



T.D. de Probabilités

Exercice 4 :

```
1 import random as rdm
2
3 def ponctualite(n):
4     p = 1 # à l'heure le jour J0
5     for j in range(1, n+1):
6         if p == 1:
7             p = int(rdm.random()<1/4)
8         else: # p == 0 (retard)
9             p = int(rdm.random()<9/10)
10    return p
11
12 def ponctualite2(n):
13     p = 1 # il est à l'heure le jour J0
14     for j in range(1, n+1):
15         p_suivant = 0 # par défaut
16         if (p == 1 and rdm.random()<1/4) or (p == 0 and rdm.random()<9/10):
17             p_suivant = 1
18         p = p_suivant
19    return p
20
21 def ponctualite3(n):
22     p = 1 # il est à l'heure le jour J0
23     for j in range(1, n+1):
24         if (p == 1 and rdm.random()<1/4) or (p == 0 and rdm.random()<9/10):
25             p = 1
26         else:
27             p = 0
28    return p
29
30 n, m = 365, 10000
31 L = [ponctualite2(n) for k in range(m)]
32 print('estimation de la probabilité de la ponctualité le jour n :', sum(L)/m)
```

Exercice 5 :

```
1 import random as rdm
2
3 def simulJeu():
4     d1, d2 = rdm.randint(1, 6), rdm.randint(1, 6)
5     nlancer = 1
6     while d1+d2 != 5 and d1+d2 != 7:
7         d1, d2 = rdm.randint(1, 6), rdm.randint(1, 6)
8         nlancer += 1
9     return d1+d2
10
11 def estimProbasPremierCinq(n = 10000):
12     L = [simulJeu() for k in range(n)]
13     return L.count(5)/n
14
15 def estimProbasPremierSept(n = 10000):
16     L = [simulJeu() for k in range(n)]
17     return L.count(7)/n
```

Exercice 6 :

```
1 import random as rdm
2
3 def simulPassage():
4     n = 1
5     while rdm.random() < 1/(n+1):
6         n += 1
7     return n
8
9 def hauteurMoyenne(m = 10000):
10    L = [simulPassage() for k in range(m)]
11    return sum(L)/m
```

Exercice 7 :

```
1 import random as rdm
2
3 def seanceGratuite():
4     seances = 0
5     LAdistinctes = []
6     while len(LAdistinctes) != 4:
7         a = rdm.randint(1,4)
8         seances += 1
9         if a not in LAdistinctes:
10            LAdistinctes.append(a)
11    return seances
12
13 def seanceGratuite2():
14    L = [0]*4 # avec L[i] = 1 si on a l'affiche n° i
15    nbPassages = 0
16    while sum(L) != 4:
17        a = rdm.randint(0,3)
18        if L[a] == 0:
19            L[a] = 1
20            nbPassages += 1
21    return nbPassages
22
23 def seanceGratuite3():
24    L = [0]*4 # nbe de fois où on a l'affiche n° i
25    seances = 0
26    while 0 in L:
27        L[rdm.randint(0,3)] += 1
28        seances += 1
29    return seances
30
31 def nbMoyenSeances(m=10000):
32    L = [seanceGratuite() for k in range(m)]
33    return sum(L)/m
```

Exercice 8 :

```
1 import random as rdm
2 import numpy as np
3
4 def tirageF(y):
5     num_ticket = rdm.randint(1,10)
6     return int(num_ticket <= y)
7
8
```

```

9 def tirageE(x):
10     UE = list(range(1, 11))
11     tick1, tick2 = rdm.sample(UE,2)
12     return int(tick1<=x and tick2 <=x)
13
14 def tirageE2(x):
15     """On suppose les x aller billets gagnant"""
16     UE = list(range(1,11))
17     tick1 = rdm.choice(U) # t1 = rdm.randint(1,10)
18     if tick1 <= x:
19         U.remove(tick1)
20         tick2 = rdm.choice(U)
21         if tick2 <= x:
22             return 1
23     return 0
24
25 def tirageE3(x):
26     UE = [1]*x+[0]*(10-x)
27     t1, t2 = rdm.sample(UE, 2)
28     return t1*t2
29
30 def jeu(x,y):
31     tE, tF = tirageE(x), tirageF(y)
32     while tE == 0 and tF ==0: # les 2 ont perdu
33         tE, tF = tirageE(x), tirageF(y)
34     if tE == 1:
35         return 1
36     else:
37         return 2
38
39 def valPourfEntier():
40     f = lambda x:(10*x*(x-1))/(90+x-x**2)
41     L = []
42     for x in range(2,10):
43         if np.floor(f(x))==f(x):
44             L.append(x)
45     return L

```

Exercice 9 :

```

1 import random as rdm
2
3 def rgPremierSucces(p):
4     x = 1
5     while rdm.random() > p:
6         x += 1
7     return x
8
9 def simulY():
10    X = rgPremierSucces(1/2)
11    Y = 1
12    for k in range(X):
13        if rdm.random() <= 1/3: # tirage d'une blanche dans U
14            return Y
15        Y += 1
16    # Attention : si aucune blanche n'a été tiré dans U,
17    # alors à la sortie de la boucle on a : Y = X+1
18    return (Y-1)+rgPremierSucces(1/4)
19
20 def estimEsperanceY(m = 10000):
21    return sum([simulY() for _ in range(m)])/m

```