POLYTECHNIQUE
MONTRÉAL

LE GÉNIE
EN PREMIÈRE CLASSE

INF8808 – Visualisation de données

Lab 2

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 lab is to create a (*line chart*) with a dynamic zoom using open data in CSV format.

Before starting, it is recommended to read chapters 5, 6 and 7 from Scott Murray's book [1] for some background information.

2 Introduction

A line chart is a type of graphic where data is represented by one or by many lines. It is extremely common and can be found in many fields where we would like to see the evolution of a parameter as a function of another. It is for example used in mathematics to illustrate the behaviour of a function of form $y = f(x)$.

In this lab, you will have to make a line chart using data about the quantity of bicycles passing each day on certain streets of Montreal in 2016. This data is collected from bicycle counters located on many of Montreal's bike paths. Figure 1 illustrates what these bicycle counters look like in practice. This data is available via the open data portal of Montréal.

Figure 1: Bicycle counter on rue Laurier

© François Démontagne, 2015

3 What to do

In this lab, you will have to complete Javascript code in order to display a line chart based on the data on the quantity of bicycles circulating daily on Montréal's streets in 2016. Additionally, you will have to complete a *brush* and a legend to accompany your graphic.

The following subsections present the different parts that you will have to complete for this lab. Note that you will have to complete the two first parts of this lab (data preprocessing and diagram creation) before starting the third and fourth parts. Make sure you complete the various `TODO` sections in the code, which can be found in the folder "`assets/scripts`".

3.1 Data preprocessing

For this first part, you will have to preprocess the data provided by the City of Montréal. The data contained in the CSV file is raw, so it is necessary to reorganize certain parts of it so they can be properly used by the D3 library. To do so, you need to complete the file "**`1-preproc.js`**", which is in the folder "`assets/scripts`". More precisely, you will have to complete these steps :

- Specify the domain of the color scale (function "`domainColor`");
- Specify the domain of the vertical and horizontal axes (functions "`domainX`" and "`domainY`");
- Convert the dates of the CSV file to JavaScript objects (function "`parseDate`")
- Sort data by street and then by date (function "`createSources`").

⚠ **Be careful!**

Make sure you convert the numbers from the CSV file to type *number* since they are initially of type *string*. To do this conversion, you can use the function `parseInt` provided by JavaScript. This conversion is important because otherwise D3 will not properly interpret the data.

3.2 Diagram

For this second part, you will have to implement the main part of the data visualization. This visualization is in fact divided into two line chart diagrams. One of them, called "focus" allows you to show the data in more detail. The other one, called "context", allows you to show a higher level overview of the data and select which data to show in the "focus" section using a *brush* (see next section). Figure 2 shows these two diagrams which compose the main visualization.

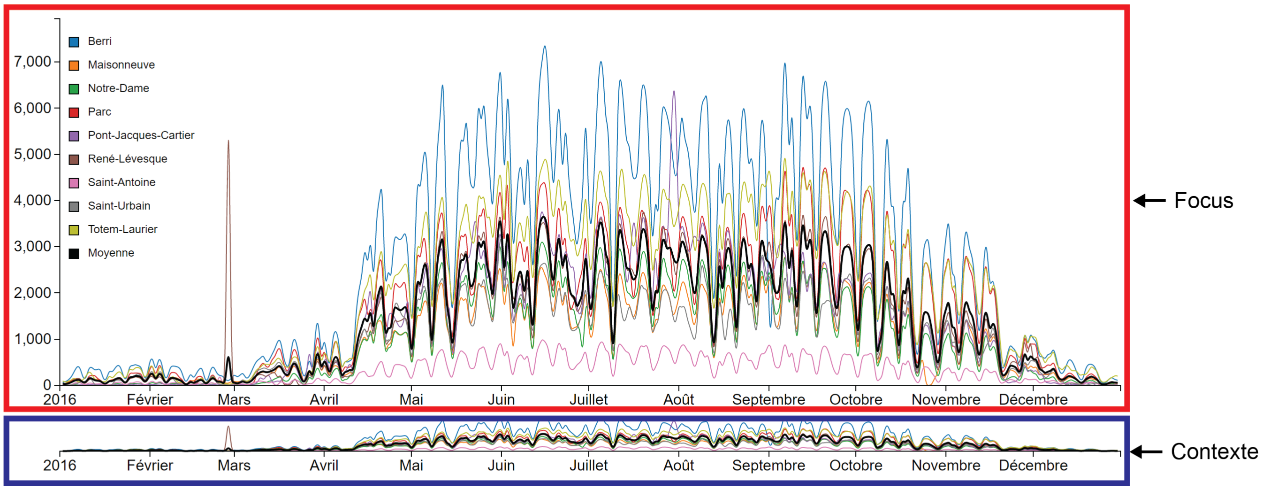

Figure 2: The "focus" (red) and "context" (blue) diagrams to implement

Each line of the line charts will be associated with a color that represents the data for a particular street. In addition, the line corresponding to the daily traffic average should be black and should be thicker than the others.

The axes are already implemented in the code. However, be sure to assign the correct data range to them. You will notice that the "focus" and "context" line charts use the same data and the same colors. The code to generate these two line charts will therefore be quite similar.

For this part, you wil have to complete the file "**`2-line-charts.js`**". You will have to implement these features:

- Create an SVG line for both line charts (function "`createLine`");
- Create the "focus" line chart (function "`createFocusLineChart`");
- Create the "context" line chart (function "`createContextLineChart`").

3.3 *Brush*

The third part of this lab consists in the implementation of a *brush*. This *brush* allows you to select a particular range of data to be displayed in the "focus" line chart. The "context" line chart allows the selection of the data range to be displayed in the "focus" line chart. The Figure 3 illustrates this mechanism. As we can see, the "context" diagram allows the selection of data associated with the month of June, then the "focus" diagram updates its domain in x to display only the data for the month of June .

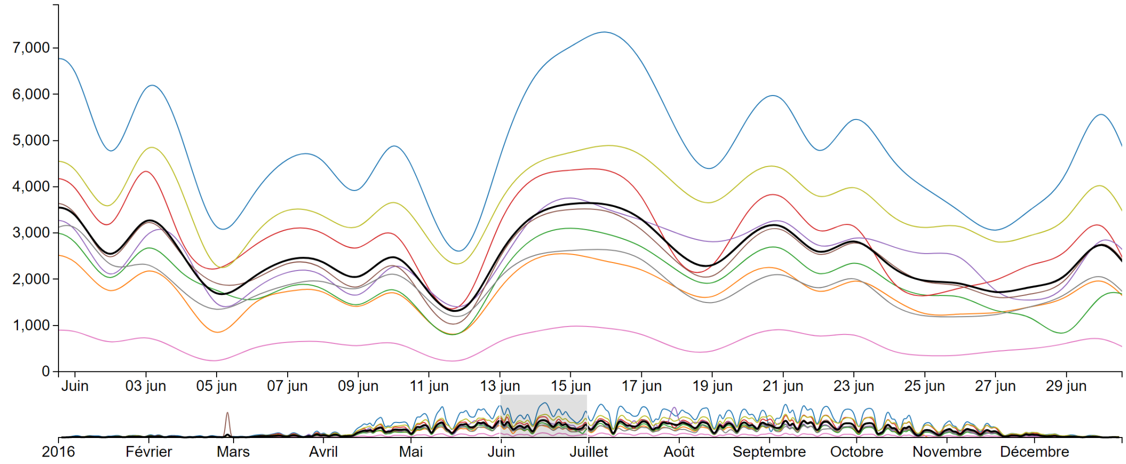

Figure 3: Selection of the month of June with the *brush*

The selection in the "context" line chart has already been implemented for you. You will only have to update the "focus" line chart based on the selection in the "context" line chart. You will have to update the horizontal and vertical scale of the "focus" line chart and to redraw its lines based on the current selection. You will have to complete the file "**`3-brush.js`**" for this part.

3.4 Legend

In this last part of the lab, you will add a legend to the visualization to make it easier to read. This legend will associate each color with the name of the street it represents. Note that this legend must be drawn in the foreground, meaning on top of the "focus" graph. Also, make sure that the colors in the legend match the color of the lines in the diagrams. Figure 4 also illustrates what the legend should look like.

As the amount of streets to display is quite large, it can be difficult to read the data for a given street. To solve this problem, we should be able to display or hide the data of certain streets as desired. You will have to complete a method which displays or hides the data for any street when the square for that street is clicked in the legend. When the data for a street is hidden, the square in the legend for this street must become white with a black outline. See Figure 5.

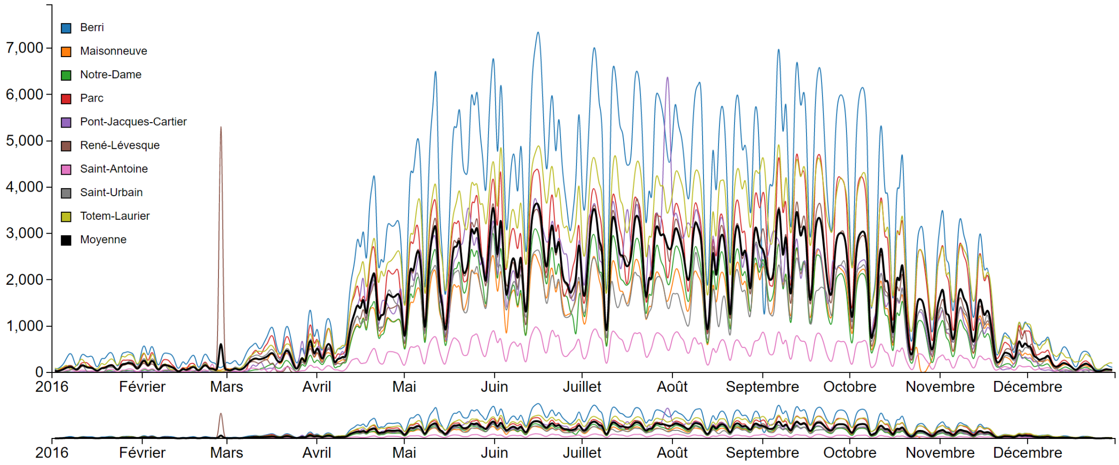

Figure 4: The visualisation's legend

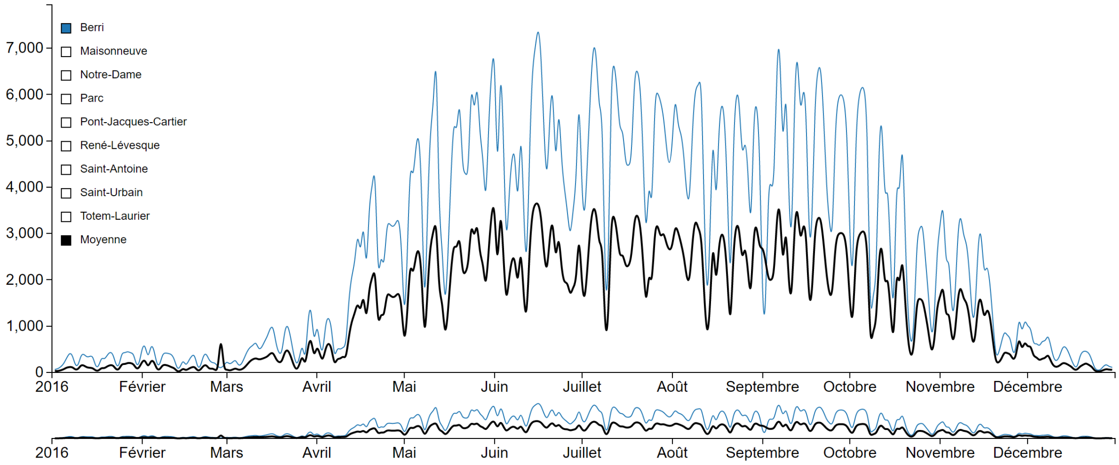

Figure 5: Visualization showing only data for rue Berri and the mean

For this part, you will have to complete "**`4-legend.js`**". You will have to:

- Display the legend for the "focus" line chart (function "`legend`");
- Make the display for the line charts' lines dynamic (function "`displayLine`").

Be careful!

Make sure to show or hide the street data in both the "focus" and "context" line charts. You must update the two line charts.

4 Submission

Here are the instructions for submitting this lab:

1. You must place your project code in a compressed ZIP file named "`TP2_studentID1_studentID2_studentID3.zip`".
2. The lab must be submitted before **11:59PM February 6** on Moodle.

5 Evaluation

Overall, you will be assessed on the respect of the following requirements. The correction is divided is as follows:

Requirements	Points
Data preprocessing	6
Diagram	6
Brush	3
Legend	4
Code quality and clarity	1
Total	20

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

References

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