📊 Cours : Analyse de Données avec Python – Pandas et Matplotlib

🎯 Objectif :

Apprendre à manipuler, analyser et visualiser des données efficacement en utilisant les bibliothèques **Pandas** (pour le traitement) et **Matplotlib** (pour la visualisation).

---

1. 📥 Introduction à l'analyse de données

L'analyse de données est un processus qui consiste à :

- Importer des données à partir de fichiers ou bases de données.
- Nettoyer et organiser les données.
- Explorer les statistiques descriptives.
- Visualiser les tendances ou anomalies.
- Extraire des conclusions exploitables.

Python est un outil puissant pour cette tâche grâce à des bibliothèques comme **Pandas**, **Matplotlib** et **Seaborn**.

---

2. 💼 Chargement de données avec Pandas

```
import pandas as pd

# Lire un fichier CSV
df = pd.read_csv("ventes.csv")

# Aperçu des données
print(df.head())

# Informations générales
print(df.info())
```

✅ `pd.read_csv()` est la méthode la plus utilisée pour charger des jeux de données tabulaires.

---

3. 🔍 Nettoyage et préparation des données

a. Supprimer les valeurs manquantes

```
df.dropna(inplace=True)  # Supprimer les lignes avec des NaN
```

b. Remplir les valeurs manquantes

```
df["prix"].fillna(df["prix"].mean(), inplace=True)
```

c. Renommer les colonnes

```
df.rename(columns={"NomProduit": "Produit"}, inplace=True)
```

d. Filtrer les lignes

```
df_filtré = df[df["Ventes"] > 1000]
```

---

4. □ Analyse statistique de base

```
print(df.describe())  # Statistiques générales (moyenne, écart-type, etc.)
print(df["Ventes"].mean())  # Moyenne des ventes
print(df["Produit"].value_counts())  # Fréquence des produits
```

---

5. □ Agrégation des données

a. Grouper les données par colonnes

```
df_groupe = df.groupby("Catégorie")["Ventes"].sum()
print(df_groupe)
```

b. Trier les données

```
df_sorted = df.sort_values("Ventes", ascending=False)
```

---

6. 📈 Visualisation des données avec Matplotlib

```
import matplotlib.pyplot as plt
```

a. Histogramme des ventes

```
plt.hist(df["Ventes"], bins=10)
plt.title("Distribution des ventes")
plt.xlabel("Montant")
plt.ylabel("Fréquence")
plt.show()
```

b. Courbe de tendance

```
plt.plot(df["Date"], df["Ventes"])
plt.title("Évolution des ventes")
plt.xlabel("Date")
plt.ylabel("Ventes")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

c. Diagramme en barres

```
df["Produit"].value_counts().plot(kind="bar")
plt.title("Ventes par produit")
plt.show()
```

---

7. □ Exemple complet d'analyse

```
import pandas as pd
import matplotlib.pyplot as plt

# Chargement
df = pd.read_csv("data.csv")

# Nettoyage
df.dropna(inplace=True)

# Analyse
top_produits = df["Produit"].value_counts().head(5)

# Visualisation
top_produits.plot(kind="bar", color="teal")
plt.title("Top 5 produits vendus")
plt.xlabel("Produit")
plt.ylabel("Nombre de ventes")
plt.show()
```

---

8. 🎁 Autres outils utiles

Bibliothèque	Utilité principale
`seaborn`	Visualisations avancées
`numpy`	Calcul numérique
`plotly`	Graphiques interactifs
`openpyxl`	Lire/écrire des fichiers Excel

✅ Résumé

Tâche	Outil utilisé
Chargement des données	`pandas.read_csv()`
Nettoyage	`dropna()`, `fillna()`
Statistiques descriptives	`describe()`, `mean()`
Agrégation	`groupby()`, `sum()`
Visualisation	`matplotlib.pyplot.plot()`

🔁 Exercice proposé :

Créez un script Python qui :

1. Charge un fichier CSV contenant des ventes.
2. Affiche la moyenne des ventes par produit.
3. Génère un histogramme des ventes.
4. Sauvegarde le graphique dans un fichier `graphique.png`.