



Je pense que le tableur est pédagogiquement plus adapté, car il permet une réactualisation en temps réel des lancers de dé ou des marches aléatoires...

Cette fiche n'est donc qu'un prétexte pour découvrir des fonctionnalités des modules `<matplotlib>` et `<random>`.

1. Lancer de dés

```
1  # -*- coding: utf-8 -*-
2  # Python3
3  #
4  # stabilisation des fréquences lors des lancers de dés
5
6  from random import randrange
7  from matplotlib import pyplot as plt
8
9  def nLancers(n):
10     L=[randrange(6)+1 for k in range(n)]
11     return(L)
12
13  def graph(n):
14     """ prépare le graphique correspondant à la fréquence
15     d'apparition des faces du dé pour <n> lancers
16     """
17     plt.title("freq sur "+str(n)+"\n lancers")
18     plt.hist(nLancers(n), bins=6, range=(.5,6.5), normed=True)
19     plt.axis([0, 7, 0, .3])
20     # pour tracer le segment (x1,y1)(x2,y2), les coordonnées sont
21     # données sous la forme [x1,x2], [y1,y2]
22     plt.plot([0,7],[.16667,.16667], 'r-')
23
24
25  plt.subplot(131)
26  graph(100)
27  plt.subplot(132)
28  graph(1000)
29  plt.subplot(133)
30  graph(10000)
31  plt.show()
```

2. Le vieux loup de mer

Extrait de *Élémentaire mon cher Watson*, Colin Bruce (p. 81)

« Rappelez-vous cet ivrogne que nous vîmes tituber dans Baker Street, la veille de Noël. Je viens d'apprendre qu'il s'appelait Davis. Il était, paraît-il marin à bord de l'*Illustrious*, un navire de Sa Majesté qui avait jeté l'ancre à Greenwich. Lorsque nous le vîmes, il tentait de retrouver le chemin de son bateau.

– Cela faisait un bon bout de chemin, commentais-je.



- Son intention n'était nullement de faire tout le trajet à pied. Il avait loué un petit canot et il avait ramé jusqu'à Fisherman's Wharf, un quai abandonné non loin d'ici, en se laissant porter par la marée montante. Visiblement, il pensait que la marée descendante lui permettrait de rentrer avant minuit, à la fin de sa permission. Mais en réalité, il n'est jamais rentré.
- Vu dans l'état dans lequel il était lorsque nous l'avons vu, il aurait pu lui arriver toutes sortes de choses.
- Et, de fait, il lui est arrivé quelque chose. La dernière partie de son chemin jusqu'au canot fut particulièrement mouvementée. Il faisait nuit noire sur Fisherman's Wharf. Le quai, très large, fait cent pas de long. Lorsqu'on arrive au bout, on tombe brutalement sur la rivière, sauf au centre, où une promenade en bois s'étend sur douze pas, de part et d'autre du milieu.
- Je ne vois pas comment il aurait pu retrouver son chemin, dis-je.

- C'est très facile pour quelqu'un qui n'a pas bu. Le quai est orienté plein nord. Au moment où il est arrivé, l'étoile Polaire devait être bien visible. Il lui suffisait de marcher vers elle, puis de sauter du bord du quai sur la passerelle, au centre de laquelle son canot était amarré.

Mais nous savons qu'il était saoul et qu'au lieu de marcher droit il faisait des zigzags. D'ailleurs d'après ce que nous avons vu et ce qu'ont constaté par la suite d'autres témoins, son trajet était une parfaite marche au hasard, au sens mathématique, c'est-à-dire qu'il ne suivait aucune règle dans ses changements de direction et n'avait aucune préférence quand au côté vers lequel il titubait. Il faisait un pas vers l'avant, suivit d'un pas de côté involontaire, d'une façon tellement aléatoire qu'il serait difficile pour un homme sobre d'en faire autant. Ainsi, bien qu'il se fût dirigé vers le nord, en direction du centre du quai, il se retrouva décalé de treize pas vers la droite lorsqu'il arriva à la rivière.

- Un nombre qui porte malheur, ajoutais-je.
- Cela lui a effectivement porté malheur. La nuit était froide, et une fine couche de glace s'était formée sur l'eau. En regardant vers le bas, il a aperçu une surface solide qu'il a dû prendre pour la passerelle en bois et il a sauté. On a retrouvé hier au fond de la rivière des clés et d'autres affaires lui appartenant. La marée a dû emporter son corps jusqu'à la mer. On ne le retrouvera sans doute jamais.
- Le pauvre homme. Cet accident semble toute fois assez banal. Pourquoi Lestrade a-t-il éprouvé le besoin de vous demander conseil ?
- C'est là que nous touchons à l'originalité de cette histoire. Davis avait pris l'avant veille une assurance-vie pour une somme de cent guinées. Il n'avait pas de famille, et la seule bénéficiaire de ce contrat était sa sœur. Il semble particulièrement curieux qu'il ait pris une assurance aussi chère, alors que l'on sait qu'il avait des dettes de jeu. La compagnie d'assurances pense qu'il a pu se suicider et refuse de payer – rappelez-vous que, par cette froide nuit de veille de Noël, plusieurs malheureux se sont jetés du haut des ponts de la Tamise.
- Lestrade doit être désolé pour la pauvre sœur. Notre inspecteur a le cœur moins dur qu'il voudrait nous le faire croire. Mais l'argument principal de la compagnie d'assurance, c'est que Davis faisait tout bonnement semblant d'être saoul et que, après s'être assuré que de nombreux témoins l'avaient vu incapable de marcher en ligne droite, il s'est jeté à l'eau volontairement. D'ailleurs le fait qu'il soit passé devant la résidence du détective le plus connu de Londres n'était peut-être pas un hasard. »

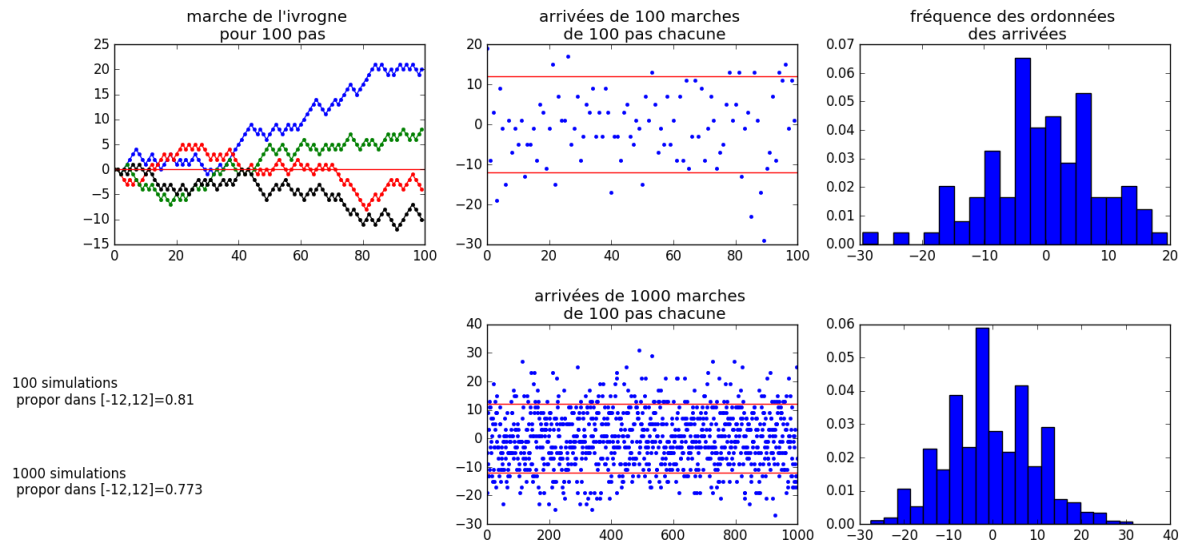
[...] Je frissonnais à l'idée de que l'ivrogne qui nous avait intrigués cette nuit là était peut-être en réalité en train de marcher froidement vers la mort, tout en sachant que nous l'observions. [...]

-
- « Très bien. J'aurais alors le plaisir de vous laissez expliquer cette affaire à Lestrade.
 - Pourquoi moi, alors que vous le feriez tellement mieux ?
 - Je dois faire une course, et Lestrade va revenir d'ici une heure. J'espère être de retour peu après, mais je suis heureux de vous confier le soin de lui expliquer. »

[...] Néanmoins, lorsque Lestrade arriva, il sembla suivre mes explications sans trop de mal.



En vous inspirant du fichier de la section précédente, essayez de programmer un algorithme qui donne une fenêtre ressemblant à celle-ci.



Voici quelques idées que j'ai utilisées :

- la fonction `randrange` du module `random`
- création de listes par compréhension
- définition de fonction pour préparer les graphiques
- la fonction `subplot` du sous module `pyplot` du module `matplotlib`



```
1  # -*- coding: utf-8 -*-
2  #Python 3
3  #
4  # marche de l'ivrogne
5
6  from random import randrange
7  from matplotlib import pyplot as plt
8
9  n = 100
10 X = [x for x in range(n)]
11
12 dy=[]
13 for i in range(n):
14     dy.append(2*randrange(2)-1) # liste de (-1) ou 1 : déplacement en y
15 Y = [sum(dy[:x]) for x in range(n)]
16
17 plt.plot(X,Y, '.b', linestyle='solid')
18 plt.plot([0,100],[0,0], 'r-')
19 plt.title("marche de l'ivrogne")
20 plt.show()
```

```
1  # -*- coding: utf-8 -*-
2  # python3
3  # marche de l'ivrogne
4
5  from random import randrange
6  from matplotlib import pyplot as plt
7
8  def decalage(n):
9      """ retourne la liste des ordonnées de l'ivrogne """
10     # liste de (-1) ou 1 : déplacement en y
11     dy=[0]+[2*randrange(2)-1 for k in range(n-1)]
12     y = [sum(dy[:x]) for x in range(n)]
13     return(y)
14
15 def graph(n):
16     """ prépare le graphique de 4 marches de <n> pas """
17     Y1, Y2, Y3, Y4 = decalage(n), decalage(n), decalage(n), decalage(n)
18     # par défaut : les abscisses sont les entiers de 0 à n-1
19     plt.plot(Y1, '.b', linestyle='solid')
20     plt.plot(Y2, '.r', linestyle='solid')
21     plt.plot(Y3, '.g', linestyle='solid')
22     plt.plot(Y4, '.k', linestyle='solid')
23     plt.plot([0,100],[0,0], 'r-')
24     plt.title("marche de l'ivrogne \n pour "+str(n)+" pas")
25
26 def arrivee(n,p):
27     """ donne l'ordonnée de l'arrivée pour <n> marches de <p> pas """
28     A = [] # liste des ordonnées de l'arrivée
29     for i in range(n):
```



```
30         y=sum([2*randrange(2)-1 for k in range(p-1)])
31         A.append(y)
32     return(A)
33
34 def graphApts(A,n,p):
35     """ prépare le graphique des arrivées de <n> marches de <p> pas """
36     plt.plot(A, '.b')
37     plt.plot([0,n],[12,12], 'r-') # droite passant par (x1,y1)(x2,y2)
38     plt.plot([0,n],[-12,-12], 'r-') # il faut écrire [x1,x2], [y1,y2]
39     plt.title("arrivées de "+str(n)+" marches \n de "+str(p)+" pas chacune")
40
41     N=100
42     plt.subplots_adjust(wspace = .2, hspace = .4)
43     plt.subplot(231)
44     graph(N)
45
46     plt.subplot(232)
47     A = arrivee(100,N)
48     graphApts(A,100,N)
49
50     plt.subplot(233)
51     plt.hist(A, range=(min(A)-.5, max(A)+.5), bins=20, normed=True)
52     plt.title("fréquence des ordonnées \n des arrivées")
53     # compte le nombre d'arrivées dans [-12,12]
54     p = len([x for x in A if -12<=x<=12])
55     plt.figtext(0.05,.3,"100 simulations \n propor dans [-12,12]="+
56                 str(p/100)[:5])
57     #plt.subplot(234)
58
59     plt.subplot(235)
60     A = arrivee(1000,N)
61     graphApts(A,1000,N)
62     plt.subplot(236)
63     plt.hist(A, range=(min(A)-.5,max(A)+.5), bins=20, normed=True)
64     p = len([x for x in A if -12<= x <=12])
65     plt.figtext(0.05,.15,"1000 simulations \n propor dans [-12,12]="+
66                 str(p/1000)[:5])
67     plt.show()
```
