UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Info. TP8: Derniers tracés

- http://matplotlib.org/api/pyplot_api.html (doc pyplot)

Rendu : sous forme d'archive

À la fin, vos réponses seront rendues dans une archive zip : cette archive contiendra un fichier `compte-rendu.txt` (avec votre **nom**, **groupe**, et les réponses aux questions précédées d'une apostrophe) et vos **fichiers pythons** (et graphiques etc).

Pour faire une **archive zip** de votre TP pour le rendu, exécutez, dans le terminal :

```
cd ~/info-spichi/
zip -r tp8.zip tp8/
```

Important : mise en place

- Suivre les consignes des TP précédents, et travailler dans un dossier tp8.

Exercice 1 – Plus loin dans les dimensions

Rappel : pour cet exercice, travaillez dans un dossier `ex1` lui même dans le dossier `tp8`.

Rappel Tp5 et 6: Bluecime veut faire des statistiques météorologiques pour des stations de ski. Afin de prévoir le format avant d'avoir reçu toutes les données, un générateur de données synthétiques a été créé.

Les recherches pour retrouver Régis ont été abandonnées. Vous êtes donc officiellement responsable du générateur de données. Vous vous rendez compte qu'avoir autant de lignes que de mesures effectuées (une pour chaque heure de chaque jour) n'est pas toujours pratique et décidez d'ajouter une nouvelle dimension au tableau de données.

Plutôt que d'utiliser les dimensions (nombre de jours*24, nombre de type de mesure) vous allez utiliser (nombre de type de mesure, nombre de jours, 24) qui vous semble plus logique.

Ouvrez le fichier `generateur_donnees.py` téléchargeable sur le site du cours. Cela peut être réalisé directement depuis le terminal avec les commandes :

```
wget https://learn.heeere.com/2020-infospichi-0b73/raw/generateur_donnees.py
```

Q1) La fonction `generer_donnees` a déjà été modifiée, adaptez `generer_temperatures`, `generer_ensoleillement`, `genere_neige`, et `generer_vent` pour qu'elles renvoient des tableaux de la bonne taille de sorte que `generer_donnees` fonctionne. Attention, les données créées doivent toujours respecter les contraintes sur les valeurs, notamment au moment des transitions entre les jours.

Votre patron vous fait remarquer que les stations de ski ont plusieurs remontées (mécaniques), donc plusieurs systèmes. Vous devez donc rajouter une dimension à vos données afin d'en tenir compte.

Q2) Dans `generateur_donnees.py`, créez la fonction `generer_donnees_station`, qui en fonction d'un nombre de remontées et d'un nombre de jours va créer un tableau de données aléatoires de dimensions (nombre de remontée, nombre de type de mesure, nombre de jours, 24) où les données de chaque remontée sont générées avec `generer_donnees`.

Suite à vos précédents travaux, votre patron vous fournit la liste finale des graphiques à fournir. Vous ferez les tracés avec des données générées pour 3 remontées sur 37 jours.

Q3) Créez un fichier `generateur_courbes.py` dans lequel vous écrirez *une fonction pour chacune des questions* suivantes, fonction que vous prendrez bien soin de tester au fur et à mesure en vérifiant bien les fichiers PDF générés.

`Dans les questions suivantes`, vous devrez générer un (ou plusieurs) fichier(s) PDF contenant le(s) graphique(s), sans marge superflue.
Dans ces questions, vous devez donc choisir des noms de fichiers (ex. `q4-tous-temps-remontee-0.pdf`), et donner dans la mesure du possible un titre explicite à chaque graphique, des légendes (si plusieurs courbes sont tracées), des noms (avec l'unité) aux axes, et faire en sorte que les axes soient de longueur constante pour chaque type de mesure.
Lorsqu'il sera demandé un tracé « en fonction du temps », c'est à dire sur plusieurs jours, vous afficherez sur l'axe correspondant uniquement le début de chaque jour (numéroté de 1 à 37).
Les clients à qui le système va être présenté sont étasuniens, vous allez donc devoir fournir des températures en Fahrenheit, des épaisseurs de neige en pouces, et une vitesse de vent en mile/h. De plus, les mesures d'ensoleillement devront être fournies en kWh/m². Une bonne solution est de faire cette modification, une fois pour toute, juste après avoir généré les données.

- Celsius -> Fahrenheit : °C × 1.8 + 32 <-> °F
- centimètre -> pouce : cm / 2.54 <-> po
- kilomètre -> mile : km × 0.6213712 <-> mi
- Watt-heure -> Joule : 1Wh <-> 3600J

Q4) Dans un fichier par remontée, tracez dans 4 graphiques (disposés sur une colonne), les courbes en fonction du temps: 1- des températures; 2- de l'ensoleillement; 3- des épaisseurs de neige; 4- de la vitesse du vent.

Q5) Dans un fichier par remontée, tracez dans 2 graphiques (disposés sur une ligne), 1- les températures minimales, moyennes, et maximales à l'ombre à 0h, 1h, … 23h sous la forme de barres horizontales; 2- même tracé mais avec les températures au soleil.

Q6) Dans un fichier, tracez dans un graphique par remontée (disposés sur une colonne), La courbe des valeurs d'ensoleillement mesurées la journée où la moyenne de l'ensoleillement a été la plus basse, et où elle a été la plus haute.

Aide: `np.argmax(tableau)` renvoie l'index de la valeur maximale dans `tableau`.

Q7) Dans un fichier par remontée, tracez dans 3 graphiques (les deux premiers sur une ligne, le troisième prenant toute la seconde ligne). 1- la proportion de mesures où chacune des jauges a contenu une plus grande épaisseur de neige que les autres, sous forme de diagramme en "camembert". 2- *l'importance* moyenne de l'épaisseur de neige de chaque jauge (donc 3 courbes) en fonction de l'heure (0h, 1h, …, 23h). Pour ce faire, (pour chaque heure 0h, …, 23h) soustrayez la moyenne des épaisseurs dans les 3 jauges (et sur tous les jours) à la moyenne des épaisseurs dans chaque jauge (moyenne aussi sur tous les jours). 3- la moyenne des épaisseurs dans les trois jauges en fonction des heures sous forme de barres verticales

Aide: `np.bincount(tableau)` qui permet de compter le nombre d'occurrences de chaque valeur d'un tableau d'entiers. `np.bincount([5,2,5])` par exemple renvoie `[0,0,1,0,0,2]`, aucun 0, aucun 1, un 2, aucun 3, aucun 4, deux 5 (s'arrête à la valeur maximale)
Aide: le 3ème argument de `plt.subplot` peut être l'indice d'un graphique ou bien un tuple contenant 2 indices. Dans ce cas, l'espace compris entre ces indices sera occupé par un seul graphique. `plt.subplot(4, 2, (1, 4))` correspond donc à graphique recouvrant entièrement les 2 colonnes des 2 premières lignes.

Q8) Dans un fichier, tracez dans un graphique par remontée (disposés sur une colonne), le nuage de points où chaque point est un jour avec la vitesse minimum du vent en x et la vitesse max en y.