inquiet, Jean-Claude se demande quelles sont ses chances de tomber sur un moniteur ou une monitrice. Il simule 100 000 semaines de cours de ski.
◗ Après réflexion, Jean-Claude cherche seulement à approcher la probabilité d'avoir une monitrice au moins trois jours dans la semaine.

✐ Corrigé :

◗
```
for k=1:100000
  t=tirage_entier(6,0,1);
  N(k)=taille(find(t==1));
end
for k=0:6
  fr(k+1)=frequence(k,N);
  afficher([k,fr(k+1)])
end
clf; bar(fr)
```

◗
```
N=100000;
f=0;
for k=1:N
    t=tirage_entier(6,0,1);
```

```
        if taille(find(t==1))>=3 then
            f=f+1/N;
        end
    end
    afficher(f)
```

Exercice 3

(Fichier : Première-Loi gt-Ex 3.sce)

Approche de la loi géométrique tronquée à partir d'une situation concrète. Le programme contient deux boucles imbriquées, on pourra commencer par simuler une fois l'expérience (boucle sur k) avant de la faire N fois (boucle sur i).

Léa a une énorme réserve de bonbons, mais un quart seulement est au citron, son parfum préféré. Elle prend un bonbon au hasard, et s'il n'est pas au citron, elle le remet et mélange, et ainsi de suite jusqu'à ce qu'elle tombe sur un bonbon au citron.
Le programme suivant simule 10 000 fois cette expérience en limitant à 30 tirages, et calcule les fréquences de chaque rang d'obtention du bonbon au citron.

> La loi géométrique tronquée
> On reproduit une expérience de Bernoulli de probabilité p jusqu'à obtention du succès. On cherche la probabilité que le succès arrive au rang k. Théoriquement le succès pourrait ne jamais arriver, on limitera donc le nombre d'expériences à un entier n pour mettre fin au programme. On dit que la loi est tronquée.
> Soit X la variable aléatoire suivant la loi géométrique tronquée de paramètres n et p.
> On convient que $X = 0$ si le succès ne se produit pas lors des n expériences.
> La fonction Scilab **`loi_geometrique(n,p)`** retourne le vecteur des $n+1$ valeurs $[p(X=0),\ldots,p(X=n)]$.
> La fonction Scilab **`loi_geometrique(n,p,k)`** retourne la probabilité $p(X=k)$.

```
n=30; p=0.25; N=10000;
X=zeros(1,n);
for i=1:N
  for k=1:n
    a=floor(rand()+p);
    if a==1 then
      X(k)=X(k)+1/N;
      break
    end
  end
end
clf; bar(X)
```

◗ Recopier ce programme dans l'éditeur. Que contient **X** au début ?
◗ Que fait **a** ? Proposer une autre façon de procéder au tirage.
◗ À quoi sert la commande **break** ?
◗ Quelle est la formule donnant la probabilité d'obtention d'un bonbon au citron au rang **k** ? Afficher dans un tableau les valeurs de **k**, les fréquences et les probabilités correspondantes.
◗ Calculer la somme des probabilités pour **k** allant de 1 à **n**. Qu'en pensez-vous ?

Corrigé :

◗ **X** est une liste (vecteur ligne) de **n** zéros. Lorsque Léa tire un bonbon au citron lors du tirage **k**, **X(k)** augmente de **1/N**, pour contenir en fin de simulation la fréquence de l'événement « Léa a tiré son premier bonbon au citron au rang **k** ».
◗ **a** prend la partie entière d'un nombre aléatoire compris entre p et 1+p. Si ce nombre est compris entre 1 et 1+p, **a** vaudra 1, sinon **a** vaudra 0.
La proportion étant ¼, on peut utiliser un tirage entier entre 1 et 4 et convenir par exemple que 1 représentera un bonbon au citron :

```
for k=1:n
    a=tirage_entier(1,1,4);
    if a==1 then
      X(k)=X(k)+1/N;
      break
    end
end
```

◗ La commande **break** fait sortir de la boucle. En effet, dès que Léa a tiré un bonbon au citron, l'expérience s'arrête. Voir, dans l'exercice 4, une autre façon de programmer.
◗

```
for k=1:n
     G(k)= (1-p)^(k-1)*p;
     afficher([k,X(k),G(k)])
 end
```

◗ La somme des probabilités : **sum(G)** donne 0.9998214179098. Pour obtenir 1, il faut ajouter la probabilité de ne jamais tirer de bonbon au citron, soit $(1-p)^{30}$.

Exercice 4 (fichier Première-Loi gt-Ex 4.sce)

Simuler la loi géométrique tronquée dans le cas général. Reprise du document ressource du ministère.

- Compléter le tableau ci-dessous. On simule **n** épreuves de Bernoulli, avec une probabilité de succès **p**. L'algorithme donne la valeur de **X**, rang du premier succès. S'il n'y a pas de succès, **X=0**.
- Exécuter le programme Scilab en prenant **n=50** et **p=0.3**.

Algorithme	Éditeur Scilab
Entrées : valeur de n, valeur de p	`n=50; p=0.3;`
Initialisations : a prend la valeur 0, k prend la valeur 0	`a=…; k=…;`

Traitement : Tant que a = 0 et k < n
a prend la valeur ent(NbrAléat + p)
k prend la valeur ………
Fin de la boucle "tant que"
	`while (a==0 & k<n)`
	`a=floor(………………);`
	`k=k+1;`
	`end`

Sortie : Si a = 1
Alors afficher message "X = " valeur de …
Sinon afficher message "X = " 0
Fin de l'instruction conditionnelle
	`if a==1 then`
	`afficher(" X =  "+string(k))`
	`else`
	`afficher("……")`
	`end`

- Enrichir ce programme pour simuler **N** expériences. On définira **X(k)** qui donne le nombre de fois que le premier succès s'est produit au rang **k**, et **X0** qui donne le nombre de fois que le succès ne s'est pas produit et, on tracera le diagramme en barres de ces valeurs. On prendra **N=1000** pour commencer.

✐ Corrigé :

- Voir le fichier Scilab.
- L'exécuter.

```
◗ n=50;p=0.3;N=1000;
X=zeros(1,n); X0=0;
for i=1:N
    k=0;a=0;
    while(a==0 & k<n)
        a=floor(rand()+p);
        k=k+1;
    end
    if a==1 then
        X(k)=X(k)+1;
    else
        X0=X0+1;
    end
end
X=[X0,X]/N;
Y=loi_geometrique(n,p);
clf;bar([0:n],[X',Y'])
```

NOTION D'ESPÉRANCE

Exercice 1 (Fichier : Première-Esp-Ex 1.sce)

Approcher l'espérance par un calcul de moyenne fait sur un grand nombre de simulations, à partir des règles du jeu de la roulette. Utilisation combinée de tests et de boucles à proposer aux élèves en les guidant.

À la roulette, il y a 37 numéros de 0 à 36. On peut miser sur un numéro (sauf sur le 0), sur le pair (2,4,…36), sur l'impair (1,3,…35), ou sur une douzaine (1-12, 13-24, 25-36). Si l'on mise sur un numéro gagnant, on récupère 35 fois sa mise. Si l'on mise sur pair ou impair et que l'on gagne, on récupère 1 fois sa mise. Si l'on mise sur une douzaine et que l'on gagne, on récupère 2 fois sa mise.

◗ Simuler 100 000 parties, en supposant que la mise `m` est de 1 € à chaque partie et que l'on a misé sur le 1 (ou impair ou douzaine 1-12) et calculer le gain moyen pour chaque façon de miser.
◗ Exécuter plusieurs fois le programme. Les résultats sont-ils significatifs ?

◗ Faire le calcul théorique de l'espérance de chacun des paris. Lequel est le moins défavorable ?

Corrigé :

◗ Simulation :

```
m=1; gN=0; gP=0; gD=0;
for n=1:100000
    T=tirage_entier(1,0,36);
    if T==1 then
        gN=gN+35*m;
    else
        gN=gN-m;
    end
    if pair(T)==%F then
        gP=gP+m;
    else
        gP=gP-m;
    end
    if 1<=T & T<=12 then
        gD=gD+2*m;
    else
        gD=gD-m;
    end
end
afficher("Gain moyen en jouant un numéro : "+string(gN/n))
afficher("Gain moyen en jouant pair ou impair : "+string(gP/n))
afficher("Gain moyen en jouant une douzaine : "+string(gD/n))
```

◗ Les réponses varient du simple au double malgré le grand nombre de tirages.

◗ Le calcul donne la même espérance de gain pour chaque pari :
(-1/37)*m

Exercice 2 (Fichier : Première-Esp-Ex 2.sce)

Simuler pour approcher une espérance. Conjecturer la méthode la moins coûteuse. Conforter un résultat théorique. On peut guider l'élève dans la première partie en classe et donner la deuxième à faire à la maison.

Une entreprise fabrique chaque jour 10 000 composants électroniques. Chaque composant présente un défaut avec la probabilité 0,002. Si le composant est repéré comme étant défectueux, il est détruit par l'entreprise et chaque composant détruit fait perdre 1 € à l'entreprise.

- Les composants sont contrôlés un à un et chaque contrôle coûte 0,1 €.
- Les composants sont regroupés par lots de 10 et on effectue un unique contrôle automatique de chaque lot qui coûte lui aussi 0,1 €. À l'issue de ce contrôle, le lot est accepté si tous les composants sont sains et globalement détruit si au moins l'un des 10 composants présente un défaut.

Dans les deux cas, on cherche le coût moyen journalier pour l'entreprise de ces dispositifs de contrôles et de destruction des composants défectueux.

- Définir un algorithme permettant de simuler les deux types de contrôle et de calculer leur coût. L'exécuter plusieurs fois.
- Utiliser l'espérance de la loi binomiale pour faire le calcul exact des coûts moyens.

✐ Corrigé :

```
//Le contrôle d'une pièce non défectueuse (p1=0.998)
//coûte 0.1€, une pièce défectueuse coûte 1.1€
N1=10000; p1=0.998; C1=0;
for n=1:N1
    a=floor(rand()+p1);
    if a==1 then
        C1=C1+0.1;
    else
        C1=C1+1.1;
    end
end
afficher(C1)
```

```
//Le contrôle d'un lot de 10 non défectueux (p2=0.998^10)
//coûte 0.1€, un lot de 10 défectueux coûte 10.1€
N2=N1/10; p2=0.998^10; C2=0;
for n=1:N2
    a=floor(rand()+p2);
    if a==1 then
        C2=C2+0.1;
    else
        C2=C2+10.1;
    end
end
afficher(C2)
```

- Dans le premier cas, il y a en moyenne $10000 \times 0,002 = 20$ pièces défectueuses. Donc le coût journalier moyen est : $10000 \times 0,1 + 20 \times 1 = 1020€$
Dans le deuxième cas, il y a en moyenne $1000 \times (1- 0,99810) = 19,82$ lots défectueux. Donc le coût journalier moyen est : $1000 \times 0,1 + 19,82 \times 10 = 298€$

Exercice 3 **(Fichier : Première-Esp-Ex 3.sce)**

Utiliser la possibilité offerte par Scilab de faire des tirages dans des ensembles (donc sans ordre ni remise) pour approcher l'espérance de gain au 421.
Permettre de valider un calcul théorique un peu compliqué. Cet exercice, à la syntaxe nouvelle, est plutôt à faire par le professeur devant la classe.

Nous allons jouer au 421. On lance 3 dés équilibrés.

- Si l'on sort 421, on gagne 10€,
- Si l'on sort 2 des 3 chiffres seulement, on gagne 2€,
- Sinon, on ne gagne rien.

Combien peut-on espérer gagner en moyenne sur 1 000 parties ?

- Écrire le programme correspondant à l'algorithme suivant (on peut aussi demander aux élèves de définir l'algorithme) :
 Initialiser la somme gagnée S à 0
 Définir les ensembles gagnants
 Pour k allant de 1 à 1000
   Simuler le lancer de 3 dés et définir l'ensemble obtenu

Si cet ensemble contient les 3 chiffres gagnants, alors
S augmente de 10
ou si cet ensemble contient 2 des chiffres gagnants, alors
S augmente de 2
Fin de si
Fin de pour
Afficher S/1000

◗ Calculer l'espérance théorique de ce jeu et comparer avec les résultats trouvés.

Corrigé :

◗ Algorithme :

```
S=0;
T=ensemble("1","2","4"); D1=ensemble("1","2");
D2=ensemble("1","4");D3=ensemble("2","4")
for k=1:1000
    J=tirage_entier(3,1,6);
    JJ=ensemble(string(J));
    if inclus(T,JJ)==%T then
        S=S+10;
    elseif inclus(D1,JJ)==%T | inclus(D2,JJ)==%T | inclus(D3,JJ)==%T then
        S=S+2;
    end
end
afficher(S/1000)
```

À noter
La simulation par la commande **`tirage_entier`** donne une liste de 3 nombres. Il faut la transformer en ensemble. Pour cela on utilise **`string`**, qui transforme un nombre en chaîne de caractères, comme on le fait dans les affichages.

◗ Le calcul théorique se fait en comptant ordre et remise, il n'est pas très simple :
Il y a 6^3 = 216 issues dont
6 pour avoir 421 (3 ordres possibles)
72 pour avoir 2 des 3 chiffres (3 × 24)

en effet pour les 2 chiffres 1 et 2 corrects, on a
6 issues avec 2 chiffres égaux, type 112 ou 122
18 issues avec 3 chiffres différents, type 123 (on exclut 124)
L'espérance est donc égale à (6 × 10 + 72 × 2) / 216 = 0,94.

INTERVALLES DE FLUCTUATION

Exercice 1 (Fichier : Première-Fluct-Ex 1.sce)

Reprise sous une forme détournée de l'exercice classique sur l'élection du Président. Peut servir d'introduction faite par le professeur à la notion d'intervalle de confiance. Utilise la fonction de répartition de la loi binomiale.

La méchante reine aimerait bien savoir si elle est plus belle que Blanche Neige. N'ayant pas confiance en son miroir, elle a fait interroger 100 sujets, et 52 ont voté pour Blanche Neige. Elle a fait ses calculs, et a conclu qu'au risque de 5%, Blanche Neige n'était pas sûre de l'emporter.
Quels calculs a-t-elle fait ? A-t-elle eu raison ? Combien aurait-elle dû interroger de sujets pour s'inquiéter vraiment ?

✐ **Corrigé :**

◗ Quels calculs a-t-elle fait ?

Elle s'est dit qu'elle pouvait considérer que le nombre B de sujets préférant Blanche Neige suivait une loi binomiale de paramètres 100 et 0,52.
Qu'au risque de 5 %, elle allait chercher l'intervalle contenant les 95 % restants, en éliminant 2,5 % de chaque côté.
Avec Scilab, elle a fait établir la fonction de répartition **`FB`** correspondant à **`B`**.
Puis elle a recherché le plus petit entier m1 tel que p (B <= m) > 0,025 et le plus petit entier M1 tel que p (B <= M) >= 0,975.

Son intervalle de confiance était [m1/100 ; M1/100]

```
N=100; p=0.52;
RB=rep_binomiale(N,p);
clf; quadrillage; bar(RB)
m=find(RB>0.025); M=find(RB>=0.975);
afficher([(m(1)-1)/N,(M(1)-1)/N])
```

```
//Remarque : on doit enlever 1 car si m(1) est le rang 43,
//alors RB(43)= p(X<= 42)

afficher([p-1/sqrt(N),p+1/sqrt(N)])
```

◗ A-t-elle eu raison ?

Oui : elle a trouvé **`m1=42`**, et **`M1=62`**, donc **`I=[0.42;0.62]`**
Donc **`I`** contient des valeurs inférieures à 50 %, il n'est pas dit que Blanche Neige l'emporte au risque de 5 %.
La sorcière a bien conclu, sachant qu'elle a fait deux approximations :
- Elle a considéré qu'il y avait remise des sujets interrogés,
- Elle a fait une hypothèse de symétrie qui n'est totalement vraie que si p = 0,5.
- Elle a vérifié si cela correspondait à ce que l'on lui avait appris à l'école :
`I=[0.52-1/10;0.52+1/10]`

◗ Combien aurait-elle dû interroger de sujets pour s'inquiéter vraiment ?

Elle se rend bien compte que son intervalle est trop grand pour être vraiment significatif. Elle retourne à ses calculs, et reprend son programme en faisant varier la valeur **`N`** du nombre de sujets interrogés.
On reprend le même programme avec :

```
N=1000, puis N=2000, …
```

Elle se dit ensuite qu'elle va faire un test pour savoir si la borne **`m1/N`** peut devenir supérieure à 0,5 et à quel moment.

```
for N=2000:3000
    RB=rep_binomiale(N,0.52);
    m=find(RB>0.025);
    if (m(1)-1)/N>=0.5 then
        afficher(N)
        break
    end
end
```

Elle a trouvé 2350.
Si elle avait fait le calcul avec $\frac{1}{\sqrt{N}}$, elle aurait trouvé $N > \left(\frac{1}{0,02}\right)^2$, soit $N > 2500$

Exercice 2 (Fichier : Première-Fluct-Ex 2.sce)

Reprendre le calcul de l'exercice 1 dans une autre situation. Peut être posé en contrôle après avoir travaillé l'exercice 1.

Un groupe de citoyens demande à la municipalité d'une ville la modification d'un carrefour en affirmant que 40 % des automobilistes tournent en utilisant une mauvaise voie. Un officier de police constate que sur 500 voitures prises au hasard, 190 prennent une mauvaise voie.

- Déterminer, en utilisant la loi binomiale sous l'hypothèse **`p=0.4`**, l'intervalle de fluctuation au seuil de 95 %.
- D'après l'échantillon, peut-on considérer, au seuil de 95 %, comme exacte l'affirmation du groupe de citoyens ?

Corrigé :

-
```
N=500;p=0.4;
RB=rep_binomiale(N,p);

m=find(RB>0.025); M=find(RB>=0.975);
afficher([(m(1)-1)/N,(M(1)-1)/N])
```

- On trouve **`[0.358 ; 0.444]`**.
  Comme **`f=0.38`**, l'affirmation est considérée comme exacte.

Fonctions Scilab utiles

POUR LES STATISTIQUES

- **`sum(n)`** retourne la somme des valeurs du vecteur n (sert à calculer un effectif total).
- **`cumsum(n)`** retourne le vecteur des valeurs cumulées croissantes du vecteur n (sert à calculer les effectifs cumulés croissants).
- **`taille(v)`** retourne le nombre de coordonnées du vecteur v.
- **`trier(v)`** ou **`trier(v,">")`** retourne trié le vecteur de nombres ou de chaînes de caractères v dans l'ordre croissant.
- **`trier(v,"<")`** retourne trié le vecteur de nombres ou de chaînes de caractères v dans l'ordre décroissant.
- **`moyenne(v)`** retourne la moyenne du vecteur de nombres v.
- **`moyenne_ponderee(v,n)`** retourne la moyenne du vecteur de nombres v pondérée par le vecteur de nombres n avec v et n de même dimension et n un vecteur de nombres positifs ou nuls mais non tous nuls.
- **`ecart_type(v)`** retourne l'écart type du vecteur de nombres v.
- **`ecart_type_pondere(v,n)`** retourne l'écart type du vecteur de nombres v pondéré par le vecteur de nombres n avec v et n de même dimension et n un vecteur de nombres positifs ou nuls mais non tous nuls.
- **`variance(v)`** retourne la variance du vecteur de nombres v.
- **`variance_ponderee(v,n)`** retourne la variance du vecteur de nombres v pondérée par le vecteur de nombres n avec v et n de même dimension et n un vecteur de nombres positifs ou nuls mais non tous nuls.
- **`mediane(v)`** retourne la médiane du vecteur de nombres v.
- **`mediane_ponderee(v,n)`** retourne la médiane du vecteur de nombres v pondérée par le vecteur de nombres n avec v et n de même dimension et n un vecteur de nombres positifs ou nuls mais non tous nuls.
- **`quartiles(v)`** retourne les deux quartiles du vecteur de nombres v.
- **`quartiles_ponderes(v,n)`** retourne les deux quartiles du vecteur de nombres v pondérés par le vecteur de nombres n avec v et n de même dimension et n un vecteur de nombres positifs ou nuls mais non tous nuls.
- **`deciles(v)`** retourne les déciles D1 et D9 du vecteur de nombres v.
- **`deciles_ponderes(v,n)`** retourne les déciles D1 et D9 du vecteur de nombres v pondérés par le vecteur de nombres n avec v et n de même dimension et n un vecteur de nombres positifs ou nuls mais non tous nuls.
- **`bar(v,n,couleur)`** trace le diagramme en barre avec v en abscisse, n en ordonnée, v et n étant des vecteurs lignes de même dimension. Par défaut, **`bar(n)`** trace le diagramme en barres de n en bleu avec 1,2,3… en abscisses.

◗ **bar(v,[n1',n2'])** trace un diagramme en barre double avec v en abscisse, n1 en ordonnée en bleu et n2 en ordonnée en vert, avec v, n1 et n2 vecteurs lignes de même dimension.
◗ **histogramme(a,n,couleur)** permet de tracer l'histogramme d'une série statistique où les valeurs de la variable sont regroupées dans des intervalles. a est le vecteur donnant les bornes des intervalles dans l'ordre croissant. n est le vecteur des effectifs ou des fréquences correspondants. **couleur** (argument optionnel) définit la couleur comme dans la fonction **plot**.

POUR SIMULER

◗ **tirage_entier(p,m,n)** retourne un vecteur de p tirages entiers pris entre m et n avec p entier positif, m et n entiers et $m \leq n$.
◗ **tirage_reel(p,a,b)** retourne un vecteur de p tirages réels pris entre a et b avec p entier positif, a et b réels et $a \leq b$.
◗ **rand(n,p)** avec n avec p entiers positifs, retourne une matrice $n \times p$ de nombres pris aléatoirement entre 0 et 1.
◗ **rand()** retourne un nombre réel pris aléatoirement entre 0 et 1.
◗ **floor(x)** retourne la partie entière du nombre réel x.
En particulier, si p est un réel compris entre 0 et 1, **floor(rand()+p)** vaudra 1 avec une probabilité p et 0 avec une probabilité $1-p$.
◗ **frequence(n,s)** retourne la fréquence de n dans la suite de nombres s avec n entier.
◗ **frequence_tirage_entier(p,m,n)** retourne la suite des fréquences de p tirages entiers pris entre m et n avec p entiers positif, m et n entiers et $m \leq n$.
◗ **ensemble("r(1)","r(2)","r(3)","v(1)","v(2)")** définit un ensemble, ici l'ensemble de trois boules rouges numérotées 1, 2, 3 (leurs valeurs) et deux vertes numérotées 1, 2 (leurs valeurs).
◗ **tirage_ensemble(n,ens)** retourne un ensemble de n éléments pris parmi ceux de l'ensemble **ens**.

POUR DÉFINIR DES LOIS

◗ **factorielle(n)** retourne la factorielle de n avec n entier positif ou nul.
◗ **arrangement(n,p)** retourne le nombre d'arrangements de p éléments pris parmi n avec n et p entiers positifs ou nuls et $p \leq n$.

- **`combinaison(n,p)`** retourne le nombre de combinaisons de p éléments pris parmi n avec n et p entiers positifs ou nuls et $p \leq n$.
- **`loi_binomiale(n,p)`** où n est un entier positif et p un réel entre 0 et 1 retourne le vecteur ligne des probabilités $p(X=k)$, pour k allant de 0 à n, lorsque X suit la loi binomiale de paramètres n et p. On peut aussi utiliser **`binomial(p,n)`**.
- **`loi_binomiale(n,p,k)`** avec k entier entre 0 et n retourne la probabilité $p(X=k)$ lorsque X suit la loi binomiale de paramètres n et p.
- **`rep_binomiale(n,p)`** retourne le vecteur ligne des probabilités cumulées $p(X \leq k)$ pour k allant de 0 à n, lorsque X suit la loi binomiale de paramètres n et p.
- **`rep_binomiale(n,p,t)`** retourne la probabilité $p(X \leq t)$ lorsque X suit la loi binomiale de paramètres n et p.
- **`loi_geometrique(n,p)`** où n est un entier positif et p un réel entre 0 et 1 retourne le vecteur ligne des probabilités $p(X=k)$, pour k allant de 0 à n, lorsque X suit la loi géométrique tronquée de paramètres n et p.
- **`loi_geometrique(n,p,k)`** avec k entier entre 0 et n retourne la probabilité $p(X=k)$ lorsque X suit la loi géométrique tronquée de paramètres n et p.
- **`loi_exp(lambda,t)`** retourne la probabilité $p(X \leq t)$ lorsque X suit la loi exponentielle de paramètre λ avec λ positif.
- **`loi_normale(t,xbar,sigma)`** retourne la probabilité $p(X \leq t)$ lorsque X suit la loi normale de paramètres $\overline{x}$ et σ avec σ positif.

POUR AFFICHER ET TRACER

- **`clf`** signifie « clear figure » et efface la figure présente sur la fenêtre graphique.
- **`quadrillage`** fait apparaître un quadrillage dans la fenêtre graphique.
- **`plot(X,Y,"*")`** trace le nuage des points de coordonnées (X(i),Y(i)) sous forme d'étoiles. On peut préciser la couleur.
- **`plot(Y,"+")`** trace le nuage des points de coordonnées (i,Y(i)) sous forme de croix.
- **`afficher("Phrase")`** affiche ce qui est écrit entre les guillemets.
- **`afficher(A)`** où A est une matrice de nombres affiche le tableau des valeurs de A.
- **`afficher("Phrase"+string(x))`** affiche la phrase et la valeur du nombre x.

MINISTÈRE
DE L'ÉDUCATION NATIONALE
reconnu d'intérêt pédagogique

2011

digiteo
Recherche en sciences & technologies de l'information

www.scilab.org