ETL Project Report

Team 1

Team Members: Allen Broce, Anthony Paige, Greg Atkinson, Stephanie Zhu

Data source: Kaggle

Topic: Starbucks Locations vs Income Level by Zip Codes (US only)

Datasets:

- File_1: Starbucks Locations Worldwide.csv (starbucks_data)
  - Link: https://www.kaggle.com/starbucks/store-locations
- File_2: US Household Income Statistics.csv (income_data)
  - Link: https://www.kaggle.com/goldenoakresearch/us-household-income-stats-geo-locations?select=kaggle_income.csv

Part I: Extract

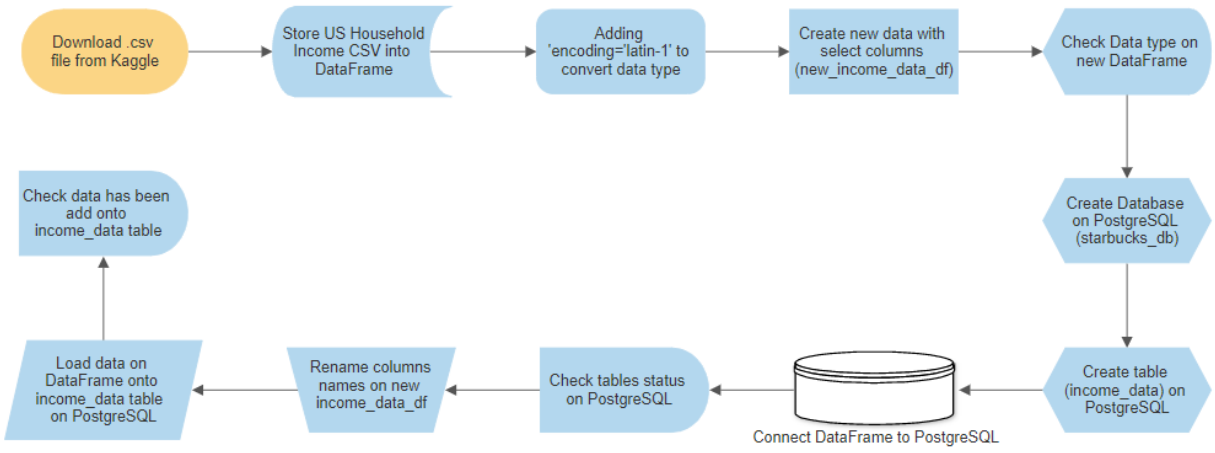
File_1: Starbucks Locations Worldwide.csv
Process:

The data for Starbucks locations was sourced from Kaggle at https://www.kaggle.com/starbucks/store-locations, and was in CSV format. It contained all Starbucks locations worldwide, so the first step in transforming it was reducing the data to US-only locations. The next step was dropping columns that seemed irrelevant, like brand (Starbucks for all? Maybe named otherwise for international locations?), phone number and time zone.

The data that we would be joining on with the income data, zip code, was the biggest obstacle for cleaning the Starbucks data. Most zip codes were the standard five characters, while some were nine characters, with the extra four digits on the right ("ZIP+4", for sector and segment). Other zip codes were only four digits because they were missing the leading zero, like 6117 for West Hartford, CT. Data Science Made Simple (https://www.datasciencemadesimple.com) had code for adding a leading 0 to these zip codes, as noted in the Jupyter notebook. Dropping rows with NA values for zip code was tried, but the data had as many rows afterwards as it did before, so there were no rows with NA values for the key data.

File_2: US Household Income Statistics.csv (income_data)

Process:



Part II: Transform

- Starbucks_data:
  - ✓ Country → Select 'US' data only
  - ✓ Selected columns → 'Store Number', 'Street Address', 'City', 'State/Province', 'Country', 'Postcode', 'Longitude', 'Latitude'
  - ✓ Rename selected columns
  - ✓ Zip Code → Remove four extra digits & extract first five digits on Zip Codes
  - ✓ Zip Code → Some zip codes start with 0, adding a leading zero back to zip codes
- Income_data:
  - ✓ Encoding of csv file → Update encoding to 'latin-1'
  - ✓ Selected columns → "id", "State_Name", "City", "Zip_Code", "Lat", "Lon", "Median", "Stdev"
  - ✓ Rename selected columns

Part III: Load

- Final database: starbucks_db
- Tables:
  - ✓ starbucks_data
  - ✓ income_data
- Targeted data to be retrieved:
  - ✓ Joining starbcuks_data & income_data with zip codes
  - ✓ Maximum & Minimum median income
  - ✓ Total number of Starbucks locations for each State
  - ✓ Total number of Starbucks locations with Maximum Median Income for each state (300,000)
  - ✓ Total number of Starbucks locations with Minimum Median Income for each state (0)

Challenges:

1. We have found that Identifying data source with correlation could be most challenging for our team;
2. The Zip Codes from stabucks_data.csv file was in the 'Zip + 4' format. We had to drop the last 4 digits on all zip codes for the US;
3. Some Zip Codes started with digit 0, and we had to add a Leading 0 back onto the Zip codes;
4. Some of the data types we selected incorrectly, and we had to cross reference with the csv file;
5. Using lowercase with columns is critical with PostgreSQL. We hit a snag using upper case and it took us a while to figure that one out.

Queries:

```sql
-- Create table structure for starbucks_data
CREATE TABLE starbucks_data (
store_number varchar PRIMARY KEY,
address TEXT,
city TEXT,
state varchar,
country TEXT,
zipcode varchar,
longitude float,
latitude float
);
```

```sql
-- Create table structure for income_data
CREATE TABLE income_data(
id INT PRIMARY KEY,
state_name TEXT,
city TEXT,
zip_code varchar,
lat float,
lon float,
median float,
stdev float
);
```

```sql
-- Rename column in income_data table
ALTER TABLE income_data
RENAME COLUMN median TO median_income

-- Joins starbucks_data & income_data tables (Income DESC)
SELECT S.state,
S.city,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code
ORDER BY 3 DESC
```

```sql
-- Joins starbucks_data & income_data tables (Income ASC)
SELECT S.state,
S.city,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code
ORDER BY 3

-- Display Minimum & Maximum Median_income
SELECT MIN(median_income) AS Min_income,
MAX(median_income) AS Max_income
FROM income_data
```

```sql
-- Display Maxmium median_income for each state
SELECT A.state,
max(median_income) AS Max_Income
FROM(
SELECT S.state,
S.city,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code) A
GROUP BY A.state
ORDER BY 1
```

```sql
-- Display Minmium median_income for each state
SELECT A.state,
MIN(median_income) AS Min_Income
FROM(
SELECT S.state,
S.city,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code) A
GROUP BY A.state
ORDER BY 1
```

```sql
--Display Starbucks locations for each state
SELECT state,
COUNT(store_number) AS Number_of_Starbucks
FROM starbucks_data
GROUP BY state
ORDER BY 1

-- Display total Number of Median Income for each state
SELECT A.state,
COUNT(median_income) AS Number_of_Income
FROM(
SELECT S.state,
S.city,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code) A
GROUP BY A.state
ORDER BY 1
```

```sql
-- Display Total Number of Maxmium Median_Income for each state
SELECT  A.state,
COUNT(DISTINCT A.zip_code) AS Total_Locations,
A.median_income
FROM(
SELECT S.state,
S.city,
I.zip_code,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code
WHERE I.median_income >= 300000) A
GROUP BY  A.state, A.median_income
ORDER BY 2 DESC
```

```sql
-- Display Total Number of Minmium Median_Income for each state
SELECT  A.state,
COUNT(DISTINCT A.zip_code) AS Total_Locations,
A.median_income
FROM(
SELECT S.state,
S.city,
I.zip_code,
I.median_income,
I.stdev
FROM starbucks_data AS S INNER JOIN
income_data AS I
ON S.zipcode = I.zip_code
WHERE I.median_income = 0) A
GROUP BY  A.state, A.median_income
ORDER BY 2 DESC
```