**ETL Project – Target Location by Demography**

**Group Members:**

William Taborn, Steven Nguyen, Kaylin McLeod, Janet Namulondo and Ganeshkumar Gurunathan

# Project Proposal:

* Scrape Target Location by City.
* Use Population Census CSV file
* Clean CSV files
* Merge by County/State
* Find correlation between Target store location and demographics.

# Data Source:

* Target Location : <https://www.allstays.com/c/target-locations.htm>
* Census: CSV file from **Kaggle.com**
* City with County Info: <https://simplemaps.com/data/us-zips>

# Extraction

# Target Locations data:

* + - Used **BeautifulSoup** to web scrap allstays website, get Target Link for each state.
    - For each state link, Used BeautifulSoup, to web scrape Target location details containing City, State, State Abbr.

# Census Data:

* + - Download Cenus CSV from **Kaggle.com**, load it in DataFrame.

# City with County Info:

* + - Download City County CSV from **simplemaps.com**, load it in DataFrame.

*Challenges and Decisions:*

*Our ideal data would have been by city since our Target location data was listed by city. However, given the time constraint, we decided to join our data at County level, a field the census data has.*

*The most ideal extract would have been an API since this would continue to avail us with data as opposed to a CSV that is static. We went back and forth deciding if an API was feasible and decided that for the foreseeable future, a CSV should be sufficient in providing census data. We would need more real time data with other data sets.*

# Transform

# Cleaning Target Location Data:

* + - Target State link consisted of both State Link and Map link, filtered out map links.
    - State link had //, that caused the link to open as file in splinter browser, also since it’s a secure website it has to have https, so appended https.
    - Used **Regular Expression (RegEx)** to clean, segregate State Abbrevation from Address Data, since the address data came with City, State Abbr, and Phone number concatenated into single string.

# Cleaning Census Data:

* + - We loaded the raw cense data into jupter notebook and used Pandas as well SqAlchemy to create data frames and clean up our data.

# Selection

* + - The census data was trimmed from 37 columns to 11 columns. We chose the 11 columns because we are primarily interested in using the demographics to get to insights about Target’s location.

# Joining

* + - We joined Target data, with City County data to get Target Location by County data.
    - Then we joined our Target Location data with the Census data, both at County Level.

# Filtering

* + - There wasn’t any filtering done for the Census data. However, we checked our data to make sure we had the correct datatypes for analyzing, as well as making sure we had no missing cells.

# Aggregating

* + - We aggregated all the columns since the raw census data had the counties split up by county political demarcations. However, since we couldn’t not determine the criteria of the splits, we aggregated the split counties and summed up the demographics to get to one row per county in each State.

# Loading:

# Relational Database

* + - We used SqAlchemy to load our data into Postgres.
    - We loaded both the transformed and raw census data, Target Location, and the Analysis into tables in postgres.

# Jupyter Notebook:

# ScrapeStoreLocation.ipynb

* + - Function: Extract, Transform and Load Target Location data, City county data.
    - Output: target\_city\_raw, target\_city\_location, target\_count\_county

# Census\_Race\_Ethnicity.ipynb

* + - Function: Extract, Transform and Load Census CSV and get Target Store Count by Gender, Race, Ethnicity.
    - Output: census\_df, census\_race\_ethnicity, target\_race\_ethnicity\_stats

# Census\_Income\_Commute.ipynb

* + - Function: Get Target Store Count by Income, Mean Commute, Employment.
    - Output: target\_income\_stats