City-Level Unemployment Map

Interactive Map of Recent (August 2023) City-Level Unemployment Data for Top 100 US Cities

Ryan McReynolds

September 26, 2023

GitHub repo: https://github.com/ryanmcr17/city-level-employment

Executive Summary

Project Plan/Proposal

The aim of this project is to clearly represent recent employment statistics 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 unemployment metrics. I will display city-specific data on an interactive map of the US with multiple visualization options for users.

Key Files/Folders

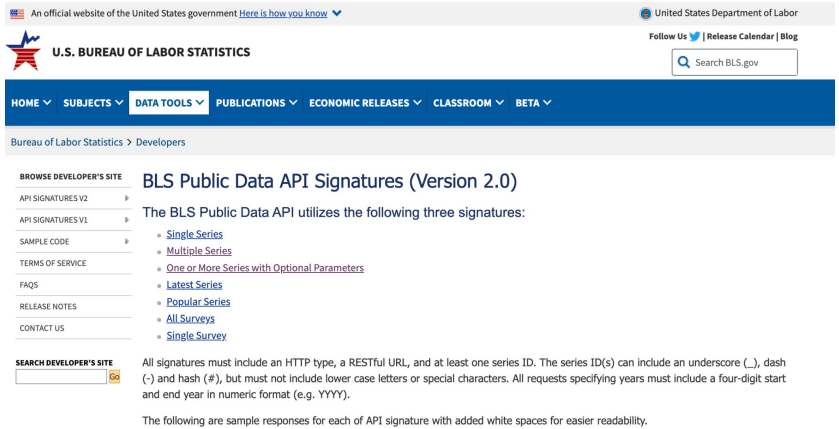
- all data processing / ETL work contained within cityDataETL/ folder
  - cityDataETL.ipynb contains the bulk of the ETL work
  - .sqlite file contains the cleaned-up database produced by the Python ETL code
  - CityUnemploymentAPI.py is separate Python code which generates a Flask API allowing users to call the city data from the SQLite DB in JSON format
- index.html is the primary file to open which loads the interactive map
- the static/ folder contains the JS file that processes the city data (pulled from the Flask API) into circle markers and builds the Leaflet map

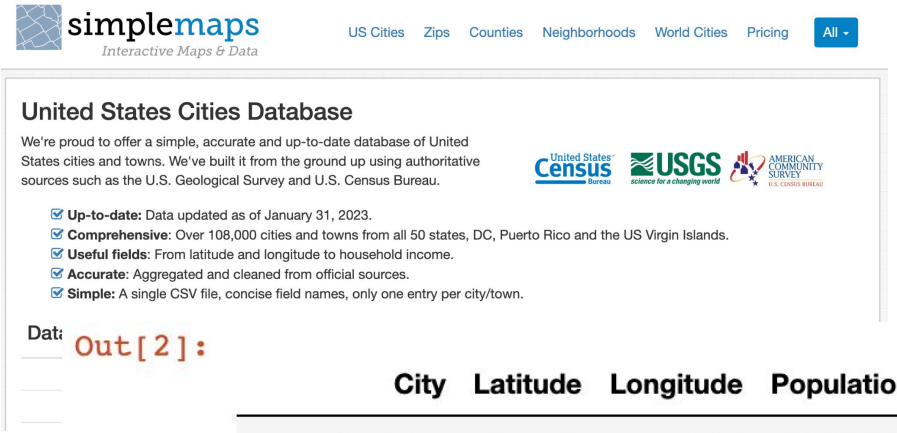
Project

Data Processing / ETL

Key Data Sources

City-level unemployment data pulled from the Bureau of Labor Statistics (BLS) API
CSV file of city lat/long coordinates and populations downloaded from https://simplemaps.com/data/us-cities





	City	Latitude	Longitude	Population
0	New York	40.6943	-73.9249	18972871
1	Los Angeles	34.1141	-118.4068	12121244
2	Chicago	41.8375	-87.6866	8595181
3	Miami	25.7840	-80.2101	5711945
4	Dallas	32.7935	-96.7667	5668165

Data Wrangling

'Area codes' for US cities/locations scraped from BLS page and parsed as .tsv string
BLS 'area codes' merged with city DF containing lat/long coordinates and population data
'Area codes' used to build out BLS API URLs for unemployment data calls

	area_type_code	area_code	area_text	display_level	selectable	sort_sequence
0	A	ST0100000000000	Alabama	0	T	1
1	A	ST0200000000000	Alaska	0	T	149
2	A	ST0400000000000	Arizona	0	T	193
3	A	ST0500000000000	Arkansas	0	T	257
4	A	ST0600000000000	California	0	T	383

	City Area Code	city_state_key
0	CT0100820000000	Alabaster, AL
1	CT0101852000000	Anniston, AL
2	CT0102956000000	Athens, AL
3	CT0103076000000	Auburn, AL
4	CT0105980000000	Bessemer, AL

	City	Latitude	Longitude	Population	Unemployment Rate Series ID	Unemployment Series ID	Labor Force Participation Rate Series ID	City Area Code
0	New York, NY	40.6943	-73.9249	18972871	LAUCT365100000000003	LAUCT365100000000004	LAUCT365100000000008	CT3651000000000
1	Los Angeles, CA	34.1141	-118.4068	12121244	LAUCT064400000000003	LAUCT064400000000004	LAUCT064400000000008	CT0644000000000
2	Chicago, IL	41.8375	-87.6866	8595181	LAUCT171400000000003	LAUCT171400000000004	LAUCT171400000000008	CT1714000000000
3	Miami, FL	25.7840	-80.2101	5711945	LAUCT124500000000003	LAUCT124500000000004	LAUCT124500000000008	CT1245000000000
4	Dallas, TX	32.7935	-96.7667	5668165	LAUCT481900000000003	LAUCT481900000000004	LAUCT481900000000008	CT4819000000000
5	Houston, TX	29.7860	-95.3885	5650910	LAUCT483500000000003	LAUCT483500000000004	LAUCT483500000000008	CT4835000000000
6	Atlanta, GA	33.7628	-84.4220	5046555	LAUCT130400000000003	LAUCT130400000000004	LAUCT130400000000008	CT1304000000000
7	Washington, DC	38.9047	-77.0163	4810669	LAUCT115000000000003	LAUCT115000000000004	LAUCT115000000000008	CT1150000000000
8	Boston, MA	42.3188	-71.0852	4208580	LAUCT250700000000003	LAUCT250700000000004	LAUCT250700000000008	CT2507000000000
9	Phoenix, AZ	33.5722	-112.0892	4047095	LAUCT045500000000003	LAUCT045500000000004	LAUCT045500000000008	CT0455000000000

Final ETL Output

Unemployment data pulled via BLS API and merged back with city location/population data
Final cleaned-up dataframe saved to SQLite DB file

	Unemployment Rate Series ID	August 2023 Unemployment Rate
0	LAUCT365100000000003	5.6
1	LAUCT064400000000003	5.7
2	LAUCT171400000000003	4.4
3	LAUCT124500000000003	1.8
4	LAUCT481900000000003	4.3

	Unemployment Series ID	August 2023 Unemployment
0	LAUCT365100000000004	234715
1	LAUCT064400000000004	117804
2	LAUCT171400000000004	61024
3	LAUCT124500000000004	4350
4	LAUCT481900000000004	32665

	city	latitude	longitude	population	unemploymentRate	unemploymentCount
0	New York, NY	40.6943	-73.9249	18972871	5.6	234715
1	Los Angeles, CA	34.1141	-118.4068	12121244	5.7	117804
2	Chicago, IL	41.8375	-87.6866	8595181	4.4	61024
3	Miami, FL	25.7840	-80.2101	5711945	1.8	4350
4	Dallas, TX	32.7935	-96.7667	5668165	4.3	32665

Flask API

Flask API

Separate Python file creates the Flask API which responds to API calls with the city dataset from the SQLite database in JSON format

```python
@app.route("/USCityUnemploymentDATA", methods=['GET'])
def unemploymentDATA():
    """Return the city-level data (location, population, unemployment) as JSON"""

    # Create our session (link) from Python to the DB
    session = Session(engine)


    # Query all city unemployment data from the database + tranfer to DF then to dictionary then return as JSON

    city_data = session.query(CityData).all()

    city_df = pd.DataFrame([(cd.city, cd.latitude, cd.longitude, cd.population, cd.unemploymentRate, cd.unemploymentCount) for

    print(city_df)


    df_to_json = city_df.to_json(orient="records")
    parsed_json = loads(df_to_json)
    json_city_data = dumps(parsed_json, indent=4)


    return json_city_data

    session.close()
```

127.0.0.1:5000/USCityUnemploymentDATA

[{ "city": "New York, NY", "latitude": "40.6943", "longitude": "-73.9249", "population": "18972871", "unemploymentRate": "5.6", "unemploymentCount": "234715" }, { "city": "Los Angeles, CA", "latitude": "34.1141", "longitude": "-118.4068", "population": "12121244", "unemploymentRate": "5.7", "unemploymentCount": "117804" }, { "city": "Chicago, IL", "latitude": "41.8375", "longitude": "-87.6866", "population": "8595181", "unemploymentRate": "4.4", "unemploymentCount": "61024" }, { "city": "Miami, FL", "latitude": "25.7840", "longitude": "-80.2101", "population": "5711945", "unemploymentRate": "1.8", "unemploymentCount": "4350" }, { "city": "Dallas, TX", "latitude": "32.7935", "longitude": "-96.7667", "population": "5668165", "unemploymentRate": "4.3", "unemploymentCount": "32665" }, { "city": "Houston, TX", "latitude": "29.7860", "longitude": "-95.3885", "population": "5650910", "unemploymentRate": "5.2", "unemploymentCount": "62158" }, { "city": "Atlanta, GA", "latitude": "33.7628", "longitude": "-84.4220", "population": "5046555", "unemploymentRate": "3.9", "unemploymentCount": "10846" }, { "city": "Washington, DC", "latitude": "38.9047", "longitude": "-77.0163", "population": "4810669", "unemploymentRate": "5.5", "unemploymentCount": "21877" }, { "city": "Boston, MA", "latitude": "42.3188", "longitude": "-71.0852", "population": "4208580", "unemploymentRate": "3.0", "unemploymentCount": "11719" }, { "city": "Phoenix, AZ", "latitude": "33.5722", "longitude": "-112.0892", "population": "4047095", "unemploymentRate": "4.2", "unemploymentCount": "38801" }, { "city": "Detroit, MI", "latitude": "42.3834", "longitude": "-83.1024", "population": "3522856", "unemploymentRate": "8.3", "unemploymentCount": "21211" }, { "city": "Seattle, WA", "latitude": "47.6211", "longitude": "-122.3244", "population": "3438221", "unemploymentRate": "3.4", "unemploymentCount": "17375" }, { "city": "San Diego, CA", "latitude": "32.8313", "longitude": "-117.1222", "population": "3084174", "unemploymentRate": "3.8", "unemploymentCount": "27621" }, { "city": "Minneapolis, MN", "latitude": "44.9635", "longitude": "-93.2678", "population": "2856952", "unemploymentRate": "3.1", "unemploymentCount": "7864" }, { "city": "Tampa, FL", "latitude": "27.9945", "longitude": "-82.4447", "population": "2683956", "unemploymentRate": "3.2", "unemploymentCount": "7337" }, { "city": "Baltimore, MD", "latitude": "39.3051", "longitude": "-76.6144", "population": "2205092", "unemploymentRate": "2.3", "unemploymentCount": "6439" }, { "city": "Las Vegas, NV", "latitude": "36.2333", "longitude": "-115.2654", "population": "2150373", "unemploymentRate": "6.3", "unemploymentCount": "20610" }, { "city": "St. Louis, MO", "latitude": "38.6359", "longitude": "-90.2451", "population": "2092481", "unemploymentRate": "4.3", "unemploymentCount": "6679" }, { "city": "Portland, OR", "latitude": "45.5371", "longitude": "-122.6500", "population": "2036875", "unemploymentRate": "3.7", "unemploymentCount": "14126" }, { "city": "Riverside, CA", "latitude": "33.9381", "longitude": "-117.3949", "population": "2022285", "unemploymentRate": "4.5", "unemploymentCount": "7153" }, { "city": "Orlando, FL", "latitude": "28.4773", "longitude": "-81.3370", "population": "1927699", "unemploymentRate": "3.0", "unemploymentCount": "5500" }, { "city": "Sacramento, CA", "latitude": "38.5677", "longitude": "-121.4685", "population": "1924167", "unemploymentRate": "4.6", "unemploymentCount": "11353" }, { "city": "San Antonio, TX", "latitude": "29.4632", "longitude": "-98.5238", "population": "1910785", "unemploymentRate": "4.2", "unemploymentCount": "33127" }, { "city": "San Jose, CA", "latitude": "37.3012", "longitude": "-121.8480", "population": "1729879", "unemploymentRate": "3.7", "unemploymentCount": "20468" }, { "city": "Pittsburgh, PA", "latitude": "40.4397", "longitude": "-79.9763", "population": "1720279", "unemploymentRate": "3.5", "unemploymentCount": "5472" }, { "city": "Cincinnati, OH", "latitude": "39.1413", "longitude": "-84.5060", "population": "1712287", "unemploymentRate": "3.7", "unemploymentCount": "5721" }, { "city": "Cleveland, OH", "latitude": "41.4764", "longitude": "-81.6805", "population": "1683059", "unemploymentRate": "4.5", "unemploymentCount": "7261" }, { "city": "Austin, TX", "latitude": "30.3005", "longitude": "-97.7522", "population": "1659251", "unemploymentRate": "3.7", "unemploymentCount": "25096" }, { "city": "Kansas City, MO", "latitude": "39.1238", "longitude": "-94.5541", "population": "1644497", "unemploymentRate": "3.6", "unemploymentCount": "9934" }, { "city": "Columbus, OH", "latitude": "39.9862", "longitude": "-82.9855", "population": "1556848", "unemploymentRate": "3.3", "unemploymentCount": "16045" }, { "city": "Charlotte, NC", "latitude": "35.2083", "longitude": "-80.8303", "population": "1516107", "unemploymentRate": "3.5", "unemploymentCount": "18784" }, { "city": "Virginia Beach, VA", "latitude": "36.7335", "longitude": "-76.0435", "population": "1500764", "unemploymentRate": "2.5", "unemploymentCount": "6173" }, { "city": "Milwaukee, WI", "latitude": "43.0642", "longitude": "-87.9675", "population": "1340981", "unemploymentRate": "4.6", "unemploymentCount": "12760" }, { "city": "Providence, RI", "latitude": "41.8230", "longitude": "-71.4187", "population": "1270149", "unemploymentRate": "3.1", "unemploymentCount": "2749" }, { "city": "Jacksonville, FL", "latitude": "30.3322", "longitude": "-81.6749", "population": "1220191", "unemploymentRate": "3.4", "unemploymentCount": "17502" }, { "city": "Salt Lake City, UT", "latitude": "40.7776", "longitude": "-111.9311", "population": "1135344", "unemploymentRate": "2.9", "unemploymentCount": "3875" }, { "city": "Raleigh, NC", "latitude": "35.8324", "longitude": "-78.6429", "population": "1062018", "unemploymentRate": "3.4", "unemploymentCount": "9451" }, { "city": "Memphis, TN", "latitude": "35.1087", "longitude": "-89.9663", "population": "1034498", "unemploymentRate": "5.4", "unemploymentCount": "15901" }, { "city": "Richmond, VA", "latitude": "37.5295", "longitude": "-77.4756", "population": "1008069",

JavaScript+Leaflet

JavaScript+Leaflet

Upon loading the HTML, JavaScript calls the Flask API, processes the JSON data on cities, creates 2 sets of Leaflet circle markers the visualize the data in different ways, and builds the Leaflet map with layers

```javascript
// save API endpoint
const baseUrl = "http://127.0.0.1:5000";
const APIurl = baseUrl + "/USCityUnemploymentDATA";
console.log('APIurl = ' + APIurl);


// D3 request to the API URL
d3.json(APIurl).then(APIresponse => createUnemploymentMap(APIresponse)).catch(console.log("No response from the API - make sure the Flask API is active/serving"));
console.log('APIresponse = ' + APIresponse);


// function to define/create the features (specific earthquakes) and to create the map
function createUnemploymentMap(city_data) {

  console.log('API response / city_data = ' + city_data);

  let cityMarkers = [];
  let cityMarkers2 = [];

  // loop through the earthquakes/'features' array creating a new circle marker with detailed popup for each quake and pushing them into the quakeMarkers array
  for (let i = 0; i < Object.keys(city_data).length; i++) {

    let city = city_data[i];
    console.log('city object = ' + city);

    let name = city.city;
    console.log('city name = ' + name);

    let coordinates = [city.latitude,city.longitude];
    console.log('coordinates = ' + coordinates);

    let population = Number(city.population);
    let population_string = population.toLocaleString();
    console.log('population = ' + population)
    console.log('population_string = ' + population_string);
```

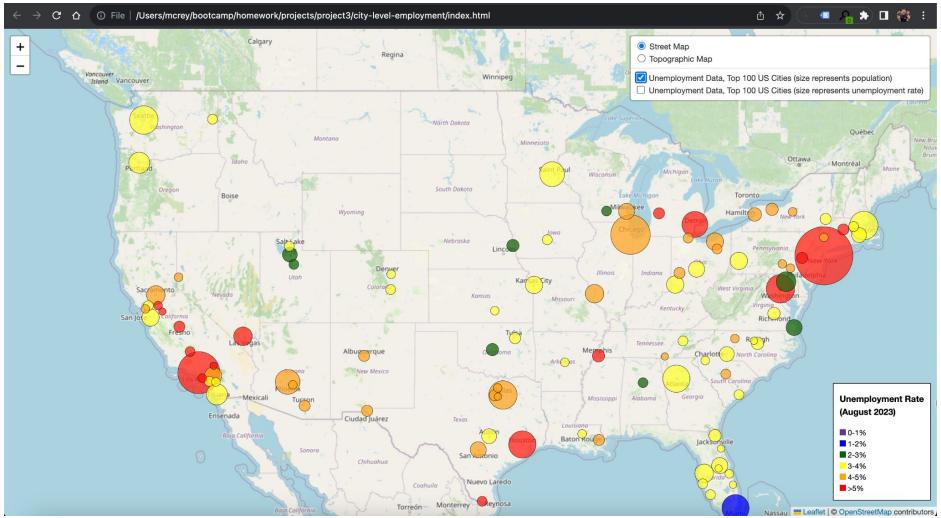
```javascript

    // create circle marker based on coordinates + attributes dict, with detailed popup showing additional info
    cityMarkers.push(
      L.circle(coordinates, featureAttributes)
      .bindPopup(`<h3>${name}</h3><hr>
      Unemployment Rate (August 2023): ${unemploymentRate}%<br>
      Unemployment Count (August 2023): ${unemploymentCount_string}`)
    );

    // save attributes dict/JSON for circle marker
    let featureAttributes2 = {
      fillOpacity: 0.75,
      fillColor: colorBasedOnRate,
      radius: Math.pow(unemploymentRate,2.5)*1000,
      stroke: true,
      color: "black",
      weight: 0.5
    };

    // create circle marker based on coordinates + attributes dict, with detailed popup showing additional info
    cityMarkers2.push(
      L.circle(coordinates, featureAttributes2)
      .bindPopup(`<h3>${name}</h3><hr>
      Unemployment Rate (August 2023): ${unemploymentRate}%<br>
      Unemployment Count (August 2023): ${unemploymentCount_string}`)
    );

  }


  // create overlay layer from quakeMarkers array
  let cityLayer = L.layerGroup(cityMarkers);
  let cityLayer2 = L.layerGroup(cityMarkers2);
```

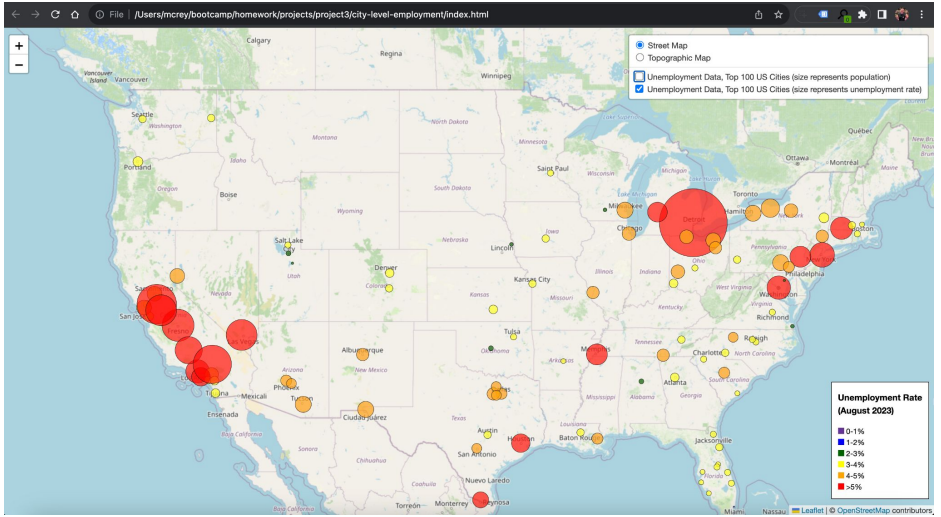
```javascript
  // create overlayMaps object to hold our quakeLayer of earthquake markers with popups
  let overlayMaps = {
    "Unemployment Data, Top 100 US Cities (size represents population)": cityLayer,
    "Unemployment Data, Top 100 US Cities (size represents unemployment rate)": cityLayer2,
  };


  // create map, passing streetmap and earthquakes layers to display on load, showing the entire world centered on [0,0]
  let myMap = L.map("map", {
    center: [39.8355, -99.0909],
    zoom: 4,
    layers: [street, cityLayer]
  });


  // add a legend showing how the marker colors represent earthquake depth

  var legend = L.control({
    position: 'bottomright'
  });

  legend.onAdd = function(map) {
    var div = L.DomUtil.create('div', 'unemployment legend');
    div.style.backgroundColor = "white";
    div.style.border = "thin solid black";
    div.style.padding = "0px 10px 10px";
    div.innerHTML = '<h3>Unemployment Rate<br>(August 2023)</h3><svg width="10" height="10"><rect width="10" height="10" fill="#663399"/></svg>  0-1%<br><svg wi
    return div;
  };

  legend.addTo(myMap);


  // create layer control with baseMaps and overlayMaps objects and add it to the map
  L.control.layers(baseMaps, overlayMaps, {
    collapsed: false
  }).addTo(myMap);
```

Final Product

Interactive Map

Upon loading the HTML, JavaScript calls the Flask API, processes the JSON data on cities, creates 2 sets of Leaflet circle markers the visualize the data in different ways, and builds the Leaflet map with layers





Conclusions and Next Steps

Analysis/Takeaways/Value

- This map enables users to easily go beyond average national unemployment data and view how that data varies at the city level for the 100 largest US cities (by population), including potential correlation in unemployment statistics within sub-regions of the country
    - The first map overlay/view seems to indicate some correlation between city size/population and higher unemployment rates, as well as generally lower unemployment rates in the Southwest, especially Florida, and in the Northwest / Mountain West
    - The second map overlay/view makes it much more clear to see that the highest unemployment rates are quite concentrated within a few smaller regions of the country, specifically Detroit, MI then California / Las Vegas, then the Northeast

Next Steps / Future Analysis

- Given additional time it would be valuable to add additional pages/options to help users explore more unemployment rate data
  - Analyzing unemployment data over longer periods of time for individual cities
  - Pulling similar data for larger areas/regions to allow users to view the data at different levels of aggregation
  - Running statistical analysis on various datasets to better describe and understand how current rates and/or rates for individual cities compare
- It would also be great to pull in additional datasets to allow users to view unemployment rate statistics within the context of other geographical data
  - GDP per capita, credit card debt per capita, median home prices, demographic data, job type data, etc

Appendix

Acknowledgements and Data Sources

- SimpleMaps used for database of US cities with populations and lat+long coordinates: https://simplemaps.com/data/us-cities
- help with SQLite database creation via Python/Pandas pulled from https://www.fullstackpython.com/blog/export-pandas-dataframes-sqlite-sqlalchemy.html
- help with Flask+Python+HTML+JS for creating web applications pulled from https://www.geeksforgeeks.org/pass-javascript-variables-to-python-in-flask/
- reverse geocoding for looking up state names from lat/long coordinates via https://geocode.maps.co/ API