Team JBT
Thomas Dunn, Jessica Lu & Brianna Pinder
CS122

Project Proposal

GOAL:

Create an interface that allows the user to query and manipulate baseball statistics in the following ways:

- Search for games with specified characteristics such as number of runs, teams playing, date/year, records
- Create fantasy teams assembled out of mixed and matched players from different eras and compare expected win totals based on wins above replacement (WAR) statistics
- Compare side by side two players or teams based on a number of statistics

SOURCES OF DATA:

Baseball Reference: The main source of baseball statistics. Contains stats for both players and games.

FanGraphs: Contains baseball stats and also some calculations of interesting statistics, as well as visuals.

MLB: The official website of the MLB with its own player and game statistics.

Baseball America and Baseball Savant: Two more sites with calculations of interesting baseball statistics and graphs.

WORK REQUIRED:

Identifying important metrics: Determine which statistics we want to grab and for what years that would help us in meeting our goal

Scraping data sources: Figuring out how to scrape the above data sources

Determining what outputs we want/writing algorithms: Determining how to save and access certain statistics in the most efficient way (SQL seems like a good idea). Additionally, we need to write an algorithm to gather WAR and compute what the expected win percentage would be for a team based on total WAR. Furthermore, need to figure out how to compare two WAR totals for two teams and determine the chances of each winning a head to head match up.

Interface / Handling Queries: We will need to design an interface for the user to select which data outputs they want from our software. For searching for games with specific parameters, they must be able to enter those in and the software must be able to return an appropriate response. For determining fantasy teams and WAR related statistics, they should be able to select a player and team. Finally, for comparing two players, the interface should let the user pick the players and have it return a side-by-side comparison of relevant stats.

POTENTIAL ALGORITHMS/STRUCTURES:

Nearest Neighbor Search: Finds closest and most similar points--closeness typically expressed in terms of a dissimilarity function: the less similar the objects, the larger the function values. This algorithm could be used to compare similar players.

Marcel Algorithm: Developed by Tom Tango, Marcel examines a player's performance over the past three seasons, weighting more recent seasons more heavily, while regressing performance toward the league mean and applying an age adjustment to generate a season forecast, both of those at equal rates per player. Major League Equivalencies (MLEs) -- translations of a player's minor league stats into major league terms -- aren't included.

Runs-To-Wins: Pythagorean Formula:

$$W\text{-}L\% = (RS^x) / (RS^x + RA^x)$$

Where x = (runs/gm)^.285, RS = runs scored, RA = runs attempted

TIMELINE:

	Class deadlines	**Outside Work**
Week 4	Proposal due	Outlining and breaking up tasks
Week 5		Write code to scrape data to create database
Week 6	Progress check-in	Write SQL code to search through database and determine calculations necessary

Week 7		Write code to create fantasy teams based on WAR stats found in the database
Week 8		Determine how to compare two players and write code to extract these statistics
Week 9		Create website interface and design organization for each of the three tasks.
Week 10		Finalize project work, test through querying
March 15	Turn in software	