INF8808 – Visualisation de données

Travail pratique 5

Auteurs:

Adrien Dessinges et Antoine Béland

Chargée de laboratoire:

Olivia Gélinas

Hiver 2020

Département de génie informatique et génie logiciel

1 Objectifs

Le but de ce travail pratique est de réaliser une carte choroplèthe en utilisant les bibliothèques D3.js et Leaflet (version 0.7.7) à partir de données ouvertes provenant d'un fichier CSV et d'un fichier TopoJSON. L'utilisateur aura la possibilité d'obtenir des informations précises sur une zone de la carte ainsi que de rechercher une zone en particulier.

Avant de commencer ce travail, il est recommandé d'avoir lu le chapitre 12 du livre de Scott Murray [1].

2 Introduction

Une carte choroplèthe est une carte où les régions sont coloriées, ombragées ou remplies d'un certain motif pour montrer une certaine valeur numérique ou une catégorie. Ce type de carte permet de comparer facilement des valeurs entre différentes régions géographiques et facilite la visualisation de tendances et de différences en fonction de l'emplacement.

En ce qui a trait à ce travail pratique, vous aurez à réaliser une carte choroplèthe à partir des résultats des élections fédérales de 2015 pour les 338 circonscriptions se trouvant à travers le pays. Les résultats détaillés pour chacune des circonscriptions proviennent d'Élections Canada, alors que les tracés des différentes circonscriptions viennent du portail des données ouvertes du Gouvernement du Canada. Puisque ces données proviennent de deux sources différentes, vous aurez à manipuler deux fichiers utilisant différents formats.

Le premier fichier que vous aurez à manipuler, soit « `data.csv` » qui se trouve dans le dossier « `data` », contient l'ensemble des résultats pour toutes les circonscriptions du pays. Chacune des lignes de ce fichier correspond aux résultats obtenus par un candidat qui s'est présenté dans une circonscription précise. Cela comprend les informations suivantes :

— Le nom de la circonscription (« `name` ») ;

— Le numéro de la circonscription (« `id` ») ;

— Le nombre de votes obtenus par le candidat (« `votes` ») ;

— Le pourcentage des votes obtenus par le candidat (« `percent` ») ;

— Le nom du candidat (« `candidate` ») ;

— Le parti politique du candidat (« `party` »).

Le deuxième fichier, qui est nommé « `canada.json` » et qui se trouve dans le dossier « `data` », contient l'ensemble des tracés nécessaires pour afficher les circonscriptions fédérales

sur une carte. Ces données sont enregistrées au format TopoJSON et chacun des polygones contenus dans le fichier possède, entre autres, les propriétés suivantes :

— Le numéro de la circonscription (« `NUMCF` ») ;

— Le nom de la circonscription en français (« `NOMSFR` ») ;

— Le nom de la circonscription en anglais (« `NOMSEN` ») ;

— Le code de la province abritant la circonscription (« `CODEPROV` »).

3 Travail à réaliser

Pour ce travail pratique, vous devrez compléter le code JavaScript nécessaire pour l'affichage d'une carte choroplèthe présentant les résultats obtenus pour les 338 circonscriptions fédérales lors des élections de 2015. Également, vous aurez à compléter la logique permettant de rechercher une circonscription précise sur la carte en effectuant un zoom automatique sur la zone concernée. Finalement, vous devrez afficher les résultats détaillés pour une circonscription sélectionnée dans un panneau d'informations se superposant à la carte. Le résultat final sera assez similaire à cette carte qui a été publiée sur le site web de Radio-Canada lors des dernières élections fédérales.

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 de la carte choroplèthe) avant de réaliser les deux dernières parties, 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 traitement sur les données provenant du fichier « `data.csv` ». En effet, vous devrez convertir les nombres provenant de ce fichier en type *number* en utilisant la fonction `parseInt`. Par la suite, vous aurez à réorganiser les données afin de faciliter leur manipulation. Par ailleurs, vous aurez à définir l'échelle de couleurs utilisée pour lier les noms des partis politiques à certaines couleurs. Pour réaliser ces éléments, vous devrez compléter le fichier « **`1-preproc.js`** » se trouvant dans le dossier « `assets/scripts` ». Plus précisément, vous aurez à compléter les fonctionnalités suivantes :

— Préciser le domaine et la plage utilisés par l'échelle de couleurs (fonction « `colorScale` ») ;

— Convertir les nombres provenant du fichier CSV en type *number* (fonction « `convertNumbers` ») ;

— Réorganiser les données provenant du fichier CSV afin qu'elles soient manipulables (fonction « `createSources` »).

3.2 Création de la carte choroplèthe

Pour cette deuxième partie, vous aurez à créer la carte choroplèthe affichant les différentes circonscriptions électorales. Avant de dessiner les tracés des circonscriptions, vous aurez à initialiser les configurations de la carte (fond de carte, niveau de zoom, position initiale) et vous devrez ajouter la couche SVG qui sera utilisée pour tracer les polygones au-dessus de la carte Leaflet. Pour vous aider, vous pouvez consulter cette exemple.

Une fois que vous aurez réussi à initialiser la carte Leaflet, vous devrez tracer les polygones correspondant aux 338 circonscriptions fédérales sur la couche SVG qui a été créée. La couleur d'une circonscription doit correspondre à celle du parti du candidat ayant été élu dans cette même circonscription. Également, l'opacité de cette couleur (propriété « `fill-opacity` ») doit être de 80 %. De plus, chacune des circonscriptions doit posséder une bordure de couleur grise (■ `#333333`). Par ailleurs, lorsqu'une circonscription est cliquée par la souris, celle-ci doit devenir sélectionnée (utiliser la classe « `selected` ») et le panneau d'informations montrant les résultats détaillées pour cette circonscription doit faire son apparition (utiliser la fonction « `showPanel` »). Il est à noter qu'il est possible de sélectionner uniquement une circonscription à la fois.

Attention

Assurez-vous que lorsque vous appliquez la classe « `selected` » à une circonscription, celle-ci se met à clignoter automatiquement. Si ce n'est pas le cas, assurez-vous de définir le style de base d'une circonscription dans le fichier CSS.

Le dernier élément à réaliser est d'effectuer la mise à jour des tracés au-dessus de la carte lorsque le zoom ou la position de cette dernière sont modifiés. En effet, vous devrez repositionner les éléments SVG aux bons endroits afin de prendre en compte les translations effectuées sur la carte. De plus, vous aurez à redessiner les circonscriptions lorsque le niveau de zoom de la carte sera modifié. Par souci de clarté, la Figure 1 illustre à quoi devrait ressembler la carte obtenue une fois cette partie complétée. Également, la Figure 2 montre l'ajustement automatique de la taille des circonscriptions en fonction du niveau de zoom.

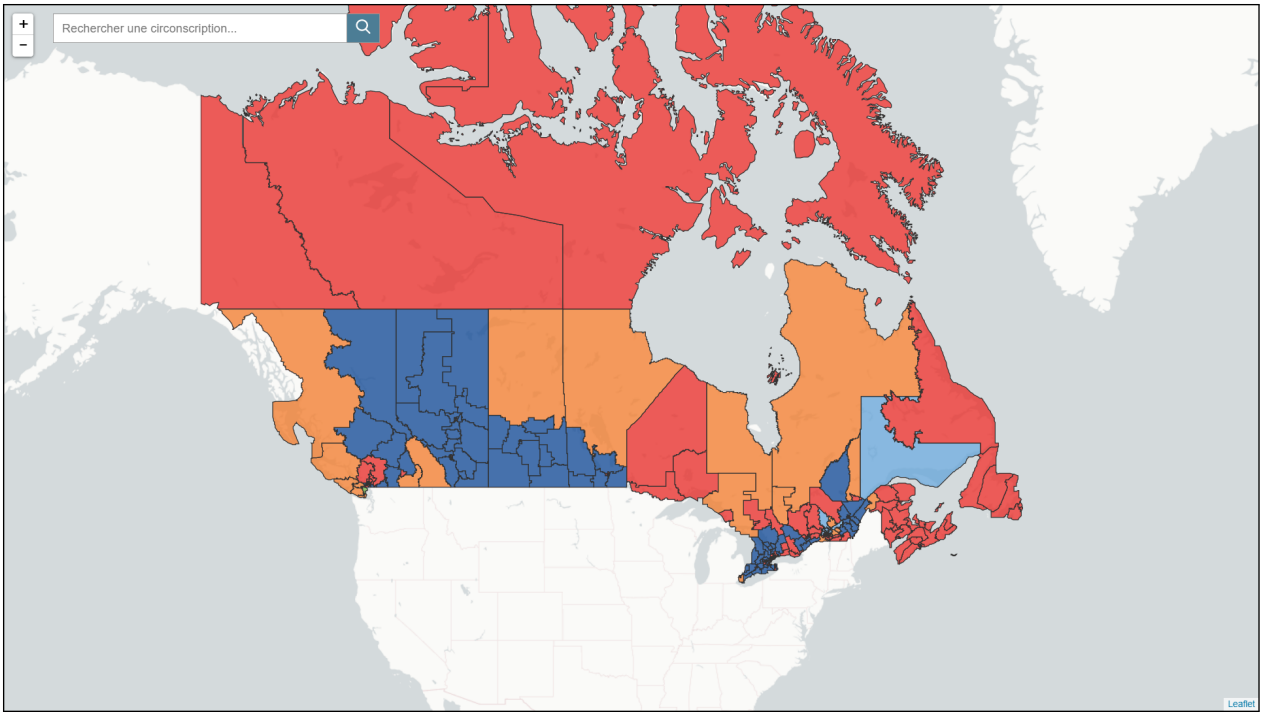


FIGURE 1 – Carte choroplèthe illustrant les résultats des circonscriptions fédérales lors des élections de 2015

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

— Initialiser le fond et les paramètres de la carte Leaflet (fonction « `initTileLayer` ») ;

— Initialiser la couche SVG responsable d'afficher les polygones au-dessus de la carte (fonction « `initSvgLayer` ») ;

— Tracer les circonscriptions sur le panneau SVG (« `createDistricts` ») ;

— Mettre à jour la position des éléments SVG et les tracés en fonction de la position et du niveau de zoom de la carte (« `updateMap` »).

3.3 Recherche d'un élément sur la carte

Pour cette troisième partie, vous aurez à compléter la recherche d'une circonscription quelconque en zoomant automatiquement sur la région correspondant à la circonscription recherchée. Comme vous l'avez sans doute remarqué, il peut être difficile de trouver une circonscription en raison de leurs nombres et de leurs tailles variées. Pour faciliter cette opération, une barre de recherche avec autocomplétion vous a été fournie afin que vous soyez en mesure de trouver une circonscription particulière.

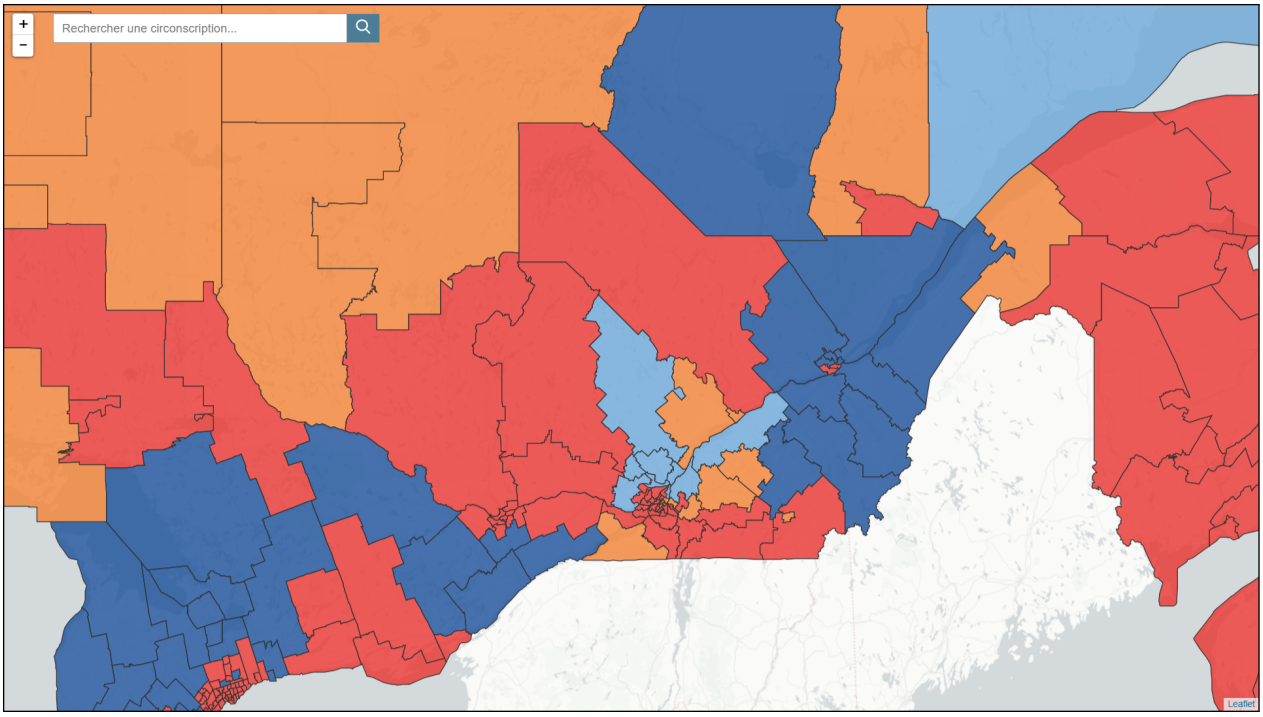


FIGURE 2 – Carte choroplèthe zoomée sur les circonscriptions du sud du Québec

Lorsqu'une circonscription valide est saisie dans la barre de recherche et qu'une recherche est lancée, un zoom automatique et animé doit se faire vers la circonscription recherchée. Pour y arriver, vous aurez à utiliser la fonction `fitBounds` de Leaflet tout en respectant les différentes contraintes demandées dans le code. Également, lorsqu'une circonscription est recherchée, celle-ci doit devenir sélectionnée (utiliser la classe « `selected` ») et le panneau d'informations montrant les résultats de cette circonscription doit être affiché (utiliser la fonction « `showPanel` »). Par souci de clarté, la Figure 3 illustre à quoi doit ressembler le résultat d'une recherche. Dans ce cas, la circonscription recherchée était « Abitibi–Baie-James–Nunavik–Eeyou ».

Pour réaliser cette partie, vous aurez à compléter la fonction « `search` » qui se trouve dans le fichier « **`3-search.js`** ».

3.4 Affichage du panneau d'informations

Pour ce qui est de la dernière partie à réaliser, vous aurez à afficher les résultats obtenus pour la circonscription sélectionnée dans le panneau d'informations exposé. Pour ce faire, vous devrez mettre à jour les informations textuelles suivantes dans le panneau :

— Le nom et le numéro de la circonscription (élément « `#district-name` ») ;

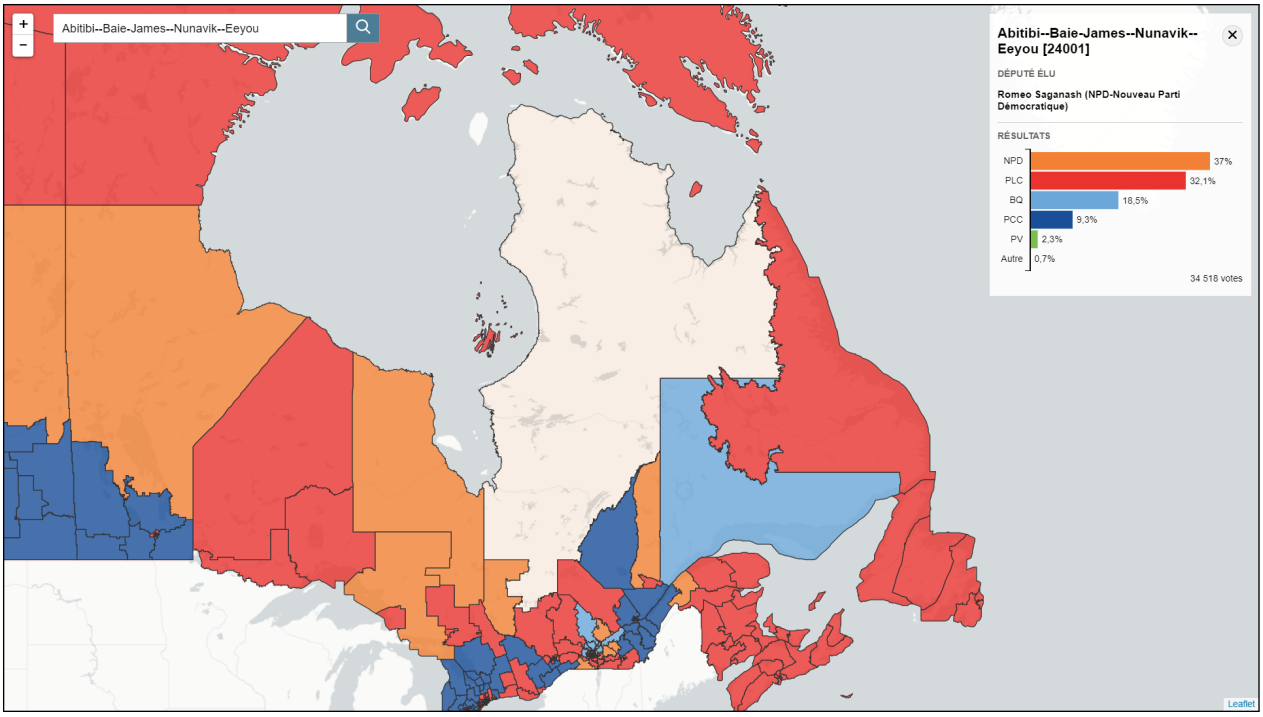


FIGURE 3 – Résultat de la recherche de la circonscription « Abitibi–Baie-James–Nunavik–Eeyou »

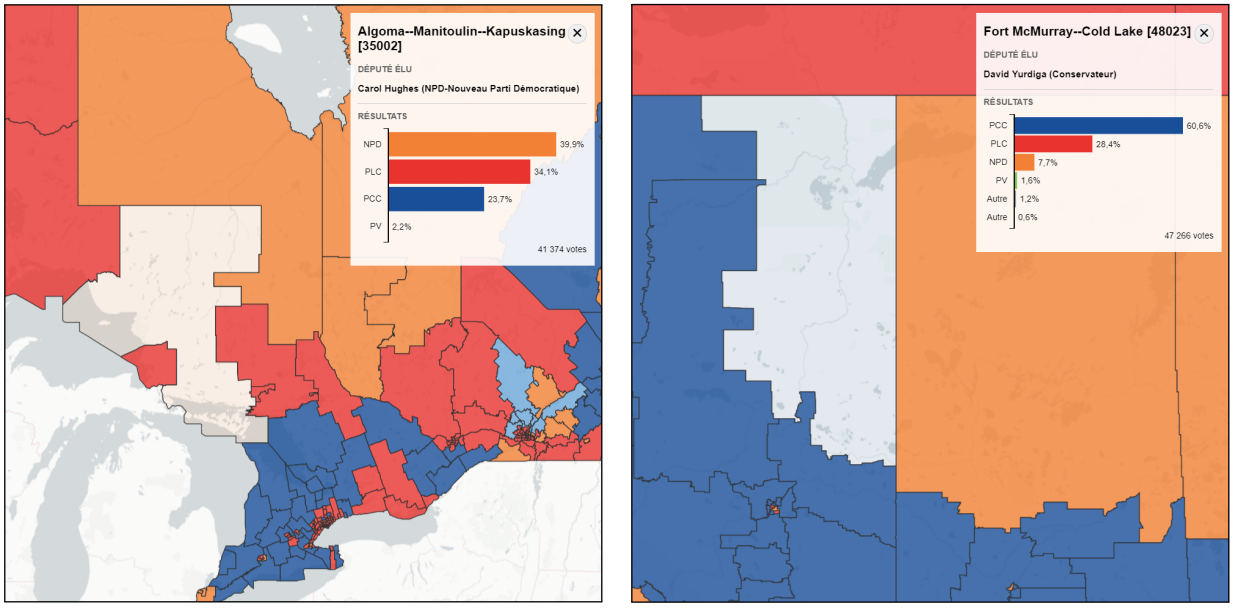
— Le nom du député élu ainsi que son parti politique (élément « `#elected-candidate` ») ;

— Le nombre total de votes pour tous les candidats (élément « `#votes-count` »).

En plus de mettre à jour les informations textuelles, vous aurez à créer et à modifier un diagramme à bandes horizontales illustrant les résultats obtenus pour les candidats s'étant présentés dans cette circonscription. Ce diagramme doit présenter les candidats en ordre décroissant de votes, c'est-à-dire présenter le candidat ayant obtenu le plus de votes en premier et ainsi de suite. La couleur des bandes du graphique doit correspondre au parti politique du candidat. Si le parti du candidat n'est pas répertorié dans l'échelle de couleurs (p. ex. Parti de l'Héritage Chrétien, Parti Rhinocéros, etc.), vous devrez colorier la bande en gris. Les pourcentages obtenus par les candidats doivent être indiqués à droite de chacune des bandes. Finalement, les noms abrégés des partis des candidats devront être indiqués à gauche des différentes barres. Pour obtenir le nom abrégé des partis, vous devez utiliser la liste « `parties` » qui vous est fournie. Dans le cas où le parti n'est pas répertorié dans la liste, remplacez le nom de ce dernier par « Autre ».

Lorsque le panneau d'informations est fermé via le bouton « ✕ », la circonscription qui était sélectionnée doit réinitialiser son affichage à celle par défaut (enlever la classe « `selected` »).

L'affichage des informations du panneau devrait être similaire à la Figure 4. Il est à noter que le rendu du diagramme à bandes doit être dynamique, c'est-à-dire que le nombre de bandes à créer doit correspondre au nombre de candidats s'étant présentés dans une circonscription. D'ailleurs, les figures 4a et 4b illustrent les derniers propos.



(a) Algoma–Manitoulin–Kapuskasing (b) Fort McMurray–Cold Lake

FIGURE 4 – Panneau d'informations illustrant les résultats pour différentes circonscriptions

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

— Mettre à jour les domaines du diagramme à bandes (fonction « `updateDomains` ») ;

— Mettre à jour les informations textuelles (fonction « `updatePanelInfo` ») ;

— Mettre à jour le diagramme à bandes (fonction « `updatePanelBarChart` ») ;

— Réinitialiser l'affichage à celle par défaut (fonction « `reset` »).

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 « `TP5_matricule1_matricule2_matricule3.zip` » ;
2. Le travail doit être remis avant **23h59**, le **19 mars 2020** 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
Création de la carte choroplèthe	8
Recherche d'un élément sur la carte	2
Affichage du panneau d'informations	6
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.