Introduction à Python pour la Data Analyse

Angelo.Steffenel@univ-reims.fr

Partie 1 – Exercices Python

Accéder à https://repl.it/languages/python3
Vous pouvez utiliser l'interface sans créer un compte. L'avantage du compte est de pouvoir enregistrer le code que vous avez fait dans ce cours.

Ex 1.1 : une somme simple

Dans la fenêtre au milieu (éditeur du code main.py), taper votre programme.

Attribuer la valeur 5 à une **variable a**, et la valeur 15 à une **variable b**.
Créer une variable somme qui fait l'addition **a+b**.
Utiliser la méthode ***print()*** pour imprimer la somme.

Une fois le code prêt, vous pouvez cliquer le bouton "run" pour exécuter votre code.

Ex 1.2 : Pair ou impair ?

Effacer le code précédent.
Utiliser la méthode ***input()*** pour demander qu'un nombre soit entrée au clavier
Attention au type de donnée
Utiliser un **test logique *if*** pour afficher à l'écran si le nombre est pair ou impair
Astuce : le reste de la division entière (%) d'un nombre pair par 2 est égale à zéro

Ex 1.3 : ne me quittes pas

Reprendre le code de l'exercice 3 et l'insérer dans une **boucle *while***.
Le code sera répété tant que la valeur lue au clavier est positive.

Ex 1.4 : listes

Cette fois-ci, créer une liste fixe avec les valeurs suivantes :
nombres = [1, 2, 3, 4, 5]
Grâce à une **boucle for**, imprimer le carré de chacun des nombres de cette liste

Ex 1.5 : listes de courses

Les listes peuvent contenir aussi des strings (chaînes de caractères).
Créer une liste comme la suivante :
courses = ['farine', 'œufs', 'beurre', 'sucre', 'PQ', 'popcorn', 'fromage']
Lire au clavier un numéro
Si le numéro est inférieur à la longueur de la liste courses, imprimer le contenu de la position désignée (ex : ***print(courses[2])*** affichera beurre).
Sinon (numéro supérieur à la longueur), afficher "Option hors de la liste" et finir l'exécution
Astuce : le test logique **if condition:** peut être suivi d'un autre bloc **else:** qui correspond à "sinon"

Partie 2 – Manipulation de données avec Pandas

Nous utilisons toujours l'interface web https://repl.it/languages/python3

Ex 2.1 – Vérification des données

Dans ce premier exercice, vous allez charger un dataset pour pouvoir obtenir un certain nombre d'informations. Pour cela, vous allez importer la bibliothèque pandas avec l'alias **pd** et charger un DataFrame **df** avec des données issues de plusieurs recensements de la population mondiale entre 1960 et 2014. Ce dataset a été obtenu à partir de la Banque Mondiale.
Effacer le code précédent et entrer ceci :

```
import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/lsteffenel/
IntroPythonData/master/world_ind_pop_data_nulls.csv")
```

Votre première tâche est celle d'utiliser ***df.head()*** et ***df.tail()*** pour afficher la première et la dernière entrée de ce dataset.

Ex 2.2 – Types de données

Pandas connaît les types de données des colonnes d'un DataFrame. Il sait aussi reconnaître les données de type **null** et **NaN** ('Not-a-Number'), qui indiquent souvent une entrée manquante. Dans cet exercice, nous continuons à utiliser le DataFrame df de l'exercice précédent, qui contient quelques valeurs NaN.
Votre défi est d'utiliser ***df.info()*** pour afficher combien d'entrées il y a en tout, et identifier s'il y a des colonnes avec des données manquantes.

Ex 2.3 – Des statistiques rapides

Grâce à ma méthode ***df.describe()***, retrouvez la valeur moyenne de la population urbaine (enregistrée en tant que pourcentage de la population totale) et la population du plus petit pays recensé.

Ex 2.4 – Nettoyer un fichier "problématique"

Dans les exercices précédents on vous a donné les instructions pour charger un fichier. Cette fois-ci, vous allez écrire vous-même cette instruction, mais il y a un problème. Le fichier ne répond pas au format CSV classique : les champs sont séparés par espaces et pas par des virgules, l'entête a plusieurs lignes, et des commentaires (marqués avec #) apparaissent partout dans le document. Voici donc les instructions pour nettoyer ce fichier :

1. Charger le fichier que se trouve à l'adresse "`https://raw.githubusercontent.com/lsteffenel/IntroPythonData/master/messy_stock.tsv`". Ce fichier contient les cotations en bourse de quatre compagnies, obtenues à partir de Yahoo Finance. Les données sont stockées dans un format avec une ligne par compagne, avec le prix de clôture de chaque mois.
2. Utiliser la méthode ***head()*** pour afficher les premières lignes du fichier.
3. Charger le fichier dans un DataFrame **df2**, mais cette fois-ci vous allez rajouter des paramètres à la méthode ***pd.read_csv()*** :
    a. Pour indiquer le séparateur des colonnes → ***delimiter=" "***
    b. Pour indiquer le caractère qui démarque les commentaires → ***comment="#"***
    c. Pour indiquer que l'entête s'étend sur trois lignes → ***header=3***
4. Afficher les premières lignes de votre dataset propre **df2**
5. Enregistrer ce dataset. Pour cela, utiliser la méthode ***pd.to_csv('propre.csv', index=False)***. Un fichier "propre.csv" a été rajouté à gauche de votre fenêtre. Vous pouvez cliquer sur ce fichier pour afficher son contenu.

Partie 3 – SQL avec Pandas

Nous utilisons toujours l'interface web https://repl.it/languages/python3
Pour cet exercice nous allons utiliser le dataset que se trouve à l'adresse "https://raw.githubusercontent.com/lsteffenel/IntroPythonData/master/tips.csv"

Ex 3.1 – SELECT

Nous allons commencer cet exercice en faisant la même requête vue en cours, i.e., faire une sélection avec juste une partie des colonnes ('total_bill', 'tip', 'smoker' et 'time' uniquement). Nous allons aussi limiter le nombre d'entrées affichées grâce à ***head()***.

tips[['total_bill', 'tip', 'smoker', 'time']].head(5)

Ex 3.2 – WHERE

Nous pouvons aussi passer des conditions afin de filtrer les données retenues. Ceci est équivalent à la clause WHERE du SQL. Voici donc un exemple de filtrage sur Pandas :

tips[tips['time'] == 'Dinner'].head(5)

Uniquement les entrées faites pendant les dîner seront affichées.

Pouvez-vous faire une requête avec une condition multiple ? Affichez seulement les entrées où il y a au moins 5 personnes à table OU dont l'addition était supérieure à $45

Astuce : utiliser le symbole | qui représente le OU logique entre deux conditions.

Partie 4 – Visualisation avec Pandas

Nous utilisons toujours l'interface web https://repl.it/languages/python3
Pour cet dernier exercice nous allons utiliser le dataset que se trouve à l'adresse "https://raw.githubusercontent.com/lsteffenel/IntroPythonData/master/aapl.csv"

Ex 4.1 – Charger les données et afficher les valeurs d'une colonne

Utiliser le code ci-dessous pour initialiser votre programme :

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

src="https://raw.githubusercontent.com/lsteffenel/IntroPythonData/master/aapl.csv"
aapl = pd.read_csv(src, index_col='Date', parse_dates=True)
close_arr = aapl['Close'].values
plt.plot(close_arr)
plt.savefig('close.png')
```

Ex 4.2 – Afficher tout le dataset

On peut afficher tout le dataset avec ***aapl.plot()***. Utilisez cette commande, suivie de plt.savefig() pour enregistrer un deuxième fichier "aapl.png"

Ex 4.3 – Choisir les colonnes et améliorer la visibilité

Pour finir, nous allons sélectionner juste quelques colonnes ('Open' et 'Close), puis changer le format d'affichage (au lieu de simples lignes, nous voulons utiliser des "area" remplies).

```
detail = aapl.loc[:,['Open','Close']]
detail.plot(kind='area')
plt.savefig('openclose.png')
```