SI 206 Final Project Writeup

By: Abhinav Tadikonda and Jack Sambursky
Github: https://github.com/abhinavt20/206final.git

The goals for your project (10 points)

Originally, the goal for our project was to calculate the quantity of drinks that could be made based on the quantity of ingredients currently in-stock at Kroger. Additionally, we planned to calculate which drinks are the most popular in order to help inform Kroger what inventory they should be investing in. However, as we started to project, we decided it would be more interesting to determine if there was a link between a country's air quality index (AQI) and their mortality rate. We predicted that the continent/country with the highest average mortality rate should also have the highest average air quality index. Using data from the API, *Air Quality Open Data Platform*, and the website, *Statistic Times,* we planned to create visualizations and data tables to summarize our findings.

The goals that were achieved (10 points)

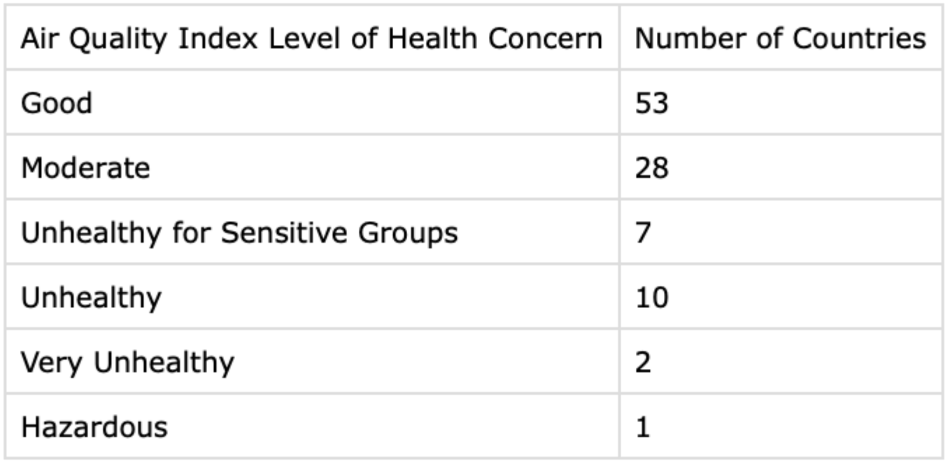
We determined that there is no significant relationship between the mortality rate and the air quality index across countries and continents alike. For example, we found that the continent with the highest average mortality rate (Europe) had the third lowest average air quality index. Also the continent with the lowest average mortality rate (South America) had the third highest average air quality index per continent. Additionally, after creating a scatter plot of the AQI vs mortality rate for each country in the data set, the line of best fit actually indicates a negative correlation between AQI and mortality rate; however, we suspect this is due to certain outliers in the data. Finally, we created an additional plot to map the number of countries that fall under certain API categories, and we found that a majority of countries fall in the "good" air quality index level with the second and third most populated categories being "moderate" and "unhealthy" respectively.

The problems that you faced (10 points)

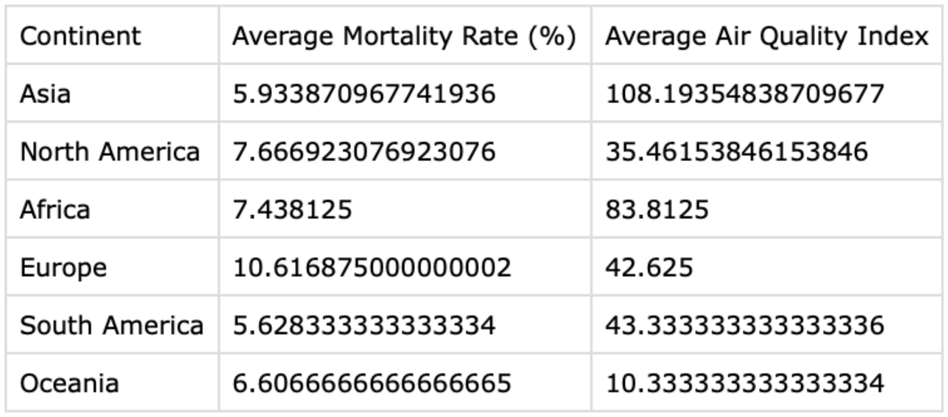
One of the biggest problems we faced was reading in the data from our API which held the air quality index for each country. In order to access the data for each country, the API required that we manually input the name of each country into the API link; however, they failed to provide a list of countries for us to index through. As such, we first read in data from the mortality rate website and stored the name of the countries in one array and the mortality rate in another. We then used the array of countries to loop through the API to gather data on the AQI for each country. When looking at our database tables it is important to note that the Country_ID column in the AirQualityTable and MortalityRateTable may look identical; however, the countries in the

AirQualityTable are not all the same as MortalityRateTable. For example, Indonesia is present in the MortalityRateTable but not in the AirQualityTable. Therefore, while it seems we may have just split one table into two, that is not the case in this instance.

File that contains the calculations from the data in the database:

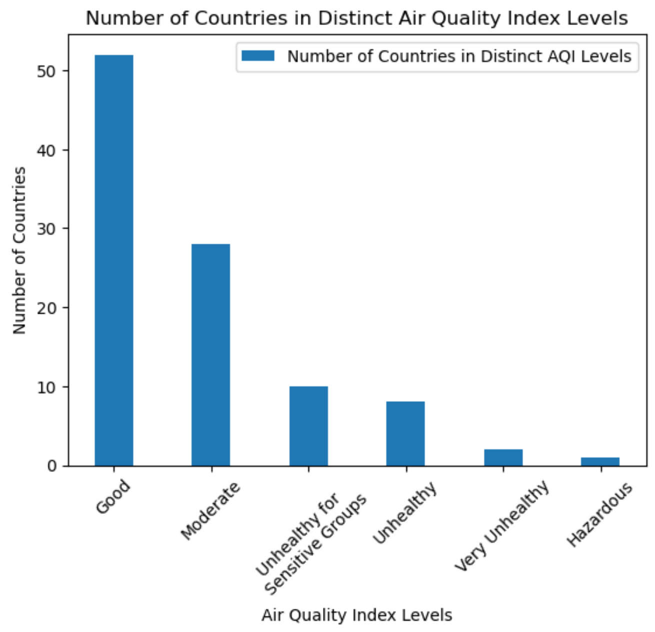


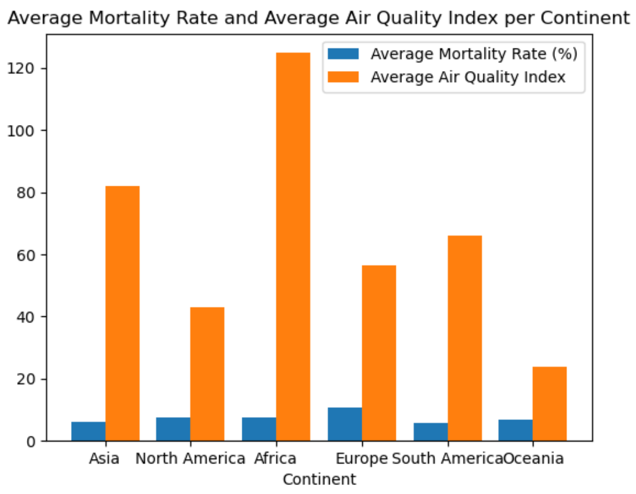
Air Quality Index Level of Health Concern	Number of Countries
Good	53
Moderate	28
Unhealthy for Sensitive Groups	7
Unhealthy	10
Very Unhealthy	2
Hazardous	1

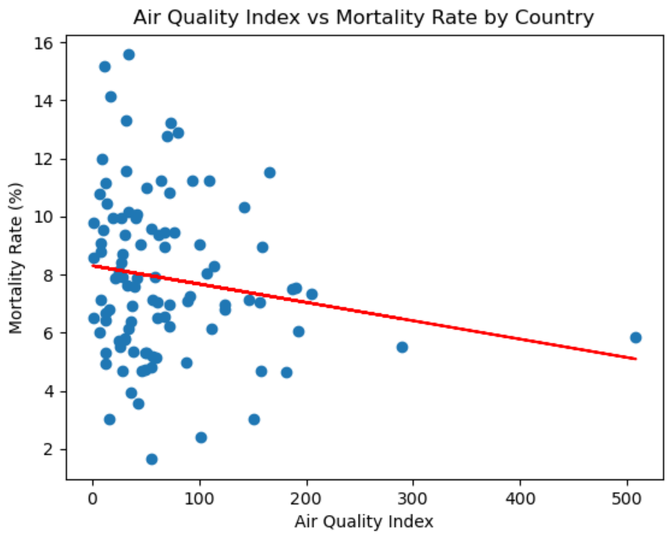


Continent	Average Mortality Rate (%)	Average Air Quality Index
Asia	5.933870967741936	108.19354838709677
North America	7.666923076923076	35.46153846153846
Africa	7.438125	83.8125
Europe	10.616875000000002	42.625
South America	5.628333333333334	43.333333333333336
Oceania	6.6066666666666665	10.333333333333334

Data Visualizations:



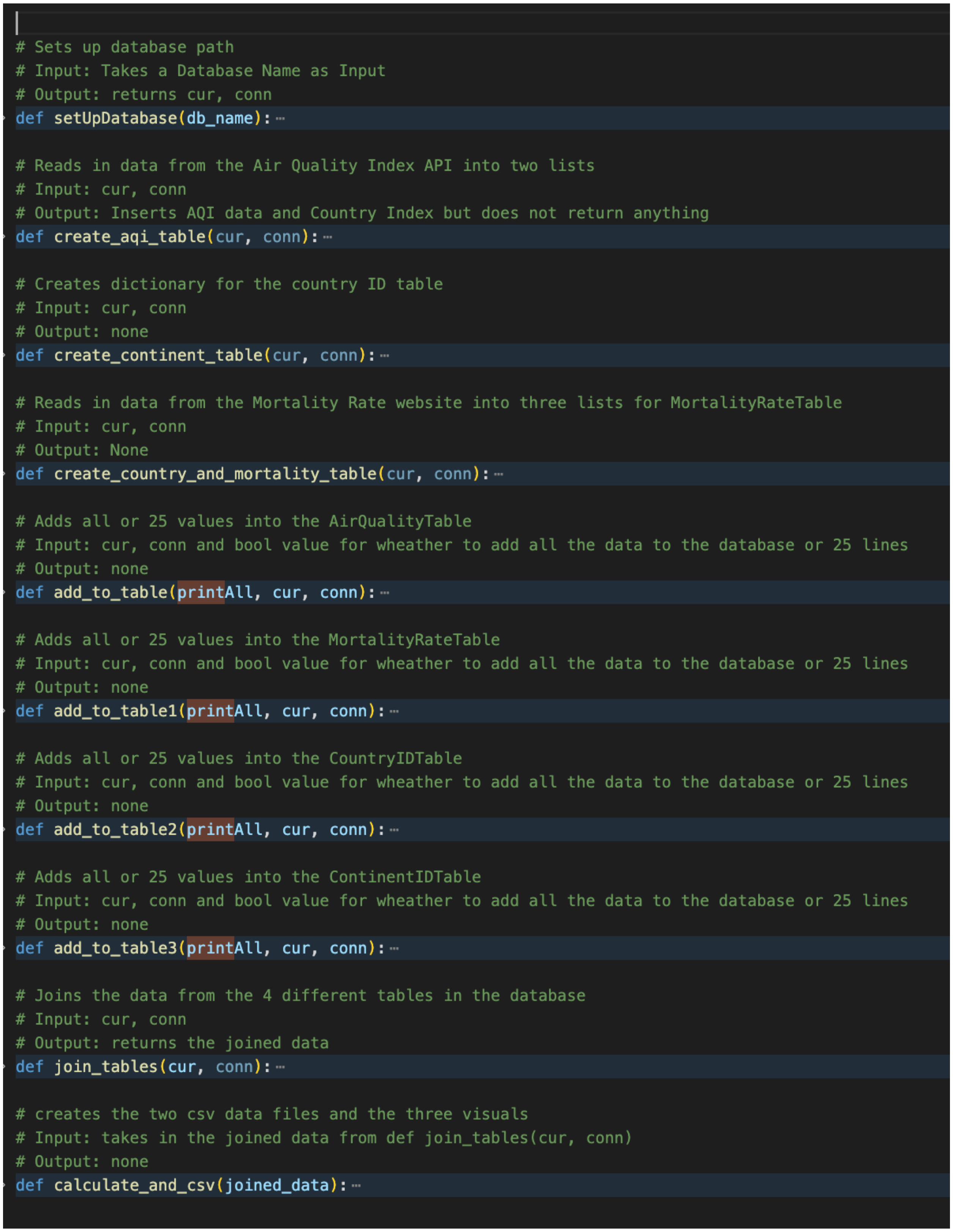




Instructions for how to run code

1. Download the api.py file
2. From terminal, cd into the directory in which you saved the api.py file
3. From terminal type in "python api.py"
4. When prompted, type "Reset" or "Continue"
    a. If you type "Reset," the data from the database will be emptied. You will then be prompted to choose if you want to add all available data to the database by typing "All" or only 25 lines of data by typing "25"
    b. If you type "continue" the program will add 25 more lines of data to the database until there is no more data left to add.
5. Repeat steps 3 and 4 until you have read in all the data

Documentation for each function including input and output



```
# Sets up database path
# Input: Takes a Database Name as Input
# Output: returns cur, conn
def setUpDatabase(db_name):⋯

# Reads in data from the Air Quality Index API into two lists
# Input: cur, conn
# Output: Inserts AQI data and Country Index but does not return anything
def create_aqi_table(cur, conn):⋯

# Creates dictionary for the country ID table
# Input: cur, conn
# Output: none
def create_continent_table(cur, conn):⋯

# Reads in data from the Mortality Rate website into three lists for MortalityRateTable
# Input: cur, conn
# Output: None
def create_country_and_mortality_table(cur, conn):⋯

# Adds all or 25 values into the AirQualityTable
# Input: cur, conn and bool value for wheather to add all the data to the database or 25 lines
# Output: none
def add_to_table(printAll, cur, conn):⋯

# Adds all or 25 values into the MortalityRateTable
# Input: cur, conn and bool value for wheather to add all the data to the database or 25 lines
# Output: none
def add_to_table1(printAll, cur, conn):⋯

# Adds all or 25 values into the CountryIDTable
# Input: cur, conn and bool value for wheather to add all the data to the database or 25 lines
# Output: none
def add_to_table2(printAll, cur, conn):⋯

# Adds all or 25 values into the ContinentIDTable
# Input: cur, conn and bool value for wheather to add all the data to the database or 25 lines
# Output: none
def add_to_table3(printAll, cur, conn):⋯

# Joins the data from the 4 different tables in the database
# Input: cur, conn
# Output: returns the joined data
def join_tables(cur, conn):⋯

# creates the two csv data files and the three visuals
# Input: takes in the joined data from def join_tables(cur, conn)
# Output: none
def calculate_and_csv(joined_data):⋯
```

All resources used

Date	Issue Description	Location of Resource	Result (did it solve the issue)
12/1/22	Needed Air Quality Index Data	https://aqicn.org/data-platform/token-confirm/3e528641-9f8e-4d20-adf6-67442e772e7d	Yes
12/1/22	Needed Mortality Rate Data	https://statisticstimes.com/demographics/countries-by-death-rate.php	Yes
12/5/22	Helped us create csv tables for our calculated data	https://www.pythontutorial.net/python-basics/python-write-csv-file/	Yes
12/6/22	Helped us understand the overall syntax for creating plots in Mathplotlib	https://matplotlib.org/stable/tutorials/introductory/pyplot.html	Yes
12/6/22	Trouble making a regression line in Mathplotlib	https://www.tutorialspoint.com/linear-regression-with-matplotlib-numpy	Yes
12/1/22	Trouble understanding how to only read in 25 lines of code to the database and general project help	Office Hours	Yes