TP4 (Python Version)

INF8808 : Data Visualisation

Department of computer and software engineering

Author : Olivia Gélinas

Teacher's Assistant : Hellen Vasques

Objectives

The goal of this lab is to create an interactive bubble chart using open data in JSON format.

Before beginning, we recommend you have completed the following readings and practice exercises :

Readings :	• Bubble Charts https://plotly.com/python/bubble-charts/ • Intro to Animations https://plotly.com/python/animations/ • Advanced Callbacks https://dash.plotly.com/advanced-callbacks
Exercices :	TP4 exercices : 1, 2, 3, 4

Introduction

A bubble chart is a type of graphic that is used to visualize data possessing many parameters and where each point is represented by a circle. It's a combination of a scatter plot and a proportional area chart. With this type of graphic, we can encode a maximum of four dimensions: the position in x, position in y, color and area of the circles.

In this lab, you will create an interactive bubble chart using data about the CO_2 emissions per inhabitant and revenue per inhabitant, provided by the open data portal of the World Bank [1]. You will have access to data

from years 2000 and 2015.

Before completing the code for the bubble chart, you will also implement a simplified version of it using a Jupyter notebook. The source for the notebook has been provided to you. You can run the notebook with Jupyter and complete it with your own code. This notebook symbolizes an important initial step in the creation of most data visualizations, which is the exploration of the data and of design possibilities.

Description

In this lab, you will have to complete the Python code using Plotly and Dash in order to display a bubble chart representing the relationship between GDP and CO2 emissions. As a preliminary step, you will implement a simple interactive app using Dash in a Plotly notebook. Afterwards, you will reuse and complete your notebook's code using the code in the archive provided with this lab.

In the bubble chart, the color of each bubble represents the continent, and their size represents the population. To make the bubble chart interactive, a button and a slider toggle between views of the data from the years 2000 and 2015. When the data is updated using the 'Play' button, the circles move to their new positions gradually with an animation.

The following subsections present the different parts that you will have to complete for this lab. We recommend completing the data exploration in the Jupyter notebook first. Then, this code will help you to complete the implementation of the data preprocessing, followed by the animated bubble chart. We recommend you implement the code for the tooltip as the last step.

File Structure

To complete this work, you will need to fill the various "TODO" sections in the files from the archive provided for the lab. The comments in the code explain in more detail the steps to take. The scripts to use are located in the "assets" directory of the archive provided for the lab.

In this lab, we provide you with and archive with 7 Python files used to accomplish the desired visualization :

- `app.py` : This file generates the HTML structure of the webpage and orchestrates the steps required to create the visualization. You do not need to modify it.

- `bubble.py`

- `hover_template.py`

- `preprocess.py`

- `server.py` : This file is used to launch the application. You do not need to modify it.

Dataset

The dataset is located in the `src/assets/data/` directory in the archive provided for the lab. The dataset contains a table of objects for years 2000 and 2015. Each object in the tables contains the following keys :

- `Country Name` : The name of the country.

- GDP : The GDP per capita in current U.S. dollars.

- CO2 : The CO2 emissions per capita in metric tonnes.

- Population : The population of the country.

- Continent : The continent of the country.

Data exploration

For this part, use the Jupyter notebook provided with this lab.

In the notebook, you will implement the code to create scatter plots for the data from 2000 and 2015. These plots will be displayed in a Dash interactive application. The specific guidelines regarding the scatter plots are contained in the Jupyter notebook. As you will see, the scatter plots are interactive : when a marker is hovered in one of the two plots, information about the given country for both years is displayed in the table at the bottom of the page.

For further indications, Figure 1 below displays both scatter plots and the application's behavior when a marker is hovered.

Figure 1 : The scatter plots resulting from the data exploration. The marker for the country Trinidad and Tobago in 2015 is hovered by the cursor.

Data preprocessing

For this second part, you will have to modify the code provided in the archive for the lab. To begin, you will have to preprocess the data we provide you. The data contained in the JSON file is raw, so it is necessary to reorganize certain parts of it so they can be properly used by the Plotly library. To do so, you need to complete the file preprocess.py.

The specific steps are :

1. Round the numbers in the data down so they have less decimal places for the display (function `round_decimals`). You can reuse part of your code from your notebook for this step.

2. Get the range of possible values for the x and y axes (function `get_range`). You can reuse part of your code from your notebook for this step.

3. Combine the data into a structure more easily readable by Plotly (function `combine_dfs`)

4. Sort the data by year and by continent for the display (function `sort_dy_by_yr_continent`)

Animated bubble chart

For this third part, the code to modify may be found in the archive for the lab. More specifically, it is in the file `bubble.py`. This part consists of generating the main part of the data visualization. For this part, you will first generate the bubble chart with an animation across the available years. The x and y axes are log scaled. The x and y axes represent the GDP and CO2 emissions per capita, respectively. The color of each circle represents the continent of the data, while its size represents the population of the data. After the bubble chart follows these specifications, you will refine its animation by simplifying its menu and by making sure the hover template is set for each animation frame. Finally, you will make some visual adjustments, such as setting the axes labels, the template and the legend for the figure.

To sum up, the steps for this part are as follows :

1. Draw the animated bubble chart (function `get_plot`). You can reuse part of your code from your notebook for this step.

2. Update the hover template so it appears on all frames (function `update_animation_hover_template`)

3. Simplify the display of the animation menu (function `update_animation_menu`)

4. Update some visual elements of the figure, such as the axes labels, the template, and the legend (functions `update_axes_labels`, `update_template`, and `update_legend`)
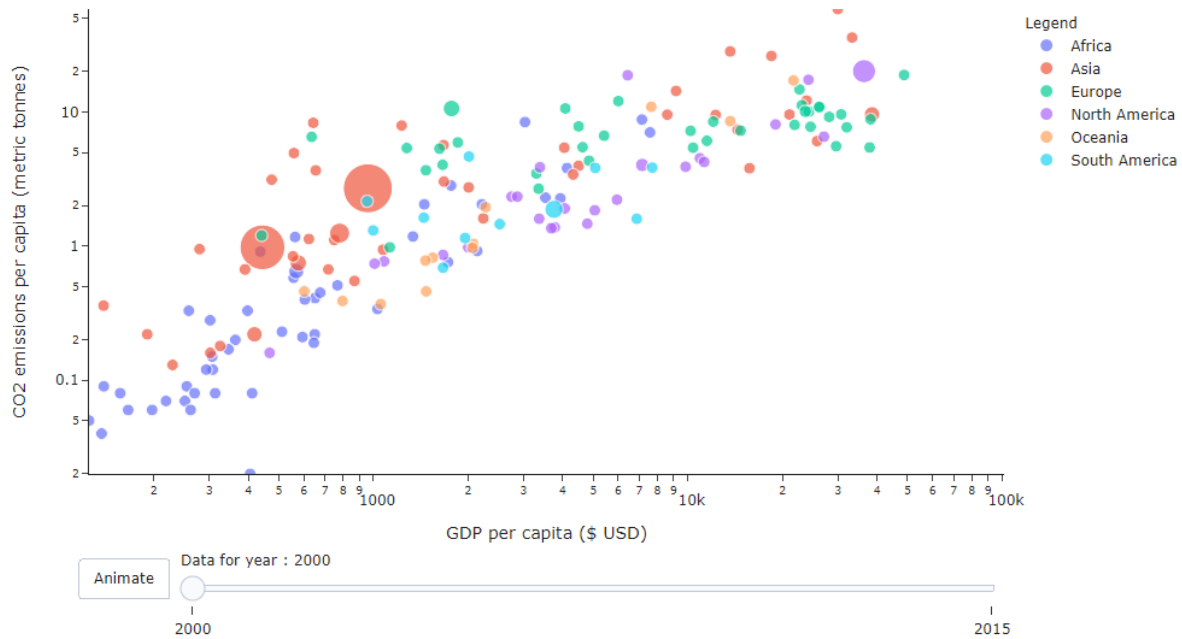
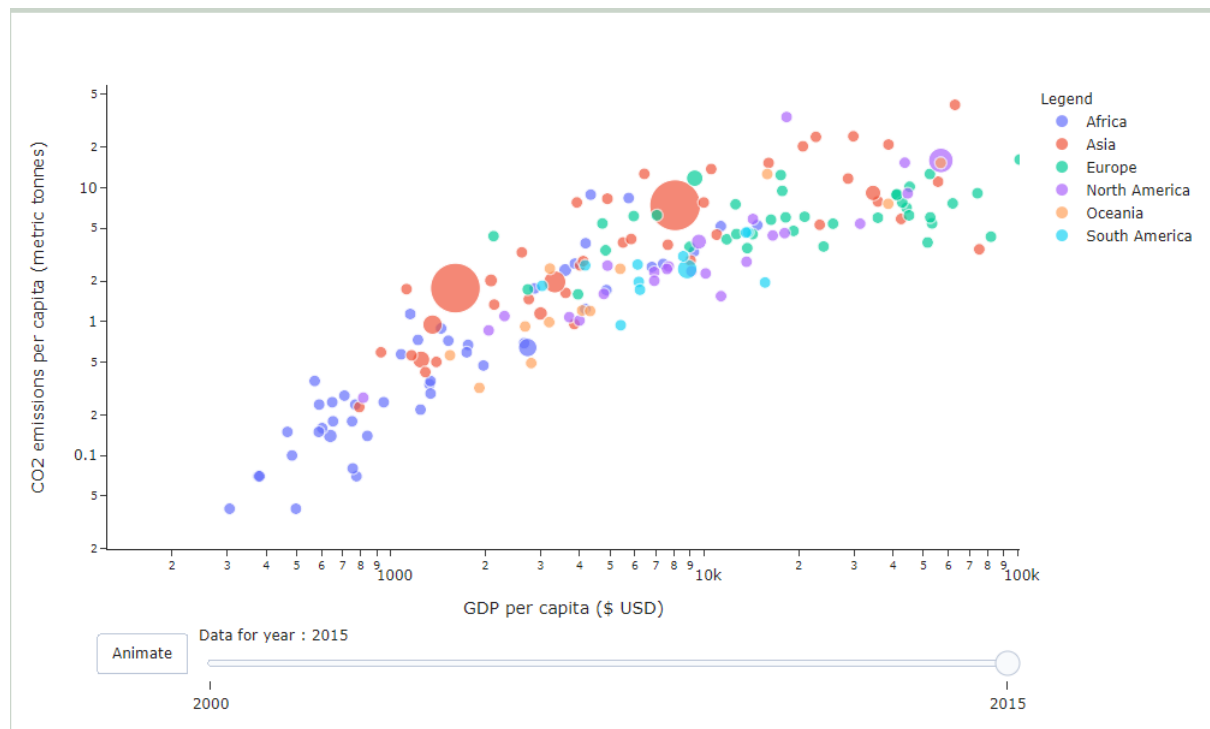*Figure 2 : The bubble chart with data from 2000*

Figure 3 : The bubble chart with data from 2015

Tooltip

For this fourth part, you will define a template used to display the tooltip that appears when the cursor hovers over a bubble. The tooltip should contain the country's name, population, GDP per capita, and CO2 emissions per capita, in that order. Each information should be preceded by a corresponding label and followed by the units of measurement where appropriate. The entirety of the code for this section can be written in the function `get_bubble_hover_template` in the file `hover_template.py`. Make sure you also handled the hove template correctly when you created the bubble chart.

See Figure 4 for the expected final result.
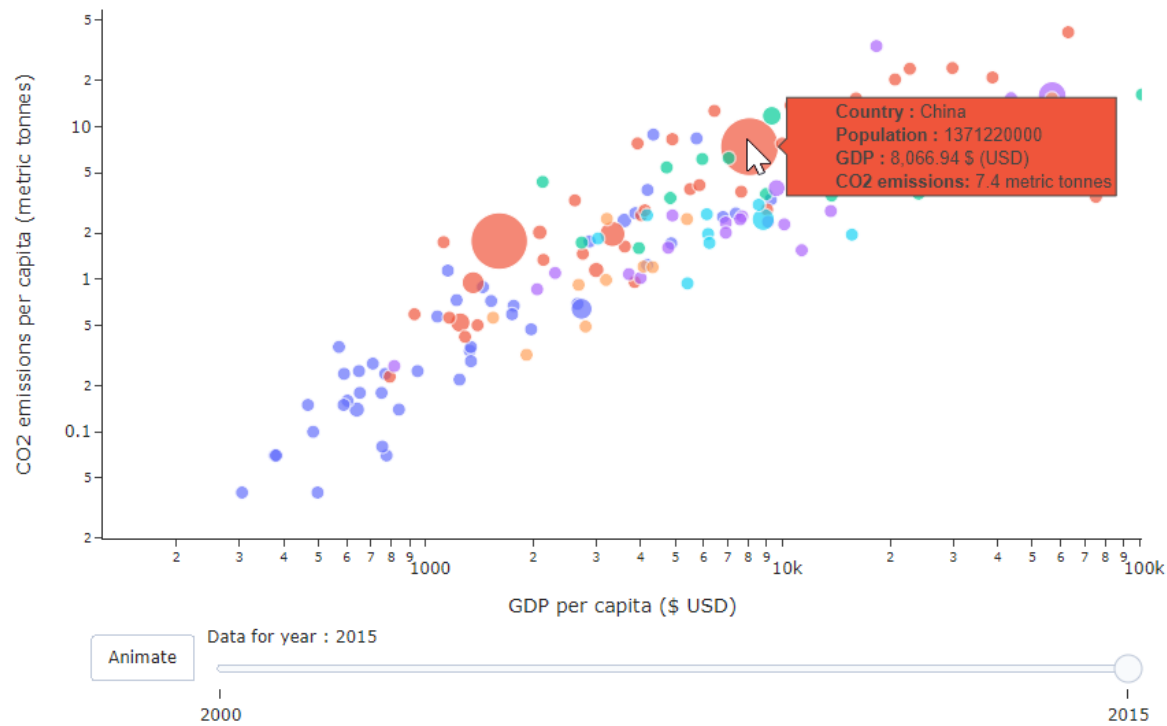
The bubble chart for 2015 with the bubble for China hovered

Submission

The instructions for the submission are :

1. You must place your project code in a compressed ZIP file named
matricule1_matricule2_matricule3_matricule4.zip

2. The lab must be submitted before **June 5th 23h59**

Evaluation

Overall, your work will be evaluated according to the following grid. Each section will be evaluated on correctness and quality of the work.

Requirement	Points
Data exploration	7
Data preprocessing	3
Animated bubble chart	7
Tooltip	2
Overall quality and clarity of the submission	1
Total	**20**

References

[1] The World Bank, "DataBank ," The World Bank. Available: https://databank.worldbank.org/home [Accessed 01 09 2020].