In our solution, we parsed through each game and checked to see if each team had enough remaining games to have a higher number of wins than the eighth ranked team in their conference. If it would be tiebreaker, we then used the first two criteria in the PDF given to see if the team in question could win the tiebreaker if it came down to it. We recognize our solution is not full, and if we had more time, we would have implemented our code to check the rest of the tiebreaker rules as well as checking against any of the teams that could’ve taken the eighth spot, instead of just the current eight ranked team. In order to read and write directly to the excel file, we used a package named openpyxl which can be downloaded through pip, or from <https://pypi.python.org/pypi/openpyxl/2.4.8> . Our solution can be run as such:

python Elimination.py --excel\_file Analytics\_Attachement.xlsx Division\_Info 2016\_17\_NBA\_Scores

The solution can also be run with other similar excel files, by simply changing the command to be “--excel\_file (name of excel file) (name of division info sheet) (name of score sheet)”. The elimination dates will be written directly to the excel file in a new sheet named “Elimination Dates”. We included a copy of the excel sheet with the code-generated elimination dates sheet.