TP5 (Python Version)

INF8808 : Data Visualisation

Department of computer and software engineering

POLYTECHNIQUE MONTRÉAL

Author : Olivia Gélinas

Teacher's Assistant : Hellen Vasques

Objectives

The goal of this lab is to create an interactive scatter map using open data in JSON and GeoJSON format.

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

Readings :	• Mapbox Choropleth Maps https://plotly.com/python/mapbox-county-choropleth/ • Scatter plots on Mapbox https://plotly.com/python/scattermapbox/
Exercices :	TP5 exercices : 1, 2, 3, 4

Introduction

A scatter map is similar to a typical scatter plot, except the markers are positioned with respect to their geographic location on a map base. This type of chart is useful when the geographic context is important to our interpretation of the data.

In this lab, you will implement an interactive scatter map using data representing pedestrian paths in Montreal's streets, as well as data representing the limits of Montreal's boroughs. The street data [1] and the borough data [2] was extracted from the City of Montreal's open data portal. The datasets contains data representing various information about pedestrian and shared streets in Montreal, as well as the geometric limits of its boroughs.

Description

In this lab, you will have to complete the Python code using Plotly and Dash in order to display a scatter map displaying the geographical location and type of each pedestrian path in Montreal. To make the chart interactive, the name of each borough and path will appear in a tooltip when hovered. You will also implement the code to display an informational panel when each marker is clicked.

To create this effect using Plotly, we will add two map layers to our figure. The first layer will be represented as a modified version of a choropleth map where each area is filled with the same color. The intention is to give the effect of an interactive map base. Normally, choropleth maps use color to encode the value of a variable corresponding to each geographic area. However, in our case, we will simply use Plotly's choropleth features to display a grey map base without any additional data encoded by the color. The second layer will be a scatter plot layer. Both these layers will be displayed using Mapbox, an open source mapping platform used by Plotly.

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 scatter map itself. The next two parts, the legend and the information panel, 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.

In this lab, we provide you with an archive containing 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.
- `callback.py`
- `helper.py` : This file contains some helper functions useful when generating the visualization. You do not need to modify it.
- `hover_template.py`
- `map_viz.py`
- `preprocess.py`
- `server.py` : This file is used to launch the application. You do not need to modify it.

Dataset

In this lab, you will have to make a scatter map from the data representing Montreal and its pedestrian paths. The locations and information about the pedestrian paths are located in a separate file from the geographic divisions of the polygons representing Montreal's boroughs. Since this data comes from two different sources, you will have to handle the two files using different formats.

The first dataset, representing data on pedestrian paths, is located in the `src/assets/data/projetpietonnisation2017.geojson` file in the archive provided for the lab. The dataset contains many properties. The following properties may be useful for this lab :

- `MODE_IMPLANTATION` : How long the pedestrian street will be implanted (e.g. permanent, temporary, etc.)
- `NOM_PROJET`: Name of the project which led to the pedestrian street

  `OBJECTIF_THEMATIQUE` : The intention behind the pedestrian street project (reading, taking photos, etc.)
- `TYPE_SITE_INTERVENTION` : The type of site where the pedestrian street project is located

The second dataset, representing data on contains all the geometries necessary to display the boroughs on a map, is located in the `src/assets/data/montreal.json` file in the archive provided for the lab. The dataset contains many properties. The following properties may be useful for this lab :

- `NOM` : Name of the borough
- `CODEID` : Unique identifier of the borough

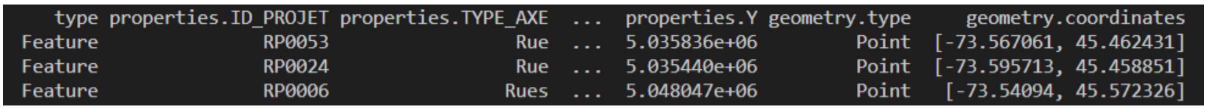
Data preprocessing

To begin, you will have to preprocess the data we provide you about the pedestrian streets and about Montreal's geography. The data contained in the GeoJSON and JSON files is raw, so it is necessary to process 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 data to a pandas dataframe (function `to_df`)
2. Simplify the names which will later be displayed in the legend (function `update_titles`)
3. Sort the data for the display (function `sort_df`)
4. Complete a utility function to get the neighborhood names (function `get_neighborhoods`)

To help complete your work for this part, Figure 1 illustrates a portion of the resulting data about pedestrian streets.



```
   type properties.ID_PROJET properties.TYPE_AXE  ...  properties.Y geometry.type      geometry.coordinates
Feature               RP0053                 Rue  ...  5.035836e+06         Point  [-73.567061, 45.462431]
Feature               RP0024                 Rue  ...  5.035440e+06         Point  [-73.595713, 45.458851]
Feature               RP0006                Rues  ...  5.048047e+06         Point   [-73.54094, 45.572326]
```

Figure 1 : Sample of heatmap data

Scatter map

For this second part, you will have to implement the main part of the data visualization. First, you will draw the map base, including the polygons that represent Montreal's neighborhoods. To do so, you may use Plotly's choropleth tracing features, although the result will not be a true choropleth map, which would use color to encode some data for each geographic region. As you will see, we provide you with a `z_vals` parameter in the `add_choro_trace`, which is a table containing always the same value for `z`. We also provided you with the color scale to use, containing only one color.The result will be that each geographic region will be the same color, but will benefit from some of the interactive features Plotly applies to Choropleth maps (such as hover tooltips).

Second, once the map base is drawn, you will have to add traces representing each type of pedestrian street to the map figure. Each type of street should be one trace, and each street should be one point in the trace.

To summarize, the steps are as follows :

1. Add the trace for the modified one color choropleth map showing the neighborhoods (function `add_choro_trace`)
2. Add the markers to the map (function `add_scatter_traces`)

Figure 2 shows what the map should resemble after these steps.

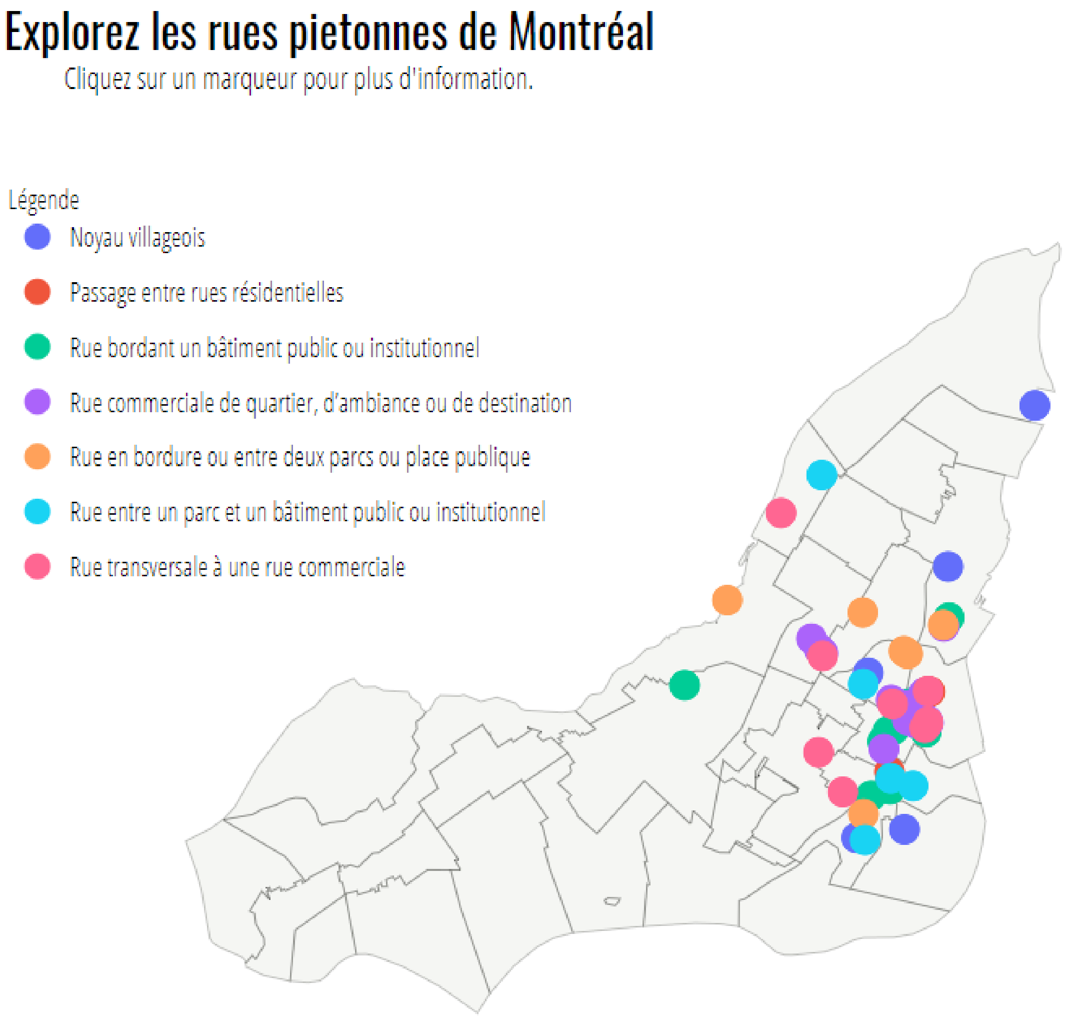

Explorez les rues pietonnes de Montréal
Cliquez sur un marqueur pour plus d'information.
Légende
● Noyau villageois
● Passage entre rues résidentielles
● Rue bordant un bâtiment public ou institutionnel
● Rue commerciale de quartier, d'ambiance ou de destination
● Rue en bordure ou entre deux parcs ou place publique
● Rue entre un parc et un bâtiment public ou institutionnel
● Rue transversale à une rue commerciale

Figure 2 : Map base with scatter markers

Tooltips

For this fourth part, you will define templates used to display the tooltips that appears when the mouse hovers over a neighborhood or a marker on the scatter map. When a neighborhood is displayed, simply its name should be displayed in the tooltip. Similarly, when a marker is displayed, simply its type of site should be displayed in the tooltip. The code to complete for this part is located in the file `hover_template.py`. The functions to fill in are `map_base_hover_template` for the display of the neighborhood names and `map_marker_hover_template` for the display of the types of sites.

Figures 3 and 4 below illustrate the tooltip behavior when the map base and a marker are hovered by the cursor.

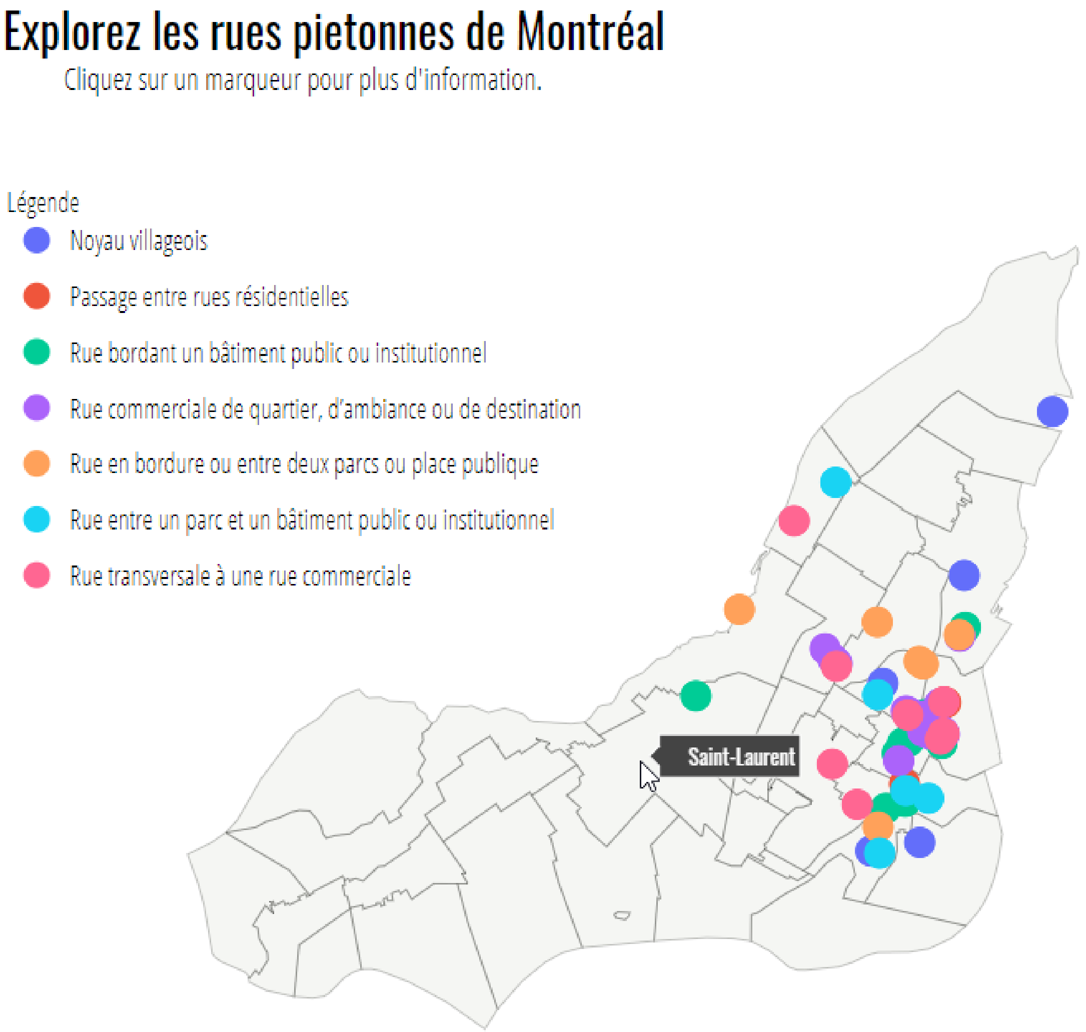
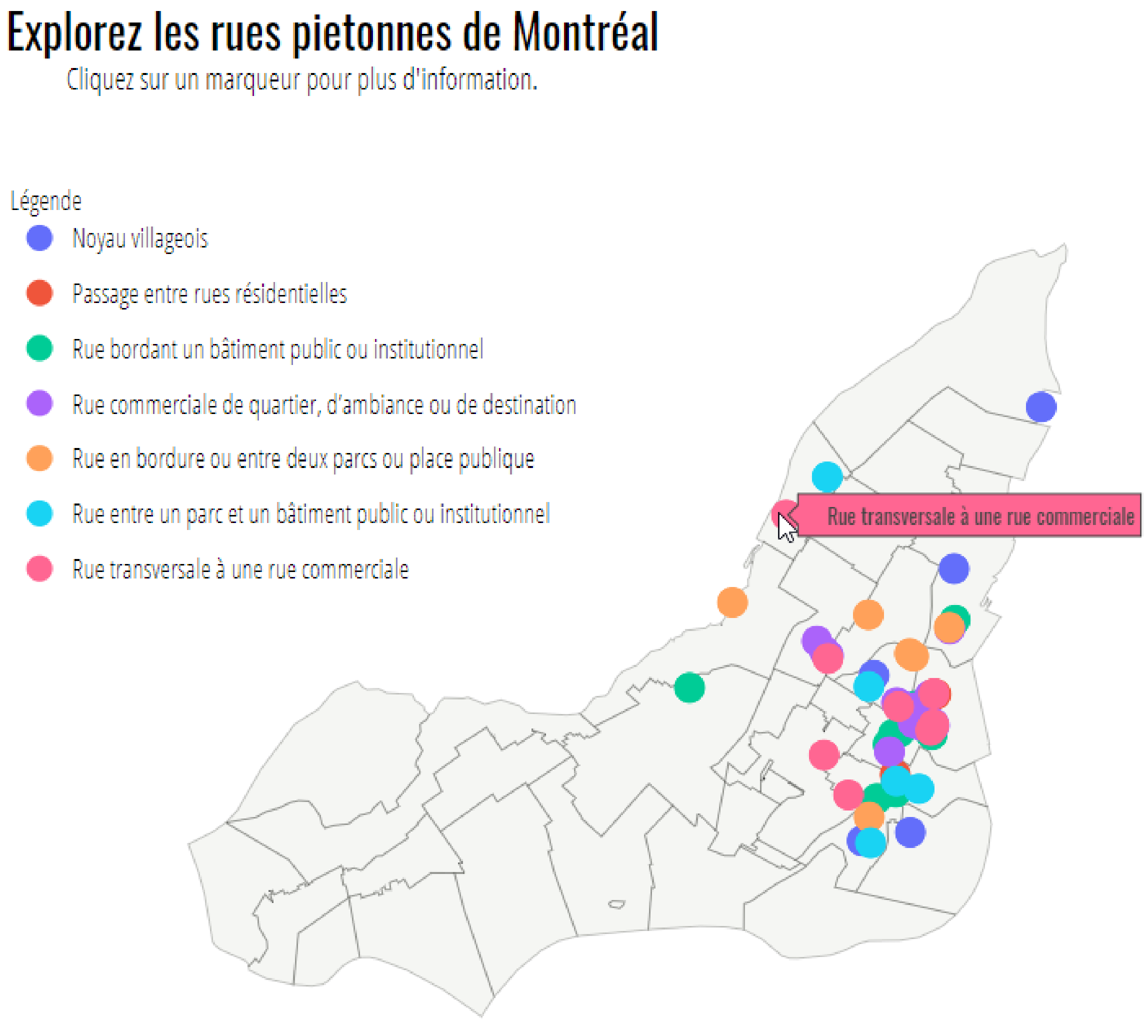

Explorez les rues pietonnes de Montréal
Cliquez sur un marqueur pour plus d'information.
Légende
● Noyau villageois
● Passage entre rues résidentielles
● Rue bordant un bâtiment public ou institutionnel
● Rue commerciale de quartier, d'ambiance ou de destination
● Rue en bordure ou entre deux parcs ou place publique
● Rue entre un parc et un bâtiment public ou institutionnel
● Rue transversale à une rue commerciale
Saint-Laurent

Figure 3 : Cursor hovering the map base



Explorez les rues pietonnes de Montréal

Cliquez sur un marqueur pour plus d'information.

Légende
● Noyau villageois
● Passage entre rues résidentielles
● Rue bordant un bâtiment public ou institutionnel
● Rue commerciale de quartier, d'ambiance ou de destination
● Rue en bordure ou entre deux parcs ou place publique
● Rue entre un parc et un bâtiment public ou institutionnel
● Rue transversale à une rue commerciale
Rue transversale à une rue commerciale

Figure 4 : Cursor hovering a marker on the scatter map

Information panel

For this fourth part, you will generate an informational panel which appears to the left of the map when a marker is clicked. Part of the structure for this part is already provided for you in the file `app.py`. Your task consists of completing the callback functions for the various contexts in which the map may be clicked. You will have to complete the functions in the file `callback.py`. The panel should contain as title the name of the project, written in the same color as its associated marker. Further, below its title, it should have a subtitle indicating the intended duration of the site (permanent, temporary, etc.). When available, the panel should also list the intended themes for the site, presented in the format of an unordered list.

Thus, the steps to follow for this part are :

1. Handle the default behavior for the map before any click events have been registered (function `no_clicks`). The panel should not be displayed until a marker is clicked.
2. Handle the behavior for the map when the map base is clicked instead of a marker. If the panel is displayed, its information should stay the same. If the panel is not displayed, it should stay hidden from view (function `map_marker_clicked`).
3. Handle the behavior when a marker is clicked, so the panel is displayed with the appropriate information (function `map_marker_clicked`)

Figure 5 gives an example of what the panel contains when one of the markers is clicked.

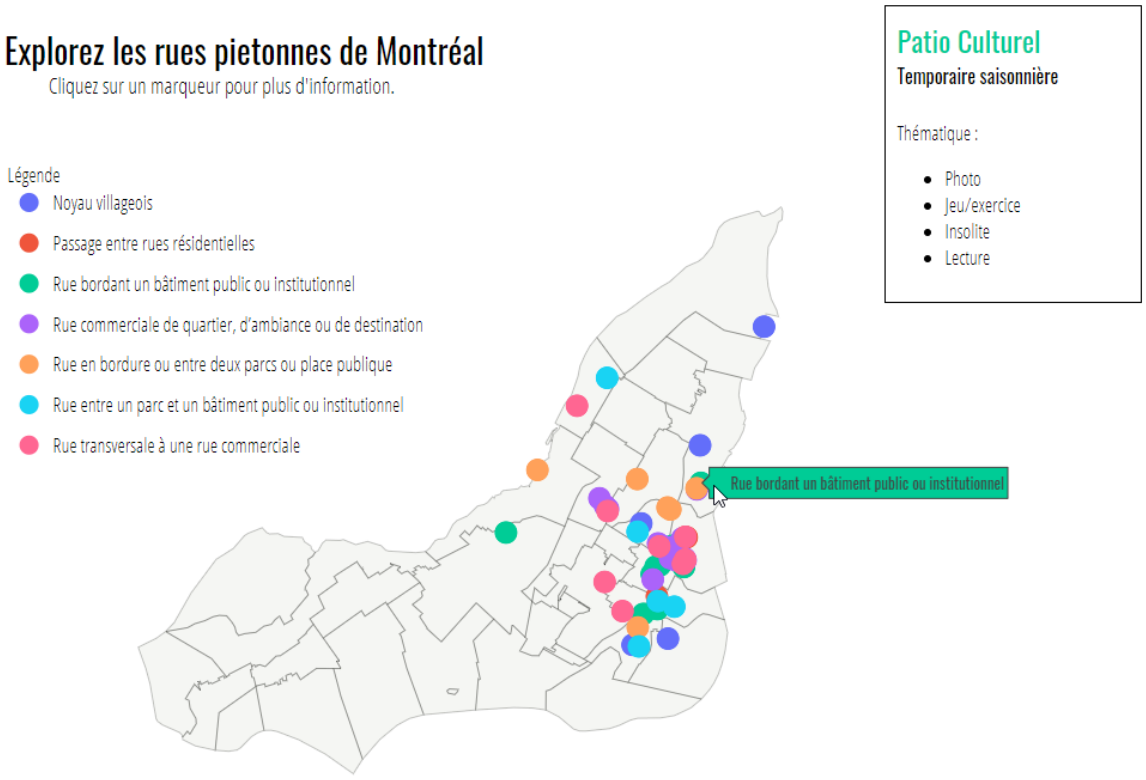

Explorez les rues pietonnes de Montréal
Cliquez sur un marqueur pour plus d'information.
Légende
Noyau villageois
Passage entre rues résidentielles
Rue bordant un bâtiment public ou institutionnel
Rue commerciale de quartier, d'ambiance ou de destination
Rue en bordure ou entre deux parcs ou place publique
Rue entre un parc et un bâtiment public ou institutionnel
Rue transversale à une rue commerciale
Rue bordant un bâtiment public ou institutionnel
Patio Culturel
Temporaire saisonnière
Thématique :
Photo
Jeu/exercice
Insolite
Lecture

Figure 5 : The information panel showing information about the "Patio Culturel" project

Submission

The instructions for the submission are :

1. You must place your project code in a compressed ZIP file named [StudentID1_StudentID2_StudentID3]

2. The lab must be submitted before [April 10th 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
Scatter map	6
Tooltips	3
Information panel	5
Overall quality and clarity of the submission	1

Requirement	Points
Total	**20**

References

[1] Service de l'urbanisme et de la mobilité / Arrondissements, "Rues piétonnes et partagées ," Montréal : Portail de données ouvertes. Available: http://donnees.ville.montreal.qc.ca/dataset/rues-pietonnes [Accessed 01 09 2020].

[2] Service des infrastructures du réseau routier - Division de la géomatique, "Limite administrative de l'agglomération de Montréal (Arrondissements et Villes liées)," Montréal : Portail de données ouvertes. Available: http://donnees.ville.montreal.qc.ca/dataset/polygones-arrondissements [Accessed 01 09 2020].