206 Final Presentation
Tourism & Yelp

Sahithi, Max, Ashleigh

https://github.com/craigash00/206Final

Note: Calculation values, tables, and visualizations in the report might differ from those in the code file since the code was ran on different days and Yelp constantly updates data about restaurants based on new reviews

Original Goals

Main Plan: Find the potential effect of restaurant quality and average temperatures in a given country on the countries’ tourism numbers, using the Yelp API, a weather API, and World Bank website

Create at least 3 visualizations and 3 calculations

Achieved Goals

- Checked if there is a relationship between restaurant ratings and price and tourism in a country
- Successfully created a python application using real world data with no starter code
- Created 5 visualizations
- Carried out 5 different calculations
- Used Yelp API, World Bank population API, and World Bank tourism data website

Issues

- Not all countries use Yelp and instead of giving no response or error when making an API call with that country, Yelp will instead give results for a similar sounding location. So, prior to collecting data we first found out which countries use Yelp and found that out of the top 10 countries, 2 of them do not use Yelp. We then skipped these 2 countries and moved to the next 2 in the list that used Yelp.
- When searching for restaurants in a country, Yelp will only provide information for one city in that country and it will choose this city automatically. We used this information and treated it only as a subsection approximate of that country since it was not possible to get random restaurant data from all over the country.

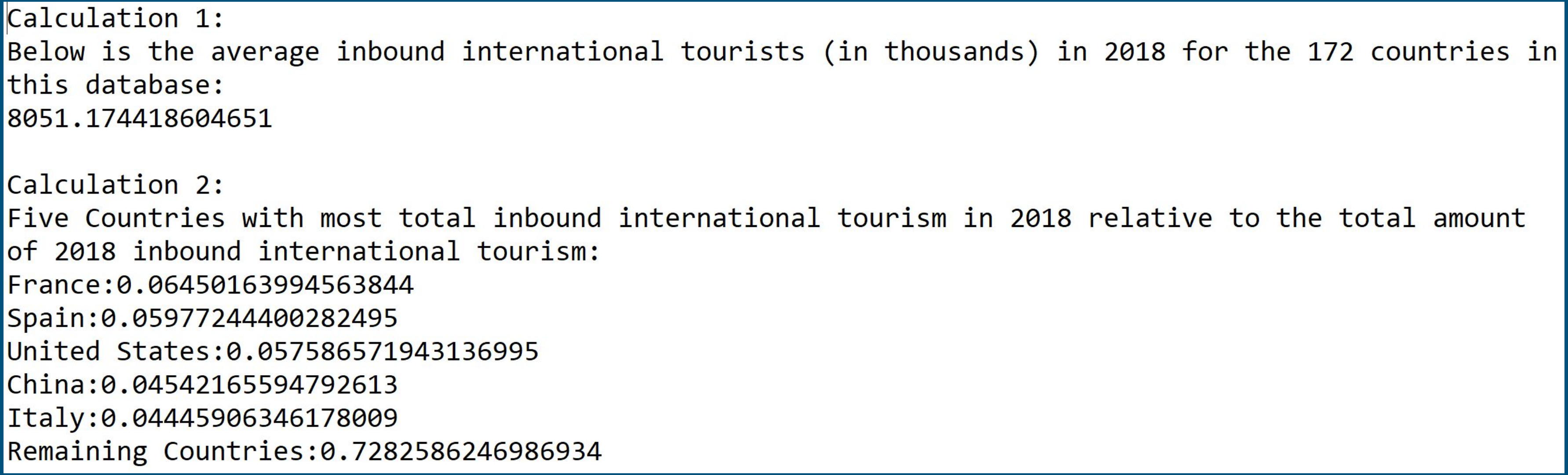
Issues

- The Yelp response object gives back a two letter country code whereas all other instances in our code use the country names. So, in order to make future joins and data selection easier, I added separate columns in the Yelp_Countries table with the country name as well as the country code
- Decided not to use Weather API because average annual weather for a large country such as the United States or China would not be very consistent.

Potential Flaws and Biases in the Data

- Since Yelp could only provide information for one city and this city was automatically chosen for us. Additionally, only 10 restaurants were chosen for each city. Therefore, these results are not entirely accurate and can’t necessarily be interpreted as representative of the whole country.
- The cities chosen by Yelp are also large metropolitan cities so price averages are not consistent with the rest of the country
- Lastly, restaurants are only listed on Yelp if they receive many reviews and if owners add their businesses to the website. So, this can often exclude smaller, local businesses.

Calculations- Tourism Website



```
Calculation 1:
Below is the average inbound international tourists (in thousands) in 2018 for the 172 countries in
this database:
8051.174418604651

Calculation 2:
Five Countries with most total inbound international tourism in 2018 relative to the total amount
of 2018 inbound international tourism:
France:0.06450163994563844
Spain:0.05977244400282495
United States:0.057586571943136995
China:0.04542165594792613
Italy:0.04445906346178009
Remaining Countries:0.7282586246986934
```

Calculations- Tourism Website

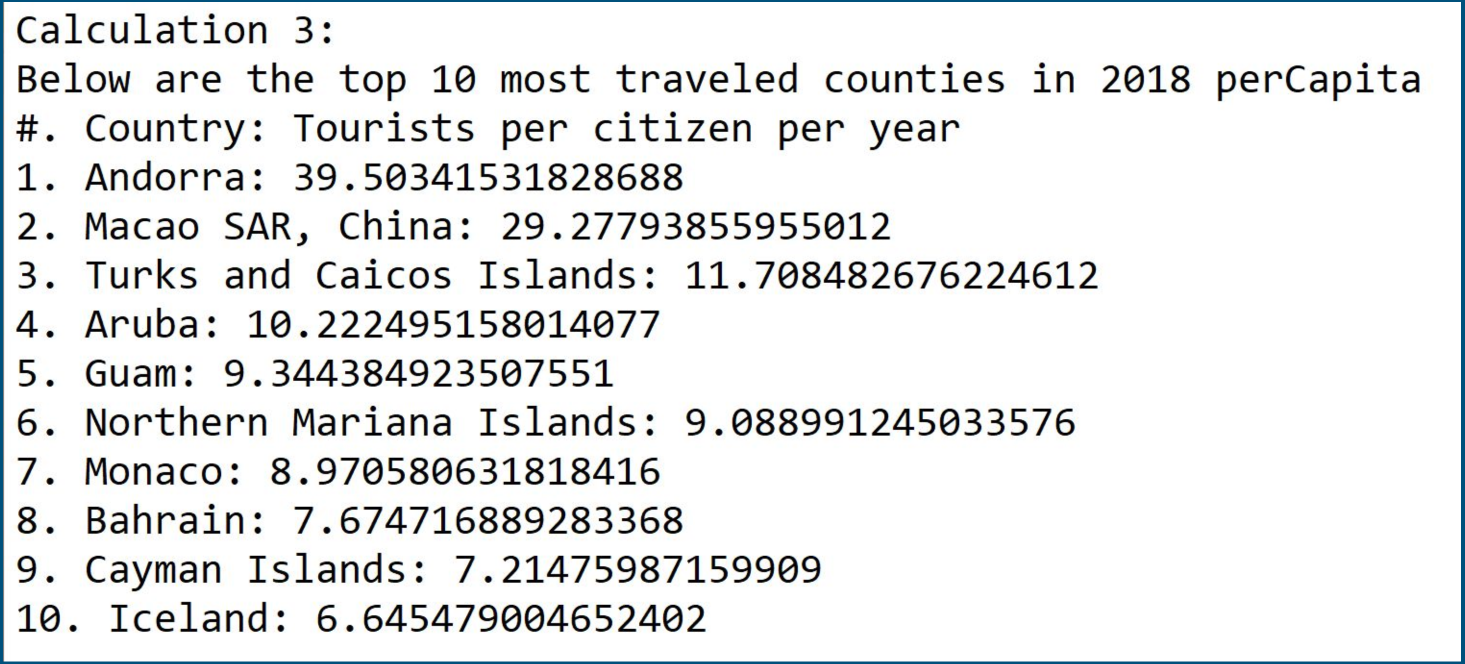
Calculation 1 Explanation:

- This calculation is created by the function calcAvgCountryTouristsinThousands(cur, conn) which returns the average number of inbound tourists in 2018 in thousands per country. The number 8051.174 means amongst 172 countries with tourism data available, the average number of people travelling to each country in 2018 was 8,051,174.

Calculation 2 Explanation:

- This calculation is created by the function calcBiggestFiveFractions(cur, conn) which returns a dictionary of length 6. This dictionary holds the 5 countries with the greatest number of inbound tourists in 2018 and the percentage of total tourism that year that is from each country. The 6th key value pair is for all the 167 remaining countries and their total percentage of total tourism. This calculation is used to create the pie chart in the visualizations section.

Calculations- Population API



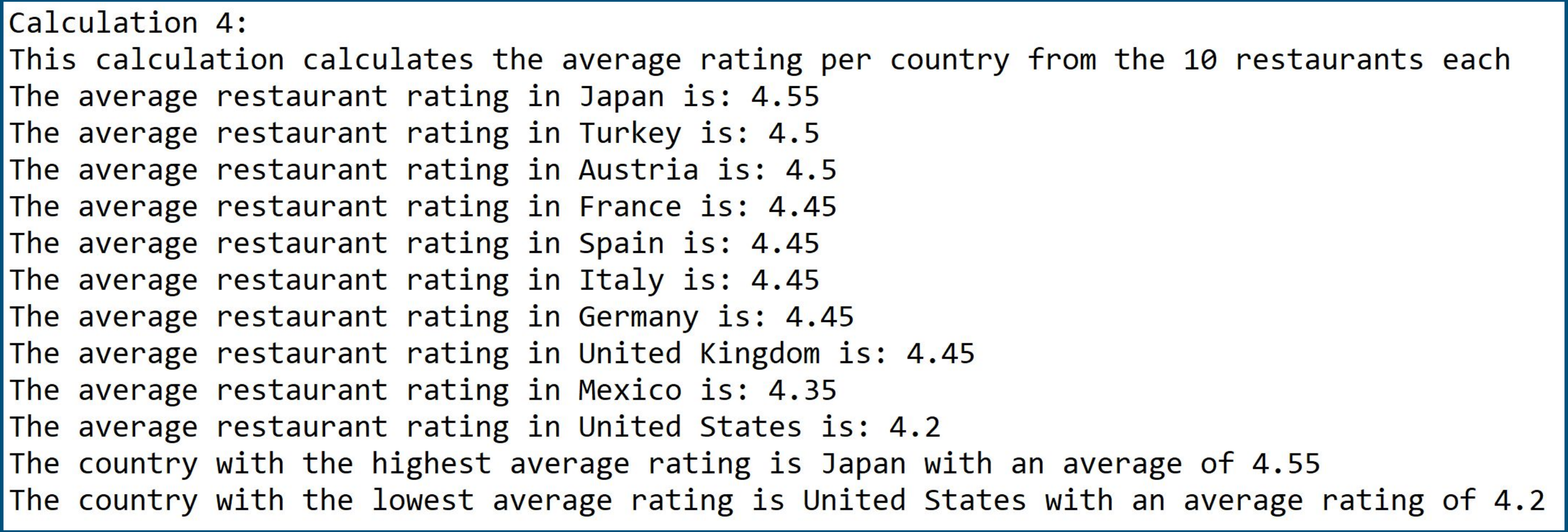
```
Calculation 3:
Below are the top 10 most traveled counties in 2018 perCapita
#. Country: Tourists per citizen per year
1. Andorra: 39.50341531828688
2. Macao SAR, China: 29.27793855955012
3. Turks and Caicos Islands: 11.708482676224612
4. Aruba: 10.222495158014077
5. Guam: 9.344384923507551
6. Northern Mariana Islands: 9.088991245033576
7. Monaco: 8.970580631818416
8. Bahrain: 7.674716889283368
9. Cayman Islands: 7.21475987159909
10. Iceland: 6.645479004652402
```

Calculations- Population API

Calculation 3 Explanation:

This calculation is created by using both the Population and Tourism databases in order to get a ‘Per Capita’ number. The numbers of Inbound Tourism and Population are selected for each country and divided to get the amount of travel per capita per country. The top ten most traveled cities by per capita numbers are then listed.

Calculations- Yelp API



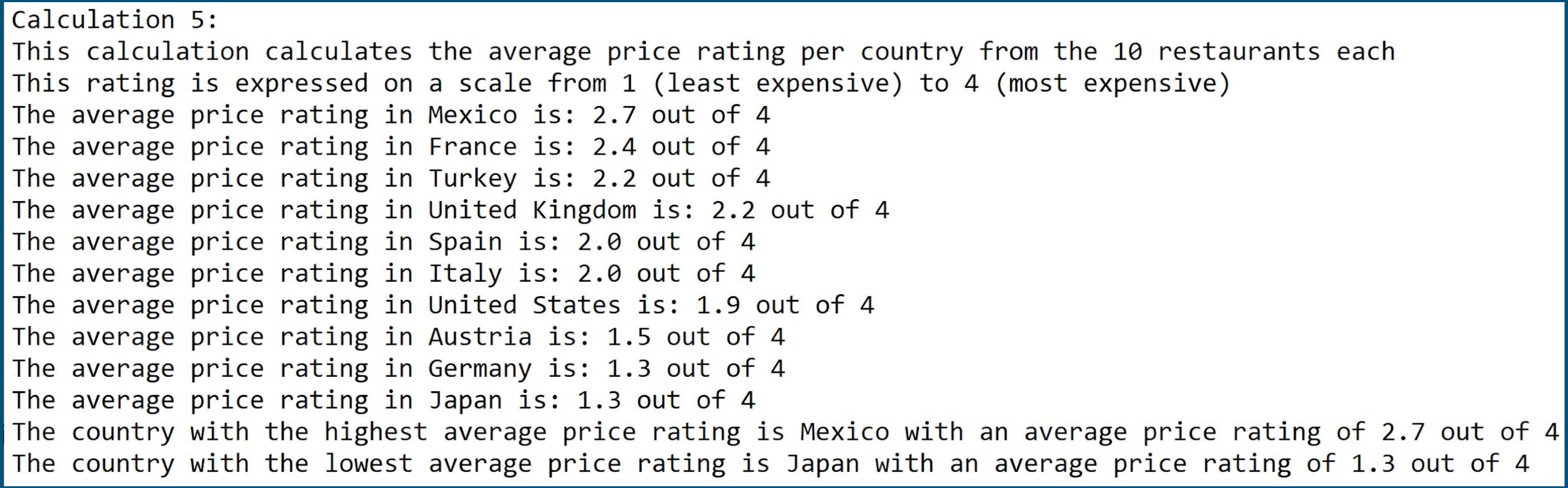
```
Calculation 4:
This calculation calculates the average rating per country from the 10 restaurants each
The average restaurant rating in Japan is: 4.55
The average restaurant rating in Turkey is: 4.5
The average restaurant rating in Austria is: 4.5
The average restaurant rating in France is: 4.45
The average restaurant rating in Spain is: 4.45
The average restaurant rating in Italy is: 4.45
The average restaurant rating in Germany is: 4.45
The average restaurant rating in United Kingdom is: 4.45
The average restaurant rating in Mexico is: 4.35
The average restaurant rating in United States is: 4.2
The country with the highest average rating is Japan with an average of 4.55
The country with the lowest average rating is United States with an average rating of 4.2
```

Calculations- Yelp API

Calculation 4 Explanation:

Using a join on the Restaurants and Yelp_Countries table, I selected the restaurant rating and country name and created a dictionary of the countries as keys and the values are the average restaurant rating for the 10 restaurants in that particular country. I did this by accumulating the rating value of each restaurant and then dividing by 10. I then sorted the dictionary based on average rating and wrote the information to a file as seen in the image above

Calculations- Yelp API



```
Calculation 5:
This calculation calculates the average price rating per country from the 10 restaurants each
This rating is expressed on a scale from 1 (least expensive) to 4 (most expensive)
The average price rating in Mexico is: 2.7 out of 4
The average price rating in France is: 2.4 out of 4
The average price rating in Turkey is: 2.2 out of 4
The average price rating in United Kingdom is: 2.2 out of 4
The average price rating in Spain is: 2.0 out of 4
The average price rating in Italy is: 2.0 out of 4
The average price rating in United States is: 1.9 out of 4
The average price rating in Austria is: 1.5 out of 4
The average price rating in Germany is: 1.3 out of 4
The average price rating in Japan is: 1.3 out of 4
The country with the highest average price rating is Mexico with an average price rating of 2.7 out of 4
The country with the lowest average price rating is Japan with an average price rating of 1.3 out of 4
```

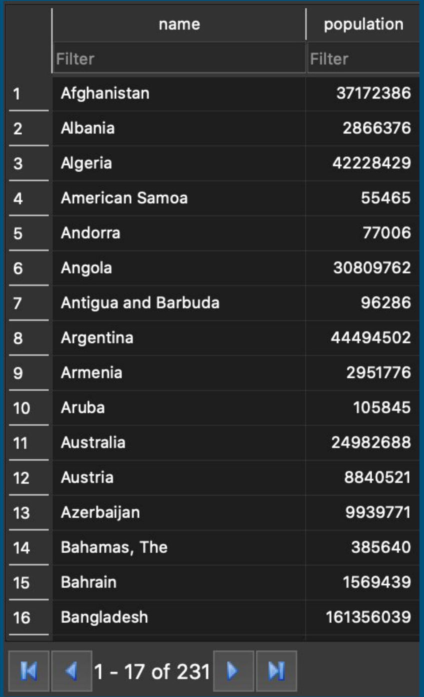
Calculations- Yelp API

Calculation 5 Explanation:

I used an almost identical process as the previous calculation however in the case of prices, this information was provided from Yelp in the form of currency symbols for the respective country. 1 symbol indicates the lowest price and 4 indicates the highest. For the restaurants that did not have a price I assumed a mid level price and gave these a value of ‘na’. I then used the number of characters in the price to indicate its rating and for those without a price ‘na’ served the purpose of indicating 2 characters (mid-level). I then created a dictionary of countries and their average price, sorted that information, and wrote the results to a file

Tables

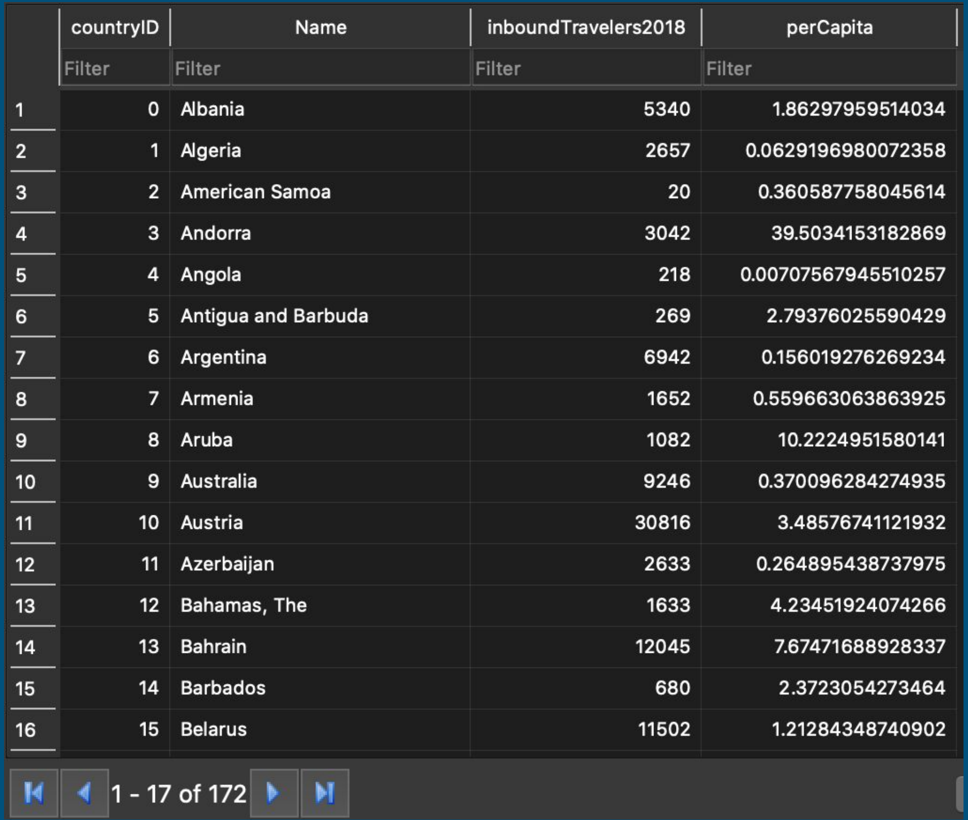
Population



	name	population
	Filter	Filter
1	Afghanistan	37172386
2	Albania	2866376
3	Algeria	42228429
4	American Samoa	55465
5	Andorra	77006
6	Angola	30809762
7	Antigua and Barbuda	96286
8	Argentina	44494502
9	Armenia	2951776
10	Aruba	105845
11	Australia	24982688
12	Austria	8840521
13	Azerbaijan	9939771
14	Bahamas, The	385640
15	Bahrain	1569439
16	Bangladesh	161356039

1 - 17 of 231

Countries

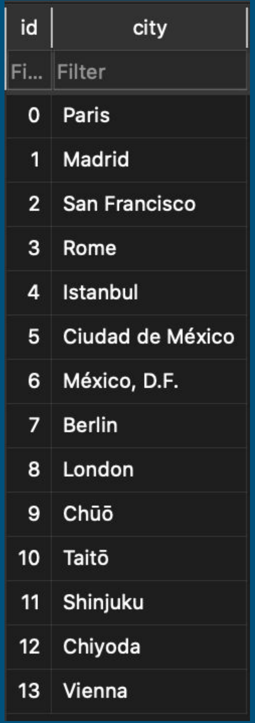


	countryID	Name	inboundTravelers2018	perCapita
	Filter	Filter	Filter	Filter
1	0	Albania	5340	1.86297959514034
2	1	Algeria	2657	0.0629196980072358
3	2	American Samoa	20	0.360587758045614
4	3	Andorra	3042	39.5034153182869
5	4	Angola	218	0.00707567945510257
6	5	Antigua and Barbuda	269	2.79376025590429
7	6	Argentina	6942	0.156019276269234
8	7	Armenia	1652	0.559663063863925
9	8	Aruba	1082	10.2224951580141
10	9	Australia	9246	0.370096284274935
11	10	Austria	30816	3.48576741121932
12	11	Azerbaijan	2633	0.264895438737975
13	12	Bahamas, The	1633	4.23451924074266
14	13	Bahrain	12045	7.67471688928337
15	14	Barbados	680	2.3723054273464
16	15	Belarus	11502	1.21284348740902

1 - 17 of 172

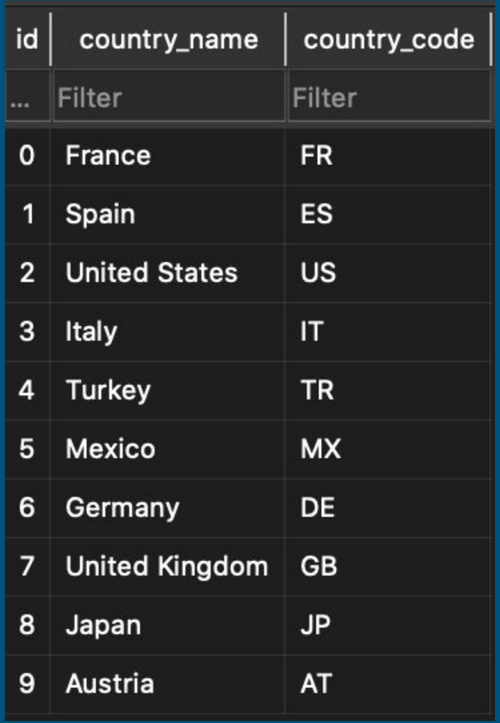
Tables

Cities



id	city
Fi...	Filter
0	Paris
1	Madrid
2	San Francisco
3	Rome
4	Istanbul
5	Ciudad de México
6	México, D.F.
7	Berlin
8	London
9	Chūō
10	Taitō
11	Shinjuku
12	Chiyoda
13	Vienna

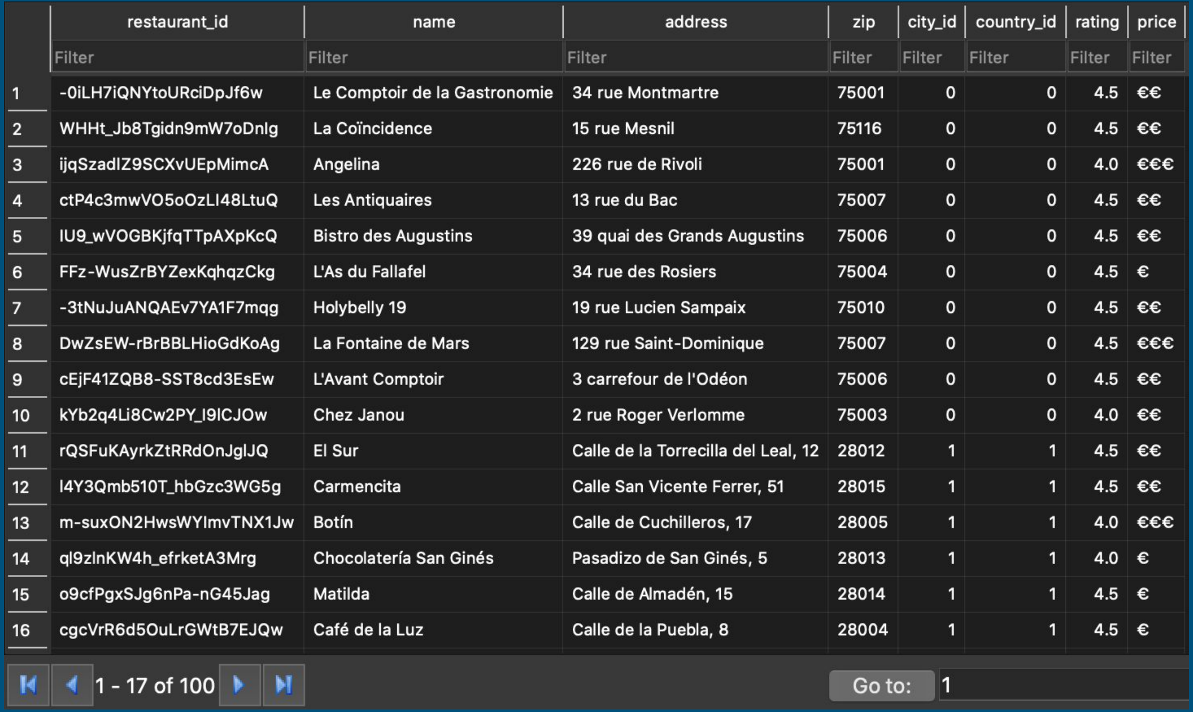
Yelp_Countries



id	country_name	country_code
...	Filter	Filter
0	France	FR
1	Spain	ES
2	United States	US
3	Italy	IT
4	Turkey	TR
5	Mexico	MX
6	Germany	DE
7	United Kingdom	GB
8	Japan	JP
9	Austria	AT

Tables

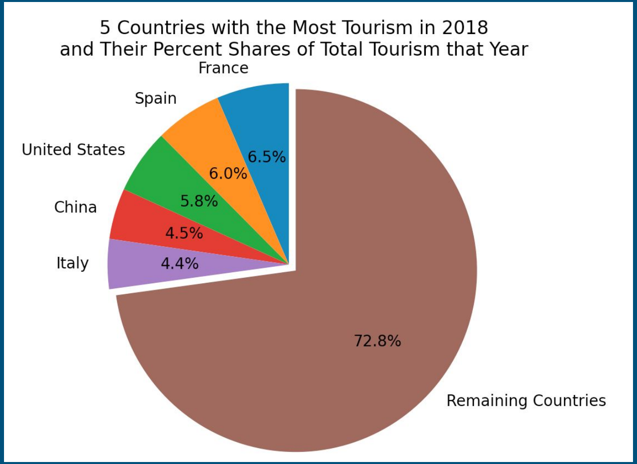
Restaurants

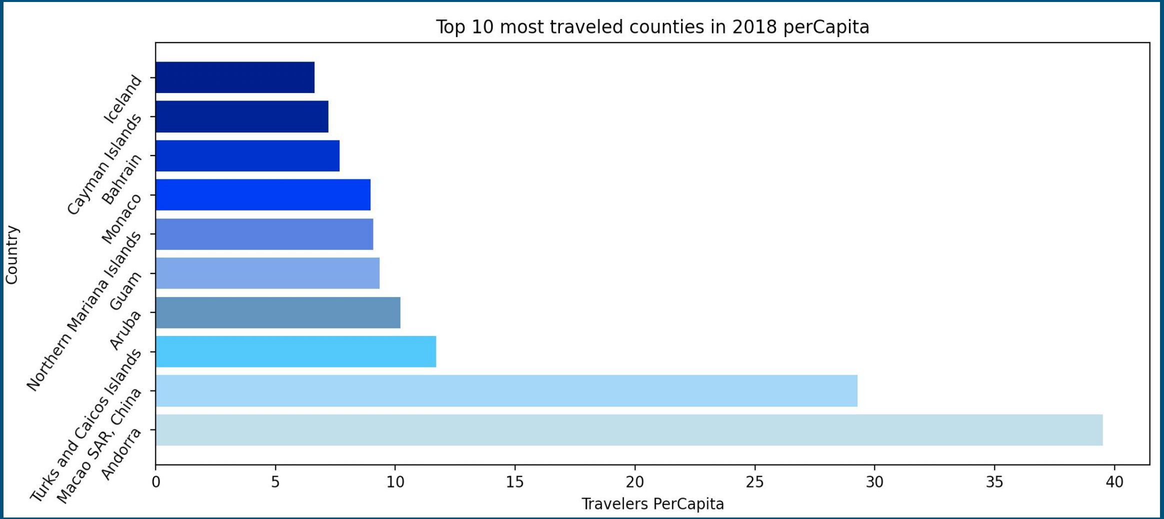


	restaurant_id	name	address	zip	city_id	country_id	rating	price
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	-0iLH7iQNYtoURciDpJf6w	Le Comptoir de la Gastronomie	34 rue Montmartre	75001	0	0	4.5	€€
2	WHHt_Jb8Tgidn9mW7oDnlg	La Coïncidence	15 rue Mesnil	75116	0	0	4.5	€€
3	ijqSzadIZ9SCXvUEpMimcA	Angelina	226 rue de Rivoli	75001	0	0	4.0	€€€
4	ctP4c3mwVO5oOzLI48LtuQ	Les Antiquaires	13 rue du Bac	75007	0	0	4.5	€€
5	IU9_wVOGBKjfqTTpAXpKcQ	Bistro des Augustins	39 quai des Grands Augustins	75006	0	0	4.5	€€
6	FFz-WusZrBYZexKqhqzCkg	L'As du Fallafel	34 rue des Rosiers	75004	0	0	4.5	€
7	-3tNuJuANQAEv7YA1F7mqg	Holybelly 19	19 rue Lucien Sampaix	75010	0	0	4.5	€€
8	DwZsEW-rBrBBLHioGdKoAg	La Fontaine de Mars	129 rue Saint-Dominique	75007	0	0	4.5	€€€
9	cEjF41ZQB8-SST8cd3EsEw	L'Avant Comptoir	3 carrefour de l'Odéon	75006	0	0	4.5	€€
10	kYb2q4Li8Cw2PY_I9ICJOw	Chez Janou	2 rue Roger Verlomme	75003	0	0	4.0	€€
11	rQSFuKAyrkZtRRdOnJglJQ	El Sur	Calle de la Torrecilla del Leal, 12	28012	1	1	4.5	€€
12	l4Y3Qmb510T_hbGzc3WG5g	Carmencita	Calle San Vicente Ferrer, 51	28015	1	1	4.5	€€
13	m-suxON2HwsWYImvTNX1Jw	Botín	Calle de Cuchilleros, 17	28005	1	1	4.0	€€€
14	ql9zlnKW4h_efrketA3Mrg	Chocolatería San Ginés	Pasadizo de San Ginés, 5	28013	1	1	4.0	€
15	o9cfPgxSJg6nPa-nG45Jag	Matilda	Calle de Almadén, 15	28014	1	1	4.5	€
16	cgcVrR6d5OuLrGWtB7EJQw	Café de la Luz	Calle de la Puebla, 8	28004	1	1	4.5	€

This table was created using the Yelp API. Tables Cities and Yelp_Countries were also created with this API. Restaurants shares a key with Cities and Yelp_Countries, this can be seen in the city_id and country_id column

Visualizations



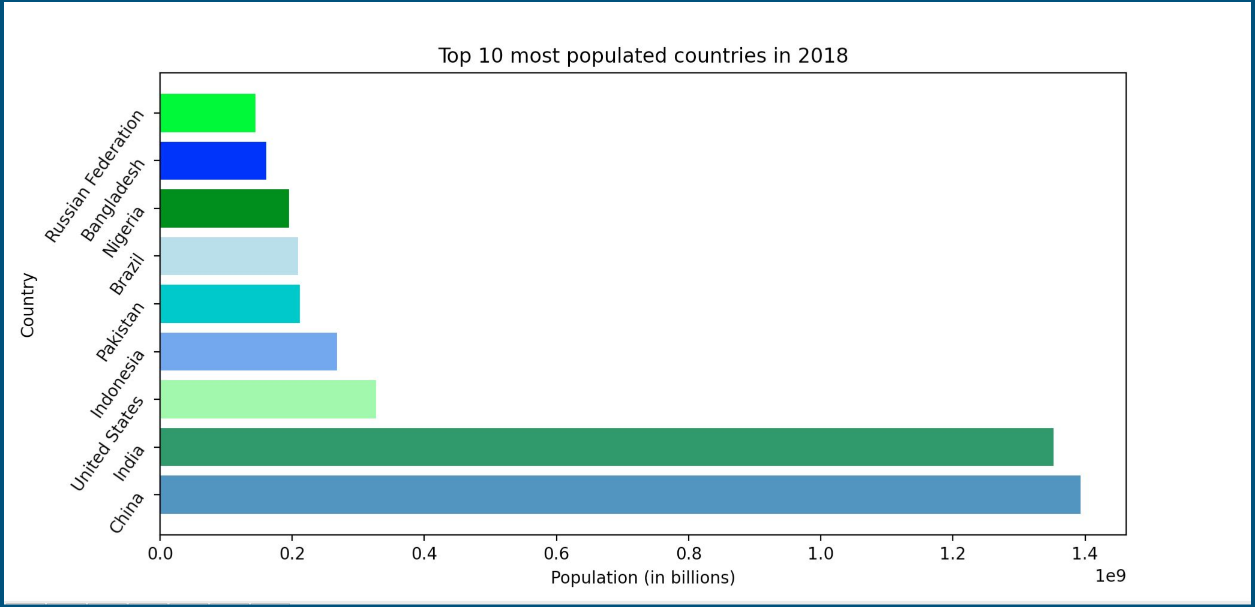


Created using Calculation 2 and drawPie(cur, conn) in visualizations.py

Takeaway: Interesting that only 5/172 countries account for over ¼ of total inbound tourists in 2018

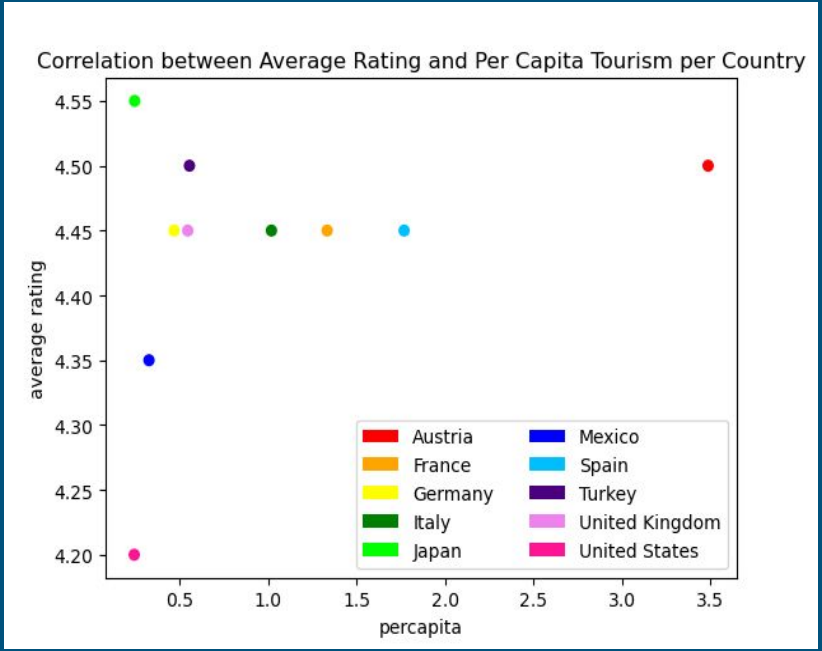
This visualization was created using the data of per capita travel by country from calculation 3. It is interesting to compare these countries to the countries in the pie chart as they are very different and smaller countries.

Visualizations



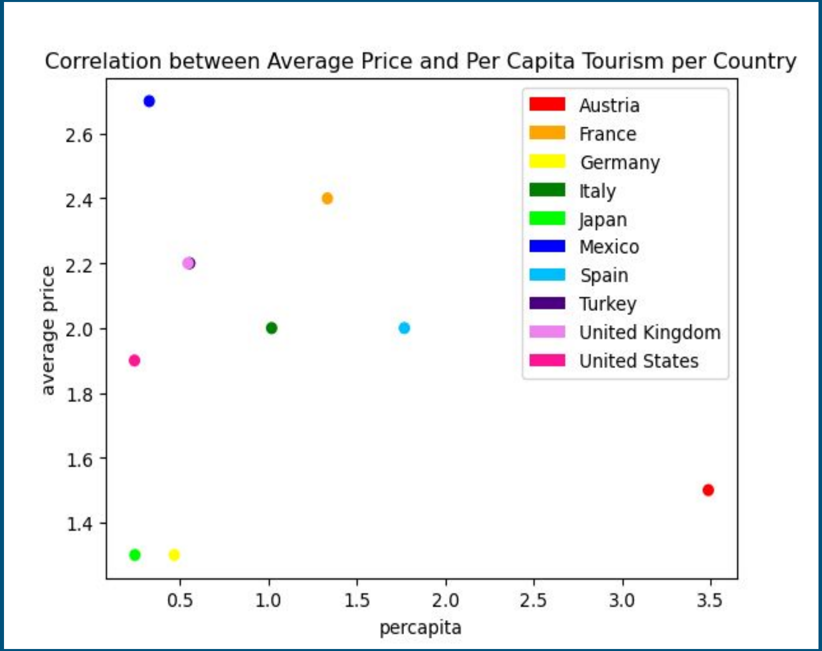
This visualization was created using the population API data. The top ten most populated countries were selected and listed in this bar chart. It is interesting how even though these countries are the largest, not all of them have a large inbound tourist rate.

Visualizations



This visualization shows the relationship between average rating and per capita tourism in each country. Each point belongs to a different country as indicated in the legend. We wanted to see if there was a correlation/relationship between frequently travelled countries and the quality of restaurants in those countries, as that is a common aspect of the vacation/travel experience. What we found is that there is not a concrete relationship as there are a few countries who have similar ratings but large differences in per capita tourism

Visualizations



This visualization shows the relationship between average price and per capita tourism in each country. Each point belongs to a different country as indicated in the legend. We wanted to see if there was a correlation/relationship between frequently travelled countries and the price of restaurants in those countries. We wanted to see if people are travelling to countries with cheaper restaurants or if restaurants in popular countries were charging higher prices because tourism is good business for them. We found that there is no clear relationship here as well since we see a variety of different combinations of price and per capita tourism

Instructions to Run Code

Run population.py 6 times:

Creates Populations table and adds 3 items in the first run, then 25 items each run until 100 rows of data are added after which the remaining is added in one run. Populations will have 231 rows after 6 runs

Run tourism.py 5 times:

Creates Countries table

Run yelp.py 5 times:

Creates 3 tables- Cities, Yelp_Countries, Restaurants. Adds 20 rows of data to Restaurants each time and will have 100 rows after running 5 times

Run join.py

Adds per capita tourism data to the Countries table. Performs calculations and creates a text file

Run visualizations.py

Creates five visualizations

Code Documentation

population.py

setUpDatabase():

Opens the file path to the database ‘finalproject.db’ and creates a connection and cursor object to that database.

fill_db():

Creates a table within the database, ‘finalproject.db’ with the columns country name and population. Connects to the World Bank api through the given url and takes up to 25 data points at a time, until over 100 have been retrieved. After 100 have been retrieved, can enter the rest of the data without stopping.

main():

calls fill_db() to populate the table.

Code Documentation

tourism.py

get_inbound_international_tourists_2018():

No inputs. Uses BeautifulSoup to fetch data on World Bank website in order to return a dictionary with keys being all countries with tourism data available and values being the number (in thousands) of inbound international tourists in 2018 for each country key.

setUpDatabase():

Opens the file path to the database 'finalproject.db' and creates a connection and cursor object to that database.

create_countries_table(cur, conn):

Takes the database cursor and connection as inputs. Returns nothing. Creates a table in the database with three columns: CountryID, Name, InboundTravelers2018.

Code Documentation

tourism.py

add_countries_from_dict(cur, conn):

Takes the database cursor and connection as inputs. Returns nothing. Checks to see if data has already been stored in database table. Depending on the CountryID already in the table, this function adds a specific group of country values to the database. For example, if CountryID 24 is already in the table, then it adds the 25th through 50th country in countryDict.

if __name__ == '__main__':

- The contents of this if statement at the end of the file calls all the functions described above and on the previous slide

Code Documentation

yelp.py:

setUpDatabase(db_name):

Takes in a database name as input and creates a database with that name in the file directory of the code file

getYelp(search_country):

Takes a country name as input and uses the API key listed in the file to make an API call to the Yelp Fusion API and returns a python object with information regarding 10 restaurants in that country

createCities(cur, conn):

Takes the database cursor and connection as inputs and creates a table called Cities in the database with two columns- id and city. It inserts rows of data into the table using the data returned by calling the getYelp function on the list of top countries

createCountries(cur, conn):

Takes the database cursor and connection as inputs and creates a table called Yelp_Countries in the database with three columns- id, country_name, and country_code

Code Documentation

yelp.py:

createRestaurants(cur, conn):

Takes the database cursor and connection as inputs and created creates a table titled Restaurants. It then uses data returned from calling getYelp on the list of top 10 countries and foreign keys from the 2 tables previously created to create a table with 8 colums- restaurant id, name, address, zip code, city_id, country_id, rating, and price. Each execution of the file will add 20 rows of data- 10 restaurants each for 2 countries at once

main():

Calls setUpDatabase using 'finalproject.db' as input and database name to create the cursor and connection. Then it calls createCities, createCountries, and createRestaurants on that cursror and connection

Code Documentation

join.py

setUpDatabase():

Opens the file path to the database ‘finalproject.db’ and creates a connection and cursor object to that database.

join_tourism():

Joins the tables population and countries through the country name and adds a column for ‘perCapita’ integer. Selects the name, inbound travelers, and population for each country. Divides the numbers of inbound travelers by the population for each country and updates the column ‘perCapita’ in the table ‘Countries’.

calcAvgCountryTouristsinThousands(cur, conn):

Takes the database cursor and connection as inputs. Returns the average inbound international tourists (in thousands) in 2018 for the 172 countries in this database. Function checks to see that all 172 countries have been added to the database before calculating the average, and if not then just returns None.

Code Documentation

join.py (cont.)

calcBiggestFiveFractions(cur, conn):

Takes the database cursor and connection as inputs. Returns a dictionary with 6 key value pairs, the first 5 being the five countries with the most tourists in 2018, and their values being the percentage of that country's tourism out of all tourism that year. The 6th key value pair is all remaining nations and the percentage of all their tourism out of total tourism that year. Function checks to see that all 172 countries have been added to the database before calculating and if not then just returns None.

averageRating(cur, conn):

Takes database cursor and connection as inputs. Selects data from the database using a join and returns a dictionary of country names as keys and average restaurant rating as the value (detailed explanation provided in calculations section)

Code Documentation

join.py (cont.)

averagePrice(cur, conn):

Takes database cursor and connection as inputs. Selects data from the database using a join and returns a dictionary of country names as keys and average restaurant price as the value (detailed explanation provided in calculations section)

writeCalculationToFile(filename, cur, conn):

Opens filename to write calculations to. Writes average inbound tourist number for all countries. Writes perCapita tourism number for top 10 countries traveled to by perCapita. Writes average restaurant rating calculation. Writes average restaurant price calculation.

main():

Takes no inputs. First calls join_tourism() and then creates a cursor and connection using setUpDatabase. Then it calls writeCalculationToFile using the filename 'calculations.txt' and a text file with the calculations will be created in the file directory of the code file

Code Documentation

visualizations.py

setUpDatabase():

Opens the file path to the database ‘finalproject.db’ and creates a connection and cursor object to that database.

percapita(cur, conn):

Selects country name and perCapita travel for the top 10 countries traveled by perCapita and makes a barchart of these using matplotlib.

population(cur, conn):

Uses the population API to select the country name and population in descending order. Makes a bar chart of the top 10 countries with highest population using matplotlib.

Code Documentation

visualizations.py

drawPie(cur, conn):

Takes the database cursor and connection as inputs. Returns nothing. Function calls calcBiggestFiveFractions(cur, conn) and uses matplotlib to input the keys and values of this dictionary into a pie chart with 6 pieces. Five pieces are for the five countries with the most tourists in 2018, and the percentage of each country's tourism out of worldwide tourism that year. The sixth piece is for the remaining countries.

drawRatingScatter(cur, conn):

Takes database cursor and connection as inputs. Uses join to select the per capita tourism values only for those countries listed in Yelp_Countries. Calls the averageRating function and stores the ratings and countries in different lists. Creates and shows a scatterplot of average rating on the y-axis and per capita tourism on the x-axis. Each point is color coded according to the country and a legend is provided accordingly.

Code Documentation

visualizations.py

drawPriceScatter(cur, conn):

Takes database cursor and connection as inputs. Uses join to select the per capita tourism values only for those countries listed in Yelp_Countries. Calls the averagePrice function and stores the pricess and countries in different lists. Creates and shows a scatterplot of average price on the y-axis and per capita tourism on the x-axis. Each point is color coded according to the country and a legend is provided accordingly.

main():

Takes no inputs. Creates a cursor and connection using the setUpDatabase function and calls the remaining functions explained in the previous 2 slides. Returns nothing but shows the plots created using the functions called

Resources

Date	Issue Description	Resource Location	Result
12/08/2020	Legend in the scatter plot would only show one label	Stack Overflow https://stackoverflow.com/questions/26558816/matplotlib-scatter-plot-with-legend	Was able to fix the legend to display all labels
12/01/2020	Unsure how to create tables with shared key	SI206 Homework 8 and Office Hours	Successfully created tables using a foreign key
11/30/2020	Trouble accessing 25 key value pairs of a dict at once for tourism.py	https://stackoverflow.com/questions/7971618/return-first-n-keyvalue-pairs-from-dict	Solved issue of sending only 25 sets of data to table at a time

Resources Continued

Date	Issue Description	Resource Location	Result
12/06/2020	Unsure of how to put percentages on each piece of the pie chart	https://matplotlib.org/3.1.1/gallery/pie_and_polar_charts/pie_features.html	Solved the issue
12/08/2020	Unsure how to ensure 'calculations.txt' would be created in the correct folder	Project2 and runestone	Solved the issue
12/07/2020	How to change bar colors in barchart.	https://stackoverflow.com/questions/3832809/how-to-change-the-color-of-a-single-bar-if-condition-is-true-matplotlib	Solved the issue