**ETL Project Report**

**French Tourism 2017:**

**E:** Source from Institut national des statistiques et des études économiques (National Institute of Statistics and Economic Studies). The site is in French, but gladly google translated it automatically. I was able to download an excel file with six sheets of different data. A lot of the data that could have be useful was not public, and pay for, but I was able to find a source with significant data for 2017 tourism information.

**T:**I read the excel file into jupyter lab, created six data frames from the excel file and sheets. For cleaning, I renamed columns, created new indexes, dropped rows through pandas. I created a database in pgAdmin and six pgAdmin tables correlating to the six data frames in jupyter. I transformed and loaded the jupyter tables into the tables in the database.

**L:** Loaded to my database, but had to rename some columns in pgAdmin tables to match the six data frame columns in jupyter.

**French Chocolate:**

**E:**I wanted to find datasets on French chocolate. I found a dataset on Kaggle that listed chocolate companies around the globes, expert ratings, and other attributes of the chocolate.

**T:** I exported the csv into jupyter notebook by using pandas. I started cleaning

the data by using “.loc” to narrow down the dataset to companies located in France. I wanted to make a separate dataset with company ID and company I used “.unique” on the column of “companies” to create the data frame. I created another dataset with the chocolate\_ID, company, review\_date, and expert rating by selecting the columns from the original dataset. The last dataset I created was an ingredients dataset I transformed the original dataset from strings into a Boolean (True/False) for each ingredient column. This was a little tricky at first I tried “.map” however the output was a list of data that I would then need to recreate into a dataframe. I decided to use “.replace” as the output was a dataframe and was much easier to work with.

**L:** Loaded data into database tables.

**French Wine :**

**E:** I located french wine review data from winemag.com. While no data was available to download I found the search results page pretty easy to scrape. One interesting problem was that the appelation data for each wine was a part of the title string I got from scraping. Luckily the data was enclosed in parenthesis and I was able to use the string.find() function to strip out that information. For some reason my scrape function returned 2 copies of each title. I was able to resolve this by stripping out every other element from the returned list (mylist[::2]). I wanted to keep review and price information as straight integers so I also stripped the dollar sign and ' Points' from the scraped information.

**T:** Since I was able to scrape the exact data I wanted there wasn't much transforming to do. The only issue was that some of the wines didn't have prices associated with them and were listed as 'N/A'. Using pandas I replaced all N/As with a 0 so they could be loaded into the associated columns as integers.

**L:**Load process was straight forward from here. I just used the pandas.DataFrame.to\_sql() function.

**French Cheese:**

**E:** My focus was on different and most popular cheeses from France. I used three sources: Opendatasoft: List of French Cheese and Eurostat: List of dairy products obtained per country in 2019.

**T:** Was able to download two data as CSV files and created a csv file for the three most popular milk sources available for cheese making. I manually inserted the three sources of data and provided a primary key number for each one and names the file milk\_source.csv.

I completed a query that indicated the quantities of cheese exported throughout European countries by milk source type.

The biggest challenge was finding “free” data. Most of the data searches came up with information I could have used for the project but was unable to access free information. It wasn’t until I found the Opendatasoft and Eurostat that I was able to get the information I needed. I was able to choose to download csv files, so the manipulation of the tables and query were pretty simple.

**L:** Loaded the data frames into pdAdmin using pandas.DataFrame.to\_csv() function.