
TP 13 : Simulation de variables aléatoires finies.

Pour faire des probabilités avec Python on utilise la bibliothèque `numpy.random` que l'on importe à l'aide de la commande

```
import numpy.random as rd
```

1 La commande `rd.randint`

- ▷ Choisissez un entier n et testez la commande `rd.randint(n)` plusieurs fois dans la console pour différentes valeurs de n . Qu'observez-vous ?
- ▷ Testez la commande `rd.randint(n,m)` pour différentes valeurs de n et de m . Qu'observez-vous ?
- ▷ Choisissez un entier p et ajoutez un troisième argument dans la commande `rd.randint(n,m,p)`. Qu'observez-vous ?
- ▷ En résumé que font les commandes `rd.randint(n)`, `rd.randint(n,m)`, `rd.randint(n,m,p)` ?

Exercice 1. 1. Ecrire une fonction dé qui simule le lancer d'un dé cubique non truqué.

2. On lance 10 fois un dé équilibré à 6 faces. Ecrire une fonction appelée `comptagesix` permettant de déterminer le nombre de fois où le 6 apparaît au cours des 10 lancers. On pourra utiliser la fonction `dé`.

Exercice 2. On considère une urne contenant 4 boules bleues numérotées de 1 à 4, 5 boules vertes numérotées de 5 à 9 et 2 boules jaunes numérotées 10 et 11. Ecrire une fonction qui simule le tirage d'une boule dans cette urne et qui renvoie la couleur de la boule tirée.

2 La commande `rd.binomial`.

- ▷ Choisissez un entier n et un nombre p compris entre 0 et 1 et tester la commande `rd.binomial(n,p)` avec différentes valeurs de p plus ou moins grande. Qu'observez-vous ?
- ▷ Choisissez un entier m et ajoutez un troisième paramètre dans la commande `rd.binomial(n,p,m)`. Qu'observez-vous ?
- ▷ Que font les commandes `rd.binomial(n,p)` et `rd.binomial(n,p,m)` ?

Exercice 3. Madame A téléphone à n personnes distinctes et la probabilité que chacune d'entre-elles réponde est la même : p . On note X le nombre de correspondant joints.

1. Construire une fonction appels qui permet de simuler cette expérience et qui renvoie le nombre de personnes jointes.
2. Elle recommence en essayant de joindre les personnes absentes la première fois. On note Y le nombre de réussites de ce deuxième essai. Compléter le programme précédent pour qu'il renvoie X et Y .

3 La commande `rd.random`

- ▷ Tester la commande `rd.random()` plusieurs fois dans la console. Qu'observez-vous ?
- ▷ Tester l'instruction `rd.random(n)` pour différentes valeurs de $n \in \mathbb{N}$. Qu'observez-vous ?
- ▷ En résumé, que fait la commande `rd.random` ?

La commande `rd.random` permet de simuler de nombreux phénomènes aléatoires pourvu qu'on la modifie correctement.

- ▷ A l'aide de la commande `rd.random`, écrire des instructions permettant de renvoyer un nombre choisi au hasard dans l'intervalle $]0, 2[$.

- ▷ A l'aide de la commande `rd.random`, écrire des instructions permettant de renvoyer un nombre choisi au hasard dans l'intervalle $]0, 7[$.

- ▷ A l'aide de la commande `rd.random`, écrire des instructions permettant de renvoyer un nombre choisi au hasard dans l'intervalle $]2, 7[$.

4 Simulation de variable de Bernoulli

Il n'existe pas de fonction `rd.bernoulli` dans Python. Cependant nous verrons ici deux manières de simuler une loi de Bernoulli : en passant par l'utilisation de `rd.binomial` et en passant par l'utilisation de `rd.random`.

4.1 A l'aide de `rd.binomial`

- ▷ Rappeler le lien entre loi binomiale de paramètre n et p et loi de Bernoulli de paramètre p .

- En déduire une commande permettant de simuler une épreuve de Bernoulli de paramètre p .

Exercice 4. On considère une urne contenant 4 boules bleues et 5 boules rouges indiscernables au toucher. Ecrire une fonction qui simule le tirage d'une boule dans cette urne.

4.2 A l'aide de `rd.random`

- ▷ Soit $p \in]0, 1[$. Si l'on prend un nombre x au hasard compris entre $]0, 1[$ quelle est la probabilité que $x < p$?

- ▷ Expliquer alors pourquoi la fonction suivante permet de simuler une épreuve de Bernoulli de paramètre p :

```

1 def Bernoulli(p):
2     x=rd.random()
3     if x<p:
4         return 1
5     else:
6         return 0

```

- ◁ En s'inspirant de ce programme : écrire un programme simulant une variable aléatoire X telle que $X(\Omega) = \{-1, 1\}$ telle que $P(X = -1) = p$ et $P(X = 1) = 1 - p$.

Exercice 5. Reprendre l'exercice 2 en utilisant la commande `rd.random`

Exercice 6. Soit X suivant une loi binomiale de paramètres n et p . Ecrire un programme simulant X sans utiliser la fonction `rd.binomial` mais en utilisant `rd.random`

5 Modélisation Python dans un problème

Exercice 7. Un mobile se déplace sur les points à coordonnées entières d'un axe d'origine 0. A l'instant 0, le mobile est à l'origine. Le mobile se déplace selon la règle suivante : s'il est sur le point d'abscisse k à l'instant n , alors à l'instant $n + 1$ il sera sur le point d'abscisse $(k + 1)$ avec la probabilité $\frac{2}{3}$ **ou** sur le point d'abscisse 0 avec probabilité $\frac{1}{3}$.

Pour tout $n \in \mathbb{N}^*$, on note X_n la variable aléatoire égale à l'abscisse de ce point à l'instant n et l'on a donc $X_0 = 0$.

1. Compléter la fonction suivante pour qu'elle simule l'expérience aléatoire étudiée et renvoie la valeur prise par X_n , où l'entier n est le paramètre d'entrée.

```
1 def simu(n):  
2     x=0  
3     for k in range(n):  
4         if .....:  
5             x=.....  
6         else :  
7             x=.....  
8     return x
```

2. Modifier la fonction précédente afin qu'elle renvoie le vecteur contenant toutes les abscisses du mobile.
3. Dans la console, exécuter la fonction avec $n = 100$. Quelle est l'abscisse minimale et maximale dans votre simulation ? N'hésitez pas à l'exécuter plusieurs fois !

On pourra utiliser les commandes `np.max` et `np.min`