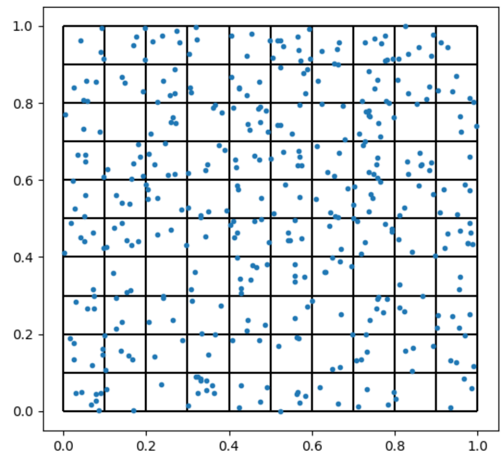
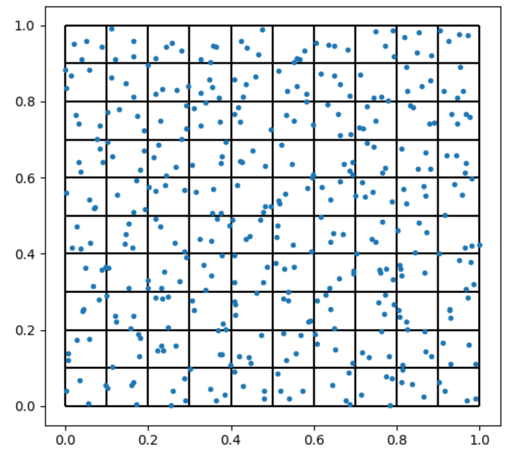
Probabilités, Statistiques, Combinatoire : TM5

Simulations géométriques, espérances





Les images ci-dessus montrent deux ensembles de 400 points pris aléatoirement dans le carré unité (l'ensemble des points du plan dont les deux coordonnées sont comprises entre 0 et 1). L'un des deux ensembles a été généré "honnêtement", en prenant 400 points indépendamment les uns des autres, chacun uniformément dans le carré ; l'autre a été généré de manière "malhonnête", selon un procédé secret.

Dans le fichier `TM5.py` fourni, vous trouverez une fonction `dessinEnsemble(S)`, qui produit un dessin similaire si on lui donne comme argument une liste de points, chaque point étant un tuple (paire) de coordonnées. Vous pouvez également ajouter un second paramètre k, qui indique alors le nombre de subdivisions du carré unité à dessiner ($k = 10$ sur les dessins ci-dessus).

5.1 Espérances

Dans les séances précédentes, vous avez écrit des fonctions qui produisent des dictionnaires de fréquences pour les résultats de variables aléatoires.

Écrivez une fonction `moyenne(f,dict)` qui prend en entrée une fonction f et un dictionnaire de fréquences ; il est supposé que la fonction prend ses valeurs au moins dans l'ensemble des clés du dictionnaire. Votre fonction `moyenneEmpirique` retournera la moyenne (pondérée par les fréquences) des images par f des clés du dictionnaire.

Ainsi, si par exemple le dictionnaire est `{1:0.34, 2:0.36, 3:0.18, 4:0.22}`, et que f est la fonction carré ($f(x) = x^2$), la valeur retournée devrait être $1 \times 0.34 + 4 \times 0.36 + 9 \times 0.18 + 16 \times 0.22 = 6.92$; avec la fonction identité ($f(x) = x$), la valeur retournée devrait être $1 \times 0.34 + 2 \times 0.36 + 3 \times 0.18 + 4 \times 0.22 = 2.48$.

Utilisez les fonctions que vous avez écrites lors des séances précédentes, pour engendrer des dictionnaires de fréquences pour 1000, puis 10000 tirages de la loi géométrique de paramètre $1/6$, ainsi que pour la loi binomiale de paramètres $(10, 0.3)$. Utilisez ensuite votre fonction `moyenne` pour comparer les moyennes (et, à peine plus compliqué : les variances empiriques) de vos dictionnaires avec les espérances et variances vues en cours pour ces lois.

5.2 Points aléatoires dans le carré

Dans le module `random`, la fonction `random.random()` retourne un nombre réel uniforme, compris entre 0 et 1.

Écrivez une fonction `pointAleatoire()`, qui retourne (sous forme de tuple de coordonnées) un point aléatoire du carré. Écrivez également une fonction `ensembleAleatoire(modele,n)`, qui retourne une liste de n points aléatoires indépendants, tous tirés par la fonction de simulation `modele` (sur le principe des séances précédentes, à ceci près qu'on ne retourne pas des dictionnaires mais des listes de points ; ainsi `ensembleAleatoire(pointAleatoire,400)` devrait retourner un ensemble de 400 points uniformes dans le carré).

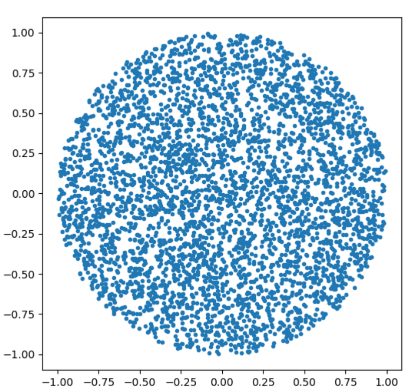
Utilisez la fonction `dessinEnsemble` pour visualiser vos ensembles de points ; esayez de préciser votre intuition sur "à quoi s'attendre" dans un ensemble de points aléatoires : est-il normal d'observeer des "zones" un peu plus denses, ou moins denses, que la moyenne ? dans quelles proportions ? Il peut être pertinent pour cela de découper le carré en une grille de telle sorte que le nombre moyen de points par "case" reste à peu près constant (par exemple, 100 cases pour 500 points, 400 cases pour 2000 points, etc).

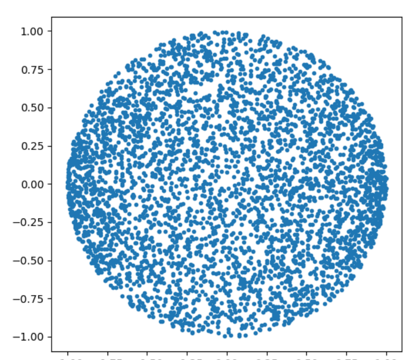
5.3 Points aléatoires dans le disque

Cherchez un moyen de tirer un point aléatoire uniforme dans le disque unité D (l'ensemble des points de coordonnées (x, y) à distance au plus 1 de l'origine, c'est-à-dire tels que l'on ait $x^2 + y^2 \leq 1$). Vérifiez, au moins visuellement (en affichant des ensembles de 1000 à 5000 points), que votre méthode n'accumule pas systématiquement les points quelque part. Sur les images ci-dessous, l'ensemble de gauche est tiré selon une bonne méthode ; celui de droite est tiré selon une "mauvaise" méthode, qui a tendance à concentrer les points à droite et à gauche (mais qui est une méthode qui peut sembler naturelle).

Si (x, y) et (x', y') sont deux points, le carré de leur distance est $(x - x')^2 + (y - y')^2$.

Lorsque deux points A et B sont choisis indépendamment, uniformément dans le disque unité, l'espérance du carré de leur distance est un nombre remarquable. Essayez de déterminer expérimentalement ce nombre (il n'est pas demandé de preuve !). Comparer à l'espérance du carré de la distance entre un point A aléatoire uniforme dans le disque, et le centre de ce disque.





5.4 Pour les curieux

Si l'on répartit N points **régulièrement** (pas aléatoirement) dans le carré unité, quel est l'ordre de grandeur de la distance minimale entre deux points de l'ensemble ? (on peut supposer que N est un carré parfait, $N = n^2$; ainsi les points peuvent être répartis sur une grille régulière ; mais on exprimera la réponse comme une fonction de N)

Écrivez deux fonctions pour mesurer des petites distances dans un ensemble de points :

— une fonction `distancePoint(p,E)`, qui prend en entrée un point (sous forme de couple de coordonnées) et un ensemble de points, et retourne la plus petite distance entre le point et un point de l'ensemble ;

— une fonction `distanceMin(E)`, qui prend en entrée un ensemble de points et retourne la plus petite distance entre deux points (difféernts) de l'ensemble. **Il existe, pour ce problème, des algorithmes de complexité $O(n \log n)$ si n désigne le nombre de points ; vous pouvez réfléchir au problème, mais utiliser un algorithme de complexité supérieure est acceptable**

Utilisez ces deux fonctions avc des ensembles aléatoires de points du carré unité, de tailles variables (pour le point dans `distancePoint`, vous prendrez un point fixe comme l'origine), et essayez de déterminer l'ordre de grandeur typique (comme fonction de la taille n de l'ensemble) de ces deux grandeurs. Comparez le résultat à ce que vous avez prévu pour des points répartis régulièrement.

Note : Si vous faites plusieurs simulations pour la même taille, vous constaterez des valeurs assez variables ; pour se faire une idée un peu plus précise du bon ordre de grandeur, il faut faire la moyenne sur un assez grand nombre de simulations pour chaque taille ; typiquement, une centaine.