**Bridget O’Connell, Saki Amagai, and Shabeehe Rizvi**

**NU Data Science Bootcamp 2020**

**ETL Project**

**Final Report**

Extract

We extracted multiple reports from two sources:

* World Happiness Reports from Kaggle
  + Five separate csv files for years 2015 through 2019.
  + Each report includes overall rank, country or region name, total score, GDP per capita score, social support score, healthy life expectancy score, freedom to make life choices score, generosity score, and perceptions of corruption score.
  + Each report was downloadable and free.
  + Link: <https://www.kaggle.com/unsdsn/world-happiness/data?select=2019.csv>
* Gender Statistics from The World Bank
  + Within The World Bank’s Data Bank there is a plethora of information under Gender Statistics. Info can be filtered by numerous factors: country or geographic regions, years (1960 to YTD 2020), and series which includes both broad and specific breakdowns of male/female rights, activities, and abilities (example, can a woman apply for a passport, open a bank account, or obtain judgment of divorce).
  + We choose to pull series on female population, female % of the labor force, and female % that complete secondary school 2015 to 2019.
  + Each report was downloadable and free.
  + Link: <https://databank.worldbank.org/reports.aspx?source=283&series=SG.DMK.HLTH.FN.ZS>
* As both reports listed country name, we felt using country abbreviations would be cleaner and more appropriate to use as a primary key once data was cleaned and move to a relational database. We found a csv file online listing country name and abbreviation.

Transform

Cleaning up the data was done in two Jupyter notebooks using pandas.

* World Happiness Reports
  + After reading in the files, we did the following to each:
    - Removed unnecessary columns
    - Renamed the kept columns to be consistent across all files
    - Added a column for the year
    - Merged the data with the country abbreviation files to bring in abbreviation which we called “code”
    - Searched for null values; while we tried to avoid manual manipulations, there were a handful of instances where a country had one name in four years and a slightly different name in the fifth year or where the country name in the happiness file varied slightly from the country name in the abbreviations file. Using a code to find the null values across all rows and columns helped me identify where I needed to take a closer look and see if a manual manipulation was needed.
  + Reorganized columns to be in a specific order left to right
  + Used pd.concat to combine all files into one
  + Added a column “happy\_id” to combine the year column and country code/abbreviation with an underscore to create a unique identifier to be used as the primary key. Example, “2015\_US”, “2016\_US” and so on.
  + Exported the data to a csv
* Gender Statistics
  + After reading in the files, I did the following to each:
    - Removed unnecessary columns
    - Renamed the kept columns to be consistent across all files
    - Added a column for the year
    - Merged the data with the country abbreviation files to bring in abbreviation which we called “code” – this was to eliminate any “countries” that would not qualify as countries.
    - Removed any duplicates in the country code and ensured that there were no null values in the code
  + Created separate DataFrames for 2015,2016,2017,2018 and 2019 for the female education, population and labor force data set. The data sets were merged by year.
  + The merged data sets were then concatenated, using pd.concat
  + Exported the data to a csv.
* For ease, we read in the finalized gender statistics data into the world happiness notebook versus copying all the coding used to clean the data. We will submit both notebooks for review.
  + Once gender statistics was read in, we did some column renaming and changed the data type of the year to a string in order to create the unique identifier below.
  + Added a column “fem\_id” to combine the year column and country code/abbreviation with an underscore to create a unique identifier to be used as the primary key. Example, “2015\_US”, “2016\_US” and so on.

Load

We chose to use a relational database, Postgres, because if we were to continue analyzing this data we’d like to see if there is a relationship between a country’s happiness rank (or one of the components of its’ rank) versus female involvement in labor and education. In pgAdmin, we created a “happiness\_db” and created tables “world\_happy” and “fem\_stats”. In the world happiness Jupyter notebook, we created an engine and connection, pushed the final data sets to SQL, and confirmed they pushed correctly by running a pd.read\_sql\_query in Jupyter and a select all query in Postgres.