

Premiers pas en Python - Visualisation & Pandas (CSV)

Enseignant : Etienne Dagorn
Mail : etienne.dagorn@univ-lille.fr



Objectifs de la séance

- Savoir lire/écrire des `.csv` avec `pandas`.
- Créer et manipuler un `DataFrame` (sélection, filtrage, tris, stats).
- Visualiser rapidement (histogrammes / camemberts) et sauvegarder ses résultats.

Astuce

Ouvrez votre dossier de TP dans VS Code. Assurez-vous que le **dossier courant** est bien celui qui contient vos fichiers `.py` et `.csv`. Sous Python :

```
import os
print(os.getcwd())    # dossier courant
```

Fonctions utiles (mémo)

Python (de base)

- `len(seq)` : taille d'une liste (`len(notes)`)
- `sum(seq)` : somme des éléments (`sum(notes)`)
- `min(seq)`, `max(seq)` : minimum / maximum
- `sorted(seq)` : copie triée (utile pour la médiane)

Matplotlib (pyplot)

- `plt.hist(data, bins=..., range=(a,b), alpha=..., edgecolor=...)` : histogramme
- `plt.xlabel("Texte")`, `plt.ylabel("Texte")` : labels des axes
- `plt.title("Texte")` : titre du graphique
- `plt.axvline(x, linestyle="-", linewidth=1)` : ligne verticale (ex. moyenne)
- `plt.legend()` : légende (si `label=` utilisé dans les tracés)
- `plt.tight_layout()` : ajuste les marges pour éviter les chevauchements
- `plt.savefig("nom.png", dpi=300, bbox_inches="tight", transparent=True)` : export
- `plt.show()` : affiche la figure

Fichiers / chemin

- `import os`
- `os.getcwd()` : dossier courant
- `os.path.exists("fichier.png")` : vérifier qu'un fichier existe
- `os.path.abspath("fichier.png")` : chemin absolu du fichier

Aide rapide

- `help(plt.hist)` / `help(plt.savefig)` : documentation en ligne dans Python
- VS Code : survol (hover), `Ctrl+Shift+Space` (signature), `F12/Alt+F12` (définition/aperçu)

1 Qu'est-ce qu'un fichier CSV ?

- **CSV** = *Comma-Separated Values* : un **fichier texte brut** où chaque ligne = un enregistrement (une *lí* ligne de tableau *z*), et les **valeurs** sont séparées par un **délimiteur** (souvent *,* ou *;* en France).

Exemple (contenu brut) :

```
prenom,note,assidue
Alice,14,0.82
Bob,10,0.63
Clara,16,0.91
```

- Correspond à un tableau :

prenom	note	assidue
Alice	14	0.82
Bob	10	0.63
Clara	16	0.91

- **Points clés**
 - **Séparateur** (sep) : *,* (anglo-saxon), *;* (souvent FR), parfois *\t* (TSV).
 - **Entête** (header) : la 1^{ère} ligne contient souvent les noms de colonnes.
 - **Encodage** (encoding) : *utf-8* recommandé ; parfois *latin-1*.
 - **Décimal** (decimal) : *.* en général ; *,* en FR (*attention* à ne pas confondre avec le séparateur).
 - **Guillemets** : si une valeur contient le séparateur, elle est souvent entre guillemets *"..."*.
 - **Ouverture** : Excel/LibreOffice peuvent changer séparateur/encodage → préférez *í* Importer *z* en précisant les options.

- **Avec pandas (read_csv)**

```
import pandas as pd

# Cas "FR" typique : séparateur ';' et virgule comme décimale
df = pd.read_csv("fichier.csv", sep=";", decimal=",", encoding="utf-8")
print(df.head(), df.dtypes)

# Forcer un type numérique (si nécessaire)
df["montant"] = pd.to_numeric(df["montant"], errors="coerce")
```

- **À retenir**

- Un CSV, c'est du *texte* : on peut l'ouvrir dans un éditeur et *voir* les séparateurs.
- Toujours vérifier `sep`, `decimal`, `encoding`.
- Contrôler rapidement `df.shape`, `df.head()`, `df.dtypes` après import.

2 Installer un package

- **Linux / macOS :**

```
python3 -m pip install --upgrade pip
python3 -m pip install matplotlib
```

- **Windows (PowerShell) :**

```
python -m pip install --upgrade pip
python -m pip install matplotlib
```

- **Vérifier** dans Python :

```
import matplotlib; print(matplotlib.__version__)
```

Dépannage rapide

- *ImportError / ModuleNotFoundError* : l'environnement Python actif n'est pas le bon. Sélectionner l'interpréteur en bas à droite (Python: **Select Interpreter**).
- *Toujours pas reconnu* : relancer VS Code après l'installation.
- *Plusieurs Pythons installés* : préférez `python -m pip ...` (ça installe dans l'interpréteur actif).

A - Première visualisation

Exercice - Histogramme des notes

Installe `matplotlib` si nécessaire puis trace un histogramme pour `[12, 14, 10, 8, 16, 15, 9]`.

- **Étapes**

1. Importer `matplotlib.pyplot`.
2. Définir la liste `notes`.
3. Appeler `plt.hist` avec un nombre de classes (`bins`) explicite.
4. Ajouter un titre et des labels d'axes.
5. Ajouter la moyenne en ligne pointillée.
6. Exporter la figure puis l'afficher.
7. Vérifier que le fichier `hist_notes.png` est bien créé.

- **Questions de vérification**

- Que se passe-t-il avec `bins=3` puis `bins=10` (regarde l'agrégat des barres) ?
- L'axe x affiche quelles valeurs (notes) ? l'axe y (effectifs par classe) ?
- Le fichier `hist_notes.png` est-il bien créé dans ton dossier de travail (`pwd`) ?
- Ajoute `alpha=0.8` dans `plt.hist(...)` pour voir l'effet de la transparence.
- Remplace `range=(0,20)` par `range=(8,16)` : que remarques-tu ?

A.2 - Comparer deux distributions

Exercice - Deux classes, un seul histogramme

On observe deux groupes de notes : `classe_A = [12, 14, 10, 8, 16, 15, 9]` et `classe_B = [11, 13, 12, 7, 17, 14, 10]`. Trace deux histogrammes sur le **même** graphique (transparence, légende) puis ajoute une ligne verticale à la moyenne de chaque classe.

1. Quelle classe a la moyenne la plus élevée ?
2. Les distributions se recouvrent-elles ?
3. Qu'est-ce que la moyenne ne montre pas ?

A.3 - Boîte à moustaches

Exercice - Boxplot

Représente `classe_A` et `classe_B` avec un **boxplot** (boîte à moustaches). Compare ce que tu vois avec l'histogramme.

1. Qu'indiquent les différentes parties du boxplot ?
2. Qu'indiquent la médiane et l'étendue interquartile ?
3. Vois-tu des valeurs atypiques (*outliers*) ?

A.4 - Statistiques descriptives par classe

Objectif : décrire et comparer deux échantillons en calculant à la main les indicateurs clés.

Données

```
A = [12, 14, 10, 8, 16, 15, 9]
B = [11, 13, 12, 7, 17, 14, 10]
```

Tâches (pour chaque classe)

1. Quelles **statistiques** calculer pour décrire l'échantillon ?
2. Calculer l'effectif n et la somme $\sum x_i$.
3. Calculer la moyenne $\bar{x} = \frac{\sum x_i}{n}$.
4. Calculer la médiane : trier, puis prendre l'élément du milieu (ou la moyenne des deux du milieu si n est pair).
5. Trouver le minimum et le maximum.
6. Calculer l'écart-type **population** $\sigma = \sqrt{\frac{1}{n} \sum (x_i - \bar{x})^2}$.

Aide minimale

```
def mediane(L):
    T = sorted(L)
    n = len(T); mid = n // 2
    return T[mid] if n % 2 == 1 else (T[mid-1] + T[mid]) / 2

def etendue(L):
    return max(L) - min(L)

def ecart_type_population(L):
    n = len(L)
    m = sum(L) / n
    var = sum((x - m) ** 2 for x in L) / n
    return var ** 0.5
```

Questions de compréhension

1. Bien que les deux classes aient la même moyenne, laquelle est la plus **dispersée** ? Que montrent vos min / max et écart-types ?
2. Si vous ajoutez une note extrême (ex. 0 dans A), que deviennent moyenne, médiane et écart-type ?