# **ETL Project**

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

### Group Members: Zach Elson, Martin Wherli, 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. 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 its data could be used to pull longitude and latitude values. The “region” column often had city data instead of …[MARTIN TO EDIT]

## **Load**

The final databases from our project are:

* all\_wines.csv