**Repository**
https://github.com/zmathews37/SI206-P3/

**Goals For the Project:**
The goals for this project were to compare various MLB stats and World Series outcomes in order to see if some of the spoken trends about baseball (home field advantage, which positions are better at hitting, etc.) hold any value. Another goal was to compare world series teams in order to see the team stats and if a particular stat was more important than the others. We planned to use the rapidAPI for baseball stats (https://rapidapi.com/theapiguy/api/mlb-data/), MLB stats API (https://appac.github.io/mlb-data-api-docs/), and MLB.com/scores site in order to collect our data.

We planned for the Rapid API to be used to get player data for each player and store them in a players data table. We planned to store stats like OPS for hitters and ERA for pitchers alongside some other important indicators of success. We planned for the MLB API to be used to get team roster information and store it in a team data table. We planned to store the top hitter for each position in the table as well as the top 5 starting pitchers and top 3 relief pitchers. We planned for the MLB.com/scores website to allow us to find the playoff matchups for a given year and store the matchups, game results, and series results in a database for each matchup/year. We also planned to store the home/away team in every matchup.

**Goals Achieved:**
We were able to make graphs comparing the home vs away runs scored and allowed; this let us quantify if home field advantage was valid during World Series games. We were also able to plot the positional stats for hitters; we got to see that the typical positions in DH and 1st base held the best hitting stats. We were also able to compare the World Series team stats to see that the teams held similar stats and there weren't too many outliers except for the larger number of homeruns hit by the 2017 Astros. We ended up using each of the APIs and the website that we planned to use.

The Rapid API was actually used to get career stats for certain players. This was used to compare positional groups' weighted average stats. The MLB API was actually used to gather team data alongside individual player's yearly data. We did not end up storing/filtering out the top 5 starting pitchers and top 3 relief pitchers per team. MLB.com/scores was used as intended with storing the game results for each World Series matchup.

**Problems that you faced:**
One problem we faced was that GitHub Copilot would suggest to use dates that had no games held on them, so the scraping would return an error. Another problem we had was that we initially could not figure out how to limit data stored from the API in the database to 25 or fewer on each file run. In terms of graphing, we encountered issues with bar charts creating 2

separate charts instead of having side by side bar charts on the same figure. We also encountered duplicate string data issues.

**Calculations From the Data:**

--- Output for Plot 1 ---

Astros Playoffs Runs Scored At Home Per Game: 3.93
Astros Playoffs Runs Scored On Road Per Game: 4.07
Astros Playoffs Runs Allowed At Home Per Game: 5.53
Astros Playoffs Runs Allowed On Road Per Game: 3.13

Cardinals Playoffs Runs Scored At Home Per Game: 4.2
Cardinals Playoffs Runs Scored On Road Per Game: 4.0
Cardinals Playoffs Runs Allowed At Home Per Game: 3.2
Cardinals Playoffs Runs Allowed On Road Per Game: 4.5

Red Sox Playoffs Runs Scored At Home Per Game: 6.14
Red Sox Playoffs Runs Scored On Road Per Game: 5.12
Red Sox Playoffs Runs Allowed At Home Per Game: 2.0
Red Sox Playoffs Runs Allowed On Road Per Game: 3.25

Giants Playoffs Runs Scored At Home Per Game: 6.86
Giants Playoffs Runs Scored On Road Per Game: 3.0
Giants Playoffs Runs Allowed At Home Per Game: 2.43
Giants Playoffs Runs Allowed On Road Per Game: 3.11

--- End Output for Plot 1 ---

--- Output for Plot 2 ---

Weighted Average of Home Runs and OPS for Each Position

| Position | Home Runs | OPS |
|---|---|---|
| C | 32.72 homeruns per 1000 at bats | 0.736 |
| 1B | 45.66 homeruns per 1000 at bats | 0.851 |
| 2B | 17.62 homeruns per 1000 at bats | 0.738 |
| 3B | 36.3 homeruns per 1000 at bats | 0.792 |
| SS | 22.72 homeruns per 1000 at bats | 0.731 |
| LF | 31.44 homeruns per 1000 at bats | 0.766 |
| CF | 27.61 homeruns per 1000 at bats | 0.743 |
| RF | 34.28 homeruns per 1000 at bats | 0.751 |
| DH | 53.02 homeruns per 1000 at bats | 0.881 |

--- End Output for Plot 2 ---

--- Output for Plot 3 ---

Comparing Team Stats of World Series teams

OPS
San Francisco Giants: 0.756
St. Louis Cardinals: 0.788
Boston Red Sox: 0.731
Houston Astros: 0.826

Home Runs
San Francisco Giants: 162
St. Louis Cardinals: 160
Boston Red Sox: 165
Houston Astros: 238

ERA
San Francisco Giants: 3.36
St. Louis Cardinals: 3.77
Boston Red Sox: 4.7
Houston Astros: 4.1

WHIP
San Francisco Giants: 1.27
St. Louis Cardinals: 1.31
Boston Red Sox: 1.37
Houston Astros: 1.27

--- End Output for Plot 3 ---

--- Output for Plot 4 ---

Average Runs Scored Per Game
In Game 1's:
Home: 5.28
Away: 3.83

In Game 2's:
Home: 4.28
Away: 3.11

In Game 3's:
Home: 4.0
Away: 4.17

In Game 4's:
Home: 3.72
Away: 4.61

In Game 5's:
Home: 3.73
Away: 4.53

In Game 6's:
Home: 4.8
Away: 3.9

In Game 7's:
Home: 3.6
Away: 4.8

--- End Output for Plot 4 ---

--- Output for Plot 5 ---

Run Differentials For Each Game:

Home Team - Away Team (Year)  Differential = Home Score - Away Score
White Sox 5 - 3 Astros (2005)  Differential = 2
White Sox 7 - 6 Astros (2005)  Differential = 1
Astros 5 - 7 White Sox (2005)  Differential = -2
Astros 0 - 1 White Sox (2005)  Differential = -1
Tigers 2 - 7 Cardinals (2006)  Differential = -5
Tigers 3 - 1 Cardinals (2006)  Differential = 2
Cardinals 5 - 0 Tigers (2006)  Differential = 5
Cardinals 5 - 4 Tigers (2006)  Differential = 1
Cardinals 4 - 2 Tigers (2006)  Differential = 2
Red Sox 13 - 1 Rockies (2007)  Differential = 12
Red Sox 2 - 1 Rockies (2007)  Differential = 1
Rockies 5 - 10 Red Sox (2007)  Differential = -5
Rockies 3 - 4 Red Sox (2007)  Differential = -1
Rays 2 - 3 Phillies (2008)  Differential = -1
Rays 4 - 2 Phillies (2008)  Differential = 2
Phillies 5 - 4 Rays (2008)  Differential = 1
Phillies 10 - 2 Rays (2008)  Differential = 8

Phillies 4 - 3 Rays (2008)  Differential = 1
Yankees 1 - 6 Phillies (2009)  Differential = -5
Yankees 3 - 1 Phillies (2009)  Differential = 2
Phillies 5 - 8 Yankees (2009)  Differential = -3
Phillies 4 - 7 Yankees (2009)  Differential = -3
Phillies 8 - 6 Yankees (2009)  Differential = 2
Yankees 7 - 3 Phillies (2009)  Differential = 4
Giants 11 - 7 Rangers (2010)  Differential = 4
Giants 9 - 0 Rangers (2010)  Differential = 9
Rangers 4 - 2 Giants (2010)  Differential = 2
Rangers 0 - 4 Giants (2010)  Differential = -4
Rangers 1 - 3 Giants (2010)  Differential = -2
Cardinals 3 - 2 Rangers (2011)  Differential = 1
Cardinals 1 - 2 Rangers (2011)  Differential = -1
Rangers 7 - 16 Cardinals (2011)  Differential = -9
Rangers 4 - 0 Cardinals (2011)  Differential = 4
Rangers 4 - 2 Cardinals (2011)  Differential = 2
Cardinals 10 - 9 Rangers (2011)  Differential = 1
Cardinals 6 - 2 Rangers (2011)  Differential = 4
Giants 8 - 3 Tigers (2012)  Differential = 5
Giants 2 - 0 Tigers (2012)  Differential = 2
Tigers 0 - 2 Giants (2012)  Differential = -2
Tigers 3 - 4 Giants (2012)  Differential = -1
Red Sox 8 - 1 Cardinals (2013)  Differential = 7
Red Sox 2 - 4 Cardinals (2013)  Differential = -2
Cardinals 5 - 4 Red Sox (2013)  Differential = 1
Cardinals 2 - 4 Red Sox (2013)  Differential = -2
Cardinals 1 - 3 Red Sox (2013)  Differential = -2
Red Sox 6 - 1 Cardinals (2013)  Differential = 5
Royals 1 - 7 Giants (2014)  Differential = -6
Royals 7 - 2 Giants (2014)  Differential = 5
Giants 2 - 3 Royals (2014)  Differential = -1
Giants 11 - 4 Royals (2014)  Differential = 7
Giants 5 - 0 Royals (2014)  Differential = 5
Royals 10 - 0 Giants (2014)  Differential = 10
Royals 2 - 3 Giants (2014)  Differential = -1
Royals 5 - 4 Mets (2015)  Differential = 1
Royals 7 - 1 Mets (2015)  Differential = 6
Mets 9 - 3 Royals (2015)  Differential = 6
Mets 3 - 5 Royals (2015)  Differential = -2
Mets 2 - 7 Royals (2015)  Differential = -5
Indians 6 - 0 Cubs (2016)  Differential = 6
Indians 1 - 5 Cubs (2016)  Differential = -4
Cubs 0 - 1 Indians (2016)  Differential = -1

Cubs 2 - 7 Indians (2016)  Differential = -5
Cubs 3 - 2 Indians (2016)  Differential = 1
Indians 3 - 9 Cubs (2016)  Differential = -6
Indians 7 - 8 Cubs (2016)  Differential = -1
Dodgers 3 - 1 Astros (2017)  Differential = 2
Dodgers 6 - 7 Astros (2017)  Differential = -1
Astros 5 - 3 Dodgers (2017)  Differential = 2
Astros 2 - 6 Dodgers (2017)  Differential = -4
Astros 13 - 12 Dodgers (2017)  Differential = 1
Dodgers 3 - 1 Astros (2017)  Differential = 2
Dodgers 1 - 5 Astros (2017)  Differential = -4
Red Sox 8 - 4 Dodgers (2018)  Differential = 4
Red Sox 4 - 2 Dodgers (2018)  Differential = 2
Dodgers 3 - 2 Red Sox (2018)  Differential = 1
Dodgers 6 - 9 Red Sox (2018)  Differential = -3
Dodgers 1 - 5 Red Sox (2018)  Differential = -4
Astros 4 - 5 Nationals (2019)  Differential = -1
Astros 3 - 12 Nationals (2019)  Differential = -9
Nationals 1 - 4 Astros (2019)  Differential = -3
Nationals 1 - 8 Astros (2019)  Differential = -7
Nationals 1 - 7 Astros (2019)  Differential = -6
Astros 2 - 7 Nationals (2019)  Differential = -5
Astros 2 - 6 Nationals (2019)  Differential = -4
Dodgers 8 - 3 Rays (2020)  Differential = 5
Dodgers 4 - 6 Rays (2020)  Differential = -2
Rays 2 - 6 Dodgers (2020)  Differential = -4
Rays 8 - 7 Dodgers (2020)  Differential = 1
Rays 2 - 4 Dodgers (2020)  Differential = -2
Dodgers 3 - 1 Rays (2020)  Differential = 2
Astros 2 - 6 Braves (2021)  Differential = -4
Astros 7 - 2 Braves (2021)  Differential = 5
Braves 2 - 0 Astros (2021)  Differential = 2
Braves 3 - 2 Astros (2021)  Differential = 1
Braves 5 - 9 Astros (2021)  Differential = -4
Astros 0 - 7 Braves (2021)  Differential = -7
Astros 5 - 6 Phillies (2022)  Differential = -1
Astros 5 - 2 Phillies (2022)  Differential = 3
Phillies 7 - 0 Astros (2022)  Differential = 7
Phillies 0 - 5 Astros (2022)  Differential = -5
Phillies 2 - 3 Astros (2022)  Differential = -1
Astros 4 - 1 Phillies (2022)  Differential = 3

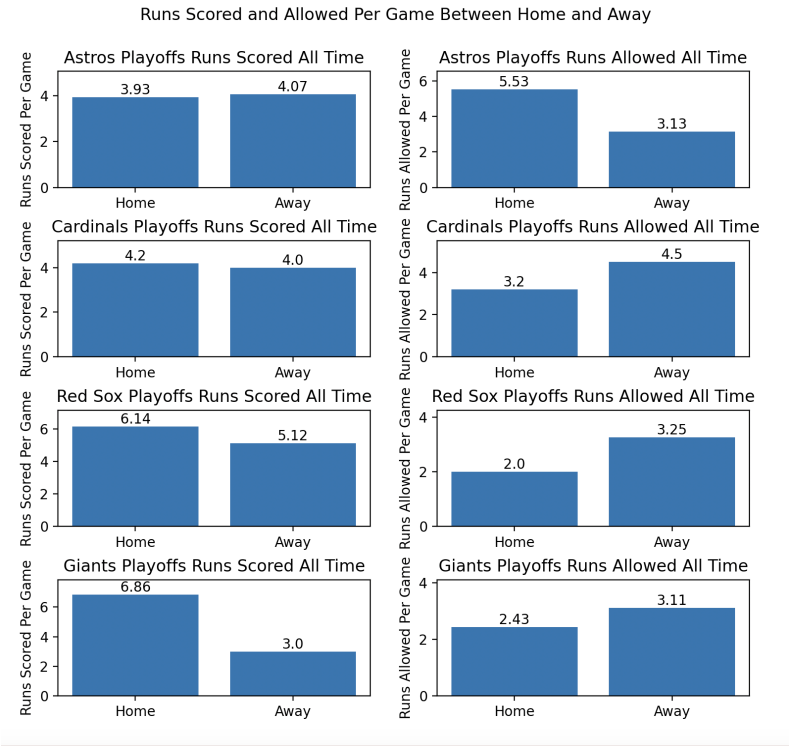
Run Differentials Distribution for Each Team

Astros: [-2, -1, -2, -1, -2, 1, 2, -4, 1, -2, 4, -1, -9, 3, 7, 6, -5, -4, -4, 5, -2, -1, 4, -7, -1, 3, -7, 5, 1, 3]
Cardinals: [5, -2, 5, 1, 2, 1, -1, 9, -4, -2, 1, 4, -7, 2, 1, -2, -2, -5]
Red Sox: [12, 1, 5, 1, 7, -2, -1, 2, 2, 5, 4, 2, -1, 3, 4]
Giants: [4, 9, -2, 4, 2, 5, 2, 2, 1, 6, -5, -1, 7, 5, -10, 1]

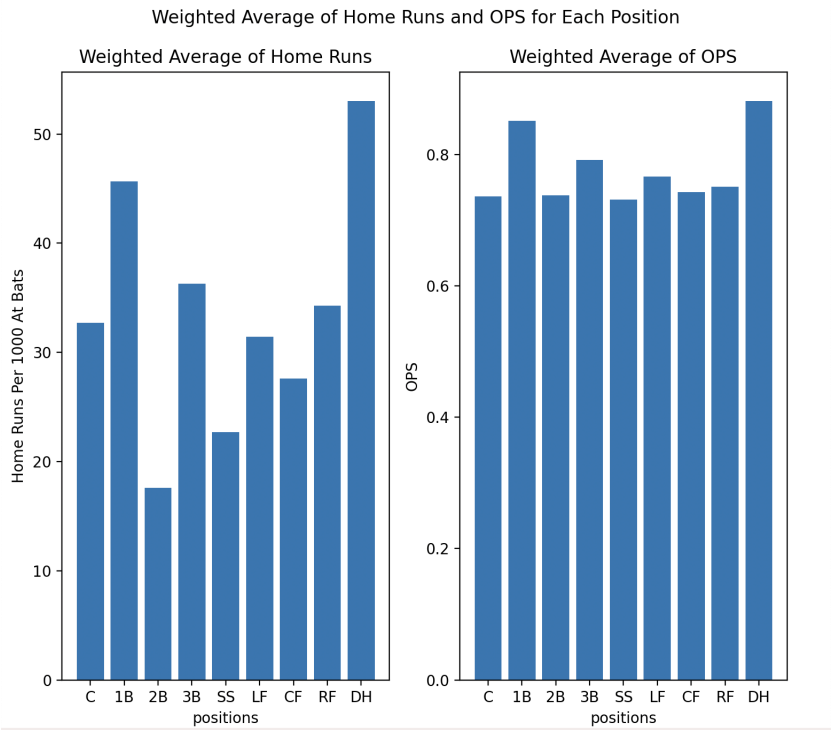
--- End Output for Plot 5 ---

## **Visualizations Created:**

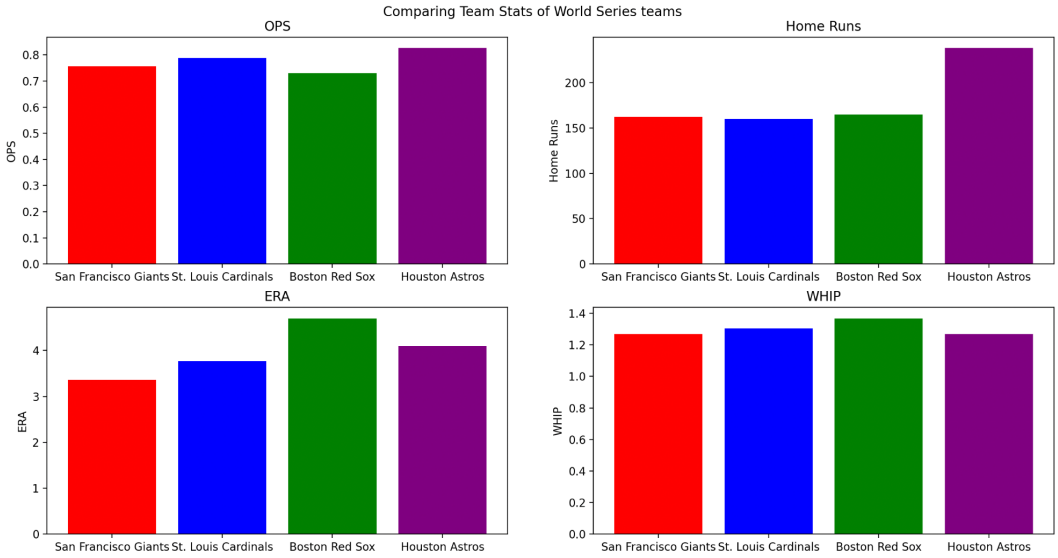
Plot 1:



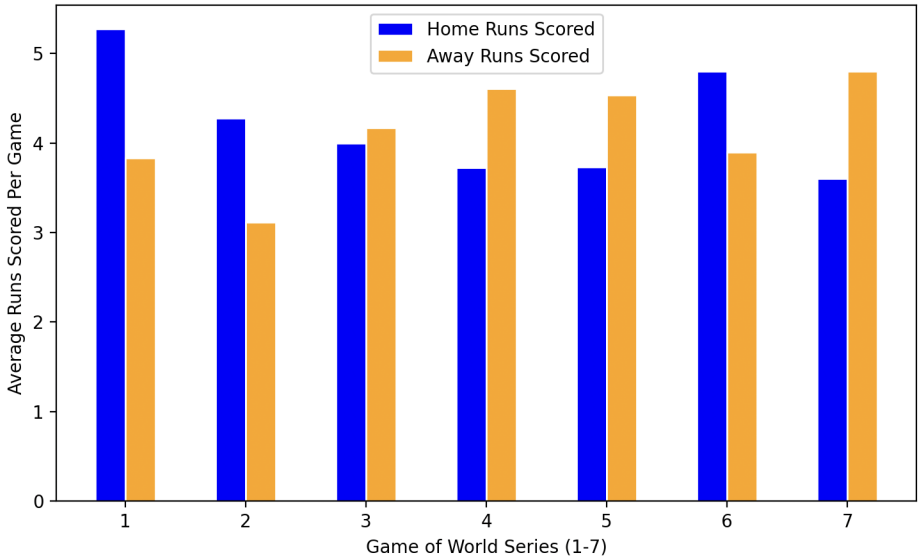

Plot 2:


Weighted Average of Home Runs and OPS for Each Position
Weighted Average of Home Runs
Home Runs Per 1000 At Bats
0
10
20
30
40
50
C
1B
2B
3B
SS
LF
CF
RF
DH
positions
Weighted Average of OPS
OPS
0.0
0.2
0.4
0.6
0.8
C
1B
2B
3B
SS
LF
CF
RF
DH
positions

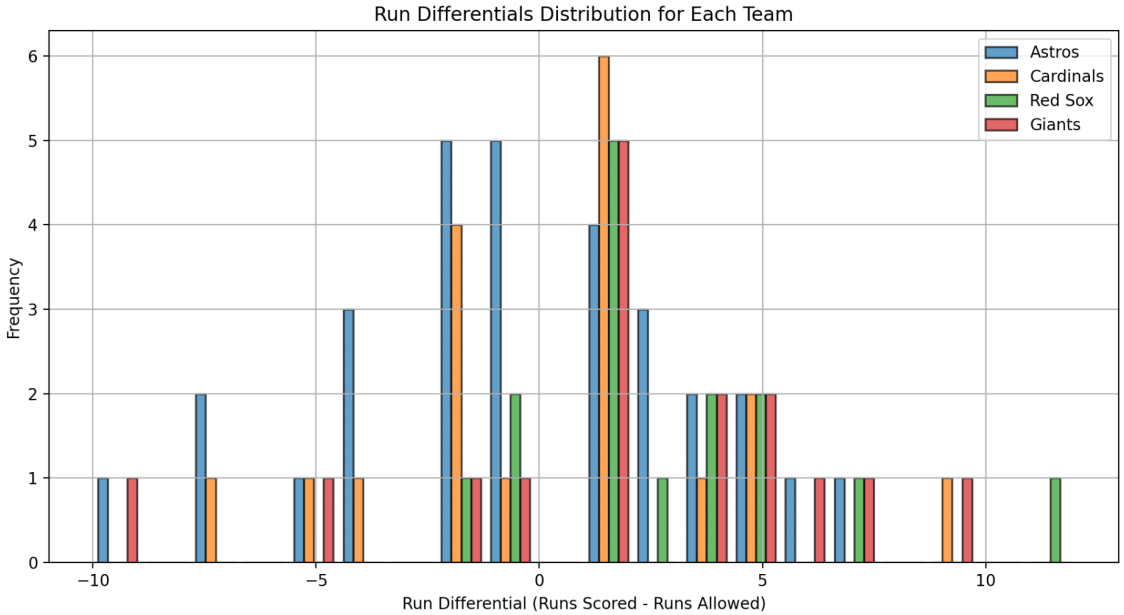
Plot 3:


Comparing Team Stats of World Series teams
OPS
OPS
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
San Francisco Giants
St. Louis Cardinals
Boston Red Sox
Houston Astros
Home Runs
Home Runs
0
50
100
150
200
San Francisco Giants
St. Louis Cardinals
Boston Red Sox
Houston Astros
ERA
ERA
0
1
2
3
4
San Francisco Giants
St. Louis Cardinals
Boston Red Sox
Houston Astros
WHIP
WHIP
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
San Francisco Giants
St. Louis Cardinals
Boston Red Sox
Houston Astros

Plot 4:


Home Runs Scored
Away Runs Scored
Average Runs Scored Per Game
Game of World Series (1-7)
0
1
2
3
4
5
1
2
3
4
5
6
7

Plot 5:


Run Differentials Distribution for Each Team
Astros
Cardinals
Red Sox
Giants
Frequency
Run Differential (Runs Scored - Runs Allowed)
0
1
2
3
4
5
6
−10
−5
0
5
10

**Instructions for Running the Code:**

1. Run mlbAPI.py until the message "Added All Players From All Specified Teams!" is printed.
2. Run rapidAPI.py until the message "All Career Stats Added" is printed.
3. Run MLBWebsite.py until the message "Done scraping all dates!" is printed.
4. Run plot1.py to view visualization 1
5. Run plot2.py to view visualization 2
6. Run plot3.py to view visualization 3
7. Run plot4.py to view visualization 4
8. Run plot5.py to view visualization 5

**Code/Functions Documentation:**

For mlbAPI.py:

get_api_full_info(url, input_headers, input_params)

- Inputs: url [string], input_headers [dictionary], input_params [dictionary]
- Outputs: response.json() [dictionary]
- Details: Gets the data from the API using the input url, headers, and params. Returns the data as a python dictionary.

get_roster(team_name, year)

- Inputs: team_name [string], year [string[
- Outputs: data["roster_team_alltime"]["queryResults"]["row"]  [dictionary]
- Details: According to the input parameters, get the team roster for the team and year specified. Return the data as a python dictionary.

get_player_statistics(player_id, position, year, team_id)

- Inputs: player_id [integer], position [string]. year [integer], team_id [string]
- Outputs: results [dictionary]
- Details: Gets the statistics for the specified player (given input parameters) from the mlbAPI and returns the statistics as a python dictionary.

get_connection()

- Inputs: None
- Outputs: (connection, cursor) [tuple containing database connection and cursor]
- Details: connects to our baseball.db database and returns the corresponding connection and cursor for it

add_player_to_Players_table(player_id, name, team_name, position, year)

- Inputs: player_id [integer] name [string], team_name [string], year [integer]
- Outputs: None
- Details: Adds a player's information to the database in table Players

add_player_to_Statistics_table(player_id, position, year, team_id)

- Inputs: player_id [integer] position [string], year [integer], team_id [string]
- Outputs: None
- Details: Adds a player's information to the database in table Statistics. Uses the get_player_statistics function to get the stats.

put_full_roster_in_database(team_name, year, iterator)
- Inputs: team_name [string], year [integer], iterator [integer]
- Output; None
- Details: Gets a full roster specified by team_name and year parameters and puts the corresponding players in the database using the add_player_to_Players_table and add_player_to_Statistics_table functions. Iterator parameter makes sure only 25 are added on each run.

main()
- Inputs: None
- Outputs: None
- Details: Creates the tables in the database if they don't exist. Calls put_full_roster_in_database to fill database.

For rapidAPI.py:
get_roster(team_name, year, teamsDict)
- Inputs: team_name [string], year [integer], teamsDict [dictionary]
- Output: Results [dictionary]
- Details: Gets the roster with the specified team_name and year parameters. Returns a python dictionary with the roster. teamsDict is used to get the team id.

get_career_stats(player_id, pitcher)
- Inputs: player_id [integer], pitcher [boolean]
- Outputs: dictionary
- Details: Gets the player career statistics from the rapidAPI. Returns a dictionary with these stats.

store_teams_dict()
- Inputs: None
- Outputs: d [dictionary]
- Details: Creates and returns a dictionary mapping key of team name to value of team id. Calls the rapidAPI to get team name and id mapping.

get_connection()
- Inputs: None
- Outputs: (connection, cursor) [tuple containing database connection and cursor]
- Details: connects to our baseball.db database and returns the corresponding connection and cursor for it

put_player_in_career_stats_table(player_id, name, position, at_bats, ops, homeruns, innings_pitched, era, whip)

- Inputs: player_id [integer], name [string], position [string], at_bats [integer], ops [float], homeruns [integer], innings_pitched [integer], era [float], whip [float]
- Outputs: None
- Details: Puts the specified player and their careers stats (parameters) into the database table CareerStats.

main()

- Inputs: None
- Outputs: None
- Details: Creates the CareerStats table if it doesn't exist. Calls above helper functions in order to populate the database.

For MLBWebsite.py:
get_connection()

- Inputs: None
- Outputs: (connection, cursor) [tuple containing database connection and cursor]
- Details: connects to our baseball.db database and returns the corresponding connection and cursor for it

put_scores_in_database(list_of_dates, iterator)

- Inputs: list_of_dates [list of strings], iterator [integer]
- Outputs: (boolean, iterator) [tuple containing boolean and integer]
- Details: Takes in a list of dates and an iterator, scrapes 25 dates from the list, and adds these dates to the database into the table called Scores. Returns a boolean indicator whether or not all dates from list have been scraped and an iterator saying how many dates have been scraped in the current file execution.

main()

- Inputs: None
- Outputs: None
- Details: Establishes connection to the database, creates the Scores table if it does not exist, and calls the function put_scores_in_database for each date list in the overarching series_to_scrape variable. Populates the database with World Series scores.

For Plot1.py:
main()

- Inputs: None
- Outputs: None
- Details: Calculates runs scored and runs allowed between home team and away team. Plots these differences for each team

For Plot2.py:
sort_data_into_positions(results)

- Inputs: results [list of tuples]
- Outputs: d [dictionary]
- Details: Takes in a list of tuples where each tuple contains position, at_bats, ops, homeruns, innings_pitched, era, and whip from the database table CareerStats. Returns a dictionary that maps position to a list of tuples where each tuple contains position, at_bats, ops, homeruns, innings_pitched, era, and whip. This effectively sorts the results variable into the positions.

get_weighted_hitting_averages(position_dict, stats_dict)

- Inputs: position_dict [dictionary], stats_dict [nested dictionary]
- Outputs: None
- Details: Uses the position_dict dictionary to calculate the weighted average stats for homeruns and OPS for each position. Stores the results in the stats_dict dictionary where the key is position and the value is a dictionary where the key is the stat type and the value is the weighted average of that stat.

main()

- Inputs: None
- Outputs: None
- Details: Utilizes the above helper functions to calculate the weighted average career stats per position. Graphs these weighted averages in a side by side bar chart.

For Plot3.py:
join_tables(team1, team2, team3, team4)

- Inputs: team1 [tuple], team2 [tuple], team3 [tuple], team4 [tuple]
- Outputs: results [list of tuples]
- Details: Each of the input tuples contains team name and team year. For each team name and year, uses a database join to extract team_name, position, at_bats, ops, homeruns, innings_pitched, era, and whip from tables Players and Statistics where the player id matches in both tables. Returns a list of tuples where each index in the list is for a different team and year and is a tuple containing the team_name, position, at_bats, ops, homeruns, innings_pitched, era, and whip.

get_team_stats(player_list)

- Inputs: player_list [list of tuples]
- Outputs: (team_ops, team_homeruns, team_era, team_whip) [tuple]
- Details: Gets the weighted average for a team's OPS, ERA, and WHIP while calculating the total number of homeruns for the team. The input is a list of tuples with each tuple corresponding to a player and their stats. The output is a tuple containing the team stats.

main():

- Inputs: None

- Outputs: None
- Details: Uses the above functions to calculate the weighted averages for OPS, ERA, and WHIp alongside the total for homeruns for each playoff team. Graphs 4 bar charts displaying these with teams having different colors.

For Plot4.py:
main()

- Inputs: None
- Outputs: None
- Details: Calculates the runs scored by the home team and runs scored by the away team for each game of various World Series. Displays the difference amongst the 7 games of World Series in a bar chart.

For Plot5.py:
plot_run_differentials(connection, cursor)

- Inputs: connection [database connection], cursor [database cursor]
- Outputs: None
- Details: Calculates the margin of victory for each team's games. Plots the average margin of victory against the frequency of occurrence for each team.

main()

- Inputs: None
- Outputs: None
- Details: Calls plot_run_differentials with database connection and cursor.

**Documentation of Resources Used:**

| Date | Issue Description | Location of Resource | Result (Did it resolve the issue?) |
|---|---|---|---|
| Entire Project | Get player career stats | https://rapidapi.com/theapiguy/api/mlb-data/ | Resolved the issue |
| Entire Project | Get team rosters and the player's year by year stats | https://appac.github.io/mlb-data-api-docs/ | Resolved the issue |
| Entire Project | Get the World Series scores/results | https://www.mlb.com/ scores Base url | Resolved the issue |

| | | | |
|---|---|---|---|
| Entire Project | Graphing help with matplotlib | GitHub Co-Pilot | Resolved the issue |
| December 4th | Plot 5 issues with formatting | ChatGPT | Resolved the issue |