Basketball Attendance Statistics From 2009 to 2018

By Davis Grininger and Daniel Conde

ETL Project: Final Report

10 August 2019

**Background**

The purpose of the project was to produce a usable set of data that we extracted, transformed and loaded to a Postgres SQL server. For our project we chose to extract professional basketball team stats and the fan attendance for each team. We collected a decades worth of data from each source in order to determine strong correlations.

**Sources of Data**

The data that was collected came from basketballreference.com and ESPN. We used Pandas web scraping function to gather the game attendance for between the years 2009 to 2018 from the ESPN website. We gathered team statistics for the same years of each basketball team from basketballreference.com. Examples of the URLs are listed below:

* <http://www.espn.com/nba/attendance/_/year/2008/sort/homePct>
* [https://www.basketball-reference.com/teams/2018.html](https://www.basketball-reference.com/teams/SAC/2018.html)

**Methods of Extraction and Transforming Data**

With the use of splinter and pandas libraries we extracted our data. The methods of extraction created a few challenges in order to transform our data to a usable data set. In the case of the data from basketballreference.com, we used the pandas library to scrape the website for data. While using pandas for webscraping it will return a large list of dataframes. Much of the data that the pandas function returns was of no use for our cause. In order to sort through this data, we used a for loop to sort through each column in each dataframe to find keywords that only existed in the tables we wanted to use. The next challenge that was presented was that many of the tables we chose to use had multiple headers. In order to solve this issue, we renamed all of the columns in the table and dropped the unwanted header by index. After dropping the header, we dropped the unwanted columns and renamed the desired columns to easily identifiable titles. The data extracted from ESPN presented a different issue. Once extracted, the data collected from ESPN was already formatted in way where it was easy to work with however, the teams names did not match with the team names from basketballreference.com. ESPN used team mascots as team identifiers and baskeballreference.com used the team abbreviation. Since we ultimately wanted to use the team abbreviation we had to change the team names in the data collected from ESPN. We did this two ways, one by using a for loop and dictionary that matched each team to a key and the the other way was to use a large for loop of conditional statements. After completing this step, it allowed us to match the data sets seamlessly by team abbreviation.

**Loading the Final Database**

The process for loading the final database with the HTML data we obtained was comparatively simpler than cleaning and transforming the data. The fully transformed data was structured into two final dataframes, with one containing statistical ratings for each basketball team and the other having both the home and away attendance draw for each team. We loaded the data frames into SQL by using the PostgreSQL adapter Psycopg2 and create\_engine from the SQLAlchemy module to connect to PostgreSQL remotely through Python in Pandas. We prepped the database before loading them with the transformed data by creating two empty tables named team\_rating and team\_attendance with the same columns as the two final dataframes. Finally, we used pd.read\_sql\_query to select all rows from each table in the database display the first five using the Pandas head.() function.