

Probabilités, Statistiques, Combinatoire : TM4

Expérimentations : théorème de la limite centrale

On suppose dans cette feuille que les variables aléatoires sont à valeurs dans $\mathbb{R}$.

Dans vos programmes, les lois des variables aléatoires seront représentées par des fonctions de simulation, comme celles que vous avez écrites lors des semaines précédentes, ou (pour les variables aléatoires ne prenant qu'un nombre fini de valeurs) par un dictionnaire dont les clés sont les valeurs possibles, et les valeurs associées, les probabilités des clés.

Lorsque vous aurez besoin d'un *échantillon* de n tirages d'une loi donnée, cela correspondra à un dictionnaire de fréquences observées sur n tirages de votre loi, obtenu par la fonction `tiragesRepetes` écrite lors d'une semaine précédente.

4.1 Normalisation d'une loi

Supposons que X soit une variable aléatoire dont l'espérance et la variance sont connues : m et v , respectivement. Pour deux constantes a et b données, quelles sont l'espérance et la variance de la variable aléatoire $Y = a.X + b$?

En supposant que $v > 0$, déterminer (en fonction de m et v) un choix de $a > 0$ et $b \in \mathbb{R}$ qui assure que Y ait pour espérance 0, et pour variance 1. La loi de la variable aléatoire Y ainsi définie sera appelée la *normalisation* de la loi de X .

Écrire une fonction `moyenne(dict)` qui, prenant en entrée un dictionnaire qui décrit une loi de probabilités, retourne l'espérance de la loi ; et une fonction `variance(dict)`, qui retourne la variance de la loi décrite par le dictionnaire passé en entrée.

Note : lorsque vos fonctions `moyenne` et `variance` sont utilisées sur un dictionnaire représentant un échantillon, on parle plutôt de *moyenne empirique* et de *variance empirique*.

Écrire une fonction de normalisation `normalisation(dict)` qui retourne (sous forme de dictionnaire) la normalisation de la loi décrite par le dictionnaire passé en entrée.

Optionnel : Écrire également une fonction qui prend en entrée une fonction de simulation d'une loi (sur le même modèle que les semaines précédentes), ainsi que l'espérance et la variance de la loi, et retourne une fonction de simulation de la loi normalisée. Mieux : écrivez une fonction qui ne prend en entrée que la fonction de simulation de la loi, et utilise l'espérance et la variance empiriques d'un échantillon pour faire la normalisation de manière approchée (attention, il ne faut pas que l'espérance et la variance empiriques soient recalculées à chaque appel de la fonction de simulation retournée).

4.2 Tracé de fonctions de répartition empiriques

Dans le fichier `dessin.py` fourni, la fonction `dessin_va(L)` s'appelle avec comme paramètre une liste de dictionnaire qui peuvent représenter soit des lois, soit des échantillons de lois. Elle permet de tracer, au moyen de la bibliothèque `matplotlib`, soit des courbes de fréquences (empiriques ou non), soit (avec un second paramètre optionnel prenant la valeur

`True`) des courbes de fréquences cumulées (correspondant à peu près à des fonctions de distribution). (L'appel à cette fonction `dessin.va` est bloquant tant que la fenêtre affichant le graphique n'est pas fermée)

En combinant les fonctions que vous avez écrites au cours des dernières semaines, produisez un tracé représentant, sur le même dessin, et à partir d'échantillons de taille 10000 ou plus, les courbes de fréquences cumulées pour les variables aléatoires *normalisées* :

- variable binomiale de paramètres n et $p = 0.5$;
- variable binomiale de paramètres n et $p = 0.2$;
- somme de n tirages (indépendants) d'un dé à 6 faces ;
- somme de n tirages indépendants d'une loi géométrique.

(Vous avez vu en cours l'espérance et la variance de toutes ces lois ; pour mémoire, la variance du lancer d'un dé à 6 faces est $35/12$)

Faire varier le paramètre n , de $n = 2$ à $n = 100$ (pas forcément de 1 en 1). Vous pouvez ajouter d'autres lois, toujours en prenant la somme de n variables aléatoires de même loi.

Qu'observez-vous ?