



# INF8808 - Visualisation de données

---

## TP2 : Line Chart

Département de génie informatique et génie logiciel  
Polytechnique Montréal  
Hiver 2017



## 1. Présentation du TP

### a. Objectifs

Le but de ce TP est de réaliser un line chart avec un zoom dynamique depuis une base de données externe. Il est recommandé d'avoir lu les chapitres 5, 6 et 7 du livre de Scott Murray avant de commencer le TP. Les données utilisées proviennent du portail de données de la ville de Montréal : <http://donnees.ville.montreal.qc.ca/dataset/velos-comptage>.

### b. Line Chart

Un **Line Chart** (ou line graph ou graphe de lignes) est un graphique où les données sont représentées par une ou plusieurs courbes. Ils sont extrêmement communs et se retrouvent 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 représenter graphiquement une courbe d'équation  $y = f(x)$ .

### c. Détail du TP

Le travail pratique est constitué d'un fichier HTML et de fichiers JavaScripts contenant le code JavaScript et d'un fichier CSV contenant les données à afficher. Le fichier CSV contient des données sur le nombre de vélos quotidiens passant dans certaines rues de Montréal quotidiennement en 2014 ainsi que la moyenne. On se propose donc de visualiser ces données à l'aide d'un line chart. Dans une première partie, on va prétraiter les données pour les rendre utilisables par D3. Dans une deuxième partie nous allons dessiner le line chart en question. Enfin dans une troisième et quatrième partie nous allons enrichir ce line chart en rajoutant la possibilité de zoomer sur une partie du graph et en ajoutant une légende permettant d'afficher ou de masquer les rues souhaitées. Il est à noter que les parties c) et d) du TP sont indépendantes et peuvent donc être réalisées dans n'importe quel ordre. Par contre il est impératif de réaliser les autres parties avant.

## 2. Travail à réaliser

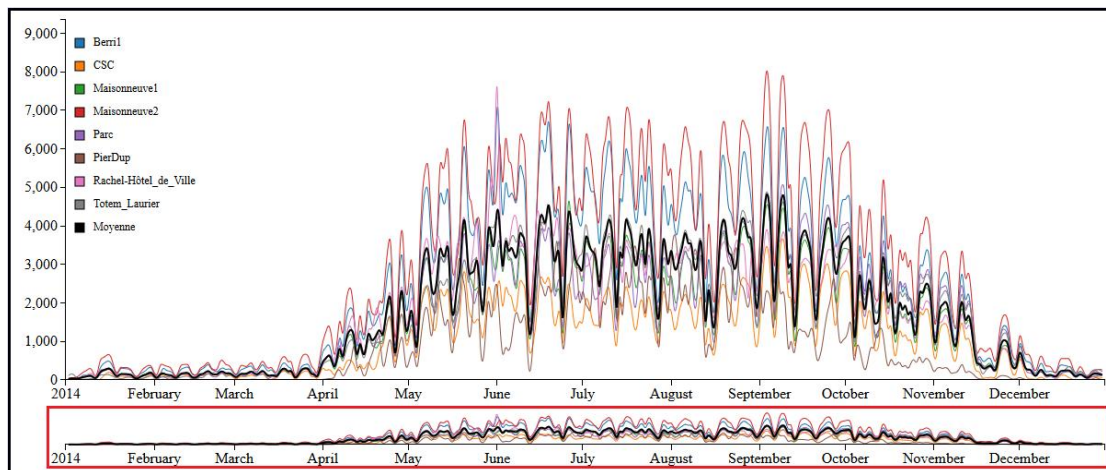
### a. Prétraitement des données

Les données contenues dans le csv sont des données "brutes". Par exemple, les dates prises dans le fichier ne sont pour d3 qu'une suite de caractère et non un objet Date et il serait intéressant de trier les données par rue puis par date. Dans cette partie il faut :

- Parser les dates du CSV en objet JavaScript (fonction **parseData**).
- Calculer la moyenne quotidienne des vélos passant dans les rues (fonction **procMean**).
- Préciser le domaine de la fonction scale **color** (fonction **domainColor**).
- Trier les données par rue puis par date (fonction **createSources**).
- Préciser le domaine des axes verticaux et horizontaux (fonction **domainX** et **domainY**).

### b. Création du Line Chart *focus* et du sous Line Chart *context*

Le line chart permet de visualiser les variations quotidiennes de l'utilisation des vélos dans les rues de Montréal. On souhaite que chaque ligne possède sa propre couleur permettant de distinguer chacune des rues affichées (**épaisseur 1px**). Une dernière ligne, noire et plus épaisse, correspond à la moyenne de la circulation quotidienne (**épaisseur 2px**).



Line chart à implémenter. Le sous Line Graph est encadré en rouge

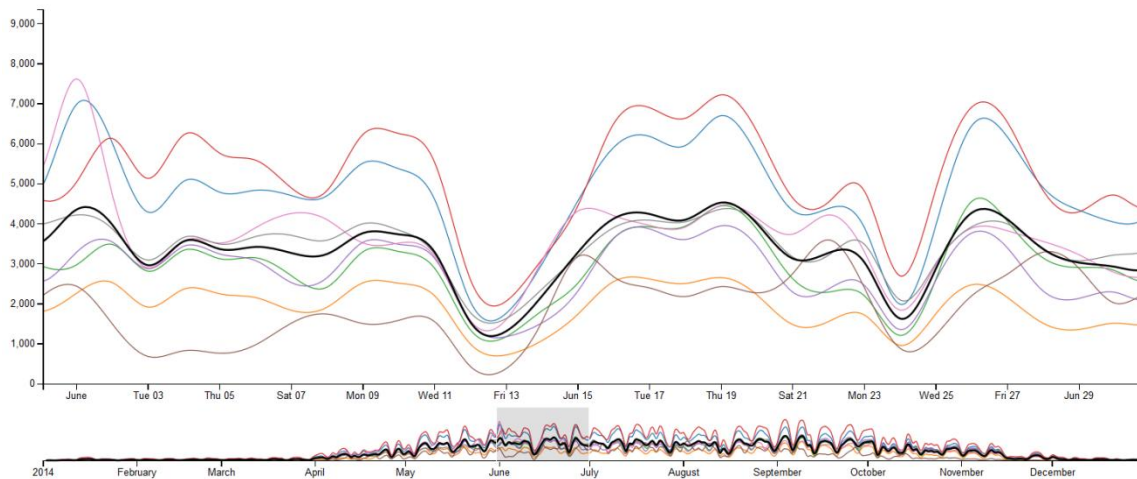
Les axes sont déjà implémentés en SVG mais il faut leur assigner la bonne plage de données (cf. partie a). De plus vous remarquerez dans le code fourni qu'il est également question d'un sous line chart. Ce dernier contient les mêmes informations que le précédent et servira simplement de base à notre zoom. Dans cette partie il faut :

- Créer les *lines* pour les deux line charts (fonctions **createLine** et **createLine2**).
- Créer les line charts (fonctions **createFocusLineGraph** et **createContextLineGraph**).

**Remarque :** utiliser la définition **clip** pour le line graph principal.

### c. Ajout du zoom

Le line chart est en fait composé de deux sous-line charts : un premier (*focus*) dans lequel sera affiché le graphe et un deuxième (*context*), plus petit, qui permettra de réduire le champ de vue à une partie du graphe seulement, ce qui revient à zoomer dessus. La sélection de la zone sur laquelle zoomer est déjà implémentée, seul le zoom en lui-même doit être implémenté. Cliquer dans *context* à l'extérieur du cadre revient à retourner à l'échelle originelle.



Line chart zoomé sur le mois de juin

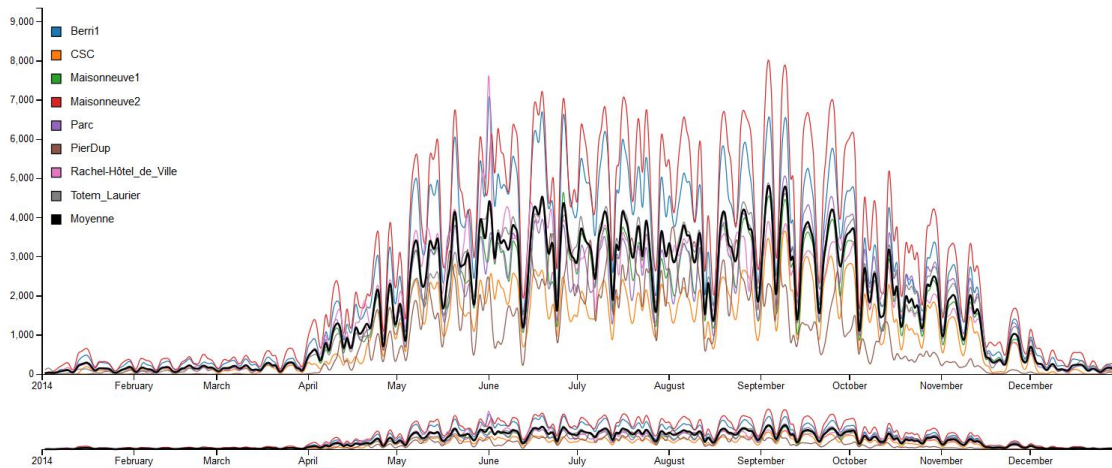
Observez comment l'échelle de temps se modifie d'elle-même. Sans le prétraitement des données et sans le parsing des dates, le changement d'échelle au niveau des mois et des jours n'aurait pas été possible.

En pratique nous allons réaliser ce zoom à l'aide d'un « brush ». A chaque fois qu'une nouvelle sélection est faite dans le sous Line Graphe, on va redessiner le Line Graphe avec les plages de données qui nous intéressent. Le principe du zoom consiste à changer le domaine et l'axe horizontal de *focus* et de mettre à jour les *paths*. On va ainsi changer l'échelle temporaire de *focus*. Par contre l'échelle verticale reste inchangée.

- Implémenter le zoom via un brush pour modifier l'échelle horizontale de *focus* (fonction **brush**).
- (Bonus) Implémenter dans **brush** une modification de l'échelle verticale.

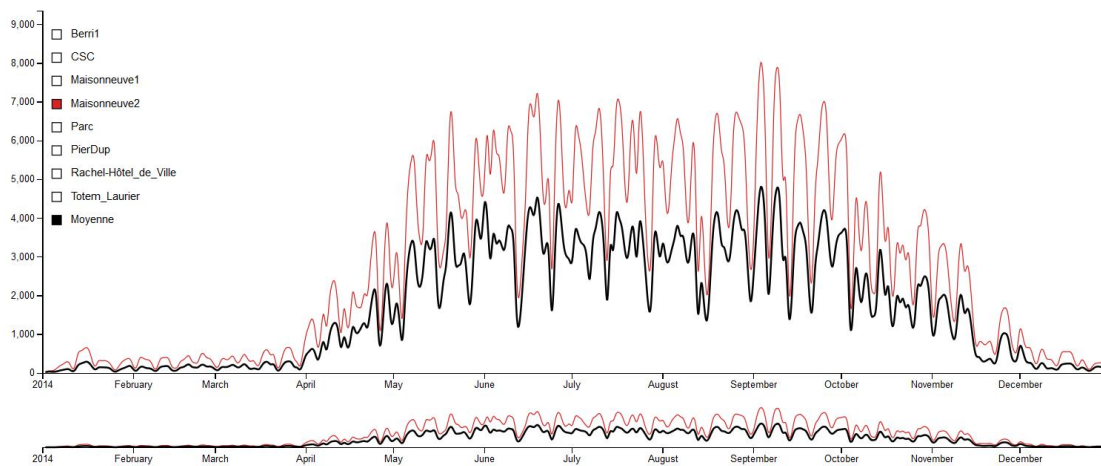
## d. Légende

Pour simplifier la lecture du graphique, on va implémenter une légende permettant de relier la couleur des lignes au nom des rues. Cette légende doit être "par-dessus" le line chart. Les couleurs de la légende doivent correspondre aux lignes du line chart.



Line chart avec légende

Le nombre de rue à afficher étant assez important, il peut être difficile de bien lire en entier une seule ligne. Ainsi on souhaite pouvoir simplement afficher ou masquer les rues. On va donc implémenter une méthode qui lorsque l'on clique sur un carré va afficher ou masquer la ligne en fonction de son état actuel (masquée ou affichée). On va également masquer dans la légende la couleur des lignes masquées. Par contre on doit laisser un cadre noir visible pour savoir ou cliquer à nouveau pour afficher la ligne après l'avoir masquée.



Line chart où on n'affiche que la rue Maisonneuve2 et la moyenne

Dans cette partie il faut :

- Afficher la légende à gauche du graphe principal (fonction **legend**).
- Rendre cette légende dynamique (fonction **dispThis**).



### 3. Fichiers Fournis

|                 |                             |
|-----------------|-----------------------------|
| TP2.html        | Squelette HTML du TP        |
| données2000.csv | Données à utiliser          |
| d3.js           | Librairie de d3 du tool-tip |
| 1-preproc.js    | Code de la partie a         |
| 2-lineGraph.js  | Code de la partie b         |
| 3-brush.js      | Code de la partie c         |
| 4-legend.js     | Code de la partie d         |

### 4. Travail demandé

Compléter le code JavaScript des différentes parties pour afficher un scatter plot. A savoir :

- Le code de preproc.js.
- Le code de lineGraph.js.
- Le code de brush.js.
- Les codes de legend.js.