**Introduction**

This project collates data from the 2018-19 NBA season. Data on salary, combine performance, game results, and overall statistics were gathered from multiple disparate sources. The 2018-2019 season was examined because it is the most recent year in which the champions cannot be brought into question. With COVID-19 suspending the 2019-2020 NBA, we felt it was most beneficial to use the most complete data set available.

**Extract**

1. Initial basic data was collected from [basketball-referenece.com](https://www.basketball-reference.com/) in the form of CSV files. It contained a complete list of NBA players, teams, and games allowing for primary keys.
2. Data obtained from sportrac contained detailed contract and personnel data.

For these datasets, Pandas was then employed to clean and condition them. Another data source we used to obtain data was spotrac.com. This website is a database that holds contract information for the NBA as well as many other leagues. To obtain this data we needed to scrape the site for the name of the player as well as their salary in dollar amount. For the first attempt at getting this data, we used the requests module. The maximum number of rows accessible using requests was 100 so we needed to import splinter and and use a ChromeDriver.

We used Beautifulsoup to parse through our HTML file. Then we used pandas to append the data to a dataframe. Once all of our data was uploaded to our Resources folder, we ran the data through a data validation file. In the data validation file we normalized the columns to be all lowercase and no spaces, instead replaced with underscores. Additionally the data validation file checked for duplicates in the dataset and the data types. Once that data validated, it was sent to a Validated Data folder in which we would use for the transformation.

**Transform**

Below is the general route that we would take when transforming our datasets for loading.

#Import dependencies

#dropping null value columns to avoid errors

#File path to dataset for df

#File path to player dataset

#Merge dataset with players to get primary key

#Drop duplicates due to merge

#Drop unneeded columns

#Access dataset only in 2018

#drop duplicates due to merge

#Ensure data types are correct for load

#Rearrange columns

#Rename columns for consistency across all datasets

#Export to csv

**Load**

The first step in the load process was to create an ERD that follows the outline of our dataset. Using QuickDBD we created the ERD that is located in the code section of our repository. Using QuickDBD export function, we exported to a PostgreSQL file to be used in our jupyter notebook. When loading we came across an issue with regards to our primary keys. The table that we were using for players as primary keys actually had duplicates due to some players in the league being on multiple teams in the 2018-2019 season. Because of this issue we needed to create a primary table of just player\_id and player name to ensure no duplicates are in the primary table

 Another issue we came across when loading our data happened with loading in the combine data for current NBA players. The issue we came across was that some of our columns had null values. The original data type we had entered for the SQL query file was integer, which would not accept a null value, we experimented with other data types and found that float worked when doing a couple other alterations to the code using the psycopg2 module.

**Discussion/Limitations/Next Steps**

**TRANSFORM;**

Didn’t have complete datasets; a limitation we found with the salary cleaning  was that it didn’t include every single player that played in the 2018 season. There were 748 players in the 2018 season and we were only able to obtain salary data on 469 players and when we merged the two dataframes together it dropped to 445 players. The reason we believe why our salary source didn’t include all players is because some players played on non-guaranteed contracts, some played on 10-day contracts and some don't make it through the full season.

**LOAD TO SQL;**

1. cursor.close() - is a required step, without a close and re-open the tables were being created and stored in the default postgresql db. It should be noted that a second instance of cursor can also be created, which is the route the team took.
2. Null Value Recognition; issues arose as to how to integrate null values from the CSVs into the database. While it was decided early on that 0 would not be used as a place filler for NULL as it would lead to data pollution and integrity issues down the line, a simple addition of null=’NULL’ was added to the copy\_from command. The [psycopg2 documentation](https://www.psycopg.org/docs/) though very robust is unfortunately not for the novice.
3. OS agnosticity;  were not discovered until kernel 10 of the export to sql notebook, where in the folder level slashes had to be removed. These would not have been discovered had it not been for the team's mixed OS work environments.
4. Debugging / Database Drop; every instance of database and code debugging required a connection close and a manual database drop within the pgAdmin console, no other method worked as efficiently, though this added minutes to every process.

**Next steps (some ideas, don’t have to include);**

1. Include playoff stats and compare with regular season stats to see if players regress or improve their play in the playoffs.
2. Look at how far teams travel throughout the year and see if there are any relationships between distance traveled and team or individual player performance (ie. west coast team traveling to east coast, does it impact their play).
3. Can compare previous years with the bubble to see if the bubble impacted play(ie. Did players perform better or worse in the bubble, did home-court matter etc..)