



INF8808 - Visualisation de données

TP4 : Bar Chart et Chord Diagram

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 des bar charts et un diagram chord depuis une base de données CSV et de comparer les deux visualisations. Il est recommandé d'avoir lu les chapitres 10 et 11 du livre de Scott Murray avant de commencer le TP. De plus, le chord diagram n'étant pas expliqué, il est fortement recommandé d'avoir vu l'explication de Mike Bostock sur les chord diagrams : <https://bl.ocks.org/mbostock/1046712>. Les données utilisées proviennent du site officiel de BIXI : <http://montreal.bixi.com/fr/donnees-ouvertes> (données d'Août 2015).

b. Bar chart, Chord Diagram

Un **Bar Chart** (ou histogramme) est un graphe représentant les données avec des colonnes. C'est l'un des graphes les plus simples à dessiner et il permet de faire des comparaisons entre plusieurs catégories.

Un **Chord Diagram** (ou diagramme de cordes) est un graphe plus complexe que le précédent et permet de représenter des interactions entre différentes catégories à travers des cordes allant d'une zone d'un cercle à une autre zone.

c. Détail du TP

Le travail pratique est constitué de deux fichiers HTML, de fichiers JavaScript contenant le code et d'un fichier CSV contenant les données à afficher. Le fichier CSV contient l'ensemble des trajets réalisés entre 10 stations BIXI en un mois à Montréal, plus précisément en Août 2015. Chaque ligne du CSV possède de nombreux attributs plus ou moins utiles. Dans notre cas, nous allons surtout nous intéresser à la station de départ et la station d'arrivée des trajets. Il est à noter que le départ et l'arrivée peuvent être identiques. On va donc visualiser ces données avec deux outils : un bar chart pour chaque station de départ et un chord diagram pour l'ensemble des données. Nous aurons ainsi la possibilité de comparer deux visualisations d'une même donnée. Nous utiliserons les layouts **histogram** et **chord** de d3 pour réaliser ces deux graphes. Les parties traitant des bar charts et du chord diagram sont totalement indépendantes.

2. Travail à réaliser – Bar chart

a. Prétraitement des données

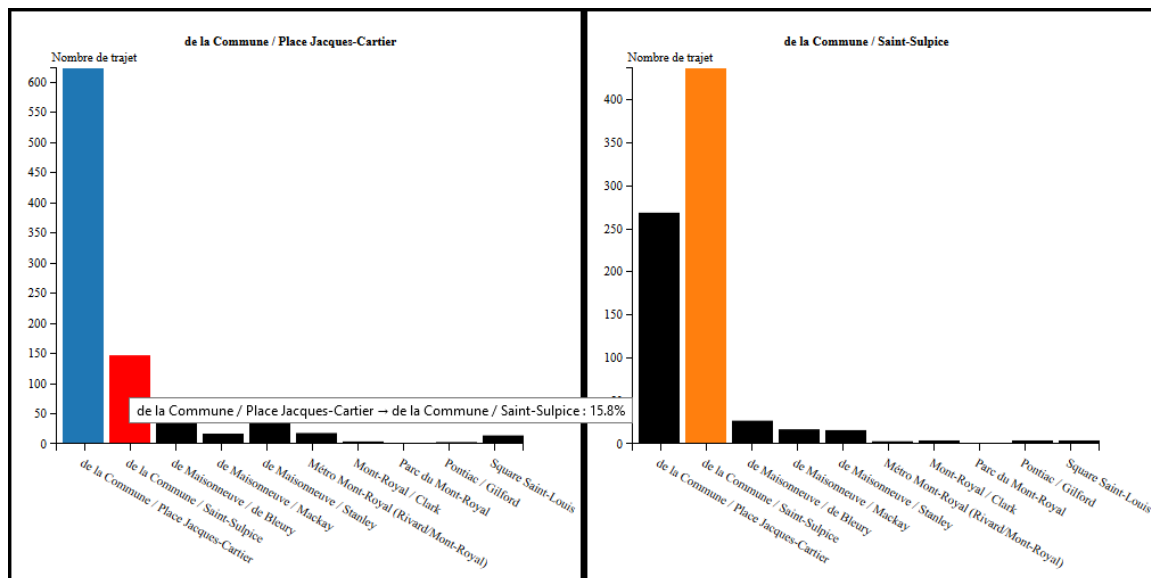
Il faut dans un premier temps convertir les données CSV pour pouvoir les utiliser avec le layout **histogram**. Nous créerons aussi dans cette partie les classiques fonctions scales.

- Compléter les fonctions contenues dans le fichier *barChartPreproc.js*.

b. Création des Bar charts

Pour ne pas écrire 10 fois le code, nous allons créer une fonction permettant de créer un bar chart en fonction des données à visualiser. Chaque bar chart possède les caractéristiques suivantes :

- Le titre du bar chart correspond à la station de départ.
- La légende correspond aux stations d'arrivées.
- Les colonnes sont en noires sauf celle correspondant à la station de départ.
- En passant la souris sur une colonne, celle-ci se colore en rouge et un texte donnant le pourcentage du trajet au sein de la station apparaît.



Bar chart pour les voyages au départ de Place Jacques-Cartier et Saint-Sulpice. On a sélectionné la colonne Saint-Sulpice dans l'histogramme Place Jacques Cartier

Dans cette partie il faut :

- Compléter les fonctions contenues dans le fichier *barChart.js*.

3. Travail à réaliser – Chord Diagram

a. Prétraitement des données

Un chord diagram utilise une matrice de liaison pour afficher les données. Une matrice de liaison est une matrice de taille $n \times n$ où n est le nombre de paramètre à considérer. Dans notre cas, cette matrice sera modélisée par un tableau de tableau (`Array[Array]`) de taille $n \times n$ où n est le nombre de stations de vélos considérées et les coordonnées $[i][j]$ correspondront au nombre de trajet partant de la station i et allant à la station j . Il est à noter que, de manière générale, $[i][j] \neq [j][i]$. Comme pour la partie a du bar chart, il faut donc dans un premier temps modifier les données du CSV afin de les convertir en une matrice de liaison.

- Compléter les fonctions contenues dans le fichier *chordDiagramPreproc.js*.

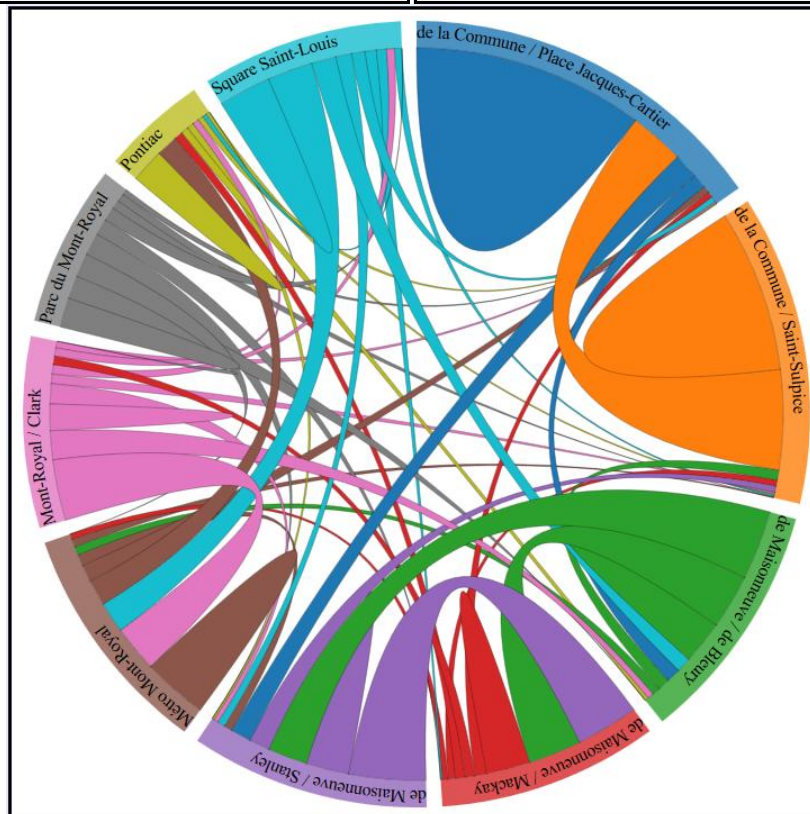
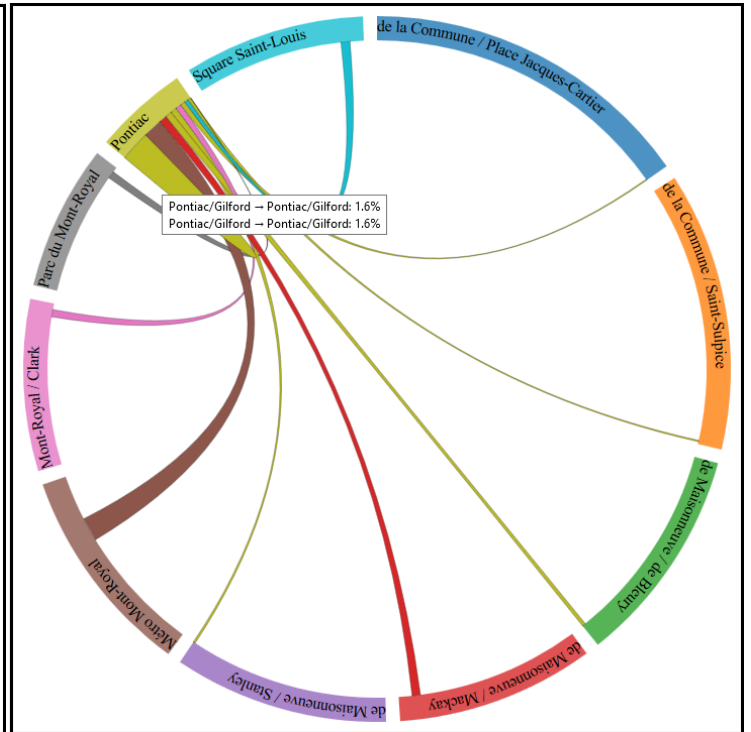
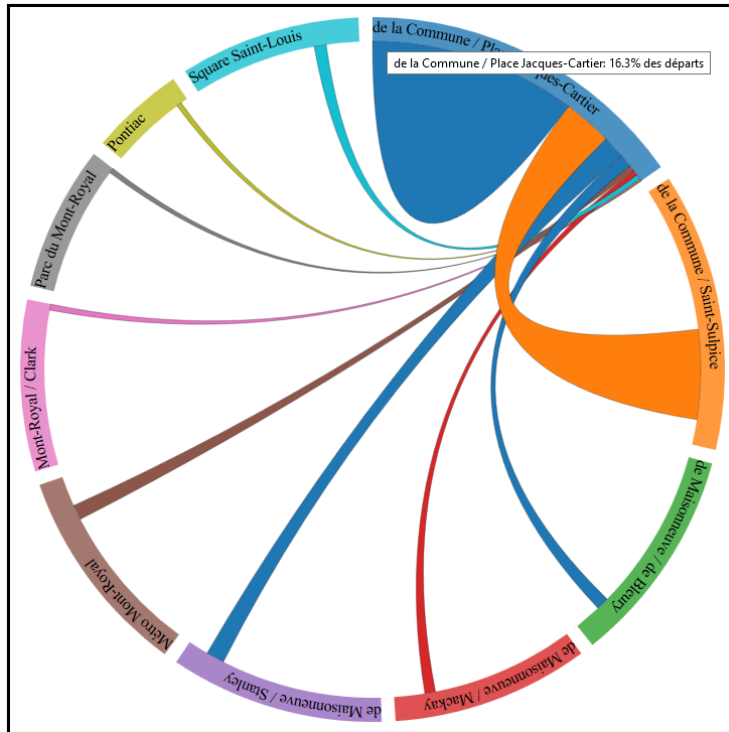
b. Création du Chord Diagram

Possédant la matrice de liaison, la création du chord diagram est simple : il suffit d'utiliser le layout **chord**. Le chord diagram doit avoir les propriétés suivantes :

- Une couleur par station, identique à celle du bar chart.
- En passant la souris sur une zone correspondante à une station, les autres zones doivent disparaître.
- Il doit y avoir le nom de la station sur les bords du chord diagram.
- En passant la souris sur le titre d'une station, on doit avoir le pourcentage de départ depuis cette station par rapport au nombre total de trajet.
- En passant la souris sur une corde, on doit avoir le pourcentage de trajet dans un sens puis dans l'autre. Le plus grand pourcentage doit être affiché en premier.
- La couleur de la corde doit être celle de la station cumulant le plus de départ entre les deux stations reliées.
- Si un nom de station est trop grand, pour être affiché, il doit être tronqué.

Dans cette partie il faut :

- Compléter les fonctions contenues dans le fichier *chordDiagram.js*.



Chord Diagram complet et avec une station sélectionnée ou un trajet sélectionné



4. Fichiers Fournis

TP4.html	Squelette HTML du TP
data.csv	Données à utiliser
d3.js	Librairie de d3.js
barChartPreproc.js et barChart.js	Code pour les bar charts
chordDiagramPreproc.js et chordDiagram.js	Code pour le chord diagram

5. Travail demandé

Compléter le code JavaScript des différentes parties pour visualiser les bar charts et le chord diagram, à savoir :

- Le code de barChartPreproc.js et barChart.js.
- Le code de chordDiagramPreproc.js et chordDiagram.js.