CSE46801 Information Visualization - Midterm

Seongouk Kim - zjqvp@unist.ac.kr

Time: 2024.04.17. (Wed), 17:30 - 18:45

This exam is **open-book exam**. You can search online, or utilize the materials on a coding session and assignments. But, **discussing with friends or utilizing AI for coding assistance (e.g. ChatGPT, CoPilot) is not allowed.**

0 Setting Node.JS (0pts)

After you unzip the file, you will see 2 directories: Prob1 and Prob2. Both problems have same structure - index.html, index.js, main.js and csv data. To run the server and start solving, follow these instructions.

1. Turn on the terminal and move to the Prob1 or Prob2. Then, run **npm install** command.
2. Run **npm start** command on the same directory. It will run the local server. If you see the message *Local server with port 3000 opened!*, you successfully set up the environment. Now, you can check your implementation at http://localhost:3000.

1 Fix the Errors in the Multiple-Line Chart implementation(15pts)

In this problem, you will be given an entire code implementation of a multiple-line chart using D3.js. It uses Life Expectancy by Country dataset which contains the following variables.

Name of the variable	Description
country_code	ISO 3166 Country code of the country where the datum is collected
country_name	The country where the datum is collected
year	The year when the datum is collected
value	Life expectancy of the country

The written code is supposed to get **Top 5 countries that have the largest difference between the minimum and maximum life expectancy**. Then, it draws 5 line charts that show the change in life expectancy over time in each country. However, this code contains several errors. Your task is to identify and correct these errors to ensure that the chart functions as intended. Here are the instructions:

- There are **5 errors** in index.html and main.js. It may be syntax errors, logical errors, or missing codes.
- Carefully review the provided code implementation.
- Identify and fix errors or issues you found. **Fixing each error is worth 3 points.**

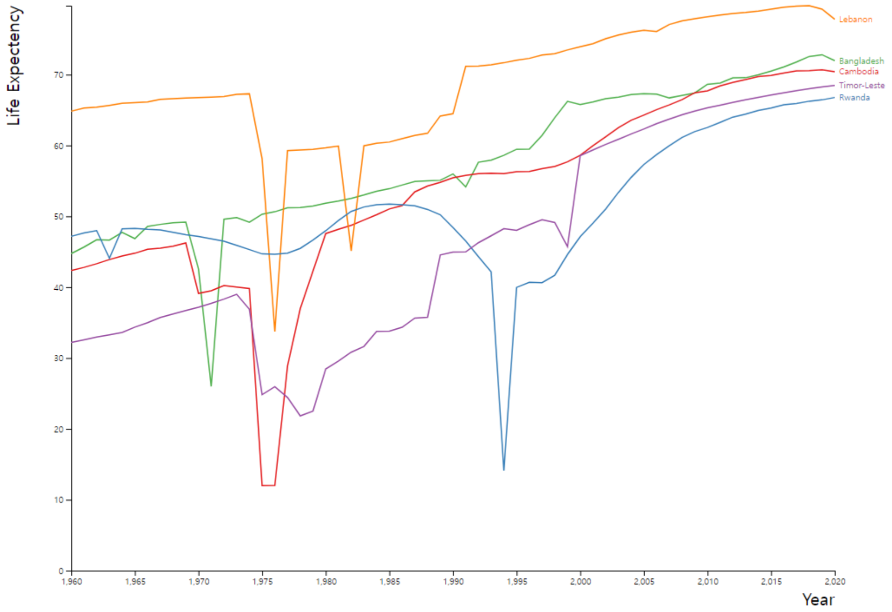

Figure 1: The completed multiple-line chart

Your result should look like Figure 1.

2 Drawing Bubble Chart (25pts)

In this problem, you will draw a bubble chart with the international COVID-19 dataset from Our World in Data. In this dataset, you will handle the variables below:

Name of the variable	Description
continent	The continent of the country where the datum is collected
location	The country where the datum is collected
date	The date when the datum is collected
population	The population of the country
life_expectancy	Life expectancy at birth in 2019
gdp_per_capita	Gross domestic product at purchasing power parity, most recent year available

You'll draw a bubble chart that shows the correlation between GDP Per Capita and Life Expectancy. Your bubble's color will show the continent of the country. And your bubble's size will show the population of the country.

Here are the instructions for implementing the bubble chart.

2.1 Data Processing

- Exclude data that have missing values in the columns related to the variables above. (2pts)
- Exclude all data except the latest data for each country. (2pts)
- Sort the data by the life expectancy. (2pts)

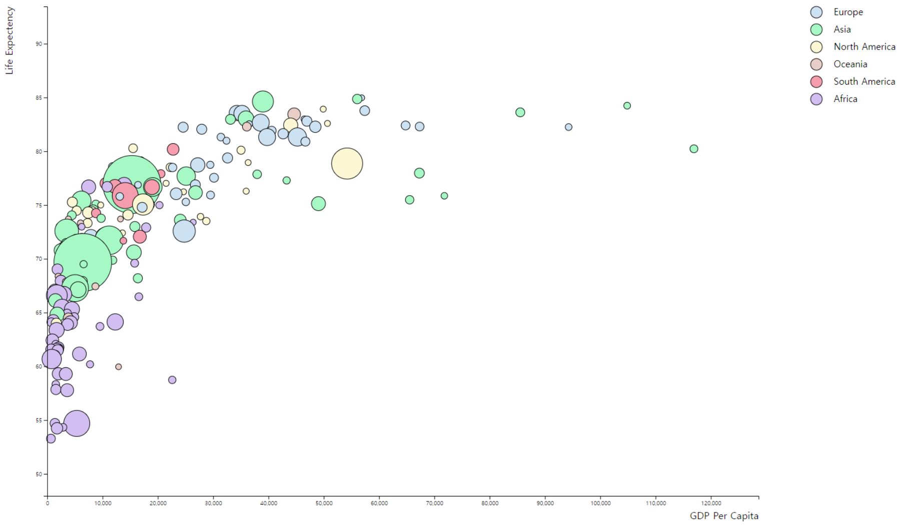

Figure 2: The completed bubble chart.

2.2 Implementing Bubble Chart

- Define a scale named xScale for the x-axis. (2pts)

  You should set the x-axis range from **0** to **110% of the highest GDP Per Capita** with evenly spaced ticks.

- Define a scale named yScale for the y-axis. (2pts)

  You should set the y-axis range from **90% of the lowest life expectancy** to **110% of the highest life expectancy** with evenly spaced ticks.

- Extract all unique continent values and save them in the variable continentList. (2pts)

  Make sure that the order of the continent follows **the order it appears in the dataset**.

  e.g.) If continent values in dataset appear like [Asia, Asia, Europe, Asia, Africa, Europe, Europe, Asia, Africa], the continentList must be [Asia, Europe, Africa]

- Define a scale named cScale for mapping colors to the bubbles. (2pts)

  Color the bubbles differently depending on the continent with the color palette:

  ```
  ['#cce1f2', '#a6f8c5', '#fbf7d5', '#e9cec7', '#f59dae', '#d2bef1']
  ```

- Define a scale named sScale for the size of the bubbles. (4pts)

  Your **bubble's radius** should range from **5** to **50**.

  And your **bubble's area** must be proportional to **the population of the country**.

- Draw Bubbles following the instructions above. (7pts)

Your result should look like Figure 2.

3 Submission

After you complete tasks, you should zip all the files into **YourStudentId_YourName.zip**. (e.g. 20251234_JohnDoe). Your file must look like this:

```
YourStudentId_YourName.zip
├─ Prob1
│  ├─ data
│  │  └─ life_expectancy_by_country.csv
│  ├─ js
│  │  ├─ index.js
│  │  └─ main.js
│  ├─ Ans1.PNG (Not essential)
│  ├─ index.html
│  └─ package.json
├─ Prob2
│  ├─ data
│  │  └─ owid-covid-data.csv
│  ├─ js
│  │  ├─ index.js
│  │  └─ main.js
│  ├─ Ans2.PNG (Not essential)
│  ├─ index.html
│  └─ package.json
└─ Instruction.pdf (Not essential)
```

You should upload this file via Blackboard.