



Département: Mathématiques & Informatique

Séance 10:

Analyse de données en Python

Module : pandas

Licence : Physique Chimie

Pr: Youssef Ouassit

Sommaire

01

IMPORTATION DES DONNEES

02

EXPLORATION DES DONNEES

03

NETTOYAGE DES DONNEES

04

FILTRAGE DES DONNEES

Introduction

Origine

Le nom « Pandas » est en fait la contraction du terme « **Panel Datas** » désignant les ensembles de données incluant des observations sur de multiples périodes temporelles.

Définition

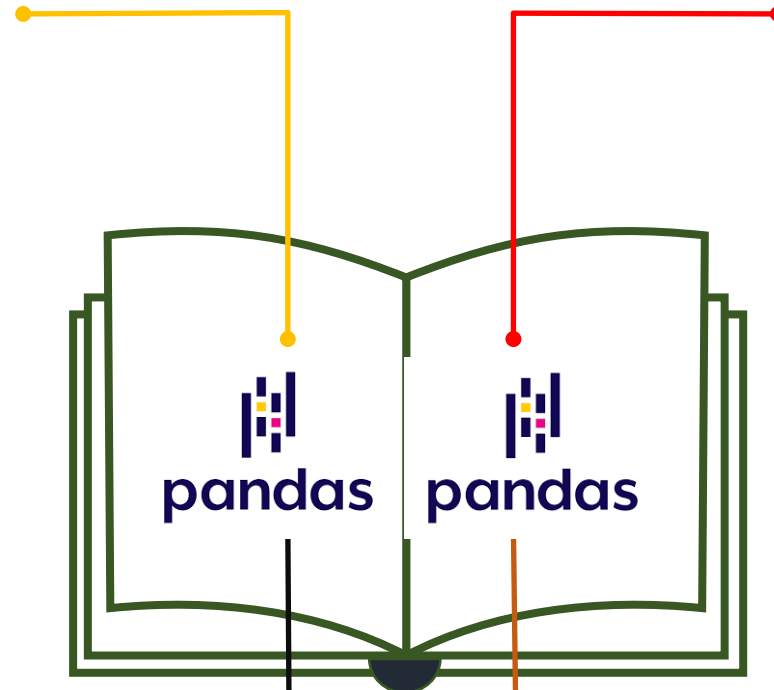
Bibliothèque écrite pour le langage python permettant la manipulation et l'analyse des données

Manipulation

Pandas manipule les données sous forme de data frame .

Type

Qualitatif et quantitatif



Pandas est une bibliothèque Python qui permet de manipuler facilement des structures de données, principalement sous forme de tables, telles que des fichiers CSV, Excel, ou des bases de données. Pandas fournit deux structures principales :

- **Series** : un tableau unidimensionnel étiqueté.
- **DataFrame** : une table bidimensionnelle (tableau avec des lignes et des colonnes).

DataFrame

Diagram illustrating a DataFrame structure with labels and indices.

Column Label/ Header (0, 1, 2, 3, 4)

Index Label (0, 1, 2, 3, 4)

Column Index (Name, Age, Marks, Grade, Hobby)

Row (0, 1, 2, 3, 4)

Row Index (0, 1, 2, 3, 4)

Column (0, 1, 2, 3, 4)

Element/ Value/ Entry (e.g., 88.78)

	0	1	2	3	4
	Name	Age	Marks	Grade	Hobby
0	Joe	20	85.10	A	Swimming
1	Nat	21	77.80	B	Reading
2	Harry	19	91.54	A	Music
3	Sam	20	88.78	A	Painting
4	Monica	22	60.55	B	Dancing

Caractéristiques:

- Structure bidimensionnelle
- Colonnes étiquetées
- Mutable
- Types de données hétérogènes
- ...

Importation des modules:

```
import numpy as np
```

```
import pandas as pd
```

1-FONCTIONS D'IMPORTATION



Le format csv



Le format CSV (Comma Separated Values) est l'un des formats de fichiers les plus courants pour le stockage et l'échange de données tabulaires (lignes et colonnes). Les fichiers CSV sont des fichiers texte dans lesquels les données sont séparées par des virgules (ou d'autres délimiteurs comme des tabulations ou des points-virgules), et chaque ligne représente une ligne de données.

Le format csv



Nom	Age	Ville	Téléphone	Salaire
Youssef	30	Rabat	0661111111	5000
Hanane	25	Casablanca	0661222222	6000
Rachid	25	Tanger	0661333333	7000
Dounia	27	Marrakech	0661444444	10000
Rachida	27	Marrakech	0661444444	9000
Lina	47	Essaouira	0661444444	4000
Ahmed	27	Fès		20000
Nabil	27	Marrakech	0661888888	8500
Nada	17	Casablanca	0661999999	5000

pd.read_csv()

Lire un fichier d'extension CSV

```
import pandas as pd  
df = pd.read_csv('etudiants.csv')
```

	Nom	Age	Ville	Téléphone	Salaire
0	Youssef	30	Rabat	0661-111111	5000
1	Hanane	25	Casablanca	0661-222222	6000
2	Rachid	25	Tanger	0661-333333	7000
3	Dounia	47	Marrakech	0661-444444	10000
4	Rachida	27	Marrakech	0661-444444	9000
5	Lina	57	Essaouira	0661-444444	4000
6	Ahmed	27	Fès	NaN	20000
7	Nabil	30	Marrakech	0661-888888	8500
8	Nada	17	Casablanca	0661-999999	5000

pd.read_table()

lire les données sous formes de fichier texte comme TSV

```
1 #Since tsv is a default seperator.  
2 df = pd.read_table('chiporders.tsv')  
3 #head() is used to display 1st 5 row in table  
4 df.head()
```

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	\$2.39
1	1	1	Izze	[Clementine]	\$3.39
2	1	1	Nantucket Nectar	[Apple]	\$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	\$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	\$16.98



pandas

pd.read_excel()

lire un fichier d'extension
xlsx

```
data=pd.read_excel('datastory.xlsx')
```

<ID>		MOT	POLISEMIQUE	SENS	PHRASE	DOMAINE	DOMAINE en clair
0	1	abaisser 01	9.0	baissér	On a~ le rideau de fer, le store. Le rideau du...	LOC	locatif, lieu

Récapitulatifs



```
import pandas as pd
```

- **pd.read_csv()** = lire un fichier CSV
- **pd.read_table()** = lire les données sous formes de fichier texte comme TSV
- **pd.read_excel()** = lire les données sous formes de fichier Excel;
- **pd.read_sql(query,connection_objet)** = lire les données du table SQL/base de données
- **pd.read_json(json_string)** = lire les données d un fichier json
- **pd.read_DataFrame(dict)** = depuis un dictionnaire, les données sont sous formes de ligne et de colonnes

2-FONCTIONS D'EXPLORATION



df.info()

➤ Informations sur les colonnes

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Nom         9 non-null     object
1   Age         9 non-null     int64
2   Ville       9 non-null     object
3   Téléphone   8 non-null     object
4   Salaire     9 non-null     int64
dtypes: int64(2), object(3)
memory usage: 492.0+ bytes
```


df.describe()

Donne une vue statistique d'ensemble

	Age	Salaire
count	9.000000	9.000000
mean	28.000000	8277.777778
std	7.968689	4841.946349
min	17.000000	4000.000000
25%	25.000000	5000.000000
50%	27.000000	7000.000000
75%	27.000000	9000.000000
max	47.000000	20000.000000

df.head(2)

Donne les n premières lignes du dataframe

	Nom	Age	Ville	Téléphone	Salaire
0	Youssef	30	Rabat	0661-111111	5000
1	Hanane	25	Casablanca	0661-222222	6000

```
df.tail( 3 )
```

Donne les n dernières lignes
du dataframe

	Nom	Age	Ville	Téléphone	Salaire
6	Ahmed	27	Fès	NaN	20000
7	Nabil	30	Marrakech	0661-888888	8500
8	Nada	17	Casablanca	0661-999999	5000

df.mean()

Donne la moyenne

```
PassengerId    446.000000
Survived        0.383838
Pclass          2.308642
Age            29.699118
SibSp           0.523008
Parch           0.381594
Fare           32.204208
dtype: float64
```

df.min() ,df.max()

Donne la valeur minimale et la valeur maximale des variables

```
titanic.min()
```

```
PassengerId      1
Survived          0
Pclass            1
Name      Abbing, Mr. Anthony
Sex            female
Age            0.42
SibSp            0
Parch            0
Ticket      110152
Fare            0.0
dtype: object
```

```
titanic.max()
```

```
PassengerId      891
Survived          1
Pclass            3
Name      van Melkebeke, Mr. Philemon
Sex            male
Age            80.0
SibSp            8
Parch            6
Ticket      WE/P 5735
Fare      512.3292
dtype: object
```



pandas

df.count()

Affiche le nombre de valeurs non nulles

```
Nom          9
Age          9
Ville        9
Téléphone    8
Salaire      9
dtype: int64
```

df[col_name]

Sélectionner une colonne particulière

```
df[ 'Nom' ]
```

```
0    Youssef
1    Hanane
2    Rachid
3    Dounia
4    Rachida
5     Lina
6    Ahmed
7    Nabil
8     Nada
Name: Nom, dtype: object
```

```
df[ [col1,col2, ...]]
```

Sélectionner plusieurs
colonnes particulière

```
df[ [ 'Nom', 'Salaire' ] ]
```

	Nom	Salaire
0	Youssef	5000
1	Hanane	6000
2	Rachid	7000
3	Dounia	10000
4	Rachida	9000
5	Lina	4000
6	Ahmed	20000
7	Nabil	8500
8	Nada	5000

RECAPITULATIFS

- **df.shape** : donne le nombre de ligne et de colonne du dataframe
- **df.info()** : donne le type de donnée
- **df.describe()** : donne une vue statistique d'ensemble
- **df.head(n)** : afficher les n premiers éléments du dataframe
- **df.tail()** : affiche les n derniers éléments du dataframe
- **df.mean()** : donne la moyenne de chaque colonnes
- **df.std()** : retourne une division standard de chaque colonnes
- **df.min()** : valeurs minimale de chaque colonnes
- **df.max()** : valeurs maximale de chaque colonnes
- **df.count()** : affiche le nombre de valeurs non nulles
- **df[col_name]** : affiche une colonne particulière



3-FONCTIONS DE NETTOYAGE



```
df=df.drop(columns=nom_col)
```

Supprimer une colonne particulière

```
df = df.drop(columns=['Ticket', 'PassengerId'])
```

Ou en place :

```
df.drop(columns=['Ticket', 'PassengerId'], inplace=True)
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S

df.isnull()

Affiche les valeurs
manquantes

	Nom	Age	Ville	Téléphone	Salaire
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	True	False
7	False	False	False	False	False
8	False	False	False	False	False

df.dropna(condition)

Supprimer les valeurs manquantes

df.dropna()

	Nom	Age	Ville	Téléphone	Salaire
0	Youssef	30	Rabat	0661-111111	5000
1	Hanane	25	Casablanca	0661-222222	6000
2	Rachid	25	Tanger	0661-333333	7000
3	Dounia	47	Marrakech	0661-444444	10000
4	Rachida	27	Marrakech	0661-444444	9000
5	Lina	57	Essaouira	0661-444444	4000
7	Nabil	30	Marrakech	0661-888888	8500
8	Nada	17	Casablanca	0661-999999	5000

df.dropna(condition)

Supprimer les valeurs manquantes

df.dropna(how='all')

	Nom	Age	Ville	Téléphone	Salaire
0	Youssef	30	Rabat	0661-111111	5000
1	Hanane	25	Casablanca	0661-222222	6000
2	Rachid	25	Tanger	0661-333333	7000
3	Dounia	47	Marrakech	0661-444444	10000
4	Rachida	27	Marrakech	0661-444444	9000
5	Lina	57	Essaouira	0661-444444	4000
6	Ahmed	27	Fès	NaN	20000
7	Nabil	30	Marrakech	0661-888888	8500
8	Nada	17	Casablanca	0661-999999	5000

df.fillna()

Remplacer toutes les valeurs manquantes

df.fillna('---')

	Nom	Age	Ville	Téléphone	Salaire
0	Youssef	30	Rabat	0661-111111	5000
1	Hanane	25	Casablanca	0661-222222	6000
2	Rachid	25	Tanger	0661-333333	7000
3	Dounia	47	Marrakech	0661-444444	10000
4	Rachida	27	Marrakech	0661-444444	9000
5	Lina	57	Essaouira	0661-444444	4000
6	Ahmed	27	Fès	---	20000
7	Nabil	30	Marrakech	0661-888888	8500
8	Nada	17	Casablanca	0661-999999	5000

df.replace(ancien, nouveau)

Remplacer les valeurs par des valeurs quelconques

```
donnees.replace(1, 'one')
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	one	0	3	Braund, Mr. Owen Harris	male	22.0	one	0	A/5 21171	7.2500	NaN	S
1	2	one	one	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	one	0	PC 17599	71.2833	C85	C
2	3	one	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

df.concat([df1, df2], axis=1)

Jointure par rapport aux lignes

```
pd.concat([df1, df2], axis=1) |
```

	Name	Sex	Ticket	Sex	PassengerId	Survived	Pclass
0	Braund, Mr. Owen Harris	male	A/5 21171	male	1	0	3
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	PC 17599	female	2	1	1
2	Heikkinen, Miss. Laina	female	STON/O2. 3101282	female	3	1	3
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	113803	female	4	1	1
4	Allen, Mr. William Henry	male	373450	male	5	0	3

4-FONCTIONS DE FILTRAGE



df[conditions]

Faire des sélections sur des colonnes

```
df[ df[ 'Salaire' ]>=7000 ]
```

	Nom	Age	Ville	Téléphone	Salaire
2	Rachid	25	Tanger	661333333.0	7000
3	Dounia	47	Marrakech	661444444.0	10000
4	Rachida	27	Marrakech	661444444.0	9000
6	Ahmed	27	Fès	NaN	20000
7	Nabil	30	Marrakech	661888888.0	8500

df[conditions]

Faire des sélections sur des colonnes

```
df[ (df['Salaire']>=7000) & (df['Age'] >= 27) ]
```

	Nom	Age	Ville	Téléphone	Salaire
2	Rachid	25	Tanger	0661-333333	7000
4	Rachida	27	Marrakech	0661-444444	9000
6	Ahmed	27	Fès	NaN	20000

df.query(condition)

Faire des sélections sur des colonnes

```
df.query('Salaire>=7000 & Age >= 27')
```

	Nom	Age	Ville	Téléphone	Salaire
2	Rachid	25	Tanger	0661-333333	7000
4	Rachida	27	Marrakech	0661-444444	9000
6	Ahmed	27	Fès	NaN	20000

`df.sort_values('col', ascending=False)`

Faire un tri selon une direction ascendante

```
donnees.sort_values('Age', ascending=False) |
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0



pandas

```
df.sort_values(['Age','Pclass'],  
ascending=[True,False])
```

Filtrage des valeurs de la
colonne Age de façon
ascendante et de la colonne
Pclass de façon descendante

```
donnees.sort_values(['Age','Pclass'], ascending=[True,False])
```

	PassengerId	Survived	Pclass	Name
803	804	1	3	Thomas, Master. Assad Alexander
755	756	1	2	Hamalainen, Master. Viljo
469	470	1	3	Baclini, Miss. Helene Barbara
644	645	1	3	Baclini, Miss. Eugenie
78	79	1	2	Caldwell, Master. Alden Gates