2^{de}	DS n°8 : Probabilités	15 Mai 2018	*Barème prévisionnel : 2-3-3-4-4-4*

Bon courage ! Optimisez le temps : essayez de répondre à toutes les questions sans s'obstiner sur aucune. Quand vous passez, laissez un peu de place pour y revenir ultérieurement...

I] Course autour du monde

Douze motos solaires sont au départ du Prix de la Terre, fameuse et non fumeuse course autour du monde. Quelle chance a t-on de trouver au hasard le tiercé gagnant (les trois motos de tête, dans le bon ordre) ?

Il faut « choisir » la première moto (celle qui gagne), il y a 12 façons de faire ce choix. Il faut ensuite choisir la deuxième moto, il n'y a plus que 10 façons de faire ce choix (la moto qui a gagné ne peut arriver second). Pour la troisième moto, il n'y a plus que 9 façons de choisir. Il y a donc 12×11×10=1320 tiercés possibles.

Comme un seul gagnera, la probabilité est de $\frac{1}{1320}\approx 0{,}000758$.

II] Jeu de cartes

On dispose d'un jeu de 52 cartes (13 cartes dans 4 couleurs) dans lequel on puise au hasard 5 cartes.

a) Combien y a t-il de mains différentes de 5 cartes ?

Les mains de cinq cartes correspondent au choix de 5 cartes parmi 52, sans tenir compte de l'ordre. Pour la $1^{ère}$ carte on a 52 possibilités, puis on en a 51 pour le 2^{de}, etc. Il y a donc, en tout, 52×51×50×49×48=311 876 200 mains ordonnées (on a aussi bien compté la suite à cœur 12345 que la suite à cœur 54321 qui, contenant les mêmes cartes, ne sont pas distinctes).

Il faut diviser maintenant ce nombre de mains ordonnées par 5×4×3×2×1=120 (il y a 120 ordres différents des mêmes 5 cartes) : il y a donc 311 876 200÷120=2 598 960 mains équiprobables.

b) Quelle est la probabilité de l'événement A l'événement « on obtient les 4 as » ? Les mains qui réalisent le carré d'as (l'événement « on obtient les 4 as ») sont au nombre de 48 (on choisit la $5^{ème}$ carte parmi les 48 qui ne sont pas des as) donc $P(carré\ d'As)=\frac{48}{2598960}=\frac{1}{54145}\approx 0{,}0000185$.

III] Jeu florentin

Galilée et son ami Cosme jouent à lancer une pièce : Galilée lance deux fois la pièce et Cosme la lance trois fois. Ils comptabilisent leurs « pile ».

- Si Galilée en obtient plus que Cosme, Cosme donne 10 *écus* à Galilée ;
- Si Galilée en obtient autant que Cosme, Cosme donne 1 *écu* à Galilée ;
- Si Galilée en obtient moins que Cosme, Galilée donne 5 *écus* à Cosme.

Le jeu est-il à l'avantage de Galilée ou de Cosme ?

(Justifier avec un arbre de probabilité où chaque branche est associée au gain algébrique de Cosme)

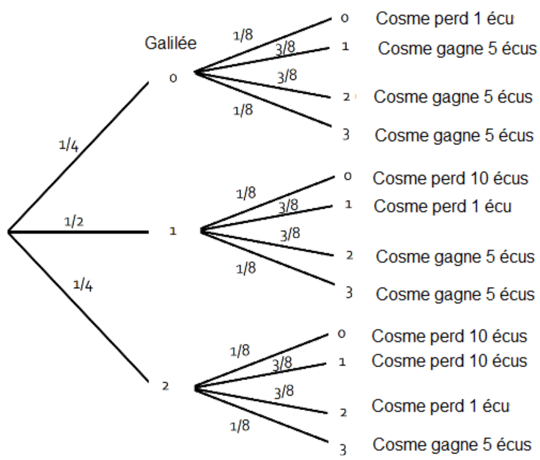
En lançant deux fois la pièce Galilée peut obtenir PP, PF, FP et FF, 4 résultats équiprobables qui amènent 0, 1 ou 2 piles. Les probabilités pour Galilée sont donc :

P(0 pile)= $\frac{1}{4}$, P(1 pile)= $\frac{2}{4}=\frac{1}{2}$ (PF ou FP) et P(2 piles)= $\frac{1}{4}$

Pour Cosme, il y a 8 résultats équiprobables qui amènent 0, 1, 2 ou 3 piles. Les probabilités pour Cosme sont donc :

P(0 pile)= $\frac{1}{8}$, P(1 pile)= $\frac{3}{8}$ (PFF, FPF ou FFP),

P(2 piles)= $\frac{3}{8}$ et P(3 piles)= $\frac{1}{8}$.



Dessinons l'arbre des probabilités et calculons :

P(Cosme gagne 5 *écus*)= $\frac{1}{4}\times\frac{7}{8}+\frac{1}{2}\times\frac{4}{8}+\frac{1}{4}\times\frac{1}{8}=\frac{1}{2}$,

P(Cosme perd 10 *écus*)= $\frac{1}{2}\times\frac{1}{8}+\frac{1}{4}\times\frac{4}{8}=\frac{3}{16}$,

P(Cosme perd 1 *écu*)= $\frac{1}{4}\times\frac{1}{8}+\frac{1}{2}\times\frac{3}{8}+\frac{1}{4}\times\frac{3}{8}=\frac{5}{16}$ ou $1-(\frac{3}{16}+\frac{1}{2})=1-\frac{11}{16}=\frac{5}{16}$ (car il s'agit du contraire de Cosme gagne 5 ou perd 10). Finalement, Cosme gagne en moyenne 0,3125 *écu* car $5\times\frac{1}{2}-10\times\frac{3}{16}-1\times\frac{5}{16}=\frac{25}{16}=0{,}3125$.

IV] Restaurant

Un certain nombre N de personnes sont dans un restaurant et commandent un menu avec entrée, plat et dessert. On a le choix entre 4 entrées, 3 plats et 2 desserts. Les choix étant aléatoires et indépendants, on s'interroge sur la probabilité p_N de l'événement E : « au moins deux commandes sont identiques » :

a) Combien de menus différents peut-il y avoir ? Notons C ce nombre.

On peut choisir 4×3×2=24 menus différents.

b) Déterminer la probabilité p_N lorsque $N=C+1$.

Lorsque $N=C$, au pire, les N personnes ont commandé les C menus différents et aucun n'a le même. Mais pour $N=C+1$, il y a forcément 2 personnes qui ont commandé le même menu. Donc probabilité p_{C+1} vaut 1 (l'événement est certain). Comme ici $C=24$, on a $p_{25}=1$.

c) Déterminer une formule donnant la probabilité p_N pour $N\leq C$

(*on peut commencer par déterminer la probabilité de F : « toutes les commandes sont différentes »*)

Si les N personnes font des commandes au hasard, il peut se trouver que ce soit n'importe quelle commande pour chacune des personnes, il y a donc $24\times24\times..\times24=24^N$ choix possibles pour la liste des N commandes, liste ordonnée par l'ordre des N personnes. Gardons-les ordonnées car toutes les listes n'ont pas la même probabilité [si il n'y a que 3 personnes, la liste non-ordonnée (1-1-2) qui peut s'écrire dans 3 ordres différents (112, 121 et 211), n'est pas aussi probable que la liste non-ordonnée (1-2-3) qui peut s'écrire dans 6 ordres différents (123, 132, 213, 231, 312 et 321)].

Combien y a t-il de listes où « deux commandes au moins sont identiques » ? Vous devez penser au contraire lorsque vous voyez écrit « au moins... ». Dans ce cas, le contraire de E : « deux commandes au moins sont identiques » est $\overline{E}$: « toutes les commandes sont différentes ». Pour avoir N commandes différentes, il y a $24\times23\times22\times21\times...\times(24-N+1)$ possibilités ordonnées.

La probabilité de $\overline{E}$ est $P(\overline{E})=\dfrac{24\times23\times22\times21\times...\times(24-N+1)}{24\times24\times24\times24\times...\times24}=\dfrac{24}{24}\times\dfrac{23}{24}\times...\times\dfrac{(24-N+1)}{24}$ et donc $p_N=P(E)=1-P(\overline{E})$. Pour 5 personnes on a $P(\overline{E})=\dfrac{24\times23\times22\times21\times20}{24^5}=\dfrac{5100480}{7962624}\approx0{,}64$, ce qui conduit à $p_5=P(E)=1-P(\overline{E})\approx0{,}36$, soit une probabilité assez prés de 50% mais il faut encore augmenter N pour arriver à dépasser 0,5.

d) Dresser un tableau donnant la probabilité p_N pour $N>1$, jusqu'à la 1ère valeur N_0 telle que $p_N>0{,}5$.

Nous avons utilisé le tableur pour réaliser ce calcul mais il suffit de calculer progressivement les valeurs de p_N en s'arrêtant quand on a dépassé 0,5.

n	1	2	3	4	5	6	7
24+1-n	24	23	22	21	20	19	18
(24-n)/24	1,0000	0,9583	0,9167	0,8750	0,8333	0,7917	0,7500
P(En)	1	0,95833	0,87847	0,76866	0,64055	0,50710	0,38033
1-P(En)	0,0000	0,0417	0,1215	0,2313	0,3594	0,4929	0,6197

$p_2=1-\dfrac{24}{24}\times\dfrac{23}{24}$; $p_3=1-\dfrac{24}{24}\times\dfrac{23}{24}\times\dfrac{22}{24}$; $p_4=1-\dfrac{24}{24}\times\dfrac{23}{24}\times\dfrac{22}{24}\times\dfrac{21}{24}$; etc.

On retrouve sur ce tableau la valeur 0,36 donnée pour p_5. On arrête le tableau à 7 car alors $p_7\approx0{,}6197>0{,}5$.

V] La cloche de l'abbé

L'abbé Barnabé a mis au point ce petit rituel matinal : il tire deux dés (faces numérotées de 1 à 6) en rejouant s'il obtient un double six (il n'aime pas le double six car cela lui fait penser à la Bête...), puis s'intéresse à la réalisation de l'événement E : « obtenir une somme égale à 2 ou 5 » (ses nombres préférés). Si E est réalisé, il sonne la lourde cloche à l'aurore, sinon c'est un moine de l'abbaye qui la sonne à sa place.

a) Quelle est la probabilité de E ?

Avec cette règle, l'abbé se retrouve avec 35 possibilités équiprobables parmi lesquels seulement 5 le conduisent à sonner la cloche. $P(E)=\dfrac{5}{35}=\dfrac{1}{7}$. Il a une chance sur 7 de devoir aller sonner la cloche.

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	x

b) Quelle est la probabilité que, pendant une semaine entière, il sonne la cloche au moins une fois ?
Pendant une semaine entière, il faut sonner la cloche 7 fois, et chaque jour il ne la sonne pas avec la probabilité $P(\overline{E})=\frac{6}{7}$. Donc la probabilité qu'il ne sonne pas la cloche pendant 7 jours est $\left(\frac{6}{7}\right)^7=\frac{279936}{823543}\approx 0{,}34$.

Finalement, la probabilité qu'il la fasse sonner au moins 1 fois pendant 7 jours est $1-\left(\frac{6}{7}\right)^7=\frac{543607}{823543}\approx 0{,}66$.

c) Quelle est la probabilité que, pendant une semaine entière, il sonne la cloche plus d'une fois ?
Cet événement est le contraire de la réunion de A : « ne pas sonner la cloche de la semaine » et B : « sonner la cloche juste une fois pendant la semaine ».
Pour la probabilité de B, on peut penser à l'arbre de probabilités qui donne les tirages de chaque jour de la semaine. Dans cet arbre il y a 7 chemins qui réalisent B, et ils ont tous la probabilité $\frac{1}{7}\times\left(\frac{6}{7}\right)^6=\frac{6^6}{7^7}=\frac{46656}{823543}\approx 0{,}056653$ et donc $P(B)=7\times\frac{6^6}{7^7}=\left(\frac{6}{7}\right)^6=\frac{46656}{117649}\approx 0{,}396569$.

La probabilité de A ou B est $P(A\cup B)=P(A)+P(B)=\left(\frac{6}{7}\right)^7+\left(\frac{6}{7}\right)^6=\frac{606528}{823543}\approx 0{,}736486$.

Finalement, la probabilité cherchée est celle du contraire de A ou B, c'est-à-dire :

$$P(\overline{A\cup B})=1-P(A\cup B)\approx 1-0{,}736486=0{,}263514$$

bonus1 (1,5pt)
Comme il se fait vieux, l'abbé voudrait, en tirant les deux dés à sa façon (en retirant lorsqu'une certaine combinaison se produit), obtenir un événement E' qui se réalise seulement une fois sur huit. Pouvez-vous l'aider ?
Pour obtenir ce genre de probabilité, il faudrait que $P(E')=\frac{4}{32}=\frac{1}{8}$. Il faut donc supprimer 4 possibilités (pour obtenir seulement 32 cas possibles) et en retenir 4 (pour constituer l'événement E' ayant la probabilité voulue) : par exemple, il peut retirer les dés s'il obtient une somme égale à 5 et aller tirer la cloche si la sonne fait 9 (cela ne va pas dans le sens de ses goûts mais on ne peut tenir compte de choses aussi irrationnelles).

	1	2	3	4	5	6
1	2	3	4	x	6	7
2	3	4	x	6	7	8
3	4	x	6	7	8	9
4	x	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

On aurait aussi pu proposer de retirer les dés chaque fois qu'il apparaît au moins un 6 (11 possibilités) ou que la somme fait 6 (5 autres possibilités) ou bien 3 (la moitié de 6) ou 9 (6 renversé, le pire!). Il ne reste plus que 16 possibilités et il suffit d'en choisir 2 pour obtenir un événement E'' qui se réalise une fois sur huit. Pour aller dans le sens des goûts irrationnels de l'abbé pour les nombres 2 et 5 (que ne faut-il pas faire parfois), on peut choisir d'aller tirer la cloche quand on obtient un double 2 ou un double 5. On a alors bien $P(E'')=\frac{2}{16}=\frac{1}{8}$.

	1	2	3	4	5	6
1	2	x	4	5	x	x
2	x	4	5	x	7	x
3	4	5	x	7	8	x
4	5	x	7	8	x	x
5	x	7	8	x	10	x
6	x	x	x	x	x	x

VI] L'urne de Pólya
On considère une urne contenant au départ a boules blanches et b boules rouges. On tire une boule de l'urne et on la remet avec une autre boule de la même couleur ; on a donc maintenant $a+b+1$ boules. On recommence un certain nombre de fois l'opération qui consiste à tirer une boule de l'urne et à la remettre avec une autre boule de la même couleur. On cherche à déterminer l'évolution de la proportion de boules blanches dans l'urne. On note $B(n)$ le nombre de boules blanches dans l'urne au bout de n tirages.
a) Écrire un algorithme ou un programme qui simule M fois cette expérience aléatoire avec N tirages, A boules blanches et B boules rouges. Cet algorithme affiche la valeur de $B(N)$ pour chaque expérience.

L'algorithme peut ressembler à ça :

saisir M, N, A et B
Pour *i* allant de 1 à M
a=A et *b*=B *(indispensable de remettre A et B aux valeurs d'origine pour chaque expérience)*
Pour *j* allant de 1 à N
R=nombre entier aléatoire entre 1 et *a*+*b*
si R<=*a* alors *a*=*a*+1
sinon *b*=*b*+1
fin du 2ème Pour
print *a*
fin du 1er Pour

b) Programmer ce programme sur la calculatrice et l'exécuter avec N=100, M=10, A=2 et B=2.
Noter les résultats et commenter : les résultats vous semblent-ils converger vers une valeur particulière ?
Voici la programmation en Python de cet algorithme, pour les valeurs particulières de M, N, A et B demandées (on n'a pas besoin de différencier A de *a* ou B de *b* puisqu'on ne prévoit pas de saisir ces valeurs). L'affichage est un peu sophistiqué, ce qui n'était pas nécessaire : on pouvait se contenter d'afficher A par l'instruction print(A).

```
from random import randint
M,N=10,100
for I in range(M):
    A,B=2,2
    for J in range(N):
        R=randint(1,A+B)
        if R<=A:
            A=A+1
        else:
            B=B+1
    print("tirage n° {} : {} boules blanches ({}% des boules)".format(I+1,A,round(A/(A+B)*100),2))
```

```
tirage n° 1 : 49 boules blanches (47% des boules)
tirage n° 2 : 66 boules blanches (63% des boules)
tirage n° 3 : 73 boules blanches (70% des boules)
tirage n° 4 : 48 boules blanches (46% des boules)
tirage n° 5 : 88 boules blanches (85% des boules)
tirage n° 6 : 18 boules blanches (17% des boules)
tirage n° 7 : 49 boules blanches (47% des boules)
tirage n° 8 : 34 boules blanches (33% des boules)
tirage n° 9 : 70 boules blanches (67% des boules)
tirage n° 10 : 43 boules blanches (41% des boules)
```

Le programme qui suit montre une version minimaliste de cet algorithme, avec utilisation de la structure des fonctions Pythons (*def fonction([arguments])* :). Cette structure, si elle amène les arguments A et B, nécessite des variables *a* et *b* distinctes (qui peuvent être nommées autrement). Attention alors de bien travailler avec ces nouvelles variables (et pas avec A et B).

```
from random import randint
def polya(M,N,A,B):
    for I in range(M):
        a,b=A,B
        for J in range(N):
            R=randint(1,a+b)
            if R<=a:a+=1
            else:b+=1
        print(a)
polya(10,100,2,2)
```

```
44
43
65
19
29
34
42
69
11
22
```

Commentaire sur les résultats. On n'en a que dix mais on voit qu'ils se répartissent assez largement : entre 18 et 88 la 1ère fois et entre 11 et 69 la 2de. Disons que le maximum pour A est 102 (les 2 de départ et les 100 qui auraient été ajoutés à A) alors que le minimum est 2 (les 2 de départ et ensuite les 100 auraient été ajoutés à B). Il n'y a pas vraiment de tendance à la convergence vers une valeur particulière. On peut s'arrêter là pour le commentaire, ou essayer des valeurs de M différentes : j'ai essayé à droite avec M=100 pour obtenir cette série de valeurs qui se répartissent entre 8 et 98, donc plus largement.

```
34 23 83 82 63 47 77 87 63 43
61 49 47 32 78 92 59 74 89 90
41 92 63 46 74 84 27 76 42 11
55 57 43 59 28 51 53 85 36 34
39 59 51 20 56 32 50 8  50 85
60 62 45 70 54 58 25 29 13 45
48 80 28 45 39 62 44 47 54 90
77 30 51 35 30 25 73 50 26 57
49 36 65 57 26 34 69 95 48 23
58 84 24 89 53 48 69 98 42 66
```

J'ai aussi essayé avec un rapport initial différent pour A et B : avec A=1 et B=2 au départ on obtient un nombre de A beaucoup plus faible en général, entre 1 (pas de A ajouté du tout!) et 85 (tout de même, A ayant été ajouté dès le départ sans doute, la proportion de A a pu grandir).

```
18 44 38 18 65 68 75 10 7  16
50 51 37 72 1  70 20 42 92 7
39 9  39 55 1  35 52 16 65 70
52 15 51 30 40 66 26 32 40 85
62 42 12 24 50 57 78 64 40 20
7  38 16 30 59 62 41 8  3  83
1  24 57 58 56 81 31 20 19 94
65 30 80 32 40 47 47 29 52 13
33 27 67 37 1  7  63 14 18 52
17 47 22 35 37 83 49 4  4  71
```

avec A=1, B=2

On commence à comprendre l'intérêt de cet algorithme qui a été développé par Pólya[1] pour modéliser le phénomène épidémique, puis qui s'est appliqué à toute sorte de contagion (tendance de l'opinion sur les marchés financiers, la compétition technologique, la localisation industrielle, etc.).

---

1 **George Pólya** est né à Budapest (Hongrie) le 13 décembre 1887 et mort à Palo Alto (États-Unis) le 7 septembre 1985. Il est l'auteur d'ouvrages théoriques sur les probabilités ou la théorie des nombres mais aussi d'ouvrages montrant l'intérêt d'une pédagogie de la découverte par construction comme son *Comment poser et résoudre un problème*, 2e éd., 1965, nouveau tirage 2007, traduction de *How to Solve It*, 1957.

bonus2 (1,5pt tant que bonus1+bonus2 $\leq 1,5$*)*

Le jeu de Jeanne consiste à jeter des dés autant de fois que nécessaire pour obtenir N fois la face 6, de la façon suivante : au départ on jette N dés, on reprend ceux qui n'ont pas fait 6 et on les rejette, et ainsi de suite jusqu'à obtenir les N faces 6. On note K la variable donnant le nombre de jets nécessaire pour obtenir les N faces 6. Écrire un algorithme ou un programme qui simule ce jeu et affiche K. Donner quelques résultats d'exécution pour $N=5$.

Ce jeu se traduit par un algorithme qui ressemble à ce qui suit :

saisir N
K=0 et F=0 *(K est le nombre de jets et F le nombre de Faces qui montrent le chiffre 6)*
Tant que F<N
D=N-F *(D est le nombre de dés à retirer)*
K=K+1 *(un jet est compté lorsqu'on lance les D dés autant qu'il y en a)*
Pour *i* allant de 1 à D
R=nombre entier aléatoire entre 1 et 6
si R=6 alors F=F+1
fin du Tant que
print K

Le programme Python peut ressembler à cela :

```
from random import randint
def jeanne(N):
    K,F=0,0
    while F<N:
        D=N-F
        K=K+1
        for i in range(D):
            R=randint(1,6)
            if R==6:F=F+1
    print(K,end=" - ")

for i in range(100):jeanne(5)
```

```
from random import randint
def jeanne(N):
    K,F=0,0
    while F<N:
        D=N-F
        K+=1
        for i in range(D):
            R=randint(1,6)
            if R==6:F+=1
    print(K)

jeanne(5)
```

```
6
```

```
17 - 8 - 14 - 10 - 14 - 10 - 13 - 26 - 12 - 15 - 11 - 7 - 8 - 5 - 13 - 8 - 15 - 19 - 18 - 22
- 7 - 19 - 14 - 11 - 9 - 10 - 6 - 13 - 8 - 6 - 30 - 14 - 17 - 10 - 19 - 8 - 10 - 6 - 17 - 34
- 12 - 17 - 10 - 11 - 7 - 26 - 5 - 15 - 10 - 9 - 13 - 14 - 12 - 7 - 13 - 22 - 20 - 15 - 9 -
6 - 7 - 11 - 14 - 14 - 9 - 16 - 4 - 6 - 12 - 8 - 26 - 12 - 22 - 8 - 15 - 12 - 10 - 6 - 6 - 2
1 - 3 - 14 - 27 - 15 - 21 - 24 - 25 - 9 - 18 - 11 - 11 - 11 - 21 - 23 - 16 - 27 - 5 - 10 - 5
- 18 -
```

Vous remarquerez que j'ai choisi une structure de fonction et que mon instruction *for i in range(100):jeanne(5)* lance 100 fois cette fonction. C'est pour obtenir un éventail des valeurs possibles. Vous remarquerez aussi que je les affiche les unes derrière les autres avec l'option end=« - » qui remplace l'option par défaut de print qui est un retour à la ligne (end=« \n »). On peut remplacer cela par la forme minimaliste de droite qui n'affiche qu'une seule valeur de K (comme c'était demandé).