

Algorithme de calcul de moyenne et d'écart type

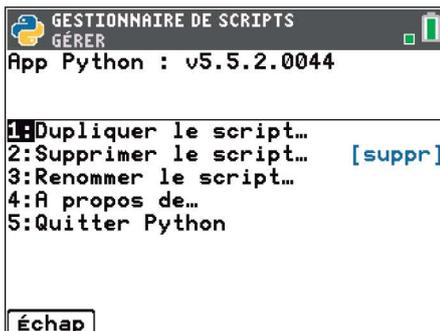
Énoncé

Une entreprise fabrique des petites voitures et les vend par lot de 100. La probabilité qu'un jouet produit présente un léger défaut de peinture a été établi à 28%.

L'entreprise examine un lot et trouve 35 % de jouets présentant un défaut. Elle se demande si sa chaîne de production est dégradée.

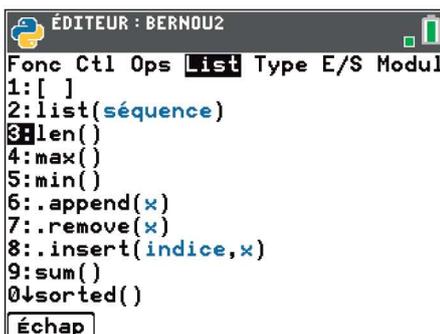
1. A l'aide de l'environnement Python, définissez la fonction **EcartType** qui prend en paramètre la liste **l** et renvoie son écart type. Les fonctions intermédiaires **Moyenne** et **Variance** sont fournies.
2. Définissez la fonction **Proportion** qui prend en paramètres **l**, **a** et **b** où **l** est la liste à examiner, **a** et **b** les bornes distinctes d'un intervalle. La fonction renvoie la proportion, en pourcentage, d'éléments de la liste dans l'intervalle [a;b].
3. A l'aide des fonctions **Echantillon**, **EcartType** et **Proportion**, observez le pourcentage de fréquence appartenant aux intervalles [p-s;p+s], [p-2s;p+2s] et enfin [p-3s;p+3s] où s est la valeur de l'écart type et p la probabilité qu'un jouet soit défectueux.

1. Définition d'EcartType en Python



Nous repartons du script précédent que nous dupliquons à l'aide de l'onglet **Gérer** dans la fenêtre de gestionnaire de script. Il s'appelle désormais **BERNOU2**.

Nous travaillerons en pourcentage arrondi à l'unité, que ce soit pour les fonctions **JouetDefaut** ou **Proportion** de la question suivante.



Les fonctions **sum** et **len** renvoient respectivement les sommes des éléments de la liste passée en paramètre et sa longueur, c'est-à-dire le nombre de ses éléments. Ces fonctions sont accessibles dans le sous menu List du menu **Fns...**.

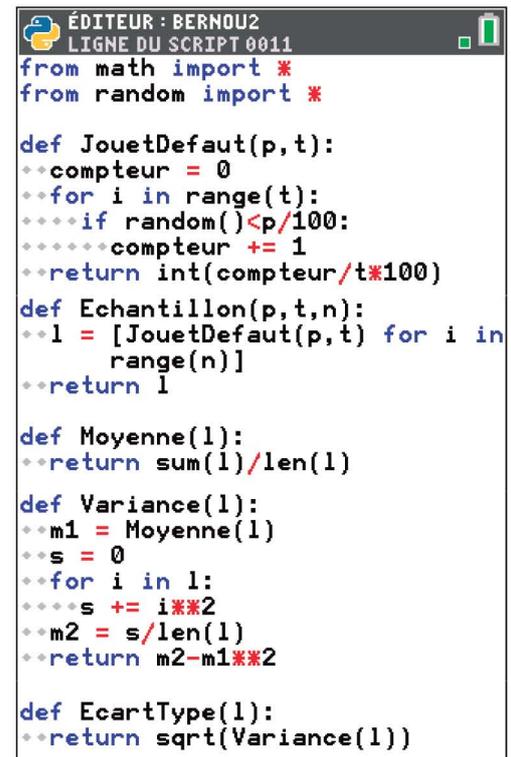
Nous avons choisi une formule de calcul de la variance qui nous permet de nous passer de l'utilisation d'une

liste intermédiaire et fonctionne par accumulateur.

L'import de la bibliothèque **math**, bien que réalisé systématiquement par réflexe, est ici indispensable pour faire appel à la fonction **sqrt** qui renvoie la racine carrée de la variance.

Nous sommes maintenant presque outillés en Python pour pouvoir passer à quelques manipulations et études de nos fréquences.

Il nous reste à définir la fonction qui va permettre de déterminer le pourcentage des valeurs dans un intervalle donné.



Algorithme de calcul de moyenne et d'écart type

2. Définition de Proportion en Python

Nous complétons notre script avec la fonction **Proportion**, qui comptabilise le nombre d'éléments de la liste **l** passée en paramètre dans un intervalle de valeurs données par les paramètres **a** et **b** distincts.

Finalement la fonction renvoie la proportion sous forme de pourcentage arrondi à l'unité en divisant la variable **p** par la longueur de la liste. C'est la fonction **int** qui arrondit à l'unité.

```
def Proportion(l,a,b):
    p = 0
    for i in l:
        if a<=i and i<=b:
            p += 1
    return int(p/len(l)*100)
```

Fns... a A # Outils Exéc Script

3. Manipulation

La suite de l'exercice consiste donc à manipuler nos fonctions en console et observer les résultats obtenus. On commence par définir la variable **p** à 28 (notre probabilité de référence), **t** (la taille d'un échantillon) à 100 et **n** à 100.

On définit **l** une première liste de fréquences issues de la fonction Echantillon.

On stocke dans **s** l'écart type de notre liste.

On utilise la touche pour appeler plus rapidement chacune de nos fonctions et réaliser les manipulations demandées.

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de BERNOU2
>>> from BERNOU2 import *
>>> p = 28
>>> t = 100
>>> n = 100
>>> l = Echantillon(p,t,n)
>>> s = EcartType(l)
>>> |
```

Fns... a A # Outils Éditer Script

Lors de notre première manipulation, nous avons obtenu que 93% des fréquences issues de nos 100 échantillons de taille 100 sont dans l'intervalle $[p-2s; p+2s]$. Il semble donc que les 35% de jouets défectueux, obtenus lors du test d'un lot ne soit pas aberrant.

On relance plusieurs fois ces manipulations avec les mêmes paramètres de taille d'échantillons et nombre de lots. Les résultats se confirment à quelques variations près.

Il est possible de relancer ensuite les manipulations en augmentant le nombre d'échantillons jusqu'à 500.

Là encore les résultats nous amèneront à la même conclusion.

```
PYTHON SHELL
>>> Proportion(l,p-s,p+s)
75
>>> Proportion(l,p-2*s,p+2*s)
93
>>> Proportion(l,p-3*s,p+3*s)
100
>>> p+2*s
36.87988738667334
>>> p+3*s
41.31983108001001
>>> |
```

Fns... a A # Outils Éditer Script

```
PYTHON SHELL
>>> l=Echantillon(p,t,n)
>>> s=EcartType(l)
>>> Proportion(l,p-s,p+s)
66
>>> Proportion(l,p-2*s,p+2*s)
96
>>> Proportion(l,p-3*s,p+3*s)
99
>>> p+2*s
37.05427103636731
>>> |
```

Fns... a A # Outils Éditer Script

```
PYTHON SHELL
>>> l=Echantillon(p,t,n)
>>> s=EcartType(l)
>>> Proportion(l,p-s,p+s)
68
>>> Proportion(l,p-2*s,p+2*s)
95
>>> Proportion(l,p-3*s,p+3*s)
100
>>> p+2*s
37.10514140472295
>>> |
```

Fns... a A # Outils Éditer Script