NYT Spelling Bee Solver

C-T Berezin

11/9/21

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#all english words
word_list <- read_csv(paste0("https://raw.githubusercontent.com/dwyl",
                             "/english-words/master/words_alpha.txt"),
                      col_names = "word")
```

```
## Rows: 370103 Columns: 1
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): word
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
word_list <- word_list %>%
  mutate(length = str_length(word)) %>%
  filter(length >= 4)

#words to remove, weren't accepted in puzzle
nonwords <- c("fluoroformol", "fluoroform", "floramour", "floramor", "formular",
  "furfural", "furfurol", "affloof", "famular", "farfara", "florula",
  "formful", "mafurra", "alfuro", "fallal", "farfal", "faroff",
  "forfar", "formol", "fulful", "fullam", "fullom", "fulmar", "furfur",
  "mafura", "afara", "afoam", "alfur", "falla", "flaff", "fluor", "flurr",
  "foram", "forma", "forra", "froom", "fural", "luffa", "mafoo",  "affa",
  "alfa", "faff", "fala", "fama", "farl", "faro", "flam", "flor", "foll",
  "fram", "frau", "fuff", "fula", "loof", "luff", "moff", "olaf", "olof",
  "raff", "ralf", "rolf")

word_list <- word_list %>% filter(!word %in% nonwords)
```

```
center_letter <- "f"
circle_letters <- c("m",
                    "u",
                    "a",
                    "r",
                    "l",
                    "o")
all_letters <- c(center_letter, circle_letters)
```

```
solving <- word_list %>%
  #filter to words with center letter
  filter(str_detect(word, center_letter) == TRUE) %>%
  #break work into letters
  mutate(lets = str_split(word, pattern=""))

solving <- solving %>%
  #check the word's letters against the puzzle's letters
  mutate(match = map(lets, ~.x %in% all_letters, lets=.x)) %>%
 #count number of matching letters
  mutate(matches = map(match, sum)) %>%
  unnest(matches)

answers <- solving %>%
  #keep words that only use the puzzle's letters
  filter(matches == length) %>%
  arrange(desc(length))

answers$word
```

```
##  [1] "froufrou" "alfalfa"  "falloff"  "formula"  "roomful"  "armful"
##  [7] "floral"   "formal"   "afoul"    "aloof"    "floor"    "flora"
## [13] "flour"    "fluff"    "forum"    "furor"    "loofa"    "offal"
## [19] "afar"     "afro"     "fall"     "farm"     "foal"     "foam"
## [25] "fool"     "fora"     "form"     "foul"     "four"     "from"
## [31] "full"     "furl"     "loaf"     "muff"     "roof"     "ruff"
```

```
pangrams <- answers %>%
  filter(length >= 7) %>%
  #check the puzzle's letters against the word's letters
  mutate(lets = map(lets, unique)) %>%
  mutate(pangram = map(lets, ~.x %in% all_letters, lets=.x)) %>%
 #count number of matching letters
  mutate(p_matches = map(pangram, sum)) %>%
  unnest(p_matches)

pangrams %>%
  #keep words that only use the puzzle's letters
  filter(p_matches == 7) %>%
  arrange(desc(length)) %>%
  pull(word)
```

```
## [1] "formula"
```