I Lecture

1.1 Découverte de la structure

Ouvrez le fichier *pays.csv* dans l'éditeur de texte de spyder et expliciter la structure de chaque ligne. Il s'agit de comprendre quelles sont les données présentes et comment elles sont séparées les unes des autres.

Exercice 1
A l'aide de la console et de l'aide python, comprendre les fonctions s.strip et s.replace quand s est une chaîne de caractère. Même question avec s.split.

Exercice 2
Créer une fonction converti qui prend en entrée une ligne de notre fichier (avec le caractère de fin de ligne) et retourne (nom_pays, population, superficie) où la population est un **float** et superficie un **int**.
On pourra se référer à l'énoncé du DM pour la syntaxe de la lecture de fichier.

1.2 Extraction des données

Exercice 3
Créer une fonction **lit_pays(chemin)** qui prend en argument le chemin du fichier csv (ce chemin est une chaîne) et retourne 3 listes : celle des noms de pays, celle des populations et celle des superficie.
On n'oubliera pas de fermer le fichier ouvert après lecture.

Exercice 4
Trouver le pays ayant la plus forte et celui ayant la plus faible densité de population et afficher leurs noms.

II Graphiques

2.1 Rappels sur l'utilisation

```
import matplolib.pyplot as plt
import numpy as np
```

La création de courbes se fait de la manière suivante : On considère deux listes X et Y de même longueur n

```
plt.plot(X, Y)
```

trace la ligne brisée qui relies tous les points de coordonnées $(X[i], Y[i])$ pour i entre 0 et $n-1$

Exercice 5
Tracer la fonction arctan sur $[-5, 5]$. On prendra une liste d'abscisses de longueur 200 (cf **np.linspace**) et on pourra utiliser la fonction **np.arctan**.

Exercice 6
Créer la liste *pop* telle que *pop*$[i]$ contient le nombre de pays ayant une population (en millions) dans $[20i, 20(i+1)[$. Créer également la liste *abscisses* des $20*(i+1)$ pour toutes les valeurs de i dont vous aurez besoin pour *pop* et afficher le graphique correspondant.
Comparer à

```
plt.hist(Lpop, abscisses)
```

où L*pop* est la liste de toutes les populations.

III Ecriture

Exercice 7
Ecrire dans le fichier *densité.csv* les données : nom du pays ; densité.
On pourra au choix donner des titres aux colonnes dans la première ligne.
Pour écrire dans un fichier situé à l'adresse `chemin` :

```
f = open(chemin, 'w')
...
f.write(ici mettre une chaine de caractères à écrire dans le fichier)
...

f.close()
```

Pour passer à la ligne, on utilise le caractère spécial $\backslash n$

Exercice 8
Ecrire une fonction **combine(L1, L2, L3)** qui prend 3 listes de même longueurs comme arguments et retourne la liste $[[L1[0], L2[0], L3[0]], \dots]$ où chaque éléments est une liste de longueur 3.

Exercice 9
Expérimenter la fonction

```
L.sort()
```

où L est une liste construite avec la fonction **combine**. On fera au moins 3 expériences pour 3 valeurs pertinentes de L1.

Exercice 10
Créer les fichiers csv *pays-par-population.csv* et *pays-par-superficie.csv* qui contiennent la liste de nos pays triées dans l'ordre croissant suivant leur titres...

IV Bonus

Trouver comment tracer un cercle de centre donné et rayon donné, comment tracer un polygone régulier à n côtés, une droite d'équation $y = ax + b$ (on donne a, b), cette même droite mais limitée à un rectangle donné (comment se donner un rectangle le plus efficacement possible ?).