TD n° 2

Traitement des données en tables (dataframes)

Bibliothèque pandas

Présentation

→ L'informatique permet de traiter des quantités d'information très importantes dans des domaines très variés.

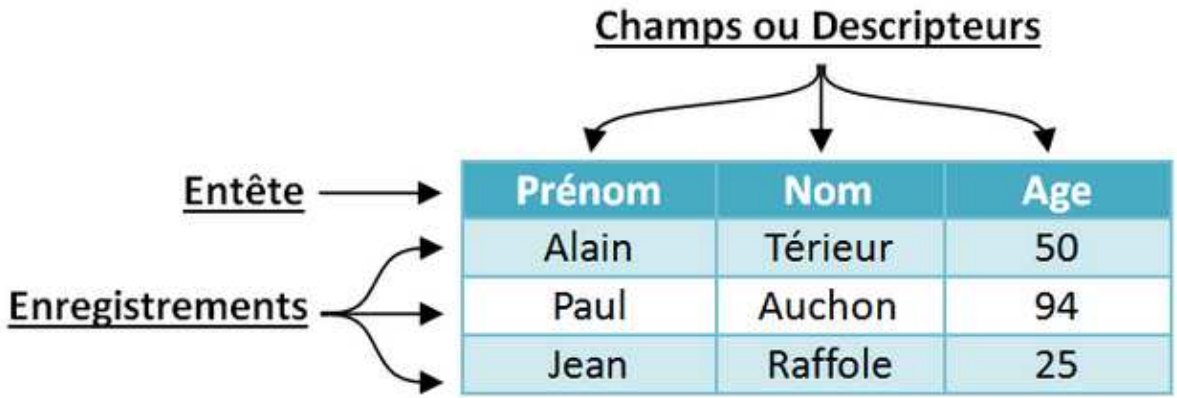
→ Pour s'en convaincre on peut regarder le site `www.data.gouv.fr`.

→ Encore faut-il que ces données soient organisées pour pouvoir les exploiter.

→ Plusieurs formats de fichiers existent dont le format en tables de type **CSV**, objet de ce cours.

Qu'est-ce qu'un fichier CSV ? ?

R107



→ CSV signifie *Comma-Separated Values* autrement dit : Valeurs séparées par des virgules.

→ Ce format permet d'organiser les données en table dans un fichier : Chaque ligne du fichier correspond à un **enregistrement**, Pour chaque ligne, les **descripteurs** (valeurs) sont séparés par des virgules.

→ En réalité les anglo-saxons ont choisi la virgule comme séparateur car leur nombre décimaux sont décrits avec un point (dot en anglais).

→ En France il n'est pas rare d'opter pour le point-virgule, mais d'autres caractères peuvent être utilisés.

Exemple : La table ci-dessus est donc décrite de la manière suivante :

```
Prenom, Nom, Age
Alain, Terieur, 50
Paul, Auchon, 94
Jean, Raffole, 25
```

Petit exercice (nous y reviendrons en TP !)

1. Télécharger le fichier `titanic.csv` depuis Moodle. Double-cliquer. Que se passe-t-il ?
2. Combien y a-t-il de descripteurs ?
3. Combien de valeurs ont été enregistrées ?
4. Utilisons un outil de filtre pour extraire les passagers de 3$^{\text{ème}}$ classe :
   a. Dans l'onglet Données, cliquez sur filtre standard.
   b. Dans Options, cochez la case : La plage contient des étiquettes de colonne
   c. Renseignez Pclass = 3
   d. Édition / Annuler : Filtrer pour revenir en arrière.
5. Faites afficher uniquement les hommes.
6. Faites afficher les survivants.

→ *Mais pour plus de souplesse dans le traitement des données, on va programmer en Python en utilisant la bibliothèque `pandas` idéale dans le traitement des fichiers CSV.*

Principales commandes pandas

Important : Le dataframe CSV doit se trouver dans le même répertoire que le fichier Python.

→ Préambule incontournable :

```
import numpy as np
import matplotlib as plt
import pandas as pd
```

→ Ouvrir un fichier : `pd.read_csv(nom_fichier)`

→ Afficher le début du dataframe : `df.head()`

→ Afficher la dimension du dataframe : `df.shape`

→ Afficher le type de données : `df.dtypes`

→ Statistiques rapides : `df.describe()`

→ Éliminer certaines colonnes : `df.drop(['col', 'col', ...])`

→ Éliminer les lignes aux données manquantes : `df.dropna(axis=0)`

→ Compter les répétitions : `df.['col'].value_count()`

→ Analyse par groupe : `df.groupby(['col'])`

Comment créer un Dataframe pandas ?

Il existe plusieurs façons de créer une trame de données pandas. Les trois principales méthodes sont les suivantes :

- → Créer une trame de données en lisant des fichiers CSV
- → Créer une trame de données à partir de tableaux `NumPy`
- → Créer une trame de données à partir d'un dictionnaire Python

Exercice 1 :

1. Télécharger le fichier `iden.csv` depuis Moodle dans un dossier BDD
2. Créer un fichier `exo1.py` dans BDD
3. Saisir `iden = pd.read_csv("ident.csv")` puis `print(iden)`
4. Qu'observe-t-on ?

Exercice 2 : interpréter chaque ligne du code suivant

```
np.random.seed(42)
mat_alea = np.random.randint(0, 101, (6,4))
etu = ["Alice", "Benjamin", "Cathy", "David", "Erika", "Franck"]
cols = ["semaine_1", "semaine_2", "semaine_3", "semaine_4"]
etu_df = pd.DataFrame(data = mat_alea, index = etu, columns = cols)

print(etu_df)
```

→ Dans Pandas, deux structures de données :

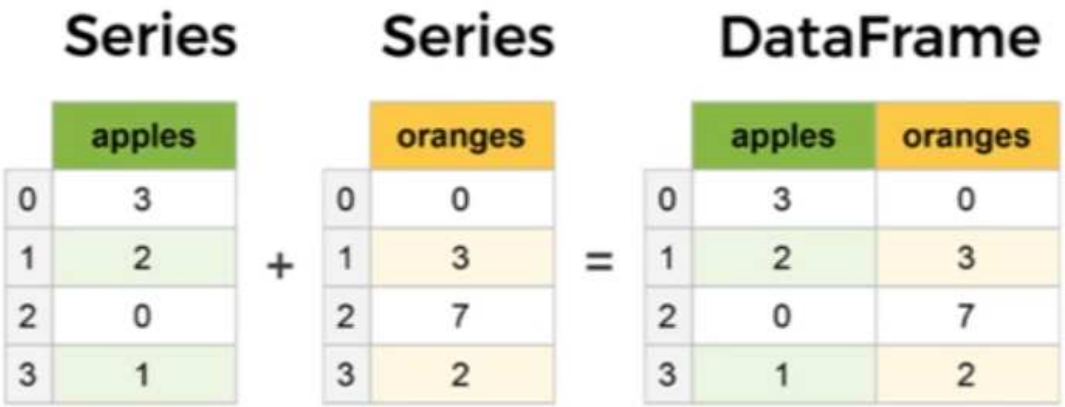
- Séries
- Dataframes

→ Série = Colonne (tableau `numpy` 1D) + index

→ Dataframe = Ensemble de séries

→ Attention l'index n'est pas forcément un range...cf Exo 2 !

→ Ainsi, un Dataframe peut se voir comme un dictionnaire qui contient des séries



Exercice 3 :

```
np.random.seed(42)
mat_alea = np.random.randint(0, 101, (6,4))
data_dict = {}
etu = ["Alice", "Benjamin", "Cathy", "David", "Erika", "Franck"]
for students, stu_data in zip(etu, mat_alea) :
  data_dict[students] = stu_data
etu_df = pd.DataFrame.from_dict(data_dict, orient = "index")
print(etu_df)
```

Conséquences « sympas » :

- → `data_df["col"]` : (série)
- → `data_df["col"][0:3]` : (indexing ou slicing)
- → `data_df["col"] < valeur` : (masque)
- → `data_df[data_df["col"] < valeur]` (indexing ou slicing booléen)
- → `data_df["col1", "col2"]` : (dataframe)
- → `data_df.iloc[0, 0]` : (indexing type numpy)
- → `data_df.iloc[0:2, 0:2]` : (slicing type numpy)
- → `data_df.loc[0:2, "col"]` : (indexing ou slicing colonne type numpy)

Exercice 4 : Facile

1. Éditer dans un tableur la liste des étudiants de la classe avec comme descripteur Nom, Prénom, Age
2. Enregistrer en CSV
3. Éditer dans Python et manipuler à votre convenance !

Exercice 5 : Moyen

Simuler avec pandas un tableau de notes aléatoires de 10 étudiants fictifs sur l'ensemble des matières de la LP du S1.

Exercice 6 : Difficile

1. Récupérer le fichier `villes.csv` et ouvrir depuis Python Pandas.
2. Extraire le sous-groupe des villes et villages de l'Hérault (34)
3. Modifier la colonne nb hab 2010 afin de créer quatre catégories :
   - Les villages de moins de 100 habitants
   - Les villages entre 101 et 2000 habitants
   - Les villes entre 2001 et 10000 habitants
   - Les villes de plus de 10000 habitants.

`https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html`