Training Piscine Python for datascience - 2

DataTable

Summary: Today, you will learn how to load, manipulate and display datatable

Version: 1.00

Contents

Chapter I

General rules

- You have to render your modules from a computer in the cluster either using a virtual machine:
  - You can choose the operating system to use for your virtual machine
  - Your virtual machine must have all the necessary software to realize your project. This software must be configured and installed.
- Or you can use the computer directly in case the tools are available.
  - Make sure you have the space on your session to install what you need for all the modules (use the goinfre if your campus has it)
  - You must have everything installed before the evaluations
- Your functions should not quit unexpectedly (segmentation fault, bus error, double free, etc) apart from undefined behaviors. If this happens, your project will be considered non functional and will receive a `0` during the evaluation.
- We encourage you to create test programs for your project even though this work **won't have to be submitted and won't be graded**. It will give you a chance to easily test your work and your peers' work. You will find those tests especially useful during your defence. Indeed, during defence, you are free to use your tests and/or the tests of the peer you are evaluating.
- Submit your work to your assigned git repository. Only the work in the git repository will be graded. If Deepthought is assigned to grade your work, it will be done after your peer-evaluations. If an error happens in any section of your work during Deepthought's grading, the evaluation will stop.
- You must use the Python 3.10 version
- Your lib imports must be explicit, for example you must "import numpy as np". Importing "from pandas import *" is not allowed, and you will get 0 on the exercise.
- There is no global variable.
- By Odin, by Thor ! Use your brain !!!

Chapter II

Specific instructions of the day

- No code in the global scope. Use functions!
- Each program must have its main and not be a simple script:

```
def main():
  # your tests and your error handling

if __name__ == "__main__":
  main()
```

- Any exception not caught will invalidate the exercices, even in the event of an error that you were asked to test.
- You can use any built-in function if it is not prohibited in the exercise.
- All your functions must have a documentation (__doc__)
- Your code must be at the norm
    - pip install flake8
    - alias norminette=flake8

For this module, we will use data from FREE SCHOOL MATERIALS FROM GAPMINDER.ORG, CC-BY LICENSE.

We encourage you to have a look at the available data if you want to train yourself to manipulate data or do data vision.

Chapter III

Exercise 00

	Exercise 00
Exercice 00: Load my Dataset	
Turn-in directory : *ex*00/	
Files to turn in : `load_csv.py`	
Allowed functions : `pandas or any lib for data set manipulation`	

Make a function that takes a path as argument, writes the dimensions of the data set and returns it. You have to handle the error cases and return None if the path is bad, bad format...

```
def load(path: str) -> Dataset: (You have to adapt the type of return according to your library)
        #your code here
```

Your script tester:

```
from load_csv import load

print(load("life_expectancy_years.csv"))
```

```
$> python tester.py
Loading dataset of dimensions (195, 302)
country 1800  1801  1802  1803  ...  2096  2097  2098  2099  2100
Afghanistan 28.2 28.2  28.2  28.2  ...  76.2  76.4  76.5  76.6  76.8
   ...
$>
```

You can display the Dataset in any format you like, the given format is not restrictive.

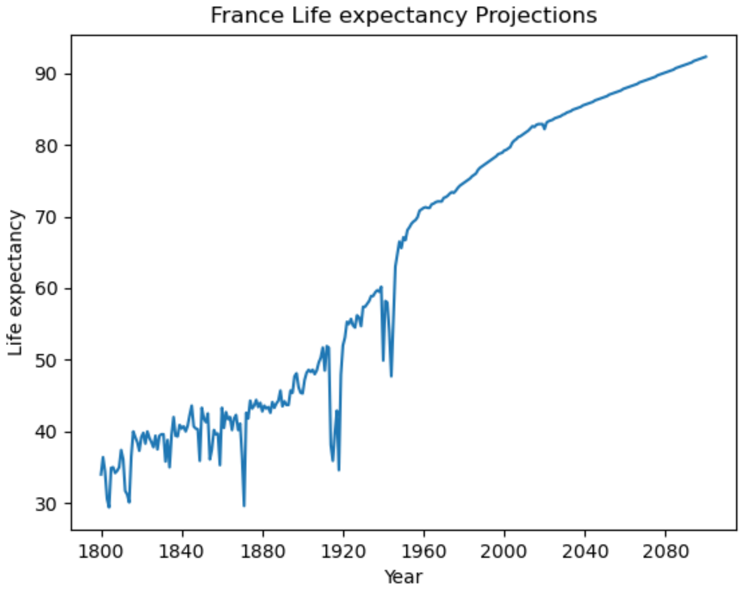
Chapter IV

Exercise 01

	Exercise 01
Exercice 01: draw my country	
Turn-in directory : *ex*01/	
Files to turn in : `load_csv.py, aff_life.py`	
Allowed functions : `matplotlib, seaborn or any lib for Data Visualization`	

Create a program that calls the load function from the previous exercise, loads the file life_expectancy_years.csv, and displays the country information of your campus. Your graph must have a title and a legend for each axis.

For `example`, for the 42 campuses in France we will have this result.

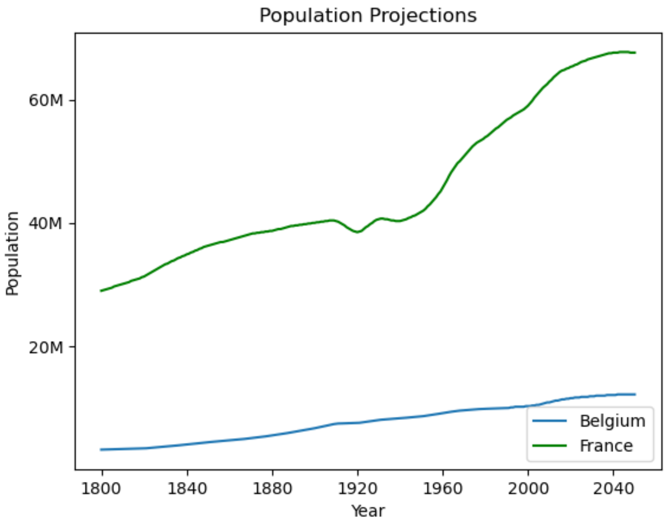


Chapter V

Exercise 02

	Exercise 02
Exercice 02: compare my country	
Turn-in directory : *ex*02/	
Files to turn in : `load_csv.py, aff_pop.py`	
Allowed functions : `matplotlib, seaborn or any lib for Data Visualization`	

Create a program that calls the load function from the first exercise, loads the file population_total.csv, and displays the country information of your campus versus other country of `your choice`. Your graph must have a title, a legend for each axis and a legend for each graph.
You must display the years from 1800 to 2050.
For `example`, for the 42 campuses in France we will have this result.



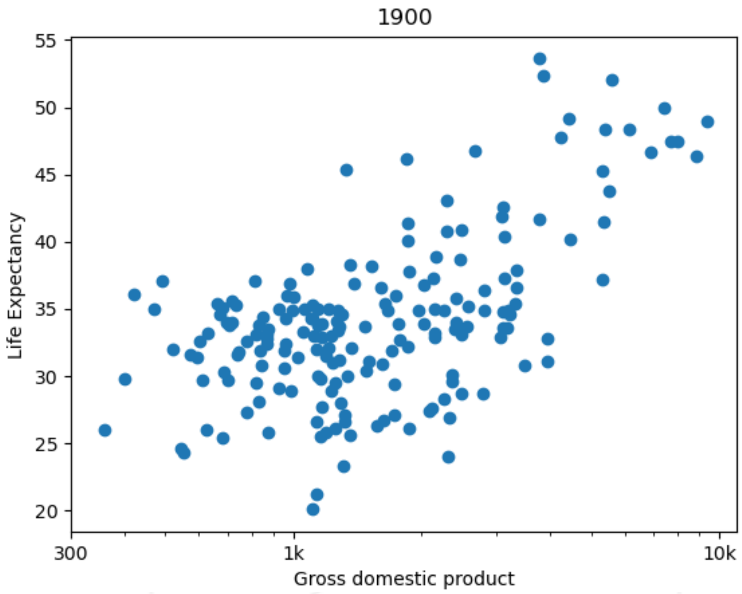
Chapter VI

Exercise 03

	Exercise 03
Exercice 03: draw my year	
Turn-in directory : *ex*03/	
Files to turn in : `load_csv.py, projection_life.py`	
Allowed functions : `matplotlib, seaborn or any lib for Data Visualization and your lib of ex00`	

Create a program that calls the load function from the first exercise, loads the files "income_per_person_gdppercapita_ppp_inflation_adjusted.csv" and "life_expectancy_years.csv", and displays the projection of life expectancy in relation to the gross national product of the year 1900 for each country.

Your graph must have a title, a legend for each axis and a legend for each graph. You must display the year 1900.



Do you see a correlation between life span and gross domestic product?

Chapter VII

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

> The evaluation process will happen on the computer of the evaluated group.