# TP3 (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 heatmap and line chart dashboard using open data in CSV format.

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

**Readings :**

- Plotly Express

  > https://plotly.com/python/plotly-express/

- Heatmaps

  > https://plotly.com/python/heatmaps/

- Line charts

  > https://plotly.com/python/line-charts/

**Exercices :** TP3 exercices : 1, 2, 3, 4, 5

# Introduction

A heatmap is a type of data visualization which shows values depicted as colors in two dimensions. The hue or intensity of the color represents the magnitude of the observation for its corresponding values in 2 dimensions.

In this lab, you will implement an interactive heatmap using data representing tree plantations over time in Montreal neighborhoods. The data was extracted from the City of Montreal's open data portal [1]. The dataset

was then modified into a version to be used in this lab. It contains data representing the date and location that public trees were planted in Montreal.

To explore the use of dashboards, we will also program a line chart to accompany the heatmap. A line chart is a type of graphic where data is represented by one or by many lines. It is extremely common and can be found in many fields where we would like to see the evolution of a parameter as a function of another. In this lab, the data presented on the line chart will depend on the heatmap's current selection.

# Description

In this lab, you will have to complete the Python code using Plotly and Dash in order to display a heatmap with an accompanying line chart to represent tree plantations in Montreal neighborhoods throughout time.

The heatmap represents how many trees were planted in each Montreal neighborhood each year between 2010 and 2020. The data displayed on the line chart depends on which cell is selected in the heatmap. When a cell on the heatmap is selected, the line chart displays tree plantations over time for the selected neighborhood and year. When no cell is selected in the heatmap or when there is no data to display for the current selection, the area where the line chart resides contains an instructional message.

The following subsections present the different parts that you will have to complete for this lab. While you code, we recommend completing the data processing first, followed by the implementation of the heatmap itself. Next, you should implement the line chart. The next two parts, the theme and the tooltip, are independent of each other.

## 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.
- `heatmap.py`
- `hover_template.py`
- `line_chart.py`
- `preprocess.py`
- `server.py` : This file is used to launch the application. You do not need to modify it.
- `template.py`

## Dataset

The dataset is located in the `src/assets/data/` directory in the archive provided for the lab. The dataset contains the following columns :

- `Arrond` : The ID of the neighborhood where the tree was planted.
- `Arrond_Nom` : The name of the neighborhood where the tree was planted.
- `Date_Plantation` : The date the tree was planted.
- `Longitude` : The longitude of the tree.
- `Latitude` : The latitude of the tree.

## Data preprocessing

To begin, you will have to preprocess the data we provide you. The data contained in the CSV 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`.

More precisely, you will have to complete these steps :

1. Convert the dates in the data to a format understandable by Plotly (function `convert_dates`)
2. Filter the data by year (function `filter_years`). Though we have data from trees planted starting in the early 20th century, we want to focus on tree plantations between 2010 and 2020 for this visualization.
3. Summarize the data to get the total count of trees planted per year per neighborhood (function `summarize_yearly_counts`)
4. Restructure the data into a format that Plotly can easily read to generate the heatmap (function `restructure_df`)
5. Write a function to easily get data to display on the line chart, corresponding to the daily amount of planted trees in a given neighborhood and year (function `get_daily_info`)

To help guide you in these steps, Figures 1 and 2 show an extract of the data after the preprocessing.

Figure 1 illustrates data in a format that can be displayed in the heatmap, while Figure 2 samples some data that can be displayed in the line chart.

| Date_Plantation | 2010-12-31 | 2011-12-31 | 2012-12-31 | 2013-12-31 | ... | 2017-12-31 | 2018-12-31 | 2019-12-31 | 2020-12-31 |
|---|---|---|---|---|---|---|---|---|---|
| Arrond_Nom | | | | | ... | | | | |
| Ahuntsic - Cartierville | 437.0 | 289.0 | 327.0 | 357.0 | ... | 936.0 | 815.0 | 851.0 | 144.0 |
| Côte-des-Neiges - Notre-Dame-de-Grâce | 401.0 | 108.0 | 458.0 | 395.0 | ... | 803.0 | 623.0 | 685.0 | 79.0 |
| LaSalle | 1.0 | 147.0 | 522.0 | 0.0 | ... | 336.0 | 48.0 | 1175.0 | 0.0 |

*Figure 1 : Sample of heatmap data*

| | Date_Plantation | Counts |
|---|---|---|
| 0 | 2017-05-16 | 4 |
| 1 | 2017-05-17 | 23 |
| 2 | 2017-05-18 | 2 |
| 3 | 2017-05-19 | 0 |
| 4 | 2017-05-20 | 0 |

*Figure 2 : Sample of line chart data for the Sud-Ouest in 2017*

# Heatmap

For this second part, you will have to implement the main part of the data visualization. You will have to display the data you preprocessed in the previous step in the heatmap. To complete this part, you will need to modify the file `heatmap.py`.

This is the main step you will have to accomplish for this part :

1. Generate the heatmap from the preprocessed data and display it (function `get_figure`)

In Figure 3 we can see what the completed heatmap should look like.

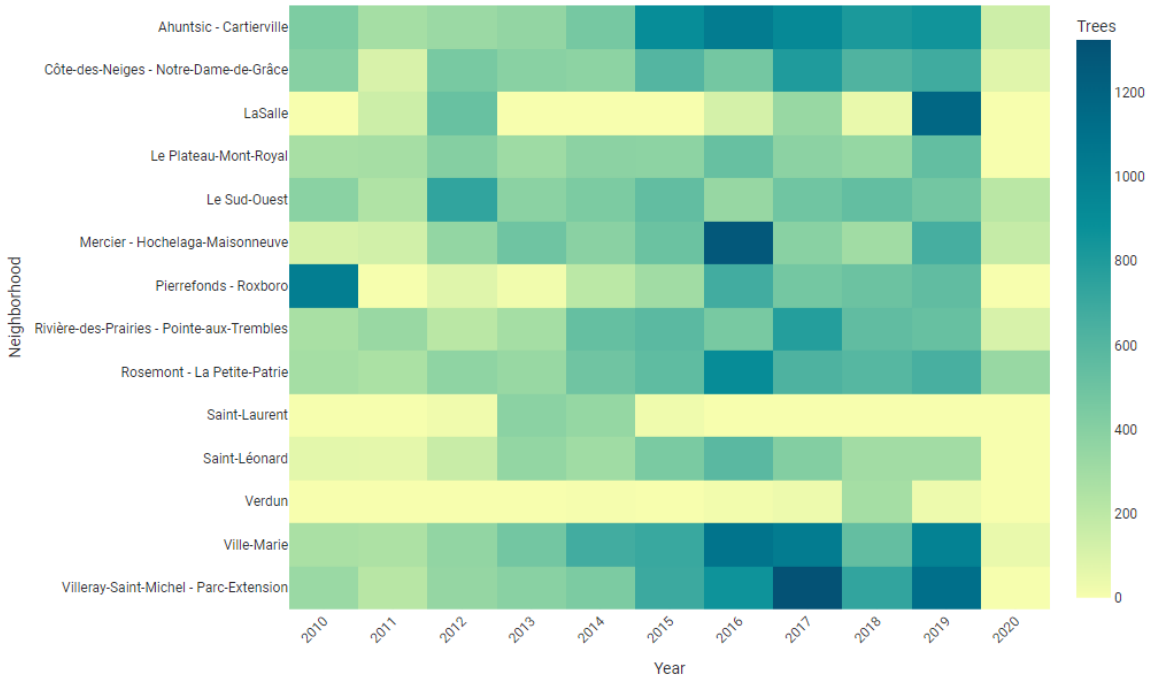


*Figure 3 : The heatmap*

# Line chart

For this third part, you will have to program the line chart which accompanies the heatmap. First, you will have to implement the empty version of the line chart, which is displayed when no cell has been selected or when the selected cell has no data to display. This empty version of the line chart should contain an informational message and a grey rectangle as a background. Second, you will implement the code to display data on the line chart when a cell is selected. When there is only data for one day to display, make sure to change the display mode of the line chart to be sure the single point is visible.

More precisely, you will have to complete these steps :

1. Write the function to display an empty figure when there is no data to display (function `get_empty_figure`)
2. Write the function to draw a grey rectangle on the empty figure when there is no data to display (function `add_rectangle_shape`)
3. When there is data to display, generate the line chart with the corresponding data for the given neighborhood and year (function `get_figure`)

The following figures give some further precisions on what is expected for this part. Figures 4, 5, and 6 are indicative of the final visual appearance and behavior of the graphics.

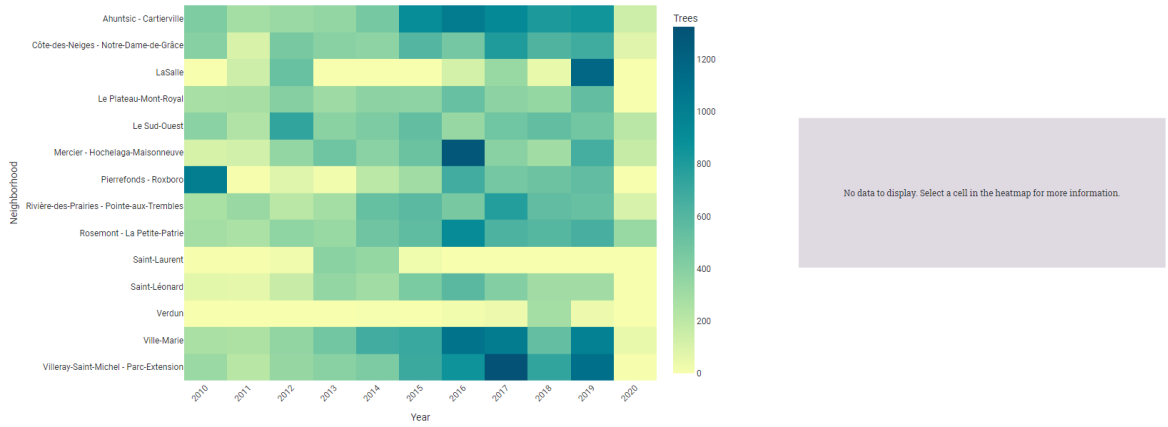


*Figure 4 : The heatmap with accompanying empty figure*

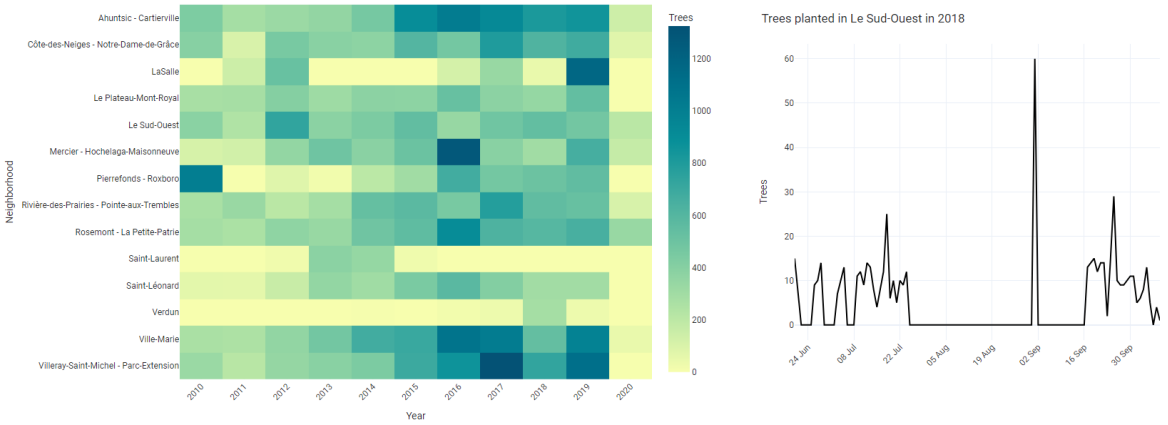


*Figure 5 : The heatmap with accompanying line chart representing daily trees planted in the Sud Ouest in 2017*

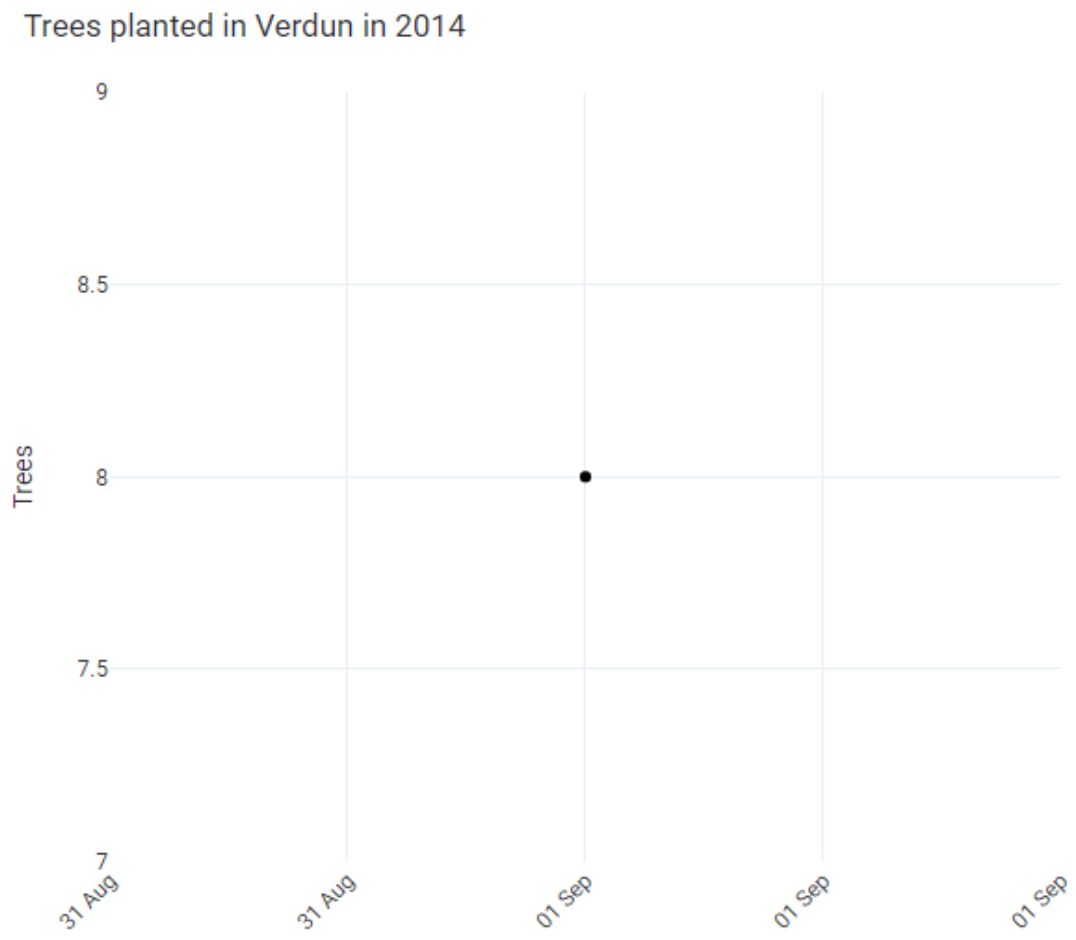


*Figure 6 : The line chart when there is only data for one day to display (here, the data for daily trees planted in Verdun in 2014)*

## Theme

For this fourth part, you will create a custom template to customize the visual display of the bar chart. To use it, you will set it as the default theme, applied on top of the `plotly_white` theme provided by Plotly. The code for this part is in the file `template.py`. You need to complete the functions `create_custom_theme` and `set_default_theme`. Make sure to closely follow the instructions in the comments of the code when defining each element of the theme. The desired values to use are defined in a variable at the beginning of the file. Make sure the colors in the heatmap and line chart resemble those in Figures 4, 5, and 6.

## Tooltip

For this fifth part, you will define a template used to display the tooltip that appears when the mouse hovers over a cell in the heatmap or over the line chart. In the heatmap, the tooltip for a cell should contain the name of the neighborhood, the year, and the amount of trees planted that year in the given neighborhood. In the line chart, the tooltip should contain the date and the amount of trees planted that day in the given neighborhood. The comments in the code give a detailed description of the contents and appearance of the tooltip. The code for this part should be written in the file `hover_template.py` in the function `get_hover_template`.

The steps you should follow to complete this part are as follows :

1. Complete the hover template for the heatmap (function `get_heatmap_hover_template`)
2. Complete the hover template for the line chart (function `get_linechart_hover_template`)

See Figures 7 and 8 for the expected final results.

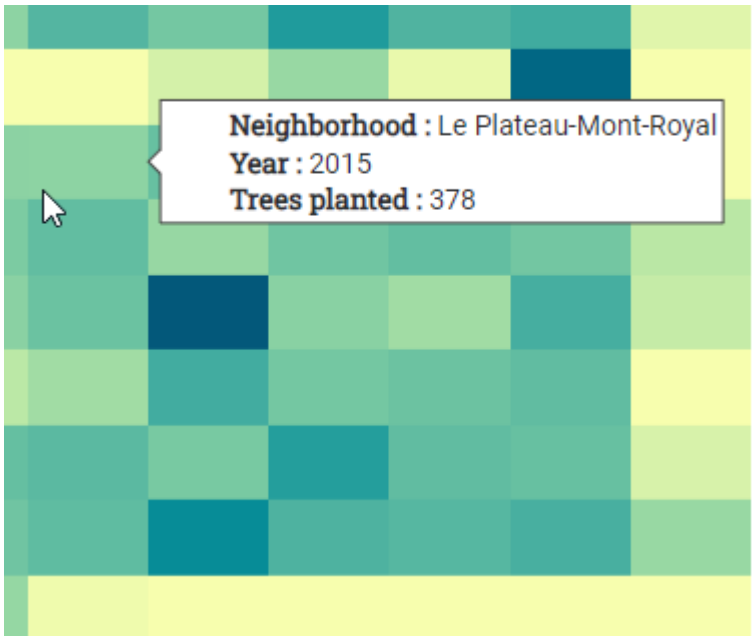


*Figure 7 : The tooltip for the heatmap for Plateau Mont-Royal in 2015*

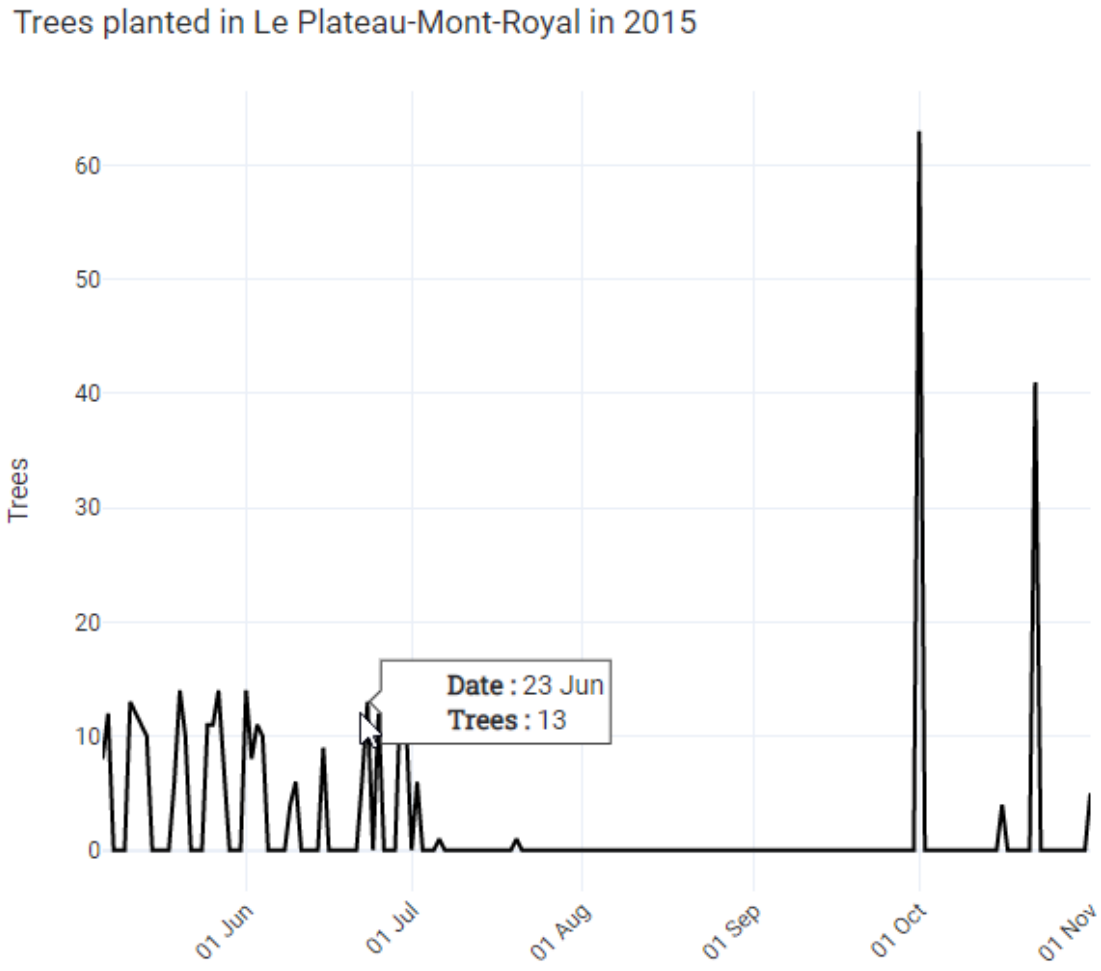


*Figure 8 : The tooltip for the line chart for Plateau Mont-Royal in 2015.*

# Submission

The instructions for the submission are :

1. You must place your project code in a compressed ZIP file named [studentID1_studentID2_studentID3.zip]

2. The lab must be submitted before [May 29th 23:59]

# 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 preprocessing | 5 |
| Heatmap | 3 |
| Line Chart | 6 |
| Theme | 2 |
| Tooltip | 3 |
| Overall quality and clarity of the submission | 1 |
| **Total** | **20** |

# References

[1] Service des grands parcs, du Mont-Royal et des sports, "Arbres publics sur le territoire de la Ville ," Montréal : Portail de données ouvertes. Available: http://donnees.ville.montreal.qc.ca/dataset/arbres [Accessed 01 09 2020].