Programming Assignment 1

Due on Jan 29, 6pm.

We strongly recommend that you complete Task 1 and 2 in the first week. First, you will use D3 to create a column chart and then you will need to update the given code for rendering a scatter plot to use the circle size as an additional channel.

For Task 3, you will explore additional data sources and visualize them by using other chart types in D3.

This programming assignment assumes that you have fulfilled all course prerequisites and followed along with the tooling videos.

Template and Submission

We will use **git** repositories for all programming assignments and project milestones. You should already be well acquainted with the process and all necessary git commands from previous courses (see Reid Holmes' CPSC 310 git tutorial as a refresher).

We created a git repository in your github student account (https://github.students.cs.ubc.ca/) that contains a template for all three tasks that should help you get started. The datasets are included in the /data folder. You will need to modify the .html, .js, and .css files as described below.

Submit the programming assignment by updating the given repository (`git push`). You can commit/push changes as often as you want before the deadline. Do not rename files or create new ones. Work alone.

Task 1: Column Chart (50%)

1. Familiarize yourself with the given template. When you start a local web server and open `task1.html` , you should see a bar chart with the top 10 countries by population. See Technical Hints on the website for more details on git and instructions for starting a local web server.

2. Change the data source

    - Load the CSV file `water_withdrawal_data.csv` that contains the yearly freshwater withdrawal (million cubic metres) for 25 countries. You don't need the file `population_data.csv` anymore.
    - Sort countries according to its water withdrawal and select the top 10.
    - Change the bar chart (x and y dimensions) to use the new data.

3. Change the fill color of the bars to a shade of grey.
4. Change the chart orientation to show vertical columns instead of horizontal bars.
    - Countries should be displayed on the x-axis and the corresponding water withdrawal on the y-axis. Make sure that the x-axis is at the bottom of the chart and the columns are not rendered upside down.
5. Incorporate a second dataset
    - Now use normalized freshwater withdrawal per capita in your column chart instead of absolute water withdrawal.
    - You will need to combine the 2 datasets `water_withdrawal_data.csv` and `population_data_all.csv`. The field *country_code* can be used as a common identifier. You can choose to merge the datasets in an external tool or load both with D3.

      Hint: You can use Promise.all to load multiple CSV files in JavaScript

      ```
      Promise.all([
          d3.csv("file1.csv"),
          d3.csv("file2.csv")
      ]).then(files => {
          // files[0] contains file1.csv
          // files[1] contains file2.csv
      })
      ```
    - Use cubic metres instead of million cubic metres on the y-axis
6. Add a chart title in HTML (SVG axis titles are optional).

> **Data sources**
>
> https://data.oecd.org/water/water-withdrawals.htm https://data.worldbank.org/indicator/sp.pop.totl
>
> "Water withdrawals, or water abstractions, are defined as freshwater taken from ground or surface water sources, either permanently or temporarily, and conveyed to a place of use. If the water is returned to a surface water source, abstraction of the same water by the downstream user is counted again in compiling total abstractions: this may lead to double counting. The data include abstractions for public water supply, irrigation, industrial processes and cooling of electric power plants."

Task 2: Scatter Plot Variations (10%)

1. Familiarize yourself with the given template. When you open `task2.html`, you should see a scatter plot of car models presenting weight on the y-axis and horsepower on the x-axis.
2. Search the web for a new dataset with at least 3 quantitative dimensions.
3. Change the starter template to use your new dataset instead of the car models data.
4. Use the radius to encode a third dimension, in addition to the circle's x and y position (e.g., between 2 and 10px depending on the data value).

Task 3: Chart Types (40%)

1. Familiarize yourself with the given template. The HTML document `task3.html` contains a 2-column layout with empty SVG drawing areas that you will populate with two different D3 visualizations.
2. Pick two datasets from the quirky collection at DataIsPlural: either browse through the newsletter archive or see the full archive in spreadsheet format. The requirements are:
    - One that can be visualized as a scatter plot and a second one that can be visualized as a line or area chart.
    - The datasets can cover different topics and it is very likely that you will only use 2-3 columns of a larger dataset. You can choose one complex dataset if you can visualize different dimensions and provide two perspectives on the data by showing 2 charts.
3. Create charts
    - Scatterplot: You can copy the D3 code from task 2 and connect the scatter plot to the new dataset.
    - Line or area chart: Implement the chart yourself or find an appropriate starter template on https://vizhub.com/ or https://github.com/d3/d3/wiki/Gallery#basic-charts. Connect the chart to the chosen dataset.
    - The two charts need to be displayed side by side
4. Tweak the charts in some substantive way. For example, a combination of smaller changes (e.g., colors, tick format) or one major change (e.g., show circle marks on top of the line).
5. Add descriptive chart titles and axis labels
6. Reflect about your design choices in `task3.html` below the charts
    - State the source of your datasets
    - What is your rationale for the fit between your chosen data and the chart types?
    - Have you used a base code? If yes, please provide the URL.

- How did you tweak the charts?

Updated 5pm on Jan 20, 2020.