

PROJET PYTHON



Le but de ce projet est de comparer des observations produites à partir d'un échantillon de vols en partance de la France.

Base de données

Pour ce projet, nous avons récupéré une base de données venant du site :
<https://www.eurocontrol.int/Economics/DailyTrafficVariation-States.html>.

Mais elle a été réduite et modifiée au préalable pour être manipulable dans le cadre du cours.

(le fichier a été nettoyé et mis en format txt et ce n'est qu'un échantillon de la vraie base)

Donc utilisez bien la base qui est fournie dans le fichier zip du projet !

En ouvrant le fichier txt, vous constaterez que vous avez, en colonne :

les jours du mois	le nombre de vols pour l'année 2022	le nombre de vols pour l'année 2021	le nombre de vols pour l'année 2019
-------------------	-------------------------------------	-------------------------------------	-------------------------------------

Vous comprenez bien qu'il vous faudra mesurer l'impact du COVID-19 sur les vols (entre 2019 et 2021) puis commenter la situation en 2022 (reprise ou non et de combien).

Tout cela sera fait à partir de quelques exercices.

Mise en place

1. Récupération des données

Pour commencer, il vous faudra récupérer les informations contenues dans le fichier texte. Comme vu en cours, il faudra utiliser les fonctions :

open(chemin), **readlines**(fichier) ou **readline**(fichier) (au choix)

2. Réorganisation des données

Comme vu en cours, les données sont groupées lignes par lignes dans la liste, ce qui fait qu'elles ne sont pas sous une forme exploitable pour la suite.

Il faudra donc les séparer et les organiser avec des méthodes telles que **liste.split()**, **liste.append()** (ou **liste = liste + [objet]**) ou des fonctions comme **del liste[index]**.

De plus, vous constaterez que les éléments sont des chaînes de caractères, il vous faudra donc les transformer en nombre avec la commande **int(objet)**

Note : vous pouvez

- soit mettre toutes les lignes dans une liste et vous aurez une grande liste contenant des sous-listes pour chaque jour ;
- soit mettre chaque ligne dans une liste et vous aurez au total 31 listes.

Cela dépendra de quelle fonction vous avez choisi pour extraire les données.

Maintenant, que les données sont organisées dans la liste ou les listes, vous pouvez commencer les exercices.

Exercice 1 - Comparaison

Comme annoncé précédemment, ceci est un exercice de comparaison des données. Votre code devra répondre aux questions et consignes ci-dessous.

Y a-t-il un jour où le nombre de vols en 2021 a dépassé celui de 2019 ?

Note : vous pouvez faire une boucle **for** contenant une boucle **if** qui s'arrêterait au premier jour qui répond à la question.

S'il n'y en a pas, la fonction doit quand même renvoyer une réponse.

De façon générale, constatez-vous que les vols sont plus importants en 2022 qu'en 2019 ?

Note : cette fois-ci, il faut voir tous les cas où cela a eu lieu.

Au 14 juillet de chaque année, quel était le nombre de vols sortant de France ? Que remarquez-vous ?

Écrivez une fonction qui renvoie le nombre de vols pour le 1 et le 31 juillet et commentez.

Note : y a-t-il des points à relever en début et en fin de mois ?

Exercice 2 - Minimums, Maximums, Moyennes

Écrivez une fonction ou un script qui renvoie le min, le max et la moyenne du nombre de vols pour chaque année.

Pour cet exercice, vous utiliserez les fonctions **min(objet)**, **max(objet)**, **mean(objet)**.

Note : pour chaque résultat, faite en sorte de l'afficher avec une phrase.

Utilisez la fonction **print(objet)** ou la fonction **str(objet)** accompagnée de l'opérateur "+".

Écrivez une fonction ou un script qui compare les informations calculées précédemment.

Note : vous ne devez pas commenter les résultats vous-même. C'est à la fonction de retourner les bons commentaires.

Exercice - Graphiques

Ceci est un exercice bonus, si les représentations graphiques vous intéressent, n'hésitez pas.

Pour cet exercice, il vous faudra importer le module pyplot de la bibliothèque matplotlib.

Utilisez la commande **import matplotlib.pyplot as plt**

Il vous faudra :

- définir une liste contenant les données sur l'axe des abscisses (de 1 à 31)
- définir 3 listes chacune contenant les données d'une seule année (si cela n'avait pas été fait avant)

Vous utiliserez les commandes :

plt.plot(x,y)

plt.show()

Vous pouvez vous amuser à personnaliser votre graphique avec les commandes et fonctions ci-dessous (il en existe d'autres, ce n'est qu'un aperçu).

plot(x,y, **paramètres**)

c = 'couleur' ('r' rouge, 'b' bleu, 'c' cyan, 'g' vert, 'k' noir) ;
marker = 'style de point' ('o' point, '+' plus, 'x' croix) ;
ls = 'style de trait entre les points' ('-' ligne continue, '--' pointillés, ':' petits points)

Ces paramètres peuvent être écrits ensembles. Exemple : `plt.plot(x,y, 'or-')`, affiche une graphique avec des points en forme de rond, reliés par un trait et tout en rouge.

Viviana SANCHES RODRIGUES RIBEIRO
viviana.sanches-rodrigues-ribeiro@etu.univ-cotedazur.fr

ms = taille des points (nombre)

lw = épaisseur du trait (nombre)

label = 'titre qui est associé à la courbe'

à la suite du plot :

plt.title("titre du graphique")

plt.legend() permet d'afficher la légende qui a été définie précédemment avec le paramètre label
--

plt.xlabel(nom de l'axe) plt.ylabel(nom de l'axe)
--

xlim(xmin, xmax) et ylim(ymin, ymax)
