Introduction : Dans ce manuel d'introduction à l'analyse de données avec Python et Pandas, nous continuerons à travailler avec le même fichier de données que celui que nous avons précédemment utilisé avec Excel. L'objectif est de montrer comment Python, avec l'aide de la bibliothèque Pandas, peut être un outil puissant pour analyser, manipuler et visualiser des données structurées.

On va utiliser le fichier CSV portant le nom ***students.csv*** contenant les données à traiter à partir de ce lien :
https://github.com/HoussemLahiani/StructeredData/blob/main/students.csv

Lecture d'un fichier CSV avec Pandas

```
import pandas as pd
# L'URL du fichier CSV en ligne
url = 'https://github.com/HoussemLahiani/StructeredData/raw/main/students.csv'
# Charger le fichier CSV depuis l'URL dans un DataFrame
df = pd.read_csv(url)
# Afficher les premières lignes du DataFrame
print(df.head())
```

```
      Nom   Age  MatierePreferee   Note 1   Note 2   Note 3
0   Alice    16    Mathematiques       85       92       78
1     Bob    17          Histoire       88       76       90
2  Claire    16          Sciences       92       95       89
3   David    17          Francais       78       84       86
4    Emma    16           Anglais       90       91       88
```

Filtrage de données avec Pandas

```
import pandas as pd
# Charger le fichier CSV dans un DataFrame
df = pd.read_csv('https://github.com/HoussemLahiani/StructeredData/raw/main/students.csv')
# Filtrer les lignes selon un critère (par exemple, notes supérieures à 90)
result = df[df[' Note 1'] > 90]
print(result)
```

```
        Nom   Age  MatierePreferee   Note 1   Note 2   Note 3
2    Claire    16          Sciences       92       95       89
5    Fabien    18          Physique       92       94       91
8  Isabelle    17          Biologie       91       88       90
```

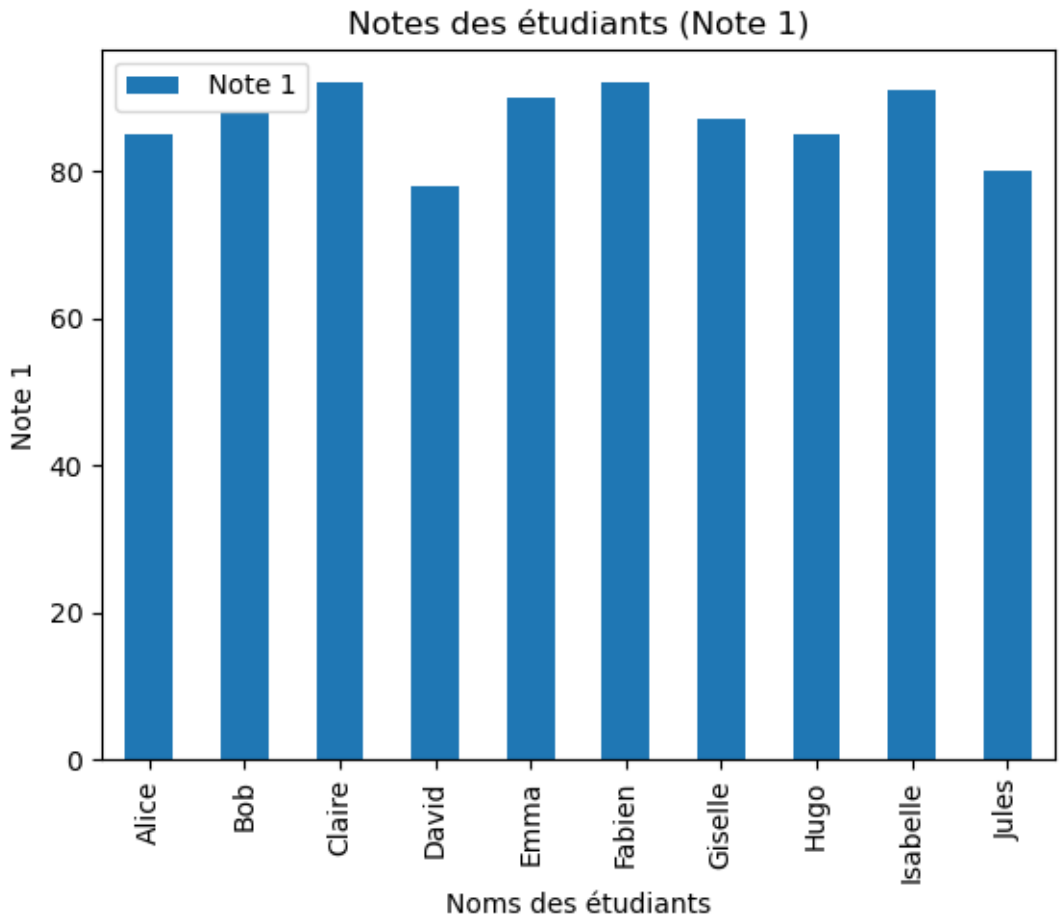
Tri des données avec Pandas

```
import pandas as pd
# Charger le fichier CSV dans un DataFrame
df = pd.read_csv('https://github.com/HoussemLahiani/StructeredData/raw/main/students.csv')
# Trier les données par colonne (par exemple, par nom)
sorted_df = df.sort_values(by='Nom')
print(sorted_df)
```

	Nom	Age	MatierePreferee	Note 1	Note 2	Note 3
0	Alice	16	Mathematiques	85	92	78
1	Bob	17	Histoire	88	76	90
2	Claire	16	Sciences	92	95	89
3	David	17	Francais	78	84	86
4	Emma	16	Anglais	90	91	88
5	Fabien	18	Physique	92	94	91
6	Giselle	17	Chimie	87	89	83
7	Hugo	16	Geographie	85	79	88
8	Isabelle	17	Biologie	91	88	90
9	Jules	18	Arts	80	82	78

Création d'un graphique avec Pandas (pour la visualisation)

```
import pandas as pd
import matplotlib.pyplot as plt
# Charger le fichier CSV dans un DataFrame
df = pd.read_csv('https://github.com/HoussemLahiani/StructeredData/raw/main/students.csv')
# Créer un graphique en barres
df.plot(x='Nom', y=' Note 1', kind='bar')
plt.xlabel('Noms des étudiants')
plt.ylabel('Note 1')
plt.title('Notes des étudiants (Note 1)')
plt.show()
```



Accéder à une valeur spécifique dans un DataFrame pandas depuis un fichier CSV

Assurez-vous de remplacer 'Nom' par le nom de la colonne que vous souhaitez accéder ('Âge', 'Matière préférée', 'Note 1', 'Note 2', 'Note 3' dans votre cas).Notez que les indices commencent généralement à 0 en Python, donc df.loc[0] vous donnera la première ligne, df.loc[1] la deuxième ligne, et ainsi de suite.

```
import pandas as pd
# Charger le fichier CSV dans un DataFrame
df = pd.read_csv('https://github.com/HoussemLahiani/StructeredData/raw/main/students.csv')
# Accéder à la valeur dans la première ligne de la colonne "Nom"
info = df.loc[0, ' Age']
print(info)
#Il est possible de récupérer toutes les lignes d'une colonne, il suffit de remplacer la partie
#index_ligne" de "loc" par ":"
```

```
16
```

Le code qui va suivre extraira les valeurs des colonnes 'nom' et 'Age' pour les lignes 0 et 1 de votre DataFrame 'iden'. Vous obtiendrez un sous-ensemble de données contenant les noms et les âges des deux premières lignes de votre DataFrame.

```
import pandas
iden=pandas.read_csv("students.csv")
info = iden.loc[[0, 1], ['Nom',' Age']]
print(info)
```

```
     Nom   Age
0  Alice    16
1    Bob    17
```

Calculer la moyenne des notes pour chaque étudiant

```
import pandas as pd
# Charger le fichier CSV dans un DataFrame
df = pd.read_csv('https://github.com/HoussemLahiani/StructeredData/raw/main/students.csv')
# Calculer la moyenne des notes pour chaque étudiant
df['Moyenne'] = df[[' Note 1', ' Note 2', ' Note 3']].mean(axis=1)
# Afficher la colonne 'Moyenne'
print(df['Moyenne'])
# Ajouter le nom de chaque étudiant devant sa moyenne
df['Nom et Moyenne'] = df['Nom'] + ' : ' + df['Moyenne'].astype(str)
# Afficher la nouvelle colonne 'Nom et Moyenne'
print(df['Nom et Moyenne'])
```

```
0    85.000000
1    84.666667
2    92.000000
3    82.666667
4    89.666667
5    92.333333
6    86.333333
7    84.000000
8    89.666667
9    80.000000
Name: Moyenne, dtype: float64
0                    Alice : 85.0
1         Bob : 84.66666666666667
```

```
2                   Claire : 92.0
3       David : 82.66666666666667
4        Emma : 89.66666666666667
5      Fabien : 92.33333333333333
6     Giselle : 86.33333333333333
7                     Hugo : 84.0
8    Isabelle : 89.66666666666667
9                    Jules : 80.0
Name: Nom et Moyenne, dtype: object
```

La bibliothèque pandas simplifie considérablement la manipulation et l'analyse de données en Python, notamment lorsqu'il s'agit de lire et de traiter des fichiers CSV. Cependant, si vous souhaitez éviter d'utiliser pandas et que vous préférez travailler directement avec Python de manière plus "brute", vous pouvez utiliser la fonction open pour lire le fichier CSV et un boucle for pour parcourir les lignes du fichier. Voici comment vous pourriez le faire :

```python
# Ouvrir le fichier CSV en mode lecture
with open('students.csv', 'r') as file:
    # Lire les lignes du fichier
    lines = file.readlines()
# Supprimer les caractères de saut de ligne des lignes
lines = [line.strip() for line in lines]
# Diviser les lignes en listes en utilisant la virgule comme séparateur
data = [line.split(',') for line in lines]
# Les noms des colonnes sont dans la première ligne
columns = data[0]

# Les données commencent à la deuxième ligne
data = data[1:]
# Créer un dictionnaire pour stocker les données
data_dict = {column: [] for column in columns}
# Remplir le dictionnaire avec les données
for row in data:
    for i, value in enumerate(row):
        data_dict[columns[i]].append(value)
# Maintenant, vous avez un dictionnaire contenant les données
print(data_dict)
```

```
{'Nom': ['Alice', 'Bob', 'Claire', 'David', 'Emma', 'Fabien', 'Giselle', 'H
ugo', 'Isabelle', 'Jules'], ' Age': [' 16', ' 17', ' 16', ' 17', ' 16', ' 1
8', ' 17', ' 16', ' 17', ' 18'], ' MatierePreferee': [' Mathematiques', ' H
istoire', ' Sciences', ' Francais', ' Anglais', ' Physique', ' Chimie', ' G
eographie', ' Biologie', ' Arts'], ' Note 1': [' 85', ' 88', ' 92', ' 78',
' 90', ' 92', ' 87', ' 85', ' 91', ' 80'], ' Note 2': [' 92', ' 76', ' 95',
' 84', ' 91', ' 94', ' 89', ' 79', ' 88', ' 82'], ' Note 3': [' 78', ' 90',
' 89', ' 86', ' 88', ' 91', ' 83', ' 88', ' 90', ' 78']}
```