Midterm Project

Due: Friday, March 14th by 11:59pm

Midterm Project Overview: The midterm project marks the transition from exploring individual components of a data system to integrating them into a full-stack application. So far, you've worked with:

- A web front end (JavaScript and D3)
- A web server (Python/Flask with Pandas)
- A database backend (DuckDB)

Now, your task is to bring these components together into a fully functional visual analytics system.

No Starter Code—You Build It from Scratch: Unlike previous labs, we're not providing support code—only the dataset. You are expected to build the system from scratch, but you may reuse code from earlier labs, including any provided support code.

What You Need to Build: Your system should implement a **single parameterized SQL query** approach, similar to the HomeFinder example from Lecture 2. That means your visualization interface will act as a wrapper around a primary SQL query. You may need additional queries (e.g., fetching min/max values for an axis, as in Lab 5: Forest Fire), but the core interaction should follow this model.

Data Pipeline: To demonstrate the full data pipeline, your system must:

1. Retrieve relevant data from the database via SQL.
2. Perform statistical analysis on the retrieved data in Python.
3. Send the processed data and computed statistics to the front end via Flask.
4. Update the visualization using JavaScript/D3.
5. Respond to user interactions by repeating steps 1–4.

Constraints and Expectations: While there are multiple ways to implement the system (e.g., performing statistical analysis in DuckDB, or downloading the full dataset from the database into a data frame), **you must follow the specified pipeline** to ensure hands-on experience with all components.

Finally, while you haven't explicitly practiced using Python for statistical analysis in previous labs, we expect you to figure this out independently. You're free to use any library that fits your needs (e.g., scikit-learn) if Pandas doesn't suit your approach.

Objectives:

- Learn to integrate the technology pieces (Javascript/D3.js frontend, Python/Flask/Pandas webserver, database backend)
- Build a visual analytics system from scratch to analyze a given dataset

How to get started:

- Download the “student outcomes” dataset
- Reuse code from previous labs as needed

Requirements:

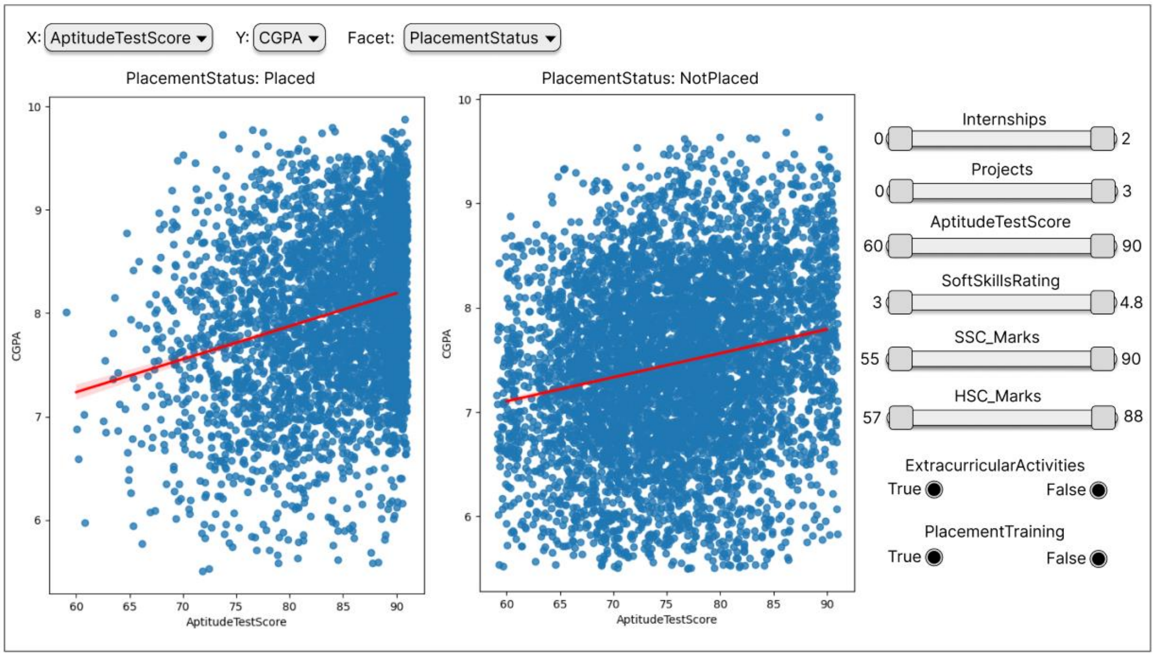
- Follow the structured data pipeline described above.
- Minimum requirement: the visual analytics system should function as a wrapper around a single parameterized SQL query.
    - If you’d like to extend the system with multiple visualizations, each with its own corresponding parameterized query, you’re welcome to do so.
- We suggest a visualization design (see below), but you are not required to use it. Feel free to be creative as long as your system adheres to the prescribed data pipeline and utilizes parameterized SQL queries for data retrieval.
    - If you create your own design, consider Camelia’s lecture on Design Methodology and think critically about task definition and interface design.
- Summary Report: a document summarizing your project, including:
    - How you built your system
        - Include task definition and interface design
        - Include the parameterized SQL query (or queries)
    - How you used it to explore/analyze the dataset
        - Include screenshots
    - Describe key insights derived from using your system to analyze the data

Teams:

- You may form your own teams.
- Each team should have three members. If your team has more or fewer than three, please notify us.

How to submit:

- Submit your project via Canvas, including:
    - Source code (.html, .js, .py, .css files)
    - Summary report



X: AptitudeTestScore ▾ Y: CGPA ▾ Facet: PlacementStatus ▾

Internships 0 – 2

Projects 0 – 3

AptitudeTestScore 60 – 90

SoftSkillsRating 3 – 4.8

SSC_Marks 55 – 90

HSC_Marks 57 – 88

ExtracurricularActivities True ● False ●

PlacementTraining True ● False ●