**Extract, Transform, Load**

**Extrac**t: where did we get our data?

We used a wine review dataset, found on kaggle.com (data scraped from WineEnthusiast). From this dataset, we extracted 2 files; one in a csv format, and one in a json format. The csv had the columns title, country, province, region\_1, region\_2, variety, winery. The json had the columns title, description, designation, points, price, taster\_name, taster\_twitter\_handle.

**Transform**: how did we transform the data?

**-Dataframes**

The first step was to load the json and csv files into pandas dataframes to facilitate the exploration process.

***-Data Encoding***

We found that the json and csv were in another encoding, so many characters would appear different and in a strange form, so we checked with two encodings to see which one fit best, and found out that UTF-8 was the better option.

***-Union of the two sets of data in one final dataframe***

We merged the two dataframes on the column “title”.

***-Eliminated the duplicated data***

We eliminated the duplicated data on the merged dataframe to have consistent data.

***-Removed useless data columns***

We eliminated the region\_2 column because 90% percent of the data was NaN and the other 10% had the same value of region\_1.

***-Eliminated NaN values***

We dropped all the rows with null values to have all data with values defined.

***-Index insertion***

We inserted a column called index using reset\_index. This created an index starting from zero, but to read the final table into pgAdmin, an index starting from 1 was required. To achieve this, we added 1 to each element of the newly created index column. This index was to be used as the primary key in pgAdmin.

**Load: The final table**

***-Database***

Created the WINE\_DB database.

***-Table creation***

With the structure defined in the complete dataframe, we created a table in PostgreSQL that will receive our data from pandas.

***-Connection***

Created the connection in Pandas to PostgreSQL, and to the wine\_db database to export the data.

***-Data export***

We exported the data from the dataframe to the PostgreSQL table with the to\_sql function.

Then we exported from PgAdmin to a csv that was exported to an excel document using the UTF-8 encoding.