**CANDY\_ETL**

Amanda Seagle, Mansi Majithia, Emmarose McCoig

**OBJECTIVE**

Our group decided to create a database of popular Halloween candy and its nutritional information that could potentially be used to see if there is any correlation between nutritional content (i.e. grams of sugar, water content, or grams of protein) and candy popularity.

**EXTRACT**

We selected two CSVs as our data sources:

1. A ranking of Halloween candy that people most prefer from FiveThirtyEight, created through a poll of 8,371 IP addresses on FiveThirtyEight.com. This CSV contains candy names, categories (‘fruity’, ‘chocolate’ etc), win percent (popularity), price percent, and sugar percent:

- Article: <https://fivethirtyeight.com/videos/the-ultimate-halloween-candy-power-ranking/>

-CSV (from github): <https://github.com/fivethirtyeight/data/blob/master/candy-power-ranking/candy-data.csv>

1. Our nutritional data was extracted from the USDA’s food composition standard reference. The USDA is a reliable and standardized source for nutritional information that provides downloadable csvs of their data. We chose to use the abbreviated SR28ABXL.Zip file because this contained the most relevant columns for our use (the non-abbreviated version contains information on alcohol, caffeine, theobromine and many other nutrient contents that we weren’t interested in examining). We selected the excel format because this would be easiest to manipulate as a csv.

- CSV: <https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/methods-and-application-of-food-composition-laboratory/mafcl-site-pages/sr11-sr28/>

We first imported the data as pandas dataframes into a Jupyter Notebook to examine the types of transformations that we would need to complete.

**TRANSFORM**

Cleaning, filtering, merging

1. The USDA data provides data on all food types, so after importing the csv into a dataframe, we identified that we would need to split the ‘Shrt\_Desc’ column to extract the relevant food category (‘Candies’) and candy name.
2. We split the ‘Shrt\_Desc’ column on commas, and performed a .loc to extract foods in the ‘Candies’ category, and then dropped columns with ‘None’ values. We ended up with 134 rows of candy, and exported this to a new csv file.
3. Part of our transformation took place in Excel, where we had to change the names that the USDA had given various candies to match the candy names in the FiveThirtyEight dataset (i.e ‘Butterfinger Candy Bar’ to ‘Butterfinger’). Since the dataset was relatively small, it was faster to do this by copy-pasting from the FiverThirtyEight CSV into the extracted USDA candy dataset in Excel rather than cleaning the names programmatically.
4. Once our candy names matched, we created dataframes for both csvs using pandas and removed foreign characters from the resulting candy names.
5. We changed the column names to increase ease of formatting to our relational database, and then merged our CSVs on our candy name columns using an inner join, thus dropping any columns with incomplete data.
6. We then dropped any irrelevant columns (like the USDA database number) and duplicates. We ended up with 34 rows.

**LOAD**

1. We loaded our final data table into a relational database in PostgreSQL called ‘candy’ (merged in pandas so no need to join in SQL). A relational database is easy to scale and maintain data integrity, so we thought for this small database with potentially many more entries, this made the most sense.