How to fix: “rbind(deparse.level, …): numbers of columns of arguments do not match” when using the Fix function in R?

Authored by

stats writer

June 28, 2024

The error "rbind(deparse.level, ...): numbers of columns of arguments do not match" in the R programming language occurs when using the Fix function. This error indicates that the number of columns in the arguments being passed to the rbind function do not match, causing the function to fail. This can be resolved by ensuring that the number of columns in the arguments are consistent and identical.

Fix in R: error in rbind(deparse.level, ...) : numbers of columns of arguments do not match

One error you may encounter in R is:

Error in rbind(deparse.level, ...) :
numbers of columns of arguments do not match

This error occurs when you attempt to use the function in R to row-bind together two or more data frames that do not have the same number of columns.

This tutorial shares exactly how to fix this error.

How to Reproduce the Error

Suppose we have the following two data frames in R:

```
#create first data frame
df1 <- data.frame(x=c(1, 4, 4, 5, 3),
y=c(4, 4, 2, 8, 10))

df1
```

```
x y
1 1 4
2 4 4
3 4 2
4 5 8
5 3 10

#create second data frame
df2 <- data.frame(x=c(2, 2, 2, 5, 7),
y=c(3, 6, 2, 0, 0),
z=c(2, 7, 7, 8, 15))

df2

x y z
1 2 3 2
2 2 6 7
3 2 2 7
4 5 0 8
5 7 0 15
```

Now suppose we attempt to use rbind to row-bind these two data frames into one data frame:

```
#attempt to row-bind the two data frames together
```

rbind(df1, df2)

Error in rbind(deparse.level, ...) :
numbers of columns of arguments do not match

We receive an error because the two data frames do not have the same number of columns.

How to Fix the Error

There are two ways to fix this problem:

Method 1: Use rbind on Common Columns

One way to fix this problem is to use the intersect() function to find the common column names between the data frames and then row-bind the data frames only on those columns:

```
#find common column names
common <- intersect(colnames(df1), colnames(df2))

#row-bind only on common column names
df3 <- rbind(df1, df2)

#view result
```

```
df3

x y
1 1 4
2 4 4
3 4 2
4 5 8
5 3 10
6 2 3
7 2 6
8 2 2
9 5 0
10 7 0
```

Method 2: Use bind_rows() from dplyr

Another way to fix this problem is to use the bind_rows() function from the package, which automatically fills in NA values for column names that do no match:

```
library(dplyr)

#bind together the two data frames
df3 <- bind_rows(df1, df2)
```

```
#view result
df3

x y z
1 1 4 NA
2 4 4 NA
3 4 2 NA
4 5 8 NA
5 3 10 NA
6 2 3 2
7 2 6 7
8 2 2 7
9 5 0 8
10 7 0 15
```

Notice that NA values are filled in for the values from df1 since column z didn't exist in this data frame.

Additional Resources

The following tutorials explain how to troubleshoot other common errors in R:

How to Fix in R: longer object length is not a multiple of shorter object length