



INF8808 - Visualisation de données

TP5 : Cartographie

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 une carte du Canada avec les données des résultats des élections de 2015 et une carte du Québec avec des données géographiques. Il est recommandé d'avoir lu le chapitre 12 du livre de Scott Murray avant de commencer le TP. Les données utilisées proviennent du site des élections canadiennes : <http://www.elections.ca/res/rep/off/ovr2015app/accueil.html>.

Les cartes utilisées dans ce TP proviennent des deux sources suivantes :

- Carte du Québec et données :
<https://www.mern.gouv.qc.ca/territoire/portrait/portrait-donnees-mille.jsp>.
- Carte des circonscriptions 2015 du Canada :
<http://open.canada.ca/data/en/dataset/56124851-71fc-4f94-8df2-40f59cd1dd46>.
- Résultat des élections fédérales de 2015 :
<http://www.elections.ca/res/rep/off/ovr2015app/accueil.html>.

b. Cartographie, TopoJSON, Leaflet

D3.js permet de réaliser des cartes en utilisant le format GeoJSON. Il est également possible d'utiliser D3 pour ajouter des éléments par-dessus d'autres cartes existantes comme par exemple Google Maps. Pour cela nous utiliserons la bibliothèque Leaflet.js (version 0.7).

c. Détail du TP

Le travail pratique est constitué de deux dossiers permettant respectivement la réalisation d'une carte du Canada et d'une Carte du Québec.

Le dossier concernant la carte du Canada contient un fichier HTML et un fichier JavaScript contenant le code, un fichier CSV contenant les résultats des dernières élections fédérales et un fichier JSON modélisant les circonscriptions du Canada.

Le dossier concernant la carte du Québec contient un fichier HTML et un fichier JavaScript contenant le code ainsi que trois fichiers JSON contenant respectivement :

- Un découpage administratif du Québec.
- Les routes traversant le Québec.
- Les aéroports et hydrobases du Québec.

2. Travail à réaliser

a. Carte du Canada

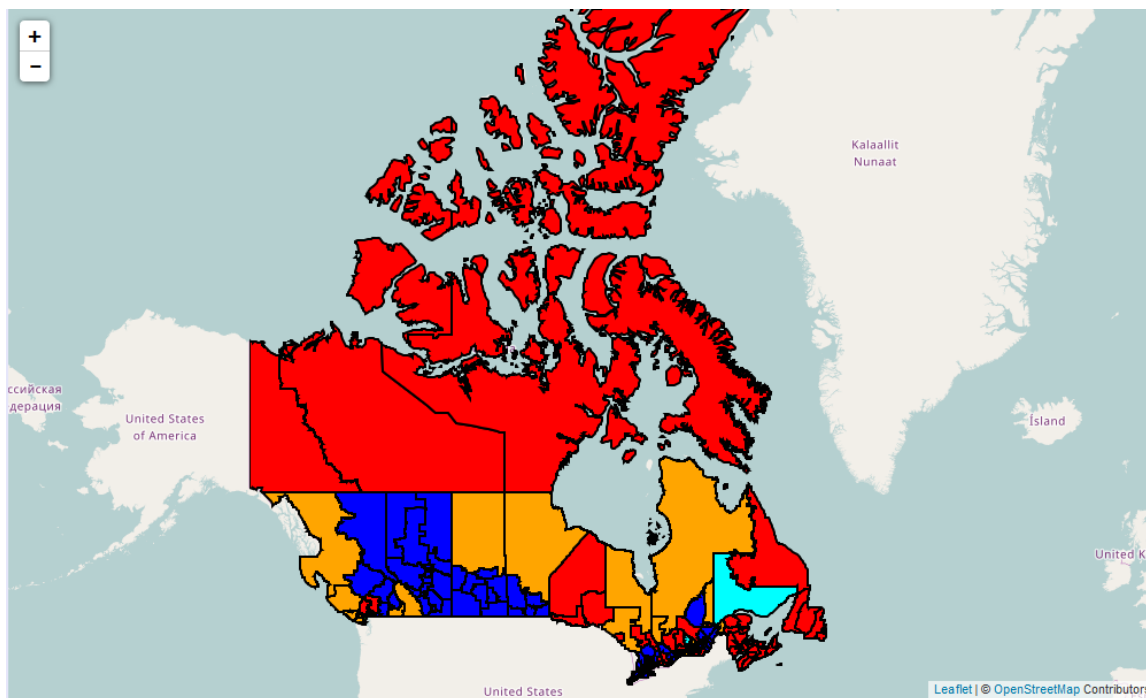
Le fichier CSV contient les résultats de tous les candidats de toutes les circonscriptions et ces résultats sont affichés par circonscription. Il va donc falloir dans un premier temps extraire dans chaque circonscription le gagnant en gardant toutes les informations le concernant.

Les circonscriptions seront dessinées avec des **path** et posséderont les attributs suivants :

Attribut	Signification
fill	Couleur du parti du candidat vainqueur
candidat	Nom du candidat vainqueur
parti	Nom du parti du vainqueur
pourcentage	Pourcentage de voix obtenu par le candidat
numeroCirconscription	Numéro de la circonscription
nomCirconscription	Nom de la circonscription

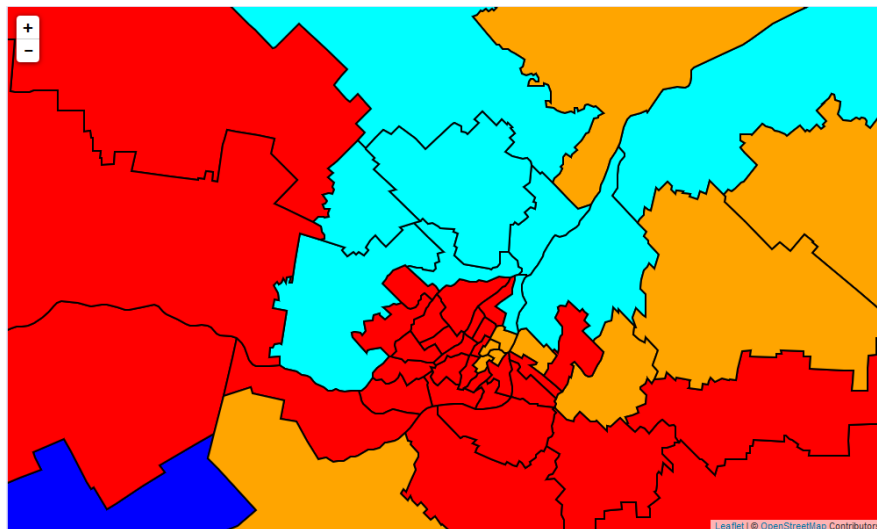
Ces attributs permettront de réaliser plus facilement la tool-tip.

Le contour des circonscriptions doit être lisible. La carte devrait alors être équivalente à la suivante :



Carte du Canada

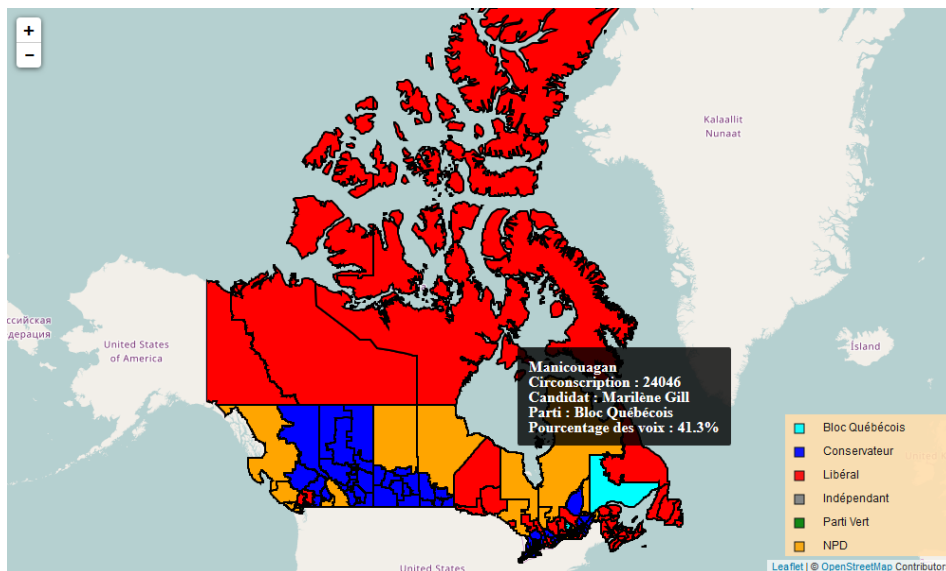
De plus la carte doit pouvoir s'actualiser en cas de zoom :



Zoom sur Montréal

Il ne reste plus qu'à créer la légende et rajouter la tool-tip qui s'affichera pour chaque circonscription. La légende doit se situer en bas à droite et la tool-tip doit contenir les informations suivantes :

- Le numéro de la circonscription.
- Le nom de la circonscription.
- Le nom du candidat élu.
- Le parti du candidat élu.
- Le pourcentage qu'a obtenu le candidat.



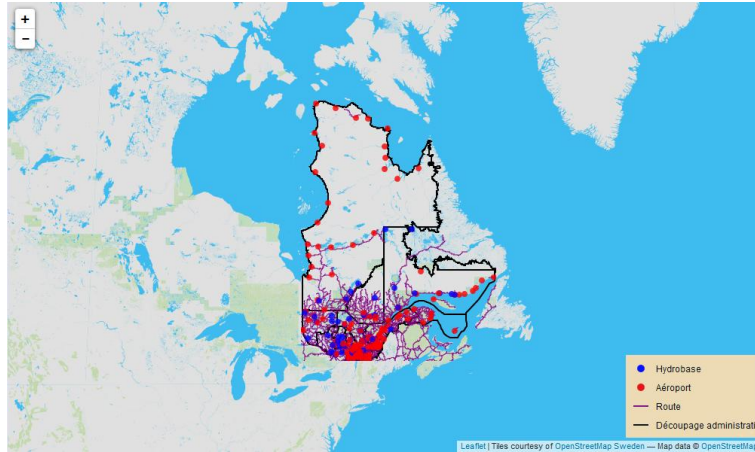
Carte du Canada avec légende et tool-tip

Enfin le SVG contenant la légende doit être légèrement transparent.

b. Carte du Québec

La carte du Québec devra afficher la province, les routes et les aéroports du Québec. Il faut globalement réaliser le même travail que pour la carte du Canada à savoir. Voici donc des indications concernant le rendu visuel à obtenir :

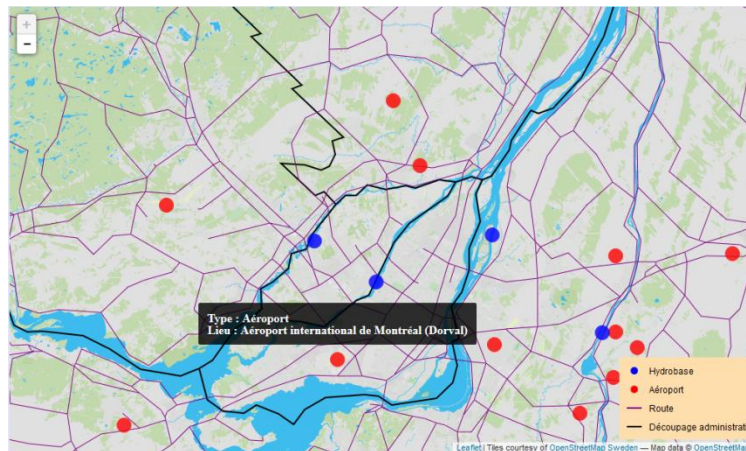
- Affichage des données
 - **Toutes** les données à afficher sont représentées par des **paths**.
 - Les routes seront de couleur violette et de largeur 1 pixel. Si le zoom de Leaflet est très faible (inférieur à 3), on ne fait pas apparaître les routes.
 - Le découpage du Québec sera noir et de largeur 1 pixel si le zoom est très faible et de 2 sinon.
 - Pour les aéroports, on utilisera des paths pour afficher des cercles. La valeur de l'attribut **d** sera alors **path.pointRadius(r)** où **r** correspond au rayon du cercle. Ce rayon est égal au niveau du zoom pour avoir des cercles de taille constante lors des zooms. Si le zoom est très faible, les aéroports ne sont pas affichés. Les aéroports seront de couleur rouge et les hydrobases seront de couleur bleu.
 - Il faudra ajouter des attributs pour les aéroports afin d'afficher sur la tool-tip le lieu et le type de l'aéroport. Les variables associées à ces deux informations dans le GeoJSON des aéroports sont respectivement : **TRP_NM_TOP** et **TRP_DE_IND**.
- Affichage de la légende
 - La légende doit afficher les types d'aéroports, les routes et le découpage.
 - Pour les aéroports, vous pouvez dessiner un cercle plutôt qu'un path.
 - Pour les routes et aéroport, vous pouvez dessiner des lignes.
- Affichage de la tool-tip
 - La tool-tip ne doit s'afficher que pour les aéroports.
 - Elle doit afficher le lieu et le type de l'aéroport.



Carte du Québec



Carte du Québec (dézoomée). L'image a été tronquée



Carte du Québec (zoom sur Montréal)



3. Fichiers Fournis

Canada.html et Quebec.html	Squelettes HTML du TP
canada.js, leafletLayout.js et legendAndTip.js	Codes pour la carte du Canada
data.csv ; canada.json	Données et carte geoJSON pour le Canada
canada.js, leafletLayout.js et legendAndTip.js	Codes pour la carte du Québec
aerobase, quebec et routes.geojson	Données pour la carte du Québec
d3.js, d3-tip.js, topojson.js et queue.js	Librairies de d3
leaflet.js	Libraires de Leaflet

4. Travail demandé

Compléter le code JavaScript des différentes parties pour visualiser la carte du Canada et la carte du Québec à savoir :

- Le code de *canada.js*, *leafletLayout.js* et *legendAndTip.js* pour la carte du Canada.
- Le code de *quebec.js*, *leafletLayout.js* et *legendAndTip.js* pour la carte du Québec.