Wordle Analysis

A brief dive into the possibility of finally getting that 1/6

Faaiz Nadeem

Contents

Background

Wordle is a web-based word game an individual can play for free at https://www.nytimes.com/games/wordle/index.html. Players have six tries to guess the five-letter word of the day. Once a guess is submitted, the tile turns green if the letter is in the correct spot, yellow if the letter is in the word but in the wrong spot, and gray means the letter is not in the word at all.

From 300,000 players (reported in the beginning of January) to now reaching over 2 million users on a global scale, it is clear that Wordle is here to stay.

The following report uses Data Analysis to determine the most probable word candidates based on the build in "words" package in R.

Data Cleaning

The "words" package consists of a list of english scrabble words as listed in the Scrabbles official tournatment and club word list. These words are collated from the word game dictionary. The dataset by default has a total of 175,393 observations with two variables which include:

-word
-word_length

The largest word length within the dataset came out to be 15. However, any words that are less than or greater than 5 in character size are completely useless to us. Therefore, a subset was created with the name of *w5* that contains all the words within the dataset that have the length of 5 characers. Below you will see the first six values from *w5*.

```
> #install.packages("words")
> library(words)
Warning: package 'words' was built under R version 4.0.5
> head(w5 <- subset(words, word_length == 5))
      word word_length
5349 aahed           5
5350 aalii           5
5351 aargh           5
```

```
5352 abaca           5
5353 abaci           5
5354 aback           5
```

Analysis and Findings

A big part about Wordle is clearly guessing which letters goes in what index of the array (1-5)This section of the report contains a commentary about the findings of our data based on questions that arose during the analysis process to help minimize the error in guessing the letters given our data.

Most Common Letters in beginning and end

Out of the five-letter words in *w5*, which letter is most commonly used at the beginning and the end of words, respectively?

```
> #Create a function to find the most repeated character in the beginning
>
> most_repeated_beginning <- function(x) {
+   tab <- table(substr(x, 1, 1))
+   names(tab)[tab==max(tab)]
+ }
>
> most_repeated_beginning(w5$word)
[1] "s"
>
> #Create a function to find the most repeated character in the end
>
> most_repeated_ending <- function(x) {
+   tab2 <- table(substr(x, 5, 5))
+   names(tab2)[tab2 == max(tab2)]
+ }
>
> most_repeated_ending(w5$word)
[1] "s"
```

Here we can see that most commonly used letter at the beginning and end of the words is ‘s’. Yet this only partly solves the issue of the first and last position, what about the other positions?

Well within the English language the majority of the words have vowels, which led to the next question:

Vowel Count?

Out of the five-letter words in *w5*, how many words consists of at least one vowel (a, e, i, o, u)?

```
> test <- w5
>
> aWords <- grep("a", test$word)
> eWords <- grep("e", test$word)
> iWords <- grep("i", test$word)
> oWords <- grep("o", test$word)
> uWords <- grep("u", test$word)
>
```

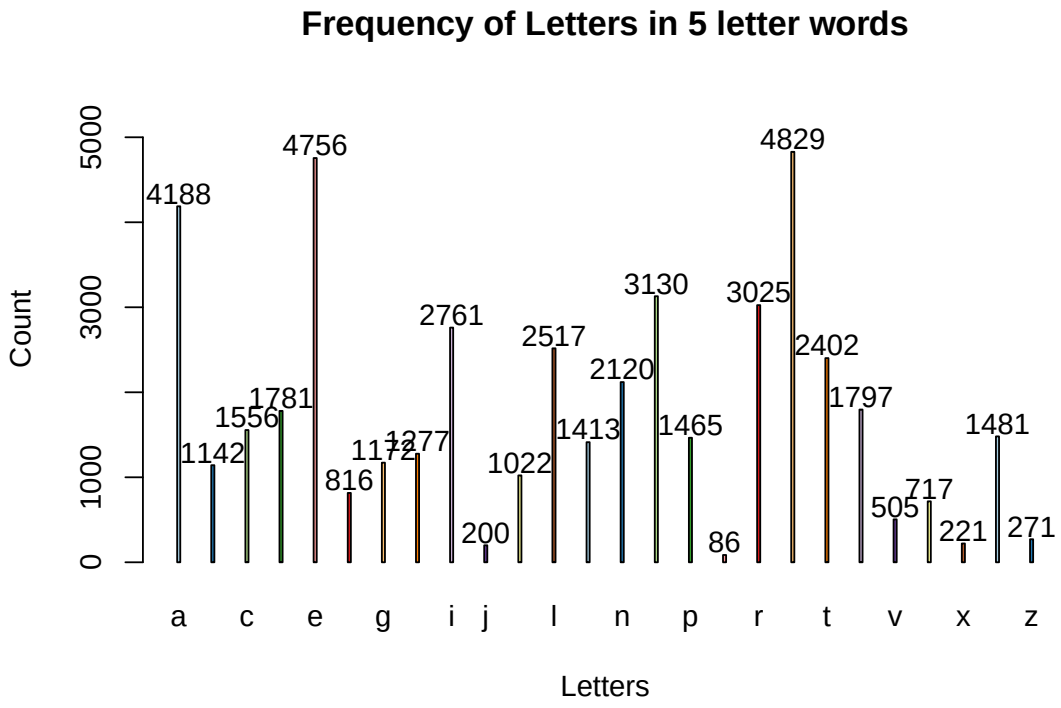
```
> collectedIndex <- c(aWords, eWords, iWords, oWords, uWords)
> results <- length(unique(collectedIndex))
> cat("There are",results,"words with vowels in them")
There are 9292 words with vowels in them
```

Out of all of the words within the *w5* data set (14,678), 63% of the words have vowels in them. Seeing this sparked curiosity towards what the most frequent letters were in the data set.

Letter Frequency

Below a bar plot was created to show the frequency of each letters in the list of five-letter words in *w5*.Considering the double letter rules that states the following "If you manage to guess a word that has a double letter and the answer does have both letters in it, then it will show you the yellow and green hints as per the location of the letter".
Double counts were not filtered out.

```
> splitData <- table(unlist(strsplit(w5$word, "")))
>
> library(RColorBrewer)
Warning: package 'RColorBrewer' was built under R version 4.0.3
> coul <- brewer.pal(12, "Paired")
> ylim <- c(0,5500)
>
> bp <- barplot(splitData, main = "Frequency of Letters in 5 letter words",
+               xlab = "Letters", ylab = "Count", col = coul,
+               space = 10, ylim = ylim)
>
> y<-as.matrix(splitData)
> text(bp, y+150, labels = as.character(y))
```



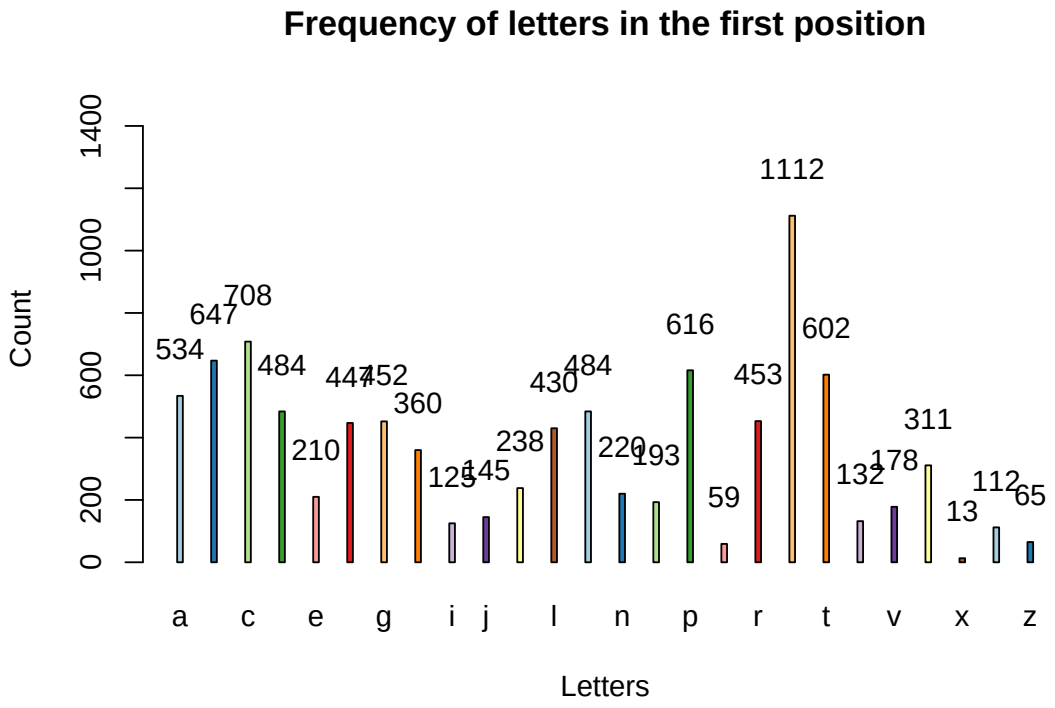
The five letters with the highest frequency include s (4829), e (4756), a(4188), o (3130), and r (3025). One word that a person could open with this would be a r o s e or arose.

Index Frequency

The bar plot above shows frequency of letters without any consideration towards location. Below are plots that show the frequency of letters from position 1-5. The following algorithm was implemented to filter out words that would not be a suitable list of candidates.

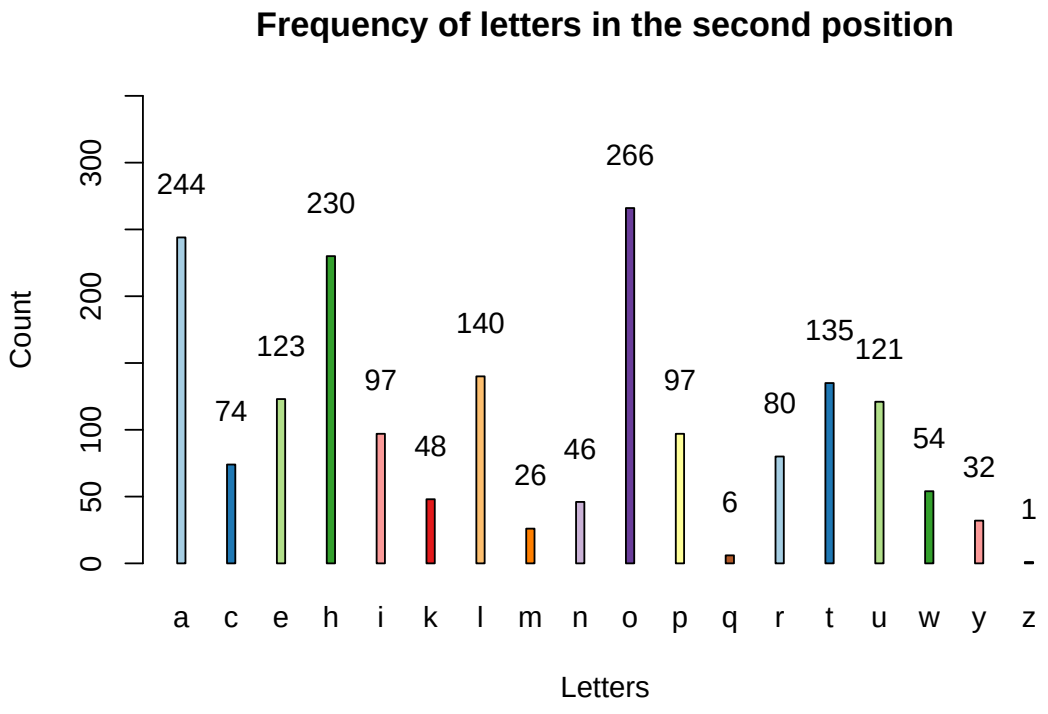
```
i. Identify the most common **2** letters used in the first position.
ii. Filter out words from ‘w5‘ that do not start with the letter identified in Step i.
iii. Repeat Steps i and ii for the 2nd, 3rd, 4th, and 5th position.

The remaining of Step 3 is a list of word candidates.
```

```
> #Data for first letter
> tab <- table(substr(w5$word, 1, 1))
> names(tab)[tab==max(tab)]
[1] "s"
>
> library(RColorBrewer)
> coul <- brewer.pal(12, "Paired")
> ylim <- c(0,1500)
>
> bp_first_letter <- barplot(tab, main = "Frequency of letters in the first position",
+                xlab = "Letters", ylab = "Count", col = coul,
+                space = 5, ylim = ylim)
>
> y<-as.matrix(tab)
> text(bp_first_letter, y+150, labels = as.character(y))
```



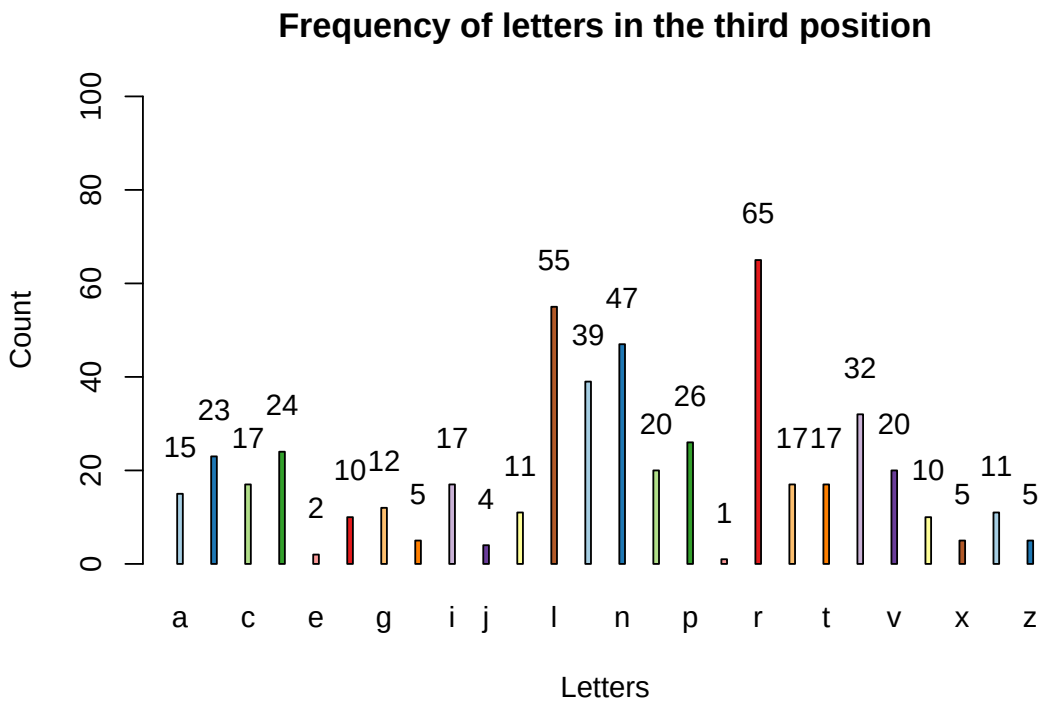
Most common two letters are ‘s’ and ‘c’; Remove all other words that do not start with ‘s’ and ‘c’

```
> library(dplyr)
Warning: package 'dplyr' was built under R version 4.0.5

Attaching package: 'dplyr'
The following objects are masked from 'package:stats':

    filter, lag
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
> w5 <- filter(w5, substr(w5$word, 1,1) %in% c("s", "c"))
>
> #Data for second letter
> tab2 <- table(substr(w5$word, 2, 2))
> names(tab2)[tab2==max(tab2)]
[1] "o"
> ylim <- c(0, 350)
>
> bp_second_letter <- barplot(tab2, main = "Frequency of letters in the second position",
+                xlab = "Letters", ylab = "Count", col = coul,
+                space = 5, ylim = ylim)
>
> y<-as.matrix(tab2)
> text(bp_second_letter, y+40, labels = as.character(y))
```



Most common two letters are ‘o’ and ‘a’; Remove all other words that do not have ‘a’ or ‘o’ in the second element

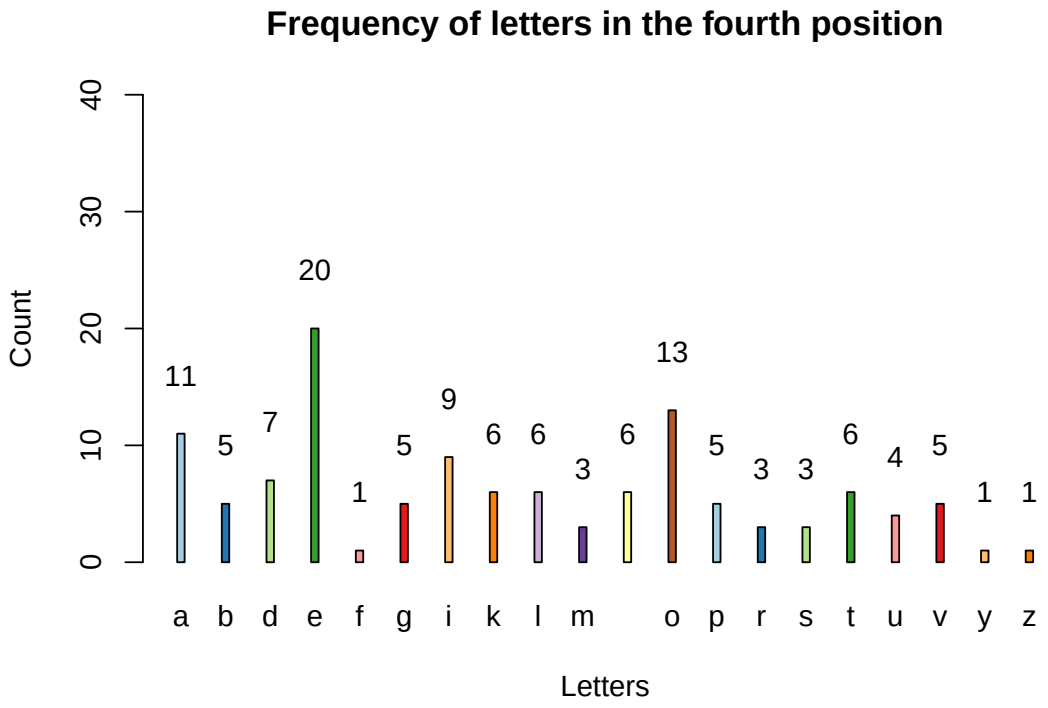
```
> w5 <- filter(w5, substr(w5$word, 2,2) %in% c("o", "a"))
>
> #Data for third letter
> tab3 <- table(substr(w5$word, 3, 3))
> names(tab3)[tab3==max(tab3)]
[1] "r"
> ylim <- c(0, 100)
>
> bp_third_letter <- barplot(tab3, main = "Frequency of letters in the third position",
+               xlab = "Letters", ylab = "Count", col = coul,
+               space = 5, ylim = ylim)
>
> y<-as.matrix(tab3)
> text(bp_third_letter, y+10, labels = as.character(y))
```



Most common two letters are 'r' and 'l'; Remove all other words that do not have 'r' or 'l' in the third element

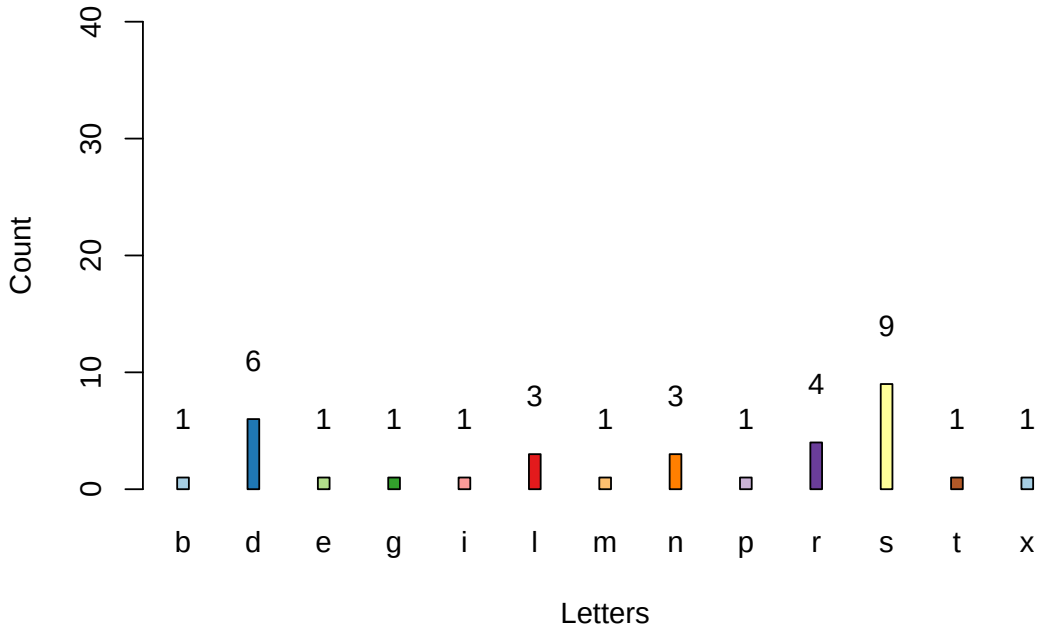
```
> w5 <- filter(w5, substr(w5$word, 3,3) %in% c("r", "l"))
>
> #Data for fourth letter
> tab4 <- table(substr(w5$word, 4, 4))
> names(tab4)[tab4==max(tab4)]
[1] "e"
>
> ylim <- c(0, 40)
>
> bp_fourth_letter <- barplot(tab4, main = "Frequency of letters in the fourth position",
+               xlab = "Letters", ylab = "Count", col = coul,
+               space = 5, ylim = ylim)
```

```
>
> y<-as.matrix(tab4)
> text(bp_fourth_letter, y+5, labels = as.character(y))
```



Most common two letters are 'e' and 'o'; Remove all other words that do not have 'e' or 'o' in the fourth element

```
> w5 <- filter(w5, substr(w5$word, 4,4) %in% c("e", "o"))
>
>
> #Data for fifth letter
> tab5 <- table(substr(w5$word, 5, 5))
> names(tab5)[tab5==max(tab5)]
[1] "s"
>
> ylim <- c(0, 40)
>
> bp_fifth_letter <- barplot(tab5, main = "Frequency of letters in the fifth position",
+               xlab = "Letters", ylab = "Count", col = coul,
+               space = 5, ylim = ylim)
>
> y<-as.matrix(tab5)
> text(bp_fifth_letter, y+5, labels = as.character(y))
```



Most common two letters are ‘s’ and ‘d’; Remove all other words that do not have ‘s’ or ‘d’ in the fourth element

After filtering out the data up to the last position, the remaining words presented are the most probable in terms of letter frequency by position.

```
> w5 <- filter(w5, substr(w5$word, 5,5) %in% c("s", "d"))
```

Conclusion

After cleaning and analyzing the data, the result is the following list of word candidates:

```
> w5
    word word_length
1  calos           5
2  cared           5
3  cares           5
4  coled           5
5  coles           5
6  cored           5
7  cores           5
8  sales           5
9  sarod           5
10 saros           5
11 soled           5
12 soles           5
13 solos           5
14 sored           5
15 sores           5
```

Working on this mini project was quite fun and eye opening. Of course this data is not the word set that Wordle uses, however, the opportunity to dig through and determine the list of words that would be most probable using data analysis was quite insightful and a fun learning experience. In the future I plan to implement the data set that Wordle actually uses to determine the "correct" word candidates. Who knows, maybe I'll finally get a 1/6 that way.