Course 1: Matplotlib

Download the source file sources1.zip, available on Moodle. Each exercise is made in a dedicated file.

Exercice 1 : Temperature Curves

The file `exercise1.py` provides average temperature data for the city of Rennes. The goal is to create Python code to generate the graph shown in Figure 1.

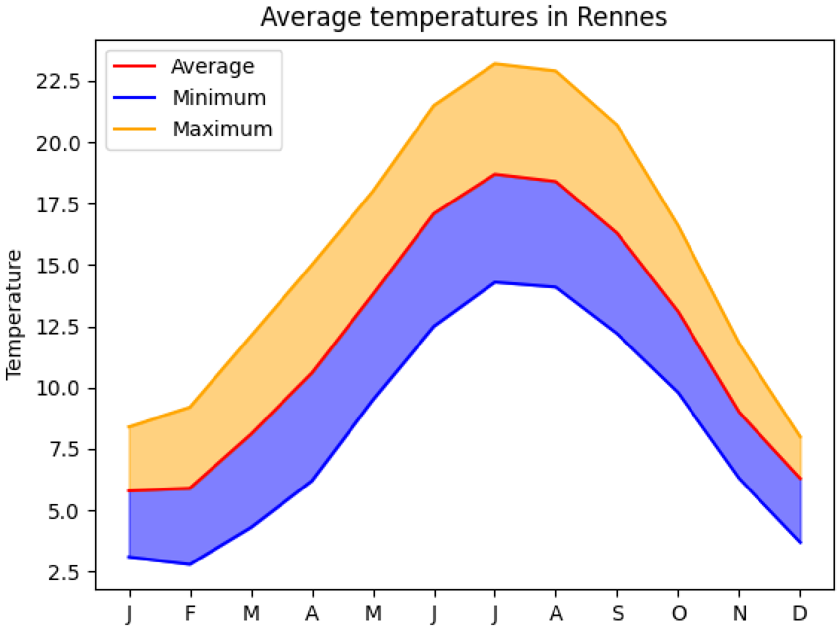

FIGURE 1 – Exercise 1

Q1 - Using the list `months`, generate a list containing only the first letter of each month.

Q2 - Plot the curve of average temperatures as a function of the initials, in red. What do you observe ?

Q3 - Plot the curve of averages without using x-values, but use `xticks` to display the initials of the months along the x-axis.

Q4 - Plot the curve of minimum temperatures in blue.

Q5 - Plot the curve of maximum temperatures in orange.

Q6 - Add a label to the y-axis.

Q7 - Add a legend positioned in the top-left corner.

Q8 - Use `fill_between` to color the spaces between the min and max curves, respectively in blue and orange, with a transparency of 0.5.

Q9 - Add the title of the graph.

Exercice 2 : Waste Analysis in Rennes with Matplotlib

Rennes Métropole provides data on the collection and treatment of household waste. The file `stats-collecte-dechets.csv` contains simplified data from 2002 to 2021 :

- `ANNEE` : the year in question ;
- `POPULATION` : the population of Rennes Métropole ;
- `COLL_DECHETS_ENS` : the total volume of collected waste in tons ;
- `COLL_OM_HAB` : the quantity of household waste collected per inhabitant, in kg ;
- `COLL_DMREC_HAB` : the quantity of recyclable waste collected per inhabitant, in kg ;
- `COLL_VEG_HAB` : the quantity of green waste collected per inhabitant, in kg ;
- `COLL_DECHETERIESENC_HAB` : the quantity of other waste collected at the recycling center per inhabitant, in kg.

The goal is to create a figure similar to the one presented in Figure 2.

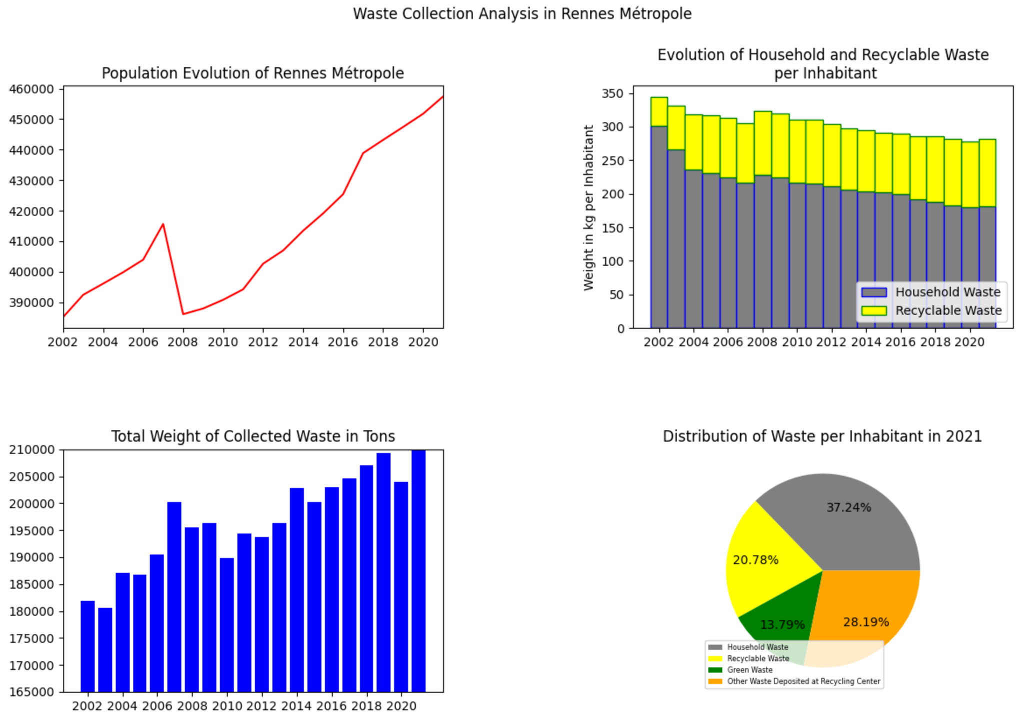

Figure 2 – Exercise 2

Q1 - Load the file `stats-collecte-dechets.csv` into a DataFrame.

Q2 - Create a figure with dimensions 14*9 containing 4 subplots.

Q3 - Plot the curve showing the evolution of the population of Rennes over the years, setting the axis limits between 2002 and 2021.

Q4 - Plot the bar chart of the total waste collection over time, in blue. Specify the y-axis limits for better visualization of the results.

Q5 - Plot the stacked bar chart of household waste and recyclable waste production per inhabitant. You can refer to[1] for inspiration.

[1] `https://www.geeksforgeeks.org/create-a-stacked-bar-plot-in-matplotlib/`

Q6 - Plot the pie chart of waste production per inhabitant in 2021. Extract the values from the DataFrame beforehand. As a reference, the values used to generate this chart are : [181.0, 101.0, 67.0, 137.0]. The legend has a font size of xx-small.

Q7 - Add a global title to the figure ; set the `wspace` and `hspace` margins to 0.5.

Exercice 3 : Age Pyramid

The file `pyramide2022.xslx` contains population data for Metropolitan France[2], based on the year of birth, for both men and women. The goal is to create a graph similar to the one shown in Figure 3.

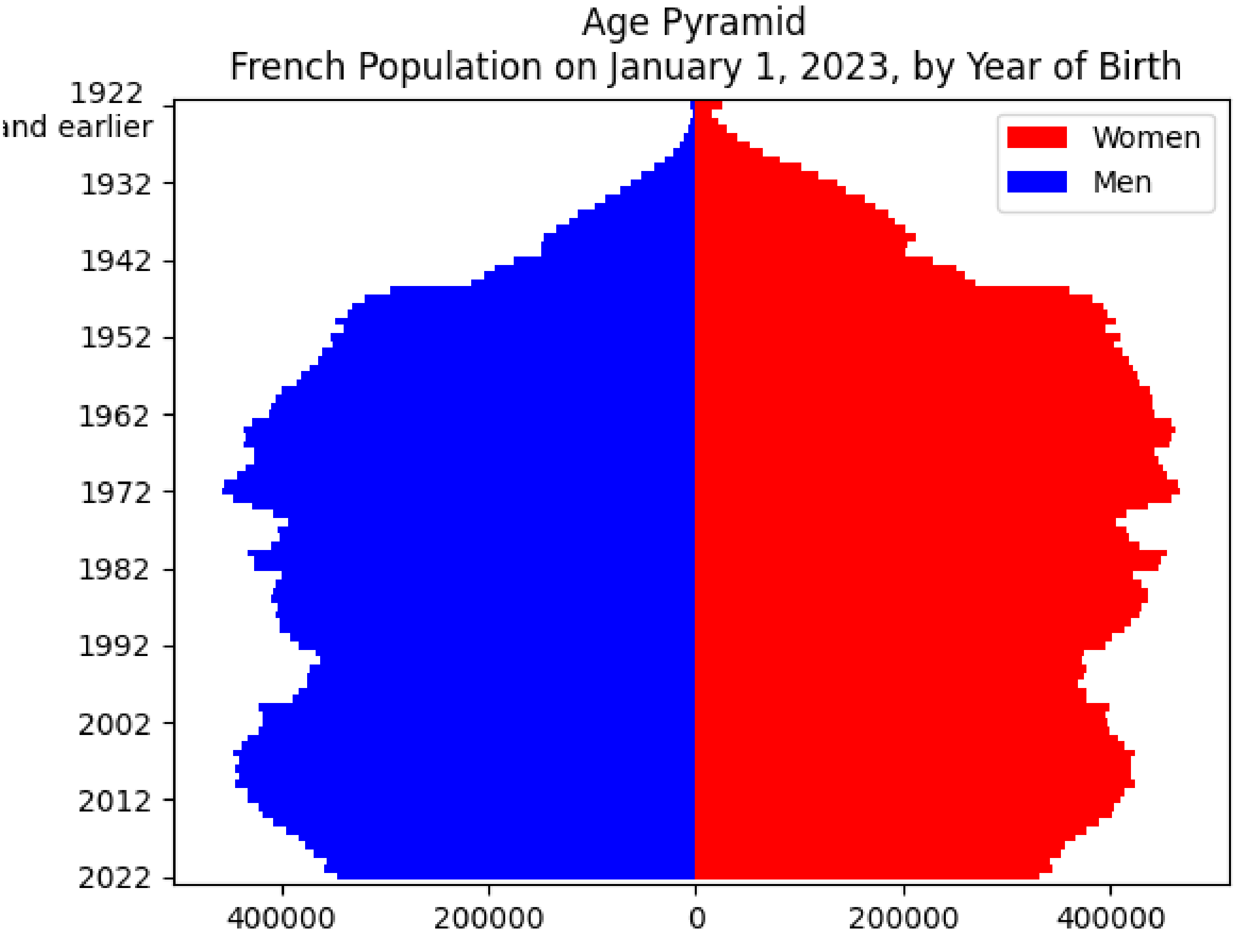

FIGURE 3 – Exercise 3

Q1 - Load the data from the file `pyramide2022.xlsx` into a Pandas DataFrame.

Q2 - Rename the columns so that their labels consist of a single word.

Q3 - Change the content of the cell "1922 et avant" to "1922" to use all dates consistently.

Q4 - Using `barh`, create a diagram identical to the one in Figure 1. Hint : To plot the men's diagrams, negative values will be plotted. However, x-axis labels are displayed with positive values.

Q5 - Adjust the y-axis labels to display "1922 et avant".

[2] `https://www.insee.fr/fr/statistiques/fichier/1892088/pop-totale-france-metro.xls`

Exercice 4 : Advanced Age Pyramid

The goal is to create a graph similar to the one shown in Figure 4, still using the `pyramide2022.xslx` file.

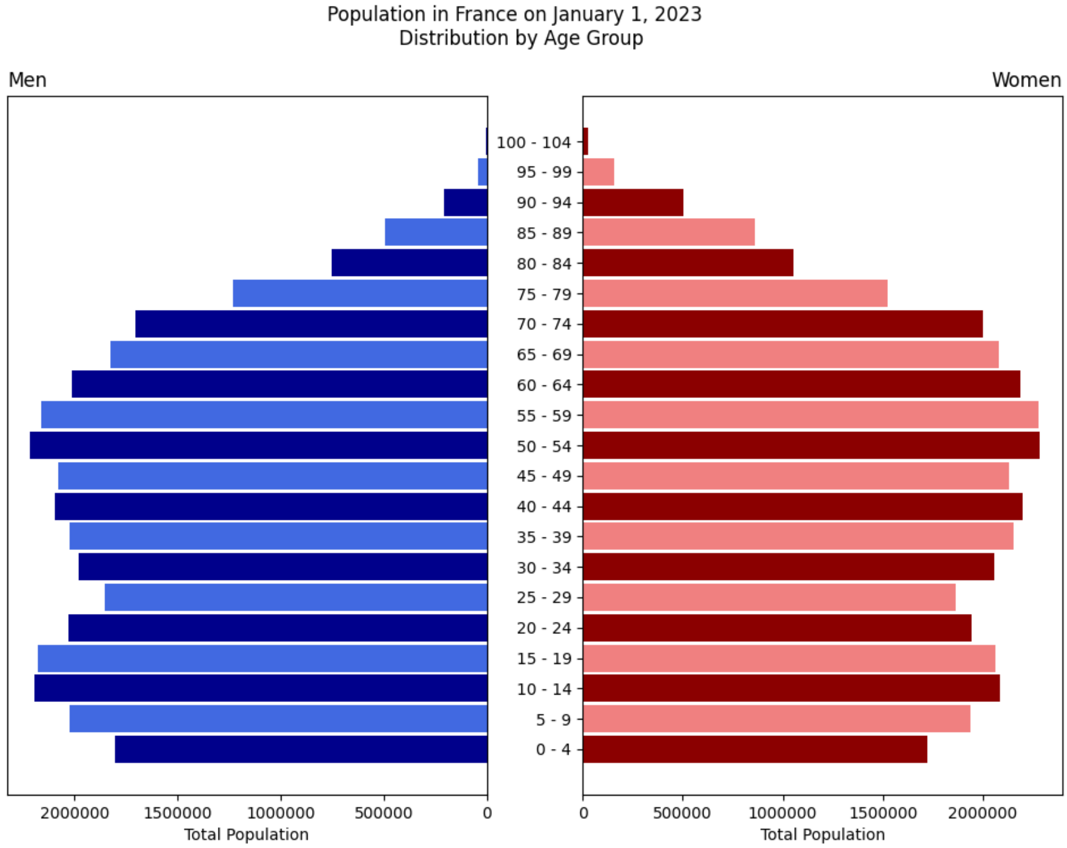

FIGURE 4 – Exercise 4

Q1 - Load the data from the file `pyramide.xslx` into a Pandas DataFrame `df1`.

Q2 - Rename the columns so that their labels consist of a single word.

Q3 - Create a DataFrame `ages` associating each age `A` with its age range `P`. Here is an excerpt from the created DataFrame :

```
   A        P
0  0    0 - 4
1  1    0 - 4
2  2    0 - 4
3  3    0 - 4
.. ...  ...
```

Note : It is considered that people over 100 years old all belong to the age range 100-104.

Q4 - Merge this age range DataFrame with `df1` to have access to the age range information for each row.

```
   Year Age   Women   Men    Total  A       P
0  2022  0   330929  345538  676467  0    0 - 4
1  2021  1   343026  357351  700377  1    0 - 4
2  2020  2   341042  356678  697720  2    0 - 4
```

```
3  2019  3  351707  368670  720377  3    0 - 4
4  2018  4  356873  376705  733578  4    0 - 4
```

Q5 - Using `groupby`, calculate the total number of men and women for each age range.

```
         Women   Men    Year
P
0 - 4    1723577  1804942  2020.0
5 - 9    1939119  2024509  2015.0
10 - 14  2087511  2196168  2010.0
15 - 19  2062755  2181207  2005.0
20 - 24  1944044  2030202  2000.0
```

Q6 - Create a diagram identical to the one in Figure 2.

Here are some useful details :

- The overall figure has a size of 12*8.
- The figure consists of 2 side-by-side graphs.
- The space between the two figures is 0.2.
- The colors used are 'darkblue' and 'royalblue' for men ; 'darkred' and 'lightcoral' for women.
- The bars have a height of 0.9.