Des probabilités avec Python

Samedi 21 janvier 2017

Buts du TP

Quelle est la moyenne des résultats obtenus quand on lance les dés de nombreuses fois ?

— le bon sens populaire dit : « pfiou, ça va varier » ;
— un bon élève de terminale doit pouvoir dire : « en moyenne, ça va tendre vers 7/2 » (notion des plus imprécises, mais c'est déjà ça) ;
— le taupin doit pouvoir dire que l'espérance après n tirages vaut toujours 7/2 ;
— un peu plus tard, ce même élève pourra dire (loi des grands nombres) que la moyenne *tend* vers 7/2 (en des sens subtils...) ;
— un peu plus tard encore, cet élève saura même décrire (grâce au théorème central limite) ce à quoi il faut s'attendre pour cette moyenne après n tirages : la loi la décrivant est proche d'une loi très simple (théorème central limite).

On souhaite :

— faire des simulations de n lancés et évaluer les moyennes obtenues ;
— observer cette moyenne lorsque n tend vers $+\infty$ (la loi des grands nombres nous dit ce qu'elle fait « probablement ») ;
— observer la distribution des valeurs sur différentes séries de 100 lancés (le théorème central limite décrit assez bien celle-ci) ;
— s'il reste du temps : vérifier expérimentalement les résultat de « l'exercice de l'ascenseur »

Exercice 1. *Créer (au bon endroit) un dossier associé à ce TP. Lancer Spyder/Pyzo/Idle, sauvegarder au bon endroit le fichier* `tdprobasouuntruccommeça.py` *; écrire une commande absurde, de type* `print( 6*7 )` *dans l'éditeur, sauvegarder et exécuter.* Si vous êtes sous Spyder, changez (via F6) les options d'exécution, pour avoir à chaque exécution une nouvelle console et garder la main dessus. (il y a donc deux cases à vérifier/cocher).

Exercice 2. *Vérifier que le premier exo a effectivement bien été fait : tout manquement donnera lieu à une agitation néfaste pour tout le monde...*

Pendant tout le TP, on pensera à noter les résultats en commentaire dans le fichier de script...

1 Premières simulations avec Python

La bibliothèque `random` fournit la fonction `randint`... qui devrait bien vous servir.

1. Écrire une fonction `tirage_de` telle que `tirage_de()` retourne une valeur parmi $[\![1, 6]\!]$, chacune avec probabilité $\dfrac{1}{6}$.
2. Écrire une fonction prenant un entier $N > 0$ en argument, réalisant N lancers consécutifs, et renvoyant la moyenne des résultats obtenus.

```
>>> serie(1000)
3.518
>>> serie(1000)
3.44
```

3. Construire un tableau constitué des résultats de 10^5 séries consécutifs de 10^2 lancers [1].

1. Vous pouvez commencer avec 10^4 séries. Et pensez à la suite, le temps qu'*il* fasse les calculs ! Quand vous passerez à la partie suivante, commentez ces calculs longs dans votre script, pour qu'ils ne soient pas réexécutés à chaque fois.

4. Parmi ces 10^5 séries :
   — Quelle est la moyenne... des moyennes ?
   — Quelle est la moyenne maximale qui a été atteinte ? Et la minimale ?
   — Combien de séries ont fourni une moyenne majorée par 3.4 ? Et minoré par 3.6 ?
   — Même chose en remplaçant 3.4 et 3.6 par 3 et 4.
   — Même chose en remplaçant « $\geqslant 4$ » par « $\geqslant 4.01$ » puis « $\geqslant 4.02$ ».

   Ces questions seront reprises à la fin de la partie 3.

2 Loi (faible) des grands nombres

Si on note Y_n la variable aléatoire égale à la moyenne après n lancers aléatoires, alors l'espérance de Y_n vaut $7/2$, et on a même d'après la loi (faible) des grands nombres :

$$\forall \varepsilon > 0, \qquad \mathbb{P}(|Y_n - 7/2| > \varepsilon) \underset{n \to +\infty}{\longrightarrow} 0.$$

1. Écrire une fonction prenant en entrée deux entiers $a, b > 0$, réalisant une série de $a \times b$ tirages, et retournant une liste constituée des moyennes obtenues (depuis le début) toutes les b séries. Cette fonction retourne donc une liste de $a+1$ flottants.

   OULALA, ça c'est une phrase qu'elle est compliquée à comprendre...

   Dans l'exemple suivant, on réalise deux séries de 1000 *tirages, en notant les résultats intermédiaires tous les* 200 *tirages.*

```
>>> ab_series(5, 200)
[0, 3.425, 3.4725, 3.5383333333333336, 3.51875, 3.5089999999999995]
>>> ab_series(5, 200)
[0, 3.4, 3.48, 3.4599999999999995, 3.47875, 3.501]
```

2. Exécuter `ab_serie(10, 10**k)` pour $k \in [\![3, 6]\!]$.
3. Représenter l'évolution de la moyenne lors de n séries (par exemple pour $n \in \{10^5, 10^6, 10^7\}$. *À n fixé, on pourra relier les points d'abscisses $100k$ ($k \in [\![0, n/k]\!]$) et d'ordonnées* `ab_series(n/100, 100)` *(en enlevant éventuellement les premiers points ; vous verrez pourquoi après un premier essai).*
4. Bonus : le théorème central limite (à venir) dit[2] que la moyenne est « proche de l'espérance », dans une zone de largeur caractéristique $2\dfrac{\sigma}{\sqrt{n}}$, avec ici $\sigma = \sqrt{\frac{35}{12}}\cdot$ Visualiser ceci en ajoutant sur les graphes précédents les courbes d'équations $y = \mu \pm \dfrac{\sigma}{\sqrt{n}}\cdot$

3 Théorème central limite

On va observer la moyenne Y (qui est une variable aléatoire) après $n = 100$ tirages. On ne touchera plus à cette valeur de n. Cette moyenne a une *espérance* de $\dfrac{7}{2}\cdot$ Par ailleurs, puisque $100 \simeq +\infty$, le théorème central limite dit que cette loi peut être approchée par une « loi normale centrée » $\mathcal{N}(7/2, \sigma^2/n)$: on va représenter la fonction de répartition de cette loi.
On va lancer de nombreuses séries de 100 tirages (comme à la fin de la première partie) et observer la distribution, en comparant visuellement l'histogramme obtenu au graphe précédent. On va enfin comparer les résultat de la fin de la première partie à ce que « prédit » la théorie.

1. La fonction de répartition de la loi $\mathcal{N}(7/2, \sigma^2/n)$ est $x \mapsto \dfrac{\sqrt{n}}{\sqrt{2\pi}} \mathrm{e}^{-n(x-7/2)/\sigma^2}$. Représenter le graphe de cette fonction.
2. Pour visualiser la distribution des valeurs expérimentales, on peut relier les points de coordonnées (v, n_v), avec v décrivant les valeurs prises par Y (donc de la forme $\dfrac{k}{100}$), et n_v le nombre de fois que v a été trouvé lors de N séries, avec par exemple $N = 10^5$:

2. Entre autres choses, et de façon différente !

```
n = 100
nb_series = 10**5

plein_de_series = [serie(n) for _ in range(nb_series)]

x = [i/float(n) for i in range(250, 451)]
y = [plein_de_series.count(i) for i in x]

pypl.plot(x, y)
```

Merci de NE PAS taper ce qui précède sans avoir VRAIMENT réfléchi au sens de ces commandes.
On aura noté la très efficace méthode de liste `count`, qui fait ce qu'on pense.

3. Pour comparer les deux courbes, on peut les représenter sur un même graphe, en imposant le maximum égal à 1. Pour la première, on enlève la constante multiplicative devant l'exponentielle. Pour la seconde, on divise toutes les ordonnées... par le maximum !
   Faites le !
4. Sachant que Y suit une loi $\mathcal{N}(7/2, \sigma^2/n)$, et en notant $X = \dfrac{\sqrt{n}}{\sigma}(Y - 7/2)$, la loi de X est $\mathcal{N}(0,1)$, et en particulier :
   $$\mathbb{P}(Y \leqslant v) = \mathbb{P}(X \leqslant v') = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{v'} \mathrm{e}^{-t^2/2} dt,$$
   avec $v' = \dfrac{\sqrt{n}}{\sigma}(v - 7/2)$.
   Vérifier ceci pour $v = 3, 4$ en évaluant cette intégrale avec `quad` (de la bibliothèque `scipy.integrate`, et en comparant aux résultats expérimentaux trouvés à la fin de la première partie.

4 À vous de jouer

Les deux théorèmes qu'on vient de voir s'appliquent dès qu'on connaît l'espérance et l'écart-type d'une variable aléatoire

1. Ouvrir un livre de probabilité ; choisir un exercice où il est question d'une variable aléatoire « simulable facilement ».
2. Reprendre les trois premières parties de ce TP !
3. Bonus : vous pouvez aussi estimer l'écart-type : voir le DM numéro 3...
4. Autre bonus[3] : déterminer la vraie distribution de probabilité de la moyenne après 100 tirages.
   Il s'agit donc de sommer des lois uniformes discrètes. Ça sent la programmation dynamique ; je conseille de considérer plutôt la distribution de la somme des tirages, histoire de rester parmi des entiers...

5 Le problème de l'ascenseur (centrale 2015)

n personnes entrent dans un ascenseur et choisissent de façon aléatoire un étage parmi p possibles. On note X le nombre d'arrêts. Ω désigne l'ensemble des applications de $[\![1,n]\!]$ dans $[\![1,p]\!]$.

1. Écrire une fonction simulant X.
2. Estimer[4] $\mathbb{E}(X)$ lorsque $n = 10$ et $p \in \{15, 30, 100\}$.
3. Qualitativement, que dire de cette espérance quand p tend vers $+\infty$, à n fixé ?
4. Pour $j \in [\![1,p]\!]$ et $f \in \Omega$, on définit $X_j(f) = \begin{cases} 1 & \text{si } j \in f([\![1,n]\!]) \\ 0 & \text{sinon} \end{cases}$
   (a) Justifier : $X = X_1 + \cdots + X_p$.

3. difficile mais intéressant d'un point de vue mathématique comme d'un point de vue informatique !
4. À l'aide de la fonction écrite à la question précédente !

(b) En déduire $\mathbb{E}(X)$.

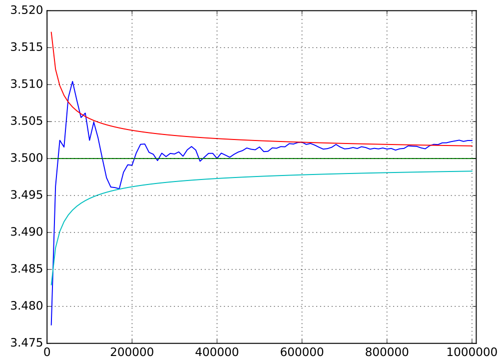
(c) Comparer cette espérance aux résultats expérimentaux.

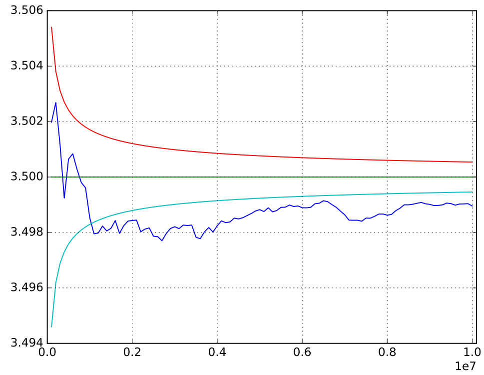
(d) Vérifier le résultat intuité à la question 3.

5. Bonus : évaluer $\mathrm{Var}(X)$.

6 De bien beaux résultats

— Loi des grands nombres, pour 10^6 et 10^7 tirages.





— Théorème central limite : 10^4 puis 10^6 séries de 100 tirages.