EXTRACT

Our first data source was *Our World in Data*, which provided us with the total number of tourist arrivals in various countries around the world. The data set looks at total number of people that arrived in a given country for the purpose of tourism each year from 1995 to 2015.

Our second data source is *World Bank*, which provided GDP and population data for each country by year.

Both of these data sources were exported from the internet into csv files. We then read the csv files into pandas and began to transform the data.

TRANSFORM

To transform the *World Bank* data, we started by dropping ‘na’ from the dataset. Then we transformed column names to make it easier to understand what a reader would be looking at. During this step we also removed extra data from the csv download that we didn’t need in our SQL database. The file originally had over 50 data fields for each country. We filtered out all data other than GDP and Population and created dataframes for each of those fields. Lastly, we dropped the columns that weren’t relevant to our data and merged the ‘Total Population’ and ‘GDP’ dataframes together.

To transform the *Our World in Data* data, we filtered out all of the years from 1995 – 2014 to only include the 2015 data, since that was the year we had population and GDP data for. Then we merged the tourism arrival data from *Our World in Data* with the GDP and Population data from *World Bank*.

LOAD

To load the data, first we created a database and tables for the data in Postgres.

To load the data into Postgres, we followed the steps below:

1. Import dependencies:
   1. from sqlalchemy import create\_engine
2. Create connection and engine:
   1. conn = "postgres:{password}@localhost:5432/tourism"
   2. engine = create\_engine(f'postgresql://{conn}')
3. Pull table names to make sure tables exist in Postgres:
   1. engine.table\_names()
4. Merge transformed data in SQL:
   1. merge.to\_sql(name='merged', con=engine, if\_exists='append', index=True)
   2. tourism\_cleaned.to\_sql(name='tourism\_2015', con=engine, if\_exists='append', index=True)
5. Perform query to make sure data has been loaded into Postgres tables
   1. pd.read\_sql\_query('select \* from tourism\_2015', con=engine).head()