ENAE 380 Project User’s Manual

Simulating NBA Seasons - William Rowe

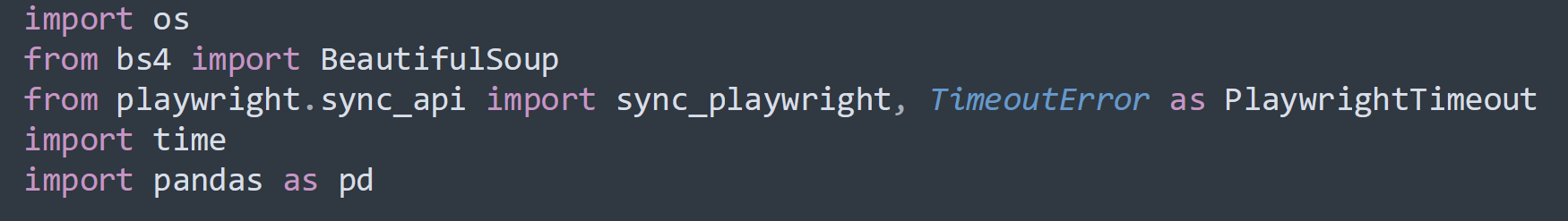
1.1 Necessary Files

All files required to implement the project include the getdata.py file, parse\_data.py file, and the test.py file. The test.py file requires an “nba\_games.csv” file to run, which is saved as a result of running getdata and then parse\_data after. The csv file is not needed to be submitted in order for the project to function, as it can be created with said python scripts, but because it takes a long time to run/create, I will submit the .csv file as well for convenience.

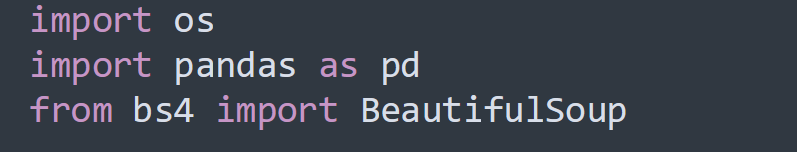
1.2 Necessary Libraries

The following libraries are required to be installed and imported before running the python code:

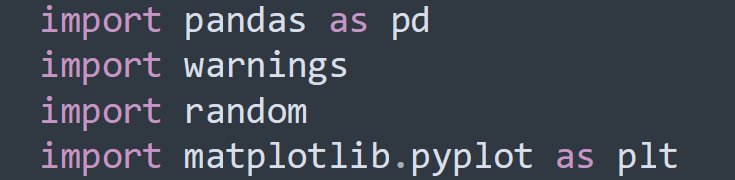
In getdata.py:

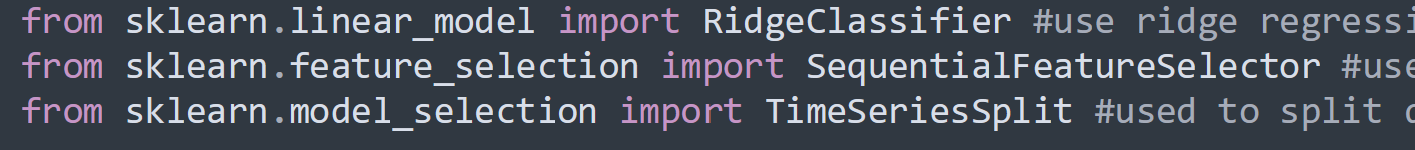


In parse\_data.py:



In test.py:





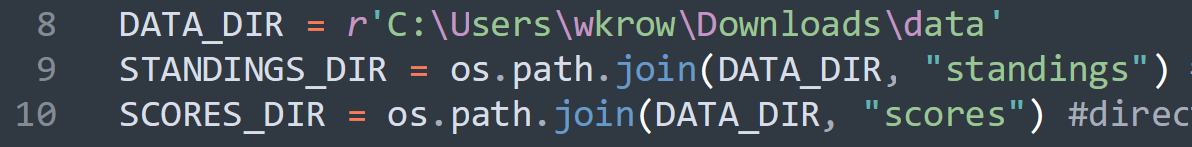




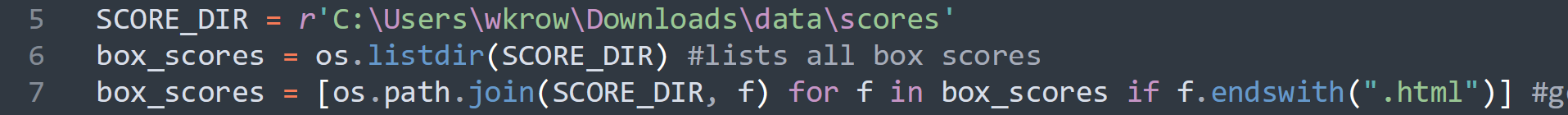
1.3 Gathering/Parsing Data into a Dataframe

Getdata.py and parse\_data.py have specific directories designated to save and take out file data to/from. You can make/edit the directory names in the script so that they save to whatever folder directory you want it to be saved to. For me, I have DATA\_DIR go to a folder called “data” in downloads, STANDINGS\_DIR go to a folder called “standings” which is located in the data folder, and SCORES\_DIR go to a folder called “scores” which is also located within the data folder. You can see where the directories were specified below:

In getdata.py:



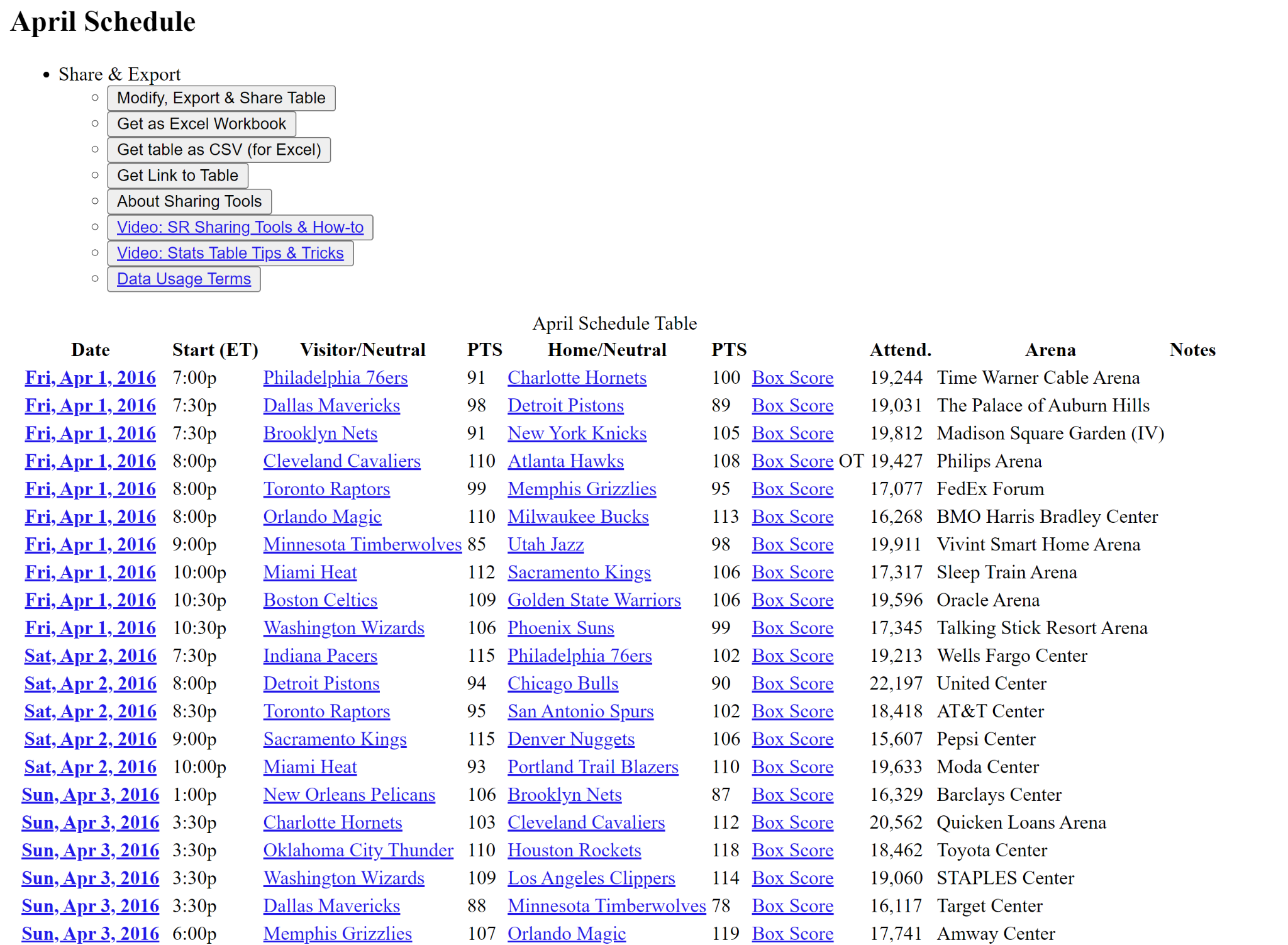
In parse\_data.py:



You can still run getdata.py for a little while and then cancel the script to see what the script is actually saving into the designated directories. First, it saves data to “standings”, then once that is finished, it starts saving data to “scores” which takes much longer to finish.

For the sake of convenience, I will show you what the standings and scores data looks like below:

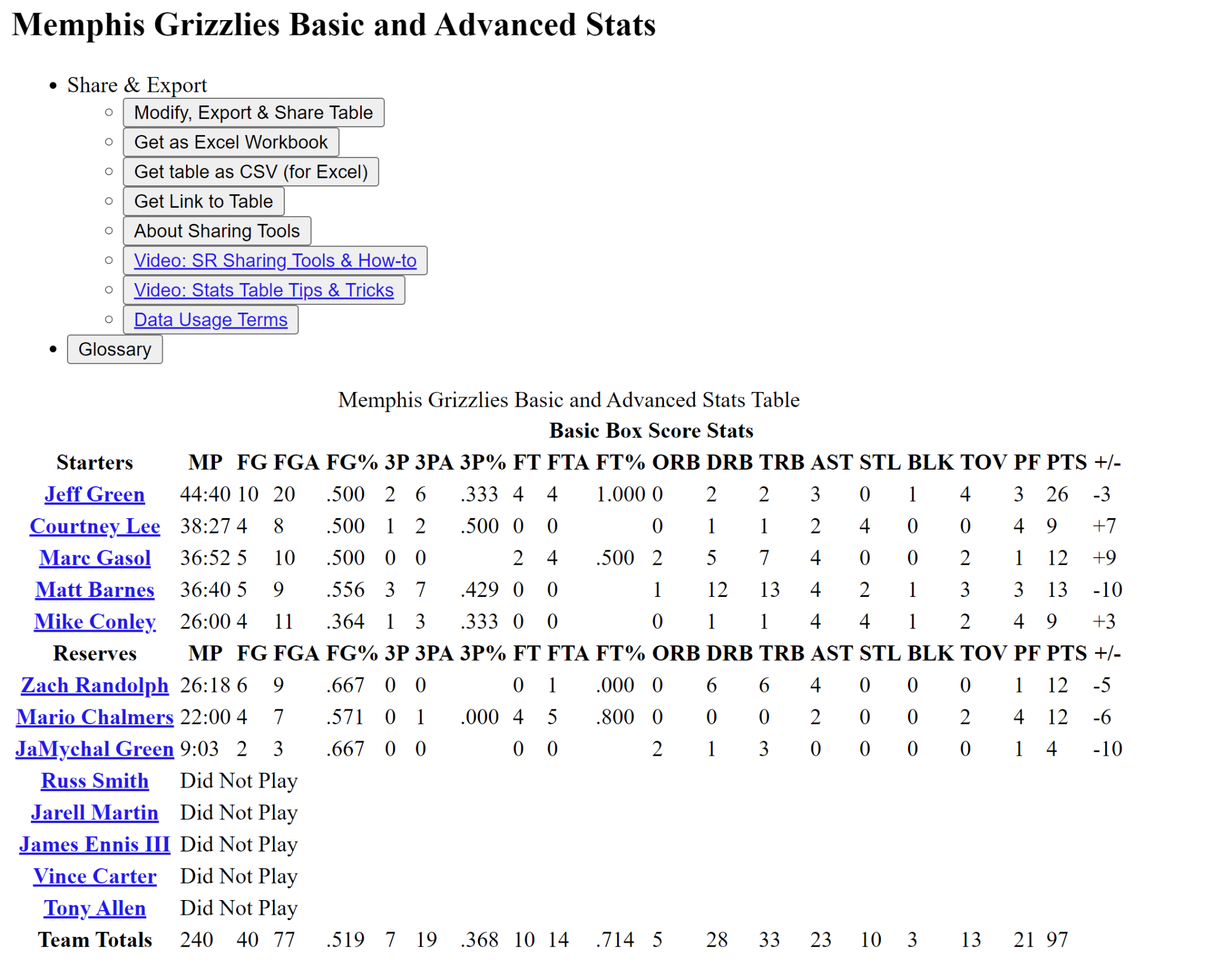
Here is what a standings file looks like:

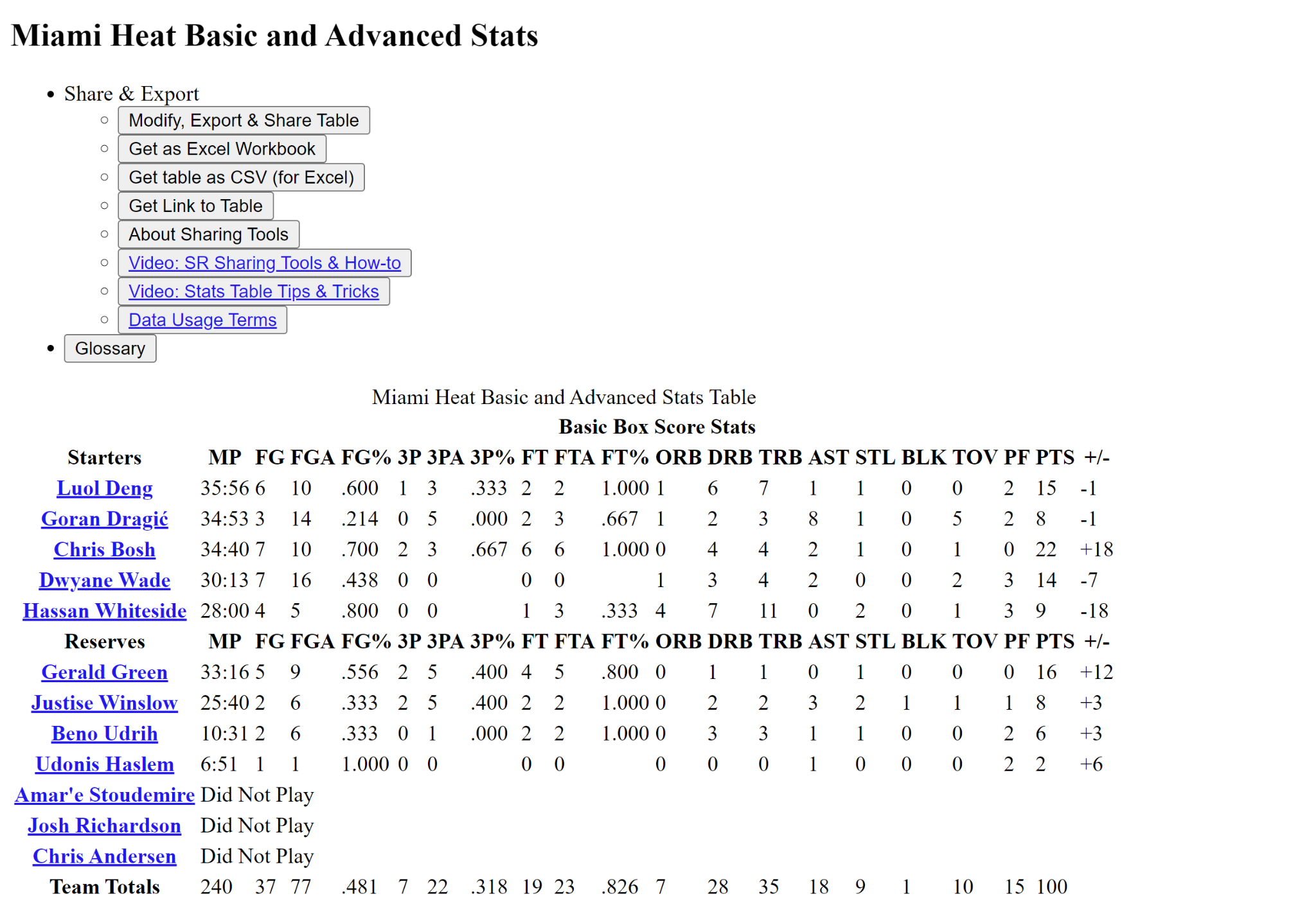


Here is what part of a scores file looks like:











1.4 User Inputs After Running test.py

After running test.py, the user will have to input a year from 2018-2021. Once the regular season simulations for the season has been completed,regular season info and a plot of simulated vs. real team win totals will appear. After X’ing out of it, the playoffs (and play-in if the inputted season is 2020 or 2021) will be simulated, display the games and results in text, and also display the play-in bracket if the season is 2020 or 2021. If the play-in bracket appears, X out of it, and the simulated playoff bracket will appear. X out of the simulated playoff bracket to see the actual playoff bracket for that year.