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.txt"

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'

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'

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()**

pick\_solution <- function(solution\_list, word\_length = 5) {  
 valid\_length\_words <- solution\_list[nchar(solution\_list) == word\_length]  
 if (length(valid\_length\_words) == 0) {  
 stop("No words of the specified length found in solution\_list.")  
 }  
 chosen\_word <- sample(valid\_length\_words, 1)  
 split\_word <- strsplit(chosen\_word, NULL)[[1]]  
 return(split\_word)  
}  
solution <- pick\_solution(solution\_list)  
print(solution)

## [1] "T" "R" "A" "Y" "S"

solution <- pick\_solution(solution\_list)  
print(solution)

## [1] "L" "Y" "T" "E" "S"

**Step 4: create two more functions**

# 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)  
}  
  
# 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")  
  
 # Main game loop  
 for (guess\_num in 1:num\_guesses) {  
 # Display remaining letters  
 cat("Letters left:", paste(remaining\_letters, collapse = " "), "\n")  
  
 # Prompt the player for a guess  
 guess <- tolower(readline(paste("Enter guess number", guess\_num, ": ")))  
  
 # 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  
 }  
  
 # Evaluate the guess  
 feedback <- evaluate\_guess(guess, solution)  
  
 # Update remaining letters  
 remaining\_letters <- setdiff(remaining\_letters, strsplit(guess, NULL)[[1]])  
  
 # Store guess and feedback in the history  
 feedback\_history[[guess\_num]] <- list(guess = guess, feedback = feedback)  
  
 # 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, collapse = ""), "\n")  
 } else {  
 cat("Index", i, "out of bounds for feedback\_history\n")  
 }  
 }  
 return(invisible())  
 }  
 }  
  
 # 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, collapse = ""), "\n")  
 } else {  
 cat("Index", i, "out of bounds for feedback\_history\n")  
 }  
 }  
}  
  
# Example usage  
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: GLEET   
## Guess and Feedback History:

knitr::opts\_chunk$set(echo = FALSE)