**Problem.**

Bike ride sharing programs are on the rise in Las Angeles, CA. In addition, crime rates are consistently high in the same region. Our customer wanted to understand the extent to which criminal complaints might be associated with bike ride sharing frequency—completely hypothetical.

**Extract.**

We found two distinct data sets: (1) a bike ride sharing data file from Kaggle and (2) crime reporting data from Data World. Both files were in .csv format. Standard ITP were used to download the two files. The crime data file had 16 total variables. The bike sharing file had 22 total variables.

Transform.

We first observed that the zip codes in the biking data file were incomplete. In order to fix this issue, we used Pandas to import the data files, created a new column in the bike dataframe, and used a for loop with Google APIs. Next did the following:

* Called json with try & except block to fill in new zip code based on lat/long in biking df
  + If except was triggered, then numpy “NAN” was inserted into the “New\_Zip” column
  + We dropped all “NAN” values from the “New\_Zip” column
* Created a new df that included “New\_Zip” and “Trip\_ID”, grouped by New\_Zip using .count() to show how many trips were associated with each zip code
* Reset index to regular index in the biking df
* Rename columns as 1\_word to include bike\_ride\_count for future insertion into PG Admin 4
  + Bike\_Ride\_Count replaces “Trip\_ID”
* Bike\_Ride\_Count now includes “Zip” and “Crime Dr.No”
* Group by “Zip” and used the .count() function
* Reset index, rename column to “Crime\_Count”

**Load.**

The following steps were followed for loading the clean data into a PostgreSQL database using PG Admin 4:

* Create new database using PG Admin 4 to generate “Bike\_Crime\_DB”
* Connect to local database “Bike\_Crime\_DB” using PG Admin 4
* Create table for Bike LA; named column “New\_Zip” to match column in Pandas DF
* Created Bike\_Ride\_Count to match Pandas DF
* Created Table “Crimes” with “Zip\_Code” as int and “Crime\_Code” as int, to match DF
* Connect to local database using PostgreSQL and password key config file
* Created engine
* Check table names on engine
* Used Pandas .toSQL to load data into database tables
* Used Pandas .pd.read\_sql\_query to validate data

**Why did we choose these data sets?**

We used the selected data sets in order to provide evidence-based data for our clients’ investigation into possible relationships between bike ride sharing programs and crime in Las Angeles, CA. The overall variables were reduced from 38 variables to 4 variables using our procedures.