**#!/usr/bin/env Rscript**

**args = commandArgs(trailingOnly=TRUE)**

*#!/usr/bin/env Rscript:* This is called a shebang line. It is used at the beginning of a script to specify the interpreter that should be used to execute the script. In this case, it indicates that the script should be run using the Rscript interpreter.

*args = commandArgs (trailingOnly=TRUE)*: This line assigns the result of the *commandArgs (trailingOnly=TRUE)* function to the variable *args*. The commandArgs function is used to retrieve command-line arguments passed to the script. By setting *trailingOnly=TRUE*, it captures only the arguments that come after the script name.

We run the script with a command like Rscript yourscript.R arg1 arg2, the args variable will be a character vector containing "arg1" and "arg2".

Ex***: Rscript --vanilla sillyScript.R iris.txt out.txt***

* Rscript: This is the command to run R scripts from the command line.
* --vanilla: The --vanilla option is used to start R in a "vanilla" mode, which means without reading any startup files or environment variables. This provides a clean and standardized environment for script execution.
* sillyScript.R: This is the name of the R script you are running.
* iris.txt: This is an example of a command-line argument. It means you are passing the file named iris.txt as an argument to your R script. Inside your script, you can access this argument using the args variable.
* out.txt: Similar to iris.txt, this is another command-line argument representing the output file name.

***Rscript --vanilla sillyScript.R iris.txt out.txt***

#!/usr/bin/env Rscript

args <- commandArgs(trailingOnly = TRUE)

# Check if there are at least two command-line arguments (input and output files)

if (length(args) < 2) {

stop("Usage: Rscript sillyScript.R input\_file output\_file")

}

# Accessing input and output file names

input\_file <- args[1]

output\_file <- args[2]