Introduction à Pandas

Par

OUHAB Abdelkrim

Introduction

- Pandas (qui signifie Python Data Analysis) est une bibliothèque conçue pour la manipulation et **l'analyse des données** .
- Elle s'intègre parfaitement avec d'autres bibliothèques Python telles que NumPy, Matplotlib et scikit-learn.

À quoi sert Pandas ?

- Lecture et écriture de données à partir de différents formats de fichiers tels que CSV, Excel et les bases de données SQL.
- Nettoyage et préparation des données (gestion des valeurs manquantes, filtrage, suppression des doublons).
- Fusion, jointure des ensembles de données.
- Analyse des données.
- Visualisation des données.

Pandas Basic Operations

Pandas simplifies data handling by enabling efficient preprocessing, cleaning, transformation, and visualization.

Data Preprocessing

Handle missing values and clean raw data for analysis.
Before (Table with Missing Values

Before (Table with Missing Values)

Name	Age	Salary_Amount
Aryan	25	NaN
Harsh	NaN	5000
Kunal	30	7000

After (Missing Values Filled)

Name	Age	Salary
Aryan	25	0
Harsh	28	5000
Kunal	30	7000

Operation Applied: Filled missing values with 0 / average using fillna()

Data Cleaning

Remove duplicates and fix column inconsistencies.
Before (Duplicate Rows & Bad Column Names):

Before

Name	Age	Salary_Amount
Aryan	25	5000
Harsh	30	7000
Aryan	25	5000

After

Name	Age	Salary
Aryan	25	5000
Harsh	30	7000

Operation Applied: Removed duplicates using drop_duplicates(), renamed columns using rename()

Data Transformation

Filter and sort data for better insights.

Before (unfiltered Data)

Product	Category	Price
Laptop	Electronics	800
Hoodie	Clothing	20
Smartphone	Electronics	500
Jeans	Clothing	40

After (Filtered & Sorted Data)

Product	Category	Price
Smartphone	Electronics	500
Laptop	Electronics	800

Operation Applied: Filtered Electronics using query(), sorted by price using sort_values()

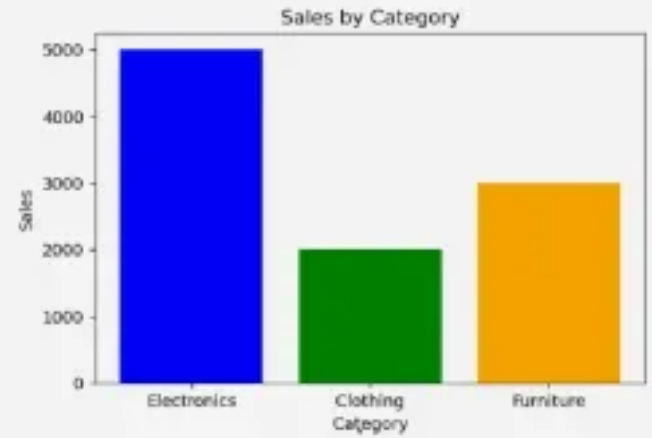
Data Visualization

Represent data graphically for insights

Before (Raw Data Table)

Category	Price
Electronics	5000
Clothing	2000
Furniture	3000

After (Bar Chart Visual)



Operation Applied: Plotted sales data using matplotlib

Installation Pandas

- Pour installer Pandas:

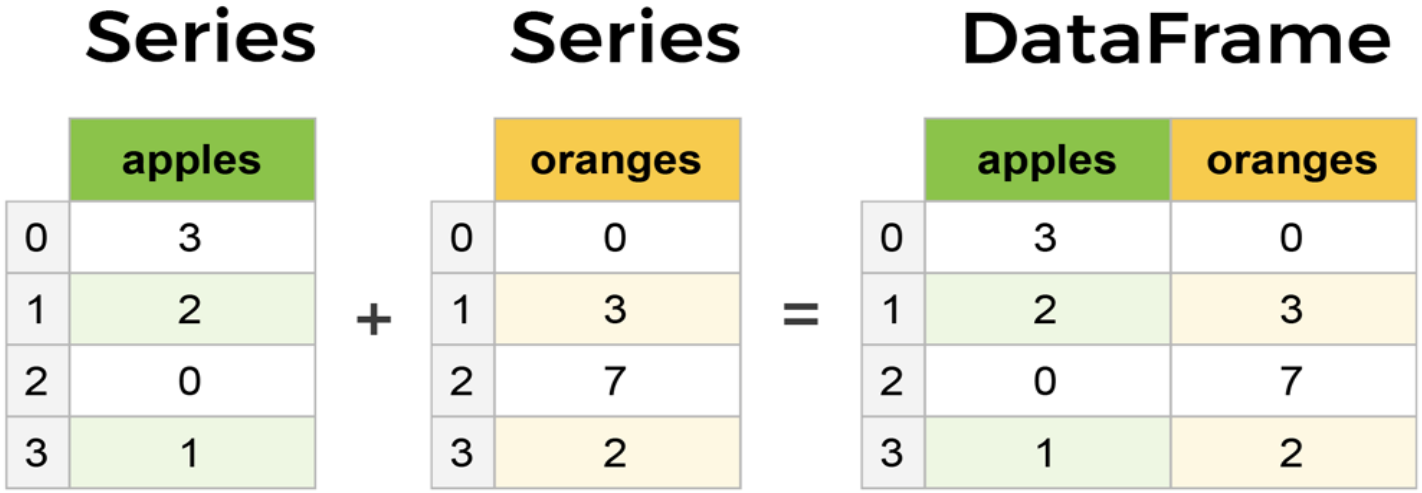
  pip installer pandas.

- Une fois Pandas installé sur le système, il faut importer la bibliothèque:

```
import pandas as pd
```

Structures de données dans Pandas

- **Séries Pandas:** tableau unidimensionnel pouvant contenir des données de tout type (nombre, chaîne de caractères, objets Python, etc.)
- **DataFrame Pandas:** structure de données bidimensionnelle (lignes et colonnes)



Création DataFrame

pandas.DataFrame(data, index , colonnes , dtype , copy)

- **data :** ndarray, series, listes, dictionnaires, autre DataFrame, etc.
- **index :** étiquettes de lignes, par défaut est np.arange(n) si aucun index n’est passé.
- **colonnes :** étiquettes de colonnes, par défaut est np.arrange(n).
- **dtype :** type de données de chaque colonne.
- **copy :** permet la copie des données.

Création DataFrame à partir d'une liste

```
import pandas as pd
liste = ['Omar', 'Ahmed', 'Fatiha']

df1 = pd.DataFrame(liste)
print("--------- DF1 ------------")
print(df1)

df2 = pd.DataFrame(liste, columns = ["nom"])
print("--------- DF2 ------------")
print(df2)
```

Output

```
--------- DF1 ------------
    0
0 Omar
1 Ahmed
2 Fatiha
--------- DF2 ------------
   nom
0 Omar
1 Ahmed
2 Fatiha
```

Création DataFrame à partir d'une liste

```
import pandas as pd
liste = [['Omar',25], ['Ahmed',45], ['Fatiha',30]]
df = pd.DataFrame(liste, columns = ["nom","age"])
print(df)
```

Output

```
    nom  age
0 Omar    25
1 Ahmed   45
2 Fatiha 30
```

Création DataFrame à partir d'un dictionnaire

```
import pandas as pd
data = {'nom':['Omar', 'Ahmed','Fatiha'], 'age':[25,45,30]}
df = pd.DataFrame(data)
print(df)
```

Output

```
    nom  age
0 Omar   25
1 Ahmed  45
2 Fatiha 30
```

Création DataFrame à partir de tableaux NumPy

```
import pandas as pd
import numpy as np
data = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
df = pd.DataFrame(data, columns= ['A', 'B', 'C'])
print(df)
```

Output

```
   A B C
0 1 2 3
1 4 5 6
2 7 8 9
```

Lecture de fichiers CSV

pd.read_csv(filepath, sep=' ,' , header='infer', index_col=None, usecols=None, engine=None, skiprows=None, nrows=None)

- **filepath:** Emplacement ou URL du fichier CSV
- **sep**: séparateur utilisé dans le fichier CSV, la valeur par défaut ', '
- **header**: numéros de ligne servant de noms de colonnes par défaut 0
- **usecols**: les colonnes sélectionnées du fichier CSV
- **nrows**: Nombre de lignes à afficher
- **index_col**: colonne utilisée comme index
- **skiprows**: les lignes à ignorer

Lecture de fichiers CSV

```
import pandas as pd
df = pd.read_csv("d:/data/unemployment.csv")
df.head()
```

Output

```
                country   nemployment_rate
0  Marshall Islands                   36.0
1      South Africa                   33.6
2          Kiribati                   30.6
3            Kosovo                   30.5
4    American Samoa                   29.8
```

Lecture de fichiers Excel

```
import pandas as pd
df = pd.read_excel("d:/data/unemployment.xlsx")
df.head()
```

Output

```
              country   nemployment_rate
0  Marshall Islands                 36.0
1      South Africa                 33.6
2          Kiribati                 30.6
3            Kosovo                 30.5
4    American Samoa                 29.8
```

Enregistrer un Dataframe comme fichier CSV

```
import pandas as pd
liste = [['Omar',25], ['Ahmed',45], ['Fatiha',30]]
df = pd.DataFrame(liste, columns = ["nom","age"])
df.to_csv('d:/data/etudiant.csv', index=False)
```

Affichage des données d'un dataframe

- La fonction head (tail) affiche par défaut les cinq premières (dernières) lignes d'un DataFrame, mais on peut également lui passer un nombre.

```
import pandas as pd
liste = [['Omar',25], ['Ahmed',45], ['Fatiha',30], ['Ali',44]]
df = pd.DataFrame(liste, columns = ["nom","age"])
print(df.head(2))
print("-----------------------")
print(df.tail(2))
```

Output

```
     nom  age
0   Omar   25
1  Ahmed   45
-----------------------
     nom   age
2 Fatiha    30
3    Ali    44
```

Concaténation de Dataframes

pandas.concat(objs, axis=0, …)

- Objs: une liste ou un tuple d'objets pandas (DataFrames ou Series) à concaténer
- Axis: axe de concaténation : 0 pour les lignes (par défaut), 1 pour les colonnes

```
import pandas as pd
liste1 = [['Omar',25], ['Ahmed',45], ['Fatiha',30]]
df1 = pd.DataFrame(liste1, columns = ["nom","age"])
liste2 = [['Ali',55], ['Fatima',35], ['Samir',30]]
df2 = pd.DataFrame(liste2, columns = ["nom","age"])
df = pd.concat([df1, df2])
```