

## Fiche P2.3 : Loi de Bernoulli & Simulation

### 1 – Loi de Bernoulli

On considère une épreuve de Bernoulli de paramètre  $p$

**Définition 1** : On dit qu'une variable aléatoire  $X$  suit la **loi de Bernoulli de paramètre  $p$**  si elle vaut 1 en cas de succès et 0 en cas d'échec à l'épreuve précédente.

- **Notation** : On note alors  $X \sim b(p)$
- **Loi de probabilité** : La loi de Bernoulli de paramètre  $p$  est donc donnée par le tableau suivant :

<b>Valeurs</b>	0	1	<b>Total</b>
<b>Probabilités</b>	$1 - p$	$p$	1

- $P(X = 1) = p$
- $P(X = 0) = 1 - P(X = 1) = 1 - p$

- **Espérance** :  $E(X) = 0 \times (1 - p) + 1 \times p = p$

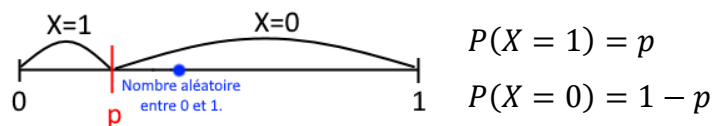
**Exemple 1** : On lance un dé bien équilibré. Soit  $X$  la variable aléatoire qui vaut 1 lorsque le dé tombe sur la face « 6 » et qui vaut 0 sinon.

- $X$  suit la loi de Bernoulli de paramètre  $p = 1/6$
- $P(X = 1) = 1/6$  et  $P(X = 0) = 1 - \frac{1}{6} = \frac{5}{6}$
- $E(X) = p = 1/6$

### 2 – Simulation d'une variable de Bernoulli

- Il est possible de **simuler** informatiquement une variable de Bernoulli c'est-à-dire de créer avec l'ordinateur une variable aléatoire qui suivra la loi de Bernoulli.
- Pour cela on utilise un générateur de nombre aléatoire qui choisi aléatoirement et de manière **uniforme** un nombre entre 0 et 1 : Avec le tableur on utilise la fonction  $ALEA()$  et avec Python la fonction  $random()$ .
- Pour créer une variable de Bernoulli  $X$  de paramètre  $p$  on utilise alors l'algorithme suivant :

« Si le nombre aléatoire est inférieur à  $p$  alors  $X$  vaut 1 sinon  $X$  vaut 0. »



. Avec le tableur, on peut entrer dans une cellule la formule suivante :  $=SI(ALEA() < p; 1; 0)$

. Avec Python, on peut créer une fonction  $bernoulli(p)$  :

```
from random import random

def bernoulli(p):
    if random() < p:
        return 1
    else:
        return 0
```

**Exemple 2** : Pour simuler le lancer d'un dé on peut utiliser les instructions précédentes avec  $p = 1/6$

### 3 – Echantillon de $n$ variables de Bernoulli

- Pour créer un échantillon de  $n$  variables de Bernoulli, il suffit de répéter  $n$  fois les instructions précédentes
  - . Avec le tableur, on utilise le **recopiage automatique** pour remplir  $n$  cellules avec la formule précédente.
  - . Avec Python on utilise une **boucle** et on peut remplir une liste avec les différents résultats obtenus :

```
def echantillon(n,p):
    L=[]
    for i in range(n):
        L.append(bernoulli(p))
    return(L)
```

- On peut alors facilement compter le nombre de succès puis donner la **fréquence** de succès.
  - . Avec le tableur, on utilise la formule = *SOMME(plage de cellule de l'échantillon)/n*
  - . Avec Python, on utilise la fonction *sum()* pour une liste : *frequence=sum(echantillon(n,p))/n*

Exemple 3 : Pour simuler 100 lancers d'un dé, on peut utiliser ces instructions avec  $n = 100$  et  $p = 1/6$ . On obtient alors la fréquence de « 6 » sur les 100 lancers.

### 4 – Etude des échantillons

- Nous allons maintenant simuler  $N$  échantillons de taille  $n$  et observer les fréquences obtenues.
  - . Avec le tableur, on utilise le recopiage automatique pour obtenir les  $N$  échantillons ainsi que la fréquence de succès dans chacun d'entre eux.
  - . Avec Python, on crée une nouvelle liste de taille  $N$  qui va simuler les  $N$  échantillons et stocker au fur et à mesure les différentes fréquences obtenues : *F=[sum(echantillon(n,p))/n for k in range(N)]*
- On observe le phénomène de **fluctuation d'échantillonnage** : La fréquence de succès varie autour de la probabilité théorique  $p$ . Plus la taille  $n$  des échantillons est grande, moins les fluctuations sont importantes.

Exemple 4 : Fréquences obtenues avec l'instruction précédente pour  $N = 1000$  et  $p = 1/6$

• Avec  $n = 20$

```
[0.15, 0.1, 0.1, 0.1, 0.2, 0.3, 0.15, 0.2, 0.15,
0.1, 0.3, 0.3, 0.1, 0.25, 0.2, 0.3, 0.15, 0.1,
0.25, 0.1, 0.05, 0.15, 0.2, 0.15, 0.1, 0.15, 0
.15, 0.05, 0.1, 0.1, 0.1, 0.2, 0.25, 0.2, 0.25,
```

• Avec  $n = 100$

```
[0.16, 0.11, 0.22, 0.16, 0.14, 0.19, 0.21, 0.23,
0.18, 0.19, 0.17, 0.15, 0.22, 0.17, 0.21, 0.15,
0.09, 0.16, 0.11, 0.22, 0.15, 0.19, 0.19, 0.14,
0.19, 0.15, 0.08, 0.15, 0.18, 0.16, 0.16, 0.23,
```

• Avec  $n = 500$

```
[0.162, 0.202, 0.164, 0.174, 0.158, 0.152, 0.164
, 0.148, 0.206, 0.148, 0.19, 0.18, 0.184, 0.194,
0.15, 0.168, 0.188, 0.174, 0.17, 0.156, 0.166,
0.172, 0.166, 0.164, 0.168, 0.19, 0.162, 0.232,
```

- On peut quantifier ces fluctuations en fonction de la taille  $n$  des échantillons avec la propriété suivante :

Propriété 1 : Soit  $p$  la probabilité théorique,  $n$  la taille des échantillons et  $s = \frac{1}{2\sqrt{n}}$  alors :

- En moyenne, environ 68% des fréquences sont dans l'intervalle  $[p - s; p + s]$
- En moyenne, environ 95% des fréquences sont dans l'intervalle  $[p - 2s; p + 2s]$
- En moyenne, environ 99% des fréquences sont dans l'intervalle  $[p - 3s; p + 3s]$

Remarque : Le nombre  $s$  coïncide environ à l'**écart-type** observé dans la série des fréquences obtenues.

Exemple 5 : Avec  $p = \frac{1}{6} \approx 0.17$ , pour  $n = 100$  on a  $s = \frac{1}{2\sqrt{100}} = 0.05$

- Environ 68% des fréquences sont dans l'intervalle  $[0.12; 0.22]$
- Environ 95% des fréquences sont dans l'intervalle  $[0.07; 0.27]$
- Environ 99% des fréquences ont dans l'intervalle  $[0.02; 0.32]$

