wordle.rmd

Step 1. Create your own custom function

```
load_dictionary <- function(filename) {
  words <- readLines(filename)
  return(words)
}
f <- "https://raw.githubusercontent.com/difiore/ada-2024-datasets/main/collins-scrabble-words-2019.txt"
f
```

```
## [1] "https://raw.githubusercontent.com/difiore/ada-2024-datasets/main/collins-scrabble-words-2019.tx
```

```
valid_list <- load_dictionary(f)
```

```
## Warning in readLines(filename): incomplete final line found on
## 'https://raw.githubusercontent.com/difiore/ada-2024-datasets/main/collins-scrabble-words-2019.txt'
```

```
solution_list <- load_dictionary(f)
```

```
## Warning in readLines(filename): incomplete final line found on
## 'https://raw.githubusercontent.com/difiore/ada-2024-datasets/main/collins-scrabble-words-2019.txt'
```

```
str(valid_list)
```

```
##  chr [1:279497] "words" "AA" "AAH" "AAHED" "AAHING" "AAHS" "AAL" "AALII" ...
```

```
str(solution_list)
```

```
##  chr [1:279497] "words" "AA" "AAH" "AAHED" "AAHING" "AAHS" "AAL" "AALII" ...
```

Step 2.Winnow your variable solution_list to only include words that are included in valid_list

```
load_dictionary <- function(filename) {
  words <- readLines(filename)
  return(words)
}
valid_list <- load_dictionary(f)
```

```
## Warning in readLines(filename): incomplete final line found on
## 'https://raw.githubusercontent.com/difiore/ada-2024-datasets/main/collins-scrabble-words-2019.txt'
```

```r
solution_list <- load_dictionary(f)
```

```
## Warning in readLines(filename): incomplete final line found on
## 'https://raw.githubusercontent.com/difiore/ada-2024-datasets/main/collins-scrabble-words-2019.txt'
```

```r
updated_solution_list <- intersect(solution_list, valid_list)
num_words_updated_solution_list <- length(updated_solution_list)
print(num_words_updated_solution_list)
```

```
## [1] 279497
```

There are 279497 words in updated solution_list vector.

Step 3:Write a custom function called `pick_solution()`

```r
#1.Write pick_solution function
pick_solution <- function(solution_list, word_length = 5) {
  #Remove words not of the specified length
  valid_length_words <- solution_list[nchar(solution_list) == word_length]
  #Check if there are words of the specified length
  if (length(valid_length_words) == 0) {
    stop("No words of the specified length found in solution_list.")
  }
  #2.Randomly choose a word from filtered list
  chosen_word <- sample(valid_length_words, 1)
  #3.Split the chosen word into a vector of single-character elements
  split_word <- strsplit(chosen_word, NULL)[[1]]
  return(split_word)
}
# 4.Use
solution <- pick_solution(solution_list)
print(solution)
```

```
## [1] "H" "O" "O" "K" "S"
```

```r
solution_custom_length <- pick_solution(solution_list, word_length = 6)
print(paste("Custom length (word_length = 6):", paste(solution_custom_length, collapse = "")))
```

```
## [1] "Custom length (word_length = 6): FLOUSE"
```

Step 4: create two more functions

```r
# 1.Helper function to evaluate a guess
evaluate_guess <- function(guess, solution) {
  feedback <- character(length(solution))
  for (i in seq_along(solution)) {
    if (guess[i] == solution[i]) {
      feedback[i] <- "*"
    } else if (guess[i] %in% solution) {
      feedback[i] <- "+"
    } else {
```

```
      feedback[i] <- "-"
    }
  }
  return(feedback)
}

# 2.Main function to play Wordle
play_wordle <- function(solution, valid_list, num_guesses = 6) {
  alphabet <- LETTERS
  remaining_letters <- tolower(alphabet)
  feedback_history <- list()

  cat("Welcome to Wordle!\n")
  cat("You have", num_guesses, "chances to guess a word of length", length(solution), "\n")

  # 3.Main game loop
  for (guess_num in 1:num_guesses) {
    # Display remaining letters
    cat("Letters left:", paste(remaining_letters, collapse = " "), "\n")

    # 4.Prompt the player for a guess
    guess <- tolower(readline(paste("Enter guess number", guess_num, ": ")))

    # 5.Check if the guess is valid
    if (!(guess %in% valid_list && nchar(guess) == length(solution))) {
      cat("Invalid guess. Please enter a valid word of the correct length.\n")
      guess_num <- guess_num - 1
      next
    }

    # 6.Evaluate the guess
    feedback <- evaluate_guess(guess, solution)

    # 7.Update remaining letters
    remaining_letters <- setdiff(remaining_letters, strsplit(guess, NULL)[[1]])

    # 8.Store guess and feedback in the history
    feedback_history[[guess_num]] <- list(guess = guess, feedback = feedback)

    # 9.Check if the puzzle was solved
    if (identical(feedback, rep("*", length(solution)))) {
      cat("Congratulations! You WON!\n")
      cat("Solution:", paste(solution, collapse = ""), "\n")
      cat("Guess and Feedback History:\n")
      for (i in seq_along(feedback_history)) {
        print(paste("Checking index", i, "of feedback_history"))
        if (i <= length(feedback_history) && !is.null(feedback_history[[i]])) {
          cat("Guess:", feedback_history[[i]]$guess, "Feedback:", paste(feedback_history[[i]]$feedback,
        } else {
          cat("Index", i, "out of bounds for feedback_history\n")
        }
      }
      return(invisible())
```

```
    }
  }

  #10.If all guesses are exhausted
  cat("Sorry, you LOST the game.\n")
  cat("Solution:", paste(solution, collapse = ""), "\n")
  cat("Guess and Feedback History:\n")
  for (i in seq_along(feedback_history)) {
    print(paste("Checking index", i, "of feedback_history"))
    if (i <= length(feedback_history) && !is.null(feedback_history[[i]])) {
      cat("Guess:", feedback_history[[i]]$guess, "Feedback:", paste(feedback_history[[i]]$feedback, col
    } else {
      cat("Index", i, "out of bounds for feedback_history\n")
    }
  }
}

#11.Use
solution <- pick_solution(solution_list)
play_wordle(solution, valid_list)
```

```
## Welcome to Wordle!
## You have 6 chances to guess a word of length 5
## Letters left: a b c d e f g h i j k l m n o p q r s t u v w x y z
## Enter guess number 1 :
## Invalid guess. Please enter a valid word of the correct length.
## Letters left: a b c d e f g h i j k l m n o p q r s t u v w x y z
## Enter guess number 2 :
## Invalid guess. Please enter a valid word of the correct length.
## Letters left: a b c d e f g h i j k l m n o p q r s t u v w x y z
## Enter guess number 3 :
## Invalid guess. Please enter a valid word of the correct length.
## Letters left: a b c d e f g h i j k l m n o p q r s t u v w x y z
## Enter guess number 4 :
## Invalid guess. Please enter a valid word of the correct length.
## Letters left: a b c d e f g h i j k l m n o p q r s t u v w x y z
## Enter guess number 5 :
## Invalid guess. Please enter a valid word of the correct length.
## Letters left: a b c d e f g h i j k l m n o p q r s t u v w x y z
## Enter guess number 6 :
## Invalid guess. Please enter a valid word of the correct length.
## Sorry, you LOST the game.
## Solution: SHIES
## Guess and Feedback History:
```

```
knitr::opts_chunk$set(echo = FALSE)
```