Project Goal: In a short period of time (roughly 4 hours), create a simple but informative baseball player evaluation metric that can be greatly expanded upon given additional time.

The Metric: A basic measure of how much a player increases their team's expected run total per at-bat, or, for pitchers, how much the opposing batter increases their team's expected run total per at-bat.

Steps:

Data Collection: Create a python script to scrape play-by-play data from baseball-reference.com and store it in a created local database. Data from years 2021 and 2022 will be collected.

Dataset Creation: Clean the collected data by removing mid-at-bat events (steals, passed-balls, etc.), and then determine the number of outs and baserunners before and after each at bat.

Assign Value to Each 'State': For each unique combination of outs/baserunners, calculate the average number of runs scored for the batting team after an at-bat in that 'state'.

Calculate the Metric: Grouping by year, team, and player, calculate the difference between the expected number of runs scored given the pre-at-bat 'state' and the observed runs scored, plus the value of the post-at-bat 'state'.

Analysis: The results of this metric were largely unsurprising on each end of the metric spectrum, but they did seem to align well with existing league metrics such as oWAR. The addition of salary/contract data for each player would allow for a more accurate dissection of player value using this metric. One major flaw seen in this metric is the fact that its average (for batters) is around .11 instead of 0. This is due to the addition of the next state's value in calculating the returns of a player's at bat. If this wasn't included, the true impact of an at-bat wouldn't be accurate. For example, given a 'state' of 1 out and a man on third, a single and a triple would receive the same score (1 – the pre-at-bat state value). However, since the triple leaves the batter's team in a more advantageous state than a single, the value of the post-at-bat state should be added to account for this. To resolve the 0-average issue, the post-at-bat state should be included in the value calculation outlined in the 3rd step above, which will require a recursive process. While this metric is extremely basic given the 4-hour time constraint, it provides numerous lanes for improvement, even in the confines of a very limited dataset. A few notable ones are opposing pitcher/batter quality and opposing defensive quality, both of which could be implemented using solely this metric and, again, basic recursion.