

# Cracking Wordle

DATA201 | LGCC  
Evelyn Kim

# About Wordle

- Daily game
- Guess the word
- English words
- 5 Letters
- 6 tries
- Strategy is key

## My Approach

- vowels
- hope

Guess the **WORDLE** in 6 tries. ✕

Each guess must be a valid 5 letter word. Hit the enter button to submit.

After each guess, the color of the tiles will change to show how close your guess was to the word.

### Examples

**W** E A R Y

The letter **W** is in the word and in the correct spot.

P **I** L L S

The letter **I** is in the word but in the wrong spot.

V A G **U** E

The letter **U** is not in the word in any spot.

A new **WORDLE** will be available each day!

## My Interest

A few weeks ago, I had no idea what Wordle even was. I'd been seeing people post a sequence of colored squares across social media, but had no idea what it meant.

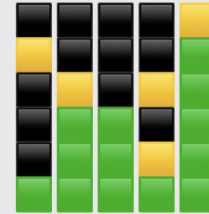
I finally bit the bullet and looked it up to see what all the buzz was about. I soon learned it's a fun game to play and share results with friends.

Most people I know play this game (my cousins and I even have a group chat to share our daily scores)

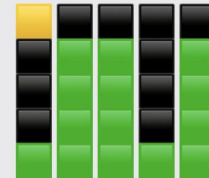
Wordle 209 4/6



Wordle 209 6/6



Wordle 209 5/6



# Research Question

What is the best strategy to start off my daily Wordle?

## Goals

- find a data-backed approach
- examine as many 5-letter English words as possible
- find helpful patterns

---

# Variables of Interest

## General Information

- Number of 5 letter words
- Letter frequency
- Common prefixes/suffixes
- Number of words with repeating letters

## Good Words

- Words including the most common letters
- Words including common prefixes/suffixes
- Combination of words that include every/most letters in the alphabet?

# About the Data

**Dataset:** <https://github.com/dwyl/english-words>

- Type: .txt file
- Size: 479k+ entries
  - Non-alpha: 370k+ words
  - 5-letter: ~16k words
- Contents: Comprehensive list of English words

# Methodologies

- **Libraries:** seaborn, matplotlib, numpy, wordcloud

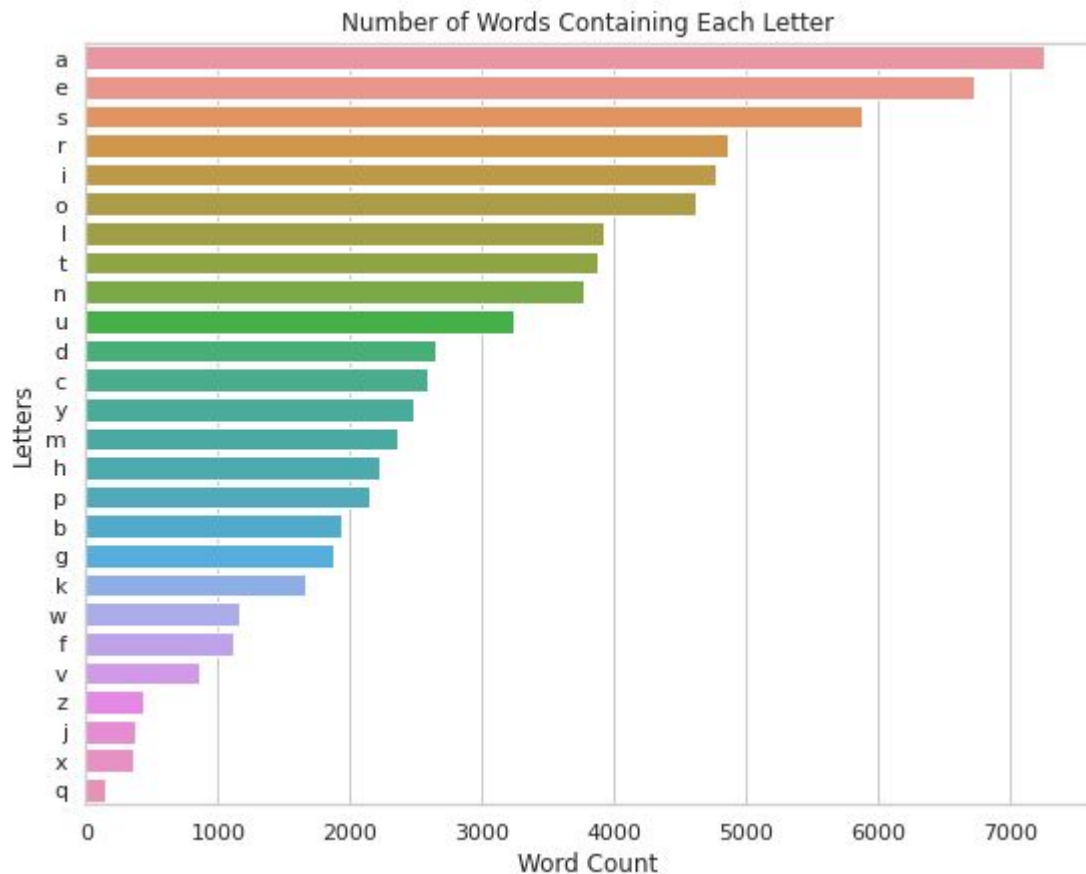
## Process

- **Imported** .txt file as a dataframe, adding a column name
- **Filtered** initial data to only examine 5-letter words
- **Created** a new dataframe to examine words by letter, with columns for words, word count, letter count, and duplicate count
- **Transformed** computed data types into integer type
- **Created** visualizations based on column values

# Results

Letters found in the most words

- A: 8392 words
- E: 7800 words
- S: 6537 words
- R: 5143 words
- I: 5067 words

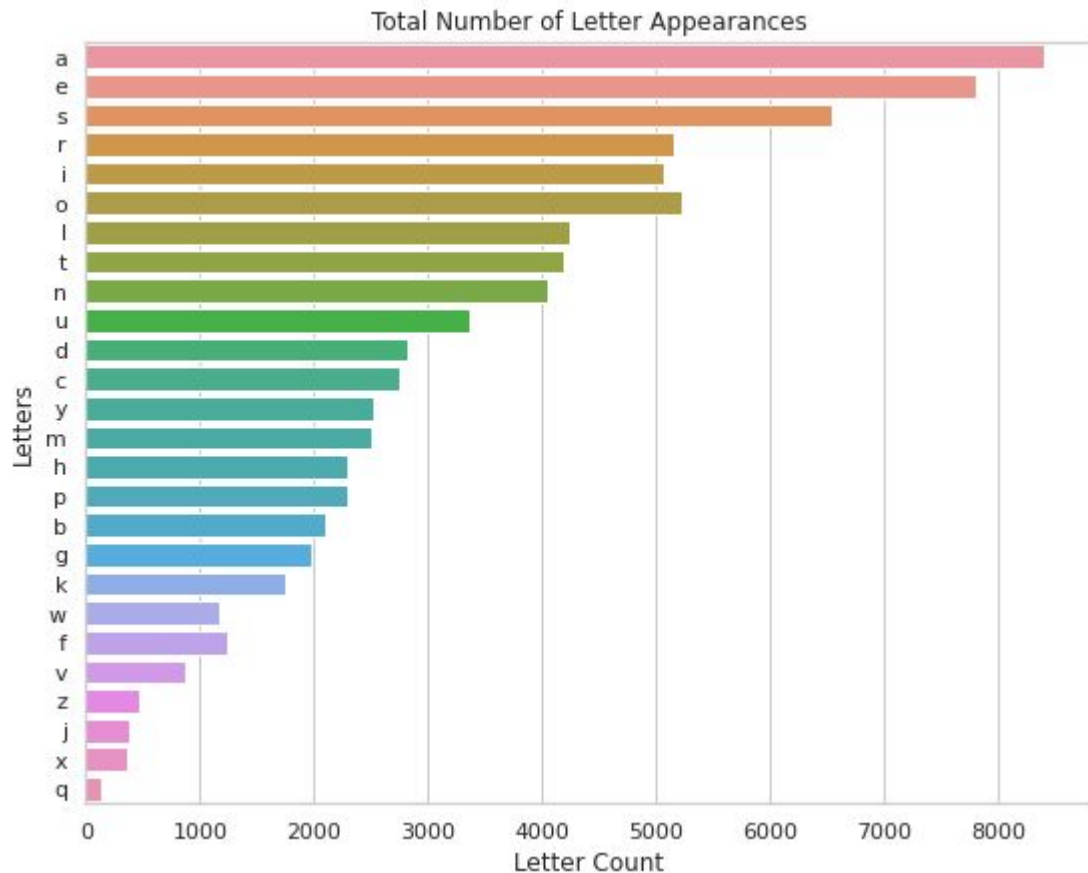




# Results

Letters that appeared the most

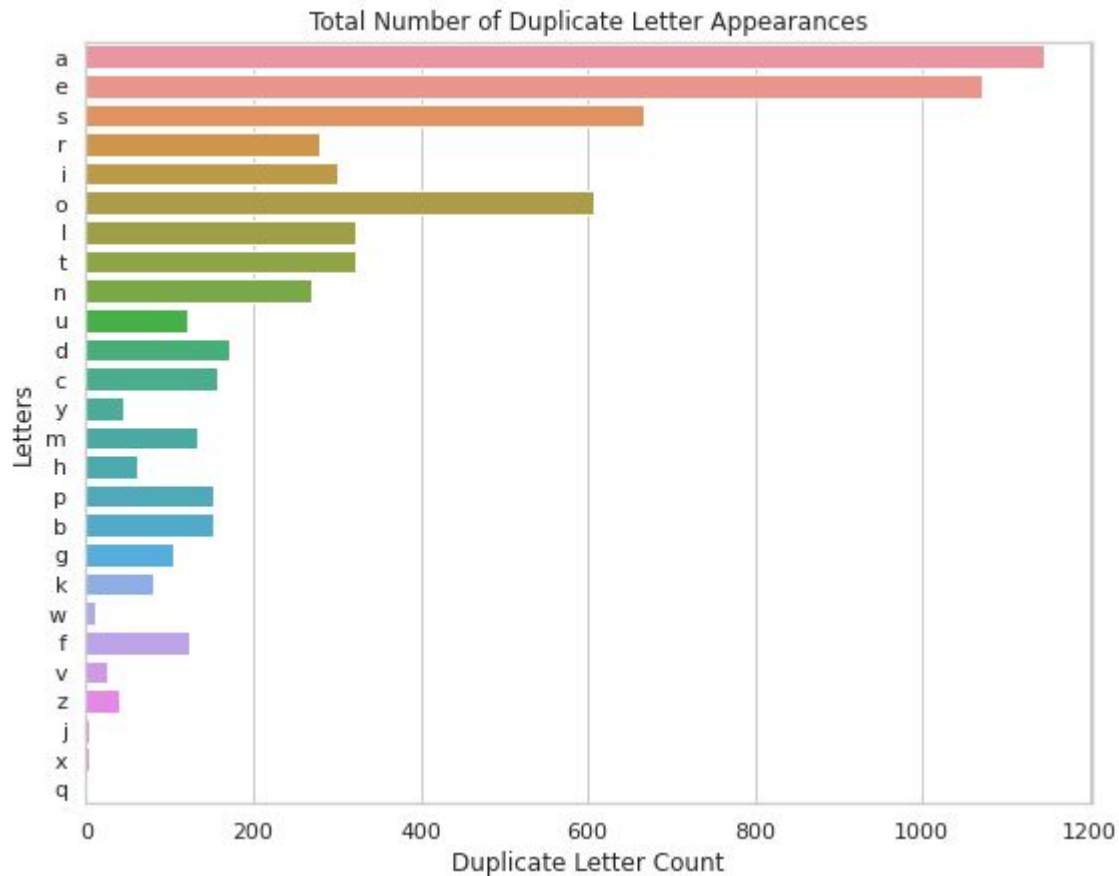
- A: 8392 times
- E: 7800 times
- S: 6537 times
- O: 5219 times
- R: 5143 times



# Results

## Most duplicated letters

- A: 1145 times
- E: 1072 times
- S: 666 times
- O: 606 times
- T: 313 times



# Discussion

- Incorporate common letters into my initial guesses
  - Common Letters: A, E, S, R, I, O, L, T, N, U
  - Potential First Guesses: raise, riots, tries
  - Potential Later Guesses: toner, tonal, count
- Keep an eye out for likely duplicate letters
  - A, E, S, O, and T were most likely to be duplicated
  - If I get stuck on a word, I can consider re-using a letter
- I'd like to examine the data further:
  - Find common prefixes & suffixes
  - Find common letter combinations
  - Find the number of words containing repeating letters (total and by letter)
  - I had difficulty creating a letter cloud based on letter frequency, so I'd like to problem solve that