**Farid Hanna**

**ETL Project**

**03/16/2020**

**Project Requirements:** Use 2 or more sources of data for extraction, transformation and loading into a database or tables/collections.

**Extract Data: -**

**First Source of Data:** I extracted data in Excel format from the below website that attempt to measure happiness by country using different methodology.

<http://happyplanetindex.org/> “The **Happy Planet Index** measures what matters: sustainable wellbeing for all. It tells us how well nations are doing at achieving long, happy, sustainable lives.”

Data Location: <http://happyplanetindex.org/resources>

**Second Source of Data:** I extracted data in csv format from the below website that shows international arrival numbers by country for the period from 1960 to 2019. I chose the international arrival numbers indicator to be able if required to perform an analysis to test if countries that are high in the happy scale index receive more travelers per year vs those that are in the low happiness scale.

Data Location: https://databank.worldbank.org/source/world-development-indicators#

**Transform Data: -**

**First Data Source Transformation:** The first data set was in Excel format and sheet and started on the second column and the data started on the 6th row. In addition, the footer contained additional information that was not part of the index or table. The python scripted had to skip the first 5 rows and 1st column as well as ignore the footer information. Also, the field names contained special characters and spaces that had to be eliminated and renamed. Below is the script in python;

#Read files and create df from "Happy Planet Index 2016" (Countries Grouped by region)

h\_path= "hpi-data-2016.xlsx"

h\_read = pd.DataFrame(pd.read\_excel(h\_path, sheet\_name='Complete HPI data',skiprows=5,skip\_footer=17))

h\_read= h\_read.drop('Unnamed: 0', axis=1)

**First Data Source Transformation:** The second data set was in csv format. The data contained columns in a range of years that were empty values. Instead of dropping those columns, I created a new data frame that contains only the relevant fields. The first 4 rows were skipped to arrive at the first columns of the data set. Also, column names were imported as floats and were converted in a rename column script.

#Read files and create df from "Data Source World Development Indicators" (All arrival of travellers by Country by Year)

t\_path= "API\_ST.INT.ARVL\_DS2\_en\_csv\_v2\_823405.csv"

t\_read = pd.read\_csv(t\_path, delimiter=',',skiprows=4)

**Merge the 2 data frames into one on the Country key**

#Merge the Two Data Frames based Country Key

country\_happy\_n\_tourism =pd.merge(left=Country\_tourism, right=Country\_indicators, left\_on='Country', right\_on='Country')

**Load Data: -**

* Connect to the Database “ETL\_Project” and load the merged Data Frame “country\_data”

#Create Engine and connection to Database

engine = create\_engine('postgres://postgres:postgres@localhost:5432/ETL\_Project')

conn = engine.connect()

country.to\_sql(name='country\_happy\_n\_tourism', con=engine, if\_exists='append', index=False)

* Check that the table has been loaded to postgres SQL database.

#Check table in sql

pd.read\_sql\_query('select \* from country\_happy\_n\_tourism', con=engine).head()