

## *Distance entre moyenne et espérance* *Loi binomiale*

Auteur : **Hélène Landart**

**Niveaux scolaires :**  
1<sup>ère</sup> / Terminale Générales

**Mots clés :**  
Algorithme/Programmation  
Probabilités/Statistiques

### ENONCE

*Étudier sur des exemples la distance entre la moyenne d'un échantillon simulé de taille  $n$  d'une variable aléatoire et l'espérance de cette variable aléatoire.*

Toutes les fonctions suivantes sont à compléter ou à écrire dans le fichier moesbin.

On n'oubliera pas d'importer le module random au début du script.

1) On considère la fonction python ci-contre:  
Compléter les pointillés pour que cette fonction simule une expérience de Bernoulli de paramètre  $p$ .

```
def bern(p):  
    c=0  
    x=random()  
    if x<...:  
        c=1  
    return c
```

2) Ecrire une fonction binom( $n,p$ ) qui renvoie le nombre de succès de  $n$  expériences de Bernoulli de paramètre  $p$ .

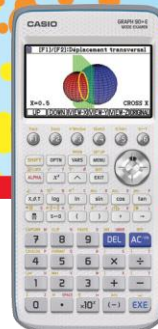
3) Compléter la fonction moy ci-contre pour qu'elle renvoie la moyenne d'un échantillon de taille  $N$  simulant une variable aléatoire de loi Binomiale  $B(n,p)$ .

```
def moy(...):  
    S=0  
    for i in range(N):  
        S=...  
    m=...  
    return m
```

4) Ecrire une fonction dist( $N,n,p$ ) qui calcule la valeur approchée à  $10^{-5}$  (on pourra utiliser la fonction round) de la distance entre la moyenne d'un échantillon de taille  $N$  simulant une variable aléatoire de loi binomiale  $B(n,p)$  et l'espérance de cette variable aléatoire. Tester cette fonction pour  $n=35$  et  $p=0.3$  pour des valeurs de  $N$  de plus en plus grandes. Que remarque-t-on?

5) BONUS: Ecrire une fonction echan( $N,n,p$ ) qui renvoie un échantillon (sous forme de liste) de taille  $N$  simulant la loi  $B(n,p)$ .

# Graph 90+E



- 1) On effectue un tirage aléatoire d'un nombre réel dans l'intervalle  $[0 ; 1[$  avec la fonction `random()`.  
Si ce nombre est inférieur à  $p$ , on obtient un succès : le résultat vaut 1. La probabilité de cet évènement est  $p$ .  
Sinon, c'est un échec : le résultat vaut 0. La probabilité de cet évènement est  $1 - p$ .

```
def bern(p):  
    c=0  
    x=random()  
    if x<p:  
        c=1  
    return c
```

- 2) Pour la fonction `binom` on simule  $n$  expériences de Bernoulli de paramètre  $p$  et on compte le nombre de succès avec la variable  $s$ .

```
def binom(n,p):  
    s=0  
    for k in range(n):  
        s=s+bern(p)  
    return s
```

- 3) Pour la fonction `moy`, on répète  $N$  fois cette expérience sur la loi binomiale et on calcule la moyenne du nombre de succès.

```
def moy(N,n,p):  
    S=0  
    for i in range(N):  
        S=S+binom(n,p)  
    m=S/N  
    return m
```

- 4) Il ne reste plus qu'à comparer l'écart entre les moyennes obtenues et l'espérance de la loi binomiale:  $np$ . On utilise pour cela la fonction `abs` (valeur absolue).

```
def dist(N,n,p):  
    d=round(abs(moy(N,n,p)-n*p),5)  
    return d
```

Voici quelques test avec  $N=100$   $N=1000$  et  $N=10000$ . On remarque que plus  $N$  est grand, plus la moyenne se rapproche de l'espérance.

```
>>>dist(100,35,0.3)  
0.17  
>>>dist(1000,35,0.3)  
0.169  
>>>dist(10000,35,0.3)  
0.0024
```

- 5) Il est possible de créer cette fonction en seulement quelques lignes comme ci-dessous:

```
def echan(N,n,p):  
    L=[binom(n,p) for i in range(N)]  
    return L
```