Andrew Huang (anhuang)
Trianna Nguyen (triannan)
SI 206 Final Project Report
April 19, 2023
Github Repository: https://github.com/AndeluPhan/ClimateSimilaritySI206

Initial Goals
The goals of this project were to collect info from the Open-Meteo API and the U.S. Cities Wiki and to calculate the similarity in climate between the inputted query city and the other cities collected. From the U.S. Cities Wiki, we want to use Beautiful Soup to collect the city name, state, population, latitude, and longitude for 100 different cities. From the Open-Meteo API, we want to collect the max temperature, minimum temperature, precipitation sum, wind speeds, and solar radiation sums from the 100 cities.

Goals Achieved
We worked with the U.S. Cities Wiki and used Beautiful Soup to collect the city name, state, population, latitude, and longitude for 100 different cities. After we gathered this information, we used the Open-Meteo API to collect the max temperature, minimum temperature, precipitation sum, wind speeds, and solar radiation sums from the 100 cities for each month over the past 5 years. From the data collected we were able to run our algorithm which calculated the similarity between one city and any other city in terms of climate.

Problems Faced
One problem we faced was finding a reliable, and affordable API that fit our project requirements. For the climate API, we could have gone with Meteostat, however their rate limit of 3 requests per second and 500 requests/month put a ceiling on how much we could experiment and gave us a small margin of error. For our collection of U.S Cities, we initially struggled finding a source of data that provided city, state, population, latitude and longitude and that list of cities should be ranked by population.

Calculations
Calculation of the climate cosine similarities between the city input and the other 99 cities. Higher cosine similarity is better (closer to 1, means better similarity to inputted query city).
Top 10 cities most similar in climate to Miami:

```
Climate cosine similarities between Miami and 99 other cities
St. Petersburg, Florida: 0.9995778885923302
Tampa, Florida: 0.9988667789549331
Orlando, Florida: 0.9984713588237715
Honolulu, Hawaii: 0.9983725918812786
Corpus Christi, Texas: 0.998340977505736
New Orleans, Louisiana: 0.9983163910834004
Jacksonville, Florida: 0.9980349723431866
Houston, Texas: 0.9969979766266618
San Antonio, Texas: 0.9958610889871632
Laredo, Texas: 0.9958609084110238
```

…. More Cities (See all 100 cities ranked in calculated_data.txt)
Bottom 14 cities (These cities are the least similar in climate from Miami)
From our results, Anchorage Alaska is the least similar in climate from Miami, and St. Petersburg, Florida is the the most similar.

```
87    Buffalo, New York: 0.9613409560927656
88    Detroit, Michigan: 0.9574444072493513
89    Toledo, Ohio: 0.9567523887015791
90    Chicago, Illinois: 0.9537107257116402
91    Lincoln, Nebraska: 0.9461214604307551
92    Aurora, Colorado: 0.9453072616546546
93    Omaha, Nebraska: 0.9410086821990036
94    Denver, Colorado: 0.9384690489032725
95    Madison, Wisconsin: 0.9346056404593973
96    Milwaukee, Wisconsin: 0.9340533755100151
97    Colorado Springs, Colorado: 0.9188461746924381
98    Saint Paul, Minnesota: 0.9064070684319944
99    Minneapolis, Minnesota: 0.9015524934374101
100   Anchorage, Alaska: 0.8210331589369426
101
102   City with the most similar climate to Miami: St. Petersburg, Florida
103   City with the least similar climate to Miami: Anchorage, Alaska
```

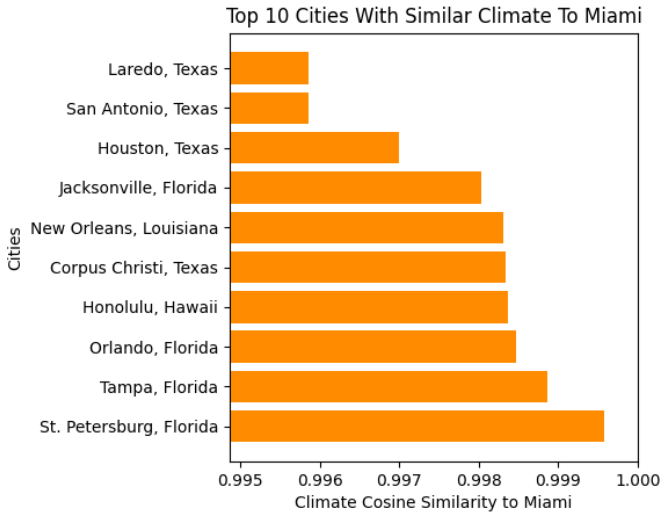
The top 10 cities calculated and their respective populations.

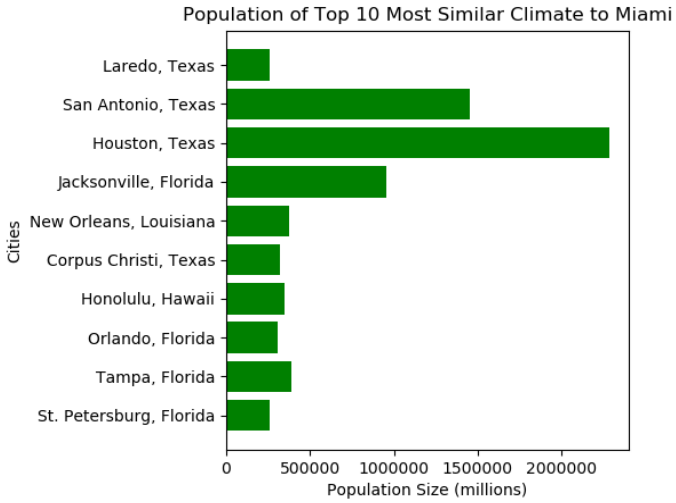
```
105   Top Ten Cities with the Most Similar Climate and their Population:
106   St. Petersburg, Florida: 258201
107   Tampa, Florida: 387050
108   Orlando, Florida: 309154
109   Honolulu, Hawaii: 345510
110   Corpus Christi, Texas: 317773
111   New Orleans, Louisiana: 376971
112   Jacksonville, Florida: 954614
113   Houston, Texas: 2288250
114   San Antonio, Texas: 1451853
115   Laredo, Texas: 256153
```

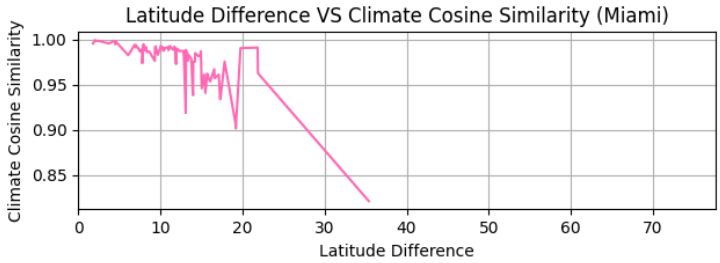
Calculation of the latitude and longitude differences between the city input and the other 99 cities. (See more in calculated_data.txt)

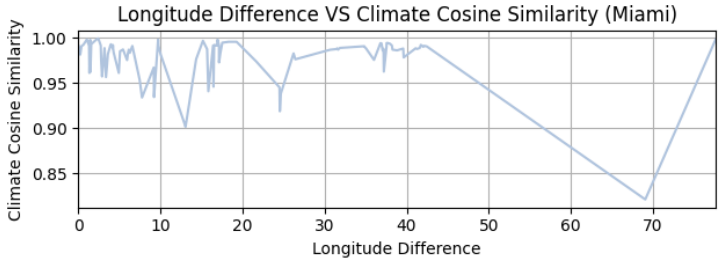
```
Cities with the closest latitude to Miami (Closest - Furthest):
('Laredo', 'Texas'), ('Corpus Christi', 'Texas'), ('St. Petersburg', 'Florida'), ('Tampa', 'Florida'),

Cities with the closest longitude to Miami (Closest - Furthest):
('Winston-Salem', 'North Carolina'), ('Pittsburgh', 'Pennsylvania'), ('Greensboro', 'North Carolina'),
```

Visualizations
Input example: Miami









Instructions for Running Code

1. Run fetchData.py 8 times to populate the database
2. Run CityClimates.py and enter in a city name when prompted in the terminal (ex. Los Angeles, Miami, Baltimore, Denver, Houston, etc.).
   * Use correct spelling for input city, look to calculated_data.txt for reference if needed.

Function Documentation

fetchData.py

- getCities
  - The inputs to this function are conn and cur. Conn is the connection to the database, and cur is the cursor of the connection. This function uses BeautifulSoup to scrape the U.S. Cities Wiki. It also creates the population table if it doesn't already exist, and inserts the data scraped into the table. The return value of this function is None.

- getClimateData
  - The inputs to this function are conn and cur. Conn is the connection to the database, and cur is the cursor of the connection. This function uses an API to get daily weather data for a city across a timespan of 5 years. The function then creates monthly averages for the city from the daily weather data. It also creates the climateData table if it doesn't already exist and inserts the monthly weather data into the table. The return value of this function is None.
- main
  - There are no inputs to this function. This function creates the connection to the database and creates a cursor to the connection. This function also runs the getCities and getClimateData functions to populate the database appropriately. getClimateData will run after there are 100 rows in the population table. If there are 100 rows in the population table, fetchData will need to be run 4 more times, for a total of 8 times to get 100 rows in both population and climateData table. This function returns None.

CityClimates.py

- calculateRankings
    - The inputs to this function are cur and chosen_city. Cur is the cursor to the connection of the database, and chosen_city is a city input when prompted from the main function. This function will select data from the population and climateData tables in the database. It will then calculate the climate cosine similarities between chosen_city and the other 99 cities in the database. This function will then write the data calculated to a separate file. It will finally plot the top 10 cities with the most similar climate to chosen_city, and it will also plot the population of the top 10 most similar climates to chosen_city. The return value of this function is a dictionary called cosine_similarities, which contains the city name as the key and the cosine similarity as the value.
- calculateSimilaritiesByLatLon
    - The inputs to this function are cur, city, and cosine_similarities. Cur is the cursor to the connection of the database, city is a city input when prompted from the main function, and cosine_similarities is the dictionary returned from calculateRankings. This function will select the latitude and longitude data from the population table and calculate the difference in latitude and longitude between the city variable and the other 99 cities in the database. It will then write the data calculated to a separate file. The function will also plot the latitude difference vs the climate similarity and the longitude difference vs the climate similarity. The return value of this function is None.
- main
    - There are no inputs to this function. This function will set up the connection to the database and the cursor to the connection. It will then get user input from the terminal to prompt the user to enter a city. After this, it will run calculateRankings and calculateSimilaritiesByLatLon. This function returns None.

Resources

Date	Issue Description	Location of Resource	Result (did it solve the issue?)
04/15/2023	Need a climate API that takes latitude and longitude as input parameters and returns historical weather data.	https://open-meteo.com/en/docs	Yes, open-meteo returned daily sums and averages for respective weather attributes.
04/15/2023	Needed a list of cities with their name, state, and population. Cities should be ranked by population.	https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population	Yes, Wikipedia provided an easy source for Beautiful Soup to get city population data.

04/14/2023	We needed an API to quickly and affordably return climate normals given latitude and longitude. The API should be quick and affordable.	https://dev.meteostat.net/guide.html	No. This API was rate-limited to 3 requests/second and only allowed 500 a month on the free tier.
04/14/2023	Needed an api where a given city returns their latitude, longitude and population.	https://api-ninjas.com/api/city	No. We decided that getting city data from a list of cities ranked by population was a better and more simple approach.
4/16/2023	We needed to find out how to calculate cosine similarity in python	https://www.geeksforgeeks.org/how-to-calculate-cosine-similarity-in-python/#	Yes. We were able to calculate cosine similarity using this resource.
4/14/2023	We needed a list of U.S Cities with their population.	https://worldpopulationreview.com/us-cities	No. We decided that it would be easier if the list of cities also had Latitude and Longitude, so we went with the U.S Cities Wiki.
4/16/2023	We wanted to find all the possible colors in matplotlib to change the colors of our plots	https://matplotlib.org/stable/gallery/color/named_colors.html	Yes. We were able to change the colors of our plots using this resource.