

# TP5 (version Python)

---

## INF8808: Visualisation de données

Département de génie informatique et génie logiciel



# POLYTECHNIQUE MONTRÉAL

**Auteure:** Olivia Gélinas

**Chargé de lab:** Hellen Vasques

## Objectifs

---

L'objectif de ce travail pratique est de créer une carte à nuages de points (*scatter map*) interactive à l'aide de données ouvertes en formats JSON et GeoJSON.

Avant de commencer, nous vous recommandons d'avoir effectué les lectures suivantes et d'avoir effectué les exercices suivants:

- Mapbox choroplèthe Maps

<https://plotly.com/python/mapbox-county-choropleth/>

**Lectures :**

- Scatter plots on Mapbox

<https://plotly.com/python/scattermapbox/>

---

**Exercices :** TP5 exercices : 1, 2, 3, 4

## Introduction

---

Une carte à nuage de points est similaire à un nuage de points typique, sauf que les points sont positionnés par rapport à leur emplacement géographique sur une carte. Ce type de graphique est utile lorsque le contexte géographique est important pour notre interprétation des données.

Dans ce travail pratique, vous implémenterez une carte à nuage de points interactive utilisant des données représentant les rues piétonnes de Montréal, ainsi que des données représentant les limites géographiques de chaque arrondissement de Montréal. Les données sur les rues [1] et les données d'arrondissement [2] ont été extraites du portail de données ouvertes de la Ville de Montréal. Les données contiennent diverses

informations sur les rues piétonnes et partagées de Montréal, ainsi que les limites géométriques de ses arrondissements.

## Description

---

Dans ce travail pratique, vous devrez compléter le code Python à l'aide de Dash et Plotly afin d'afficher une carte à nuage de points montrant l'emplacement géographique et le type de chaque rue piétonne à Montréal. Pour rendre le graphique interactif, le nom de chaque arrondissement apparaîtra centré dans son polygone lorsqu'il est survolé. Vous implémenterez également le code pour afficher un panneau d'information lorsqu'un point est cliqué.

Pour créer cet effet en utilisant Plotly, nous allons ajouter deux couches de carte à notre figure. La première couche sera représentée comme une version modifiée d'une carte choroplèthe où chaque zone est remplie de la même couleur. L'intention est de donner l'effet d'une base cartographique interactive. Normalement, les cartes choroplèthes utilisent la couleur pour encoder la valeur d'une variable correspondant à chaque zone géographique. Cependant, dans notre cas, nous utiliserons simplement les fonctionnalités choroplèthes de Plotly pour afficher une base de carte grise sans aucune donnée supplémentaire encodée par la couleur. La deuxième couche sera une couche de nuage de points. Ces deux couches seront affichées à l'aide de Mapbox, une plate-forme de cartographie *open source* utilisée par Plotly.

Les sous-sections suivantes présentent les différentes parties que vous aurez à compléter pour ce travail pratique. Pendant que vous codez, nous vous recommandons de compléter le traitement des données d'abord, suivi de la mise en œuvre de la carte à nuage de points lui-même. Les deux parties suivantes, la légende et le panneau d'information, sont indépendantes les uns des autres.

## Structure des fichiers

Pour effectuer ce travail, vous devrez remplir les différentes sections **TODO** dans les fichiers de l'archive fournie pour le travail pratique. Les commentaires dans le code expliquent plus en détail les étapes à suivre.

Dans ce travail pratique, nous vous fournissons une archive contenant 7 fichiers Python utilisés pour accomplir la visualisation souhaitée:

- **app.py**: Ce fichier génère la structure HTML de la page Web et orchestre les étapes requises pour créer la visualisation. Vous n'avez pas besoin de le modifier.
- **callback.py**
- **helper.py**: Ce fichier contient des fonctions utiles lors de la génération de la visualisation. Vous n'avez pas besoin de le modifier.
- **hover\_template.py**
- **map\_viz.py**
- **preprocess.py**
- **server.py**: Ce fichier est utilisé pour lancer l'application. Vous n'avez pas besoin de le modifier.

## Données

Dans ce travail pratique, vous devrez faire une carte à nuage de points à partir des données représentant Montréal et ses rues piétonnes. Les emplacements et les informations sur les rues piétonnes sont situés dans un fichier et les divisions géographiques des polygones représentant les arrondissements de Montréal sont situés dans un autre fichier. Étant donné que ces données proviennent de deux sources différentes, vous devrez gérer les deux fichiers en utilisant des formats différents.

Le premier jeu de données, représentant les données sur les rues piétonnes, se trouve dans le fichier `src/assets/data/projetpietonnisation2017.geojson` dans l'archive fournie pour le travail pratique. L'ensemble de données contient de nombreuses propriétés. Les propriétés suivantes peuvent être utiles pour ce travail pratique:

- **MODE\_IMPLANTATION**: Combien de temps la rue piétonne sera implantée (par exemple, permanente, temporaire, etc.)
- **NOM\_PROJET**: Nom du projet qui a conduit à la rue piétonne
- **OBJECTIF\_THEMATIQUE**: L'intention derrière le projet de rue piétonne (lecture, prise de photos, etc.)
- **TYPE\_SITE\_INTERVENTION**: Le type de site où se situe le projet de rue piétonne

Le second jeu de données contient toutes les géométries nécessaires pour afficher les arrondissements sur une carte. Il se trouve dans le fichier `src/assets/data/montreal.json` dans l'archive fournie pour le travail pratique. L'ensemble de données contient de nombreuses propriétés. Les propriétés suivantes peuvent être utiles pour ce travail pratique:

- **NOM**: Nom de l'arrondissement
- **CODEID**: Identifiant unique de l'arrondissement

## Prétraitement des données

Pour commencer, vous devrez prétraiter les données que nous vous fournissons sur les rues piétonnes et sur la géographie de Montréal. Les données contenues dans les fichiers GeoJSON et JSON sont brutes. Il est donc nécessaire de traiter certaines parties de celles-ci afin qu'elles puissent être correctement utilisées par la bibliothèque Plotly. Pour ce faire, vous devez compléter le fichier `preprocess.py`.

Plus précisément, vous devrez effectuer ces étapes:

1. Convertissez les données en un dataframe pandas (fonction `to_df`)
2. Simplifiez les noms qui seront affichés dans la légende (fonction `update_titles`)
3. Triez les données pour l'affichage (fonction `sort_df`)
4. Complétez une fonction utilitaire pour obtenir les noms des quartiers (fonction `get_neighborhoods`)

Pour vous aider à terminer votre travail pour cette partie, la figure 1 illustre une partie des données obtenues sur les rues piétonnes.

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: Exemple de données de heatmap

## Carte à nuage de points

Pour cette deuxième partie, vous devrez implémenter la partie principale de la visualisation de données. Tout d'abord, vous allez dessiner la base de la carte, y compris les polygones qui représentent les quartiers de Montréal. Pour ce faire, vous pouvez utiliser les fonctionnalités de traçage choroplèthes de Plotly, bien que le résultat ne soit pas une véritable carte choroplèthe, qui utiliserait la couleur pour encoder certaines données pour chaque région géographique. Comme vous le verrez, nous vous fournissons un paramètre `z_vals` dans `add_choro_trace`, qui est une table contenant toujours la même valeur pour `z`. Nous vous avons également fourni l'échelle de couleurs à utiliser, ne contenant qu'une seule couleur. Le résultat sera que chaque région géographique sera de la même couleur, mais bénéficiera de certaines des fonctionnalités interactives qui s'appliquent aux cartes choroplèthe de Plotly (telles que les info-bulles de survol) .

Deuxièmement, une fois la base de la carte dessinée, vous devrez ajouter des traces représentant chaque type de rue piétonne. Chaque type de rue doit être une *trace* et chaque rue doit être un point dans la *trace*.

Pour résumer, les étapes sont les suivantes:

1. Ajoutez le tracé de la carte choroplèthe montrant les quartiers (fonction `add_choro_trace`)
2. Ajoutez les points à la carte (fonction `add_scatter_traces`)

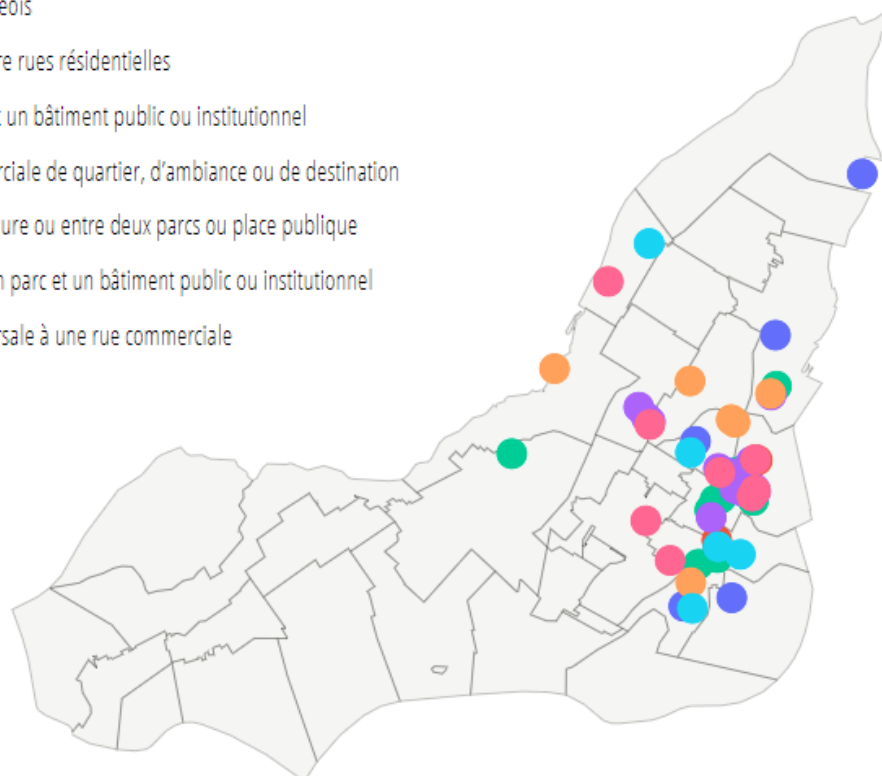
La figure 2 montre à quoi la carte doit ressembler après ces étapes.

### Explorez les rues piétonnes 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: Base de carte avec des points*

## Info-bulles

Pour cette quatrième partie, vous définirez des modèles permettant d'afficher les info-bulles qui apparaissent lorsque la souris survole un quartier ou un point sur la carte à nuage de points. Lorsqu'un quartier est affiché, son nom doit simplement être affiché dans l'info-bulle. Lorsqu'un point est affiché, il suffit d'afficher son type de site dans l'info-bulle. Le code à compléter pour cette partie se trouve dans le fichier `hover_template.py`. Les fonctions à remplir sont `map_base_hover_template` pour l'affichage des noms de quartier et `map_marker_hover_template` pour l'affichage des types de sites.

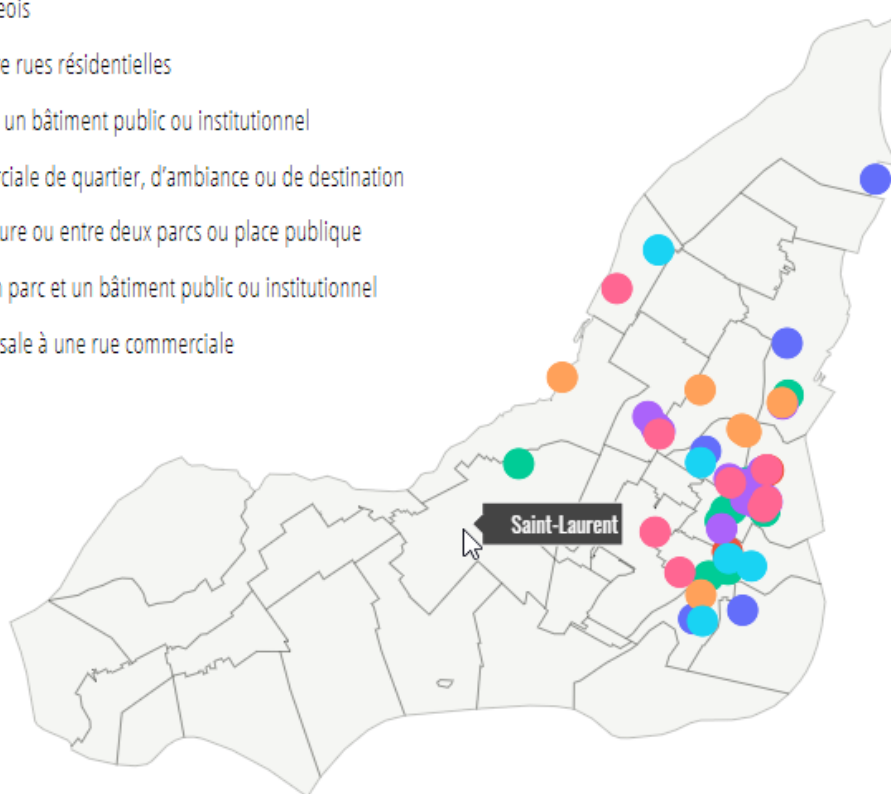
Les figures 3 et 4 ci-dessous illustrent le comportement de l'info-bulle lorsque la base de la carte et un point sont survolés par le curseur.

## Explorez les rues piétonnes 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 3: Curseur survolant la base de la carte*

## Explorez les rues piétonnes 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

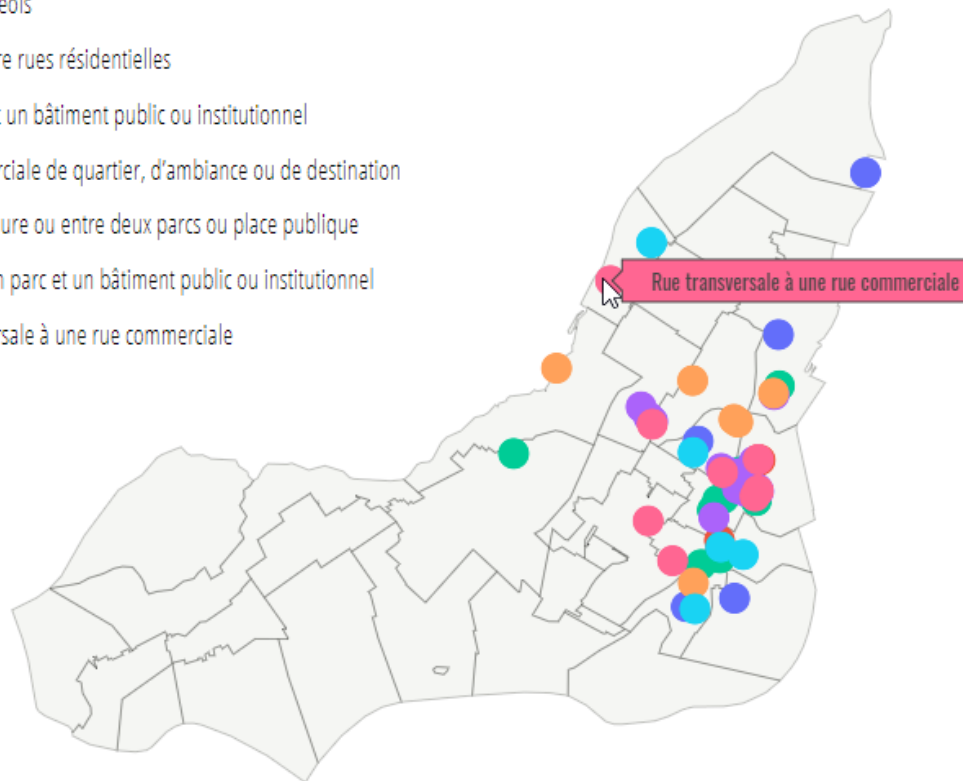


Figure 4: Curseur survolant un point sur la carte à nuage de points

## Panneau d'information

Pour cette quatrième partie, vous allez générer un panneau d'information qui apparaît à gauche de la carte lorsqu'un point est cliqué. Une partie de la structure de cette partie est déjà fournie pour vous dans le fichier `app.py`. Votre tâche consiste à remplir les fonctions pour les différents contextes dans lesquels la carte peut être cliquée. Vous devrez compléter les fonctions dans le fichier `callback.py`. Le panneau doit contenir comme titre le nom du projet, écrit dans la même couleur que son point associé. De plus, sous son titre, il devrait avoir un sous-titre indiquant la durée prévue du site (permanent, temporaire, etc.). Lorsqu'ils sont disponibles, le panneau doit également lister les thèmes prévus pour le site, présentés sous la forme d'une liste non ordonnée.

Ainsi, les étapes à suivre pour cette partie sont:

1. Gérez le comportement par défaut de la carte avant que les événements de clic aient été enregistrés (fonction `no_clicks`). Le panneau ne doit pas être affiché tant que vous n'avez pas cliqué sur un point.
2. Gérez le comportement de la carte lorsque l'utilisateur clique sur la base de la carte au lieu d'un point. Si le panneau est affiché, ses informations doivent rester les mêmes. Si le panneau n'est pas affiché, il doit rester caché de la vue (fonction `map_marker_clicked`).

- Gérez le comportement lorsqu'un point est cliqué, de sorte que le panneau s'affiche avec les informations appropriées (fonction `map_marker_clicked`)

La figure 5 donne un exemple de ce que contient le panneau lorsqu'un des points est cliqué.

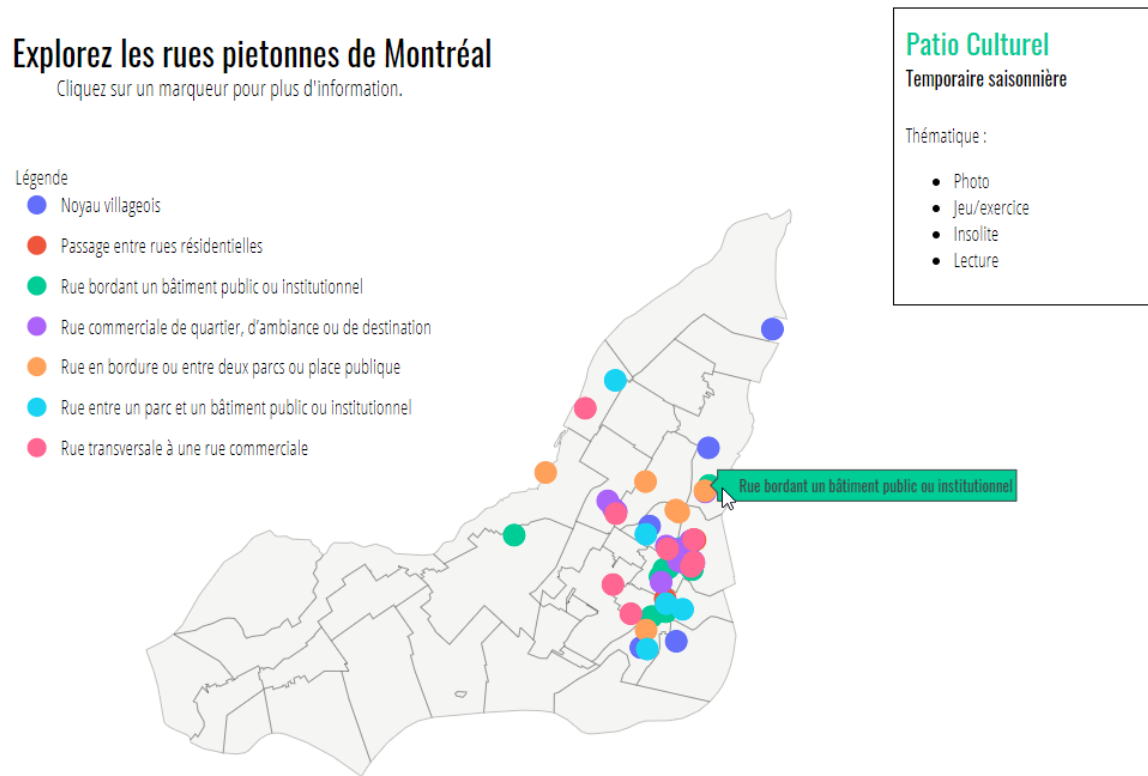


Figure 5: Le panneau d'information présentant des informations sur le projet "Patio Culturel"

## Soumission

Les instructions pour la soumission sont:

- Vous devez placer le code de votre projet dans un fichier ZIP compressé nommé [Matricule1\_Mtricule2\_Matricule3.zip]
- Le travail pratique doit être soumis avant le [10 Avril 23:59]

## Évaluation

Dans l'ensemble, votre travail sera évalué selon la grille suivante. Chaque section sera évaluée sur l'exactitude et la qualité du travail.

** Exigence **	** Points **
Prétraitement des données	5
Carte à nuage de points	6
Info-bulles	3

<b>** Exigence **</b>	<b>** Points **</b>
Panneau d'information	5
Qualité globale et clarté de la soumission	1
<b>** Total **</b>	<b>** 20 **</b>

## Références

---

[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].