

# How to use write.xlsx in R (With Examples)

Authored by  
**stats writer**

November 29, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to use write.xlsx in R (With Examples)*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=101725>

write.xlsx is an R package that allows users to easily write data from R into an excel spreadsheet. To use it, you must first install and load the package into your R workspace. After that, you must use the write.xlsx() function to write the data you want to the excel spreadsheet. This function requires the data you want to write and the name of the excel file you want to create. Once the write.xlsx() command is executed, the new excel file will be created in the specified location. An example of using write.xlsx() can be seen below: write.xlsx(data, "my\_data.xlsx") This will create an excel file named "my\_data.xlsx" with the data specified in the data variable.

You can use the **write.xlsx** function in R to write a data frame to an Excel workbook.

This function uses the following basic syntax:

```
write.xlsx(x, file, sheetName = "Sheet1", ...)
```

where:

**x**: Name of data frame

**file**: path to output file

**sheetName**: Sheet name to appear in workbook. Default is "Sheet1"

The following step-by-step example shows how to use the **write.xlsx** function in practice.

## Step 1: Install & Load xlsx Package

First, we must install and load the **xlsx** package in order to use the **write.xlsx** function:

```
install.packages('xlsx')
```

```
library(xlsx)
```

## Step 2: Create the Data Frame

Next, let's create the following data frame in R:

```
#create data frame
```

```
df <- data.frame(team=c('A', 'B', 'C', 'D', 'E'),
```

```
points=c(99, 90, 86, 88, 95),
```

```
assists=c(33, 28, 31, 39, 34),
```

```
rebounds=c(30, 28, 24, 24, 28))
```

```
#view data frame
```

df

team points assists rebounds

1 A 99 33 30

2 B 90 28 28

3 C 86 31 24

4 D 88 39 24

5 E 95 34 28

### Step 3: Use write.xlsx to Export Data Frame to Excel File

Next, let's use `write.xlsx()` to write the data frame to a file called `my_data.xlsx`:

```
#write data frame to Excel file
```

```
write.xlsx(df, 'my_data.xlsx')
```

The file will automatically be written into the current .

If I navigate to the current working directory, I can find this Excel file:

	A	B	C	D	E	F	G
1		team	points	assists	rebounds		
2	1	A	99	33	30		
3	2	B	90	28	28		
4	3	C	86	31	24		
5	4	D	88	39	24		
6	5	E	95	34	28		
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

The values in the Excel workbook match those from the data frame.

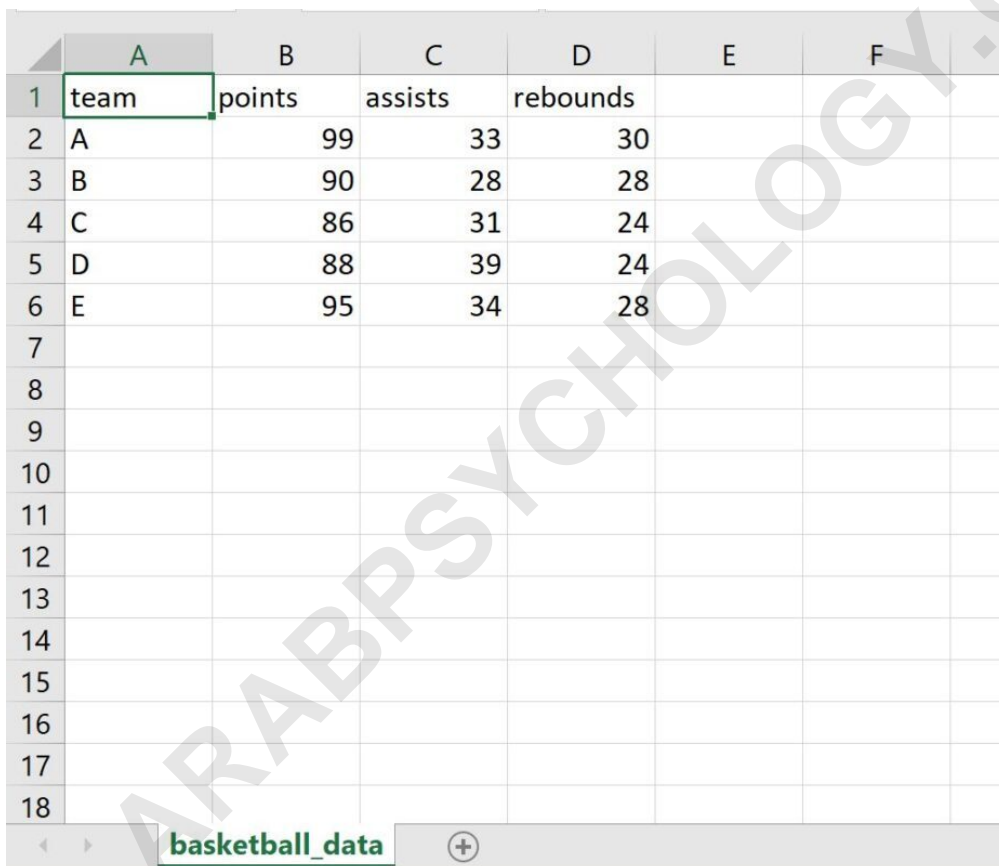
### Step 4 (Optional): Use write.xlsx with Custom Arguments

Note that you can also use the following syntax to change the sheet name in the Excel workbook and suppress the row names:

**#write data frame to Excel file**

```
write.xlsx(df, 'my_data.xlsx', sheetName = 'basketball_data', row.names=FALSE)
```

If I navigate to the current working directory, I can find this Excel file:



	A	B	C	D	E	F
1	team	points	assists	rebounds		
2	A	99	33	30		
3	B	90	28	28		
4	C	86	31	24		
5	D	88	39	24		
6	E	95	34	28		
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						

Notice that the sheet name has changed and the first column no longer contains the row numbers.

The following tutorials explain how to export other types of files in R: