**Notes on Math Modeling Problem**

* The objective of Worlde is guess a real 5 letter word using a maximum of 6 attempts. These words must actually exist in the English language, so any guesses are that aren’t actual words don’t count. **An idea I have is for the guess, have their be a legitimate word that is mostly composed of vowels. This could help narrow down which vowels are included quickly**
* When you look at that picture in Figure 1, youre supposed to read each row as a guess the person has entered. There are 3 colors that will appear depending on your guess.
  + If the tile is green that means that letter is **both** included in the actual word **and** is in the correct location.
  + If the tile is yellow, that means that the letter is included in the word, but is in the **wrong location**.
  + Finally, is the tile is gray, then the letter is not in the word at all.
* There are two modes: “Regular” and “Hard”.
  + The Regular mode puts no restrictions on what words you are or aren’t allowed to guess. For example, say the word was “apple” and you guess “lions”, the letter “l” in your guess would appear yellow, but you there is nothing preventing your next guess from being “piano” even though that doesn’t have “L” in it at all. **This is nice because you can use any words in your next guesses to find more information.**
  + **Hard Mode** places the restriction on your future guesses if you find a letter that is located in the real word (either yellow or green), meaning that all your following guess **must** include those letters. Figure 1 was played on hard mode.
* Included in this problem is a data set that includes information on the success rate of these word puzzles.

**Requirements for the problem**

1. Every day, the number of people who get the correct answer in 1, 2,3, 4, 5, 6, or not at all changes. Create a model that can **a)** explain this variation and **b)** create a prediction interval for the results reported on March 1, 2023. **This sounds like a time series classification model. We need to be able to classify a person in to groups ranging from 1-6 to correspond with the number of guesses it took for them to determine the word and an X group for those who didn’t get it at all, so a total of 7 groups.**
   1. From here, examine what attributes of the word of day made it more difficult for Hard Mode players to guess (It sounds like every player receives the same word regardless of mode). Were there any times where the Hard Mode players did better than the regular ones, if so why/how?
2. Develop a model that allows you to **input a date** and **a word**  and based on these two inputs, output the success percentages for each of the 7 groups.
   1. Explain what uncertainties are associated with your model and its predictions. **Maybe during the periods where most students are in school like January-April and September-December the performance is better.**
   2. Produce a prediction for the word “EERIE” on the date March 1, 2023.
   3. Finally produce a confidence interval for each of the group success percentage outputs. Use this to explain how confident you are in your model’s performance.
3. Develop a model that can measure a word’s difficulty. Have the output of the model explain which characteristics of the word contributed to its difficultly like “Number of vowels:”, “Number of “rare” letters:” (Define what a rare letter is), “Overall difficulty:“ (**Figure out what makes a word more difficult. It is the number of values, the inclusion of certain letters, an over abundance of specific letters? Consider taking every solution word in your data set, look at each letter included in that word and then compare that word to the amount of successes that occurred. Maybe you could take every solution word and create a data set that has 26 entries. The first column is the letter. The second column is the number of times that letter occurred. From there, you can determine which letters occur most often. Maybe you can somehow turn this into a likelihood that a word contains a specific letter. Then for each word, youd have a function that assesses each letter in that word, determines how likely that individual letter is to be present in any word, and then at the end of the word have a value \*that will be bigger than 100%\* and from there have that be the score for how “hard” a word is. You can then make groups somehow. Maybe scale that “difficulty score” so that all numbers are between 0 and 100. From there, you could plot each word and its score to visualize a distribution. Then the closer the score is to the mean/median the lower the difficult is. Kind of like Z scores. Like a word that is 1 Z-score away is considered “easy”, 2 Z-scores is medium, more than 2 z scores is hard**)
4. Describe some interesting features of the data set. ***This would be something like your “School = Better Performance” theory.***
5. In addition to your 25 max page analysis of this problem, write a 1 to 2 page letter to the New York Times that summarizes your results, gives a prediction for the word “EERIE” on March 1, 2023. **Your names SHOULD NOT appear on this letter. Actually sound like youre talking to someone. Be professional, but it should sound like one side of a conversation. Use the first person like “we have discovered that…” or “our biggest concern is that…”. Don’t sound robotic.**

**Side Notes:**

* Consider making a column that accounts for the number of people who played on the “regular” mode by just doing “Number of Reported results” – “Number in hard mode”. I don’t know if you would need to scale or separate the data so that you have success percentages for those in the hard group. **The biggest problem I see with doing that, is that it assumes that those who completed the problem on hard mode had an equal distribution on the number of attempts before a success. This is a major assumption because that would imply that being on a harder mode doesn’t make you perform worse. I am not a fan of this data splitting idea, but I don’t know how we’re supposed to determine which words were more difficult for the platers on hard mode without doing something like that.**