

Moyenne, écart-type

Auteur : Thomas Léchenne

Niveaux scolaires :
Seconde

Mots clés :
Algorithme / Programmation
Probabilités / Statistiques

ENONCE

Pour des données réelles ou issues d'une simulation, lire et comprendre une fonction écrite en Python renvoyant la moyenne m , l'écart type s , et la proportion d'éléments appartenant à $[m-2s, m+2s]$.

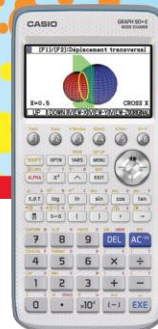
Le programme dans son intégralité :

```
moyenne.py 001/015 ▶
def A(L):
    N=len(L)
    M=0
    for I in range(N):
        M+=L[I]
    m=M/N
    V=0
    for J in range(N):
        V+=(L[J]-m)**2
    s=(V/N)**0.5
    C=0
    for K in range(N):
        if abs(L[K]-m)<=2*s:
            C+=1
    return [m,round(s,3),C/N]
FILE RUN SYMBOL CHAR A↔a ▶
```

Décortiquons cette fonction Python.
Elle se nomme A et a pour argument la variable L. Au vu de la ligne suivante qui définit N comme la longueur de L, L est une liste de valeurs.

```
moyenne.py 001/002
def A(L):
    N=len(L)
FILE RUN SYMBOL CHAR A↔a ▶
```

Graph 90+E



La variable M, initialisée à 0 à la troisième ligne, va, à chaque itération de la boucle Pour, être augmentée de la i+1-ème valeur de la liste. Ainsi la variable M stockera la somme des valeurs de la liste L.

La variable m prend la valeur de la somme des valeurs de la liste divisée par le nombre de valeurs, N. La variable m représente donc la moyenne des valeurs de la liste L.

La variable V, initialisée à 0, est augmentée, à chaque itération de la nouvelle boucle, de l'écart au carré entre la j+1-ème valeur de la liste L et de la moyenne m. Donc, V stockera la valeur de la somme $\sum_{i=1}^n (x_i - \bar{x})^2$.

La notation $**0.5$ signifie "puissance 0,5" et correspond à la racine carrée.

On a donc la variable s qui vaut $\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{V(X)}$ c'est-à-dire l'écart-type.

La variable C est initialisée à 0. Elle est ensuite incrémentée d'un à chaque fois que l'écart entre une valeur de la liste et la moyenne est inférieure ou égale à 2s.

C comptabilise donc le nombre de valeurs de la liste qui appartiennent à l'intervalle $[m - 2s; m + 2s]$.

```
moyenne.py 001/005
def A(L):
    N=len(L)
    M=0
    for I in range(N):
        M+=L[I]
```

```
moyenne.py 004/010
for I in range(N):
    M+=L[I]
m=M/N
V=0
for J in range(N):
    V+=(L[J]-m)**2
s=(V/N)**0.5
```

```
moyenne.py 008/014
for J in range(N):
    V+=(L[J]-m)**2
s=(V/N)**0.5
C=0
for K in range(N):
    if abs(L[K]-m)<=2*s:
        C+=1
```

Graph 90+E



La commande round permet d'arrondir la valeur de s (ici à 3 chiffres après la virgule).

La fonction renvoie donc les valeurs de la moyenne, de l'écart-type (arrondi au millième) et de la proportion des valeurs de la liste qui appartiennent à l'intervalle $[m - 2s; m + 2s]$.

On peut aussi tester cet algorithme avec une liste de valeurs.

```
moyenne.py 015/015 ◀▶
V+=(L[J]-m)**2
s=(V/N)**0.5
C=0
for K in range(N):
    if abs(L[K]-m)<=2*s:
        C+=1
return [m,round(s,3)]
FILE RUN SYMBOL CHAR A↔a ▶
```

```
>>>A([1,2,3,3,3,5])
[2.8333333333333334, 1]
>>>A([10,11,12,15,11])
[11.222222222222222, 1]
>>>A([10,11,11,12,10])
[11.3, 2.452, 0.9]
>>>
RUN A↔a CHAR
```