Notes On Data

Derek R Neilson

September 26, 2024

Abstract

This document contains notes on the data. The notes are intended to demonstrate how I filter and manipulate the data, and are purely for my instructor to review.

1 Introduction

In data analysis, the ability to effectively filter and manipulate data is crucial for extracting meaningful insights. This document outlines the methodologies and tools I employ to pre-process and analyze the dataset. The primary focus is on cleaning the data, handling missing values, and transforming data to suit the analytical objectives. These notes serve as a comprehensive guide for understanding my data processing workflow.

2 Data Collection

The data was collected from `https://fdc.nal.usda.gov/`. The dataset is 2.9GB and is labeled `Branded` and is in JSON format. I chose this dataset because it is large and one can assume that it has the most rows because it is so large.

To download the data, I used the following commands:

```
wget https://fdc.nal.usda.gov/fdc-datasets/FoodData_Central_branded_food_json_2024-04-18.zip
unzip FoodData_Central_branded_food_json_2024-04-18.zip
rm FoodData_Central_branded_food_json_2024-04-18.zip
```

Listing 1: Download, Extract, and Remove Zip File

As shown in Listing 1, the commands download, extract, and remove the dataset file.

It is worth noting that I am using git to track changes in the code and data. The git commands will not be shown in this document for brevity.

3 Data Inspection

I received the following files after extracting:

- `brandedDownload.json` - I am assuming that this is the main file
- `foundationDownload.json` - I am assuming that this is a supporting file

The first step to inspecting the data is to view it.

```
less brandedDownload.json # the output is too large to show here and is not useful
# I am going to use jq to view the data
jq . brandedDownload.json # this results in a segmentation fault because the file is too large
# I am going to use a stonger server to veiw the data
# For security resons, the ip address and username are redacted
sftp -P port username@ip_address
put brandedDownload.json DEV/Project/Data
put foundationDownload.json DEV/Project/Data
bye
ssh username@ip_address -P port
```

Listing 2: View the Data

As shown in Listing 2, the file is too large to view on my local machine. I will use a stronger server to view the data. Note that there is a assumption that all commands that follow are run on the server. From here on, I will refer to the server as being the machine that I am using to view the data. To get the data on the server, I used sftp to transfer the file to the server.

```
jq . brandedDownload.json | less # failed because the file is too large
jq --stream . brandedDownload.json | less # this works because it streams the data
jq --stream . foundationDownload.json | less
```

Listing 3: View the Data on the Server

After looking at the head of the data, I can see that the data is in unstructured JSON format. I will use the CSV data insted. The JSON data will not be included in this document for size reasons.

4 Data Collection (CSV)

```
rm brandedDownload.json foundationDownload.json # remove the JSON files
wget https://fdc.nal.usda.gov/fdc-datasets/FoodData_Central_branded_food_csv_2024-04-18.zip #
  ↪ download the CSV file
unzip FoodData_Central_branded_food_csv_2024-04-18.zip # unzip the file
mv FoodData_Central_branded_food_csv_2024-04-18/* . # move the files to the current directory
```

Listing 4: Download, Extract, and Remove Zip File

As shown in Listing 4, the commands download, extract, and remove the dataset file in CSV format. I will use the CSV data for the rest of the analysis.

```
total 2853M
-rw-r--r-- 1 derek derek  870M Apr  5 12:07 branded_food.csv
drwxr-xr-x 3 derek derek    1M Sep 26 09:01 build
-rw-r--r-- 1 derek derek    1M Apr  5 12:12 Download API Field Descriptions.xlsx
-rw-r--r-- 1 derek derek  123M Apr  5 12:18 food_attribute.csv
-rw-r--r-- 1 derek derek    1M Apr  5 12:09 food_attribute_type.csv
-rw-r--r-- 1 derek derek  351M Apr  5 12:18 food.csv
-rw-r--r-- 1 derek derek 1387M Apr  5 12:23 food_nutrient.csv
-rw-r--r-- 1 derek derek  124M Apr  5 12:18 food_update_log_entry.csv
-rw-r--r-- 1 derek derek    1M Sep 25 19:22 makecsv.py
-rw-r--r-- 1 derek derek    1M Apr  5 12:09 measure_unit.csv
-rw-r--r-- 1 derek derek    1M Apr  5 12:08 microbe.csv
-rw-r--r-- 1 derek derek    1M Sep 26 08:10 notes.tex
-rw-r--r-- 1 derek derek    1M Apr  5 12:08 nutrient.csv
-rw-rw-r-- 1 derek derek    1M Apr  5 12:12 nutrient_incoming_name.csv
-rw-r--r-- 1 derek derek    0M Sep 26 09:05 sizes.log
```

Listing 5: list the files

I then looked at the head of each csv file to see what was in the files. I will not include the output here for brevity. `head -n 1 *.csv`. The only file that I am interested in is `branded_food.csv`. I will use this file for the rest of the analysis. But I also noticed that none of the files had caloric information. As a result, I will use a external dataset to get the caloric information. From a quick search, I found a api `https://platform.fatsecret.com/platform-api`. I will use this api to get the caloric information for each food. It dose cost money to use this api so I will calculate the cost of using this api before I use it. I stord the key in a external file called `.env`. I will not include the key in this document for security reasons. I will use a python script to get the caloric information for each food. At this time I will make a virtual environment to run the script. `python3.12 -m venv .venv` and `source .venv/bin/activate`. I will keep track of any dependencies that I use in a `requirements.txt` file. The next step is to validate the data. Reefer