Correction du TP1

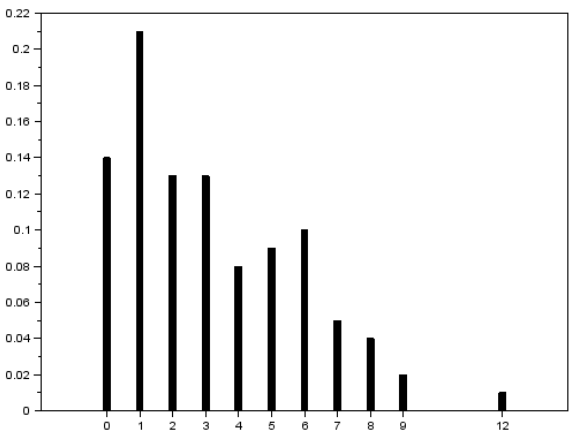
5. Exercices

Exercice 1.
On souhaite étudier la loi d'une variable $Y = |\lfloor X \rfloor|$ où $X \hookrightarrow \mathcal{N}(0,4)$. Pour cela, on réalise 100 simulation de la loi de Y et on construit le diagramme en bâton de l'échantillon obtenu , entre autres :

1. X suit une loi normale (loi à densité) donc est une variable continue, en revanche, une partie entière est un entier donc $\lfloor X \rfloor$ prend des valeurs entières. Ainsi, $Y = |\lfloor X \rfloor|$ prend des valeurs entières positives. C'est donc aussi une variable aléatoire discrète.
2.
```
// Création d'une matrice de taille 1 *100 dont les coefficients sont 100 simulations d'une
//normale de paramètres 0 et 4:

-->X=grand(1,100,'nor',0,4);

// Création d'une matrice de taille 1*100 dont les coefficients sont 100 simulations de
// la variable Y:

-->Y=abs(floor(X));
```
On simule 100 réalisations d'une variable aléatoire donc les résultats diffèrent d'un étudiant à l'autre.
Les effectifs que vous obtiendrez ne seront pas les mêmes que les miens. Les effectifs, les fréquences et le diagramme en bâton diffèreront aussi.
3. • La moyenne empirique de l'échantillon des simulations de Y : `-->m=mean(Y)` est une approximation de l'espérance de Y. On obtient $E(Y) \simeq 3,3$.
• La variance empirique de l'échantillon des simulations de Y : `-->v=mean(Y.^2)-mean(Y)^2` est une approximation de la variance de Y. On obtient $V(Y) \simeq 6$.
4.
```
// Création du tableau modalité-effectif de la série statistique Y:

-->M=tabul(Y,"i")
 M  =

    0.     14.
    1.     21.
    2.     13.
    3.     13.
    4.     8.
    5.     9.
    6.     10.
    7.     5.
    8.     4.
    9.     2.
    12.    1.
```
La première colonne est la liste des modalités de Y, la seconde est la liste des effectifs correspondants.
5.
```
// Création de la liste x=[x1,x2,....,xp] des modalités de Y:

-->x= M(:,1)
 x  =

    0.
    1.
    2.
```

```
    3.
    4.
    5.
    6.
    7.
    8.
    9.
    12.

// Création de la liste n=[n1,n2,....,np] des effectifs de Y:

-->n=M(:,2)
 n  =

    14.
    21.
    13.
    13.
    8.
    9.
    10.
    5.
    4.
    2.
    1.

// Création de la liste des fréquences de Y:

-->f=n/sum(n) // ou f=n/100
 f  =

    0.14
    0.21
    0.13
    0.13
    0.08
    0.09
    0.1
    0.05
    0.04
    0.02
    0.01
```

6.
```
// Création du diagramme en bâton de la série statistique Y: on peut changer la largeur
// des bandes par l'instruction width (0,8 par défaut) et la couleur par color (bleu par défaut)

-->bar(x,f, width=0.2, color='black')
```



On peut faire une estimation de $P(Y = k)$ en considérant la fréquence d'apparition du nombre

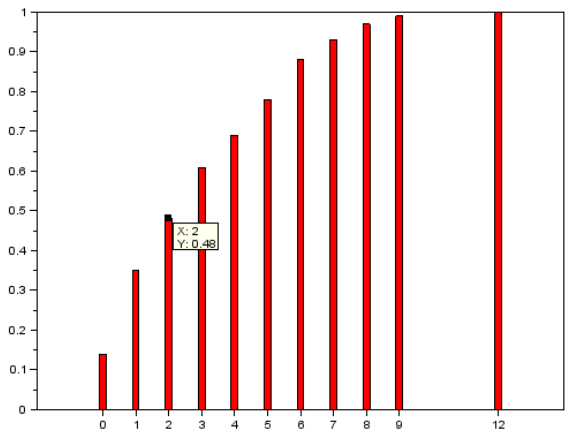
k sur un grand nombre de simulations de Y (ici, on fait 100 simulations, ce n'est pas vraiment suffisant....)
Le diagramme en bâton donne dont $P(Y=0) \simeq 0,14$, $P(Y=1) \simeq 0,21$, $P(Y=8) \simeq 0,05$.

7. D'après le diagramme en bâton, le mode de Y est la modalité 1.
La médiane de Y est la valeur m pour laquelle moins de 50% de l'échantillon a des vaelurs strictement inférieures à m et moins de 50% a des valeurs strictement supéreiues à m .
d'après le diagramme en bâton, on a $m = 3$.
On peut le vérifier en tapant `median(Y)`.

La moyenne et la médiane sont très proches, ce qui est logique car les valeurs de Y sont condensées autour de 3, il n'y a pas de valeurs extrémales importantes à prendre en compte.

8.
```
// Construction du vecteur des fréquences cumulées de Y :

-->F=cumsum(f)
 F  =

    0.14
    0.35
    0.48
    0.61
    0.69
    0.78
    0.88
    0.93
    0.97
    0.99
    1.
```

9.
```
// construction du diagramme des frequences cumulees :

-->scf(1), bar(x,F,width=0.2,color='red')
```



La lecture du tableau F ou du diagramme des fréquences cumulées donne $m = 3$, $d_1 = 0$, $d_2 = 1$, $d_3 = 1$, $d_4 = 2$, $d_5 = 3$, $d_6 = 3$, $d_7 = 5$, $d_8 = 6$, $d_9 = 7$, $d_{10} = 12$.

Exercice 2.

1.
```
function x=simulX()
        if rand() < 1/3 then
            x=grand(1,1,'nor',13,2)
        else
            x=grand(1,1,'nor',7,2)
        end
endfunction
```

Bien sûr, lorsqu'on définit une fonction, on ne fait qu'indiquer à Scilab qu'il a une nouvelle fonction à sa disposition, il ne se passe donc rien à l'exécution, si ce n'est que Scilab mémorise cette nouvelle définition.
Si on veut utiliser la fonction, il faut faire appel à cette fonction par la commande `-->simulX()`
Scilab vous renvoit alors le résultat x d'une simulation de X.

2.
```
y=zeros(1,100)
for k=1:100 do
    y(k)= floor(simulX())
end
disp(y)
```
Après exécution, on obtient :
```
y  =


         column  1 to 19

    9.    6.    8.    11.    10.    7.    7.    12.    12.    7.    11.    8.    10.    5.
    7.    10.    7.    4.    16.

         column 20 to 38

    9.    5.    6.    14.    7.    5.    8.    10.    6.    7.    6.    8.    10.    6.    5.
   5.    10.    13.    8.

         column 39 to 56

    7.    5.    6.    9.    10.    16.    12.    9.    13.    11.    15.    5.    11.    12.
   16.    12.    7.    12.

         column 57 to 75

    9.    6.    7.    5.    11.    3.    8.    7.    11.    6.    13.    15.    12.    8.    12.
  5.    14.    13.    14.

         column 76 to 94

    3.    11.    4.    4.    11.    11.    4.    9.    10.    7.    12.    16.    7.    8.    5.
 3.    12.    10.    6.

         column  95 to 100

    12.    8.    7.    5.    4.    14.
```

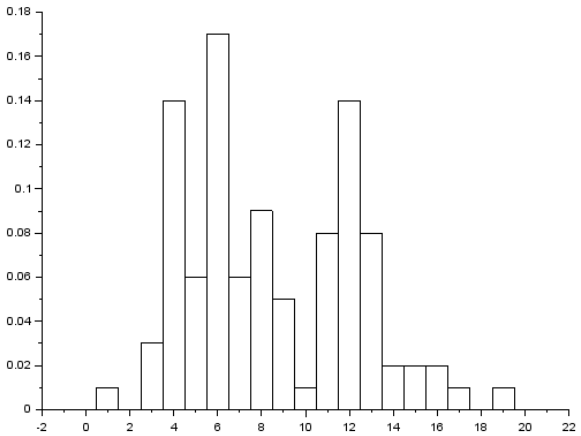
3. — `// Moyenne de Y:`
```
-->m=mean(y)
 m  =

    8.85

// Ecart type de Y:

-->sigma=sqrt(mean(y.^2)-mean(y)^2)
 sigma  =

    3.376613
```
— `// Rangement des valeurs de Y dans l'ordre croissant :`
```
-->gsort(y,'g','i')
 ans  =


         column  1 to 20

    3.    3.    3.    4.    4.    4.    4.    4.    5.    5.    5.    5.    5.    5.    5.
   5.    5.    5.    5.    6.
```

```
        column 21 to 40

   6.    6.    6.    6.    6.    6.    6.    6.    7.    7.    7.    7.    7.    7.    7.
 7.    7.    7.    7.    7.

        column 41 to 60

   7.    7.    8.    8.    8.    8.    8.    8.    8.    8.    8.    9.    9.    9.    9.
  9.    9.    10.    10.    10.

        column 61 to 77

   10.    10.    10.    10.    10.    10.    11.    11.    11.    11.    11.    11.
  11.    11.    11.    12.    12.

        column 78 to 94

   12.    12.    12.    12.    12.    12.    12.    12.    12.    13.    13.    13.
  13.    14.    14.    14.    14.

        column  95 to 100

   15.    15.    16.    16.    16.    16.
```

On lit alors : $m = 8$, $Q_1 = 6$,et $Q_3 = 11.5$.

— `// Construction de l'histogramme de Y :`

```
-->histplot(-0.5:20.5,y)
```



4. La moyenne et la médiane indiquaient déjà que les valeurs seraient centrées autour de $8 - 9$. L'écart-type et l'intervalle interquartile : $Q_3 - Q_1 = 5,5$ indiquent que les données sont plutôt dispersées d'environ $4 - 5$ unités autour de la valeur centrale.

   En revanche, ces données n'indiquaient pas que les valeurs de fréquences maximales (dont le mode) se trouvaient à 2 unités de la valeur centrale et que la valeur centrale forme plutôt un creux. L'évolution des fréquence entre les différentes valeurs n'était pas non plus représentée.

Exercice 3.
Simulation d'une loi usuelle :

1. Soit l'épreuve e : "lancer un dé " et l'évènement A : "obtenir 6" que l'on considère comme le succès de l'épreuve e. P(A)=1/6
   On réalise une succession de 6 épreuves e indépendantes et X compte le nombre de succès donc X suit une loi binomiale de taille 6 et de paramètre 1/6.
2. `// Simulation de 40 simulations [x(1), x(2), ...,x(40)] indépendantes suivant la loi B(6,1/2) :`

```
-->x=grand(1,40,'bin',6,1/2)
 x  =


         column  1 to 20

    2.    2.    4.    4.    1.    3.    3.    3.    0.    2.    4.    4.    3.    3.    3.    3.
  2.    3.    2.    3.

         column 21 to 40

    3.    1.    4.    2.    1.    2.    2.    2.    3.    2.    4.    3.    5.    1.    4.    5.
    2.    5.    1.    3.
```

3. `[I,O,N]=dsearch(x,[-0.5:1:6.5])`

```
 N  =

    0.
 O  =

    1.    5.    11.    14.    5.    4.    0.
 I  =


         column  1 to 20

    3.    4.    4.    3.    5.    6.    4.    4.    2.    3.    3.    5.    5.    4.    3.    3.
  3.    4.    4.    2.

         column 21 to 40

    3.    3.    4.    6.    6.    6.    4.    4.    5.    2.    4.    3.    4.    4.    2.    5.
  3.    1.    4.    2.
```

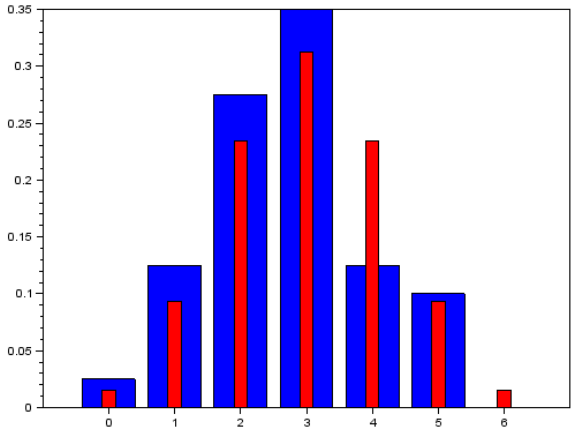
$N = 0$ signifie qu'aucune valeur de X est en dehors des classes $[-0,5;0,5[$, $[0,5;1,5[$, ...$[5,5;6,5[$. C'est forcément le cas puisque le support de la binomiale est $X(\Omega) = [\![0;6]\!]$.
D'ailleurs, appartenir à la classe $[-0,5;0,5[$ signifie valoir 0 par exemple, puisque une binomiale prend des valeurs entières.
O donne l'effectif de chaque classe : l'effectif de la modalité 0 (de la classe modale $[-0,5;0,5[$) est 1, l'effectif de la modalité 1 (de la classe modale $[0,5;1,5[$) est 5,etc....ce qui se calcule à la main en comptant les modalités et les effectifs de X obtenu ci-avant.

4. `// Calcul des fréquences des modalités de X:`

```
-->f=O/sum(O) // ou f=O/40
 f  =

    0.025    0.125    0.275    0.35    0.125    0.1    0.

--> // diagramme en barre de la série statistique X:

-->bar(0:6,f,width=0.8)
```

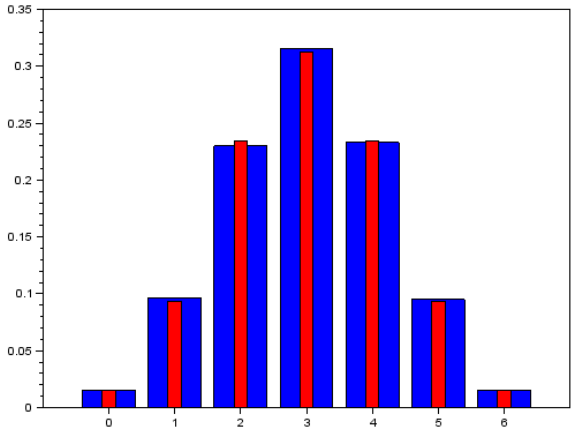
5. `// Création du vecteur de la loi théorique d'une B(6,1/2):`

```
-->p=binomial(1/2,6)
 p  =

    0.015625    0.09375    0.234375    0.3125    0.234375    0.09375    0.015625

-->// Diagramme en bâton de la loi de X:

->bar(0:6,p,width=0.2,color='red')
```



Les graphiques ne se superposent pas entièrement car le nombre de simulations de la binomiale est insuffisant pour que la fréquence d'apparition d'une valeur k soit proche de la probabilité théorique $P(X = k)$.
Il faut recommencer l'expérience avec 10000 simulations de la binomiale par exemple pour obtenir une loi empirique (diagrammes des fréquences) proche de la loi théorique :

```
-->x=grand(1,10000,'bin',6,1/2);

-->[I,O,N]=dsearch(x,[-0.5:1:6.5]);

-->f=O/sum(O) // ou f=O/10000
 f  =

    0.015    0.0963    0.2299    0.3153    0.233    0.0949    0.0156

-->clf ,bar(0:6,f,width=0.8)

-->bar(0:6,p,width=0.2,color='red')
```



Exercice 4.

1.
```
function s=SUM(M)
    [n,m]=size(M)
    s=0
    for i=1:n do
        for j=1:m do
            s=s+M(i,j)
```

```
        end
    end
endfunction
```

2.
```
function c=CUMSUM(x)
    n=length(x)
    c=zeros(1,n)
    c(1)=x(1)
    for i=2:n do
        c(i)=c(i-1)+x(i)   // ou c(i) = SUM(x(1:i))
    end
endfunction
```

3.
```
function m=MEDIANE(x)
    y=gsort(x,'g','i')
    n=length(y)
    if floor(n/2)==n/2 then
        m=(y(n/2)+y(n/2+1))/2
    else
        m=y((n+1)/2)
    end
endfunction
```