

How can I write multiple lines to a text file using the -R command?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I write multiple lines to a text file using the -R command?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=150050>

The -R command is a useful tool that allows users to write multiple lines to a text file in a single command. This command can be used in various programming languages and operating systems to efficiently add and edit content in a text file. By using the -R command, users can easily write and append multiple lines of text without the need for manually entering each line. This helps save time and effort, and also minimizes the chances of errors. Overall, the -R command provides a convenient and efficient way to input and manage text content in a file.

How do I write lines of strings to a text file in R with an extension .txt file? R provides several ways to write multiple lines in sequence to a text file by using writeLines(), sink(), cat() and write_lines() from tidyverse package.

1. Quick Examples

The following are quick examples of how to write multiple lines to a text file in R.

```
# Quick Examples
```

```
# Example 1 - using writeLines()
```

```
file_conn = file("/Users/admin/textFile.txt")
```

```
writeLines(c("I Love R Programming", "I live in USA"), file_conn)
```

```
close(fileConn)
```

```
file.show("/Users/admin/textFile.txt")
```

```
# Example 2 - Using with out file connection
```

```
writeLines(c("I Love R Programming", "I live in USA"),
```

```
"/Users/admin/textFile.txt")
```

```
# Example 3 - Using string
```

```
text ='I Love R Programming\nI live in USA'
```

```
writeLines(text, "/Users/admin/textFile.txt")
```

```
# Example 4 - using sink()
```

```
sink("/Users/admin/textFile.txt")
```

```
cat("I Love R Programming")
```

```
cat("\n")
```

```
cat("I live in USA")
```

```
sink()
```

```
# Example 5 - Using cat()
```

```
cat("I Love R Programming", file="outfile.txt", sep="\n")
```

```
cat("I live in USA", file="/Users/admin/textFile.txt", append=TRUE)
```

```
# Example 6 - Using tidyverse
library(tidyverse)
c('I Love R Programming', 'I live in USA') %>%
write_lines( "/Users/admin/textFile.txt")
```

2. Using writeLines()

R base function `writeLines()` is used to write the sequence of multiple lines to the text file. This method accepts Vector with the lines you would like to write or string. To create a vector use `c()` function. The option of `writeLines()` is roughly ten times faster than the `sink()` and `cat()` methods explained below.

2.1 Syntax of writeLines()

Following is the syntax of the `writeLines()` function.

```
# Syntax of writeLines()
writeLines(text, con = stdout(), sep = "\n", useBytes = FALSE)
```

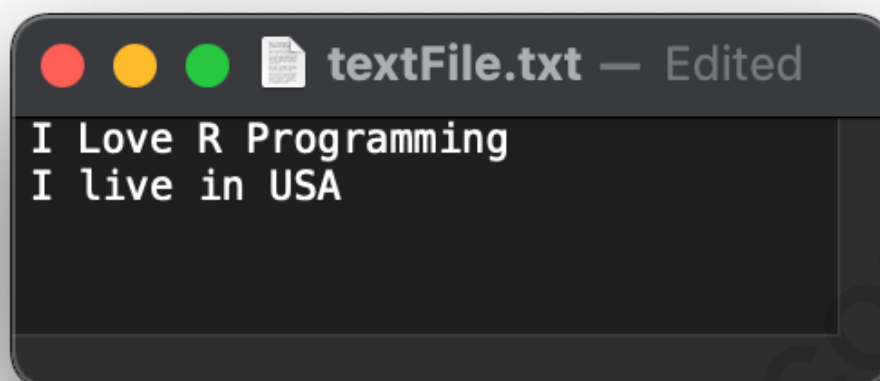
<code>text</code>	A character vector or character string
<code>con</code>	A File connection object or a file name in a string.
<code>sep</code>	character string. A string is to be written to the connection after each line of text.
<code>useBytes</code>	Default is set to False.

2.2 Write Lines to Text File Example

Here, I have using a lines I wanted to write to a text file in a Vector. If the file you are trying to write not exists, it automatically creates it and write the contents.

```
# Example 1 - using writeLines()
file_conn = file( "/Users/admin/textFile.txt")
writeLines(c("I Love R Programming", "I live in USA"), file_conn)
close(fileConn)
```

This creates a `textFile.txt` at the specified location and the contents of the file would be.



Now, let's simplify this without creating a file connection. If the `con` is a character string, it automatically gets a file connection which is opened for the duration of the function call and closes it once the write completes.

```
# Example 2
writeLines(c("I Love R Programming", "I live in USA"),
           "/Users/admin/textFile.txt")
```

If you have a string with multiple lines separated by new line character, you can use this to write.

```
# Example 3
text = 'I Love R Programming\nI live in USA'
writeLines(text, "/Users/admin/textFile.txt")
```

3. Using `sink()` to Write Lines

You can also use `sink()` to write lines to a `.txt` file.

```
# Example 4
```

```
sink("/Users/admin/textFile.txt")
cat("I Love R Programming")
cat("\n")
cat("I live in USA")
sink()
```

4. Using cat() to Write Lines to Text File

Another approach would be using `cat()`, this is not a recommended option as it opens and closes a file for every `cat()` function you call. This approach might be inefficient. Use `append=TRUE` to append lines to the existing text file.

```
# Example 5 - Using cat
cat("I Love R Programming",file="/Users/admin/textFile.txt",sep="\n")
cat("I live in USA",file="/Users/admin/textFile.txt",append=TRUE)
```

5. Using tidyverse package

Package `tidyverse` provides `write_lines()` function that can be used to write lines to a text file. In order to use `tidyverse` first you need to install it using `install.packages('tidyverse')` and load it using `library(tidyverse)`

```
# Example 6 - Using tidyverse
library(tidyverse)
c('I Love R Programming', 'I live in USA') %>%
write_lines("/Users/admin/textFile.txt")
```

6. Conclusion

In this article, you have learned how to write multiple string lines to a text file in R by using `writeLines()`, `sink()`, `cat()`, and method from `tidyverse` package.

Related Articles

References