MJ Carnahan

Brittany Clemmons

Meagan Colley

ETL PROJECT REPORT

### Extract:

* We searched Kaggle to find our datasets. We settled on two datasets that observed honey production and pesticide use in the United States. The first dataset was an overview honey production across while the other controlled for pesticides used. It was a nice pairing because many columns overlapped and shared similar data.
* The data was presented in multiple csv files.

### Transform:

* The first step was to load the proper csv files in a jupyter notebook and convert them into Pandas dataframes. Because we were working with multiple csv files, we had multiple datasets in separated in multiple dataframes.
* Next, dataframes sourced from the same csv were merged together using the “append” function provided from Pandas.
* We chose to merge our datasets using data from the columns “State” and “Year” but had to filter the data by year to create a common time frame.
* In the honey production dataset, the data included the years 1998-2007. In the pesticide dataset, the data included years 1998-2012. We removed the years 2008-2012 in the larger data set using this line of code:
  + honey\_df = honeydf [ honeydf ['year'] < 2007 ]
* The last step of transforming the data was to drop irrelevant columns, such as ones with repeating data, and renaming the columns so they would load nicely into PostgreSQL.

### Load:

* Although we originally intended to use PostgreSQL, we decided to use MongoDB because the setup for a NoSQL database was simpler to complete than the SQL database.
* In using SQL, we ran into an error where the column names “state” and “year” are considered SQL keyword functions and we could not use those titles for our column names. We resolved the error by renaming our columns in the final dataframe and attempted to reload the data. Alternatively, in the time it took to resolve the SQL error, MaryJane successfully determined how to load data into MongoDB.
* In order to successfully utilize MongoDB, we imported PyMongo, from PyMongo imported MongoClient and lastly established our database connection and client.
* Next, we named our database in Honey\_db in MongoDB, and named our collections honey\_db and honey2\_db.
* We had to convert the dataframes into dictionaries for MongoDB to read them in. We used this line of code (interchanging the names of the dataframes):
  + data\_dict = pesticidedf.to\_dict("records")
* We inserted both collections into MongoDB using this line:
  + collection2.insert\_one({ "index" : "state", "data" : data\_dict2 })’
* Lastly, we imported the dataframes as CSV files which isn’t necessary when using MongoDB but we did so in anticipation of using SQL