

Chapitre 7 - 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 : 1 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

- soit 1 EA d'univers Ω
- on doit maintenant définir la chance de réussite de chaque issue
- la loi de probabilité p est 1 fonction
- elle indique pour chaque issue $i \in \Omega$ la probabilité $p_i \in [0, 1]$ de réalisation
 - $p : \Omega \longrightarrow [0, 1]$
 - $p(\Omega) = 1$ et $p(\emptyset) = 0$
 - HP (Σ -additivité) : pour tout I dénombrable, et les A_i disjoints 2 à 2

$$p\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} p(A_i)$$
- l'ensemble (Ω, p) s'appelle 1 espace probabilisé

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 se réaliser, 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

- $\Rightarrow p(B) = 0.09$

ex 2 : équiprobabilité

- on lance 1d6 équilibré
- $p(\text{le_resultat_est_pair}) = \frac{\text{nbre_cas_favorable}}{\text{nbre_cas_possible}} = \frac{3}{6} = 0.5$

2.2 opération sur les évènements

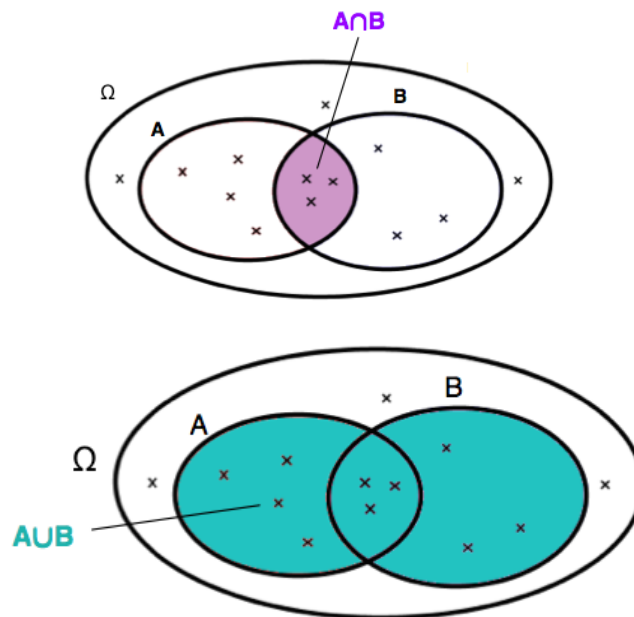
définition - notation

- contraire de A : \bar{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

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

visualisation graphique



ex : lancé de 2 dés

- on lance 2d6 équilibrés ; définir l'univers et 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 (a et b sont compris dans le choix)
 - `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 :

```
1 # chargement du module random
2 import random
3
4 # lancé d'1 dé
5 def lancerUnDe(n):
6     d = random.randint(1,n)
7     return d
8
9 # lancé de plusieurs dés
10 def lancerDeDes(nbDes,nbFaces):
11     listeDesDes = [] # liste des dés (vide au départ)
12     for i in range(nbDes):
13         d = lancerUnDe(nbFaces) # on lance un dé
14         listeDesDes.append(d) # on ajoute ce dé à la liste
15     return listeDesDes
16
17 print(lancerDeDes(10,6))
```

- résultat 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 1 salade de fruit avec 12 ingrédients choisis au hasard (qui peuvent être répétés)
- programme python pour la recette :

```

1  import random
2
3  ma_liste_de_fruit = ["apple", "banana", "cherry"]
4
5  # choix des ingrédients
6  def recette(liste_fruit, nb_ingredient):
7      recette = []                # recette vide
8      for i in range(nb_ingredient):
9          d = random.choice(liste_fruit) # on choisit 1 fruit
10         recette.append(d)           # on l'ajoute à la recette
11     return recette
12
13
14 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 cet 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

```

1 import random
2 import math
3
4 def lancer_un_de(n):
5     d = random.randint(1,n)
6     return d
7
8 def somme_face(nb_de,nb_face):
9     liste_de_de = [] #la liste des dés, pour l'instant vide
10    for i in range(nb_de):
11        d = lancer_un_de(nb_face) #on lance un dé
12        liste_de_de.append(d) #on ajoute ce dé à la liste
13        somme = sum(liste_de_de)
14    return somme
15
16 def frequence_echantillon(taille_echantillon,somme_visee,nb_de,↵
    nb_face):
17     compteur = 0
18     for i in range(taille_echantillon):
19         if somme_face(nb_de,nb_face)==somme_visee:
20             compteur += 1
21     f = compteur/taille_echantillon
22     return f
23
24 print('recherche de la probabilité de d\'obtenir 25 avec 8d6')
25 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