

PYTHON: Espérance, variance et écart type d'une variable aléatoire, compréhension de liste.

Auteur : Eddy HOMRI

Niveaux scolaires:
1^{ère}/ Terminale Générales

Mots clés :
Algorithme / Programmation
Probabilités / Statistiques

ENONCE

1) Soit X une variable aléatoire.

Soit $x = [x_1, \dots, x_n]$ la liste des valeurs possibles de X .

Soit $p = [p_1, \dots, p_n]$ la liste des probabilités correspondantes, c'est-à-dire $P(X = x_i) = p_i$

Écrire un script en Python avec trois fonctions:

- esp(x,p) qui renvoie l'espérance de X
- var(x,p) qui renvoie la variance de X
- sig(x,p) qui renvoie l'écart type de X

2) Soit X une variable aléatoire prenant les valeurs 2 ; 5 ; 7 et 9.

La loi de probabilité de X est résumée dans le tableau ci-dessous :

x_i	2	5	7	9
$P(X = x_i)$	$\frac{1}{5}$	$\frac{3}{10}$	$\frac{3}{10}$	$\frac{1}{5}$

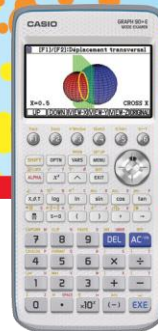
Utiliser le script de la question précédente pour déterminer l'espérance, la variance et l'écart-type de X .

3) a) A partir des listes x et p , que doit on écrire en langage python pour définir une nouvelle liste en compréhension r telle que

$$r = [x_1 * p_1, \dots, x_n * p_n]?$$

Tester votre résultat sur les listes de la question précédente.

Graph 90+E



a) Espérance:

Commençons par rappeler la formule de l'espérance mathématique de X :

$$E(X) = \sum_{i=1}^n x_i p_i$$

Pour calculer cette somme nous allons utiliser une boucle POUR qui portera sur l'indice i . Une boucle POUR est la commande for en langage python.

Nous devons définir le nombre d'itérations (répétitions) nécessaire pour obtenir la valeur de l'espérance mathématique de X .

La variable x étant une liste nous utilisons la commande `len(x)` qui donnera sa taille (le nombre d'éléments présents dans la liste).



En python l'indice du premier terme de la liste est 0 et l'indice du dernier terme est "taille de la liste-1".

Nous allons donc utiliser `for i in range(len(x))`

Après avoir défini les caractéristiques de la boucle, il nous faut une variable pour obtenir la valeur de l'espérance mathématique de X . Pour cela, nous allons utiliser une variable e que l'on initialise à 0 (en dehors de la boucle) et à qui on ajoutera $x_i * p_i$ à chaque tour dans la boucle.

On construit ensuite la boucle POUR.

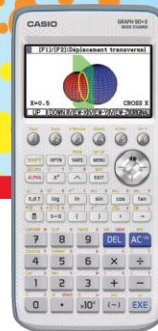


`x[i]` signifie que l'on prend la valeur présente dans la liste x à la position i .

Voici la fonction `esp(x,p)` qui renvoie l'espérance de X .

```
def esp(x,p):  
    e=0  
    for i in range(len(x)):  
        e=e+x[i]*p[i]  
    return e
```

Graph 90+E



b) Variance:

Commençons par rappeler la formule de la variance de X :

$$V(X) = \sum_{i=1}^n (x_i - E(X))^2 p_i$$

De la même manière, pour calculer cette somme nous allons utiliser une boucle POUR qui portera sur l'indice i .

Pour cela nous allons avoir besoin de deux variables:

e : l'espérance de X autrement dit **esp(x,p)**



On pourrait écrire directement **esp(x,p)** dans la boucle à la place d'utiliser la variable e mais dans ce cas le calcul de l'espérance serait fait n fois.

v , qui sera la variable pour obtenir la valeur de la variance de X , que l'on initialise à 0 et à qui on ajoute $(x_i - e)^2 * p_i$ à chaque tour dans la boucle

Voici la fonction **var(x,p)** qui renvoie la variance de X

```
def var(x,p):
    e=esp(x,p)
    v=0
    for i in range(len(x)):
        v=v+(x[i]-e)**2*p[i]
    return v
```



Il peut être intéressant ici de Copier / Coller la définition de la fonction **esp** en utilisant **CLIP** **SHIFT** **8** pour la sélectionner puis **PASTE** **SHIFT** **9** pour la coller. Il suffit ensuite de modifier ce qui doit l'être.

c) Ecart-type:

Commençons par rappeler la formule de l'écart-type de X :

$$\sigma = \sqrt{v}$$

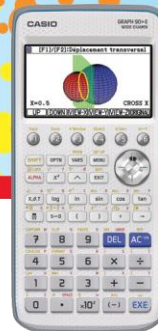
Nous n'avons pas besoin ici de variable. En revanche, nous allons avoir besoin de la fonction racine carrée. Il faut donc importer le module **math**. On rajoute la ligne **from math import *** au tout début du script.

```
proba.py 001/014
from math import*

def esp(x,p):
    e=0
    for i in range(len(x)):
        e=e+x[i]*p[i]
    return e

FILE RUN SYMBOL CHAR A↔a ▶
```

Graph 90+E



La ligne `from math import *` est directement inscrite dans notre catalogue accessible par les touches `SHIFT` puis `4`.

On définit ensuite la fonction `sig` :

```
def sig(x,p):  
    return sqrt(var(x,p))
```

2. Pour lancer le script, accéder à l'onglet `{RUN}` en appuyant sur la touche `F2`.

Il faut enregistrer le script pour pouvoir le lancer. Si vous ne l'avez pas fait, la calculatrice vous invite à le faire en appuyant sur la touche `F1`.

On définit deux listes `x` et `p` avec les valeurs de l'énoncé puis on appelle les trois fonctions.

On obtient: $E = 5,8$, $V = 5,56$ et $\sigma \approx 2,36$

```
* SHELL Initialized *  
>>>from proba import  
>>>x=[2,5,7,9]  
>>>p=[1/5,3/10,3/10,1]
```

```
>>>esp(x,p)  
5.8  
>>>var(x,p)  
5.56  
>>>sig(x,p)  
2.357965224510319
```

3. a) Pour créer la liste `r`, il suffit d'écrire:

```
r=[x[i]*p[i] for i in range(len(x))].
```

En testant directement dans le SHELL avec les listes de la question 2 on obtient bien le bon résultat:

```
>>>r=[x[i]*p[i] for i  
>>>r  
[0.4, 1.5, 2.1, 1.8]
```

b) En utilisant la fonction `sum`, cela donne:

```
def esp(x,p):  
    r=[x[i]*p[i] for i in range(len(x))]  
    return sum(r)
```

On pourra faire la même chose pour la fonction `var`.