POLYTECHNIQUE
MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE

INF8808 – Visualisation de données

Lab 5

Authors:

Adrien Dessinges et Antoine Béland

Teacher's assistant:

Olivia Gélinas

Winter 2020

Department of computer and software engineering

1 Objectives

The goal of this practical work is to make a choropleth map using the libraries D3.js and Leaflet (version 0.7.7). You will use open data from a CSV file and from a TopoJSON file. The user will be able to obtain information about areas of the map as well as to search for a particular area. Before starting this work, it is recommended to have read chapter 12 of Scott Murray's book [1].

2 Introduction

A choropleth map is a map where regions are colored, shaded or filled with a certain pattern to show a certain numerical value or category. This type of map makes it easy to compare values between different geographic regions and makes it easier to see trends and differences based on location.

In this lab, you will have to make a choropleth map from the results of the 2015 federal elections for the 338 districts across the country. The detailed results for each electoral district come from Elections Canada, while the traces from different districts come from the Government of Canada's open data portal. Since this data comes from two different sources, you will have to handle the two files using different formats.

The first file, i.e. "`data.csv`" which is in the folder "`data`", contains all of the results for all districts in the country. Each of the lines in this file corresponds to the results obtained by a candidate who ran for a specific district. This includes the following information :

- The name of the district ("`name`") ;
- The ID of the district ("`id`") ;
- The number of votes obtained by each candidate ("`votes`") ;
- The percentage of votes obtained by each candidate ("`percent`") ;
- The name of the candidates ("`candidate`") ;
- The party of the candidate ("`party`").

The second file, which is named "`canada.json`" and which is located in the "`data`" folder, contains all the plots necessary to display the federal districts on a map. This data is saved in TopoJSON format and each of the polygons contained in the file has, among others, the following properties :

- The number of the district ("`NUMCF`") ;

— The name of the district in French ("NOMSFR") ;

— The name of the district in English ("NOMSEN") ;

— The code for the districts province CODEPROV").

3 What to do

For this lab, you will have to complete the JavaScript code necessary to display the choropleth map presenting the results obtained for the 338 federal districts during the elections of 2015. Also, you will have to complete the logic allowing to search for a district on the map by automatically zooming in on the area concerned. Finally, you will have to display the detailed results for a selected district in an information panel superimposed on the map. The final result will be quite similar to this map which was published on the Radio-Canada's website for the 2015 federal election.

The following subsections present the different parts that will need to be completed for this work. It should be noted that it is necessary to complete the first two parts of this work (data preprocessing and creation of the choropleth map) before carrying out the last two parts, which are independent. Make sure to complete the different "TODO" found in the files located in the "assets/scripts" folder.

3.1 Data preprocessing

For this first part, you will have to process the data coming from the file "data.csv". You will have to convert the numbers coming from this file into type *number* using the function parseInt. After, you will have to reorganize the data in order to facilitate its manipulation. In addition, you will have to define the color scale used to link the names of political parties to certain colors. To complete these tasks, you will have to complete the file "**1-preproc.js**" located in the folder "assets/scripts". More specifically, you will have to complete the following features :

— Specify the domain and range used by the color scale (function "colorScale") ;

— Convert numbers from the CSV file to type *number* (function "convertNumbers") ;

— Reorganize the data from the CSV file to be able to manipulate it (function "createSources").

3.2 Choropleth map

For this second part, you will have to create the choropleth map showing the different electoral districts. Before drawing the boundaries plots, you will have to initialize the map configurations (basemap, zoom level, initial position) and you will have to add the SVG layer which will be used to draw the polygons above the Leaflet map. To help, you can consult this example.

Once you have successfully initialized the Leaflet map, you will have to draw the polygons corresponding to the 338 federal districts on the SVG layer that was created. The color of a district must match that of the party of the candidate who was elected in that same district. Also, the opacity of this color (property "`fill-opacity`") must be 80 %. In addition, each district must have a gray border (■ `#333333`). When a district is clicked with the mouse, it must become selected (use the class "`selected`") and the information panel showing the detailed results for this district must appear (use the function "`showPanel`"). Note that it is possible to select only one district at a time.

> ⚠ **Be careful !**
>
> Make sure that when you apply the class "`selected`" to an electoral district, the latter starts to flash automatically. If not, be sure to define the basic style of a district in the CSS file.

The task to complete is to update the traces above the map when the zoom or the position are modified. Indeed, you will have to reposition the SVG elements in the right place to take into account the translations made on the map. In addition, you will have to redraw the divisions when the zoom level of the map is changed. Figure 1 illustrates what the resulting map should look like once this part is completed. Also, Figure 2 shows the automatic adjustment of the size of the districts according to the zoom level.

To complete this part, you will need to modify the file "**`2-map.js`**". You will have to implement the following features :

- Initialize the background and parameters of the Leaflet map (function "`initTileLayer`") ;
- Initialize the SVG layer responsible for displaying the polygons above the map (function "`initSvgLayer`") ;
- Trace the districts on the SVG panel ("`createDistricts`") ;
- Update the position of SVG elements and traces according to the position and zoom level of the map ("`updateMap`").

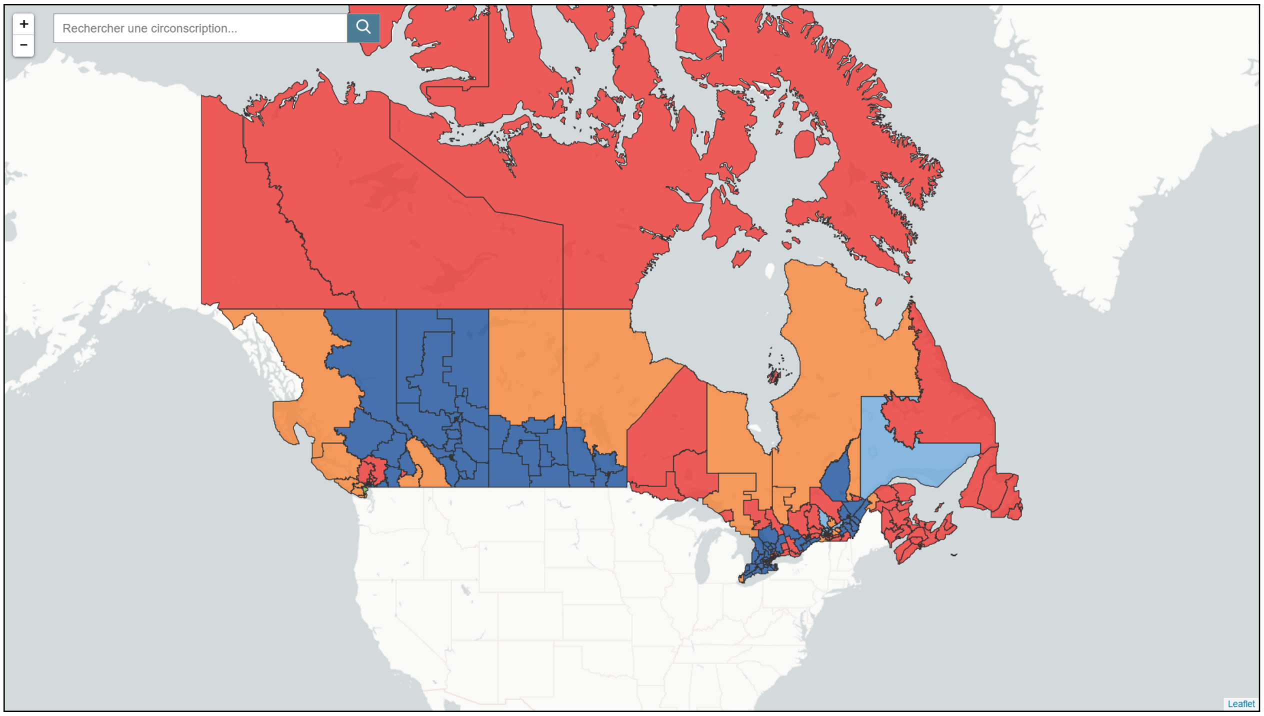


FIGURE 1 – Choropleth map showing the results of the federal districts during the 2015 elections

3.3 Search

For this third part, you will have to complete the search for any district by automatically zooming in on the region corresponding to the district sought. As you may have noticed, it can be difficult to find a riding because of their varied numbers and sizes. To facilitate this operation, a search bar with autocompletion has been provided to you so that you are able to find a particular district.

When a valid district is entered in the search bar and a search is launched, an automatic and animated zoom must be made to the district sought. To get there, you will have to use the `fitBounds` function of Leaflet while respecting the various constraints requested in the code. Also, when a district is sought, it must become selected (use the class "`selected`") and the information panel showing the results of this district must be displayed (use the function "`showPanel`"). For clarity, Figure 3 illustrates what the search result should look like. In this case, the electoral district sought was "Abitibi–Baie-James–Nunavik–Eeyou".

To complete this part, you will have to complete the function "`search`" which is in the file "**`3-search.js`**".

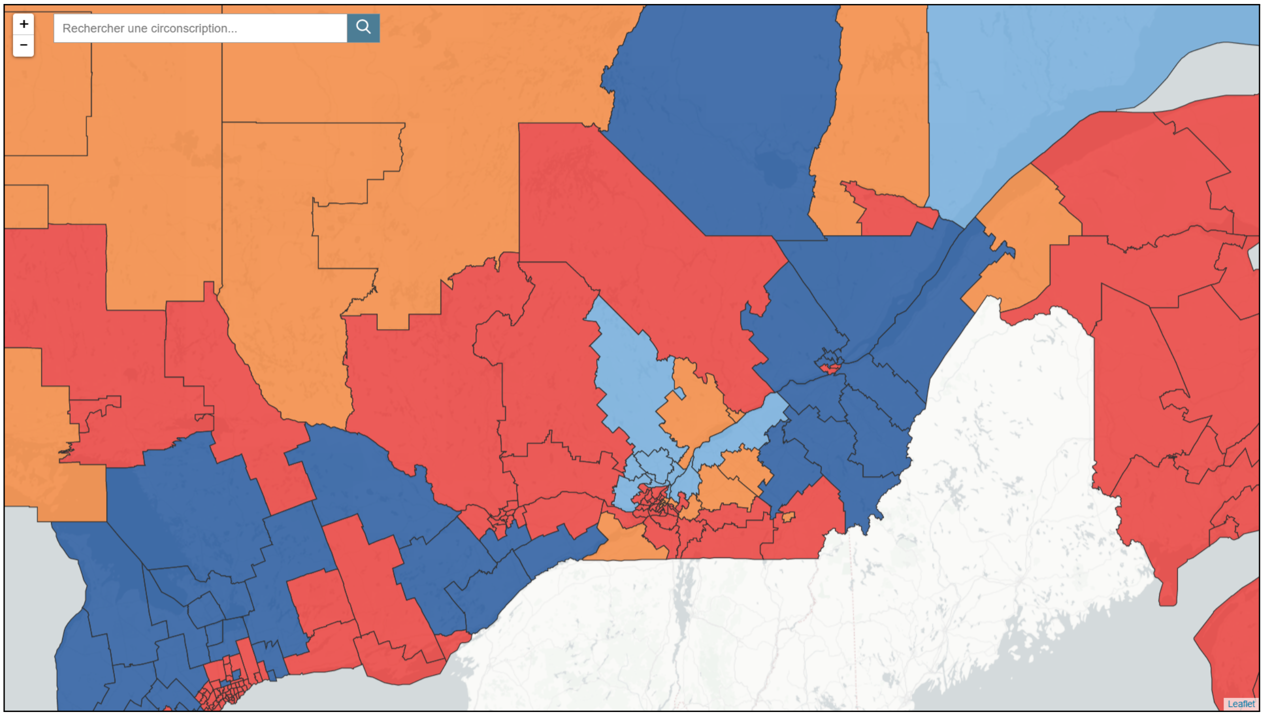


FIGURE 2 – Choropleth map zoomed on the ridings of southern Quebec

3.4 Information panel

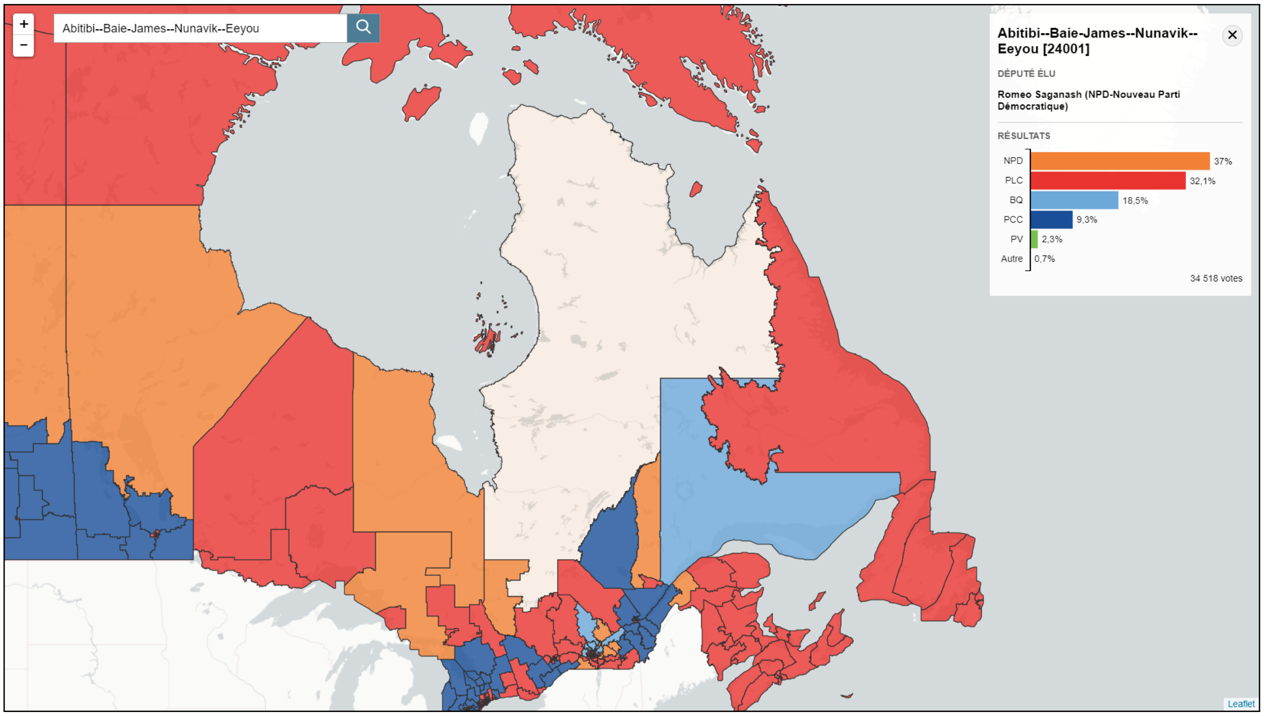
For this last part, you will have to display the results obtained for the selected district in the information panel. To do this, you will need to update the following text information in the panel :

— The name and number of the district (`"#district-name"`);

— The name of the elected deputy and his or her party (`"#elected-candidate"`);

— The total number of votes for each candidate (`"#votes-count"`).

In addition to updating the textual information, you will have to create and modify a horizontal bar graph illustrating the results obtained for the candidates having presented themselves in this district. This diagram should present the candidates in descending order of votes, i.e. present the candidate with the most votes first and so on. The color of the stripes on the graph must correspond to the political party of the candidate. If the candidate's party is not listed in the color scale (e.g. Christian Heritage Party, Rhinoceros Party, etc.), you will need to color the stripe in gray. The percentages obtained by the candidates must be indicated to the right of each of the bands. Finally, the abbreviated names of the parties of the candidates must be indicated to the left of the various bars. To obtain the abbreviated


Abitibi--Baie-James--Nunavik--Eeyou

Abitibi--Baie-James--Nunavik--Eeyou [24001]

DÉPUTÉ ÉLU

Romeo Saganash (NPD-Nouveau Parti Démocratique)

RÉSULTATS

FIGURE 3 – Reuslt of the search for the district "Abitibi–Baie-James–Nunavik–Eeyou"

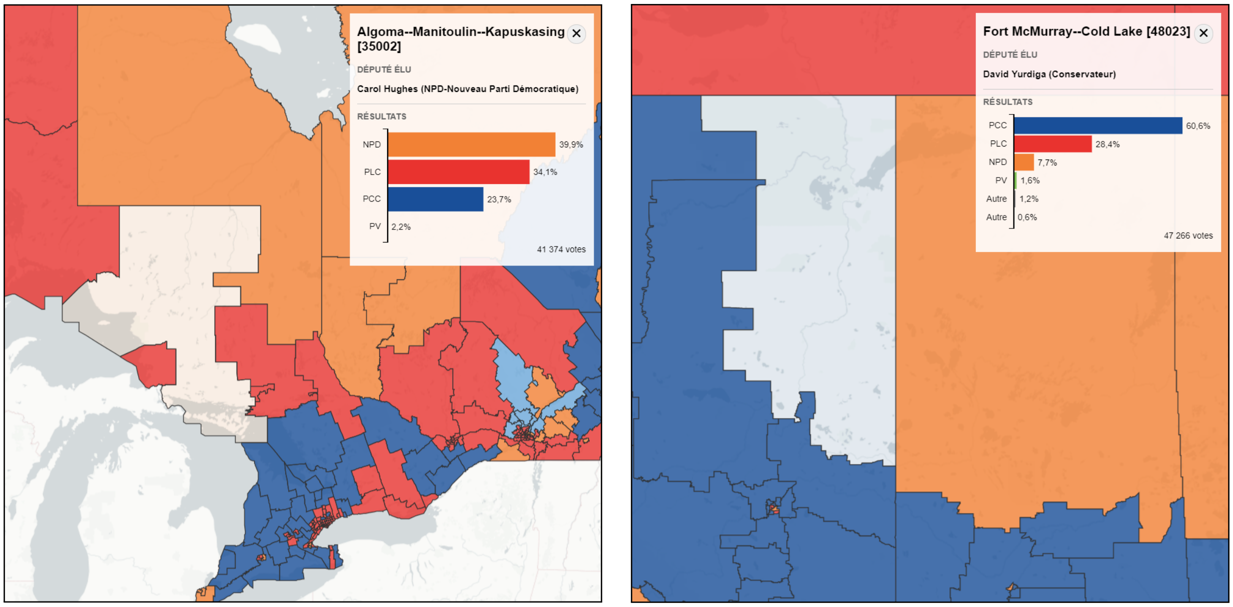
name of the parties, you must use the list "parties" which is provided to you. If the party is not listed, replace the party's name with "Other".

When the information panel is closed via the "✕" button, the district which was selected must reset its display to the default (remove the class "selected").

The display of the panel information should be similar to Figure 4. Note that the rendering of the bar graph must be dynamic, that is to say that the number of bands to be created must correspond to the number of candidates having presented themselves in a district. The figures 4a and 4b illustrate these last remarks.

To complete this part, you will need to complete the file "**4-panel.js**". More precisely, you will have to :

- Update the domains of the bar charts (function "updateDomains") ;
- Update the textual information (function "updatePanelInfo") ;
- Update the bar chart (function "updatePanelBarChart") ;
- Reinitialize the display to the default (function "reset").


Algoma--Manitoulin--Kapuskasing [35002]
DÉPUTÉ ÉLU
Carol Hughes (NPD-Nouveau Parti Démocratique)
RÉSULTATS
NPD
39,9%
PLC
34,1%
PCC
23,7%
PV
2,2%
41 374 votes
Fort McMurray--Cold Lake [48023]
DÉPUTÉ ÉLU
David Yurdiga (Conservateur)
RÉSULTATS
PCC
60,6%
PLC
28,4%
NPD
7,7%
PV
1,6%
Autre
1,2%
Autre
0,6%
47 266 votes

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

FIGURE 4 – Information panel showing results for different districts

4 Submission

Here are the submission guidelines for this lab :

1. You must submit the code for your project in a compressed folder in ZIP format named "`TP5_studentID1_studentID2_studentID3.zip`";
2. This lab must be submitted before **11 :59PM**, **March 19 2020** sur Moodle.

5 Évaluation

Globally, you will be evaluated based on the requirements for the lab listed below. The following guideline will be used :

Requirement	Points
Data preprocessing	3
Choropleth map	8
Search	2
Information panel	6
Code quality and clarity	1
Total	20

This lab is worth **7,5 %** of the course grade.

Références

[1] S. Murray, *Interactive Data Visualization for the Web : An Introduction to Designing with D3.* O'Reilly Media Inc., 2013.