TP D'INFORMATIQUE 4
Bibliothèques et fichiers

L'objectif de ce TP est d'étudier l'utilisation des bibliothèques, et la lecture et écriture de fichiers. Les fichiers à utiliser, dont le fichier `exemples.py` contenant les différents exemples mentionnés dans le sujet, sont disponibles sur Cahier de Prépa ou dans le dossier `L:\ELEVES-MPSI\Ressources\ITC\TP4`.

1 Calcul numérique

Une bibliothèque est une collection de fonctions, que l'on peut importer. Aucune connaissance n'est exigée par le programme d'ITC sur le contenu des bibliothèques usuelles de Python, mais il faut savoir les utiliser lorsqu'un sujet vous en donne la documentation.

1. Importer toutes les fonctions de la bibliothèque math en commençant le script Python par `import math`.
2. Le document `debuter_python.pdf` liste des fonctions et constantes de la bibliothèque `math`. En déduire une fonction `transformer_liste` prenant en argument une liste de flottants, et renvoyant une nouvelle liste dans laquelle chaque flottant w est remplacé par $\dfrac{\cos^2(2\pi w)}{2}$.
3. La bibliothèque `numpy` permet de manipuler efficacement des tableaux à plusieurs lignes et plusieurs colonnes (voire plus de dimensions), appelés `ndarray`. On accède à leurs éléments comme pour une liste (`T[i]`, `len(T)`, boucle `for`), mais leur taille est fixe (pas d'`append`) et les opérations comme `+` et `*` agissent élément par élément.

   L'exemple 1 (lisez-le !) présente comment transformer une liste standard en `ndarray`, et montre le comportement des calculs sur les `ndarray` (on rappelle que `alt+entrée` permet d'exécuter la sélection).

   (a) En déduire une variante `transformer_tableau` de la fonction précédente prenant en argument un `ndarray`, et renvoyant un `ndarray`. Cette fonction ne devra pas contenir de boucle.

   (b) Écrire une fonction `moyenne` prenant en argument un `ndarray` de nombres à une dimension, et renvoyant la moyenne de ces nombres. On pourra utiliser la fonction `np.sum` calculant la somme d'un `ndarray`.

   (c) Si l'on considère n points du plan $(x_1, y_1), \dots, (x_n, y_n)$, la méthode classique de la régression linéaire consiste à les approcher par la droite d'équation $y = ax + b$, avec

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \quad \text{et} \quad b = \overline{y} - a\overline{x},$$

   où $\overline{x}$ désigne la moyenne des $(x_i)_{1\leq i\leq n}$ et $\overline{y}$ celle des $(y_i)_{1\leq i\leq n}$.
   Écrire une fonction `coeff_regression_lineaire` prenant en argument deux `ndarray` `X` et `Y` de même taille contenant respectivement les x_i et les y_i, et renvoyant les coefficients a et b obtenus par régression linéaire. Cette fonction ne devra pas contenir de boucle.
   Sur les tableaux `np.array([2.3, 4.7, 5.6, 8.5])` et `np.array([7.2, 6.8, 5.6, 3.9])`, la fonction doit ainsi renvoyer le couple $(a, b) \approx (-0.56, 8.80)$.

2 Tracé de courbes

La bibliothèque `matplotlib.pyplot` permet d'afficher des courbes de fonctions. Ainsi, le code de l'exemple 2 affiche la courbe représentative de la fonction $x \mapsto e^{-x^2}$ entre $a = -10$ et $b = 10$.

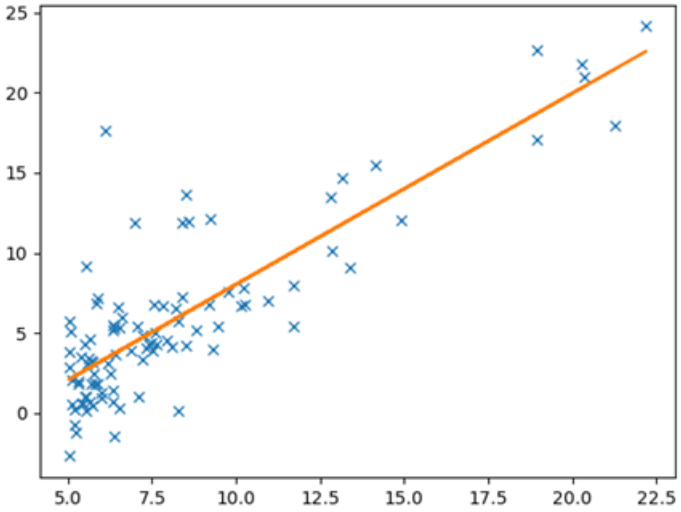
1. Tracer la courbe de la fonction cosinus entre 0 et 2π.
2. L'exemple 3 illustre l'utilisation des fonctions de `numpy` pour condenser le code du tracé, ainsi que la syntaxe pour afficher un titre et choisir la couleur de la courbe.
   Tracer en rouge la courbe constituée des points de coordonnées $(\cos(t)(1+\cos(t)), \sin(t)(1+\cos(t)))$ pour t parcourant $[0, 2\pi]$, avec le titre « cardioïde de paramètre 1 ».

3 Lecture de fichier

Un fichier permet de stocker des données sur le disque dur de l'ordinateur. Il est possible d'accéder à de telles données au sein d'un script Python, comme le montre l'exemple 4. L'exécution de cet exemple nécessite d'abord de placer le fichier `notes.txt` dans le répertoire de travail de Pyzo, dont vous pouvez connaître le chemin en entrant la commande `cd` dans la console de Pyzo.

1. Avec l'explorateur de fichier de Windows, copier le fichier `notes.txt`, le placer dans le répertoire de travail de Pyzo. Ouvrir ce fichier texte et observer comment sont organisées les données qu'il contient.
2. Exécuter le code de l'exemple 4 et observer comment les listes `L1` et `L2` ont été construites à partir des données du fichier.
3. Le fichier `profits.txt` contient plusieurs lignes donnant chacune la population d'une ville et le profit réalisé dans cette ville, séparés par une virgule.

   Placer de la même façon le fichier `profits.txt` dans votre répertoire de travail, et écrire une portion de code Python créant deux listes, `X` contenant les populations et `Y` les profits.
4. On veut à présent représenter chaque ville par un point dont l'abscisse est la population et l'ordonnée le profit réalisé. Tracer sur un même graphique le nuage de points obtenu et la droite l'approximant par régression linéaire. Vous devez obtenir la figure suivante :



4 Écriture dans un fichier

Il est également possible d'écrire des données dans un fichier au moyen d'un script Python, comme l'illustre l'exemple 5.

1. On souhaite dans cette question modifier les notes du fichier `notes.txt` en leur appliquant une *correction gamma*. Pour un coefficient $\gamma \in \mathbb{R}_+^*$ donné, cela revient à remplacer une note x par $20 \times \left(\frac{x}{20}\right)^\gamma$. Cela a pour effet d'augmenter les notes pour $\gamma < 1$, ou de les diminuer pour $\gamma > 1$.

   (a) Écrire une fonction `correction` prenant en argument le coefficient `gamma` et la note `x`, et renvoyant la note modifiée, arrondie au dixième supérieur. On pourra utiliser la fonction `np.ceil`, calculant la partie entière supérieure. L'appel `correction(0.8, 10)` doit renvoyer `11.5`.

   (b) Écrire une portion de code Python créant et remplissant un fichier `notes2.txt` ayant la même structure que `notes.txt` mais où le coefficient $\gamma = 0.8$ a été appliqué aux notes.
2. Créer un fichier `premiers.txt` contenant les 1000 premiers nombres premiers (un par ligne). La dernière ligne doit être 7919.

5 Analyse de texte

Écrire une fonction `analyse_texte` prenant en argument un nom de fichier (supposé contenir un texte en français), et renvoyant le nombre de lignes, le nombre de caractères, et le nombre de mots (qu'on suppose séparés par des espaces sur chaque ligne). On doit ainsi obtenir `(2, 59, 10)` sur le fichier `hello world.txt`.