Solution Review: Handling TXT files

In this review, we provide a detailed analysis of the solution to this problem.

We'll cover the following

Solution: Input/Output from File

main.r

data.txt

All code files are copied to end of the page...

Explanation

This exercise is similar to exercise 13, however, now we have to take input from a file and also output on file.

The main execution of the code given above starts from **line number 20**. We first open the file located at the given `path`. We then read all the lines present in the file and store in `myVector`. Next, we pass this vector to the function `evenOdd()`. This function returns us the `myOutputVector` vector, which we write on file `output/outData.txt`.

In the next lesson, we will learn how to handle `.csv` files.

Code Files Content !!!

⇓ ⇓

```
---------------------------------------------------------------------------
|  main.r [1]
---------------------------------------------------------------------------


evenOdd <- function(myVector) # a function that returns a vector
                      # whose each element tells even or odd corresponding to input vector
{
  myOutputVector <- vector("character", 0)
  for(i in myVector)
  {
      if(as.integer(i) %% 2 == 0)
      {
        myOutputVector <- c(myOutputVector, "even")
      }
      else
      {
        myOutputVector <- c(myOutputVector, "odd")
      }
  }
  return(myOutputVector)
}

# Driver Code
path <- "data.txt"

fileData <- file(path, open = "r") # open the file located in path
lines <- readLines(fileData) # read lines of the file

myVector <- vector("numeric", 0) # vector to store data

for (i in 1:length(lines)) # iterate over all the lines
{
  myVector <- c(myVector, lines[i]) # append lines in myVector
}

result <- evenOdd(myVector) # store output in result
write(result, "output/outData.txt") # write result on file


---------------------------------------------------------------------------
|  data.txt [1]
---------------------------------------------------------------------------


1
2
3





*****************************************************************************
```