

Group: Alisa Rubinstein, Emily Vanek, Lea Srebrenik
Github Repo: <https://github.com/alisasara/206-FINAL-PROJECT.git>

The goals for your project (10 points)

The goals that were achieved (10 points)

The problems that you faced (10 points)

Amongst the many goals we had for this project, the main one was to create a working travel guide based on weather, flights, and hotels. However, after beginning our project, we ran into some issues with our selected APIs and decided it would be best to shift from our original goal. Problems we ran into included not being able to access the tripadvisor API and another API only providing limited data. Our current project still serves as a travel guide but with different overarching criteria: weather, events, and restaurants instead of flights and hotels. We used two APIs, one for weather and one for events, and for restaurants, we used Beautiful Soup to web scrape the Yelp Top 100 US Restaurants page. The travel guide works by asking the user to input their "ideal temperature," and then it proceeds to find a list of cities that best matches the imputed temperature. From there, the code provides the events and top rated restaurants in the state of the city. If there are multiple top rated restaurants in a state, the code also provides the one with the highest ranking. In terms of the individual work we each did, Alisa worked on the event API, event calculations, event textfile, and event visualization. Emily worked on the weather API, weather calculations, weather textfile and weather visualization. Lea worked on the restaurant calculations, visualizations, and textfile using Beautiful Soup. We shared one GitHub repository that contains our own python files titled by our names, the weather, event, and restaurants textfiles, the travel database where we based our calculations from, and our main function python file that connects our work together by their individual main functions.

Your file that contains the calculations from the data in the database (10 points)

The calculations that each of us conducted from the data in the database are located within our own self titled python files. And that code is then called all together in the main function and file.

- The events calculations function is called "events_calculations(cur, conn)" and it finds the state with the most events.
 - The function creates a dictionary from the list of events in the main seatgeek table with the keys being the state and the value being the number of events in each state. Then the function sorts the dictionary in ascending order so that the first item in the dictionary is the state with the most events. To find the state associated with the id, the function uses the SELECT statement to grab the state associated with the state_id.
- The weather calculations function is called "weather_calculations(cur, conn)" and it finds the cities and temperatures that match a user's input ideal temperature and returns both in separate lists.
- The restaurants calculations is split into two parts, one function, "get_restaurant_count_state(state, cur, conn)" that takes a user's imputed state and returns the amount of top rated restaurants in that state using SELECT COUNT. The

second part, “highest_rank(state,cur,conn)” calls upon the first function to get the count of top rated restaurants in a state, and then returns the restaurant with the highest ranking in the state.

Screenshots of write-up files:

33 lines (33 sloc) 1.11 KB	100 lines (100 sloc) 4.98 KB
1 There are 17 events happening in NY	1 There are 5 Yelp top rated restaurants in Arizona
2 There are 7 events happening in NC	2 There are 4 Yelp top rated restaurants in Virginia
3 There are 7 events happening in FL	3 There are 4 Yelp top rated restaurants in California
4 There are 7 events happening in OH	4 There are 2 Yelp top rated restaurants in Illinois
5 There are 4 events happening in CT	5 There are 3 Yelp top rated restaurants in Texas
6 There are 4 events happening in MA	6 There are 2 Yelp top rated restaurants in South Carolina
7 There are 3 events happening in CA	7 There are 2 Yelp top rated restaurants in Idaho
8 There are 3 events happening in ON	8 There are 3 Yelp top rated restaurants in Nevada
9 There are 3 events happening in TN	9 There are 4 Yelp top rated restaurants in Tennessee
10 There are 3 events happening in GA	10 There are 5 Yelp top rated restaurants in Oregon
11 There are 3 events happening in ME	11 There are 4 Yelp top rated restaurants in California
12 There are 3 events happening in WI	12 There are 3 Yelp top rated restaurants in Nevada
13 There are 3 events happening in NJ	13 There are 2 Yelp top rated restaurants in New York
14 There are 3 events happening in MI	14 There are 4 Yelp top rated restaurants in Tennessee
15 There are 3 events happening in PA	15 There is 1 Yelp top rated restaurant in Alaska
16 There are 3 events happening in KY	16 There are 3 Yelp top rated restaurants in Kentucky
17 There are 2 events happening in OK	17 There are 4 Yelp top rated restaurants in Oklahoma
18 There are 2 events happening in NV	18 There are 2 Yelp top rated restaurants in North Carolina
19 There are 2 events happening in TX	19 There are 4 Yelp top rated restaurants in Washington
20 There are 2 events happening in MD	20 There are 5 Yelp top rated restaurants in Pennsylvania
21 There are 2 events happening in DC	21 There are 2 Yelp top rated restaurants in New York
22 There are 2 events happening in VA	22 There are 2 Yelp top rated restaurants in South Carolina
--	23 There are 5 Yelp top rated restaurants in Arizona

5 lines (5 sloc) 258 Bytes
1 Below are the cities that best match your ideal temperature of 66 degrees:
2 Phoenix: The temperature in Phoenix is 66.88.
3 Austin: The temperature in Austin is 65.48.
4 El Paso: The temperature in El Paso is 65.3.
5 Glendale: The temperature in Glendale is 66.54.

The visualization that you created (i.e. screen shot or image file) (10 points)

We created visualizations for each of our files and individual parts using matplotlib. You can see screenshots of them below.

For accurate calculations, have all 100 rows loaded into the database tables.

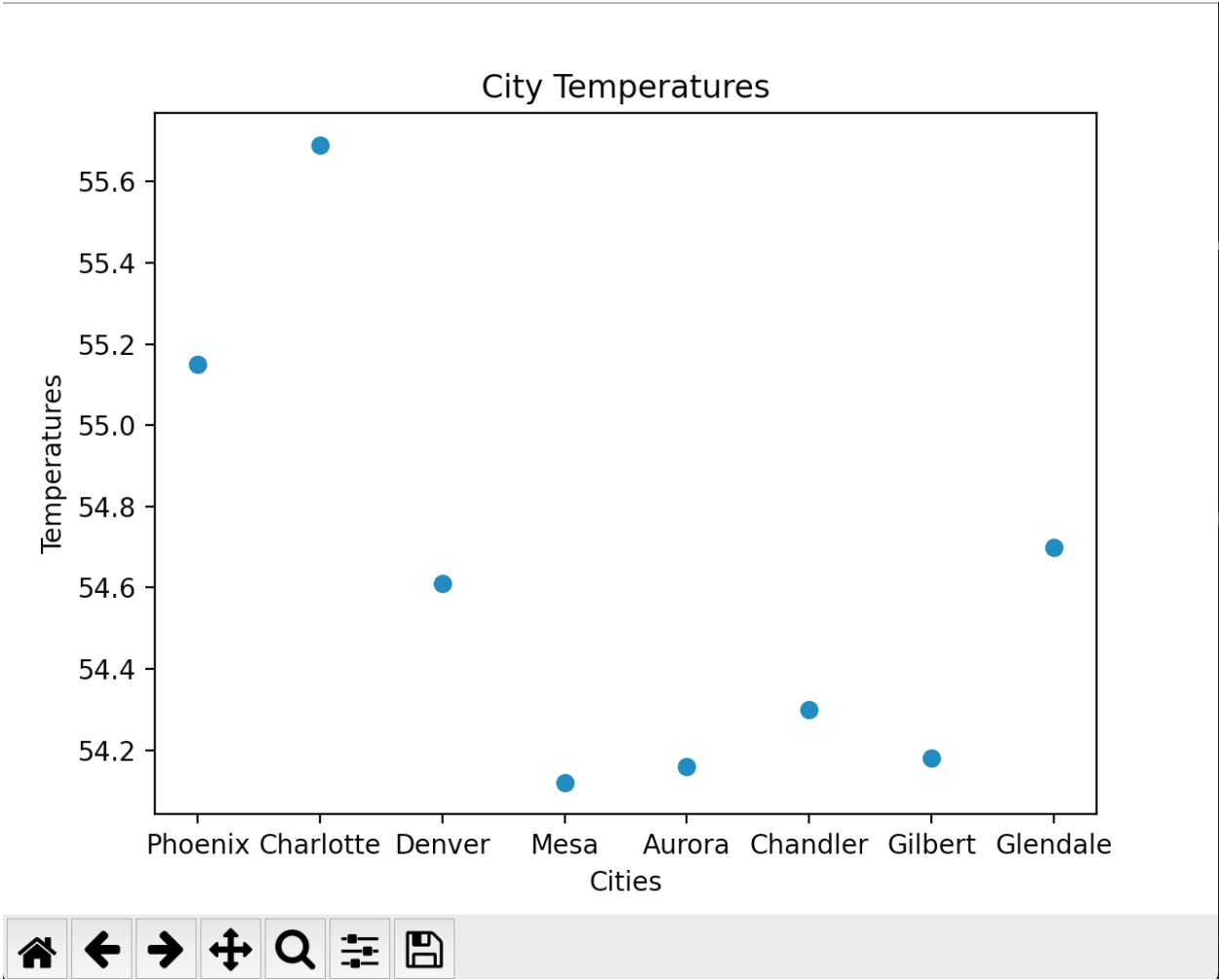


Figure 1

Figure 2

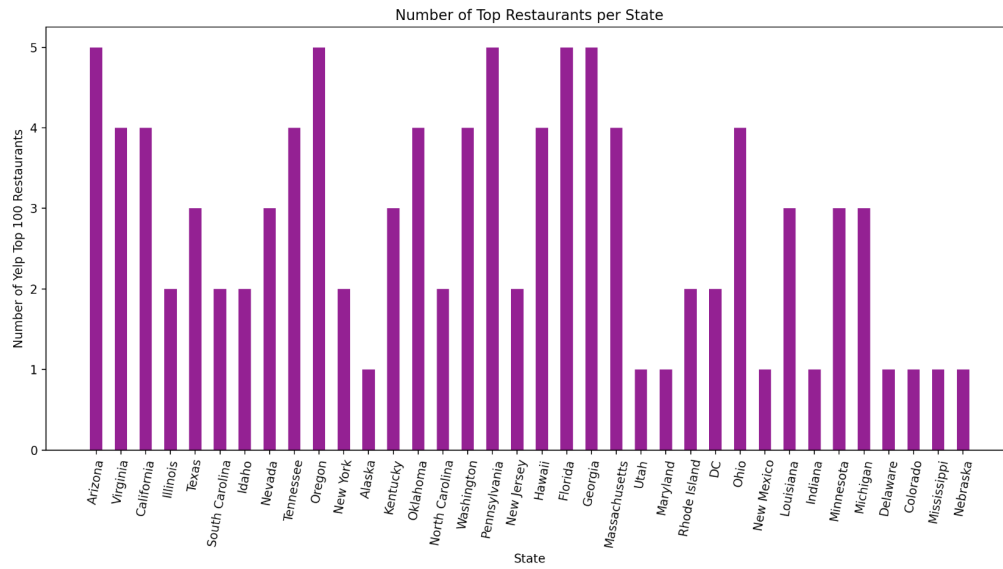
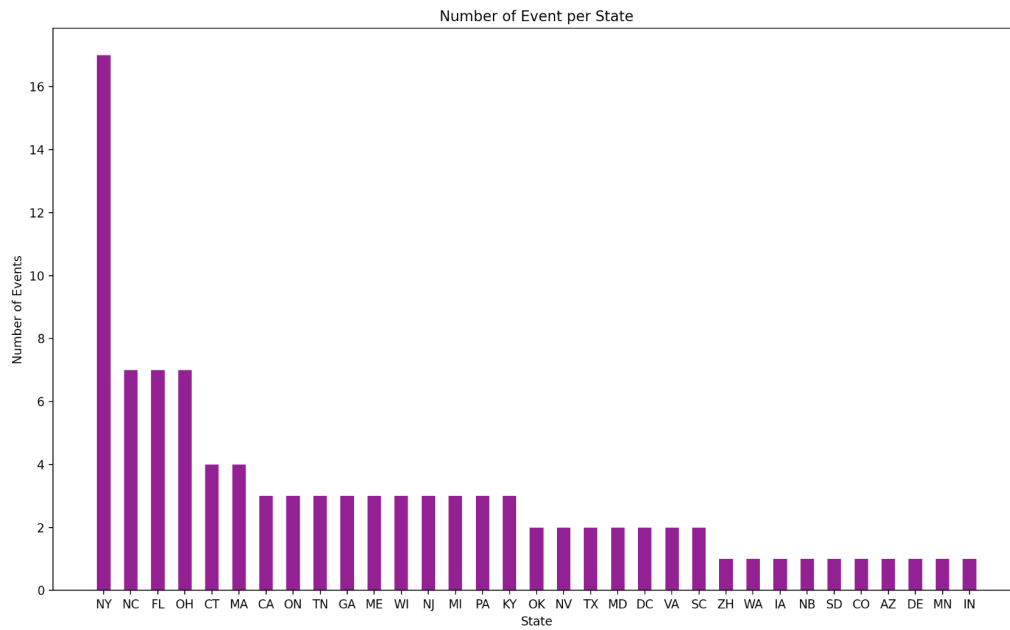
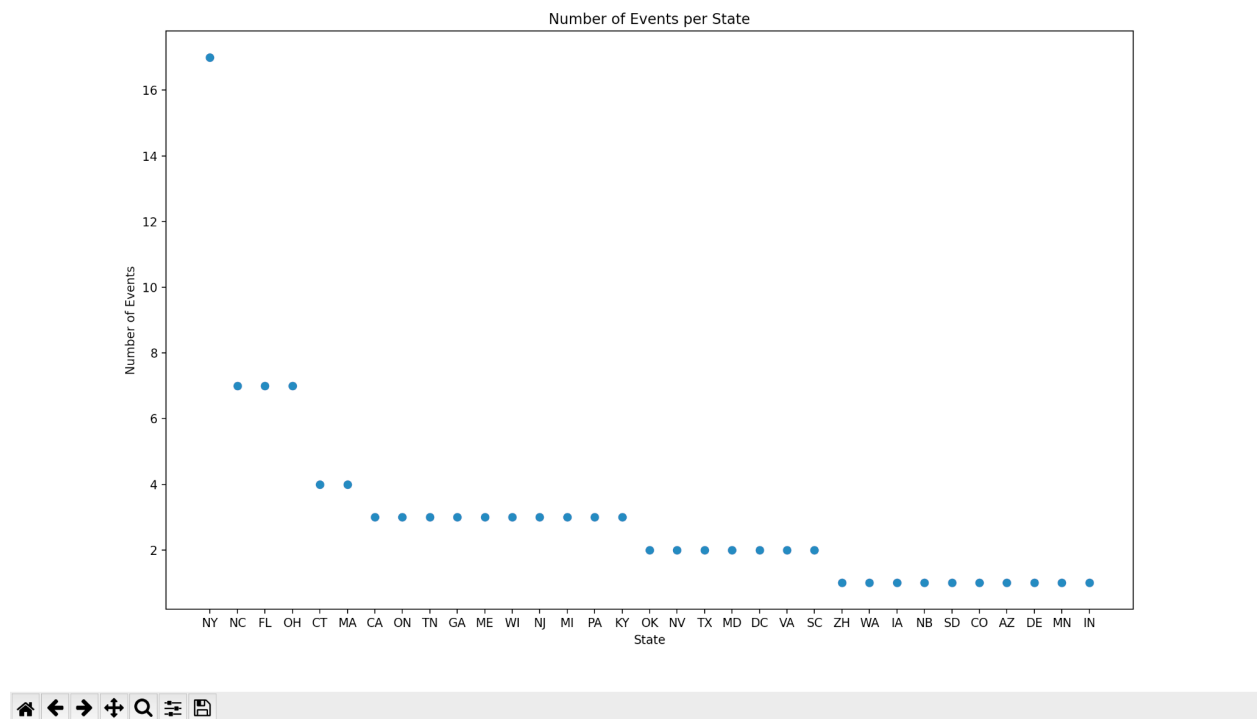


Figure 1

Figure 2





Instructions for running your code (10 points)

Main function: When first opening up the code, the main function must be run a total of four times in order to load all 100 items into the database. The first initial four runs do not require the entire process of the terminal to be completed and once the first input statement appears, the terminal can be trashed. When running the code for the 5th time, the user must then answer the first input question which asks for their most ideal temperature. The code will return a list of the temperatures in the database which fall within one degree of the imputed temperature and the corresponding cities. If no temperatures are found, the user will be prompted with the same question until the code can properly run. Next, a scatter plot will appear with the cities and corresponding temperatures. Additionally, a text file, called weather.txt will be created listing the temperatures and cities that best match the imputed ideal temperature. Next the code will print the state with the most number of events and generate a bar graph and scatter plot with all the states in the specific table and the number of events in the state. At the same time, a text file, called events.txt, will be generated that lists the number of events in each state. Lastly, an input statement will appear, asking the user to input a state to find out the Yelp top rated restaurants in that state according to the Yelp Top 100 List of 2022. The code will print the number of restaurants in that state. It will also print the name of the restaurant with the highest ranking in the state. The input statement is case sensitive and requires the state to be typed beginning with a capital letter. At the same time, a bar graph will be generated with the total number of top restaurants per state in the given table as well as a text file, called restaurants.txt that lists the number of top restaurants in each state.

Documentation for each function that you wrote. This includes the input and output for each function (20 points)

emily.py:

```
def main():
    cur, conn = setUpDatabase('TRAVEL_db')
    create_weather_table(cur, conn)
    create_weather_info(cur, conn)
    weather_calculations(cur, conn)
```

alisa.py

```
def main():
    cur, conn = setUpDatabase('TRAVEL_db')
    create_events_table(cur, conn)
    create_cities_table(cur, conn)
    create_types_table(cur, conn)
    create_states_table(cur, conn)
    create_venues_table(cur, conn)
    events_calculations(cur, conn)
    add_events_from_json(cur, conn)
    seatgeek_visualization_and_txt(cur, conn)
```

Lea.py

```
def main():
    url = 'https://www.yelp.com/article/yelps-top-100-us-restaurants-2022'
    data = get_website_data(url)

    cur, conn = setUpDatabase('TRAVEL_db')
    create_restaurants_table(cur, conn)
    create_restaurant_states_table(cur, conn)
    create_city_table(cur, conn)
    create_cuisine_table(cur, conn)
    set_up_table(data, cur, conn)

    state = input("Enter a state to view top rated restaurants: ")
    get_restaurant_count_state(state, cur, conn)
    - Output: count of restaurants per state or None

    highest_rank(state, cur, conn)
    - Output: tuple with (name of restaurant, ranking) or None

    restaurants_visualization_and_csv(cur, conn)
```

You must also clearly document all resources you used. The documentation should be of the following form (20 points)

- **Date:** 11/2-11/6
 - **Issue description:** Issue finding free api that also generated 100 pieces of data
 - **Location of resource:** Lecture office hours session
 - **Result:** GSI instructors assisted in helping me find a new events API after struggling to find a flights api which did not work, leading me to find a new idea. In class I not only found a new api, but I also better understood how to read the API documentation to have it make at least 100 requests at a time
-
- **Date:** 11/7
 - **Issue description:** Limiting the code to input 25 items into the database as a time
 - **Location of resource:** Online office hours
 - **Result:** I gained a better understanding of the process needed to code this section of the data and was able set up a new variable, ids, which allowed me to proceed with the functions needed to limit the data to 25 items at a time,
-
- **Date:** 11/8
 - **Issue description:** Calculations and finding the corresponding state to the given ID found in the dictionary created in the events python file
 - **Location of resource:** UMSI peer tutoring
 - **Result:** Assistance and guidance in using SQL SELECT statements to grab data from a table where the given id was equal to the answer to my calculations