Data Cleanup & Analysis for ETL Project

**1. Extract: your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc).**

We extracted two datasets ‘Wine Reviews’ and ‘Countries of the World’ from Kaggle. Both are in csv format.

Wines Source: <https://www.kaggle.com/zynicide/wine-reviews>.

Countries Source: <https://www.kaggle.com/fernandol/countries-of-the-world>.

**2. Transform: what data cleaning or transformation was required.**

First, we identified that both datasets contain the column “country” which will allow us to join the two.

Then we utilized MySQL to create the database ‘wine\_climate’ with the tables ‘wines’ and ‘country\_climates.’ Subsequently read in the csv files in Python and imported pandas and sqlalchemy. We took the following steps to clean and organize the two datasets. While these steps do not reflect the exact order in which the code was written out, this is the most succinct way to understand the overall process we took:

1. After reading in the ‘Wines’ data file, we renamed the “unnamed:0” column to “id” so we could sort the column by their number in ascending order. And renamed “points” to “score.”
2. Then created a new dataframe called ‘wines’ containing the columns “id,” “country,” “designation,” “score,” “price,” and “variety.”
3. After reading in the ‘Countries’ data file, we created a new dataframe called ‘country\_climate’ containing the columns “countries” and “climate.”
4. Next we dropped the null value in the “country” column from both dataframes.
5. In order to join the two datasets, the column “country” was spelled out differently in the two datasets. One contained an uppercase for the “C” in “Country” while the other contained a lowercase “c.” We changed the uppercase to lowercase so the column name is consistent in the two dataframes.
6. Lastly, we established a connection between the Python code and MYSQL so the data could be transformed and populated in the respective tables ‘wines’ and ‘country\_climates’ via MySQL.
7. When verifying in MYSQL for the data in both tables, initially only the ‘country\_climate’ table populated data so we had to troubleshoot why MySQL could not read in data from the ‘wines’ table (this issue was only occurring for PCs and not Mac computers). We noticed the file contained words with accents and symbol, so we then updated the Python code in two places with “encoding='latin-1.'” The first being the line of code where the file was originally read in and also the line of code for establishing a database connection between Python and MySQL. This update enabled MYSQL to read in words containing symbols and accents in the ‘wines’ dataframe and allowed the entire table with data to appear.

**3. Load: the final database, tables/collections, and why this was chosen.**

The final tables created in MySQL are called ‘wines’ and ‘country\_climate.’ In the ‘wines’ table, the columns include “country,” “designation,” “score,” “variety,” and “climate.” In the ‘country\_climate’ table, the columns include “country” and “climate.” After noticing the two tables spelled “United States” differently: one as “U.S.” and the other “United States,” we updated the variable “United States” in the ‘wines’ table so the two tables’ naming convention is consistent. Lastly, we joined the two tables by grouping the column “country.”

To further the exercise, we thought it would be interesting to look at how the countries rank by wine scores. We averaged the wine scores for each country then ranked the countries in

descending order. Surprisingly, the top 3 countries that have the highest rated wines are: England, India, and Austria.