

# 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](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:

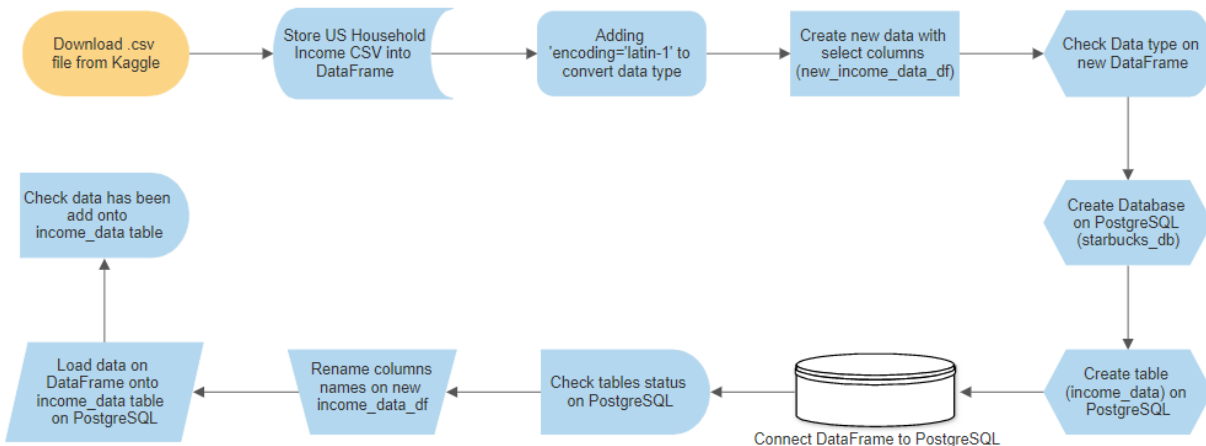
The data for Starbucks locations was sourced from Kaggle at <https://www.kaggle.com/starbucks/store-locations>, and was in CSV format. It contained all Starbucks locations worldwide, so the first step in transforming it was reducing the data to US-only locations. The next step was dropping columns that 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.

# ETL Project Report

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", "Stddev"
  - ✓ Rename selected columns

## Part III: Load

- Final database: starbucks\_db
- Tables:
  - ✓ starbucks\_data
  - ✓ income\_data
- Targeted data to be retrieved:
  - ✓ Joining starbucks\_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)

# ETL Project Report

## 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:

```
1  -- Create table structure for starbucks_data
2  CREATE TABLE starbucks_data (
3  store_number varchar PRIMARY KEY,
4  address TEXT,
5  city TEXT,
6  state varchar,
7  country TEXT,
8  zipcode varchar,
9  longitude float,
10 latitude float
11 );
```

```
35 -- Rename column in income_data table
36 ALTER TABLE income_data
37 RENAME COLUMN median TO median_income
38
39 -- Joins starbucks_data & income_data tables (Income DESC)
40 SELECT S.state,
41 S.city,
42 I.median_income,
43 I.stdev
44 FROM starbucks_data AS S INNER JOIN
45 income_data AS I
46 ON S.zipcode = I.zip_code
47 ORDER BY 3 DESC
```

```
64 -- Display Maxmium median_income for each state
65 SELECT A.state,
66 max(median_income) AS Max_Income
67 FROM(
68 SELECT S.state,
69 S.city,
70 I.median_income,
71 I.stdev
72 FROM starbucks_data AS S INNER JOIN
73 income_data AS I
74 ON S.zipcode = I.zip_code) A
75 GROUP BY A.state
76 ORDER BY 1
```

```
13 -- Create table structure for income_data
14 CREATE TABLE income_data(
15 id INT PRIMARY KEY,
16 state_name TEXT,
17 city TEXT,
18 zip_code varchar,
19 lat float,
20 lon float,
21 median float,
22 stdev float
23 );
```

```
49 -- Joins starbucks_data & income_data tables (Income ASC)
50 SELECT S.state,
51 S.city,
52 I.median_income,
53 I.stdev
54 FROM starbucks_data AS S INNER JOIN
55 income_data AS I
56 ON S.zipcode = I.zip_code
57 ORDER BY 3
58
59 -- Display Minimum & Maximum Median_income
60 SELECT MIN(median_income) AS Min_income,
61 MAX(median_income) AS Max_income
62 FROM income_data
```

```
78 -- Display Minmium median_income for each state
79 SELECT A.state,
80 MIN(median_income) AS Min_Income
81 FROM(
82 SELECT S.state,
83 S.city,
84 I.median_income,
85 I.stdev
86 FROM starbucks_data AS S INNER JOIN
87 income_data AS I
88 ON S.zipcode = I.zip_code) A
89 GROUP BY A.state
90 ORDER BY 1
```

# ETL Project Report

```
92 --Display Starbucks locations for each state
93 SELECT state,
94 COUNT(store_number) AS Number_of_Starbucks
95 FROM starbucks_data
96 GROUP BY state
97 ORDER BY 1
98
99 -- Display total Number of Median Income for each state
100 SELECT A.state,
101 COUNT(median_income) AS Number_of_Income
102 FROM(
103 SELECT S.state,
104 S.city,
105 I.median_income,
106 I.stdev
107 FROM starbucks_data AS S INNER JOIN
108 income_data AS I
109 ON S.zipcode = I.zip_code) A
110 GROUP BY A.state
111 ORDER BY 1

113 -- Display Total Number of Maxmium Median_Income for each state
114 SELECT A.state,
115 COUNT(DISTINCT A.zip_code) AS Total_Locations,
116 A.median_income
117 FROM(
118 SELECT S.state,
119 S.city,
120 I.zip_code,
121 I.median_income,
122 I.stdev
123 FROM starbucks_data AS S INNER JOIN
124 income_data AS I
125 ON S.zipcode = I.zip_code
126 WHERE I.median_income >= 300000) A
127 GROUP BY A.state, A.median_income
128 ORDER BY 2 DESC

130 -- Display Total Number of Minmium Median_Income for each state
131 SELECT A.state,
132 COUNT(DISTINCT A.zip_code) AS Total_Locations,
133 A.median_income
134 FROM(
135 SELECT S.state,
136 S.city,
137 I.zip_code,
138 I.median_income,
139 I.stdev
140 FROM starbucks_data AS S INNER JOIN
141 income_data AS I
142 ON S.zipcode = I.zip_code
143 WHERE I.median_income = 0) A
144 GROUP BY A.state, A.median_income
145 ORDER BY 2 DESC
```