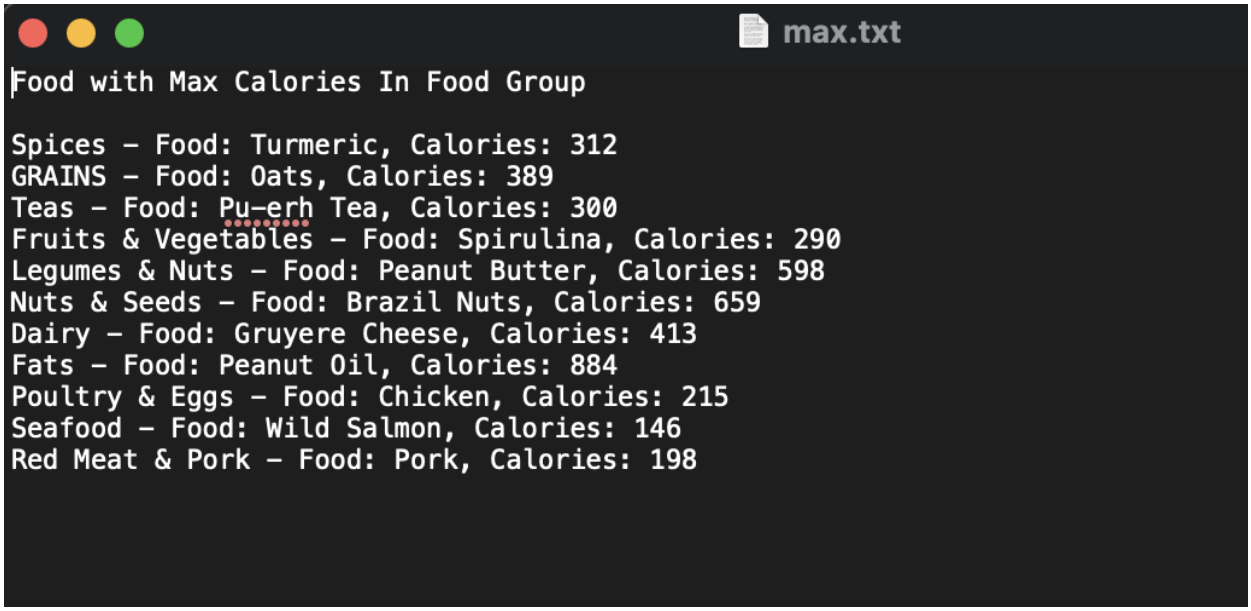
# SI 206 Final Project Report

Gabriel Liss - gabeliss@umich.edu
Emmaline Lehman - emmaline@umich.edu

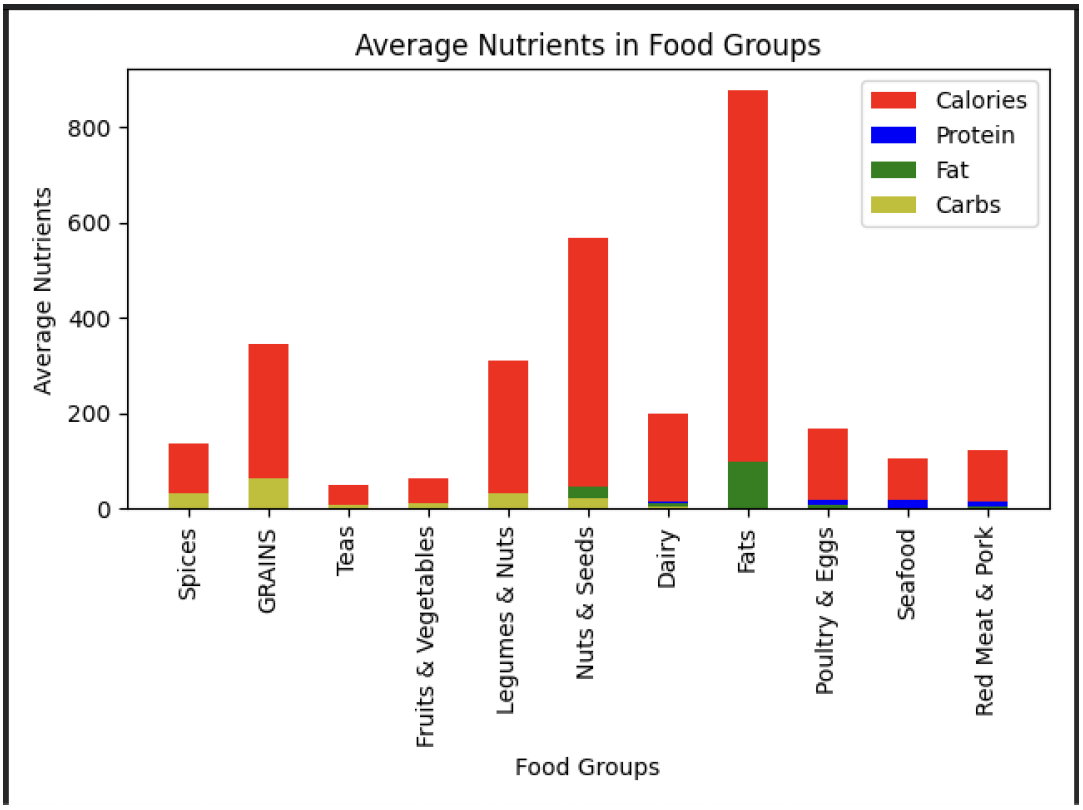
Our project aimed to analyze the top 100 healthiest foods. The original goals for our project included finding out nutritional information about the different types of food groups and the relationships between them. We wanted to learn about the average calories, protein, fat, and carbs in each food group. Additionally, our team wanted to pinpoint the specific food with the most calories in each food group. This project allowed our team to attempt to gain experience with pulling data from a website, pulling data from an API, creating a database, calculating information from the database, and displaying our information in a visual manner.

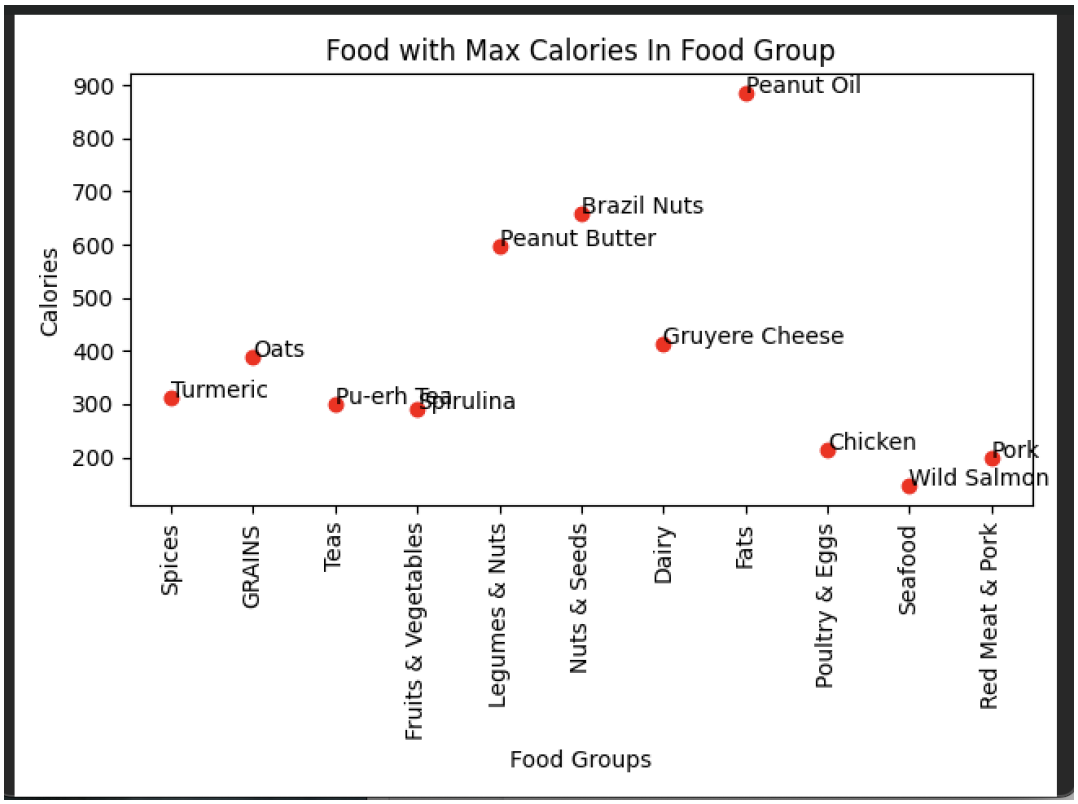
We were pleased to be able to achieve all of the aforementioned goals. Our team experienced great success in creating visually appealing graphs that display the nutritional information we wanted to learn more about, as stated in the paragraph above. We also felt that we became much better developers in the process. Some problems we encountered were figuring out how to set up the database and organize our tables. We were unsure at first what information we should store in each table, and how we would be able to connect the tables using joins. We also had difficulty figuring out how to only store 25 items for each execution of the program. We decided to create a variable that counts how many items are in each table, and use loops to ensure we only added the amount of information we needed.

**Below are the two files containing our calculations from the database:**

**Below are the two visualizations we created:**



**Instructions for running the code:**

To run the code from scratch, make sure the food.db file is deleted. Then, the first time food.py is run, it will create the tables and insert the first 25 items. Then, run food.py 3 more times to ensure the tables have 100 items. On the fourth run, the program will also output the two visualizations in png files.

**Documentation for functions:**

setUpDatabase - takes db_name as input which will be the name of the database file created. The function will then create the database and return the cursor and connection.

scrapeWebsite - takes the cursor and connection to the database as input. This function scrapes a predetermined website using beautiful soup. It then creates the Foods and Categories tables in the database if they do not already exist, and it will add 25 items to the tables. The function does not return anything.

readAPI - takes the cursor and connection to the database as input. This function gets items from the Edamam API. It then creates the Nutrients table in the database if it does not already exist, and adds 25 items to the table. The function does not return anything.

calculateAverages - takes the cursor and connection to the database as input. This function uses the Categories and Nutrients tables to calculate the average calories, protein, fat, and carbs for each food category. This information is stored in a dictionary which is then returned.

writeAverages - takes in a dictionary containing the information of average nutrients for each food category. This function outputs the dictionary to a text file called averages.txt. It does not return anything.

createAveragesGraph - takes in a dictionary containing the information of average nutrients for each food category. This function creates a chart using matlab to display the information in the dictionary. It outputs the chart to a file called averages.png. It does not return anything.

calculateMax - takes the cursor and connection to the database as input. This function joins the Categories, Foods and Nutrients tables to calculate the food item with the most calories for each food category. This information is stored in a dictionary which is then returned.

writeMax - takes in a dictionary containing the information of the food item with the most calories for each food category. This function outputs the dictionary to a text file called max.txt. It does not return anything.

createMaxGraph - takes in a dictionary containing the information of the food item with the most calories for each food category. This function creates a chart using matlab to display the

information in the dictionary. It outputs the chart to a file called max.png. It does not return anything.

## Resource Documentation

| Date | Issue Description | Location of Resource | Result: Did it solve the issue? |
|---|---|---|---|
| 12/4/22 | Forgot how to use joins in SQL | w3schools - https://www.w3schools.com/sql/sql_join.asp | Yes, we were able to successfully join our tables |
| 12/7/22 | Labels on x-axis overlapped and were difficult to read | Stack Overflow - https://stackoverflow.com/questions/10998621/rotate-axis-text-in-python-matplotlib | Yes, we were able to change the labels so every label could be read |
| 12/8/22 | Did not know how to add annotations to points in scatter plot | Mathworks - https://www.mathworks.com/help/matlab/ref/scatter.html | Yes, we were able to successfully add the annotations |

Git Repo: https://github.com/gabeliss/foodgroup