Brian Good and Daniel Malis

GITHUB REPO: https://github.com/dmalis18/si206_final_project

In addition to your API activity results, you will be creating a report for your overall project.
The report must include:
1. The goals for your project including what APIs/websites you planned to work with and what data you planned to gather (10 points)

We wanted to look at MLB Draft Data and see if we could find any interesting trends related to team or draft pick success rates. Some of the questions we wanted to answer were whether higher draft picks reach the major league at a higher rate, are some teams better at drafting than others, what draft years had the most MLB players, and have teams gotten better at drafting over time. We planned to use the PyBaseball API and the MLB Data API. The Py Baseball API would be used to gather draft data, such as player name, team, overall pick, if they reached the majors, and their position. The MLB Data API would be used to see which teams actually existed in our entire window of 1990 until 2015 to make the data consistent.

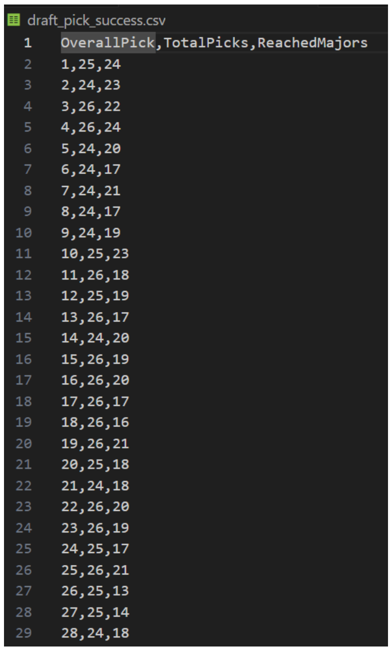
2. The goals that were achieved including what APIs/websites you actually worked with and what data you did gather (10 points)

Our goals were all met and we found conclusive answers to each of the 4 main questions we had. We used the PyBaseball API as planned to collect draft data and this proved crucial to reaching our goals about draft pick success rate and how well some teams drafted. From our data we found that higher draft picks tend to reach the major leagues at a much higher rate than lower draft picks which supports our initial hypothesis. We also found that most teams were relatively even in draft success rates, although the Yankees and Red Sox had the most draft success which did lead to 8 world series rings in a 25 year period. Most draft classes yielded the same number of major league players, but teams got better at drafting and signing players over time between 1990 and 2015.

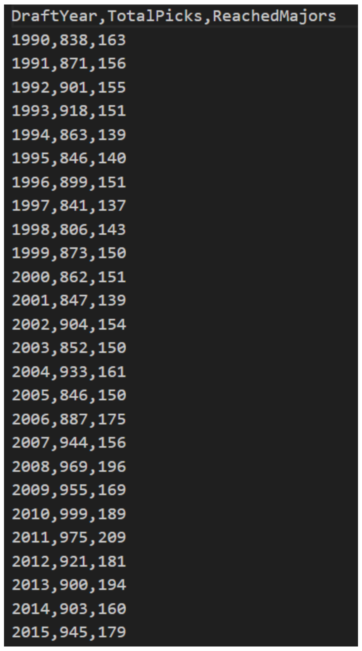
3. The problems that you faced (10 points)

The PyBaseball API had so much data that it was hard to find an alternative API and resource that had data we were missing. Additionally, we did not have full access to the MLB Data API Documentation which made it very difficult to gather useful data and find the correct requests. Lastly, multiple players share a name with other players which made finding meaningful statistics very difficult as it was hard to tell if two players were the same person. The only unique characteristic is what draft pick and year each player was selected.

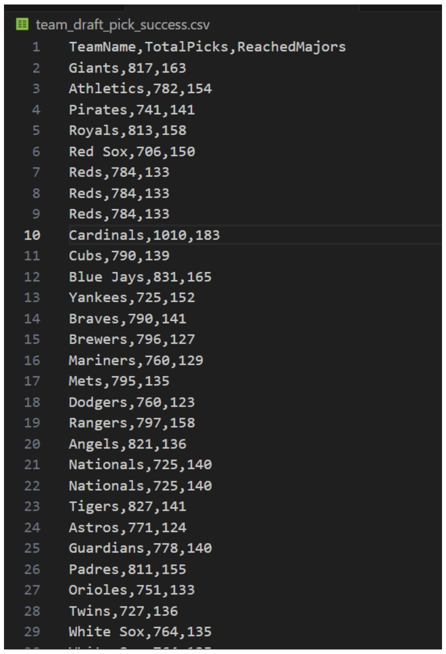
4. The calculations from the data in the database (i.e. a screen shot) (10 points)



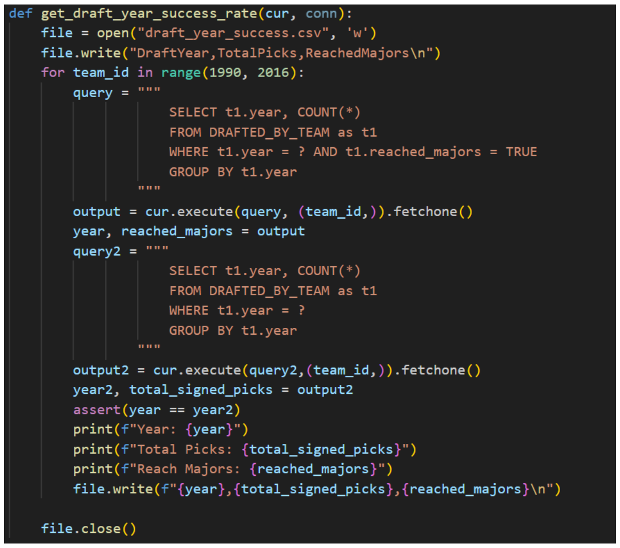
```
OverallPick,TotalPicks,ReachedMajors
1,25,24
2,24,23
3,26,22
4,26,24
5,24,20
6,24,17
7,24,21
8,24,17
9,24,19
10,25,23
11,26,18
12,25,19
13,26,17
14,24,20
15,26,19
16,26,20
17,26,17
18,26,16
19,26,21
20,25,18
21,24,18
22,26,20
23,26,19
24,25,17
25,26,21
26,25,13
27,25,14
28,24,18
```



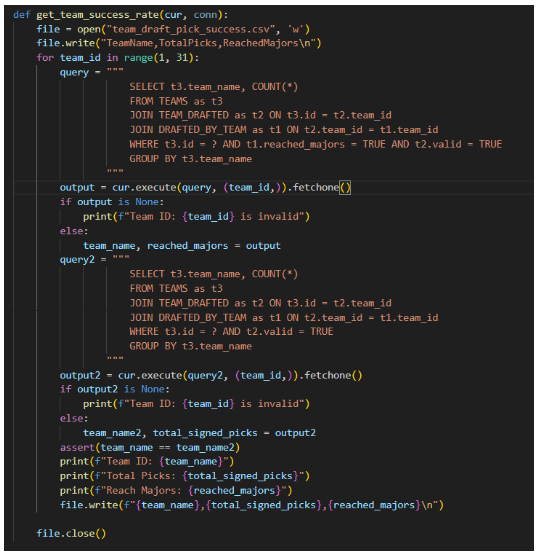
```
DraftYear,TotalPicks,ReachedMajors
1990,838,163
1991,871,156
1992,901,155
1993,918,151
1994,863,139
1995,846,140
1996,899,151
1997,841,137
1998,806,143
1999,873,150
2000,862,151
2001,847,139
2002,904,154
2003,852,150
2004,933,161
2005,846,150
2006,887,175
2007,944,156
2008,969,196
2009,955,169
2010,999,189
2011,975,209
2012,921,181
2013,900,194
2014,903,160
2015,945,179
```



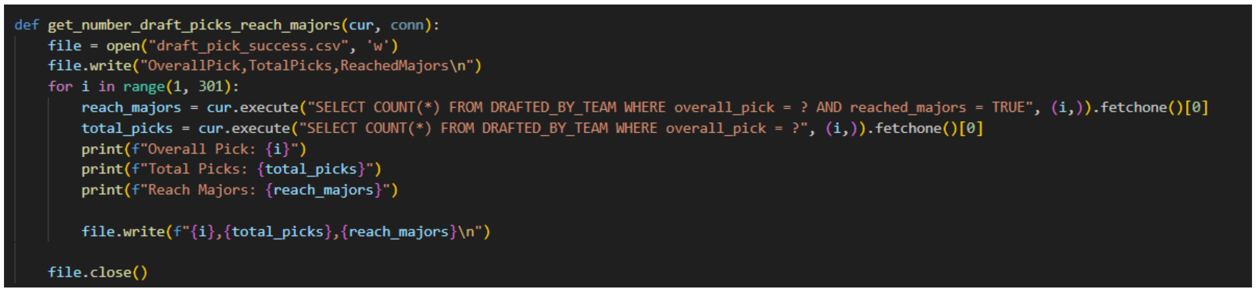
```
TeamName,TotalPicks,ReachedMajors
Giants,817,163
Athletics,782,154
Pirates,741,141
Royals,813,158
Red Sox,706,150
Reds,784,133
Reds,784,133
Reds,784,133
Cardinals,1010,183
Cubs,790,139
Blue Jays,831,165
Yankees,725,152
Braves,790,141
Brewers,796,127
Mariners,760,129
Mets,795,135
Dodgers,760,123
Rangers,797,158
Angels,821,136
Nationals,725,140
Nationals,725,140
Tigers,827,141
Astros,771,124
Guardians,778,140
Padres,811,155
Orioles,751,133
Twins,727,136
White Sox,764,135
```



```python
def get_draft_year_success_rate(cur, conn):
    file = open("draft_year_success.csv", 'w')
    file.write("DraftYear,TotalPicks,ReachedMajors\n")
    for team_id in range(1990, 2016):
        query = """
                    SELECT t1.year, COUNT(*)
                    FROM DRAFTED_BY_TEAM as t1
                    WHERE t1.year = ? AND t1.reached_majors = TRUE
                    GROUP BY t1.year
                """
        output = cur.execute(query, (team_id,)).fetchone()
        year, reached_majors = output
        query2 = """
                    SELECT t1.year, COUNT(*)
                    FROM DRAFTED_BY_TEAM as t1
                    WHERE t1.year = ?
                    GROUP BY t1.year
                """
        output2 = cur.execute(query2,(team_id,)).fetchone()
        year2, total_signed_picks = output2
        assert(year == year2)
        print(f"Year: {year}")
        print(f"Total Picks: {total_signed_picks}")
        print(f"Reach Majors: {reached_majors}")
        file.write(f"{year},{total_signed_picks},{reached_majors}\n")

    file.close()
```

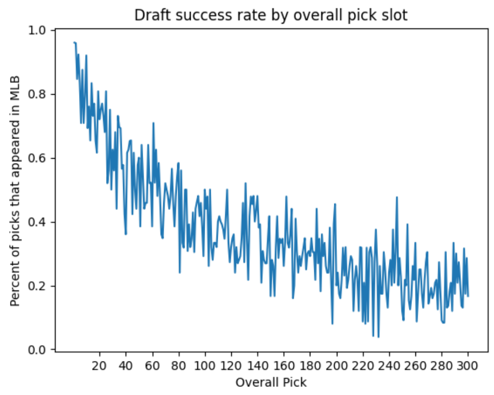


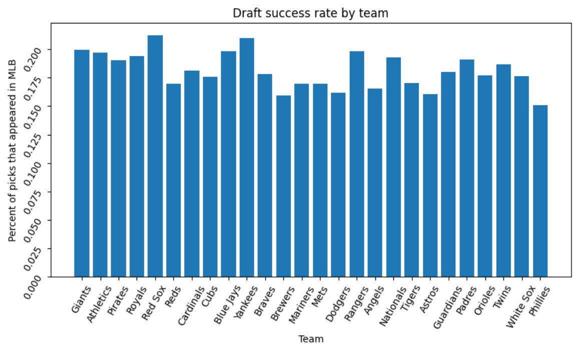
```python
def get_team_success_rate(cur, conn):
    file = open("team_draft_pick_success.csv", 'w')
    file.write("TeamName,TotalPicks,ReachedMajors\n")
    for team_id in range(1, 31):
        query = """
                    SELECT t3.team_name, COUNT(*)
                    FROM TEAMS as t3
                    JOIN TEAM_DRAFTED as t2 ON t3.id = t2.team_id
                    JOIN DRAFTED_BY_TEAM as t1 ON t2.team_id = t1.team_id
                    WHERE t3.id = ? AND t1.reached_majors = TRUE AND t2.valid = TRUE
                    GROUP BY t3.team_name
                """
        output = cur.execute(query, (team_id,)).fetchone()
        if output is None:
            print(f"Team ID: {team_id} is invalid")
        else:
            team_name, reached_majors = output
        query2 = """
                    SELECT t3.team_name, COUNT(*)
                    FROM TEAMS as t3
                    JOIN TEAM_DRAFTED as t2 ON t3.id = t2.team_id
                    JOIN DRAFTED_BY_TEAM as t1 ON t2.team_id = t1.team_id
                    WHERE t3.id = ? AND t2.valid = TRUE
                    GROUP BY t3.team_name
                """
        output2 = cur.execute(query2, (team_id,)).fetchone()
        if output2 is None:
            print(f"Team ID: {team_id} is invalid")
        else:
            team_name2, total_signed_picks = output2
        assert(team_name == team_name2)
        print(f"Team ID: {team_name}")
        print(f"Total Picks: {total_signed_picks}")
        print(f"Reach Majors: {reached_majors}")
        file.write(f"{team_name},{total_signed_picks},{reached_majors}\n")

    file.close()
```

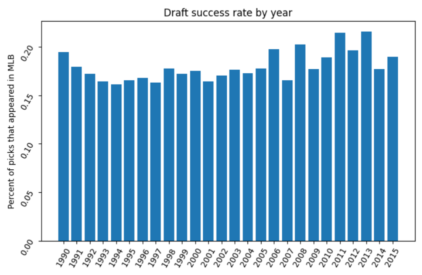


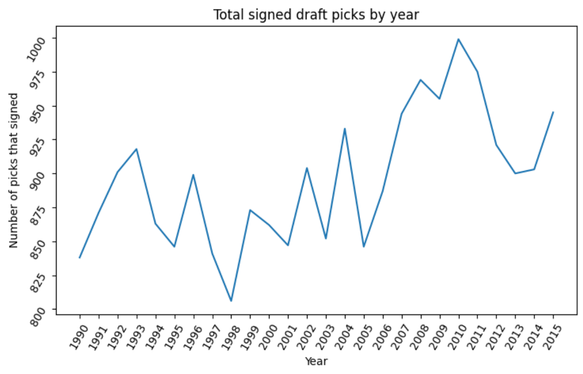
```python
def get_number_draft_picks_reach_majors(cur, conn):
    file = open("draft_pick_success.csv", 'w')
    file.write("OverallPick,TotalPicks,ReachedMajors\n")
    for i in range(1, 301):
        reach_majors = cur.execute("SELECT COUNT(*) FROM DRAFTED_BY_TEAM WHERE overall_pick = ? AND reached_majors = TRUE", (i,)).fetchone()[0]
        total_picks = cur.execute("SELECT COUNT(*) FROM DRAFTED_BY_TEAM WHERE overall_pick = ?", (i,)).fetchone()[0]
        print(f"Overall Pick: {i}")
        print(f"Total Picks: {total_picks}")
        print(f"Reach Majors: {reach_majors}")

        file.write(f"{i},{total_picks},{reach_majors}\n")

    file.close()
```

5. The visualization that you created (i.e. screen shot or image file) (10 points + 30 points for bonus visualizations)









6. Instructions for running your code (10 points)

1. Open the main.py file
2. Make sure you pip install PyBaseball, sqlite3, os, requests, numpy, csv, and matplotlib
3. Run the get_all_needed_draft_data and read_active_teams functions to make the necessary API Calls that stored our data into csv files first
4. Use the various populate functions to fill the database with the respective data from the APIs 25 entries at a time until the entire csv file is written to the database
5. Use the get functions to get calculations from the databases
6. Use the plot functions to create the visualizations
7. Run the current state to fill data into the database, make the respective queries, and plot various data on graphs

7. Documentation for each function that you wrote. This includes describing the input and output for each function (20 points)

Completed in code

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	Issue Description	Location of Resource	Result
11/27/23	Creating a venv in Ubuntu	https://www.linode.com/docs/guides/create-a-python-virtualenv-	Solved the issue

		on-ubuntu-18-04/	
11/27/23 + 12/1/23	Documentation for Pybaseball API	https://github.com/jldbc/pybaseball	Resolved issue
12/1/23	Documentation for MLB Data API	https://appac.github.io/mlb-data-api-docs/	Resolved issue
12/1/23	DB Browser installation	https://sqlitebrowser.org/	Resolved issue

Coding:

Most of the coding was done as peer programming