Technical Report:

Olympics ETL Project

Prepared by:

Mark Hirschenberger, Katie Schuman, and Jackson Yaggi

**Introduction:**

The goal of this project was to provide a comprehensive database compiled from various data publicly available online. This report outlines our process for extracting the data from its original sources, transforming it into forms more conducive to a future data analysis, and loading the data into one PostgreSQL database.

We sourced our data from three databases on kaggle.com. We designed our database tables to be able to perform analysis comparing instances of UFO sightings with Winter Olympics figure skating medal winners to see whether a correlation exists.

**Summary:**

The first step is to convert the three CSV's to Pandas DataFrames. After the conversion of the raw CSV files is complete, the DataFrames must be cleaned. Unnecessary columns must be deleted, and letter casing for the 'city' columns must all be consistent (upper case). Next, the "120 years of Olympic history: athletes and results" dataset will need to be filtered to display only figure skaters in the Winter Olympics. In the UFO table, the 'state/province' column must be renamed to 'state\_province', to avoid errors in SQL.

**Details of ETL:**

Extract:

We started off our project by deciding what kind of data we wanted to analyze. After doing some searching on kaggle, we found some Olympics data that looked too interesting to pass up. We also stumbled across a UFO dataset and had to bring that in as well. We did run into some trouble with an API key for data on SportRadar.com. We contacted an employee named Scott who let us know that we could only access the data for sales purposes. We ended up just using the Kaggle data, all of which was in .csv files. We were able to read in these files using a pd.read\_csv function in pandas and then converting that into a readable data frame for the user to see.

Transform:

Then we had to decide how to transform this data. The initial .csv file was fairly broad, including data from all athletes across multiple Olympic sports (Fig.1):

Fig.1

We decided that we did not want any losers in our Olympics data, so we filtered our Medals column so that anyone who did not win a medal was removed. We also decided that we only wanted figure skaters so we removed all athletes that were not Figure Skaters from the data set. Fig.2 below shows how these transformations were accomplished using pandas, and Fig.3 shows our final figure skating dataframe after all adjustments were made.

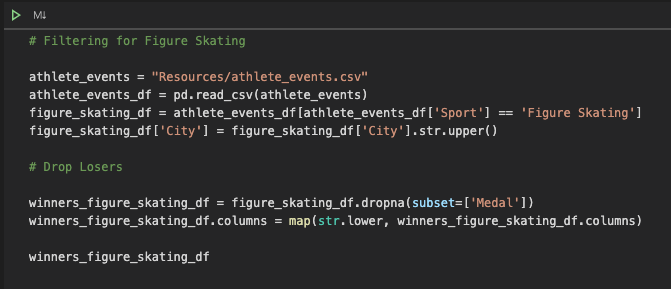
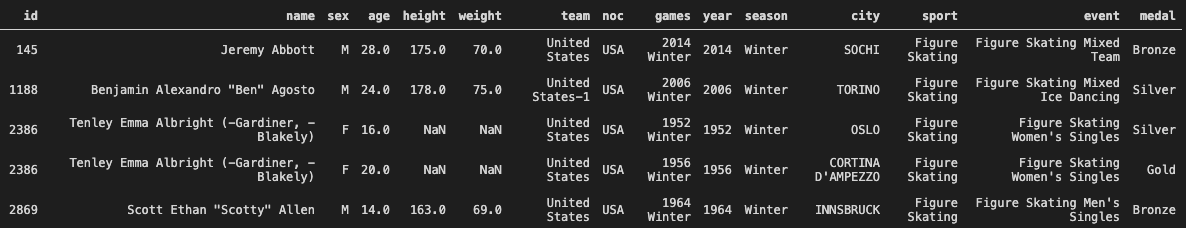
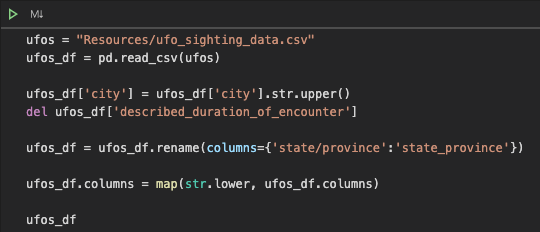
Fig.2

Fig.3

As you might expect, the dataset containing the UFO sightings was riddled with issues. We ended up removing a column due to the data inside being nearly the same as the column directly next to it. We had to rename a column due it not being compatible in Postgres. Finally, we wanted the City data to be consistent with one another so we used an .upper function so that all of the city names were capitalized (Fig.4).

Fig.4

Load:

We did run into some issues Loading the data into Postgres. The first issue we came across was trying to keep user information private. We ended up creating a user and password in Postgres keep from exposing any private information (Fig.5):



Fig.5

Next we set up all of our tables in Postgres by using a simple Create Table function and inserting our column names in here. Then we used the .to\_sql function to send the data from our jupyter notebook into Postgres (Fig.6):

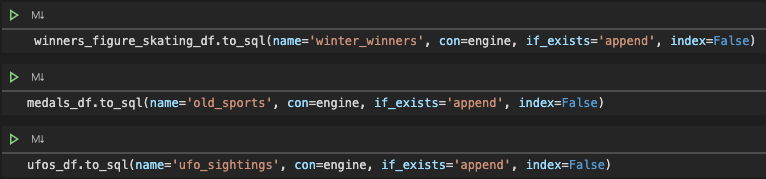


Fig.6

A big problem arose when we tried to push all of our tables due to the column names not lining up. We had to go back into our Pandas file and make all of our column names lowercase. Next the UFO data was giving us some issues. There were letters in the latitude and longitude column as well as non-numeric characters in the “length\_of\_encounter\_seconds” column. The fix for this was simply changing the data types in Postgres to TEXT instead of INT. After these changes, we were able to successfully see our manipulated tables in Postgres.

**Results and Discussion:**

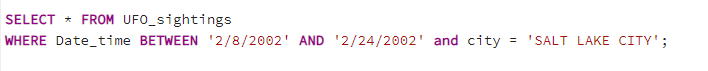
With the three datasets cleaned and filtered, a data analyst could explore and find more information surrounding the Winter Olympic Games over the years. By joining both the "Winter Olympics Figure Skating Winners" and "Olympic Sports and Medals, 1896-2014" and analyst could find more information on each figure skater's participation in the Olympic Games.

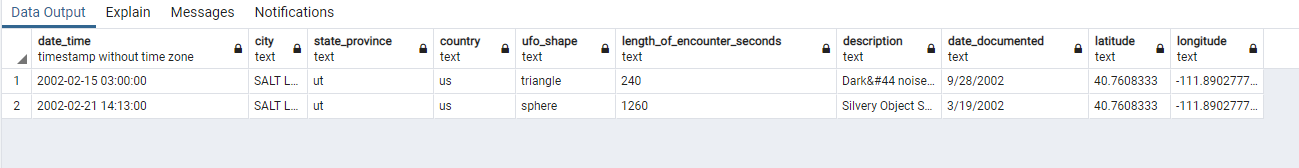
UFO researchers could join the "UFO Sightings around the world" table with the "Olympic Sports and Medals, 1896-2014" table on 'city' to determine how active UFOs were during the year in which a particular city hosted the Olympic Games.

If an analyst wanted to see the number of UFO sightings that occurred during each Olympic Games, they would join the Old\_sports and UFO\_sightings tables on 'city' and then on 'country.' To further drill down, they would need to set the 'Date\_time' between the start and end date for the Olympic Games that occurred in that city.

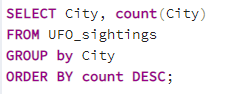
Sample Queries:

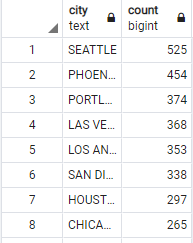
Query 1: UFO Sightings that took place during the 2002 Winter Olympic Games:





Query 2: UFO Sightings Count By city





**Conclusion and Acknowledgments:**

This project showed the value of pulling together multiple data sources in order to find unique insights. The ETL process is necessary for ensuring this data is easily accessible and formatted correctly for future data analysis.

Special thanks to Scott from SportRadar for redirecting our data gathering away from his organization’s API and into the magical world of UFO reporting. As always, thanks to Raul for answering our questions, and to the National UFO Reporting Center for maintaining a vigilant watch and thorough database.