DS10

Devoir Surveillé du 09/03/2020

Durée : 1h00

La qualité de la rédaction, la clarté et la précision des raisonnements interviendront pour une part importante dans l'appréciation des copies. Les résultats doivent être encadrés.
La calculatrice n'est pas autorisée.

Exercice 1

Doudou le hamster n'a que trois occupations dans sa cage : il dort (état 1), il court dans sa roue (état 2), et il mange (état 3). Chaque minute, il change d'activité ou continue celle qu'il était en train de faire. On suppose que :

- s'il dort à un instant donné, il y a 9 chances sur 10 qu'il dorme encore la minute suivante, mais s'il se réveille, il va soit manger, soit courir, les deux étant aussi probables l'un que l'autre.
- s'il mange, il y a 3 chances sur 10 qu'il parte dans sa roue, et 7 chances sur 10 qu'il retourne se coucher.
- s'il court, il y a 8 chances sur 10 qu'il aille se coucher à l'instant suivant, et sinon, il court encore un peu.

Supposons que Doudou dort à l'instant 0, et notons X_0 la variable aléatoire certaine égale à 1, et pour tout n, notons X_n la variable aléatoire indiquant l'activité de Doudou à la n-ième minute.

1. Soit $A = (a_{i,j}) \in \mathscr{M}_3(\mathbb{R})$ définie par $a_{i,j} = P_{[X_n=j]}(X_{n+1} = i)$. Déterminer explicitement A.
2. On rappelle que l'instruction `grand(n,"markov",A',1)` simule n minutes d'activités d'un Doudou initialement au repos. On considère le script suivant :

```
n = 100
c = [0,0,0]
for i = 1:3
      for k = 1:10000
            a = grand(n,"markov",A',1)
            if a(n) == i then
                  c(i) = c(i)+1
            end
      end
end
disp(c/10000)
```

   En l'exécutant, on obtient :

```
0.8813     0.045     0.0737
```

   Qu'effectue ce programme ? En particulier, que contient la variable `c` ?

3. En exécutant le script suivant :

```
[P,D] = spec(A)
P(:,1) = P(:,1)/sum(P(:,1))
disp(P,'P=',D,'D=')
```

   on obtient :

```
D=

    1.     0               0
    0      0.05 + 0.05i    0
    0      0               0.05 - 0.05i
```

```
P=

    0.8839779   - 0.4743416 - 0.1581139i   - 0.4743416 + 0.1581139i
    0.0441989   - 0.3162278 + 0.1581139i   - 0.3162278 - 0.1581139i
    0.0718232     0.7905694                  0.7905694
```

Commenter. Quel résultat vu en TP observe-t-on ici ? Que dire de la suite (X_n) ?

Exercice 2

1. Soit $\lambda > 0$. On admet que si $U \hookrightarrow \mathscr{U}([0,1])$, alors la variable $X = -\dfrac{1}{\lambda}\ln(1-U)$ suit une loi exponentielle de paramètre λ.

   Écrire une fonction `exponentielle(lbd)` simulant une loi $\mathscr{E}(\lambda)$ à partir de la fonction `rand()`.

2. Soit $\lambda > 0$. On admet que si $X \hookrightarrow \mathscr{E}(\lambda)$, alors $Y = \lfloor X \rfloor + 1 \hookrightarrow \mathscr{G}(1-e^{-\lambda})$.

   (a) Écrire une fonction `geom(p)` simulant une loi $\mathscr{G}(p)$ à partir de la fonction `exponentielle`.

   (b) On souhaite construire un vecteur `x` de taille 10000 contenant 10000 simulations de la loi $\mathscr{G}(0.2)$. Compléter pour cela le programme suivant :

```
x = ...
for k = 1:10000
     x(k) = ...
end
```

   (c) En exécutant l'instruction suivante :

```
disp(mean(x), stdev(x)^2)
```

   on obtient :

```
ans =
       5.0404
ans =
       20.133581
```

   Ces résultats sont-ils conforme à vos attentes ? Expliquer.

   (d) On exécute les instructions suivantes :

```
p = 0.2 ; v = p*(1-p)^[0:29]
```

   Que contient `v(i)` pour $i = 1, \dots, 30$?

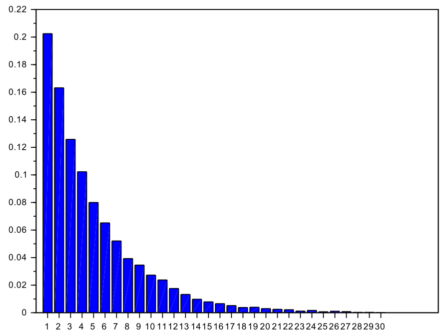
   (e) On rappelle les commandes suivantes :

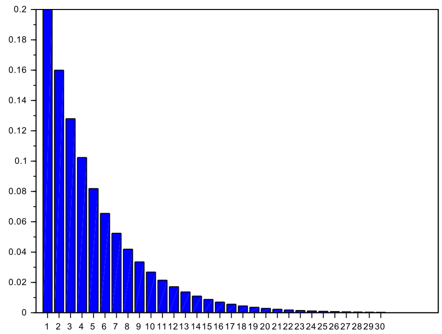
   - Si `x` est un vecteur, `y=tabul(x,"i")` est une matrice à deux colonnes, la première contenant les valeurs prises par les composantes de `x` rangées dans l'ordre croissant et la seconde contenant le nombre d'occurrences de chaque valeur.
   - Si `x` et `y` sont des vecteurs de même taille, `bar(x,y)` trace un diagramme en bâtons d'abscisse `x`, d'ordonnée `y`.

   En exécutant le script :

```
y = tabul(x,"i")
bar(y(:,1),y(:,2)/length(x))
scf()
bar([1:30],v)
```

on obtient les diagrammes en bâtons suivants :





Que contiennent les vecteurs `y(:,1)` et `y(:,2)/length(x)` ? À quoi correspondent chacun des diagrammes en bâtons ? Que peut-on conclure de ces diagrammes ?

Exercice 3

1. On a entré les commandes suivantes :

```
--> A = [1,4,2;3,2,8]; min(A,'r')
```

   et on a obtenu :

```
ans = 1.   2.   2.
```

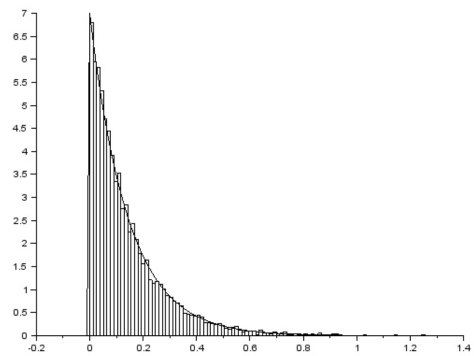
   Expliquer ce que fait la commande `min(A,'r')`.

2. Soit $n \in \mathbb{N}$ avec $n \geq 2$.

   Soit $X_1, X_2, \ldots, X_n$ n variables aléatoires indépendantes et suivant la même loi exponentielle de paramètre 1. On pose $I_n = \min(X_1, \ldots, X_n)$.

   (a) On considère le programme `Scilab` suivant qui renvoie le graphique ci-dessous :

```
n=7
X=grand(n,10000,'exp',1)
I=min(X,'r')
histplot(100,I)
function y=f(x)
      if x<0 then y=0
         else y=n*exp(-n*x)
      end
endfunction
x=-0.1:0.01:1
fplot2d(x,f)
```



   Expliquer en détails ce que fait ce programme.

   (b) Que peut-on conjecturer quant à la loi de I_n ?

Exercice 4

Soient a, b deux entiers strictement positifs. Une urne contient initialement a boules rouges et b boules blanches. On effectue une succession d'épreuves, chaque épreuve étant constituée des trois étapes suivantes :

- on pioche une boule au hasard dans l'urne,
- on replace la boule tirée dans l'urne,
- on rajoute dans l'urne une boule de la même couleur que celle qui vient d'être piochée.

Après n épreuves, l'urne contient donc $a+b+n$ boules. Pour tout $n \in \mathbb{N}^*$, on note X_n le nombre de boules rouges qui ont été **ajoutées** dans l'urne (par rapport à la composition initiale) à l'issue des n premières épreuves.

On souhaite simuler l'expérience grâce à `Scilab`.

1. Compléter la fonction suivante, qui simule le tirage d'une boule dans une urne contenant x boules rouges et y boules blanches et qui retourne la valeur 0 si la boule est rouge et 1 si elle est blanche.

```
function res = tirage(x,y)
    r = rand()
    if ......... then
       res = 0
    else
       res = 1
    end
endfunction
```

2. Compléter la fonction suivante, qui simule n tirages successifs dans une urne contenant initialement a boules rouges et b boules blanches (selon le protocole décrit ci-dessus) et qui retourne la valeur de X_n :

```
function Xn = experience(a,b,n)
    x = a
    y = b
    for k=1:n
        r = tirage(x,y)
        if r = = 0 then
           x = .........
        else
           ...........
        end
    end
    Xn = .........
endfunction
```

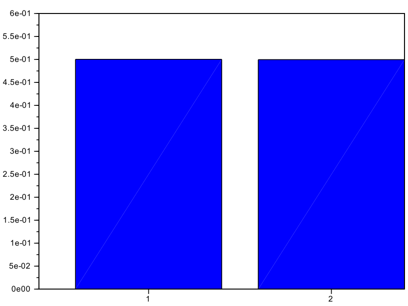
3. Écrire une fonction `Scilab` d'en tête :

```
function loi = simulation(a,b,n,m)
```

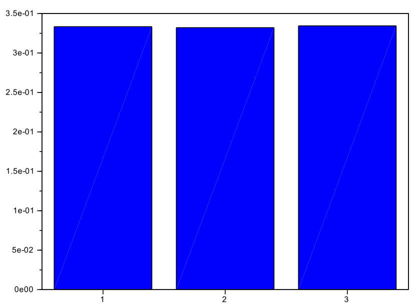
qui fait appel `m` fois à la fonction précédente pour estimer la loi de X_n. Le paramètre de sortie sera un vecteur contenant les approximations de $P(X_n=0), P(X_n=1), \ldots, P(X_n=n)$.

4. On s'intéresse ici au cas où $a=b=1$. On utilise la fonction `simulation` avec des valeurs de n entre 1 et 5 et on affiche à chaque fois l'estimation de la loi de X_n sous forme d'un diagramme en bâtons.

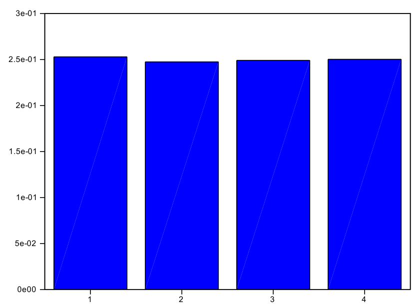
```
-> bar(simulation(1,1,1,100000))
```



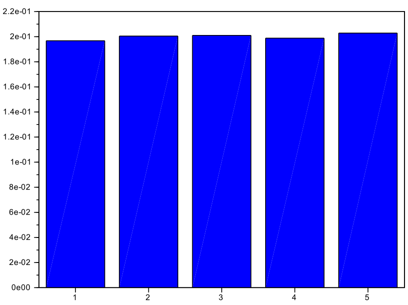
```
-> bar(simulation(1,1,2,100000))
```



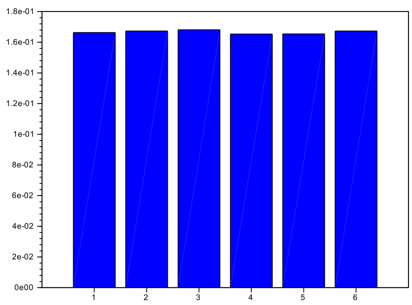
```
-> bar(simulation(1,1,3,100000))
```



```
-> bar(simulation(1,1,4,100000))
```



```
-> bar(simulation(1,1,5,100000))
```



À l'aide de ces résultats, conjecturer la loi de X_n.