

How do I create a multi-line comment in R?

Authored by
stats writer

July 2, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I create a multi-line comment in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165913>

Creating multi-line comments in R allows users to add explanatory or descriptive notes within their code without affecting the program's functionality. This is useful for documenting the code and making it easier for others to understand. To create a multi-line comment in R, begin with a hash symbol (#) followed by a space before each line of comment. Alternatively, you can use the command "comment()" to insert a multi-line comment. Here are some examples:

```
# This is a multi-line comment  
# that spans multiple lines  
# and can be used to explain the code.
```

```
# Another way to create a multi-line comment is by using the command:  
comment("This is a multi-line comment that can be inserted using the comment() command.  
It can also include special characters like $, %, &, etc.")
```

```
# Multi-line comments can also be used to temporarily disable a block of code without deleting it.  
# For example:  
# comment out this block of code  
# to prevent it from executing.
```

Create a Multi-Line Comment in R (With Examples)

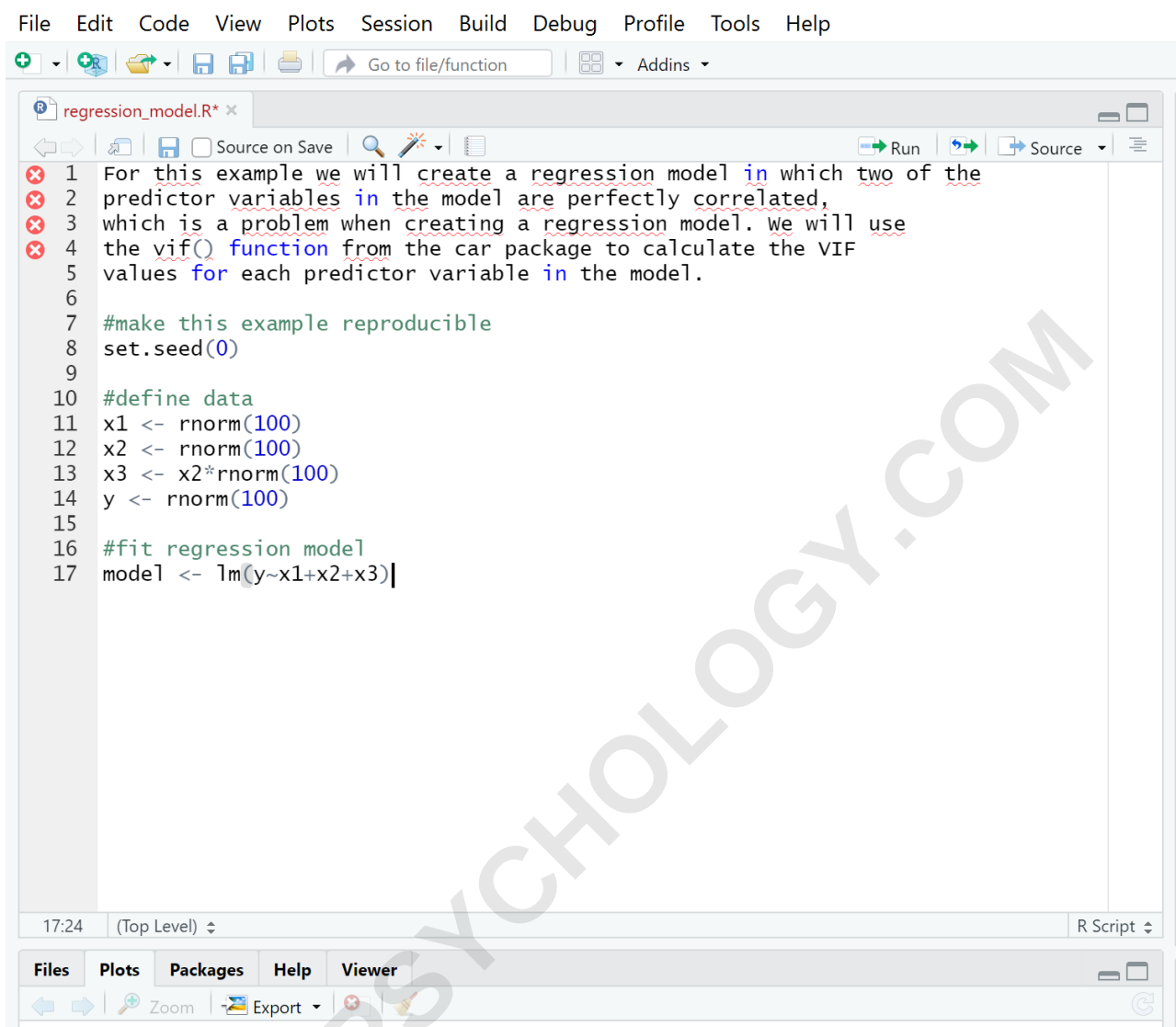
The easiest way to create a multi-line comment in RStudio is to highlight the text and press Ctrl + Shift + C.

You can just as easily remove the comment by highlighting the text again and pressing Ctrl + Shift + C.

The following example shows how to do this in practice.

Example: Create Multi-Line Comment in R

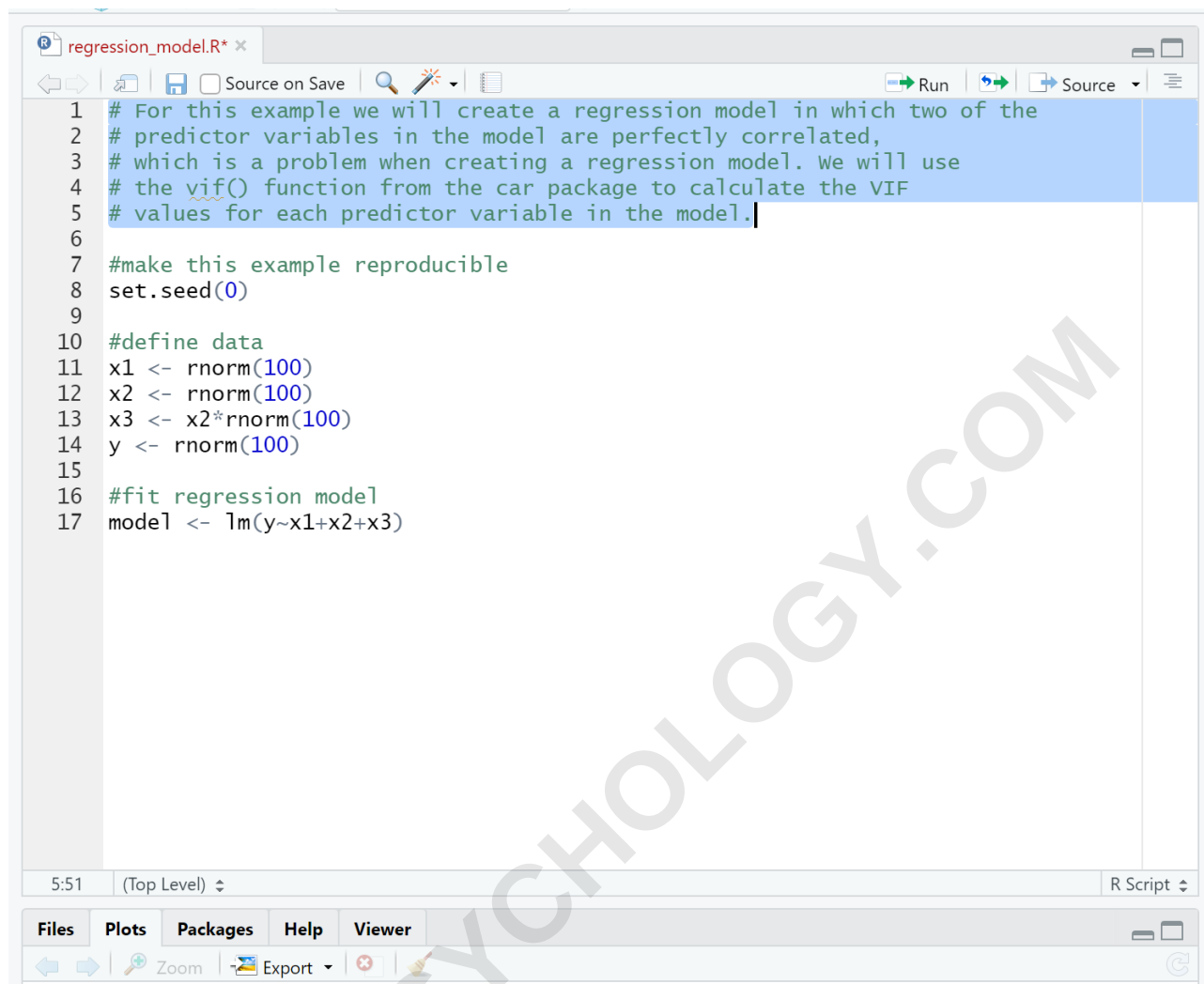
Suppose we have the following script in RStudio:



```
1 For this example we will create a regression model in which two of the
2 predictor variables in the model are perfectly correlated,
3 which is a problem when creating a regression model. We will use
4 the vif() function from the car package to calculate the VIF
5 values for each predictor variable in the model.
6
7 #make this example reproducible
8 set.seed(0)
9
10 #define data
11 x1 <- rnorm(100)
12 x2 <- rnorm(100)
13 x3 <- x2*rnorm(100)
14 y <- rnorm(100)
15
16 #fit regression model
17 model <- lm(y~x1+x2+x3)
```

The first five rows of the script simply describe what the rest of the script does. Thus, we want to turn these rows into comments.

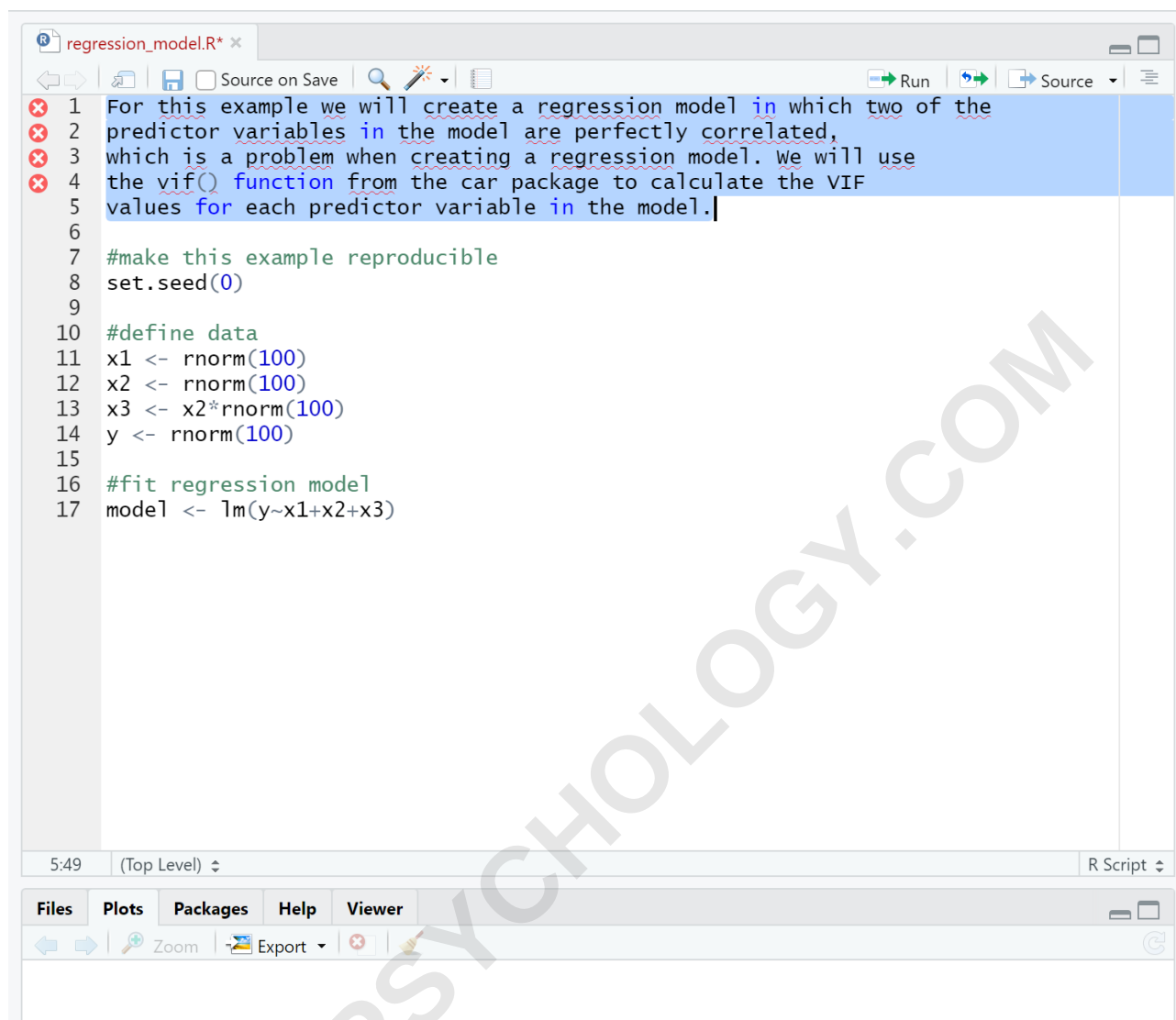
To turn the first five lines into a multi-line comment, we can highlight each of the first five rows and press **Ctrl + Shift + C**:



```
1 # For this example we will create a regression model in which two of the
2 # predictor variables in the model are perfectly correlated,
3 # which is a problem when creating a regression model. We will use
4 # the vif() function from the car package to calculate the VIF
5 # values for each predictor variable in the model.
6
7 #make this example reproducible
8 set.seed(0)
9
10 #define data
11 x1 <- rnorm(100)
12 x2 <- rnorm(100)
13 x3 <- x2*rnorm(100)
14 y <- rnorm(100)
15
16 #fit regression model
17 model <- lm(y~x1+x2+x3)
```

Notice that the # symbol has been placed in front of each line, which indicates that the entire line is a comment.

We can just as easily uncomment the first five rows by highlighting each row and pressing Ctrl + Shift + C once again:



```
1 For this example we will create a regression model in which two of the
2 predictor variables in the model are perfectly correlated,
3 which is a problem when creating a regression model. We will use
4 the vif() function from the car package to calculate the VIF
5 values for each predictor variable in the model.
6
7 #make this example reproducible
8 set.seed(0)
9
10 #define data
11 x1 <- rnorm(100)
12 x2 <- rnorm(100)
13 x3 <- x2*rnorm(100)
14 y <- rnorm(100)
15
16 #fit regression model
17 model <- lm(y~x1+x2+x3)
```

Note that the # symbol has been removed from each of the first five lines, which indicates that each line is no longer a comment.

Note that you can press command + Shift + C to perform the exact same action in OS-X.

Additional Resources