

Simulations de lois usuelles

Probabilités

Le but de ce TD est de passer en revue quelques lois de probabilités usuelles, et :

- de simuler quelques unes de leurs caractéristiques (probabilité d'obtenir un élément de l'espace des observables, espérance, variance),
- de comparer les résultats simulés avec les résultats donnés ou démontrés en cours.

————— ○ —————

Rappels : soit X une variable aléatoire réelle définie sur l'univers Ω . On note $X(\Omega) = \{x_1, \dots, x_p\}$ l'univers image.

$$\text{Espérance de } X : \mathbb{E}[X] = \sum_{k=1}^p x_k \mathbb{P}(X = x_k)$$

$$\text{Variance de } X : \text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

L'espérance représente le gain moyen qu'on peut espérer obtenir après un grand nombre d'essais.

La variance mesure l'étalement moyen de X autour de son espérance.

————— ○ —————

1 Loi uniforme

1.1 Rappels sur la loi uniforme

Une variable aléatoire qui peut prendre n valeurs possibles $x_1, x_2, \dots, x_n$ suit une loi uniforme lorsque la probabilité de n'importe quelle valeur x_i est égale à $\frac{1}{n}$.

Un exemple simple de loi discrète uniforme est le lancer d'un dé non biaisé à 6 faces. Les valeurs possibles de x sont 1, 2, 3, 4, 5, 6 ; et à chaque fois que le dé est lancé, la probabilité d'un score donné est égale à $\frac{1}{6}$.

1.2 Caractéristiques de la loi uniforme

On choisit le cas pour lequel $X(\Omega) = \{x_1, \dots, x_n\} = \{1, 2, \dots, n\}$.

$$\mathbb{P}(X = x) = \frac{1}{n} \quad \forall x \in \{1, 2, \dots, n\}$$

$$\text{Espérance : } \mathbb{E}[X] = \frac{n+1}{2}$$

$$\text{Variance : } \text{Var}(X) = \frac{n^2 - 1}{12}$$

Les formules ci-dessus ne sont pas au programme, mais vous pouvez les démontrer. Je rappelle quelques éléments de calcul pour l'espérance et la variance : $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ et $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$.

1.3 Simulation informatique de la loi uniforme

Activité 1 écriture d'une fonction simulant une loi uniforme

nom : uniforme

arguments : n (int)

effet : doit renvoyer un élément de l'ensemble des observables $X(\Omega) = \{1, 2, \dots, n\}$, lorsqu'on effectue un tirage pour une loi uniforme.

Pour rappel, la méthode `np.random.randint(1,n)` renvoie un entier aléatoire entre 1 inclus et n exclus.

Activité 2 écriture d'une fonction simulant une estimation de probabilité**nom :** probabilite_unif**arguments :** x,n,N (int,int,int)**effet :** doit renvoyer une estimation de la probabilité d'obtenir $X \in X(\Omega) = \{1, 2, \dots, n\}$, lorsqu'on effectue N tirages pour une loi uniforme.On effectuera des tests avec un grand nombre de tirages (typiquement $N = 100000$).

Les résultats obtenus sont-ils logiques ?

Activité 3 écriture d'une fonction simulant le calcul de l'espérance d'une loi uniforme**nom :** esperance_unif**arguments :** n,N (int,int)**effet :** doit renvoyer une estimation de l'espérance d'une loi uniforme avec $X \in X(\Omega) = \{1, 2, \dots, n\}$. On simulera N tirages.

Les résultats sont-ils cohérents avec la formule donnée préalablement ?

Activité 4 écriture d'une fonction simulant le calcul de la variance d'une loi uniforme**nom :** variance_unif**arguments :** n,N (int,int)**effet :** doit renvoyer une estimation de la variance d'une loi uniforme avec $X \in X(\Omega) = \{1, 2, \dots, n\}$. On simulera N tirages.

Les résultats sont-ils cohérents avec la formule donnée préalablement ?

2 Loi de Bernoulli

2.1 Rappels sur la loi de Bernoulli

La loi de Bernoulli désigne la loi de probabilité d'une variable aléatoire discrète qui prend la valeur 1 (succès) avec la probabilité $p \in [0, 1]$ et 0 (échec) avec la probabilité $q = 1 - p$.

Si X représente la variable aléatoire discrète qui prend la valeur 1 en cas de succès et 0 en cas d'échec, alors l'ensemble des valeurs de la variable aléatoire s'écrit $X(\Omega) = \{0, 1\}$. Pour rappel, $X(\Omega)$ s'appelle l'univers image. On a alors $\mathbb{P}(X = 1) = p$ et $\mathbb{P}(X = 0) = 1 - p$.

Par exemple, dans le jeu du pile ou face, le lancer d'une pièce de monnaie bien équilibrée tombe sur pile avec une probabilité $\frac{1}{2}$ et sur face avec une probabilité $\frac{1}{2}$. Une pièce peut ne pas être équilibrée et dans ce cas, on obtient pile avec une probabilité $p \neq \frac{1}{2}$ et face avec une probabilité $q = 1 - p \neq \frac{1}{2}$. En désignant pile par 1 et face par 0, on obtient une loi de Bernoulli.

2.2 Caractéristiques de la loi de Bernoulli

$$\begin{cases} \mathbb{P}(X = 1) = p \\ \mathbb{P}(X = 0) = 1 - p \end{cases}$$

$$\text{Espérance : } \mathbb{E}[X] = p$$

$$\text{Variance : } \text{Var}(X) = p(1-p)$$

2.3 Simulation informatique de la loi de Bernoulli

Activité 5 écriture d'une fonction simulant une loi de Bernoulli**nom :** bernoulli**arguments :** p (float)**effet :** doit renvoyer un élément de l'ensemble des observables $X(\Omega) = \{0, 1\}$, lorsqu'on effectue un tirage pour une loi de Bernoulli, caractérisée par un succès de probabilité p.

Pour rappel, la méthode random() renvoie un réel aléatoire entre 0 et 1. Elle est appellable par np.random.random()

Activité 6 écriture d'une fonction simulant une estimation de probabilité**nom :** probabilite_ber**arguments :** k,p,N (int,float,int)**effet :** doit retourner une estimation de la probabilité d'obtenir la valeur k, pour tout $k \in X(\Omega)$ quand X est une variable aléatoire qui suit une loi de Bernoulli de paramètre p. N correspond au nombre de tirages (ou « expériences ») effectués pour estimer cette probabilité.

On effectuera des tests avec un grand nombre de tirages (typiquement $N = 100000$).
Les résultats obtenus sont-ils logiques ?

Activité 7 écriture d'une fonction simulant le calcul de l'espérance d'une loi de Bernoulli de paramètre p. On simulera N tirages.**nom :** esperance_ber**arguments :** p,N (float,int)**effet :** doit renvoyer une estimation de l'espérance d'une loi de Bernoulli de paramètre p. On simulera N tirages.

Les résultats sont-ils cohérents avec la formule du cours ?

Activité 8 écriture d'une fonction simulant le calcul de la variance d'une loi de Bernoulli**nom :** variance_ber**arguments :** p,N (float,int)**effet :** doit renvoyer une estimation de la variance d'une loi de Bernoulli, caractérisée par un succès de probabilité p.
On simulera N tirages.

Les résultats sont-ils cohérents avec la formule du cours ?

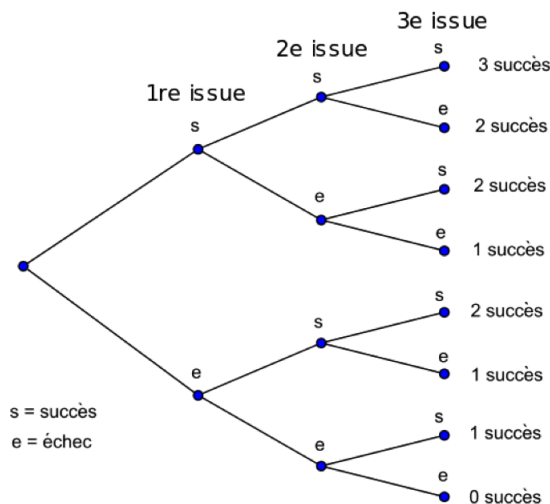
3 Loi binomiale

3.1 Rappels sur la loi binomiale

On considère la situation où une expérience aléatoire (deux résultats possibles et une probabilité fixe) est répétée un certain nombre de fois de manière indépendante; notons n ce nombre de fois. Cette répétition indépendante d'épreuves de Bernoulli s'appelle un *schéma de Bernoulli* ou simplement des *épreuves de Bernoulli*.

Une **loi binomiale** décrit le nombre de fois où le succès apparaît sur les n expériences effectuées. Le nombre de succès obtenus étant une valeur aléatoire, une loi binomiale est décrite grâce à la donnée des probabilités que le succès apparaisse précisément k fois sur les n essais.

Une manière visuelle de trouver ces probabilités est de construire un arbre de probabilité. La figure ci-dessous présente un arbre de probabilité pour une loi binomiale associée à $n = 3$ épreuves de Bernoulli.



Chaque épreuve est représentée par deux branches : l'une pour le succès, l'autre l'échec. À chaque extrémité, on rajoute deux branches (succès et échec) pour l'épreuve suivante. On recommence jusqu'au nombre total d'épreuves. À chaque extrémité finale, on peut compter le nombre de succès obtenus. Il suffit de multiplier le nombre de fois où il y a k succès par la probabilité d'obtenir k succès pour obtenir la probabilité correspondante de la loi binomiale. Ainsi la loi binomiale, de paramètres n et p , est la loi de probabilité discrète d'une variable aléatoire X vérifiant

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1 - p)^{(n-k)}$$

pour $k \in \{0, 1, 2, \dots, n\}$ avec $\binom{n}{k} = C_n^k = \frac{n!}{k!(n-k)!}$.

3.2 Caractéristiques de la loi binomiale

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1 - p)^{(n-k)}$$

$$\text{Espérance : } \mathbb{E}[X] = np$$

$$\text{Variance : } \text{Var}(X) = np(1-p)$$

3.3 Simulation informatique de la loi binomiale

Activité 9 écriture d'une fonction simulant une loi binomiale

nom : binomiale

arguments : n, p (int, float)

effet : doit renvoyer un élément de l'ensemble des observables $X \in X(\Omega) = \{0, 1, 2, \dots, n\}$, lorsqu'on effectue n tirages pour une loi binomiale, caractérisée par un succès de probabilité p .

Activité 10 écriture d'une fonction simulant une estimation de probabilité

nom : probabilite_bin

arguments : k, n, p, N (int, int, float, int)

effet : doit renvoyer une estimation de la probabilité d'obtenir k succès de l'ensemble des observables lorsqu'on effectue n tirages pour une loi binomiale. La loi de Bernoulli est caractérisée par un succès de probabilité p , et on effectue au total N tirages (ou « expériences »).

On effectuera des tests avec un grand nombre de tirages (typiquement $N = 100000$).

Les résultats obtenus sont-ils logiques ? On pourra estimer toutes les probabilités possibles pour $n=3$ et $p=0.5$, et les comparer avec les résultats attendus.

Activité 11 écriture d'une fonction simulant le calcul de l'espérance d'une loi binomiale**nom** : `esperance_bin`**arguments** : `n,p,N` (int,float,int)**effet** : doit renvoyer une estimation de l'espérance d'une loi binomiale de paramètres `n` et `p`. On simulera `N` tirages.

Les résultats sont-ils cohérents avec la formule du cours ?

Activité 12 écriture d'une fonction simulant le calcul de la variance d'une loi binomiale**nom** : `variance_bin`**arguments** : `n,p,N` (int,float,int)**effet** : doit renvoyer une estimation de la variance d'une loi binomiale de paramètres `n` et `p`. On simulera `N` tirages.

Les résultats sont-ils cohérents avec la formule du cours ?

4 Loi hypergéométrique (plus difficile)

4.1 La loi hypergéométrique

La loi hypergéométrique de paramètres associés `n`, `p` et `N`, est une loi de probabilité discrète, décrivant le modèle suivant :

On tire simultanément (c'est à dire sans ordre, et sans remise) `n` boules dans une urne contenant $N_1 = pN$ boules gagnantes et $N_2 = qN = (1 - p)N$ boules perdantes (soit un nombre total de boules valant `N`). Il est ici implicite que N_1 et N_2 sont des entiers.

On compte alors le nombre de boules gagnantes extraites et on appelle `X` la variable aléatoire donnant ce nombre.

On peut illustrer cette loi par l'exemple suivant : un lac renferme 100 poissons dont un quart sont des brochets. On pêche 10 poissons *sans ordre et sans remise à l'eau*, et on considère que pêcher un brochet est un succès (la boule gagnante).

On a alors `N = 100`, `p = 0.25` et `n = 10`.

4.2 Caractéristiques de la loi hypergéométrique

Les caractéristiques de la loi hypergéométrique (comme la loi elle-même) n'est pas à votre programme de mathématiques. Les résultats ne sont pas présentés.

4.3 Simulation informatique de la loi hypergéométrique

Je vous propose trois façons de modéliser une loi hypergéométrique. À vous de choisir le point de vue qui vous convient le mieux lorsqu'il s'agira d'écrire le code associé.

1. On crée un vecteur de taille `N`, et on place pN fois un 1 en position aléatoire. On somme ensuite, parmi les `n` premiers éléments du vecteur, le nombre de fois où l'on trouve 1.
2. On remplit une liste initialement vide, avec des entiers aléatoires distincts compris entre 0 et `N-1` inclus, pour obtenir une liste de longueur `N`. On comptabilise ensuite les boules gagnantes sur l'intervalle `[0,n[` : c'est, sur cet intervalle, le nombre de boules avec un nombre inférieur à pN .
3. On fabrique une liste avec les boules gagnantes, puis perdantes, les unes à la suite des autres, en bloc. Ensuite, on tire au hasard une boule, et on met la boule dans la liste 'tirage'. En retirant la boule, on la supprime effectivement de la liste de départ (utiliser `pop`).

Pour rappel, si `L=[1,2,3]`, la commande `L.pop(2)` retourne l'entier 3, et laisse `L` égale à `[1,2]`.

Activité 13 écriture d'une fonction simulant une loi hypergéométrique**nom** : hyper_geometrique**arguments** : n,p,N (int,float,int)**effet** : doit renvoyer un élément de l'ensemble des observables $X \in X(\Omega) = \{0, 1, 2, \dots, pN\}$, lorsqu'on effectue un tirage pour une loi hypergéométrique.**Activité 14** écriture d'une fonction simulant le calcul de l'espérance d'une loi hypergéométrique**nom** : esperance_hyper_geom**arguments** : n,p,N,M (int,float,int,int)**effet** : doit renvoyer une estimation de l'espérance d'une loi hypergéométrique de paramètres n, p et N. On simulera M tirages.**Activité 15** écriture d'une fonction simulant le calcul de la variance d'une loi hypergéométrique**nom** : variance_hyper_geom**arguments** : n,p,N,M (int,float,int,int)**effet** : doit renvoyer une estimation de la variance d'une loi hypergéométrique de paramètres n, p et N. On simulera M tirages.

4.4 Problématiques étudiées

On se propose de résoudre deux problématiques :

- Tant qu'à partir à la pêche au saumon, le pêcheur aimerait bien savoir l'espérance de la loi décrite avec les paramètres précédents, à savoir le nombre de saumons qu'il peut espérer obtenir.
- Vérifier que lorsque N tend vers l'infini, la loi hypergéométrique converge vers une loi binomiale de paramètres n et p.

Pour ce dernier point, intuitivement, pour N grand, tirer simultanément n boules revient à effectuer n fois une épreuve de Bernoulli dont la probabilité de succès serait p (p est la proportion de boules gagnantes dans l'ensemble des boules) : il est très peu probable de retomber sur la même boule, même si on la replace dans l'urne.

5 Loi géométrique (plus difficile)

5.1 La loi géométrique

La loi géométrique est la loi du nombre X d'épreuves de Bernoulli indépendantes de probabilité de succès p nécessaire pour obtenir le **premier succès**. X est la variable aléatoire donnant le rang du premier succès. L'univers image de la loi est alors $X(\Omega) = \{1, 2, 3, \dots\}$.

Bien évidemment, plus p est petit, plus la probabilité d'obtenir un succès est faible, et plus la valeur de X risque d'être élevée.

5.2 Simulation informatique de la loi hypergéométrique

Activité 16 écriture d'une fonction simulant une loi hypergéométrique**nom** : geometrique**arguments** : p (float)**effet** : doit renvoyer un élément de l'ensemble des observables $X \in X(\Omega) = \{1, 2, 3, \dots\}$, lorsqu'on effectue un tirage pour une loi géométrique.