TNM048—Information Visualization

Interactive Visualization

January 17, 2017

Jimmy Johansson jimmy.johansson@liu.se
Carlo Navarra carlo.navarra@liu.se

1 Introduction

In this lab you will implement a set of coordinated linked views that display data from the Better Life Index using commonly used Information Visualization techniques: a Scatter Plot, a Parallel Coordinates plot, and a Choropleth Map.
To pass this exercise, functional code as well as your explanations of how it works are required. The laboratory exercise should be performed individually or in a group of maximum two students. After completing all tasks, present your results to the laboratory assistant.

2 The Setup

The lab will be done using Html5 and the D3.js JavaScript library. Download the file lab1.zip from the course homepage. The file contains the different files that you will use and integrate with your own code.

3 The Data

The data set used in this exercise is the Better Life Index with high inequality (www.oecd.org/statistics/OECD-Better-Life-Index-2013-definitions.pdf). The data set is multivariate and stored in a comma separated value file (lab1/data/OECD-better-life-index-hi.csv).

Task 1:
Once downloaded, extract the compressed folder "lab1" and familiarize yourself with the file structure and the data.

4 Scatter Plot

A Scatter Plot view represents the relationships between two dimensions of a multivariate data set by plotting dots on a two dimensional space, additional data dimensions can be mapped on the size and/or color of the dots.
The Scatter Plot code is in the file sp.js in lab1/js/ folder. The scatter plot object is instantiated in the file main.js in lab1/js/.

Task 2:
Choose the data attributes to visualize and define the scale of the scatter plot axes in the parser `d3.csv()` using the operator `domain()`.

Task 3:
In the `draw()` method add the scatter dots by defining the `x` and `y` coordinates and the radius data values using the method `.attr()` in combination with the `circle` attributes.

Now you should be able to see the data displayed in the Scatter Plot.

Task 4:
Add the data value name to the axes in order to give context to the visualization.

5 Parallel Coordinates

A Parallel Coordinates plot shows all dimensions of the multivariate data set at once by mapping the data values onto several vertical axes each representing one dimension.
Modify the file pc.js in the folder lab1/js to implement a parallel coordinates plot.

Task 5:
In the data importer modify the operator `x.domain(...)` to extract the list of dimensions from the data and create a scale for each them.

Task 6:
In the method `draw()` add the data and append the polyline primitive `path` to the background lines and foreground lines.

Task 7:
Use the operator `.each` in combination with the operators `.select` and `.call` to assign the scale for each axis.

If all the previous steps have been completed correctly you should be able to see the lines and the axes of the Parallel Coordinates.

6 Choropleth Map

In a choropleth map each region or bordered area is coloured according to a corresponding value in the data set. Edit the file map.js in lab1/js/ folder.

Task 8:
Create a color scheme to shade each country with an unique color using the operator `.style()`. Use this color scheme for the other two components in order to have the same color consistency. (TIP: create an array of object `{country:... ;color:...}`).

7 Brushing and Linking

Brushing means selecting a subset of the data items, usually, in order to highlight them. In parallel coordinates this implies selecting a range of values on one of the axes (dimensions). A brushing technique that toggles the foreground lines of the parallel coordinates is already implemented in the method `brush` in the js/pc.js file, spend few minutes to analyse and understand it.

Brushing is most interesting in connection with linking. The Linking technique connects the selecting behaviour between the different views of an application so that selections made in one view are reflected

in all others. For instance by brushing lines in a parallel coordinates view, the brush effect (highlighting, etc.) should be applied on those points in the other views that represent the same data items.

Task 9:
Implement the method `selectDot()` in sp.js and `selectLine()` in pc.js so that when a dot is selected in the scatter plot a line in the parallel coordinates plot is also selected and vice-versa. (TIP: the methods are declared as public so they can be called from other classes)

Task 10:
Modify the methods `brush()` in pc.js and `selectDot()` in sp.js so that when brushing occurs in the parallel coordinates plot the corresponding dots are selected in the scatter plot.

8 Tool-tip

A tool-tip is a common graphical user interface element used to display information about an item that is hovered with the mouse pointer.

Task 11:
To increase the understanding of the data displayed, implement a tool-tip system displaying the name of the country or the data value for each view. This can be done using the `.on("mousemove",...)` and `.on("mouseout",...)` methods in combination with the .transition operator applied on a div.