

How can I export data as a CSV file in R using the write.csv() function?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I export data as a CSV file in R using the write.csv() function?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=149975>

The "write.csv()" function in R is used to export data as a CSV file. CSV (Comma Separated Values) is a common file format for storing tabular data. To use the "write.csv()" function, you need to first create a data frame in R with the desired data. Then, specify the name and location of the file where you want to save the data. The function will automatically convert the data frame into a CSV file and save it in the specified location. This allows for easy sharing and importing of data between different programs and platforms. Using the "write.csv()" function is a simple and efficient way to export data as a CSV file in R.

Use `write.csv()` to export R DataFrame to a CSV file with fields separated by comma delimiter, header (column names), rows index, and values surrounded with double quotes. You can also override this default behavior and export CSV without a header, without a row index or number, with no quotes, etc.

CSV file format is the easiest way to store scientific, analytical, or structured data (two-dimensional with rows and columns). Data in CSV is separated by delimiter most commonly comma (,) but you can also use any character like pipe, tab e.t.c

In my last article, I explained how to import a CSV file into a Data Frame, in this article, I will explain how to write or export a DataFrame to a CSV file by using different methods and their options.

1. Quick Examples

The following are quick examples of how to write/export a CSV file in R with and without header, with and without row number/index, encoding, etc.

```
# Quick Examples
```

```
# Example 1: Export to CSV without row names
```

```
write.csv(df,file='/Users/admin/new_file.csv', row.names=FALSE)
```

```
# Example 2: Export to CSV with blank for NA
```

```
write.csv(df,file='/Users/admin/new_file.csv',na="")
```

```
# Example 3: Export to CSV without quotes
```

```
write.csv(df,file='/Users/admin/new_file.csv',quote=FALSE)
```

```
# Example 4: Export to CSV without header
```

```
write.table(df,file='/Users/admin/new_file.csv',col.names=FALSE)
```

```
# Example 5: Write with UTF-8 Encoding
```

```
write.csv(df,file='/Users/admin/new_file.csv',fileEncoding = "UTF-8")
```

```
# Example 6: Load readr package
```

```
library(readr)
```

```
# Write to CSV file
```

```
write_csv(df, "/Users/admin/new_file.csv")
```

```
# Example 7: Load data.table package
```

```
library(data.table)
```

```
# Write to CSV file
```

```
fwrite(df, "/Users/admin/new_file.csv")
```

Let's create an R DataFrame, run these examples and explore the output.

```
# Create DataFrame
```

```
df <- data.frame(
```

```
id = c(10,11,12,13),
```

```
name = c('sai','ram','deepika','sahithi'),
```

```
gender = c('M','M',NA,'F'),
```

```
dob = as.Date(c('1990-10-02','1981-3-24','1987-6-14','1985-8-16')),
```

```
state = c('CA','NY',NA,NA)
```

```
)
```

```
df
```

Yields below output.

	id	name	gender	dob	state
1	10	sai	M	1990-10-02	CA
2	11	ram	M	1981-03-24	NY
3	12	deepika	<NA>	1987-06-14	<NA>
4	13	sahithi	F	1985-08-16	<NA>

2. Export to CSV in R Using write.csv()

R base functions provide a `write.csv()` to export the DataFrame to a CSV file. By default, the exported CSV file contains headers, row index, missing data as NA values, and columns separated by comma delimiter.

Following is the syntax of the `write.csv()`

```
# Syntax of write.csv()  
write.csv(input_dataframe, file=path, row.names = FALSE, ...)
```

2.1 Export CSV Example

Now, write the R DataFrame to a CSV file using the above syntax.

```
# Export DataFrame to CSV file  
write.csv(df, file='/Users/admin/new_file.csv')
```

It exports the contents from the R DataFrame object `df` to a CSV file called `new_file.csv` at the specified location. In the below examples, I will cover how to exclude these in the exported CSV file.

```
new_file.csv x
```

1	""	"id"	"name"	"gender"	"dob"	"state"
2	"1"	10	"sai"	"M"	1990-10-02	"CA"
3	"2"	11	"ram"	"M"	1981-03-24	"NY"
4	"3"	12	"deepika"	NA	1987-06-14	NA
5	"4"	13	"sahithi"	"F"	1985-08-16	NA
6						

Export or write CSV

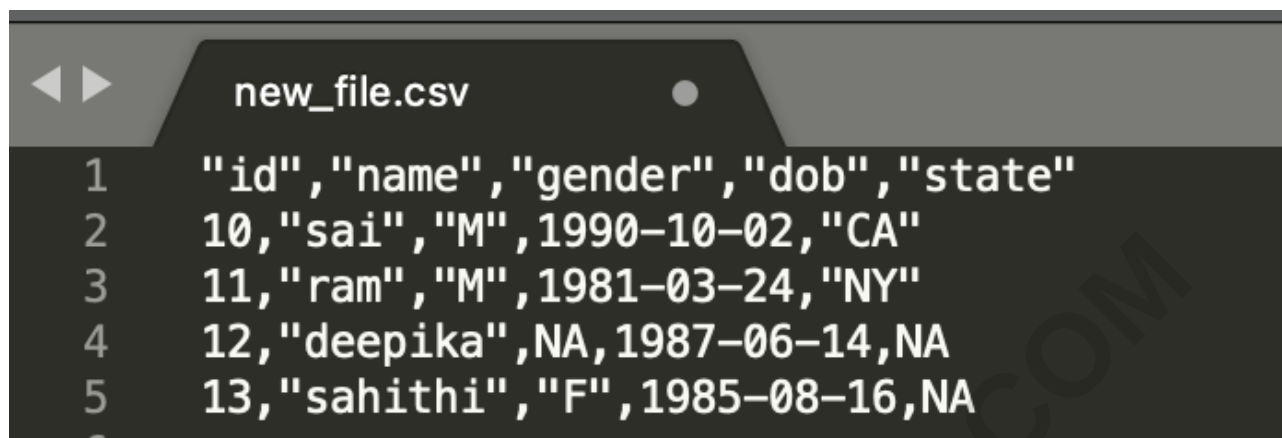
If you want to write with different column names, rename DataFrame column names in R prior to writing.

2.2. Export CSV without Index or Row number

To export without row numbers/index use `row.names` argument on R `write.csv()` function, by default this argument is set to `TRUE` mean it exports the row numbers. The below examples write/export CSV without row numbers in R.

```
# Export to CSV without row names
```

```
write.csv(df,file='/Users/admin/new_file.csv', row.names=FALSE)
```



The screenshot shows a CSV file named 'new_file.csv' with the following content:

	id	name	gender	dob	state
1	10	sai	M	1990-10-02	CA
2	11	ram	M	1981-03-24	NY
3	12	deepika	NA	1987-06-14	NA
4	13	sahithi	F	1985-08-16	NA

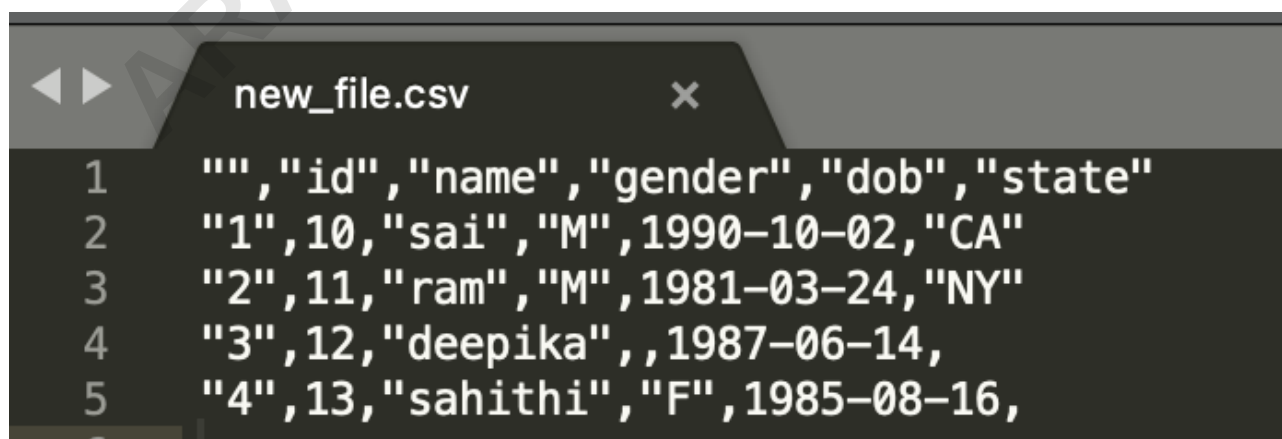
Export CSV without Index or Row Number

2.3. Write CSV NA as Blank (Empty String)

By default, all missing values are represented as NA on R DataFrame, and by exporting this to CSV also results in NA values for missing data. But sometimes you may want to export by replacing NA with Blank or an empty string. You can change this default behavior by exporting `na` argument with the blank or empty string on `write.csv()`.

```
# Export to CSV with blank for NA
write.csv(df,file='/Users/admin/new_file.csv',na='')
```

This writes or exports the CSV with a blank for NA values.



The screenshot shows a CSV file named 'new_file.csv' with the following content:

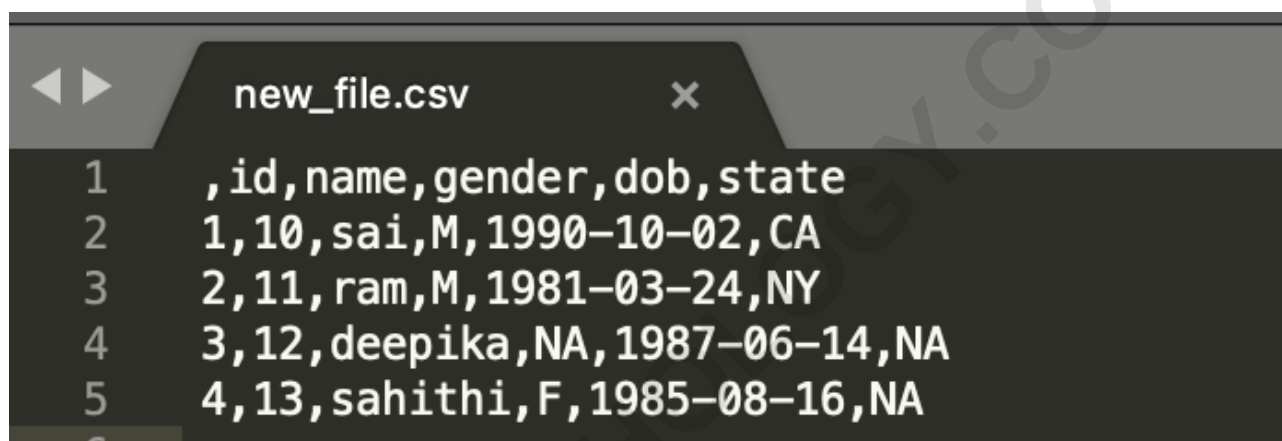
	id	name	gender	dob	state
1	10	sai	M	1990-10-02	CA
2	11	ram	M	1981-03-24	NY
3	12	deepika		1987-06-14	
4	13	sahithi	F	1985-08-16	

Export CSV with a blank or empty string for NA

2.4. Export CSV without quotes

As you notice above export outputs, it writes every cell within quotes, If you want to write without quotes in CSV use `quote=FALSE` argument. By default, this is set to `TRUE` hence, `write.csv()` always exports with quotes.

```
# Export to CSV without quotes
write.csv(df, file='/Users/admin/new_file.csv', quote=FALSE)
```



R write CSV without quotes

2.5. Write CSV file with Encoding

Use `fileEncoding='UTF-8'` as argument to `write.csv()` to export DataFrame to CSV with UTF-8 encoding.

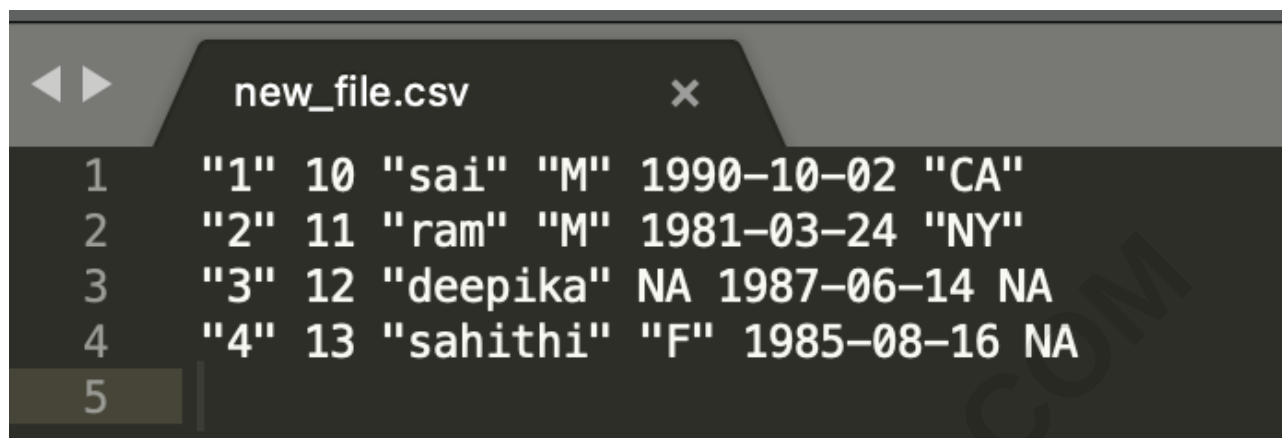
```
# Write with UTF-8 Encoding
write.csv(df, file='/Users/admin/new_file.csv', fileEncoding = "UTF-8")
```

3. Export CSV without Header using write.table()

By using `write.csv()` you can't export without a header in R however, you can use `write.table()` function with `col.names=FALSE` argument to export the data from DataFrame to CSV without header or column names.

```
# Export to CSV without header
```

```
write.table(df, file='/Users/admin/new_file.csv', col.names=FALSE)
```



1	"1"	10	"sai"	"M"	1990-10-02	"CA"
2	"2"	11	"ram"	"M"	1981-03-24	"NY"
3	"3"	12	"deepika"	NA	1987-06-14	NA
4	"4"	13	"sahithi"	"F"	1985-08-16	NA
5						

Export CSV without header

4. Use write_csv() from readr package

If you are working with larger data, you should use the `write_csv()` function from the `readr` package. `readr` is a third-party library hence, in order to use `readr` library, you need to first install it by using `install.packages('readr')`. Once installation completes, load the `readr` library in order to use this `write_csv()` method. To load a library in R use `library("readr")`.

```
# Load readr package
```

```
library(readr)
```

```
# Write to CSV file
```

```
write_csv(df, "/Users/admin/new_file.csv")
```

5. Use fwrite() from data.table package

Finally, let's use the `data.table` package to write CSV in R. Using this is the best approach when you are writing very large data sets.

```
# Load data.table package
```

```
library(data.table)
```

```
# Write to CSV file
```

```
fwrite(df, "/Users/admin/new_file.csv")
```

6. Conclusion

In this article, I have explained several R examples of how to export a CSV file without a header, without row numbers, without quotes e.t.c. Also, explained how to write CSV using write_csv() from readr and fwrite() from data.table package in R.

Related Articles

References

ARABPSYCHOLOGY.COM