CS 109 Final Project

Conditional Expectation of 4x4 Boggle Simulation

By Kyle Orciuch & Max Lau

Question: Given that the board has the condition $\{A = a,\ B = b,\ \ldots,\ Y = y,\ Z = z\}$, where a, b, … , y, z, etc. are the number of times each letter appears on the board, what is the conditional expected score based on 10k+ Boggle game simulations?

Methods: We'll start with explaining how we randomly set up the Boggle board for each simulated game. The following list of 16 uppercase strings represents the 6-sided dice which will be randomly shuffled:

```
cubes = [ 'AAEEGN', 'ABBJOO', 'ACHOPS', 'AFFKPS', 'AOOTTW', 'CIMOTU',
'DEILRX', 'DELRVY', 'DISTTY', 'EEGHNW', 'EEINSU', 'EHRTVW', 'EIOSST', 'ELRTTY',
'HIMNQU', 'HLNNRZ']
```

We will assume that the 16 dice can be placed into each position with equal probability and that the random selection of each die face will also occur with equal probability. The board is represented as a 4x4 two dimensional array that looks like this:

```
{{O, O, O, S}, {R, R, E, T}, {M, T, C, L}, {N, E, S, E}}
```

After the board has been randomly generated, we call our recursive backtracking function to conduct a depth-first search to exhaustively find all possible words that are greater or equal to length 3 and are found in the provided dictionary. Here is an example of our console output from boggle.py:

```
********* Running Game:  0  *********
['ELRTTY' 'ABBJOO' 'EEINSU' 'DISTTY' 'CIMOTU' 'AOOTTW' 'DEILRX' 'HLNNRZ'
 'HIMNQU' 'EHRTVW' 'AAEEGN' 'EEGHNW' 'ACHOPS' 'EIOSST' 'DELRVY'
'AFFKPS']
[['T', 'A', 'N', 'I'], ['I', 'A', 'L', 'N'], ['I', 'R', 'E', 'G'], ['S', 'S', 'D', 'A']]
all words:  {'glairs', 'laird', 'sealing', 'laired', 'airless', 'lingers', 'alar', 'serai', 'sears',
'gleds', 'ani', 'nares', 'algae', 'tared', 'lards', 'arles', 'ana', 'dreg', 'dean', 'taler', 'siren',
                                        ... (truncated output)
'sera', 'liner', 'lean', 'rales', 'renin', 'tirling', 'dear', 'lin', 'lines', 'led', 'tales', 'alined',
'nines', 'iris', 'egads', 'talar', 'lariat', 'ire', 'lears', 'lard', 'lars', 'glen', 'sis', 'airline', 'ired',
'ale', 'ged', 'ais'}
number of words:  253
score:  476
********* Running Game:  1  *********
```

```
['EEINSU' 'EEGHNW' 'EIOSST' 'ELRTTY' 'CIMOTU' 'ACHOPS' 'AFFKPS' 'AAEEGN'
 'HLNNRZ' 'DEILRX' 'HIMNQU' 'EHRTVW' 'DISTTY' 'ABBJOO' 'AOOTTW' 'DELRVY']
[['I', 'N', 'O', 'Y'], ['M', 'A', 'F', 'G'], ['H', 'R', 'U', 'V'], ['S', 'B', 'T', 'R']]
all words:  {'oaf', 'tug', 'guano', 'arf', 'foy', 'fubs', 'sha', 'ram', 'harmin', 'train', 'mart',
'turfy', 'tufa', 'goa', 'bran', 'fain', 'harm', 'yogurt', 'tubs', 'inarm', 'arbs', 'info', 'furan',
'trug', 'oar', 'urb',
                                          ... (truncated output)
'gonia', 'guars', 'bra', 'bur', 'buran', 'rut', 'ham', 'shrub', 'oars', 'turf', 'shrug', 'brain', 'furs',
'fur', 'main', 'amin', 'rami', 'haut', 'foam'}
number of words:  104
score:  123
```

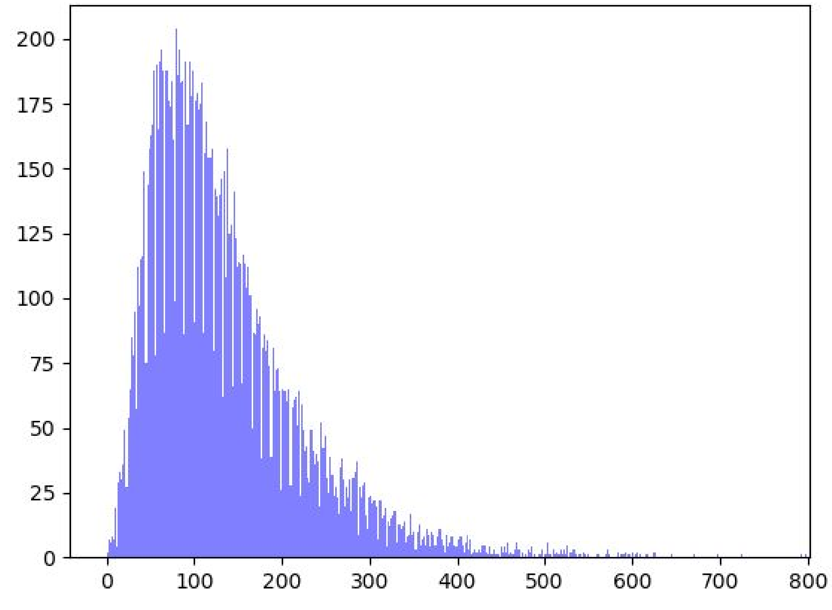
As our simulated games are completed, we send our results to a .txt log file to store our results for future use (~10 mb worth). Since the runtime was pretty expensive, we wrote merge_data.py to merge log files from different runtimes so that either one of us could run the boggly.py program and combine our log files.

Findings:

***Note: y-axis corresponds to # of games, x-axis corresponds to score**

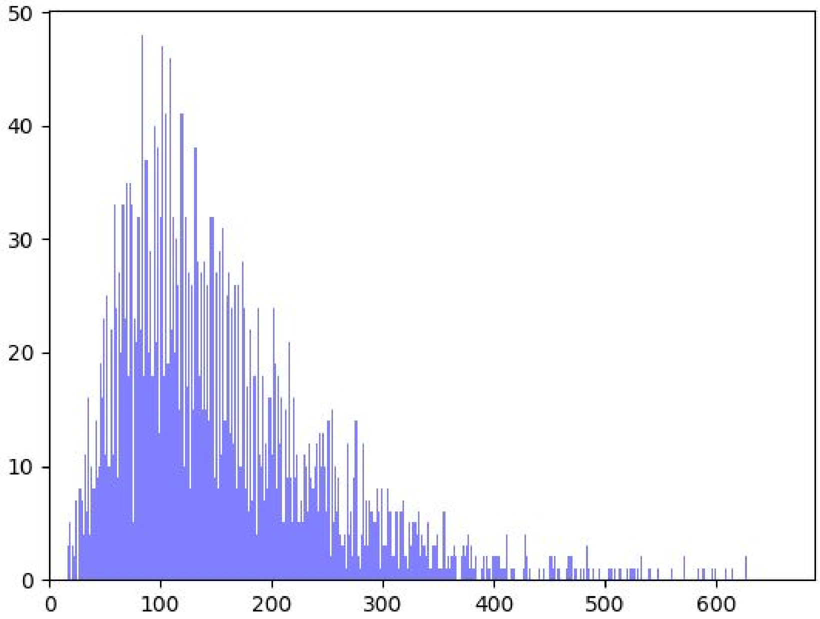
Sample Output:

```
C:\Users\maxus\Desktop\109\Boggle\cs109_final_project-main>py Merge_data.py
Number of simulated games:  15057
```



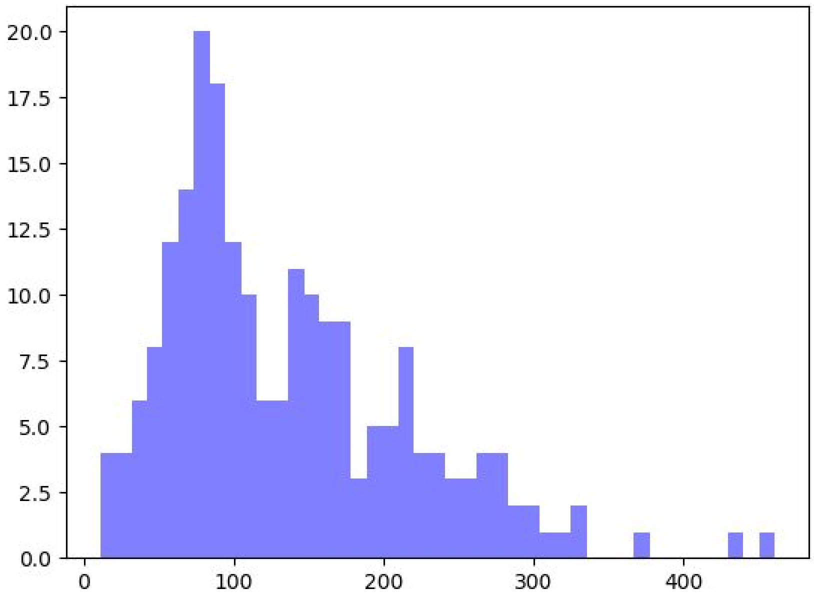
```
**********************************
Welcome to the Boggle Board Bash
Sample usage: aab = all boards with 2 A's and 1 B.
Press * to quit.

aa
```

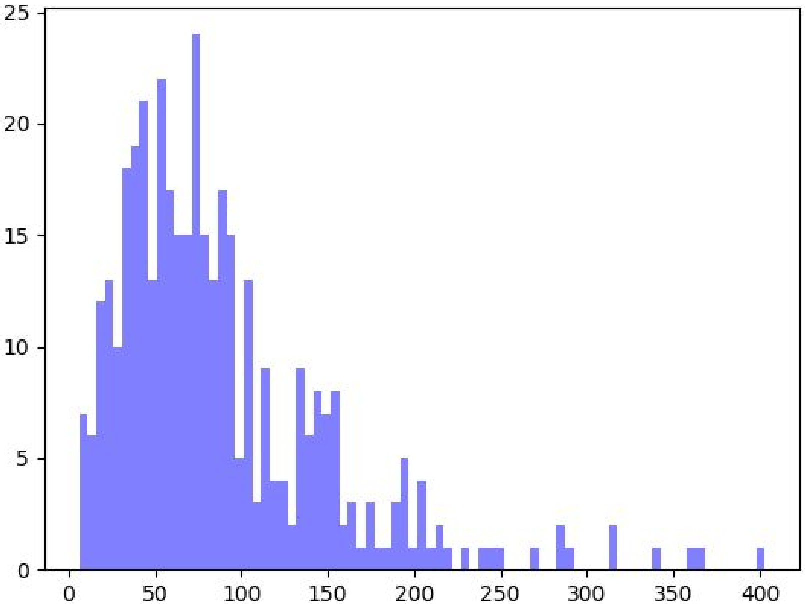


```
Number of boards with condition:  3126
Expected score:  156.84293026231606
```

abcd



Number of boards with condition: 213
Expected score: 136.4882629107981
qz

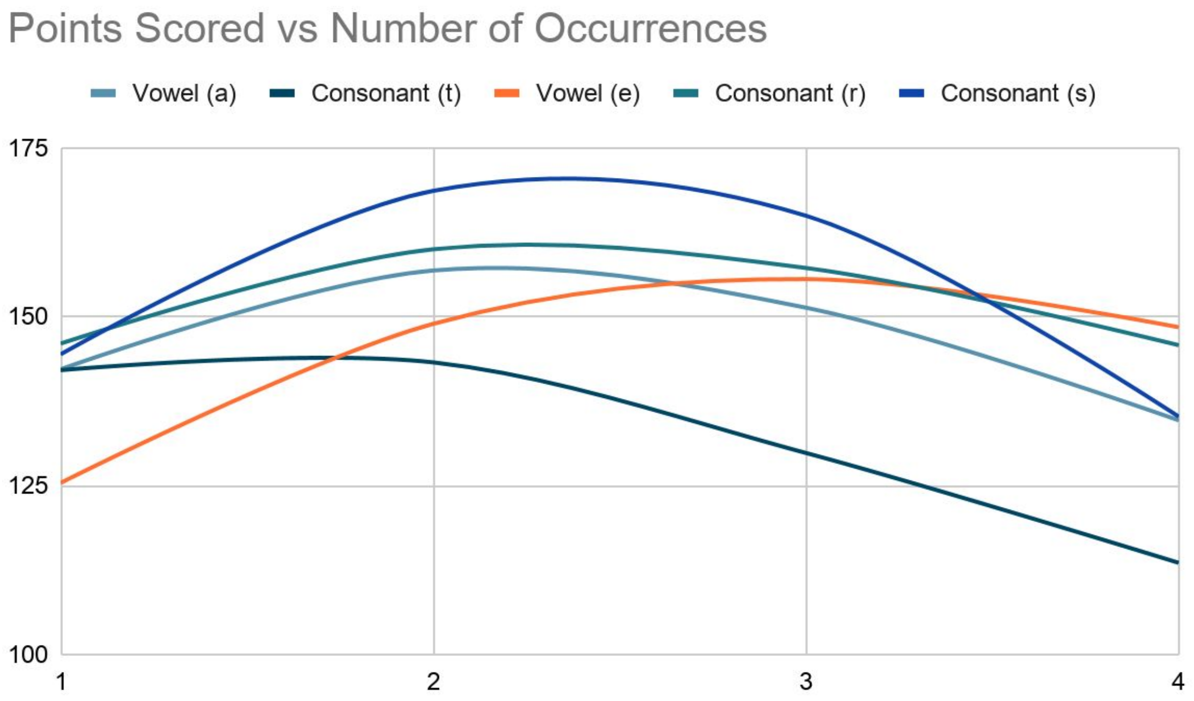


Number of boards with condition: 392

```
Expected score:  87.7984693877551
bb
Number of boards with condition:  0
Expected score:  0
*
```

End Sample Output

Conclusion: The expected value with or without any given conditions is shaped like a right-tailed Poisson Distribution. One of the biggest takeaways from this project is that the expected score value appears to be positively correlated given an increasing number of occurrences of letters on a Boggle board from roughly 1-2 occurrences, and negatively correlated given roughly 3-4 occurrences. Our observations show that in order to maximize the expected score, it would be desirable to get a board with no more than 3 occurrences of a particular letter; however, we acknowledge that more replication is necessary to say for sure.



We've also observed that inputting more conditions will decrease the sample size of the distribution and won't be as helpful as inputting only a couple letters. If we were to do this again with more time, we would focus on getting more data so that the data queries with more than 4 letters could still provide high fidelity calculations.