ETL Project Report

**Objective:**

The goal of this project is to perform what is commonly referred to as ETL. ETL stands for extract, transform, and load. This is a practice that is crucial in creating data that is clean and usable for the purpose of analyzing, storing, etc. The process can be broken down into its individual parts: extracting or acquiring the data from a source or sources, transforming or cleaning it into a form that is more readable and usable, and finally loading the data into a database or table.

**Extract:**

Data is available in a variety of formats from a variety of sources. For this project, data was collected from Kaggle, a free-to-use website that allows the open sharing of databases that cover diverse topics, and NYC Open Data, a free-to-use website that contains various NYC datasets in various formats. The first dataset utilized outlines recent NYC property sales around the five boroughs. This dataset was downloaded from Kaggle, in the format of a CSV file. The second dataset utilized is a NYC tree census. This dataset contains information about the trees located throughout the five boroughs of NYC. This was downloaded from the NYC Open Data site, in the format of a raw JSON file.

[Add used dataset url here?]

**Transform:**

To transform this data into a usable form, Pandas, a popular Python library, was utilized. Pandas proved to be very effective here due to its ability to easily load data from a CSV (or other) file into a pandas dataframe, where data can be manipulated into a form the user finds relevant to their analysis. The CSV file was read into a pandas dataframe. [NYC property sales data only contained a borough ID and was missing the borough name. To join the NYC property sales data and the NYC tree data, a borough name column is needed, so we added a borough name dataframe and merged it to the existing dataframe. ]. The JSON file in its raw form is a series of nested dictionaries, and therefore a quick look through the raw file is necessary. The file was broken down into two sections: “meta” and “data.” The data could be easily loaded into the pandas dataframe, however columns titles proved to be more difficult. An empty list was created, followed by a for loop that would cycle through the meta data and grab all the column titles and populate the list with them. This yielded the dataframe with the correct corresponding columns titles. Not all the columns are relevant to us, therefore the next step in our cleaning was dropping any columns that will not be used later. This left us with [insert what we kept]. In order for the data to be placed in SQL tables (in the hopes of performing a join later), the columns needed to be renamed to match the column names of tables that will eventually be made using Postgres. Additionally, as a precaution, any null values in our dataframes were dropped. Due to SQL being a relational database, the tables will need to preferably possess something that matches to be able to specifically join on.

**Load:**

After we were satisfied with the format of our dataframes, they were loaded into tables in Postgres (pgAdmin4) by forming a connection to the jupyter notebook through the creation of an engine (ORM). This involved creating tables that we will eventually load our data into. Each table is assigned a primary key (typically id) for relational purposes. We created tables with the number and names of columns present in the dataframes we previously created.

**Conclusion:**

The goal(s) of the project were completed successfully. The motivation was not to perform an analysis on the data itself, but rather to prepare that data in such a way that an analysis would not be limited by any inconsistencies or formatting errors due to how the data is stored. Now that these two datasets have had ETL performed on them, they are potentially ready for an analysis. A possible use for the cleaned datasets is to see if there are any correlations between trees located in NYC, and the property values at various locations. Specifically, for example, are there more trees present in high-value areas? Are the health of the trees correlated with the property values? A potential hypotheses to test here would be that if the trees are located in an area with higher value properties, then the health of the trees would be better than that of trees in low value areas.