

# EECS E6893 Big Data Analytics - Fall 2024

## Homework Assignment 3: Data Visualization

Due Friday, November 1st, 2024, by 5:00pm

### Assignment 3 guidelines:

Link to Dataset in Part I Task: <https://github.com/vega/vega/blob/main/docs/data/seattle-weather.csv>

Link to Dataset in Part II Task: <https://github.com/plotly/datasets/blob/master/auto-mpg.csv>

Dataset for Part III Task: Les Misérables dataset on CourseWorks

Submit a **single** pdf with code screenshots and output screenshots. Properly label all the plots and give titles to all the plots (even if not mentioned explicitly). It is expected that x and y axes are labeled properly, any legends used are displayed with correct color coding, any scaling done is displayed on the plots. The plots must be clear and readable. Axes tick labels should be clearly visible.

Also submit all html files for the code written as a single zip folder with name as UNI\_Name\_HW3.zip. Your submission should be **one** PDF file with screenshots of results and code AND **one** zip file containing all html files. Submission that are not in the right "format" will not be graded.

### **Part I Tasks: (20 marks)**

#### **Problem 1 (10 marks)**

Create webpages with HTML, CSS and plot various plots as asked in each subpart with d3.js. (follow tutorial for help). Use the **Seattle weather** dataset for getting the output. Paste the code snippets and output pictures in your submission file.

1. Plot basic histogram for the wind variable using D3, set bins = 10. Label the X and Y axis properly.
2. Now, change the number of bins to 25.
3. Now, plot a pie chart using D3 to show the distribution of the weather variable, properly label the sectors with percentage and annotate each sector with corresponding weather variable.
4. Now, plot a line graph using D3 to show the trend of the precipitation variable with respect to the date variable. Label the X and Y axis properly.

5. Add a paragraph to the webpage using the `<p>` tag to note down the main observations from the above obtained plots in 1, 2, 3, 4 (for eg. distribution, count, important understanding etc).
6. Use the internal style sheet to change the style (font size, font color, indentation) of the paragraph of the web page you created in 5.

**Homework Submissions:** Screenshot of your code and the output result.

### **Problem 2 (10 marks)**

Now that we have built some basic static graphs using D3.js. We will be moving on to build some dynamic graphs (interactive graphs) to get more understanding of the dataset. Use the **Seattle weather** dataset for getting the output. Paste the code snippets and output pictures in your submission file.

1. Using the same histogram from 1.2.1 add an interactive component such that we can toggle through the bins size. (We should be able to select any bin size from range 5-25 with increments of 5)
2. Add a drop-down menu with precipitation, temp\_max, temp\_min and wind variables to get histograms for each of these variables upon selection of each variable from the dropdown menu.

**Homework Submissions:** Screenshot of your code and the output result.

## **Part II Tasks: (30 marks)**

In this part you are going to create a basic webpage using HTML, CSS and Javascript and deploy the web page on Apache web server. You will be working on Data Preprocessing, Data Visualization and deploying the website on Apache web server. Use the **Auto mpg** dataset for getting the output. Paste the code snippets and out pictures in your submission file.

### **Problem 1 (10 marks)**

Using the **Auto mpg** dataset. First you need to do preprocessing of the data to get required columns for data visualization.

2.1.1 Load the csv file to a Javascript array. Display the top 5 rows of the Javascript array in the webpage as a table in HTML.

2.1.2 Create a new array to get the count of the number of cars of each model year and display the thus obtained Javascript array as a table in HTML.

2.1.3 Create a new array to get the total number of cylinders for each model year and display the thus obtained Javascript array as a table in HTML.

2.1.4 Create a new array to get the count of the number of cars of each acceleration and display the thus obtained Javascript array as a table in HTML.

**Homework Submissions:** Screenshot of your code and the output result.

### **Problem 2 (10 marks)**

Now that you have preprocessed the data and have new required arrays you will have to visualize the data in **a single HTML webpage** using various plots in D3.

2.2.1 Using the array created in 2.1.2, plot a pie chart using D3 along with the table from 2.1.2 to get the distribution of the model years amongst the cars, properly label the sectors with percentage and annotate each sector with corresponding model year.

2.2.2 Using the array created in 2.1.3, plot a line graph using D3 along with the table from 2.1.3 to see the total number of cylinders for each model year, label the x and y axis.

2.2.3 Using the array created in 2.1.4, plot a histogram using D3 along with the table from 2.1.4 for acceleration with bins equal to 10 label x and y axis appropriately.

**Homework Submissions:** Screenshot of your code and the output result.

**Problem 3 (10 marks)**

Now that our webpage is ready with the visualized plots. We will deploy the webpage on the Apache web server. Follow the tutorial or correct installation and deployment steps

2.3.1 Follow the tutorial for installing Apache HTTP server

2.3.2 Create a Virtual Host

2.3.3 Host your webpage (webpage from Part II problem 2) on the virtual host.

**Homework Submissions:** Screenshot of your code and the output result.

### **Part III Tasks: (25 marks)**

In this part you are going to create a basic webpage using HTML, CSS and Javascript and deploy the web page on Apache web server. You will be working on Graph Analysis and deploying the website on Apache web server. Use the Le Miserables dataset for getting the output. Paste the code snippets and out pictures in your submission file.

#### **Problem 1 (25 marks)**

Les Misérables (1862). Hugo's novel follows the lives and interactions of numerous characters in France leading up the 1832 June Rebellion in Paris.

- 3.1.1 Plot a Network graph that shows character co-occurrence in Les Misérables.
- 3.1.2 Add labels to each node. (i.e. each node should show the name of the character)
- 3.1.3 It can be difficult to observe micro and macro features simultaneously with complex graphs. If you zoom in for detail, the graph is too big to view in its entirety. Use fisheye distortion (magnifies the local region around the mouse, while leaving the larger graph unaffected for context) for the above network graph. (you can use D3's fisheye plugin).
- 3.1.4 Add a <p> tag to your HTML file and infer what you see from the graph.

Now that our webpage is ready with the visualized plots. We will deploy the webpage on the Apache web server. Follow the tutorial or correct installation and deployment steps.

- 3.1.5 Add a <p> tag to your HTML file and infer what you see from the graph.
- 3.1.6 Host your webpage (from 3.1.3) on the virtual host.

**Homework Submissions:** Screenshot of your code and the output result.

#### **Part 4 Tasks: (25 marks)**

- *This part of assignment uses polygon API (from previous assignment) to fetch data form a RestAPI and display that dynamically on a webpage.*

#### **Problem 1 (10 marks): Finance data visualization from Polygon API**

Create webpages with HTML, CSS and display the stock price for any 3 stocks of your choice for the current timestamp and the difference of current and previous timestamp. You can use colors to depict the increase or decrease in the price. Your visualization should be dynamic and should display the result by appending one below the another on the webpage itself.

**Homework Submissions:** Screenshot of your code and the output result.

#### **Problem 2 (15 marks): Network Graph for Companies**

Create a webpage to display a network graph for companies based on their closing stock price for the previous years. Companies to use: IBM (center of the network), Apple, Google, Intel, Microsoft and Nvidia.  
Run a loop to update the network to run 3 times (closing price for 2023, 2022, 2021) and displaying the graph/network one below the other and the webpage should also update without reloading.

#### **Pointers:**

Fetch data for the closing price of each stock for the last day in the financial year and store it, use those values and find correlation (think how this can be done) between them. The network edge nodes' distance from the vertex (IBM) should correlate to their closing price of that year.

Make limited API calls to avoid getting rate-limited, it is okay to call the API, store the data and then once you receive all the stock data you can display the graph/network.

**Homework Submissions:** Screenshot of your code and the output result.