



**POLYTECHNIQUE
MONTRÉAL**

LE GÉNIE
EN PREMIÈRE CLASSE

INF8808 – Visualisation de données

Travail pratique 4

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 bandes (*bar chart*) et un [diagramme à cordes](#) (*chord diagram*) à partir de données ouvertes provenant d'un fichier JSON. L'utilisateur aura la possibilité de modifier les données utilisées par le diagramme à bandes et pourra sélectionner les éléments à mettre en évidence dans le diagramme à cordes.

Avant de commencer ce travail, il est recommandé d'avoir lu le chapitre 11 du livre de Scott Murray [1].

2 Introduction

Un diagramme à bandes est un graphe représentant les données avec une barre. La hauteur d'une barre permet d'encoder la valeur à représenter. C'est l'un des graphes les plus simples à dessiner et il permet de faire des comparaisons entre plusieurs catégories. Le diagramme à cordes, quant à lui, est beaucoup plus complexe et permet de visualiser des liens (cordes) entre plusieurs catégories (arcs de cercle). La largeur d'une corde permet d'encoder la valeur partant d'une catégorie et se dirigeant vers une autre. La longueur d'un arc de cercle, qui représente une catégorie, encode le cumulatif des valeurs des cordes entrantes et sortantes pour cette même catégorie.

En ce qui a trait à ce travail pratique, vous aurez à réaliser les deux types de graphiques mentionnés en utilisant les données sur le nombre de trajets réalisés entre dix stations de BIXI pour le mois d'août 2015 à Montréal. Ces données proviennent du [portail des données ouvertes](#) de BIXI Montréal. Celles-ci ont été regroupées dans le fichier «`bixi-destinations.json`» se trouvant dans le dossier «`data`».

Le fichier JSON fourni est constitué d'une liste qui contient dix éléments, c'est-à-dire un élément pour chacune des stations de BIXI. Chacun de ces éléments comporte deux propriétés : «`name`» et «`destinations`». La propriété «`name`» correspond au nom de la station de départ alors que la propriété «`destinations`» est une liste indiquant le nombre de trajets qui a été effectué vers des stations d'arrivée à partir de la station de départ. Par souci de clarté, la Figure 1 montre le format d'un élément de la liste du fichier JSON fourni. Dans cet exemple, la station de départ est «`Mont-Royal / Clark`», 71 trajets sont retournés à la station de départ tandis que 22 autres se sont dirigés à la station «`Pontiac / Gilford`».

```

{
  "name": "Mont-Royal / Clark",
  "destinations": [
    {
      "name": "Mont-Royal / Clark",
      "count": 71
    },
    {
      "name": "Pontiac / Gilford",
      "count": 22
    }
  ]
}

```

FIGURE 1 – Exemple du format utilisé pour les éléments de la liste du fichier JSON fourni

3 Travail à réaliser

Pour ce travail pratique, vous devrez compléter le code JavaScript nécessaire à la réalisation des deux graphiques demandés qui utiliseront les données de BIXI Montréal. L’affichage du diagramme à bandes se fera sous l’onglet « Vue détaillée » alors que celui du diagramme à cordes s’effectuera sous l’onglet « Vue générale ».

Les sous-sections qui suivent présentent les différentes parties qui devront être réalisées pour ce travail. Il est à noter qu’il est nécessaire d’effectuer la première partie de ce travail (prétraitement des données) avant de réaliser les deux suivantes, 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 définir les différentes échelles qui seront utilisées par le diagramme à bandes. Également, vous devrez créer la matrice d’adjacence nécessaire pour dessiner le diagramme à cordes.

Une matrice d’adjacence est une matrice de taille $n \times n$ où n est le nombre de paramètres 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 BIXI considérées. Un élément $m_{i,j}$ de la matrice correspond au nombre de trajets partant de la station i et allant à la station j . Il est à noter que de manière générale, $m_{i,j} \neq m_{j,i}$. À titre d’exemple, la

Figure 2 montre une représentation équivalente entre une matrice d'adjacence (2a) et un graphe orienté (2b) afin d'illustrer le passage d'une forme à une autre.



(a) Matrice d'adjacence

(b) Graphe orienté

FIGURE 2 – Exemple de deux représentations équivalentes

Contrairement aux deux derniers travaux pratiques, vous ne devrez pas effectuer de conversion sur les données fournies. En effet, puisque les données proviennent d'un fichier au format JSON, il n'est pas nécessaire de réaliser des conversions de types puisque cette information est conservée dans le format de stockage contrairement au format CSV.

Pour réaliser cette partie, 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 x et y (fonctions « `domainX` » et « `domainY` »);
- Créer la matrice d'adjacence nécessaire (fonction « `getMatrix` »);
- Calculer le nombre total de trajets réalisés (fonction « `getTotal` »).

3.2 Réalisation du diagramme à bandes

Pour cette deuxième partie, vous aurez à réaliser le diagramme à bandes qui affichera le nombre de trajets vers d'autres stations pour une station de départ spécifié. Vous devrez compléter les fonctions se trouvant dans le fichier « **2-bar-chart.js** » afin que le tout soit fonctionnel. Les sous-sections suivantes détaillent les étapes qui devront être complétées pour réaliser le diagramme.

3.2.1 Création du diagramme

La première étape consiste en la création du diagramme à bandes pour les données ayant comme station de départ «De la Commune / Place Jacques-Cartier» (par défaut). Pour ce faire, vous devrez dessiner les barres du graphique pour chacune des stations d'arrivée en vous assurant que la couleur de la barre correspond à la couleur de la station associée. Également, vous aurez à dessiner les axes x et y du graphique. Par souci de clarté, la Figure 3 montre à quoi devrait ressembler le graphique obtenu une fois cette étape complétée.

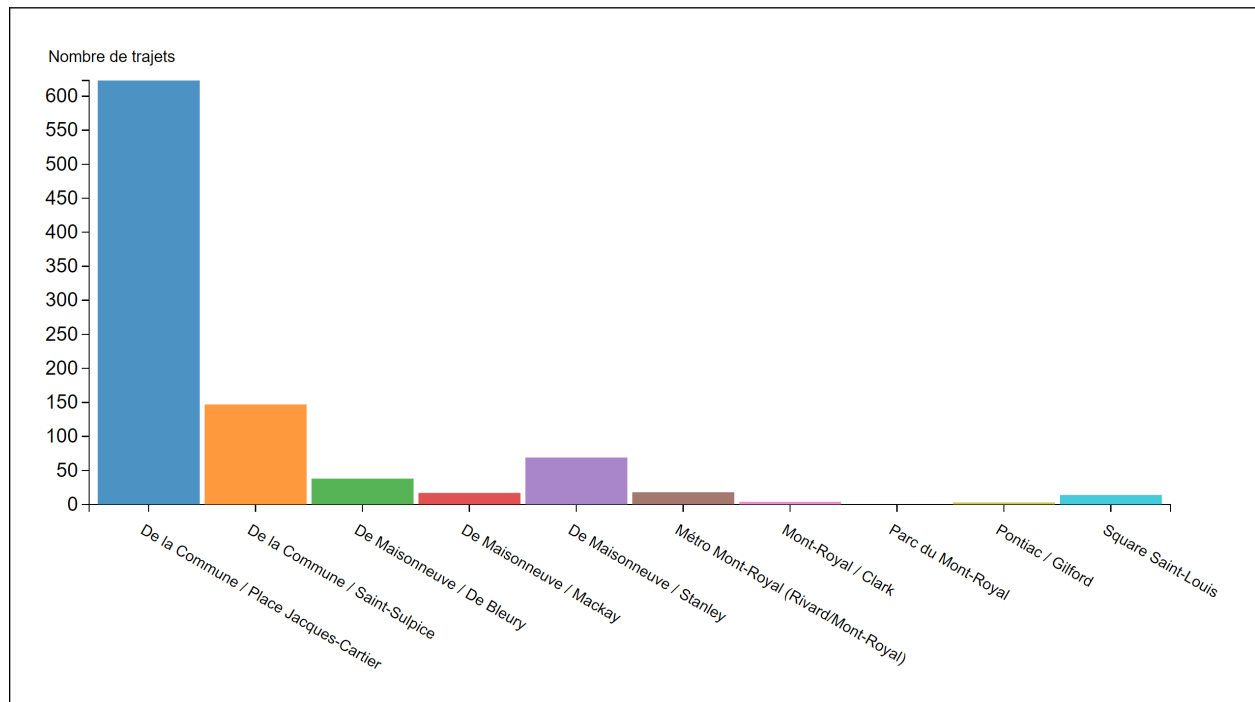


FIGURE 3 – Diagramme à bandes utilisant comme station de départ «De la Commune / Place Jacques-Cartier»

Pour la création des axes, vous pouvez utiliser les classes «axis x» et «axis y» avec les groupes responsables de dessiner les axes x et y respectivement.

Pour réaliser cette partie, vous devrez compléter les éléments suivants :

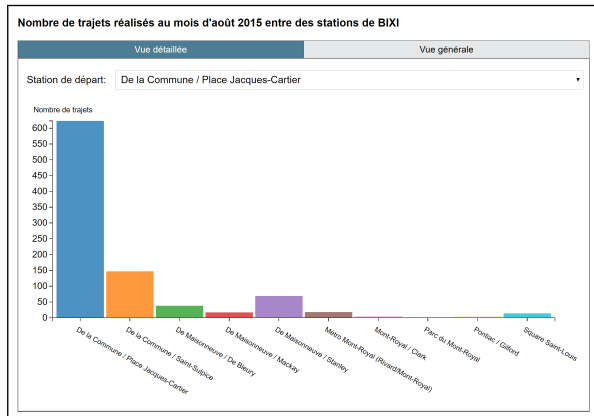
- Dessiner les axes x et y associées au graphique (fonction «createAxes»);
- Dessiner les barres du graphique (fonction «createBarChart»).

Attention

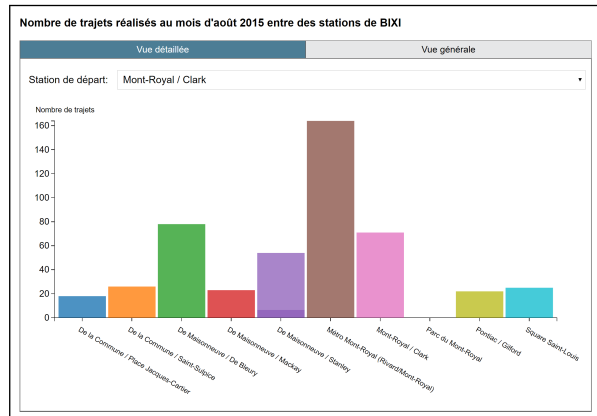
Assurez-vous d'indiquer un titre à l'axe des y du graphique (p. ex. « Nombre de trajets »). Également, assurez-vous que l'orientation des noms des stations sur l'axe des x soit à 30 degrés (voir Figure 3).

3.2.2 Réalisation de la transition

La deuxième étape consiste à réaliser une transition lorsque les données utilisées par le graphique sont modifiées. Lorsqu'une nouvelle station de départ est sélectionnée par l'utilisateur via la liste déroulante, les données du diagramme doivent être mises à jour pour utiliser celles associées à la station de départ choisie. En ce sens, une transition d'une durée d'une seconde doit être réalisée pour mettre à jour la hauteur des barres ainsi que la graduation de l'axe des y . La Figure 4 illustre d'ailleurs l'affichage attendu pour les stations de départ « De la Commune / Place Jacques-Cartier » (4a) et « Mont-Royal / Clark » (4b).



(a) De la Commune / Place Jacques-Cartier



(b) Mont-Royal / Clark

FIGURE 4 – Diagramme à bandes utilisant différentes données

Pour réaliser la fonctionnalité demandée, vous aurez à compléter la fonction « transition ».

3.2.3 Implémentation de l'infobulle

La troisième étape à réaliser consiste en l'ajout d'une infobulle lorsqu'une barre du graphique est survolée par la souris. Pour ce faire, vous devrez compléter le texte qui devra

être affiché lors de ce survol en indiquant le nombre de trajets associés à la station ainsi que le pourcentage correspondant au nombre de trajets arrivant à cette station par rapport au nombre total de trajets partant de la station de départ. Le rendu de l’infobulle devrait être similaire à la Figure 5 (données utilisant comme station de départ «De Maisonneuve / Stanley»). Il est à noter que le pourcentage doit être formaté en utilisant la fonction « `formatPercent` » qui est fournie.

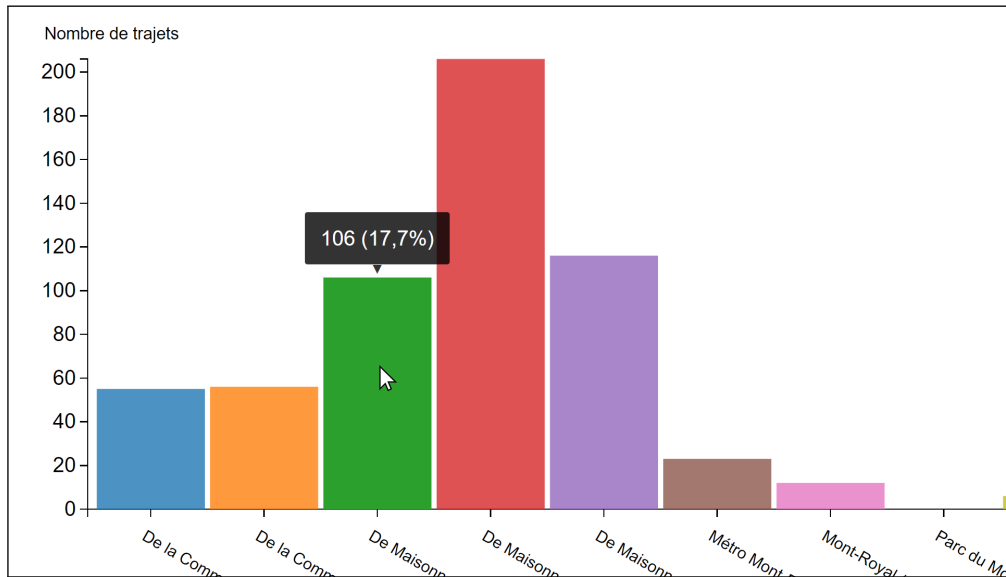


FIGURE 5 – Infobulle affichée lorsqu’une barre est survolée par la souris

Pour implémenter la fonctionnalité demandée, vous aurez à compléter la fonction « `getToolTipText` ».

3.3 Réalisation du diagramme à cordes

Pour cette troisième et dernière partie, vous devrez réaliser le diagramme à cordes qui permettra de visualiser le nombre de trajets effectués entre les dix stations de BIXI. Afin que le tout fonctionne correctement, vous devrez compléter les fonctions se trouvant dans le fichier « **3-chord-diagram.js** ». Pour vous aider, vous pouvez jeter un coup d’œil à cet [exemple de code](#) pour créer ce type de graphique en utilisant D3. Les sous-sections qui suivent décrivent les étapes à réaliser pour obtenir le diagramme demandé.

3.3.1 Création des groupes

La première étape consiste en la création des différents groupes utilisés par le diagramme. Un groupe correspond à un arc de cercle et est associé à une station de BIXI. Chacun des groupes devra être colorié de la couleur de la station qui lui est associée et devra indiquer le nom de la station en blanc. De plus, le nom des stations devra suivre la courbure de l'arc de cercle du groupe en utilisant un élément « `textPath` ». Dans le cas où un nom de station de BIXI serait trop long, celui-ci devra être tronqué afin que le texte puisse entrer dans le groupe (nécessaire pour les stations « Métro Mont-Royal [Rivard/Mont-Royal] » et « Pontiac / Gilford »). La Figure 6 illustre l’affichage qui est attendu une fois la création des groupes complétée.



FIGURE 6 – Groupes du diagramme à cordes

Lorsqu'un groupe sera survolé par la souris, celui-ci devra afficher une infobulle en utilisant la balise « `title` ». Cette infobulle devra indiquer le nom de la station ainsi que le pourcentage de trajets ayant comme départ cette station par rapport au nombre total de

trajets. La Figure 7 montre d'ailleurs le format de l'infobulle qui est attendu. Assurez-vous d'utiliser la fonction « `formatPercent` » pour formater correctement le pourcentage.

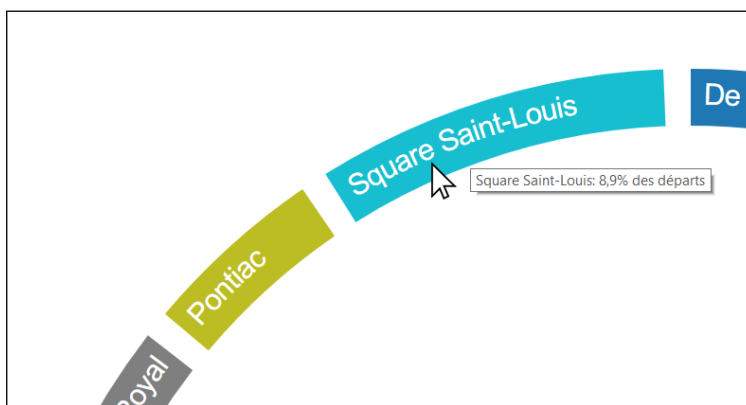


FIGURE 7 – Infobulle à afficher lorsqu'un groupe est survolé par la souris

Pour réaliser cette étape, vous aurez à compléter la fonction « `createGroups` ».

3.3.2 Création des cordes

La deuxième étape consiste à créer les cordes du diagramme qui permettront de représenter le nombre de trajets entre des stations de BIXI. Chacune des cordes devra être coloriée de la couleur de la station cumulant le plus de départs entre les deux stations reliées et devra avoir une opacité de 80 %. La Figure 8 montre le graphique qui devrait être obtenu une fois la création des cordes complétée.

Lorsqu'une corde sera survolée par la souris, l'opacité de cette dernière devra passer à 95 % et une infobulle devra être affichée grâce à la balise « `title` ». Cette infobulle devra indiquer le pourcentage du nombre de trajets réalisé dans un sens comme dans l'autre. Par souci de clarté, la Figure 9 montre à quoi devrait ressembler l'infobulle demandée. Assurez-vous de formater les pourcentages en utilisant la fonction « `formatPercent` ».

Pour réaliser cette étape, vous aurez à compléter la fonction « `createChords` ».

3.3.3 Sélection d'un groupe à afficher

La troisième étape consiste à afficher uniquement les cordes d'un groupe en particulier lorsque ce dernier est survolé par la souris dans le but de faciliter la lisibilité du graphique.

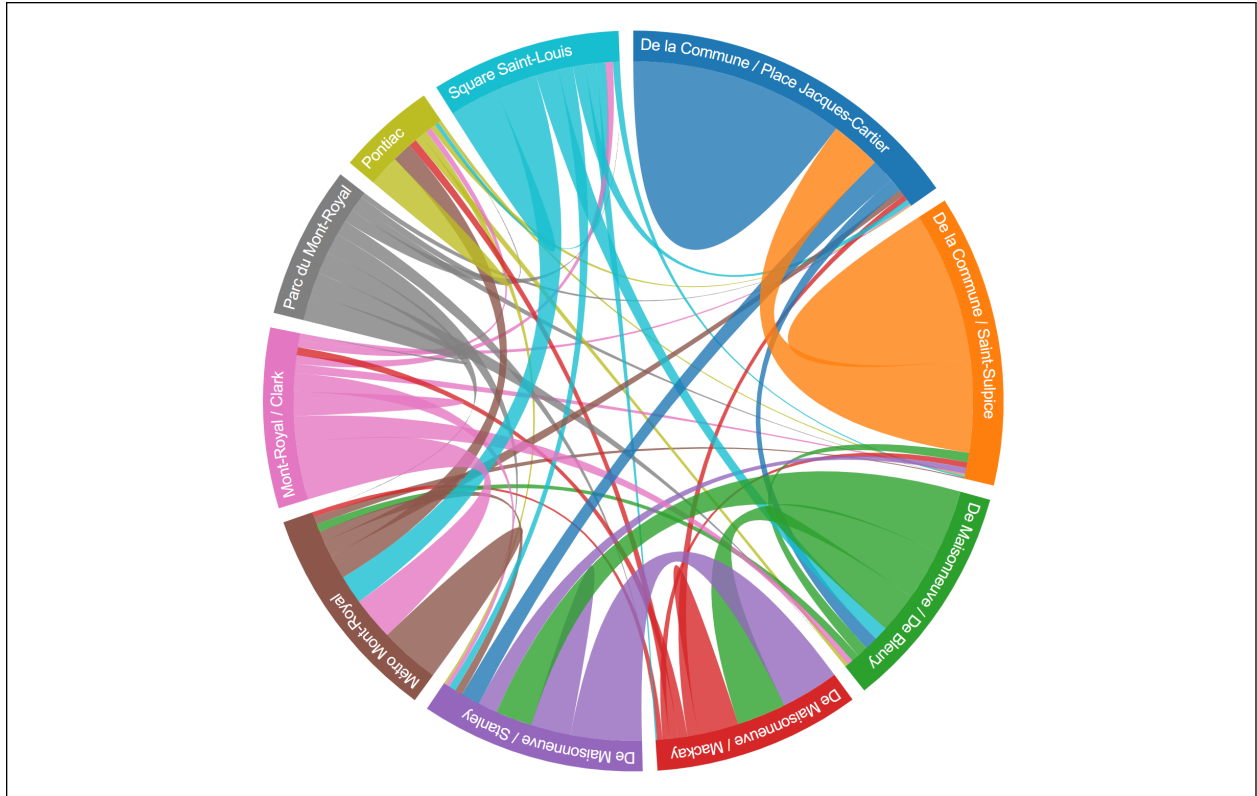


FIGURE 8 – Diagramme à cordes utilisant les données fournies

En ce sens, lorsqu'un groupe est survolé par la souris, les cordes entrantes et sortantes de ce groupe devront être affichées avec une opacité de 80 % alors que l'opacité de toutes les autres devra être modifiée à 10 %. Cette sélection devra être conservée tant que la souris se trouve à l'intérieur du cercle du graphique. La Figure 10 illustre à quoi devra ressembler la sélection d'une station particulière. Lorsque la souris sortira du cercle, l'affichage du diagramme doit être réinitialisé à celle par défaut, tel qu'illustré à la Figure 8.

Pour réaliser cette fonctionnalité, vous aurez à compléter la fonction « initializeGroups-
Hovered ».

i Pour vous aider

Afin de vous simplifier la tâche, vous pouvez réaliser cette fonctionnalité en utilisant des classes CSS.

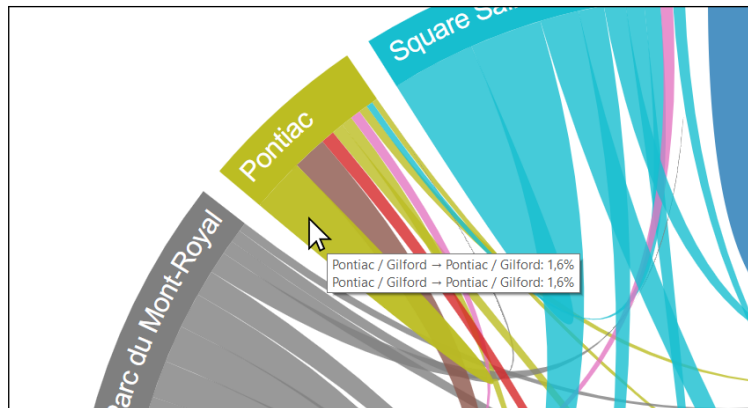


FIGURE 9 – Infobulle à afficher lorsqu’une corde est survolée par la souris

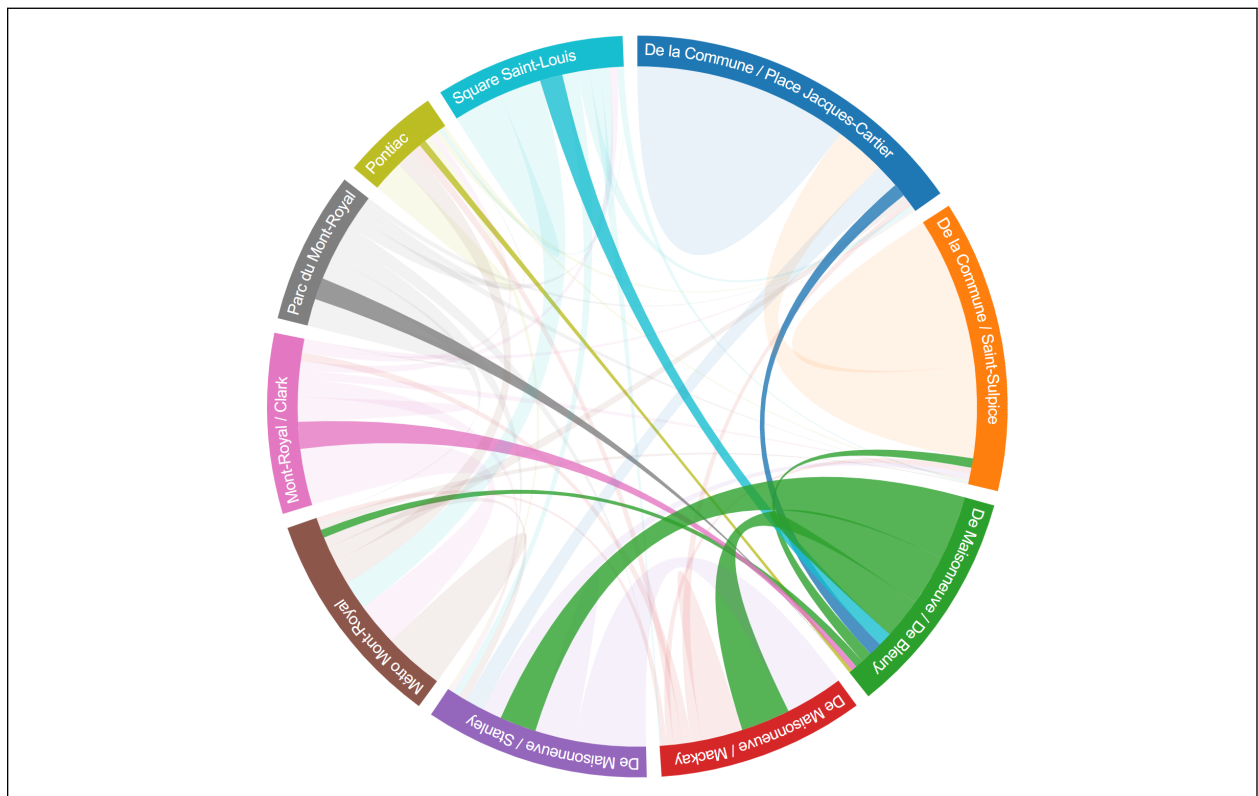


FIGURE 10 – Affichage des informations de la station « De Maisonneuve / De Bleury »

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 « TP4_matricule1_matricule2.zip ».
2. Le travail doit être remis avant **8h30**, le **21 février 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	3
Réalisation du diagramme à bandes	
Création du diagramme	4
Réalisation de la transition	1
Implémentation de l'infobulle	1
Réalisation du diagramme à cordes	
Création des groupes	4
Création des cordes	3
Sélection d'un groupe à afficher	3
Qualité et clarté du code	1
Total	20

Ce travail pratique a une pondération de **10 %** 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.