Programmation Python
TP n° 3 : Analyse de données

Objectifs d'apprentissage

Dans ce TP, vous acquerrez une expérience pratique dans :

- la manipulation de fichiers CSV en Python ;
- l'analyse de données ;
- tracer des graphiques.

Consignes :

- N'oubliez pas de continuer à typer, tester et documenter toutes vos réponses !
- Vous pouvez travailler dans un ipy notebook ou dans un fichier .py.
- Vous pouvez utiliser Python classique ou le package pandas.
- Réfléchissez à comment vous pourriez présenter vos solutions.

Nous allons nous exercer sur des données importées depuis un fichier CSV contenant des relevés Météo-France de février 2019, disponibles en ligne sur `https://donneespubliques.meteofrance.fr/donnees_libres/Txt/Synop/Archive/synop.201902.csv.gz`.

1. Télécharger et décompresser le fichier. Importer le fichier CSV comme DictReader.

Nettoyage des données

Nous n'aurons pas besoin de conserver l'ensemble des colonnes de chaque enregistrement. Seules les colonnes suivantes nous intéressent :

- `numer_sta` : l'identifiant de la station météo
- `ff` : la vitesse du vent moyen les dix dernières minutes, en m/s
- `t` : la température en Kelvin
- `u` : l'humidité en pourcentage
- `rr1` : les précipitations dans la dernière heure, en mm
- `date` : la date du relevé sous le format `AAAAMMJJhhmmss`

2. Créer une liste d'enregistrements qui ne contient que ces colonnes. Chaque enregistrement sera un namedtuple (`from collections import namedtuple`) à six champs. Renommer `uid` le champ `numer_sta`, `humidite` le champ `u`, `precipitations` le champ `rr1`, `temperature` le champ `t` et `vitesse_vent` le champ `ff`.
3. Certains enregistrements sont incomplets, certains de leurs champs étant alors égaux à `mq` : on ne considèrera pas les enregistrements dont l'un des champs est égal à `mq`. Ecrire une fonction qui enlève ces enregistrements. Vous devriez normalement récupérer 12 860 enregistrements complets.
4. Toutes les données sont actuellement encodées sous forme de chaînes de caractères. Convertissez-les en des types appropriés, et convertir la vitesse du vent en km/h et la température en degrés Celsius, sachant qu'une température de x Kelvin équivaut à $x - 273.15$ degrés Celsius. Pour les dates, utiliser `from datetime import datetime` et la fonction `datetime.strptime`.
5. Comparer les 20 premières lignes des données converties et des données originales. Expliquer pourquoi les données converties contiennent des valeurs telles que `temperature=0.4000000000000341`. Ces valeurs posent-elles un problème ? (Voir aussi chapitre 15 du tutoriel Python et cet article.)

Statistiques

6. Écrire une fonction renvoyant la température minimale relevée.
7. Écrire une fonction renvoyant l'identifiant de la station météo ayant relevé la vitesse maximale de vent.

   Le fichier `https://donneespubliques.meteofrance.fr/donnees_libres/Txt/Synop/postesSynop.csv` contient les noms et coordonnées de chaque station d'observation.
8. Télécharger et importer ce fichier CSV et créer un dictionnaire dont les clés sont les numéros des stations et les valeurs sont des tuples contenant les noms des stations et leur coordonnées. Où se trouve la station météo ayant relevé la vitesse maximale de vent ?
9. Écrire une fonction renvoyant le taux d'humidité moyen sur l'ensemble des relevés.
10. Écrire une fonction renvoyant le niveau de précipitation moyen relevés par les stations ayant un identifiant compris entre `60000` et `69999`.

Recherche d'une station

11. Écrire une fonction filtrant la liste des relevés pour ne renvoyer que les relevés d'une station d'identifiant donné en argument.
12. Utiliser les fonctions de tri de la librairie standard de Python afin de trier la liste des enregistrements par numéro d'identification croissant.
13. Modifier la fonction de filtre pour qu'elle commence par trouver (en temps logarithmique) un enregistrement de la station d'identifiant donné en argument, puis qu'elle construise dans un second temps la liste des relevés de cette station.

Fusion de tables

Comparons les relevés de février 2019 avec ceux de février 2009. Un fichier similaire pour février 2009 est disponible ici : `https://donneespubliques.meteofrance.fr/donnees_libres/Txt/Synop/Archive/synop.200902.csv.gz`
Plus précisément, on cherche à comparer les relevés de température des stations en activité lors de ces deux mois (les identifiants des stations météo n'ont pas changé depuis, mais certains stations ont disparu et d'autres sont apparues), pour chaque relevé.

14. Écrire une fonction permettant de fusionner les deux tables extraites des fichiers de 2009 et de 2019, pour conserver dans chaque enregistrement l'identifiant de la station, la date du relevé, et deux champs `t2009` et `t2019` donnant le relevé de température (différent de `mq`) en février 2009 et février 2019, respectivement.
15. Utiliser la table fusionnée pour en déduire lequel des deux mois a été le plus chaud.

Graphes

16. Créer un graphique qui montre, pour une station donnée, la progression de la température en février 2009 et en février 2019.
17. Entraînez-vous à créer tout autre graphique d'intérêt à partir de ces tableaux. Par exemple, un graphique comme dans la question précédente, mais prenant la moyenne de toutes les stations ; un graphique qui compare les températures moyennes par jour en 2009 et 2019 (ce qui devrait permettre d'obtenir un graphique plus lisse) ; un graphique à barres qui montre le nombre de jours de gel par station ; ...