



**POLYTECHNIQUE  
MONTRÉAL**

LE GÉNIE  
EN PREMIÈRE CLASSE

# INF8808 – Visualisation de données

## Travail pratique 2

Auteurs:

Adrien Dessinges et Antoine Béland

Chargé de laboratoire:

Jean-Philippe Corbeil

Hiver 2019

Département de génie informatique et génie logiciel

# 1 Objectifs

Le but de ce travail pratique est de réaliser un diagramme à lignes brisées (*line chart*) avec un zoom dynamique utilisant des données ouvertes fournies dans un fichier au format CSV.

Avant de commencer ce travail, il est recommandé d'avoir lu les chapitres 5, 6 et 7 du livre de Scott Murray [1].

## 2 Introduction

Un diagramme à lignes brisées est un graphique où les données sont représentées par une ou plusieurs courbes. Il est extrêmement commun et se retrouve dans de nombreux domaines où l'on souhaite observer l'évolution d'un paramètre en fonction d'un autre. Il est notamment utilisé en mathématique pour illustrer le comportement d'une fonction de la forme  $y = f(x)$ .

En ce qui a trait à ce travail pratique, vous aurez à réaliser un diagramme à lignes brisées utilisant les données sur le [nombre de vélos passant quotidiennement dans certaines rues de Montréal](#) pour l'année 2016. Ces données sont issues des différents vélos-compteurs qui se trouvent à plusieurs endroits sur les pistes cyclables de la ville. D'ailleurs, la Figure 1 illustre à quoi ressemblent les vélos-compteurs utilisés par la ville. Ces données sont disponibles via le [portail des données ouvertes](#) de la ville de Montréal.



FIGURE 1 – Vélo-compteur se trouvant sur la rue Laurier

© François Démontagne, 2015

## 3 Travail à réaliser

Pour ce travail pratique, vous devrez compléter le code JavaScript nécessaire pour l’affichage d’un diagramme à lignes brisées utilisant les données sur le nombre de vélos circulant quotidiennement dans les rues de la ville de Montréal en 2016. Également, vous aurez à réaliser un *brush* et une légende accompagnant le graphique.

Les sous-sections qui suivent présentent les différentes parties qui devront être réalisées pour ce travail. Il est à noter qu’il est nécessaire de compléter les deux premières parties de ce travail (prétraitement des données et création des diagrammes) avant de réaliser la troisième ou la quatrième partie, qui sont indépendantes. Assurez-vous de compléter les différents « TODO » qui se trouvent dans les fichiers se trouvant dans le dossier « assets/scripts ».

### 3.1 Prétraitement des données

Pour cette première partie, vous devrez effectuer un certain traitement sur les données fournies par la ville de Montréal. En effet, les données contenues dans le fichier CSV sont des données brutes. Il est donc nécessaire de réorganiser certains éléments afin que ces mêmes données puissent être utilisées correctement par la bibliothèque D3. Pour ce faire, vous devrez compléter le fichier « **1-preproc.js** » se trouvant dans le dossier « assets/scripts ». Plus précisément, vous aurez à compléter les éléments suivants :

- Préciser le domaine de l’échelle de couleurs (fonction « `domainColor` »);
- Préciser le domaine des axes verticaux et horizontaux (fonctions « `domainX` » et « `domainY` »);
- Convertir les dates du fichier CSV en objet JavaScript (fonction « `parseDate` »);
- Trier les données par rue puis par date (fonction « `createSources` »).



#### Attention

---

Assurez-vous de convertir les nombres provenant du fichier CSV en type *number* puisqu’il s’agit de chaîne de caractères initialement (*string*). Pour réaliser cette conversion, vous pouvez utiliser la fonction `parseInt` de JavaScript. Cette conversion est nécessaire puisque sinon, D3 n’interprétera pas correctement les données.

---

## 3.2 Création des diagrammes

Pour cette deuxième partie, il est demandé de réaliser la visualisation de données. Cette visualisation est en fait divisé en deux diagrammes à lignes brisées. Un, appelé « focus », permet d’afficher les données en détail alors que l’autre, appelé « contexte », permet de sélectionner les données à afficher dans le « focus » à l’aide d’un *brushing* (voir section suivante). Par souci de clarté, la Figure 2 illustre les deux parties de la visualisation à réaliser.

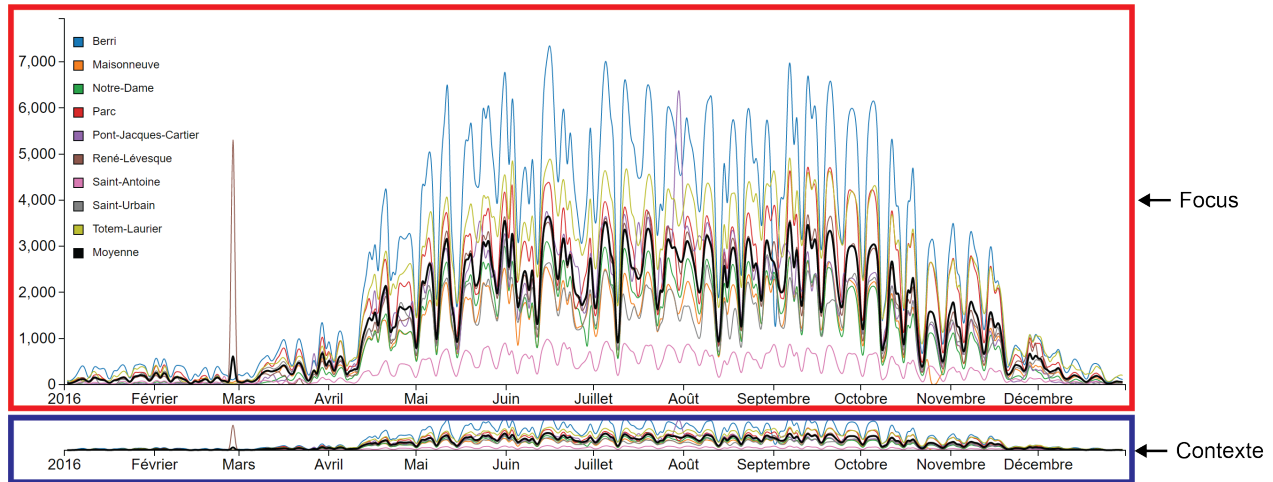


FIGURE 2 – Les diagrammes « focus » (rouge) et « contexte » (bleu) à réaliser

Chacune des lignes des graphiques devra être associée à une couleur qui représente les données pour une rue particulière. De plus, la ligne correspondant à la moyenne de la circulation quotidienne devra être noire et devra être plus épaisse que les autres.

Les axes, quant à elles, sont déjà implémentés dans le code. Cependant, assurez-vous de leur assigner la bonne plage de données (voir §PRÉTRAITEMENT DES DONNÉES). Vous remarquerez que les graphiques « focus » et « contexte » utilisent les mêmes données et les mêmes couleurs. Le code pour générer ces deux graphiques sera donc assez similaire.

Pour réaliser cette partie, vous devrez compléter le fichier « **2-line-charts.js** ». Vous aurez à compléter les fonctionnalités suivantes :

- Générer une ligne SVG pour les deux graphiques (fonction « `createLine` »);
- Créer le diagramme à lignes brisées « focus » (fonction « `createFocusLineChart` »);
- Créer le diagramme à lignes brisées « contexte » (fonction « `createContextLineChart` »).

### 3.3 Implémentation du *brush*

La troisième partie de ce travail consiste en la réalisation d'un *brushing*. Ce dernier permet de sélectionner une plage de données particulière qui sera affichée. Dans le cas présent, le diagramme « contexte » permettra la sélection de la plage de données qui devra être affichée dans le diagramme « focus ». La Figure 3 illustre les derniers propos. Effectivement, le diagramme « contexte » a permis la sélection d'un intervalle pour cibler les données associées au mois de juin, puis le diagramme « focus » a mis à jour son domaine en  $x$  pour afficher uniquement les données du mois de juin.

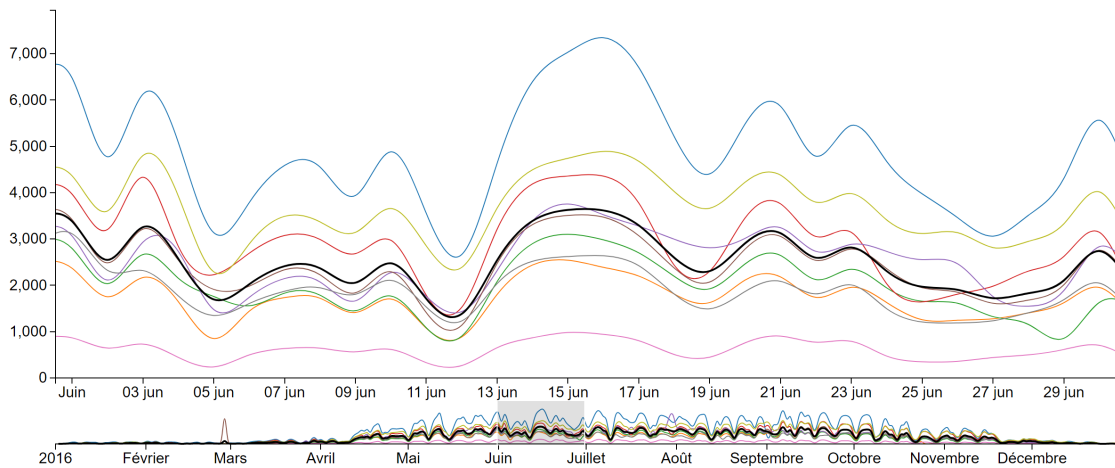


FIGURE 3 – Sélection du mois de juin à l'aide du *brush*

La sélection de la zone dans le graphique « contexte » a déjà été réalisée pour vous. Vous aurez uniquement à mettre à jour le graphique « focus » à partir de la zone qui aura été sélectionnée dans le graphique « contexte ». En ce sens, vous aurez à mettre à jour l'échelle horizontale et verticale du graphique « focus » et à redessiner ses lignes à partir de la zone sélectionnée. Vous aurez donc à compléter le fichier « **3-brush.js** » pour réaliser cette fonctionnalité.

### 3.4 Création de la légende

En ce qui conserve la dernière partie du travail à réaliser, une légende devra être ajoutée pour simplifier la lecture de la visualisation. En effet, cette légende permettra d'associer chacune des couleurs à un nom de rue particulier. Il est à noter que cette légende devra être dessinée au premier plan, c'est-à-dire au-dessus du graphique « focus ». Également, assurez-

vous que les couleurs de la légende correspondent bels et bien aux lignes des diagrammes. La Figure 4 illustre d’ailleurs à quoi la légende devrait ressembler.

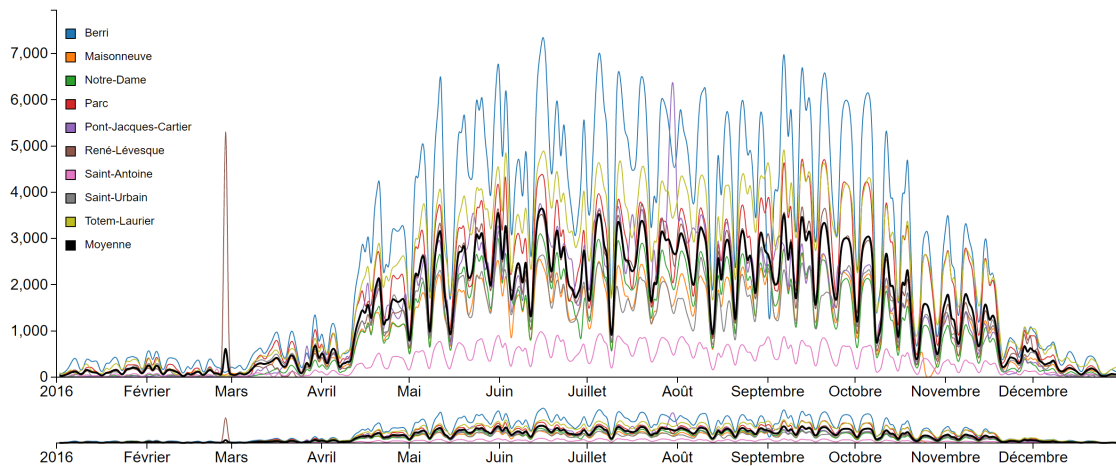


FIGURE 4 – Légende associée à la visualisation

Le nombre de rues à afficher étant assez important, il peut être difficile de bien lire les données associées à une rue particulière. En sens, il serait intéressant de pouvoir afficher ou masquer les données de certaines rues au besoin. Pour ce faire, vous devrez réaliser une méthode permettant d’afficher ou masquer les données associées à une rue quelconque lorsque le carré associé à cette même rue est cliqué dans la légende. Il est à noter que lorsque les données associées à une rue sont masquées, le carré dans la légende qui est associé à cette même rue doit devenir blanc, mais doit conserver un cadre noir. Par souci de clarté, la Figure 5 illustre les derniers propos.

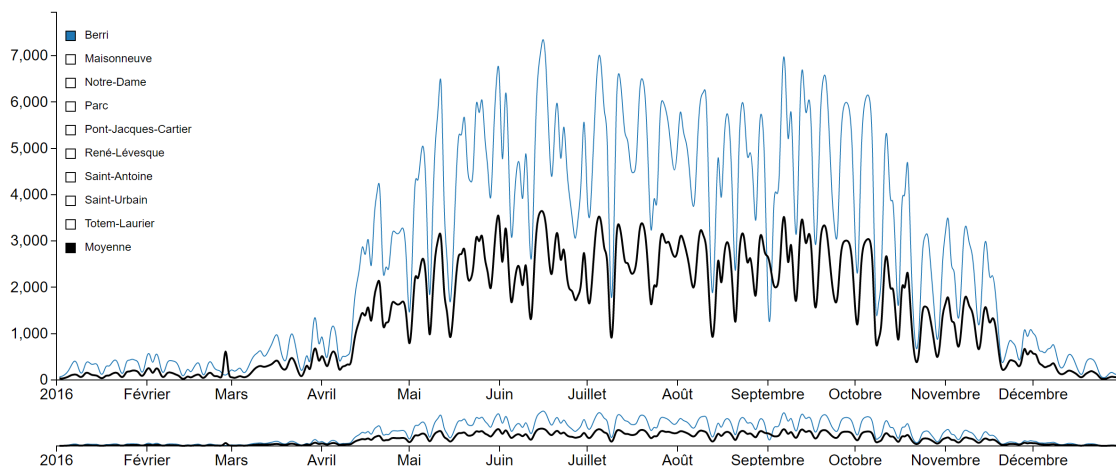


FIGURE 5 – Visualisation affichant uniquement les données pour la rue Berri et la moyenne

Pour réaliser cette partie, vous aurez à compléter le fichier « **4-legend.js** ». Plus précisément, vous devrez compléter les éléments suivants :

- Afficher la légende à gauche du graphique « focus » (fonction « `legend` »);
- Rendre l’affichage des lignes des graphiques dynamiques (fonction « `displayLine` »).



### Attention

---

Assurez-vous d’afficher ou de masquer les données des rues dans les graphiques « focus » et « contexte ». En effet, vous devez mettre en jour les deux graphiques.

---

## 4 Remise

Voici les consignes à suivre pour la remise de ce travail pratique :

1. Vous devez placer le code de votre projet dans un dossier compressé au format ZIP nommé « `TP2_matricule1_matricule2.zip` ».
2. Le travail doit être remis avant **8h30**, le **31 janvier 2019** sur Moodle.

## 5 Évaluation

Globalement, vous serez évalué sur le respect et le bon fonctionnement des exigences demandées. Plus précisément, le barème de correction est le suivant :

Exigences	Points
Prétraitement de données	6
Création des diagrammes	6
Implémentation du <i>brush</i>	3
Création de la légende	4
Qualité et clarté du code	1
Total	20

Ce travail pratique a une pondération de **7,5%** sur la note du cours.

## Références

- [1] S. Murray, *Interactive Data Visualization for the Web : An Introduction to Designing with D3*. O'Reilly Media Inc., 2013.