À faire individuellement ou par petits groupes de deux ou trois.

Suite de l'analyse du fichier nat2018.csv

Exercice 1. Nouvelle initialisation des variables

Créez un nouveau fichier pour cette série d'exercice, qui devra encore une fois charger le contenu du fichier **`nat2018.csv`**. Reprenez également la **`dataclass DataEntry`** qui avait été créée lors de la série précédente.

a. En utilisant un **`filter()`** et une fonction lambda, filtrez encore une fois le contenu du fichier **`nat2018.csv`**. Si vous avez des difficultés, reprenez ce qui a été présenté en cours.

b. En utilisant une compréhension de liste, créez une liste **`boys: List[DataEntry]`** qui contiendra toutes les lignes représentant les prénoms donnés aux garçons.

c. Toujours avec une compréhension de liste, faites de même pour créer la liste **`girls: List[DataEntry]`**.

Exercice 2. Total des naissances par année

Pour cette partie, nous souhaitons créer une structure de données qui pourra nous permettre de stocker l'information suivante : le nombre total de naissances par année.

a. Selon vous, quelle structure de données serait la plus appropriée pour stocker cette information ?

b. Si vous n'avez pas répondu un **`dict`** à la question précédente, que représenteraient les clés et les valeurs de ce dictionnaire ?

c. En utilisant une compréhension de dictionnaire, créez un dictionnaire **`births_per_year: Dict[int, List[int]]`** où:

- Une clé représente une année de naissance ;
- La valeur correspondante est une liste vide.

d. En utilisant un parcours «classique» dans une liste, complétez les données de ce dictionnaire. **Attention** : les valeurs associées aux clés sont des listes dont les éléments sont les **`data_entry.count`** pour chaque prénom (masculin ou féminin) qui aurait été donné pour une année précise. En particulier, cela signifie que la longueur de cette liste doit être égale au nombre de prénoms différents données sur cette année.

Exercice 3. Utilisation de la fonction `map()`

Le dictionnaire **`births_per_year`** ne donne pas encore le nombre total de naissances par années. Pour cela, il faut calculer la somme des valeurs de la liste associées à chaque clé du dictionnaire. Nous allons faire ceci en utilisant la fonction d'ordre supérieur **`map()`** et une fonction lambda.

Observez le code ci-après :

```
1 births: List[str] = list(map(lambda x: ..., births_per_year.items()))
```

L'objectif de ce dernier exercice est de compléter la fonction lambda. Voici quelques indications pour comprendre cette instruction:

- Pour rappel, la fonction **`map(f, l)`** va appliquer la fonction **`f`** à tous les éléments de la liste **`l`**.
- **`births_per_year.items()`** retourne une liste de tuples **`(key, value)`**, où **`key`** est un entier et **`value`** est une liste d'entiers.

- La fonction lambda n'accepte qu'un seul paramètre : c'est un tuple **`(key, value)`**.
- Pour chaque couple **`(key, value)`**, la fonction lambda doit retourner une chaîne de caractères dont le contenu sera le suivant : **`"key: sum_of_values"`**.
- Une fois que la fonction **`map()`** a fini son exécution, le tout est ensuite stocké dans une liste **`births`**.

Exercice 4. Bonus: Affichage d'un graphe

Avec les données produites, il est possible de produire un graphe avec Python en utilisant la bibliothèque **`plotnine`** qui elle-même utilise la bibliothèque **`pandas`** très largement utilisée pour faire de l'analyse de données de façon un peu plus optimisée.

Le code suivant affiche donc ce graphe :

```
from plotnine import ggplot, aes, geom_line
import pandas as pd

# ... Code construit jusqu'à maintenant

years = []
births_count = []
for item in births:
    year, birth_count = [int(x) for x in item.split(': ')]
    years.append(year)
    births_count.append(birth_count)

naissances_annees = {"Annee": years, "Naissances": births_count}
naissances_par_annees = pd.DataFrame(naissances_annees)

plot = ggplot(naissances_par_annees) + aes(x = "Annee", y = "Naissances") + geom_line()
print(plot)
```

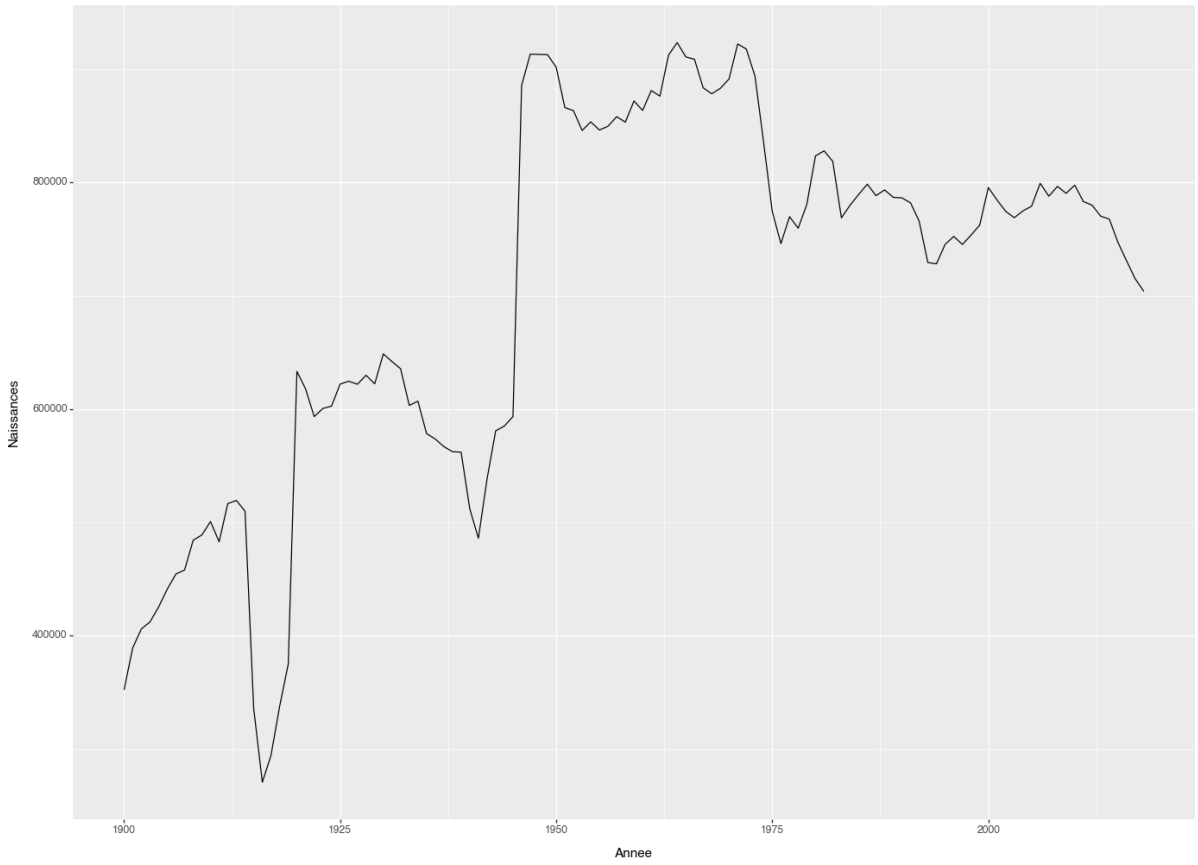

Figure 1: Nombre total de naissances par année.

Pour pouvoir utiliser ce morceau de programme, il faut installer la bibliothèque **`plotnine`**. Pour cela:

- Vérifiez la présence d'un dossier intitulé **`venv`** ou **`.venv`** dans votre dossier de travail (qui devrait s'appeler **`ICC_Prog`** si vous n'avez rien changé depuis le début du semestre).
  - Si vous avez un tel dossier:
    * Ouvrez Visual Studio Code, puis cliquez sur **`Terminal > New Terminal`**.
    * Saisissez la commande suivante : **`source venv/bin/activate`**. Remplacez **`venv`** par **`.venv`** ou toute autre variante si votre dossier s'appelle autrement.
    * Saisissez la commande suivante : **`pip3 install plotnine`**. Cela devrait télécharger la bibliothèque et la rendre accessible **uniquement** dans votre répertoire de travail.
  - Si vous n'avez pas un tel dossier:
    * Ouvez Visual Studio Code, puis cliquez sur **`View > Command Palette`**.
    * Dans la fenêtre ainsi ouverte, saisissez **`Python: Create Environment`** et cliquez sur la suggestion qui vous est donnée.
    * Choisissez **`Venv`**.
    * Choisissez la version de Python 3 la plus récente qui vous est proposée.
    * Cela devrait vous créer un répertoire **`venv`**. Suivez alors les étapes qui se trouvent plus haut.
- Reprenez le code donné plus haut (qui se trouve aussi sur Moodle pour un copier-coller plus facile sans les numéros de ligne) et testez-le pour voir si vous obtenez le même graphe.