

Pandas

Fiche de synthèse

❗ **Important.** Pandas est une ouverture : cette bibliothèque d'analyse de données fait en une ligne les traitements de tables écrits avec des boucles dans le chapitre CSV.

① LIRE UNE TABLE

```
import pandas as pd

df = pd.read_csv("notes.csv")
print(df.shape)      # (6, 5)
print(df.head(3))    # 3 premières lignes
```

Définition 1 (DataFrame et Series)

- Un **DataFrame** est un tableau à deux dimensions : chaque colonne porte le nom donné par l'en-tête, chaque ligne a un numéro, son **index**, qui commence à 0.
- Une **Series** est une colonne : une suite de valeurs, chacune avec son index.

`shape` donne le nombre de lignes et de colonnes. Pandas reconnaît les nombres, contrairement au module `csv`. Une valeur manquante s'affiche `NaN`. Avec un point-virgule comme délimiteur, on écrit `pd.read_csv("notes.csv", sep=";")`.

② COLONNES ET LIGNES

```
df["maths"]          # une colonne
df["maths"].mean()   # sa moyenne
df[["prénom", "nsi"]] # deux colonnes
df.loc[2]            # la ligne d'index 2
df.loc[2, "nsi"]     # une seule case
```

Une colonne calcule sans boucle `mean`, `max`, `min`, `sum` et `count`, le nombre de valeurs présentes. Pour plusieurs colonnes, on donne la liste de leurs noms, d'où les doubles crochets.

③ SÉLECTIONNER ET TRIER

```
df[df["maths"] >= 12]
df[(df["classe"] == "1A") & (df["nsi"] >= 15)]
df.sort_values(by="nsi", ascending=False)
```

`df["maths"] >= 12` compare chaque note à 12 et donne une série de booléens ; seules les lignes associées à `True` sont gardées. Pour combiner deux conditions, on écrit `&` pour « et »,

`|` pour « ou », et on met chaque condition entre parenthèses. `sort_values` renvoie un nouveau `DataFrame` et laisse `df` intact.

④ VALEURS MANQUANTES

`df.isna().sum()` compte les cases vides de chaque colonne. `df.fillna(v)` les remplace par `v` ; `df.dropna()` supprime les lignes incomplètes. Le choix dépend de la situation : remplacer une note absente par 0 serait injuste.

⑤ CALCULER

```
matieres = ["maths", "nsi", "physique"]
df["moyenne"] = df[matieres].mean(axis=1)
df.groupby("classe")[matieres].mean()
df.describe()
```

- Une affectation crée une colonne, calculée pour toutes les lignes à la fois ; `axis=1` calcule la moyenne de chaque ligne, en ignorant les valeurs manquantes.
- `groupby("classe")` regroupe les lignes de même classe et calcule la moyenne de chaque groupe.
- `describe` donne, pour chaque colonne numérique, l'effectif, la moyenne, l'écart type, le minimum, les quartiles et le maximum.

⑥ TRACER UN GRAPHIQUE

```
import matplotlib.pyplot as plt

df.plot(kind="bar", x="prénom", y=matieres)
plt.show()
```

Le paramètre `kind` choisit le graphique : `"bar"` pour des barres, `"line"` pour une courbe, `"scatter"` pour un nuage de points.

À retenir.

- `pd.read_csv` charge une table dans un `DataFrame`.
- `df["col"]` donne une colonne, `df.loc[i]` une ligne.
- `df[condition]` sélectionne, `sort_values` trie, `groupby` regroupe.