**Python for Data Analysis.**
ESILV S1 2021–2022
[Tutorial 2: NumPy and Panadas]

## NumPy

### Exercise 1 [NumPy initialization arrays]

1. Use this line of code to create a NumPy array. Display its dimension and type.
   *np.array([[1,2,3],[4,5,6],[7,8,9]])*

2. Create a 1-dimensional array of zeros

3. Create a 2-dimensional array of zeros and visualize its length using different functions. See its size, dimensions and types.

4. What are the "*zeros_like*" and "*empty*" functions for? Explain.

5. What are the *'linespace'* and *'arange'* functions for? Explain.

6. Use the "*random*" function to create a 1D and 2Darray array. Check their dimensions using the functions seen previously.

### Exercise 2 [Mathematic operations]

1. Create a NumPy array, representing the sinus of another array. Do the same for cosines and logarithm. Save each result in a variable.
2. Use sinus and cosine array, to compute those functions:
   $$f(x) = sinx \text{ x } cosx$$
   $$f(x) = cosx^2 - sinx^2$$
   How about this one:
   $$sin(x)^2 + cos(x)^2$$
   What are your remarks concerning the last one?

3. Add a constant *C = 1.5 to* every element of the sinus array.

4. Explore the *'dot'* and *'@'* functions. What are they used for?

### Exercise 3 [I/O operations using NumPy]

1. Create a random multi-dimensional array (or use one that has already been created) and save it as a text file. Now, save this array as a NumPy object (.npy extension)
2. Read those files again using *open()* function.

3. Try to read it again, using NumPy function *'loadtxt'*. Explain the difference between them.

**Exercise 4 [Use Case, Mona Lisa]**

*Loading Mona Lisa data*

1. Load 'monalisa.txt' file as a NumPy array and explore it. (Dimensions, values, types, etc.)

2. Show the portrait of the Mona Lisa. To do this import matplotlib using this command:
   *import matplotlib.pyplot as plt*
   Then use *'imshow'* command to display it:
   *plt.imshow(monalisa)*

3. Use the 'cmap' parameter to display it in black & white.

4. Load 'monalisa.npy' file as a NumPy array, display it and explore it. (Dimensions, values, types, etc.) Why do you think the second photo is better?

*Slicing – Using monalisa.npy*

5. Crop the image to show only Mona Lisa's head

6. Pixelize Mona Lisa's image on 3 different levels, and explain the result

7. Rotate the Mona Lisa 90°

8. Display the reflection of the Mona Lisa

9. Hide Mona Lisa's face using a white box. Replace the white box with a randomly generated box

*Transformation - Use the monlisa.txt file*

10. Create a filter on the image to have a darker Mona Lisa

11. Create a vertical gradient from right to left allowing the image to be darker on the left

12. Create a horizontal gradient that will make the Mona Lisa darker on top

## Pandas

### Introduction [Traffic accidents database]

https://www.data.gouv.fr/fr/datasets/base-de-donnees-accidents-corporels-de-la-circulation/

For each personal injury accident, information describing the accident is entered by the law enforcement unit that intervened at the scene of the accident. These seizures are brought together in a sheet called the personal accident analysis report. All of these files constitute the national file of traffic accidents known as the "BAAC File" administered by the National Intermenstrual Road Safety Observatory "ONISR".

The databases, extracted from the BAAC file, list all of the bodily injury accidents occurring during a specific year in mainland France, in the overseas departments (Guadeloupe, Guyana, Martinique, Réunion and Mayotte since 2012) and in the other overseas territories (Saint-Pierre-et-Miquelon, Saint-Barthélemy, Saint-Martin, Wallis-et-Futuna, French Polynesia and New Caledonia; available only from 2019 in open data) with a simplified description. This includes accident location information, as entered as well as information regarding the characteristics of the accident and its location, the vehicles involved and their victims.

The databases from 2005 to 2019 are now annual and made up of 4 files (Characteristics - Locations - Vehicles - Users) in csv format.

Links to databases:

- https://www.data.gouv.fr/fr/datasets/r/be2191a6-a7cd-446f-a9fc-8d698688eb9e
- https://www.data.gouv.fr/fr/datasets/r/e4c6f4fe-7c68-4a1d-9bb6-b0f1f5d45526
- https://www.data.gouv.fr/fr/datasets/r/08b77510-39c4-4761-bf02-19457264790f
- https://www.data.gouv.fr/fr/datasets/r/96aadc9f-0b55-4e9a-a70e-c627ed97e6f7

1. We'll use vehicles database. Import the data using pandas library.
   ```
   pd.read_csv(url)
   ```

### Exercise 1 [Explore pandas dataframe basic functions]

1. How many columns do we have in the dataframe? How many rows do we have?
2. What are the types of columns in the dataframe? Use *info()* function
3. Calculate the mean of numeric columns
4. Calculate the maximum and minimum value of numeric column
5. Use *describe()* function. What can you observe?
6. Display the first 30 values, then last 30.

**Exercise 2 [Explore traffic accidents vehicles]**

Read the documentation to understand the meanings of each column
Documentation: https://www.data.gouv.fr/fr/datasets/r/6cade01c-f69d-4779-b0a4-20606069888f

1. Then display the column names and rename it with the names of your choice. For the following questions, we use these column names:

   *[*
   *"Num_Acc", "sens_de_Circulation", "catV",*
   *"nb_occupants", "obstacle_fixe", "obstacle_mobile",*
   *"choc", "manoeuvre" , "num_vehicule"*
   *]*

2. Get the number of null values by column. The, display its proportion in the dataframe. For example: 0.08% of col_1 are null

*Exploratory data analysis - Filters*

3. Select 'obstacle_fixe' column. What is the type of returned value?

4. Convert the returned value to a NumPy array

5. Compute the maximum value of the NumPy array

6. Select 'catV' & 'sens_de_Circulation' columns together

7. Select all the columns containing ‘obstacle’ in their name

8. Select all rows with null values for 'sens_de_Circulation' column

9. Sort the data frame using the accident number column

10. Sort the values using the accident number column, then the vehicle category column

11. Select the 4153rd row. Then select row having index '4153'. What is the difference between both?

12. Set Num_Acc as index
    a. Make sure to use *'inplace=True'* to save the value in the dataframe
    b. Can you still get the row having index '4153'? Explain.
    c. Finally, Restore the index as it was before.

*Exploratory data analysis - Queries*

13. Select accident that have a vehicle number W23

14. Select the top 5 accidents that caused the most damage

15. Count the number of damaged vehicles, by vehicle category and number of occupants

16. From the previous result, select category vehicle '40', having '140' occupants

17. Calculate the number of occupants by vehicle category

18. Compute the number of damages per vehicle category

19. Select all the lines with a category vehicle between 37 and 40 inclusive. (These values correspond to public transport)

20. Create a new column with a Boolean value, showing whether the accident happened in public transport or not

21. Check that the new column is working well. For this, it is necessary to verify that the number of occupants in vehicles excluding public transport is equal to 0. Contrary to what we can see in the data in the other group.

22. Create a new dataframe, which contains only public transport accidents

23. Obtain the average number of occupants by vehicle category, and sort the result

24. Map the vehicle category, obstacles (fixed and mobile) using the dictionary defined in mapping.py (you can use the mapping.py file, but you can also create your own mappings).

*Heat map – vehicle category VS accident moving obstacle*

25. Create a new database, projecting the category of the vehicle in relation to the moving obstacle. The values of the data frame will be the sum of the respective occupant numbers. This database can be referred to as a heat map of the number of accidents by type of transport and obstacle.

26. Using the previous results, obtain the most dangerous mode of transport for the public

27. Sort the heatmap from the most dangerous to the least dangerous means of transport

28. Get the most dangerous moving obstacle when using public transportation

29. Sort the heat map from the most dangerous to the least dangerous entity (means of transport, and moving obstacle)