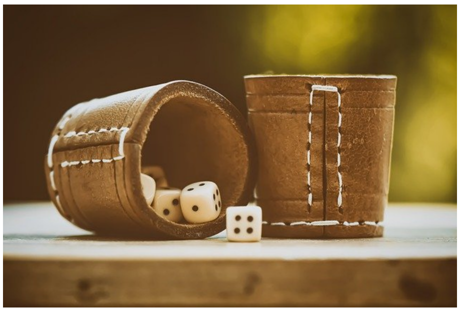
Chapitre 8 - Probabilité



1 loi de probabilité - modélisation

1.1 expérience aléatoire

définition

1 expérience aléatoire (EA) est 1 expérience qui a les caractéristiques suivantes :

- les résultats possibles sont connues
- le résultat n'est pas connu à l'avance
- répétable indéfiniment sans changement

vocabulaire

- EA : voir supra
- issue : une des possibilités de l'EA
- univers, noté Ω : ensemble des issues possibles

ex :

- EA : on jette une pièce équilibrée qui fait Pile ou Face
- issue : Pile ou Face
- univers : $\Omega = \{ \text{Pile} ; \text{Face} \}$

1.2 loi de probabilité

définition

1 expérience aléatoire (EA) est 1 expérience qui a les caractéristiques suivantes :

- les résultats possibles sont connues
- le résultat n'est pas connu à l'avance
- répétable indéfiniment sans changement

ex :

- EA : on jette 1d6 pipé (le 6 a 3 fois plus de chance de sortir que les autres)
- univers : $\Omega = \{1, 2, 3, 4, 5, 6\}$
- loi de probabilité :

X = k	1	2	3	4	5	6	
p(X=k)							

cas particulier : équiprobabilité

- si pour l'EA, chaque issue a la même probabilité de sortie, on parle de **loi équiprobable**
- ex : on lance 1d6 équilibré

2 évènement

2.1 définition - propriété

définition

- **évènement** : sous-ensemble de l'univers (regroupement de 1 ou plusieurs issues)
- **probabilité d'1 évènement** : probabilité associé ce sous-ensemble (en fonction de la loi)

ex 1

- 1 étude sur le groupe sanguin donne :

X = k	A	B	AB	O
p(X=k)	0.45	0.09	0.04	0.42

- $\Longrightarrow p(B) = 0.09$

ex 2 : équiprobabilité

- on lance 1d6 équilibré
- $p(le\_rsultat\_est\_pair) = \dfrac{nbre\_cas\_favorable}{nbre\_cas\_possible} = \dfrac{3}{6} = 0.5$

2.2 opération sur les évènements

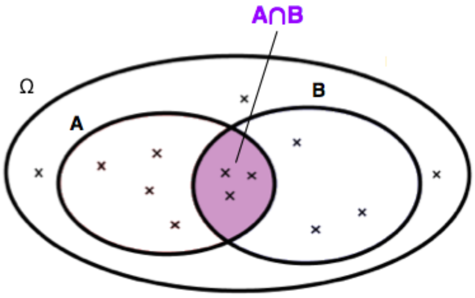
définition - notation

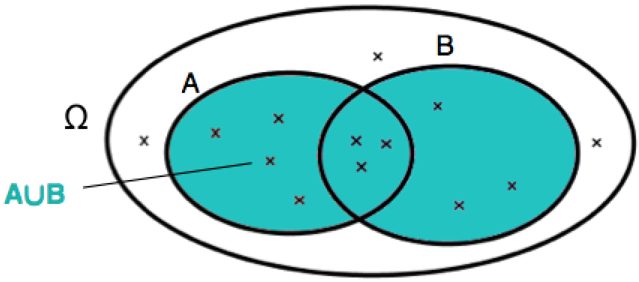
- **contraire de A** : $\overline{A}$
- **réunion de A et B** : $A \cup B$
- **intersection de A et B** : $A \cap B$
- si $p(A \cup B) = 0$ on dit que A et B sont **disjoints**

propriété

- $\boxed{p(A \cup B) = p(A) + p(B) + p(A \cap B)}$
- **inversion 1** : $\overline{A \cup B} = \overline{A} \cap \overline{B}$
- **inversion 2** : $\overline{A \cap B} = \overline{A} \cup \overline{B}$

visualisation graphique





ex : lancé de 2 dés

- on lance 2d6 équilibrés ; définir l'univers, la loi de probabilité

- A = la somme est paire et B = le premier dé est impair
- calculer $p(A)$, $p(B)$, $p(\overline{A})$, $p(A \cap B)$, $p(A \cup B)$ et $p(A \cup \overline{B})$

3 simulation - estimation

3.1 échantillon - simulation

définition

- on considère 1 EA que l'on refait plusieurs fois
- **échantillon** : l'ensemble des résultats des EA
- **taille de l'échantillon** : nombre de fois où on a refait l'EA

simulation informatique

- **simulation informatique** : au lieu de faire (physiquement) l'EA, il est plus rapide (et moins cher) de la simuler par ordinateur
- en python :
  - random.random() : donne 1 nombre aléatoire entre 0 et 1
  - random.randint(a,b) : donne 1 nombre entier aléatoire en a et b
  - random.choice{} : choisit 1 élément de l'ensemble au hasard
  - on visitera le site d'émilie sur le sujet

ex 1 : lancés de d6

- on lance 1 d6 10 fois
- programme python :

```python
# chargement du module random
import random

# lancé d'1 dé
def lancerUnDe(n):
  d = random.randint(1,n)
  return d

# lancé de plusieurs dés
def lancerDeDes(nbDes,nbFaces):
  listeDesDes = []              # liste des dés (vide au départ)
  for i in range(nbDes):
    d = lancerUnDe(nbFaces)     # on lance un dé
    listeDesDes.append(d)       # on ajoute ce dé à la liste
  return listeDesDes

print(lancerDeDes(10,6))
```

- résultats d'1 échantillon : [5, 1, 1, 1, 5, 2, 6, 5, 4, 2]

ex 2 : salade de fruits

- on dispose de 3 fruits : apple, banana et cherry
- on fabrique une salade de fruit avec 12 ingrédients choisis au hasard (qui peuvent être répété)
- programme python pour la recette :

```python
import random

ma_liste_de_fruit = ["apple", "banana", "cherry"]

# choix des ingrédients
def recette(liste_fruit, nb_ingredient):
  recette = []                        # recette vide
  for i in range(nb_ingredient):
    d = random.choice(liste_fruit)    # on choisit 1 fruit
    recette.append(d)                  # on l'ajoute à la recette
  return recette


print(recette(ma_liste_de_fruit,12))
```

- résultats d'1 échantillon :

```
['banana', 'cherry', 'apple', 'banana', 'cherry', 'banana', 'cherry', 'apple', 'cherry', 'cherry', 'banana', 'banana']
```

ex 3 : lancé d'1 pièce de monnaie équilibrée

- voir mon article sur petitfuté.com

3.2 fluctuation - estimation

définition - propriété

- lorsque l'on répète 1 EA, les échantillons ne sont pas identiques ; c'est ce que l'on appelle la **fluctuation d'échantillons**
- cependant, grâce à la **loi des grands nombres**, on peut préciser les choses

théorème de la loi des grands nombres

- soit 1 EA où on suit l'évènement A ; on réalise 1 échantillon de taille n
- $p = p(A)$, la probabilité de réalisation de A
- f_A, la fréquence de A dans l'échantillon
- la loi des nombres nous dit 2 choses :
  - $f_A \longrightarrow p$ lorsque $n \longrightarrow \infty$
  - il est fort probable (à 95% de chance) que $f_A \in [p - \frac{1}{n}, p + \frac{1}{n}]$

utilisation de la loi des grands nombres par un exemple

- on lance 8d6 et on cherche la probabilité p que la somme 25
- **question** : comment estimer p ?
- **réponse** :

- réaliser un échantillon de taille 10000
- calculer la fréquence f d'apparition de 25 dans l'échantillon
- d'après la loi des grands nombres, il y a 95% de chance que $p \in f \pm 0.01$
- ceci est 1 **estimation** relativement précise et fiable de p
-

```python
import random
import math

def lancer_un_de(n):
    d = random.randint(1,n)
    return d

def somme_face(nb_de,nb_face):
    liste_de_de = []    #la liste des dés, pour l'instant  vide
    for i in range(nb_de):
        d = lancer_un_de(nb_face) #on lance un dé
        liste_de_de.append(d)     #on ajoute ce dé à la liste
        somme = sum(liste_de_de)
    return somme

def frequence_echantillon(taille_echantillon,somme_visee,nb_de,nb_face):
    compteur = 0
    for i in range(taille_echantillon):
      if somme_face(nb_de,nb_face)==somme_visee:
        compteur += 1
    f = compteur/taille_echantillon
    return f

print('recherche de la probabilité de d\'obtenir 25 avec 8d6')
print('p appartient l\'intervalle [ ',frequence_echantillon(10000,3,8,6)-1/math.sqrt(10000), ',' ,frequence_echantillon(10000,3,3,2)+1/math.sqrt(10000),' ]')
```

-

```
recherche de la probabilité de d'obtenir 25 avec 8d6
p appartient l'intervalle [  -0.01 , 0.136  ]
```

4 Un peu de python

4.1 pour aller plus loin

- regarder le site d'émilie python
- regarder le site du petit futé informatique