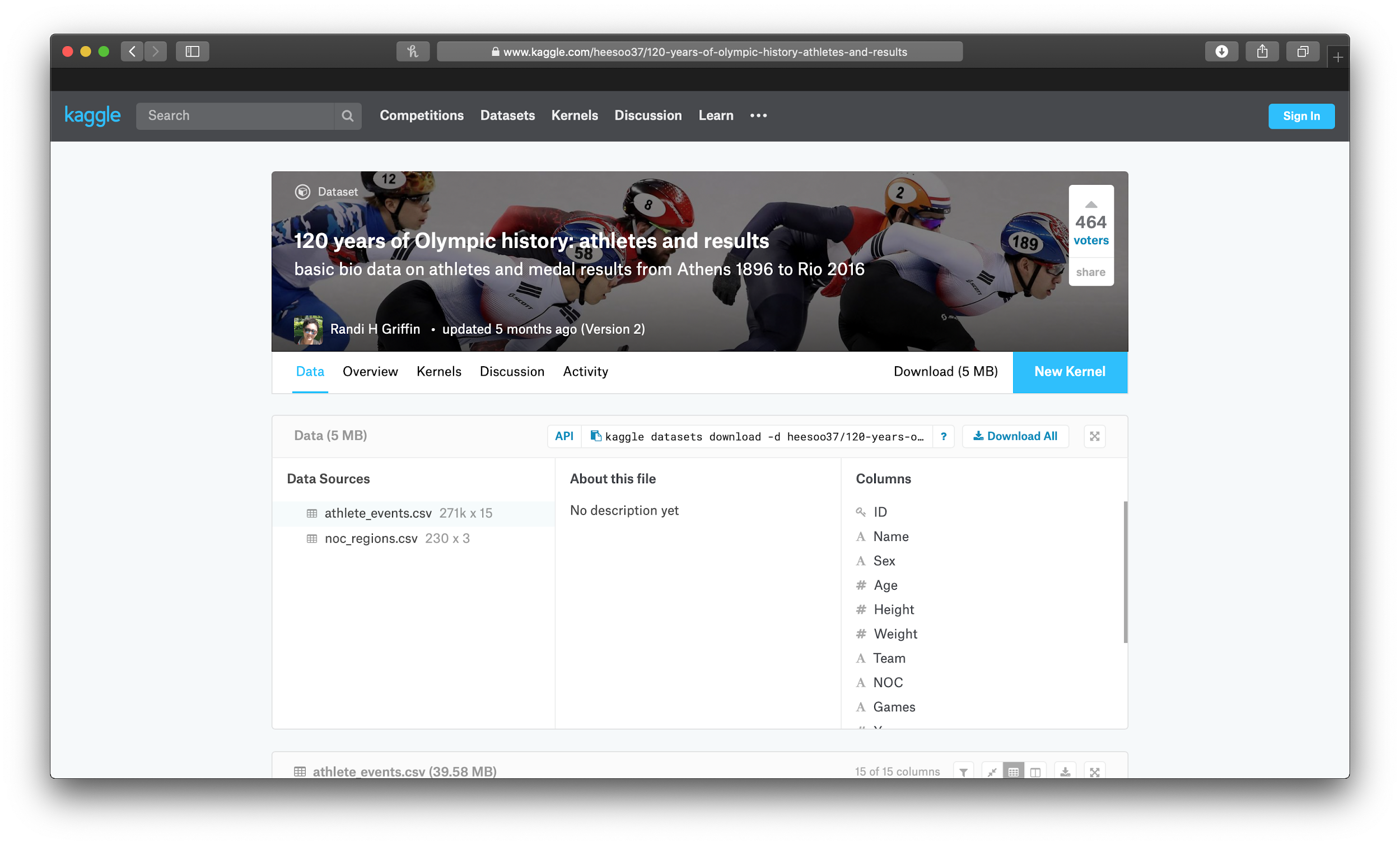
Harita Vellalath, Iqbal Singh, Rohit Rao, Jeremy Coy 11/3/2018

ETL Project Write-Up

Extract



Initial data extraction was performed using Kaggle. After finding various data sources available through the website, we determined that the datasets we wanted to analyze and manipulate would be the 120 years of Olympic history: athletes and results (<https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results>) and statistics from the World Bank Poverty Report (<https://www.kaggle.com/danofer/wb-poverty>), which is also available on Kaggle. Since Kaggle gave us the option to extract the data in the form of a CSV, we opted to download both datasets using this option for ease of use and versatility. After downloading the documents, we imported both into Python using Pandas.

Transform

In order to merge our two datasets together, we first needed to locate a common field we could use to link them together. Our first dataset listed the performance of different players in the olympics, such as their height/weight, what country they were from, and what kind of medal they won (if any). Our second dataset listed different country names and provided different metrics to measure poverty in those countries. The one field we found in both tables was “Country Name,” which is the field we used to merge the tables together.

After merging the tables together, we had to select the poverty metrics that were most descriptive/useful for the end users. We ultimately decided on the “MPI” and the “Headcount Ratio,” as we felt that these metrics painted the best picture of the financial state of the countries in the dataset.

We then decided to go one level deeper by counting up the number of gold medals won by each country in both datasets, in the event that our end users would be interested in finding the correlation between a country’s financial state and its performance in the olympics. How this information differs from the first table is that it provides a different definition of olympic success (total number of medals won vs. total number of gold medals won).

Our next table measured the total number of medals won by males vs. females. This table was simple, but it allowed end users to shift their focus from country demographics to gender demographics, in case they were interested in doing so.

Our next and final table gave a breakdown of the total number of medals won by males vs. females, by country. We were able to construct this table by aggregating our data by both “country” and by “gender”.

During the transformation process, our biggest challenge was putting ourselves in the shoes of our end users, by coming up with tables that provided information that could be used to answer a variety of questions. It was important that our tables answer questions, as opposed to simply providing data. It was also important that our tables flowed well into one another, in case our end users wanted to merge different tables together to answer any questions of their own.

Loading

To complete this step in the process, we first needed to create a database. We did so through mySQL Workbench. Because we had previously created dataframes of the tables that we used for analysis, we simply had to load those dataframes and the database in the SQL server will be filled with tables containing the transformed results. As shown below, we used Pandas to take the dataframes with the transformed data to load in to the database we created.

Although this step sounds fairly simple, we ran into one issue along the way. During the transformation of the data, and some clean up/removal that we did, the indexes of some of the tables were skewed. Because of this, the dataframes could not be loaded due to a lack of a valid index. We resolved this by rewriting the dataframes to include a numeric index and broke up any groupings we had in the dataframe. This caused the dataframes to be compatible with SQL tables and we were able to load the dataframes after making this fix.