Task: Develop an algorithm that predicts the optimal starting word for the game Wordle based on word entropy, and solve the game in the minimum possible number of trials.

Solution:

To solve Wordle, we assign each word a probability of occurrence for each possible pattern, based on a uniform distribution over the set of allowed words. We then order the words by their probability of occurrence and try them successively while assimilating previous information on the letters present/absent in the solution word. The algorithm aims to minimize the number of trials required and takes an average of four trials to reach convergence.

We limit our guesses to the set of allowed words using the corpus extracted from Wordle's website. We restrict the sample space further by obtaining the list of possible solutions and the words the game accepts as guesses.

In our first attempt, we naively assumed that entropy was the best metric for ranking the words. The algorithm starts with a user input opener, removes incompatible words from the sample space, computes a new ranking for the remaining compatible words, picks the best-ranked word, and repeats these steps until finding the solution or reaching nine attempts.

The greedy algorithm chooses the word that reaches the maximum value of { g(r_w) · H(X_w) }.

We simulated all 2309 games to test the performance of our algorithm against each case.

By using vectorized operations inside Python's Numpy module, we make use of SIMD instructions and decrease execution times substantially instead of having to iterate over every single letter of all possible words(5*12792 iterations).

Problem- Iteration (0):

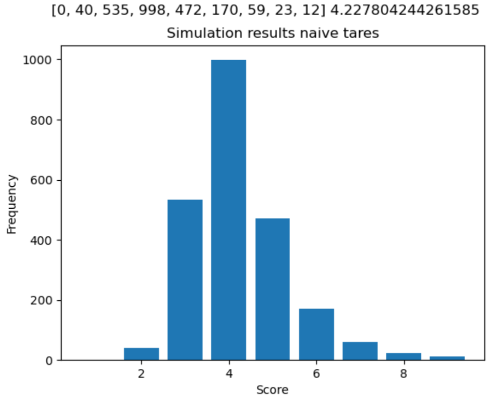
> Find the solution to any Wordle puzzle in as few attempts as possible using the words from the list of allowed words

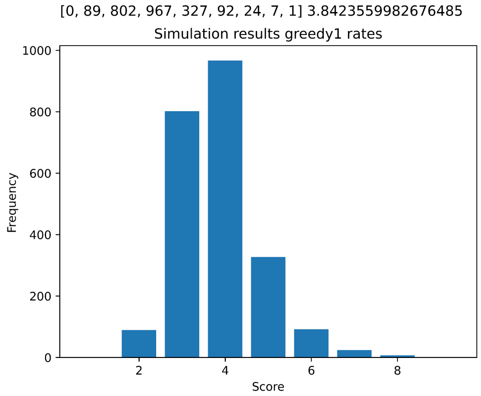
Problem - Iteration 1.0:

> Find the solution to any Wordle puzzle in as few attempts as possible using only words from the list of *solutions*.

We obtained a significantly better score for problem P1 after testing our Naïve Algorithm. We removed many words from the set of allowed words, which removed uncertainty from the problem. Our score was better than the Greedy Algorithm and P0.

Results:





Incorporating word frequencies significantly improved the average scores and changed the best starters. The highest-ranked starter was RATES, which was almost identical in structure to TARES but was a much more common word. The average score for RATES was 3.843, and less than 1% of the solutions required more than six attempts.

Problem P1 obtained a score as low as 3.527 after opening with SLATE and using the Naïve Algorithm, missing only seven solutions. While using the frequency of words improved the scores significantly, the function g could not be adjusted to reach such good scores because the solution set had been handpicked.

Future directions include optimality analysis, developing a robust and time-efficient algorithm, using decision tree/genetic algorithms to further increase accuracy, and controlling for more frequent starter words.

References:

1. https://arxiv.org/pdf/0903.1659.pdf
2. Solving Wordle using information theory