# **ETL Project**

# Preparing wine lovers to visit the wineries recognized by *Wine Enthusiast* for top-rated wines

### Group Members: Zach Elson, Martin Wehrli, Katherine Sullivan

## **Extract**

Data for this project came from two sources, Wine Enthusiast Magazine ([www.winemag.com](http://www.winemag.com)) and Google ([www.google.com](file:///C:\Users\koegs\Desktop\class\ETL%20project\ETL-project\ETL-current\www.google.com)). The wine data was downloaded from Kaggle.com in the form of 16 separate CSVs. The data in Kaggle was originally sourced from the Wine Enthusiast Magazine website.

Hotel and restaurant data was extracted using Google’s Geocode API platform (<https://maps.googleapis.com/maps/api/geocode/json>) and performing JSON requests . Wine region and country information from the wine dataset was used to return latitude and longitude for each winery. This latitude and longitude data was then used to pull the nearest lodging and restaurants to each winery using Google’s Place API platform (<https://maps.googleapis.com/maps/api/place/nearbysearch/json>).

## **Transform**

Step 1: Build and Clean Combined Wine Dataset*Transforming* – Initially, we combined 16 CSV files into one larger dataset “all\_wines4.csv”.   
  
*Cleaning* – Each of the 16 CSV files required UTF-8 encoding during the building of the larger dataset. The only other cleaning required at this point was to add an index to the combined wine dataset.

Step 2: Create and Clean SQL Wine Database, Tables and Queries in PostGres PGAdmin  
*Transforming* – With PostGres PGAdmin we created an “all\_wines” table and imported data from “all\_wines4.csv”. We then created a query for wines with rated 95 with a price of $30 or less. Lastly, these results were exported to the CSV file “thirty\_dollars.csv”.  
  
*Cleaning* – Several cells within the large wine CSV ended with single quotes. This is the default escape character in PGAdmin. Once escape character was set to “\” these cells stopped erroring out and the table was populated from the “all\_wines4.csv” file.

Step 3: Build and Clean Dataset of Hotels and Restaurants in Proximity to Wineries  
*Transforming* – Datasets of hotels and restaurants nearest to each winery were built using the results from the Google API data extractions.

*Cleaning* – The file “thirty\_dollars.csv” required cleaning before the data could be used to pull Google API longitude and latitude values. Specifically, some of the “country” and “region” columns were modified, so Google APIs would recognize the relevant city.

The “country” column for all wines produced in the United States was manually changed from “US” to the specific state where the wine was produced. This was done so the Google API’s would correctly identify the state of origin.

The “region” column reflects the dynamic within wine region definition where sometimes a wine region is both an actual town and a geographically defined wine region. (e.g. Burgenland in Austria is both a city and a recognized wine region). In this case, no cleaning was required. However, some wine regions are geographically defined but are not actual cities. In this case, the wine region was replaced with a city within the wine region (e.g. Kremstal is an Austrian wine region but Krem is the city within the Kremstal wine region recognized by Google APIs).

Additionally, the region listed for U.S. wines was the respective state, but the wine region and city were reflected in the “subregion” and “subsubregion” columns. In these instances, the “region” column was changed to reflect the corresponding city.

Finally, both the original and cleaned data are reflected in the csv to demonstrate the before and after cleaning data values. The original date is reflected in the “country” and “region” columns. The cleaned data is reflected in the “country\_clean” and “region\_clean” columns.

## **Load**

[We need to decide if we are going to combine the datasets into one or leave them as three and link with keys