

How do I use the na.rm argument in R? Can you provide some examples?

Authored by
stats writer

May 2, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I use the na.rm argument in R? Can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=141923>

The na.rm argument in R is used to remove any missing or "NA" values from a data set before performing a calculation or analysis. This argument is commonly used in functions such as mean(), sum(), and median() to exclude missing values from the calculation. To use the na.rm argument, simply include it in the function along with the data set. For example, mean(x, na.rm = TRUE) will calculate the mean of the data set x without including any NA values. Similarly, sum(y, na.rm = TRUE) will sum all the values in the data set y while ignoring any NA values. Overall, the na.rm argument is a useful tool for ensuring accurate and complete analysis of data in R.

Use na.rm in R (With Examples)

You can use the argument na.rm = TRUE to exclude missing values when calculating descriptive statistics in R.

**#calculate mean and exclude missing values
mean(x, na.rm = TRUE)**

**#calculate sum and exclude missing values
sum(x, na.rm = TRUE)**

**#calculate maximum and exclude missing values
max(x, na.rm = TRUE)**

**#calculate standard deviation and exclude missing values
sd(x, na.rm = TRUE)**

The following examples show how to use this argument

in practice with both vectors and data frames.

Example 1: Use na.rm with Vectors

Suppose we attempt to calculate the mean, sum, max, and standard deviation for the following vector in R that contains some missing values:

```
#define vector with some missing values
```

```
x <- c(3, 4, 5, 5, 7, NA, 12, NA, 16)
```

```
mean(x)
```

```
NA
```

```
sum(x)
```

```
NA
```

```
max(x)
```

```
NA
```

```
sd(x)
```

```
NA
```

Each of these functions returns a value of NA.

To exclude missing values when performing these calculations, we can simply include the argument na.rm = TRUE as follows:

```
#define vector with some missing values
```

```
x <- c(3, 4, 5, 5, 7, NA, 12, NA, 16)
```

```
mean(x, na.rm = TRUE)
```

```
7.428571
```

```
sum(x, na.rm = TRUE)
```

```
52
```

```
max(x, na.rm = TRUE)
```

```
16
```

```
sd(x, na.rm = TRUE)
```

```
4.790864
```

Notice that we were able to complete each calculation successfully while excluding the missing values.

Example 2: Use na.rm with Data Frames

Suppose we have the following data frame in R that contains some missing values:

```
#create data frame
```

```
df <- data.frame(var1=c(1, 3, 3, 4, 5),  
var2=c(7, 7, NA, 3, 2),  
var3=c(3, 3, NA, 6, 8),  
var4=c(1, 1, 2, 8, NA))
```

```
#view data frame
```

```
df
```

```
var1 var2 var3 var4
```

```
1 1 7 3 1
```

```
2 3 7 3 1
```

```
3 3 NA NA 2
```

```
4 4 3 6 8
```

```
5 5 2 8 NA
```

We can use the `apply()` function to calculate descriptive statistics for each column in the data frame and use the `na.rm = TRUE` argument to exclude missing values when performing these calculations:

```
#calculate mean of each column  
apply(df, 2, mean, na.rm = TRUE)
```

```
var1 var2 var3 var4  
3.20 4.75 5.00 3.00
```

```
#calculate sum of each column  
apply(df, 2, sum, na.rm = TRUE)
```

```
var1 var2 var3 var4  
16 19 20 12
```

```
#calculate max of each column  
apply(df, 2, max, na.rm = TRUE)
```

```
var1 var2 var3 var4  
5 7 8 8
```

```
#calculate standard deviation of each column  
apply(df, 2, sd, na.rm = TRUE)
```

```
var1 var2 var3 var4  
1.483240 2.629956 2.449490 3.366502
```

Once again, we were able to complete each calculation successfully while excluding the missing values.