kuikops

MEC – Data Engineering

TP Python / Git

L'objectif de ce TP est de manipuler des données à l'aide de **Pandas** et **NumPy**, en appliquant divers filtres, jointures et opérations courantes en analyse de données. Vous devrez :

1. Télécharger un fichier CSV depuis une source externe.
2. Charger et traiter les données dans un **DataFrame**.
3. Effectuer des jointures, des filtres, et diverses transformations sur ces données.
4. Répondre à des questions en réalisant des tâches de programmation spécifiques liées à l'analyse des données.

URL des fichiers à utiliser

Nous allons travailler sur deux fichiers disponibles en ligne :

1. **Cities.csv** :

   Téléchargeable ici : https://people.sc.fsu.edu/~jburkardt/data/csv/cities.csv

2. **Population_data.csv** :

   Téléchargeable ici : https://people.sc.fsu.edu/~jburkardt/data/csv/hw_200.csv

Fichier 1 : cities.csv

Le fichier contient des informations géographiques sur des villes :

- **"City"** : Nom de la ville
- **"LatD"** : Latitude en degrés
- **"LatM"** : Latitude en minutes
- **"LatS"** : Latitude en secondes
- **"NS"** : Hémisphère nord ou sud (N/S)
- **"LonD"** : Longitude en degrés
- **"LonM"** : Longitude en minutes
- **"LonS"** : Longitude en secondes
- **"EW"** : Hémisphère est ou ouest (E/W)
- **"Population"** : Population de la ville
- **"Area"** : Surface en km^2 de la ville

Fichier 2 : Population_data.csv

Le fichier contient des informations démographiques fictives :

- **"City"** : Nom de la ville
- **"Population_2020"** : Population de la ville en 2020
- **"Growth_rate"** : Taux de croissance démographique en pourcentage

Questions :

1. Ajouter une colonne "Latitude totale" et "Longitude totale" dans le fichier cities.csv

Calculez la latitude et la longitude totale en combinant les colonnes LatD, LatM, LatS (latitude) et LonD, LonM, LonS (longitude) en degrés décimaux. Ajoutez ces colonnes au DataFrame df_cities.

- Latitude totale : LatD + (LatM / 60) + (LatS / 3600)
- Longitude totale : LonD + (LonM / 60) + (LonS / 3600)

2. Filtrer les villes situées dans l'hémisphère Nord

Filtrez le DataFrame df_cities pour obtenir uniquement les villes situées dans l'hémisphère Nord (NS == 'N').

3. Fusionner les deux DataFrames sur la colonne "City"

Réalisez une jointure entre les DataFrames df_cities et df_population sur la colonne "City". Gardez toutes les villes de df_cities même si elles n'ont pas de correspondance dans df_population.

4. Calculer la population projetée pour 2025

En utilisant la colonne Growth_rate (taux de croissance), calculez la population projetée pour 2025 pour chaque ville. Ajoutez une nouvelle colonne "Population_2025" dans le DataFrame fusionné.

Formule :

$$\text{Population\_2025} = \text{Population\_2020} \times (1 + \frac{\text{Growth\_rate}}{100})^5$$

5. Afficher les villes avec une population projetée supérieure à 1 million en 2025

Filtrez et affichez les villes pour lesquelles la population projetée en 2025 dépasse 1 million.

6. Créer un graphique de la population projetée pour 2025

Utilisez **Matplotlib** pour créer un graphique à barres des 10 villes ayant la population projetée la plus élevée en 2025.

7. Calculer la densité de population des villes

Calculez la densité de population (habitants par km^2) pour chaque ville en utilisant les colonnes Population et Area. Ajoutez une nouvelle colonne "Densité_population" au DataFrame df_cities.

Formule :

$$\text{Densité\_population} = \frac{\text{Population}}{\text{Area}}$$

8. Trouver la ville avec la plus grande et la plus petite population

Utilisez les fonctions de Pandas pour trouver la ville avec la plus grande et la plus petite population dans le DataFrame df_cities.

9. Filtrer les villes avec une densité de population supérieure à 5000 habitants/km^2

Affichez toutes les villes pour lesquelles la densité de population dépasse 5000 habitants/km^2.

10. Trouver la moyenne et la médiane de la population des villes

Calculez la **moyenne** et la **médiane** de la population des villes dans le DataFrame df_cities.

11. Créer une nouvelle colonne pour la population normalisée

Normalisez les valeurs de la colonne Population entre 0 et 1, et ajoutez une nouvelle colonne "Population_normalisée" dans df_cities.

Formule de la normalisation :

$$\text{Population\_normalisée} = \frac{\text{Population} - \min(\text{Population})}{\max(\text{Population}) - \min(\text{Population})}$$

12. Utiliser NumPy pour créer un tableau de statistiques descriptives

Utilisez **NumPy** pour générer un tableau de statistiques descriptives (moyenne, écart-type, minimum, maximum) pour la colonne Population du DataFrame df_cities.