Data Visualization Project — Baseball

Due November 8th, November 15th, November 29th at 11:55pm

(70, 65, 65 points)

Objectives:

- Use the skills that you have acquired thus far to design a program in a modular way
- Become familiar with graphing and data visualization in python
- Implement a larger program that works with real-world data

Turn In:

- (November 8th) `pseudocode.pdf`, `baseball.ipynb`
    - pseudocode (Task 1)
    - reading data; utility functions (Task 2)
    - **All partner choices at this point are final**
- (November 15th) `baseball.ipynb`, pdfs of the 4 graphs that you produced
    - Task 3
- (November 29th) `baseball.ipynb`, pdfs of the 4 graphs that you produced, plus pdfs of any graphs you produced for the extra credit.
    - Task 4
    - Extra credit

Turn these in to the corresponding links on moodle.

Your code must be commented at each deadline. At each deadline, it must run without errors. You may not turn in any tasks late. Your code must follow the guidelines set forth in previous homeworks in terms of style and structure.

You may work with a partner if you choose to, but **you may not create or switch groups after the first checkpoint.**

<u>You will not receive a final grade for this project if you do not come to interview grading following the final checkpoint.</u>

What is given:

You are given the file `battingData1950Present.csv`. This is a comma separated file with 74,501 rows. It has one row of headers and the rest of the rows correspond to statistics for a given player during a given season with a given team.

Notice that 1) not all columns are the same type and 2) one column ("sacFly") doesn't always have a value. We'll have to deal with this later.

```
battingData1950Present.csv (first 5 rows)
```

```
yearID,playerID,nameFirst,nameLast,name,Games,AB,R,H,doubles,triple
s,HR,RBI,BB,HBP,stolenBases,caughtStealing,SO,sacFly,position
1950,aberal01,Al,Aber,"Cleveland
Indians",1,2,0,0,0,0,0,0,1,0,0,0,1,,P
1950,abramca01,Cal,Abrams,"Brooklyn
Dodgers",38,44,5,9,1,0,0,4,9,0,0,0,13,,OF
1950,adamsbo03,Bobby,Adams,"Cincinnati
Reds",115,348,57,98,21,8,3,25,43,0,7,0,29,,2B
1950,adamsbo03,Bobby,Adams,"Cincinnati
Reds",115,348,57,98,21,8,3,25,43,0,7,0,29,,3B
```

BattingData object:

We provide you with the following code that will help the legibility of your project by giving names to column numbers. This code appears in the starter notebook that we've provided. See lecture 17 and lab 10 for examples of how to use a similar object.

```
class BattingData:
     year = 0
     player_id = 1
     first_name = 2
     last_name = 3
     team_name = 4
     games = 5
     at_bats = 6
     runs = 7
     hits = 8
     doubles = 9
     triples = 10
     home_runs = 11
     rbi = 12
     walks = 13
     hbp = 14
     stolen_bases = 15
     caught_stealing = 16
     strike_outs = 17
     sac_flies = 18
     position = 19
```

Task 1: Your Plan (Thursday, November 8th at 11:55pm)
Read through the rest of this write-up. Before you begin implementing this project, write down a plan of how you are going to accomplish each step. You should write your plan as pseudocode,

with the exact format up to you. You should be as detailed as possible. This plan must cover **all** tasks in this project through task 4b. It must be **at least** 1 page/ 50 lines long.

Task 2: Read the data and create the user interface (Thursday, November 8th at 11:55pm)
2a. main() function

- Always begin your code by writing out the main() function.
- From your main() call a function named *read_data()* that takes no argument but returns a list of lists, one list per row in your original file, excluding the headers.
- When you have read in your data, this is a great place to check its dimensions.

2b. read_data(): How should we read the data in the csv file?

- You will first have to provide the path to where your "data" folder is located on your computer.
- Read all the batting data from all the file in the "data" folder.
- *What mode will you open this file in at this step?*
- You need to read in every row from the csv file, convert its values to the appropriate type, and append it to a list of lists.
    - Values that are numbers should be converted to ints
    - Values that are empty strings should be converted to the integer 0
    - Values that are strings (player ids, player names, team names, positions) should be left as strings
    - Warning! Make sure that you are not changing the order of the values in your rows!
- Make sure to ignore the headers when you read the data
- You need to return this list of lists to the main( ) function

2c. Make sure all columns of our array have values and ensure that your data is the correct types

After you have read in your data, take a look at some of the rows of your list of lists. All columns should have values, all columns that are numbers should be type int, and all columns that are strings should be strings.

Make sure you understand how to access each column of your array using the variables in the BattingData object (provided).

2d. Implement your User Interface (UI)
After reading the data the next step to would be to provide a user interface so that the user can choose what they would like to do with the given data!

Define a function named *get_menu_choice()* that takes a list of strings (the options that the user can pick between) and returns the choice entered by the user.

If the user enters an invalid choice, the program should exit.

In the final version of this project, the user will be able to choose a number and see the results of their choice (discussed in later tasks).

<u>At this point, you don't need to have the functionality behind the options implemented, but your menu should otherwise work. We recommend using print statements to make sure that your program is correctly routing the user's choice.</u>

Example:

```
1: Runs scored in a given year for all players (no cutoff)
2: Runs scored in a given year for all players (cutoff = 100)
3: Graph team presence over time
4: Plot homeruns over time (percentiles)
5: [FILL ME IN]
6: [FILL ME IN]
7: Extra Credit #2 (if implemented)
8: Extra Credit #3 (if implemented)

> 1
```

2e: Utility functions to deal with your lists of lists.

Lastly for this task, you will implement 4 utility functions that will help you greatly in the rest of your project.

- **get_matching_rows**: takes your list of lists, an integer column index, and a target value. Returns a new list of lists that contains all rows from your data where the given column's value matches the target value.
- **get_column_values:** takes your list of lists and an integer column index. Returns a new list that contains the values of the target column from all of the rows.
- **get_unique_values:** takes a list of values. Returns a new list that contains all unique values in the original list.
- **get_unique_column_values:** takes your list of lists and an integer column index. Returns a new list that contains the unique values of the target column from all of the rows. (feel free to call your other utility functions to accomplish this!)

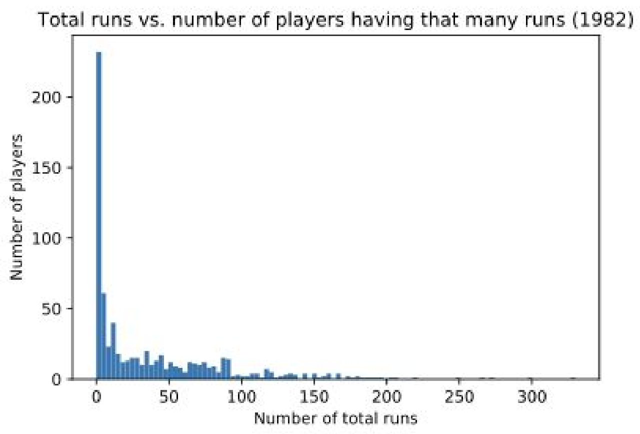
Task 3: Making your first graphs! (Thursday, November 15th at 11:55pm)

For this task your will implement the functionality behind the first 2 options in your menu.

3a: Histogram of total runs scored of all players for a specific year (no cutoff, no positions)

Here, you will be creating a histogram with 100 bins for the number of runs scored in a specific year for all players. For the given year, for each player in the dataset, you will need to sum the runs from all rows belonging to them to calculate their total runs.

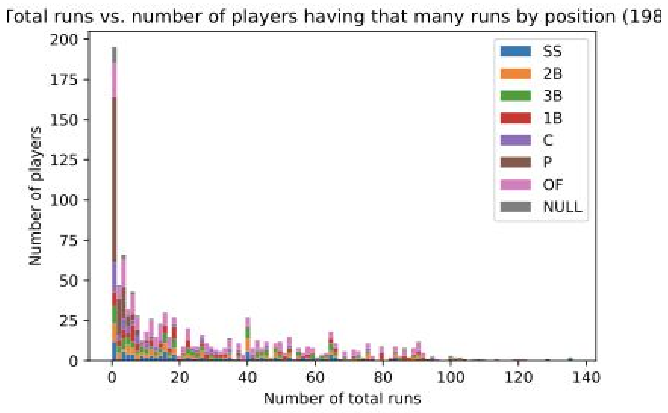
This is because some players play multiple positions or for multiple teams in one given year.



Then, you will take a list of all of the total run values, and give these values to matplotlib's hist() function.

To accomplish this, you will find the get_matching_rows and get_unique_column_values functions useful.

3b: Histogram of runs scored in the lifetimes of all players (no cutoff, WITH positions)
Your next task is the same as number 3a, but this time, we will separate the data based on the position of each player. We would like to group all Pitchers together, all Catchers together, all Short Stops together, and so on and so forth.



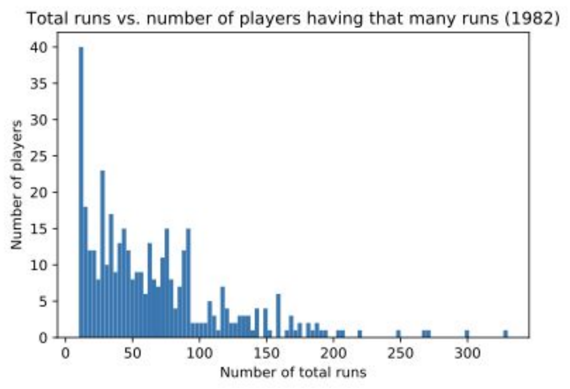
For each player having each position, you will do as you did in 3a and sum their runs over their lifetimes. Once you have all the lifetime runs for all the different positions, you can once again use matplotlib's hist() function to create a stacked histogram. You should continue to use 100 bins.

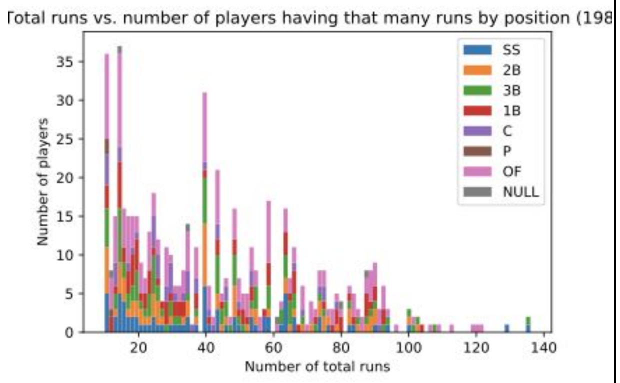
You will see that some players have position "NULL". These are season/player combinations for which we are missing the data. You don't need to worry about this.

3c: Runs scored in the lifetimes of all players (with a cutoff but without positions AND with a cutoff and positions)
Your next task is the same as numbers 3a and 3b, but this time we will impose a cutoff, meaning that we will only be graphing the data from players whose total lifetime runs value is

greater than the cutoff. We've shown you a graph with cutoff set to 10, but you should design your function so that you can pass in any number as the cutoff.

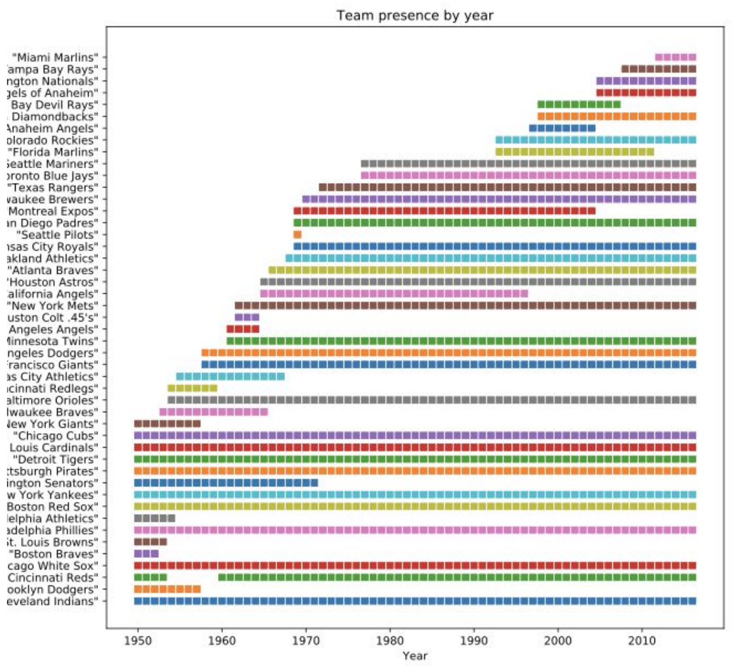
For this task, you should modify the functions that you wrote for 3a and 3b so that you can use them for either no cutoff (equivalent to cutoff = 0), or with a cutoff! Don't write new functions!





Task 4: Graphing the rest of your graphs! (Thursday, November 29th at 11:55pm)

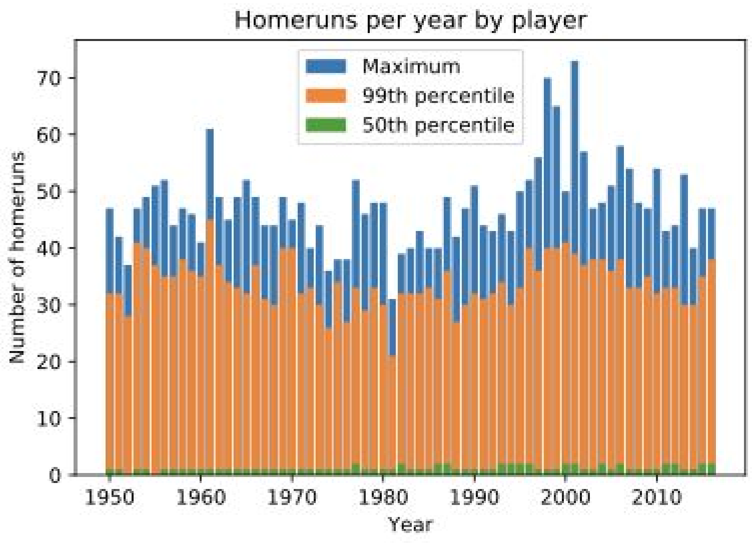
Task 4a: Creating a time plot of when teams were active



For this task, you will be creating a scatter plot showing when each team was active. The y-axis will be labeled with team names and the x-axis will be the years that they were active. A team is considered to be active during a given year if there is any player from that team with statistics for that year. It doesn't matter to us what order the teams are in, but the data that you graph should be correct.

Feel free to use whatever marker you like best in your scatter plot. Make sure to look at the examples for creating scatter plots in the matplotlib document.

4b: Plot homeruns over time (percentiles)
For this task you will create a bar graph with the following series (graphed in this order) : the maximum number of homeruns scored every year, the 99th percentile of # of homeruns scored every year, and the 50th percentile of homeruns scored every year. Take a moment to think about what this data tells us.



4c: Explore the data and choose two more statistics to create graphs of.
Add these to your menu as options corresponding to choices 5 and 6. At least one of these graphs should be a kind of graph that you have not yet used in this project (something other than a histogram, bar plot, or scatter plot).

Extra Credit (up to 20 points, due on Thursday, November 29th at 11:55pm):

Some of the extra credit questions should be answered in a separate document (`extra_credit.pdf`), some in your code. Each question indicates where you should answer it.

1) **Analyzing the graphs that you created for task 3. (total runs vs. number of players having that many runs, with and without cutoff, not separated and separated by positions).**

a) (`extra_credit.pdf`) When you compare two graphs from the same year, one separated by positions, one not, what is else is different about these graphs?
b) (`code`) Write code that helps illuminate the sources of these differences; produce interpretable output and comment your code for how this helps you.
c) (`extra_credit.pdf`) What has the code that you have written revealed about the differences between these graphs?

2) **A player's on base percentage is a metric often used to measure how good a player's season is.**

On base percentage is calculated according to: (hits + walks + hit by pitch) / (at bats + walks + hit by pitch + sacrifice flies)

Consider one row in your data set to be equal to one season. (for the purposes of this question, we will say that players who switched position or team mid-season had "multiple" seasons). Find the 20 best seasons according to on base percentage.

Next, find the 20 best seasons if you limit the data so that you only consider seasons that are at at least the 25th percentile for at bats and at least the 25th percentile for games played. How does this change your answer?

Write at least 3 sentences in the comments explaining the results from these two calculations and how they changed. Which list of seasons do you think should be used when considering best seasons?

3) (`code`) **Create a line graph that visualizes on base percentage over a single player's career. Make sure to only graph one point per year! (Think about what to do if a player has multiple rows for a given year!)**

4) (`extra_credit.pdf`) **Analyze the graphs that you created for task 4. What conclusions can you make from these graphs? What sort of additional evidence might you want to support these conclusions? How could you improve these graphs to make them more easily readable?**