Project Proposal

- Name
    - City-Level Employment Statistics: Interactive Visualizations of Employment Statistics for Top US Cities
- Label
    - city-level-employment
- Goal/Plan
    - The aim of my project is to clearly represent recent employment statistics and trends for individual large US cities. I plan to provide a dashboard/page through which users can visualize city-level data from the federal Bureau of Labor Statistics (BLS) around key employment/unemployment metrics. I will display city-specific data on an interactive map of the US and provide options for users to select specific metrics to visualize. I also plan to provide options for users to select specific cities in order to view charts showing city-specific details, including the paths/trends of key metrics over recent histor.

Outline/Plan of Work to Do (WIP)

- Data
    - Dataset(s)
        - BLS Local Area Unemployment Statistics
            - Series ID composition (specify each individual area and metric required)
                - Metric codes
                - Area/location codes
                    - Area/location type codes
            - Example API call/response
            - Sample Python code
        - SimpleMaps US Cities database
            - CSV download into input_data/ folder
    - API call / data processing steps
        - Read downloaded CSV file, create lookup table/DF of top US cities with Lat/Long coordinates and populations
        - Copy BLS Area Codes into table/DF, parse for just city codes, create lookup table/DF of codes for large US cities

        - Copy BLS Metric Codes into lookup table/DF, refine to most useful metrics to offer/display
        - Iterate through top 100 cities and selected metrics to construct API call(s) for JSON data
        - Parse JSON data/response for city+metric values and store in table/DF
    - Challenge 6 / Python API Challenge
    - Challenge 11 / Web Scraping Challenge
    - Challenge 3 / Python Challenge
- Database
    - Process data into SQLite database
        - Draw ERD
        - Create SQLite tables
    - Challenge 10 / SQL Alchemy Challenge
    - Project 2 / Crowdfunding ETL
- Flask API
    - Determine options/interactions/routes to offer, including HTML/CSS/JS
        - ? Separate from dropdowns/selections for the user, or are those just created through Flask initially and then modified with JS/HTML/CSS customization ? (routes for cities, dropdown for metrics..?)
    - Challenge 10 / SQL Alchemy Challenge
- Visualizations/Dashboard
    - Provide options/interactions with multiple charts/visualizations (including a map, ideally)
        - Select a metric from dropdown menu to change marker circle sizes/colors based on the selected metric's latest values
        - Select a city to see charts of that city's key employment metrics over time
    - Use a JS library not reviewed in class
    - 14 / Belly Button Challenge + 15 / Leaflet Challenge
- Presentation
    - Build slides
    - Project 1 for slide template

Project Requirements

- Data and Delivery (25 points)

    - Data components used in the project are clearly documented. (5 points)
    - The dataset contains at least 100 unique records. (5 points)
    - A database is used to house the data (SQL, MongoDB, SQLite, etc.). (5 points)
    - The project is powered by a Python Flask API and includes HTML/CSS, JavaScript, and the chosen database. (10 points)
- Back End (25 points)
    - The page created to showcase data visualizations runs without error. (7.5 points)
    - A JavaScript library not shown in class is used in the project. (7.5 points)
    - The project conforms to one of the following designs: (10 points)
        - A Leaflet or Plotly chart built from data gathered through web scraping.
        - A dashboard page with multiple charts that all reference the same data.
- Visualizations (25 points)
    - A minimum of three unique views present the data. (5 points)
    - Multiple user-driven interactions (such as dropdowns, filters, or a zoom feature) are included on the final page. (5 points)
    - The final page displays visualizations in a clear, digestable manner. (5 points)
    - The data story is easy to interpret for users of all levels. (10 points)
- Group Presentation (25 points)
    - All group members speak during the presentation. (5 points)
    - The content is relevant to the project. (5 points)
    - The presentation maintains audience interest. (5 points)
    - Content, transitions, and conclusions flow smoothly within any time restrictions. (10 points)