

INF 13 - Variables aléatoires discrètes.

1. DEUX COMMANDES SUPPLÉMENTAIRES.

On doit connaître les deux commandes suivantes, de la bibliothèque `numpy.random` toujours importée sous l'alias `rd.random`.

- `rd.poisson(x)` simule une variable aléatoire qui suit une loi de Poisson de paramètre x . On peut, comme pour les autres simulations de probabilités, utiliser un argument supplémentaire `size =` pour simuler un échantillon.
- `rd.geometric(x)` fait la même chose avec une loi géométrique.

2. QUELQUES EXERCICES

Exercice 1 On part avec une urne avec une unique boule bleue. On lance dès jusqu'à obtenir un 6. A chaque lancer on rajoute une boule noire dans l'urne. A la fin, on tire une boule dans l'urne : on gagne si c'est la boule bleue, on perd sinon.

1. Écrire une fonction Python qui simule cette expérience. Elle renvoie `True` si on a gagné. On n'utilisera pas les nouvelles commandes.
2. Réécrire la fonction avec la commande `rd.geometric`.
3. Écrire un programme qui prend en entrée un entier n et renvoie l'approximation de la probabilité de victoire qu'on fait en tentant n expériences. Comparer avec la probabilité théorique.

Exercice 2 Écrire une fonction qui simule la somme de deux variables aléatoires de Poisson de paramètre λ et μ . Représenter l'histogramme obtenu pour un grand nombre de simulations. Comparer l'histogramme avec une VA qui suit une loi $\mathcal{P}(\lambda + \mu)$.

3. APPROXIMATION DE LA LOI DE POISSON / LOI BINOMIALE

Pour $\lambda > 0$ fixé (attention, le mot clé `lambda` est réservé en Python, il faudra en utiliser un autre, comme `lambda_`), tracer les histogrammes obtenus pour des échantillons assez grands de loi binomiales $B(n, \frac{\lambda}{n})$ pour $n = 10, 100, 1000, \dots$. Superposez les à l'histogramme d'échantillons de même taille de lois de Poisson de paramètres λ . Que remarque-t-on?