Group 7- ETL Project Report

Group Members

Tahsin Rahman

Uzair Khan

Sean Dee

Laurie Walker

Overview

This Extract Transform and Load project will seek to prepare a dataset, allowing us to examine the success of countries in the Summer Olympics from 1960 until the present, as measured by total medals won at each Olympic Games. In particular, it will facilitate an analysis of key variables that contribute to a country’s success in an individual Olympic games.

 In particular, we want to examine how a country’s wealth (as measured by GDP), its population size, and a home advantage (whether a country is hosting the games) might affect the number of medals it wins at a single Olympic games.

Extract

Our project draws on two main data sources:

Our first main source will be a historic dataset on the Olympics, available on [Kaggle](https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results?select=athlete_events.csv). This dataset provides information on each athlete at each event in each Olympic games, such as the athlete's country, event, medal for that event, year of the event, host city of Olympic games. In the interests of time, we downloaded this document directly from Kaggle as a CSV file.

Our second source of information will be from the World Bank website, which will provide time series data on GDP and population from 1960 until 2016.

The World Bank population data can be found at:

<https://data.worldbank.org/indicator/SP.POP.TOTL>

The World Bank GDP data can be found here:

<https://data.worldbank.org/indicator/NY.GDP.MKTP.CD>

As in the case of the Olympic data, this was downloaded directly from the website as a CSV.

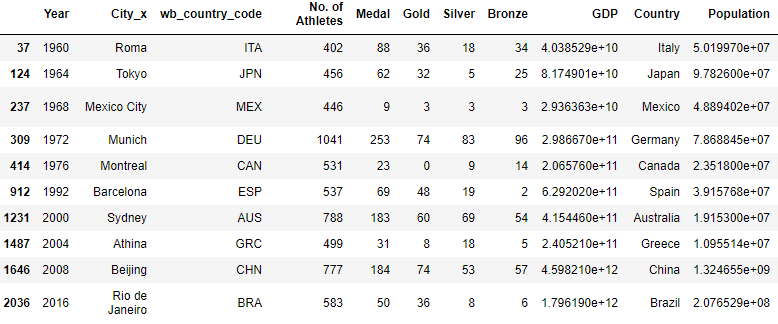
Our assumption was that the World Bank data would be accurate and complete, and this was generally true, although there some missing observations and data for some countries (notably Taiwan), was not available.

Likewise, the Olympic data appears fairly comprehensive and complete. An issue we anticipated was the changes of countries’ names and arrangements over time- for example East Germany (a major winner of Olympic medals in its own right) no longer exists while other countries have gained independence such as former Soviet states like Georgia, Armenia, Kazakhstan etc. These will pose difficulties in how we treat this historic data- likewise care will be needed when seeking to match the country data from the two different sources. Fortunately, the Olympic data whilst recording different country names over time, offers the data with a consistent National Olympic Committee (NOC) for all events- so that whilst it might record separate results for athletes of East and West Germany the NOC code for both is the modern, reunited Germany. The World Bank also seems to have done something similar, recording changing country names over time against a consistent country code (where that country exists)

A second issue was in matching the data from the Olympic dataset with the World Bank data. Both data sets had different country names and abbreviations e.g Great Britain-GBR compared to the United Kingdom-UK. In the end, we created a single master-list to match the two as a CSV file, which was a fairly tedious and manual process.

Transform

Our objective was to transform the data into a form so that each country, for each Olympic games, would have all observations such as medals won, GDP, population in that year and any other observations all in a single row, similar to the image below. All transformations were done in Jupyter Notebook, using pandas.

 In order to do this several transformations were applied.

For the Olympic data, we needed the data grouped by country and Olympic games before it could be aggregated. However, before we could aggregate by summing, information regarding medals won by individual athletes for each country at each Olympic games first had to be converted to a numeric value instead of a string. This was done by converting a medal result of “Gold” into a 1 in a separate column for gold medals, and replacing NaN values with 0’s olympic\_data['Gold'] = np.where(olympic\_data['Medal']=='Gold', 1, 0). We also included an aggregation providing the count of individual athletes representing a country at each Olympic game.

The World Bank data was first transformed by pivoting. In their original formats, both the GDP and population datasets have each year of observations as a column, with the most rightward column representing the final year, 2016. In order to have appropriate dimensions to merge with our Olympic data, we need to pivot this data so that each of the data for each country is a single row. This can be done using the [melt](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.melt.html?__cf_chl_captcha_tk__=657d9828448b137b7646ca7e08745f4a4582013f-1597316306-0-AevGGmEKPtWM1_bkDpABZ69QpHlKOriI-4omP9d7ld3okse3JsUXK_BSQV_1hfVpyCyutU3b7sDhWW8dkAmOAwYdZnjbXGFkedqBGD9EJVAHLv4YD-EYOU7vHqURApJtYHFRxzn1Rx29SdhtfFTe_vIQaU7s2BM4plATd_HXpuY-6q45wVGs1GgF-WREoGXYlGbUgHXUyfff04TED10B_ps32KvbC_H-IQH-56WIZJVWzJNRs-fvqmwYSXhYFfiXej9oLHvWb1v_RTMsUuWVNXCNEkZWqwvqhfSp-uqYGFB46flPU_8ZAQYa3jCf4NtRh4LSvhZhkkqIjaz20-6JQ6Z7jJ3IJZ4LL4j5sDuKnIeLR8TcrFOLUuPV_De8aBThBqCwokEIUNodp41ypFUE7p11SilWqJAbtQd8zLvYRF2Als03LuZUjMiu4rQKDhLWwqs9bERTk1_AaB9fgYEaPv18ZKhYvUGlNXEUJIAQP3mi2sMXtinIumjWLi3nr4AUo2wNyFfMVuC4kA84dn7ZzPyzrbzQSDAJizMa-F8JJSrg5CXXADv-8T90rut8kvFTjw) function df.melt(id\_vars=["Country Name", "Country Code", "Indicator Name", "Indicator Code"]. Finally, once the separate data sources are in the appropriate dimensions for a time series analysis, we will integrate the Olympic data with the World Bank data by merging them in pandas using a left merge, on the basis of common country code (from our created master-list) and year of the Olympic games.

To facilitate initial merges between the Olympic Games data and World Bank data, it was necessary to manually create a csv document, mapping the country name and code in the Olympic database (NOC, National Olympic Committee), to the country code in the World Bank database. Given the difference between country names, and the differences in country code between the different organisations, this was quite a tedious, time consuming and manual process. For speed, this was done in excel rather than in Pandas, and the final document read into Pandas as *noc\_country.csv*.

Before attempting merges or aggregations requiring numeric data types, we will also check the data types using the command df.dtypes. Individual columns can then be converted into the necessary data types as required.

After merging, we need to check whether there have been any failed merges or null values. In order to check for null or NaN values in columns, where a value was required, eg. for country code, we regularly ran the command print(df.isnull().sum()) . This provided a printout of the total number of NaN’s for each column in a DataFrame. Where this occurred in a critical column such as country code, we may inspect the individual countries affected and replace the NaN with the command np.where(olympic\_data['NOC']=='SGP', 'Singapore', olympic\_data['country']).

Finally, some countries and events and all their associated rows will be dropped from our dataframe, either because they are outside our analysis (earlier than 1960, Winter Olympic Games), or because there is no World Bank data (Taiwan, Palestine, Eritrea etc.)  This can be done using the command olympic\_data.drop(olympic\_data[olympic\_data["Year"]<1960].index, inplace=True).

Potential issues with data will be identified by running commands such as df.head(), df.dtypes,print(df.isnull().sum()) and df.shape.

Load

Our group decided to use PostgreSQL as *Relational Database.*

PostgreSQL offers much greater flexibility for storing data. Ability to connect to the database via Jupyter notebook. The tables in the database are as follows:

Tables:

Country - Year, Country\_Code, Country\_Name, Host

Medals - Year, Country\_Code, Medal, Gold, Silver, Bronze

Country\_Stats - Year, Country\_Code, GDP, Population

The relationship between the two tables will be joined by the country\_code and year. The Country\_Code is set as the primary key in the Country table in the database since this code is unique to the countries and it is also set as a foreign key in the Medals and Country\_Stats. We could query the data in the tables to find out information on Countries who won a medal, if they are the host country, and if the GDP and Population is a contributing factor to winning olympic medals.

Limitations and Discussion

As a practical consideration, this ETL Project relied on downloading CSV files directly from websites, and then reading them into a pandas data frame. This makes it a manual and time-consuming process to locate and download new data and refresh our ETL. A more sophisticated ETL project would depend on direct SQL queries or API connections, making the refresh process more efficient.

As noted previously, one issue with the data is that many of the countries no-longer exist. Whilst we have resolved this issue in our data by attributing observations to modern successor states (where possible), it does mean that other potential factors such as political orientation are obscured. Given our period spans most of the Cold War, a period where many countries (particularly Eastern Bloc countries) sought international prestige and athletic contests were imbued with ideological significance, this may be an important omission.