

# TP Statistiques et Matplotlib

## Objectif

L'objectif de ce TP est de prendre en main certaines fonctionnalités de la bibliothèque Matplotlib. À la fin de ce TP, vous serez capable de construire des représentations graphiques, en particulier, dans le domaine des Statistiques et vous aurez travaillé/élaboré des stratégies rencontrées lors de l'étude de données statistiques.

## I. Premières manipulations (Débutants)

1. En vous aidant de l'aide mémoire, importer la bibliothèque matplotlib.
2. Placer 3 points en forme de croix rouge dont vous aurez choisi les coordonnées.
3. Créer deux listes possédant le même nombre de valeurs, une pour les valeurs en abscisses et l'autre pour les valeurs en ordonnées puis tracer la ligne polygonale passant par ces points.

### Question supplémentaire

Écrire un programme où l'on crée une fonction avec l'expression de votre choix, puis, à l'aide d'une boucle, on réalise un tableau de valeurs avec une variable variant de 1 à 5 avec un pas de 0,2 (les données seront stockées dans deux listes, une pour les valeurs de la variable, une autre pour les images). Afficher enfin la ligne polygonale que l'on obtient avec les deux listes précédentes.

## II. Représenter des données (Intermédiaires)

1. Dans un nouveau script, importer le fichier `departement` qui contient, par exemple, `numero` (une liste avec le numéro de chaque département hors Corse et DOM-TOM) et `population` (une liste avec la population, en milliers d'habitants, de chaque département hors Corse et DOM-TOM en 2014).
2. Afficher un histogramme composé de 25 classes des populations des départements.
3. Afficher un nuage de points (en forme de rond vert) ayant pour abscisse le numéro des départements et pour ordonnées leur population.

### Question supplémentaire

Écrire un algorithme, puis le programme associé, qui permettrait de déterminer la valeur médiane d'une série statistique (dont les données seraient stockées dans une liste). L'idée est de créer une fonction `mediane`. L'aide mémoire est précieux ! Utiliser votre programme pour déterminer la population médiane des départements étudiés.

## III. Travailler avec des séries de données (Experts)

1. Dans un nouveau script, importer le fichier `departement` qui contient, par exemple, `numero` et `population` (voir ci-avant) mais aussi `commune` (une liste avec l'effectif des villes de plus de 50000 habitants de chaque département hors Corse et DOM-TOM).
2. Écrire un programme, qui crée une nouvelle liste `commune_cumule` qui contient les effectifs cumulés croissants du nombre de communes de plus de 50000 habitants parmi les départements étudiés.
3. Poursuivre ce programme pour faire en sorte de déterminer graphiquement la médiane de cette série. Il faudra afficher la ligne polygonale souhaitée mais aussi le segment attendu dont l'ordonnée des points sera calculée à l'aide de la liste des effectifs cumulés croissants. (On pourra aussi traiter les cas du premier quartile et du troisième quartile)

**Mémo supplémentaire :** la fonction `grid()`, à placer avant le `show()`, permet de faciliter la lecture.

4. Lorsque l'on souhaite mettre en relation trois grandeurs numériques, le graphique à bulles est un outil très utile ! Afficher le graphique à bulles donnant la population par département et une bulle plus ou moins imposante en fonction du nombre de villes de plus de 50000 habitants. Dans le fichier `departement`, la liste `commune2` permet un affichage plus adapté pour la surface des bulles et des couleurs ont été stockées dans `couleur`.

**Mémo supplémentaire :** `scatter([xA, xB ...], [yA, yB ...], s = [val1, val2 ...], c = [coul1, coul2...], alpha = val0)` permet de créer un graphique à bulle donc les points ont pour coordonnées (xA;yA), (xB;yB) ... de surface proportionnelle aux valeurs val1, val2 ... de couleur donnée par coul1, coul2 ... d'opacité donnée par val0 (nombre flottant compris entre 0 et 1). Les deux derniers arguments sont facultatifs.

5. Compléter votre graphique en ajoutant un titre, un légende pour chaque axe en utilisant : `title("titre")`, `xlabel("legende1")`, `ylabel("legende2")` avant le `show()`.