Starbucks and Happiness Report 2019

OCTOBER 19

Brian Lee, Don Leshem, Ellaine Ho
Columbia University Data Analytics

Executive Summary

Starbucks Locations and World Happiness Report

The purpose of this project is to create a link between the World Happiness Report with Starbucks locations around the globe, aggregating the data from both reports to create a database that will host locations by country, as well as the yearly Happiness Report for each corresponding country to examine the correlation between happiness around the country and number of locations.

Data Sources

To make this possible, we used two types of datasets obtained from Kaggle.com. The first dataset that we utilized was a World Happiness Report in 2015, 2016, and 2017. We also utilized a dataset that illustrates all Starbucks locations globally. In addition, we also created a file for country codes in corresponding ISO Alpha-2 format to create a link between two datasets. The datasets that we obtained resulted in 5 CSV files that we will use to extract and transform data.

Data Format

Starbucks Location	World Happiness Report	ISO Alpha-2 Country Code
Brand	Country	Name
Store Number	Region	Code
Store Name	Happiness Rank	
Ownership Type	Happiness Score	
Street Address	Standard Error	
City	Economy (GPD per Capita)	
State/Province	Family	
Country	Health (Life Expectancy)	
Postcode	Freedom	
Phone Number	Trust (Government Corruption)	
Timezone	Dystopia Residual	
Longitutde		
Latitude		

Data Extraction and Transformation

In observing the Starbucks Location and World Happiness Report dataset, we determined that the common link will be the country location. However, we needed to create a common link between both datasets since the country location illustrated within the Starbucks Location file is in ISO Alpha-2 format, while the World Happiness Report illustrates country locations in full, unabbreviated text format.

Starbucks Location

For this dataset, we first dropped any unnecessary data columns that we do not plan on using within the final database. This mostly included columns that contained missing data, or city or longitude/latitude specific information that are not relevant to our project purpose. Additionally, we utilized the ISO Alpha-2 Country Code to link the country code listed on the Starbucks Location file to the corresponding unabbreviated country name. We also renamed all columns in lowercase and remove spaces in the format to prevent any database formatting errors. The final format included 6 columns that will be linked based on the unabbreviated country name.

Revised
brand
store_number
store_name
state_province
country_code
country

World Happiness Report

For this dataset, we loaded reports from 3 years of data that we hoped to transform into the database. We dropped any unnecessary columns, appended a year specific column to all corresponding data frames, and then renamed the columns to prevent any database format errors. After all 3 separate data frames have been cleaned, we concatenated all three together into a joined data frame that we will use for the final database. For consistency purposes, we also renamed some country names to the corresponding Alpha-2 ISO format to prevent any errors when linking to our final database. The final data format included 11 revised columns that will be linked based on the unabbreviated country name.

Revised
country_code
happiness_rank
happiness_score
economy_gdp_per_capita
family
health_life_expectancy
freedom
trust_government_corruption
generostiry
dystopia_residual
year

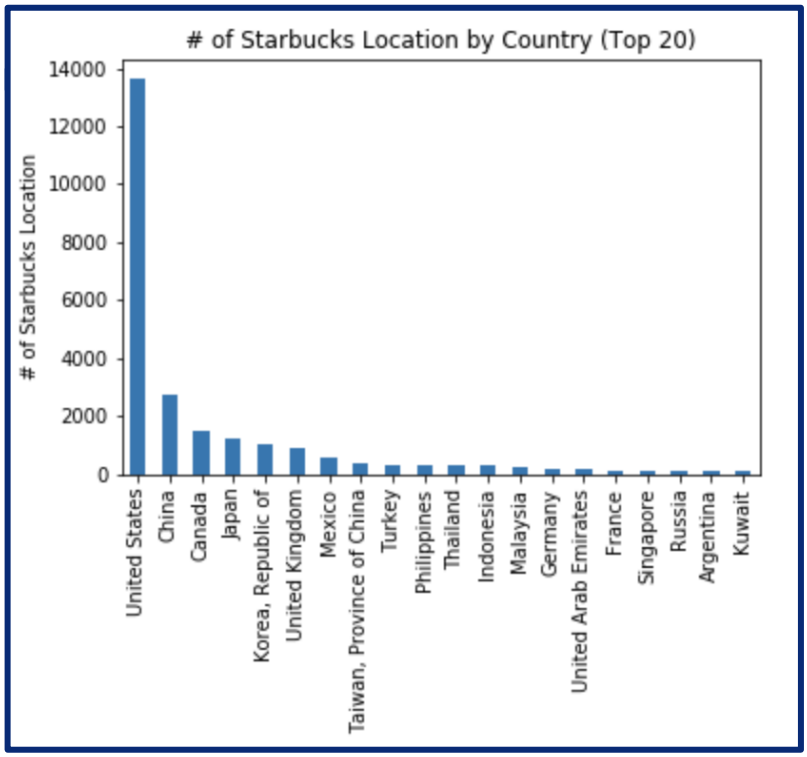
Country Codes

Aggregating from both the Starbucks Location and World Happiness Report, we created a country list that will be used for the final database. Because there were inconsistencies with the country names (ex: Hong Kong vs Hong Kong S.A.R), we exported the country list from both reports to a separate CSV file, compared for differences, and made changes in our respective data frames before creating the database link. The final list of country codes included the name of the country that will be used as the primary key.

Revised country country_code

Visualization and Normalization

To ensure that our data across both datasets are consistent before we generate a database link, we created a few visualizations and checks:



	country	happiness_rank
0	Norway	1
1	Denmark	2
2	Iceland	3
3	Switzerland	4
4	Finland	5
5	Netherlands	6
6	Canada	7
7	New Zealand	8
8	Sweden	9
9	Australia	10
10	Israel	11
11	Costa Rica	12
12	Austria	13
13	United States	14
14	Ireland	15
15	Germany	16
16	Belgium	17
17	Luxembourg	18
18	United Kingdom	19
19	Chile	20

Data Dictionary

Starbucks Location

Column Name	Data Type	Example
brand	String	Starbucks
store_number	String	47370-257954
store_name	String	Meritxell, 96
state_province	String	7/AJ
country_code	String	AD

World Happiness Report (Joined)

Column Name	Data Type	Example
country_code	String	AD
happiness_rank	String	1
happiness_score	Float	7.537
economy_gdp_per_capita	Float	1.6289
family	Float	1.533524
health_life_expectancy	Float	0.796667
freedom	Float	0.635423
trust_government_corruption	Float	0.635423
generosity	Float	0.315964
dystopia_residual	Float	0.362012
year	Integer	2017

Country Code

Column Name	Data Type	Example
Country	String	Andorra
Country_code	String	AD

Database Link ERD Visualization

Database Link

We chose to utilize SQLAlchemy to generate a connection to our PostgreSQL database. Before we append the data onto our database, we first created a database on PostgreSQL called “Country-Happiness”. Next, we created a schema file to create tables within Postgres. Finally, we created a connection through SQLAlchemy to directly append all dataframes to our corresponding tables:

- country_code (primary key)
- starbucks_locations
- happiness

The country_code table was selected as the primary key to create links between both datasets.

SQL Schema (before database connection)

```
DROP TABLE IF EXISTS Starbucks_Locations;
DROP TABLE IF EXISTS Happiness;
DROP TABLE IF EXISTS country_code;

create table Country_code(
Country VARCHAR,
Country_Code VARCHAR,
primary key (Country_Code`)
);
create table Starbucks_Locations(
Brand VARCHAR,
Store_Number VARCHAR,
Store_Name VARCHAR,
State_Province VARCHAR,
Country_Code VARCHAR,
foreign key(Country_Code) references Country_code (Country_Code)
);
create table Happiness(
Country_Code VARCHAR,
Happiness_Rank float,
Happiness_Score float,
Economy_GDP_per_Capita float,
Family float,
Health_Life_Expectancy float,
Freedom float,
Trust_Government_Corruption float,
Generosity float,
Dystopia_Residual float,
Year real,
foreign key(Country_Code) references Country_code (Country_Code)
);
```

Queries (after database connection)

1) Collect all data from Starbucks_Locations

```
# a view of Starbucks_Locations table
dataframe = psql.read_sql("SELECT * FROM Starbucks_Locations", engine)
dataframe.head()
```

	brand	store_number	store_name	state_province	country_code	country
0	Starbucks	47370-257954	Meritxell, 96	7	AD	Andorra
1	Starbucks	22331-212325	Ajman Drive Thru	AJ	AE	United Arab Emirates
2	Starbucks	47089-256771	Dana Mall	AJ	AE	United Arab Emirates
3	Starbucks	22126-218024	Twofour 54	AZ	AE	United Arab Emirates
4	Starbucks	17127-178586	Al Ain Tower	AZ	AE	United Arab Emirates

2) Collect all data from Happiness

```
# a view of Happiness table
dataframe = psql.read_sql("SELECT * FROM Happiness", engine)
dataframe.head()
```

	country	happiness_rank	happiness_score	economy_gdp_per_capita	family	health_life_expectancy	freedom	trust_government_corruption	generosity	d
0	Switzerland	1.0	7.587	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	
1	Iceland	2.0	7.561	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	
2	Denmark	3.0	7.527	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	
3	Norway	4.0	7.522	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	
4	Canada	5.0	7.427	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	

Queries (after database connection) cont.

3) Collect list of all countries

```
# a view of country_code table
dataframe = psql.read_sql("SELECT * FROM country_code", engine)
dataframe.head()
```

	country
0	Afghanistan
1	Albania
2	Algeria
3	Andorra
4	Angola

4) Collect a list of countries and the number of Starbucks locations in each country

```
# number of Starbucks stores by country
dataframe = psql.read_sql("SELECT country, count(*) as counter FROM Starbucks_Locations group by country", engine)
dataframe.head()
```

	country	counter
0	Indonesia	268
1	Brunei Darussalam	5
2	Luxembourg	2
3	Czech Republic	28
4	Sweden	18