

# How can I read a text file and convert it into a DataFrame?

Authored by  
**stats writer**

June 24, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I read a text file and convert it into a DataFrame?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=149996>

To read a text file and convert it into a DataFrame, you can use a programming language such as Python or R, which have libraries and functions specifically designed for data analysis. Firstly, import the necessary libraries such as Pandas for Python or `data.frame` for R. Then, use the appropriate function, such as `read_csv()` for a comma-separated value (CSV) text file or `read_table()` for a tab-delimited text file, to read the file and store it in a DataFrame object. This will allow you to manipulate and analyze the data in a tabular format, making it easier to perform various operations such as sorting, filtering, and statistical analysis. By converting a text file into a DataFrame, you can efficiently handle and extract useful information from large datasets.

R base package provides several functions to load or read a single text file (TXT) and multiple text files into R DataFrame. Text file with extension `.txt` is a human-readable format that is sometimes used to store scientific and analytical data. When storing data in text files the fields are usually separated by a tab delimiter.

In my previous article, I explained how to import a CSV file into Data Frame and import an Excel file into Data Frame. In this article, I will explain how to read a text file by using `read.table()` into Data Frame with examples? To export to Text File use `wirte.table()`

## 1. Quick Examples of Read Text File

Following are quick examples of how to read a text file to DataFrame in R.

```
# Quick Examples

# Read text file
df = read.table('/Users/admin/file.txt',sep='t')

# Read multiple text files
list_files = list('/Users/admin/file.txt', '/Users/admin/file2.txt')
df = read.table(list_files,sep='t')

# Read text file with header
df = read.table('/Users/admin/file.txt',sep='t', header=TRUE)

# Read text file with custom columns
col_names= c('id_col','name_col','dob_col','gender_col')
df = read.table('/Users/admin/file.txt',sep='t', header=TRUE,col.names = col_names)

# Skip first 2 rows
df = read.table('/Users/admin/file.txt',sep='t', skip = 2)
```

## 2. Read TEXT File in R using read.table()

`read.table()` is a function from the R base package which is used to read text files where fields are separated by any delimiter. If you have a comma-separated CSV file use `read.csv()` function.

### 2.1 Syntax of read.table()

Following is the syntax of the `read.table()` function.

```
# Syntax of read.table()
read.table(file, header = FALSE, sep = "", quote = "'",
dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
row.names, col.names, as.is = !stringsAsFactors,
na.strings = "NA", colClasses = NA, nrows = -1,
skip = 0, check.names = TRUE, fill = !blank.lines.skip,
strip.white = FALSE, blank.lines.skip = TRUE,
comment.char = "#",
allowEscapes = FALSE, flush = FALSE,
stringsAsFactors = default.stringsAsFactors(),
fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

### 2.2 Read TEXT File Example

I have a text file with a tab delimiter and I will use `sep='t'` argument with `read.table()` function to read it into DataFrame.

```
# Read text file
df = read.table('/Users/admin/file.txt', sep='t')
print(df)
```

### 2.3 Read Multiple Text Files

In order to read multiple text files in R, create a list with the file names and pass it as an argument to this function.

```
# Read multiple text files
list_files = list('/Users/admin/file.txt', '/Users/admin/file2.txt')
df = read.table(list_files, sep='t')
```

```
print(df)
```

## 2.4 Read File with Header

If you have a text file with a header then you have to use `header=TRUE` argument, Not specifying this will consider the header row as a data record.

```
# Read text file with header
df = read.table('/Users/admin/file.txt', sep='t', header=TRUE)
print(df)
```

## 2.5 Assign new Columns

When you don't want the column names from the file header and wanted to use your own column names use `col.names` argument which accepts a Vector, use `c()` to create a Vector with the column names you desire.

```
# Read text file with custom column names
col_names= c('id_col', 'name_col', 'dob_col', 'gender_col')
df = read.table('/Users/admin/file.txt', sep='t', header=TRUE, col.names =
col_names)
print(df)
```

Alternatively, you can also rename columns in DataFrame right after creating the data frame.

## 2.6 Skip Rows of a TXT file

Sometimes you may need to skip a few rows while reading the text file to R DataFrame. You can do this by using the `skip` argument.

```
# Skip first 2 rows
df = read.table('/Users/admin/file.txt', sep='t', skip = 2)
df
```

## 3. Read TEXT File using `read.delim()`

You can also use `read.delim()` to read a text file into DataFrame.

```
# using read.delim()
df = read.delim('/Users/admin/file.txt',header = TRUE, sep = "t")
df
```

## 4. Use read\_tsv() to Read Tab Delimiter Text File

If you are working with larger files, you should use the `read_tsv()` function from `readr` package. Since `readr` is a third-party library, you'll need to install it first with `install.packages('readr')`. After the installation is complete, load the `readr` library using `library("readr")` to utilize the `read_csv()` function.

```
# Load readr
library("readr")

# Read CSV into DataFrame
read_csv = read_tsv('/Users/admin/file.txt')
print(read_csv)
```

## 5. Conclusion

In this article, you have learned how to read or import data from a single text file (txt) and multiple text files into a DataFrame by using `read.table()` and `read.delim()` and `read_tsv()` from the `readr` package with examples.

## Related Articles

## References