

TD3 – Fonctions et comparaison de groupes

Python pour économistes – Université de Lille, L2 Économie

Question fil rouge. Comment écrire des fonctions Python pour comparer des groupes de pays à partir d'indicateurs OWID ?

Livrables. Un fichier `td3_nom_prenom.py`, un tableau texte, un graphique `comparaison_pib_td3.png` et une interprétation de cinq lignes.

Objectifs du TD

- Distinguer paramètre, argument, affichage et valeur de retour.
- Écrire et tester des fonctions simples sur des listes numériques.
- Comparer deux groupes avec moyenne, médiane, minimum et maximum.
- Réutiliser une fonction pour produire un graphique `matplotlib`.
- Corriger des erreurs liées aux noms de variables et à `return`.

Progression du TD

Les exercices 1 à 13 sont obligatoires. Les notions statistiques obligatoires sont : moyenne, médiane, minimum, maximum et proportion. Les quartiles et l'écart-type restent facultatifs. Le point Python central est la fonction : paramètre, appel, `return` et test.

Source des données

Les valeurs de PIB par habitant sont des extraits pédagogiques arrondis d'Our World in Data.

```
1 https://ourworldindata.org/grapher/gdp-per-capita-worldbank.csv
```

Logique 1 – Automatiser une statistique

PSEUDO-CODE

```
1 # Objectif : comparer deux listes de valeurs
2 # Entrees : liste_groupe_a, liste_groupe_b
3 # Définir une fonction moyenne(liste)
4 # Définir une fonction mediane(liste)
5 # Appliquer les fonctions aux deux groupes
6 # Comparer les resultats
7 # Sortie : tableau + phrase d'interpretation
```

A. Données et rappels

Exercice 1 – Créer deux groupes de pays

OBLIGATOIRE

Créer deux listes de PIB par habitant, exprimées en milliers de dollars internationaux et arrondies pour l'exercice.

```
1 groupe_europe = [46.0, 54.0, 34.0, 38.0, 22.0, 61.0]
```

```
2 groupe_monde = [46.0, 76.0, 19.0, 10.0, 7.0, 3.0]
```

Ajouter deux listes de noms pour documenter les valeurs :

```
1 pays_europe = ["France", "Germany", "Spain", "Italy", "Poland", "Sweden"]
2 pays_monde = ["France", "United States", "China", "Brazil", "India", "Ethiopia"]
```

Exercice 2 – Prédire quelques opérations sur les listes

OBLIGATOIRE

Sans écrire de fonction, prédire puis vérifier :

- `len(groupe_europe)` ;
- `sum(groupe_monde)` ;
- `sorted(groupe_monde)` ;
- `groupe_europe[0] > groupe_monde[0]`.

Ajouter une phrase qui explique pourquoi ces petits contrôles sont utiles avant de définir une fonction.

Exercice 3 – Rappel intuitif

OBLIGATOIRE

Ajouter dans le script trois commentaires :

- la moyenne résume un niveau général ;
- la médiane est la valeur centrale après tri ;
- min et max donnent l'étendue des valeurs observées.

B. Fonctions

Exercice 4 – Lire une fonction avant d'en écrire une

OBLIGATOIRE

Avant d'écrire des fonctions statistiques, lire ce petit exemple.

```
1 def ajouter_trois(x):
2     resultat = x + 3
3     return resultat
4
5 print(ajouter_trois(10))
```

Répondre en commentaires :

- quel est le paramètre de la fonction ?
- quel est l'argument utilisé lors de l'appel ?
- quelle ligne renvoie une valeur ?
- quelle ligne affiche une valeur ?

Exercice 5 – Modifier une fonction courte

OBLIGATOIRE

Compléter puis modifier cette fonction.

```
1 def convertir_en_dollars(valeur_milliers):  
2     return ...  
3  
4 print(convertir_en_dollars(46.0))
```

La fonction doit transformer 46.0 en 46000.0. Modifier ensuite son nom ou son paramètre, puis vérifier que l'appel utilise exactement le même nom. L'objectif est de repérer les erreurs de nom avant les fonctions statistiques.

Exercice 6 – Fonction moyenne**OBLIGATOIRE**

Écrire une fonction moyenne(valeurs) qui renvoie la moyenne d'une liste. La tester sur les deux groupes.

```
1 def moyenne(valeurs):  
2     return sum(valeurs) / len(valeurs)
```

La tester d'abord sur [10, 12, 14] pour vérifier que le résultat attendu est 12.0. Afficher ensuite les résultats des deux groupes avec une unité : milliers de dollars par habitant.

Exercice 7 – Fonction médiane guidée**OBLIGATOIRE**

Pour une liste de longueur paire, on peut prendre la moyenne des deux valeurs centrales. Compléter la fonction :

```
1 def mediane_paire(valeurs):  
2     valeurs_triees = sorted(valeurs)  
3     n = len(valeurs_triees)  
4     milieu_droit = ...  
5     milieu_gauche = ...  
6     return ...
```

Tester d'abord la fonction sur [8, 10, 12, 14] : la médiane attendue est 11.0. Tester ensuite la fonction sur les deux groupes.

Exercice 8 – Débogage de fonction**OBLIGATOIRE**

Les deux fonctions suivantes ressemblent à des solutions, mais elles posent problème. Identifier l'erreur, formuler une hypothèse, puis écrire la correction.

```
1 def moyenne_bug(valeurs):  
2     resultat = sum(valeur) / len(valeurs)  
3     return resultat  
4  
5 def moyenne_sans_return(valeurs):  
6     print(sum(valeurs) / len(valeurs))  
7  
8 m = moyenne_sans_return([10, 12, 14])  
9 print(m + 1)
```

Indice : le premier cas concerne un nom de variable ; le second permet de distinguer afficher une valeur et renvoyer une valeur.

Exercice 9 – Min, max et proportion

OBLIGATOIRE

Pour chaque groupe, calculer :

- la valeur minimale ;
- la valeur maximale ;
- la proportion de pays au-dessus de 30 milliers de dollars par habitant.

Conseil : utiliser un compteur et une boucle.

C. Comparer et interpréter

Logique 2 – Comparer deux groupes par un graphique

PSEUDO-CODE

```
1 # Objectif : comparer visuellement deux resultats
2 # Entree : deux listes de valeurs et une fonction moyenne
3 # 1. Calculer la moyenne du groupe Europe
4 # 2. Calculer la moyenne du groupe Monde
5 # 3. Placer les deux resultats dans une liste
6 # 4. Tracer une barre par groupe
7 # 5. Sauvegarder la figure
8 # Sortie : comparaison lisible
```

Exercice 10 – Graphique de comparaison

OBLIGATOIRE

Utiliser les fonctions écrites plus haut pour construire un graphique simple.

```
1 import matplotlib.pyplot as plt
2
3 groupes = ["Europe", "Monde"]
4 moyennes = [moyenne(groupe_europe), moyenne(groupe_monde)]
5
6 plt.figure()
7 plt.bar(groupees, moyennes)
8 plt.ylabel("Milliers de dollars par habitant")
9 plt.title("PIB par habitant moyen - extrait OWID")
10 plt.tight_layout()
11 plt.savefig("comparaison_pib_td3.png", dpi=300)
```

Modifier ensuite le code pour tracer les médianes à la place des moyennes.

Exercice 11 – Déboguer un graphique

OBLIGATOIRE

Voici deux fragments qui ressemblent au code précédent mais posent problème. Identifier l'erreur probable, puis proposer une correction.

```
1 # Fragment A
2 groupes = ["Europe", "Monde"]
3 moyennes = [moyenne(groupe_europe)]
4 plt.bar(groupees, moyennes)
5
6 # Fragment B
7 plt.bar(groupees, medianes)
```

Dans le fragment A, comparer la longueur des deux listes. Dans le fragment B, vérifier si la variable `medianes` a bien été créée avant son utilisation.

Exercice 12 – Tableau texte

OBLIGATOIRE

Afficher un petit tableau texte. Il doit contenir les colonnes `groupe`, `moyenne`, `mediane`, `min`, `max` et `part_au_dessus_30`. L'objectif est la lisibilité, pas la perfection typographique.

Exercice 13 – Question économique

OBLIGATOIRE

Ajouter cinq commentaires à la fin du script :

- quel groupe a la moyenne la plus élevée ?
- quel groupe a la médiane la plus élevée ?
- pourquoi moyenne et médiane peuvent-elles raconter deux choses différentes ?
- pourquoi le choix des pays influence-t-il fortement le résultat ?
- quelle limite voyez-vous à travailler sur seulement six pays par groupe ?

D. Entraînement facultatif

Pour aller plus loin 1 – Statistiques supplémentaires sans pression

Choisir au moins deux questions si le parcours obligatoire est terminé.

- Ajouter un pays dans chaque groupe et vérifier que les fonctions marchent encore.
- Calculer l'étendue : $\text{max} - \text{min}$.
- Écrire une fonction `proportion_au_dessus(valeurs, seuil)`.
- Tester le seuil de 50 au lieu de 30.
- Écrire une fonction `resume(valeurs)` qui affiche moyenne, médiane, min et max.
- Comparer la moyenne avant et après ajout d'une valeur très élevée.
- Calculer un premier quartile avec une méthode simple de votre choix.
- Calculer l'écart-type à partir d'une formule fournie par l'enseignant.
- Écrire une phrase expliquant pourquoi l'écart-type mesure la dispersion.
- Transformer les valeurs en dollars au lieu de milliers de dollars.

À retenir

- Une fonction sert à nommer une suite d'opérations que l'on veut réutiliser.
- Le paramètre est écrit dans la définition ; l'argument est la valeur fournie lors de l'appel.
- `return` renvoie une valeur réutilisable, alors que `print()` affiche seulement.
- Une fonction testée peut servir ensuite dans un tableau ou un graphique.
- Tester une fonction sur un cas simple permet de détecter une erreur avant de l'appliquer aux données.