

## CS424 - Visualization & Visual Analytics (Fall 2023)

Instructor: Fabio Miranda

Course webpage: <https://fmiranda.me/courses/cs424-fall-2023/>

### Assignment 4

The goal of this project is to build a complete web-based visualization system leveraging outcomes from the previous assignments. We will use JavaScript and a visualization library to visualize and interactively analyze a dataset. Your visualization should be hosted on a publicly available website and you can use any web hosting service you prefer. Options include [GitHub](#), [Heroku](#), and [Google Sites](#). Once again, we will use GitHub to manage documentation and code. You should create a repository for your team on GitHub Classroom (see Piazza for invitation link). We will use GitHub to manage code and documentation.

### Tasks

Up until now, you have created isolated visualizations to explore a particular dataset. The tasks below are designed so that you can integrate your previous outcomes into a single web-based visualization system to explore the dataset you chose. You are free to re-use your previous assignments datasets and outcomes.

#### Task 0: Setting up a local server

Before moving your system to the web, it is easier to develop using a local server (i.e., a server running on your own computer). There are several ways to achieve that but, for the purposes of this assignment, the most straightforward one is to simply use Python's HTTP servers. In the folder where you will store your code, type in the command line:

```
python -m http.server
```

That will serve all files in your folder through the address 0.0.0.0:8000 . Open the address in a browser and you will be able to see the served files.

#### Task 1: Creating an empty page

Create a web page to host your visualizations. You should start by defining the div elements that will serve as the parent element for your individual visualizations (check [here](#) to refresh your memories). Ideally, you should also define the *size* of each element in a style sheet -- make sure you use percentage as opposed to size in pixels. After you have outlined the position of your visualization elements in your screen, create a JavaScript file (main.js) that will contain the code to (1) load your data and (2) generate your visualizations. Again, revisit [this](#) lecture to make sure you are familiar with the interaction between HTML and JavaScript.

For a possible starting point, you can check [this](#) template. This is just a template, and you should not expect it to satisfy *all* the requirements of the assignment. In other words, do not simply paste your previous Vega-lite specifications into the .js file, without adequately addressing questions of interaction, size, and positioning. The zip file contains three files: `index.html` containing six div elements, `style.css` defining the position of the divs, and `script.js` with the code to generate the Vega-lite visualizations.

In the `index.html` file, notice that we are including the appropriate Vega libraries:

```
<script src="https://cdn.jsdelivr.net/npm/vega@5"></script>
<script src="https://cdn.jsdelivr.net/npm/vega-lite@5"></script>
<script src="https://cdn.jsdelivr.net/npm/vega-embed@6"></script>
```

Following the steps outlined [here](#), the `script.js` file calls the `vegaEmbed` function to embed specifications into the HTML file.

Notice that the example provided in the template loads raw data from HTTP addresses. However, depending on your data, that might not be feasible, as you might need to access preprocessed data files instead.

For a list of useful resources, click [here](#).

## Task 2: Preprocess and host data files

Given the size of your data, you should not expect that Vega-lite will support your raw data. Instead, you might need to preprocess the file and save *visualization appropriate* data files. By *visualization appropriate*, I mean data that only contains attributes and aggregations that will be used in the visualization. Based on your previous assignments, store the precomputed data (either as JSON or CSV files) in a folder that is served by your local server.

## Task 3: Using previous visualizations

In the previous assignment, you have already created a set of interactive visualizations. Re-use these visualizations in your new web page, making sure you *maintain* the same level of interactivity, i.e., if you brush a particular view, the linked views are updated. In the previous project you had to create at least **two** interactive visualizations (one using single view and another using multiple views) -- you should now port them to your new integrated visualization interface.

Make sure to appropriately modify the div elements so that the size and position of your visualizations are adequate.

## Task 4: New multiple linked view

Create a new multiple linked view visualization, using a different interaction mechanism (i.e., manipulating the data, manipulating the visual mapping, manipulating the view) or method (i.e., aggregation, filtering, change mapping, selection, navigation, spatial arrangement) from the ones you used in the previous task. Again, make sure your plots avoid common problems, such as overplotting.

Even though the template provides you with a starting point, you should go beyond that. Consider adding HTML elements (such as [drop down menus](#)) to enable and/or enhance user interaction. For example, a reasonable (yet simple) approach would be to have a drop down menu to select attributes for a bar chart. You should carefully think about the interplay of HTML and Vega-lite interactions.

## Task 5: Spatial visualization

Migrate or create a new visualization leveraging the spatial information from your dataset. For extra points, you should link the spatial view with a reasonable subset of the views from the previous tasks.

## Task 6: Web deployment

Select a web hosting service and make yourself familiar with the steps needed in order to upload files to the service. Free options include [GitHub](#), [Heroku](#), and [Google Sites](#). Make sure you upload your final results to the hosting service so that they are publicly available.

## Task 7: UTK (extra)

Now that you have created Vega-lite visualizations, explore the possibility of migrating them to the [UTK framework](#). As a starting point, check the [quick start guide](#) and [tutorials](#). For 5 extra points in the assignment, you should successfully (1) load your data or a subset of it into UTK, (2) create a map-based visualization, and (3) create a linked 2D visualization within the environment using Vega-lite. If you are not able to create new visualizations using UTK, you can simply write the steps (and errors) encountered in the process. You do not need to upload your UTK-based visualizations to the web hosting service; instead, simply add your screenshots (and JSON specifications) to the GitHub.

---

## Deliverables

In this project, there will be four deliverables:

- 1) A [markdown document](#) (.md), with a clear description of the dataset, summarizing the most important points, questions, data transformations, encodings, interactions and initial findings (feel free to re-use text from previous assignments). Also include a link to your visualization solution (see deliverable 4). You should include screenshots with meaningful captions. **Please put some effort into it as outcomes like this can be helpful when building up a portfolio.**
- 2) Source code and any necessary data files.
- 3) A png image showing your visualization interface. I might use your screenshot to link your web page in the course notes, so

please make sure to have a good screenshot.

4) A web page hosting your visualization interface. The web page should be publicly accessible.

5) A ~5 minute presentation, presenting your visualization interface. For the presentation, you should present your interface, interactions and findings, **without** relying on a Powerpoint presentation. In other words, for your presentation you should simply open the webpage hosting your visualization (see task 1) and present it live.

---

## Grading

Your submission will be graded according to the quality and coverage of the results and presentation. You need to make sure that both your deliverables (presentation and documentation) clearly show your findings. Visualizations need to be well constructed, with good color and font choices and proper labeling. Your submission and presentation will be graded according to the following:

- Visualizations:
  - Excellent (A): well-rounded and thoughtful visualizations, clear description.
  - Good (B): good description of the visualizations but could be more detailed.
  - Poor (C): missing or incomplete descriptions and / or visualizations.
- Presentation:
  - Excellent (A): clearly described the data and the path between sketches and visualizations, good discussion regarding design decisions.
  - Good (B): good description of the data and visualizations, but could be more detailed.
  - Poor (C): missing or incomplete descriptions and / or visualizations.

Points will be deducted if your visualizations suffer from errors or broken features, overly simplistic designs, incomplete write-up.

If you successfully create visualizations using UTK, you will get 5 extra points in the assignment. If you document your process of using UTK (even if not successful), you will get 5 extra points.