

Lab 1: Line and Bar Graphs

In this lab, you will be using Processing to draw a line graph and a bar chart, with a toggle button that can switch between the two. This is your first lab that displays a visualization based on input data, and there are a few concepts that you need to explore: (1) reading and parsing data; (2) mouse hovering for highlighting visual elements;

Basic Requirements for This Assignment:

1. We are giving you a simple comma-delimited file (CSV) called “data.csv”. This file will have the following properties:
 - a. It has two columns, the first column displays categorical (ordinal) data, and the second column displays quantitative data.
 - b. The first row has labels for each column.
 - c. There are around 10 rows of data.
2. In Processing, do the following:
 - a. Parse the CSV file and read in the data.
 - b. Create a button at the upper-right-hand corner of the canvas. This button should toggle between two states (similar to the button that you created in lab 0 except that it shouldn't resize).
 - i. In state 1, the canvas should display the data in the CSV file as a line graph
 1. The line graph should display each data item as a circle.
 2. The circles should be connected by lines.
 - ii. In state 2, the canvas should display the data in the CSV file as a bar chart
 1. The bar chart should display each data item as a bar
 - iii. In both the line chart and bar graph, the axes need to be labeled using the labels L1 and L2. The X-axis should display the categorical dimension of the data, and the Y-axis should show the numeric values.
 - c. Implement mouse hovering. This means that when the user moves the mouse over a circle (in the line graph), or a bar (in the bar chart), the circle or bar should be highlighted. Furthermore, additional text information should be displayed (like a tool tip). The additional text information should be the original value of the data item, displayed as “(Apple, 24)” for showing both the x and y values of the data item.
 - i. For testing intersection, test the geometry of the mouse position with the rectangles or circles.
 - d. Do not assume a fixed range in either the X or Y axis. For instance, do not assume that the values in the Y-axis are always between 0 and 100. Determine the lower and upper bound based on what you read from the data and scale the Y-axis accordingly. Similarly, do not assume that the number of data items in the X-axis is always below a certain number. Your visualization should adapt to the data that it is given and display the data in the most appropriate manner.
 - e. The system needs to be resizable in that you should not assume a fixed dimension for your canvas. In other words, if the canvas size becomes larger, your visualization should become larger. Conversely, if the canvas becomes smaller, your visualization shouldn't be “cut off”.
 - i. To add resizing to your Processing code, add the line in the setup function:

```
frame.setResizable(true);
```

- ii. Note that you need to remove this line if you want to post this program online. Since this line of code is not standard Processing code, it will cause your program to not run in a browser via Processing.js.
- iii. Pay attention to the size of the spacing (e.g., the space between the bars). The spacing should not be a static number (e.g., hard coded to 5 pixels), but should be dynamically determined based on the canvas size and the number of data elements.
- f. Your code needs to make use of object-oriented concepts. Specifically, you need to implement at least one class for the line graph, and one class for the bar chart. If you think more classes are helpful, feel free to add more classes.

Additional notes:

1. When you demo your assignment to us, the TAs and I might ask you to use a different CSV file. These new datasets can vary in terms of the number of data items, as well as the range of possible values in the y-axis. Be sure to put some thought into what sizes your bars and circles should be when there are only 2 data items vs. when there are 200.
2. Also, as noted in the lecture, if you implement the barchart well, you can reuse it for your assignment 1, which calls for integrating treemaps with a barchart.