

**Getting help beyond the help files**

# We all encounter problems when writing R code

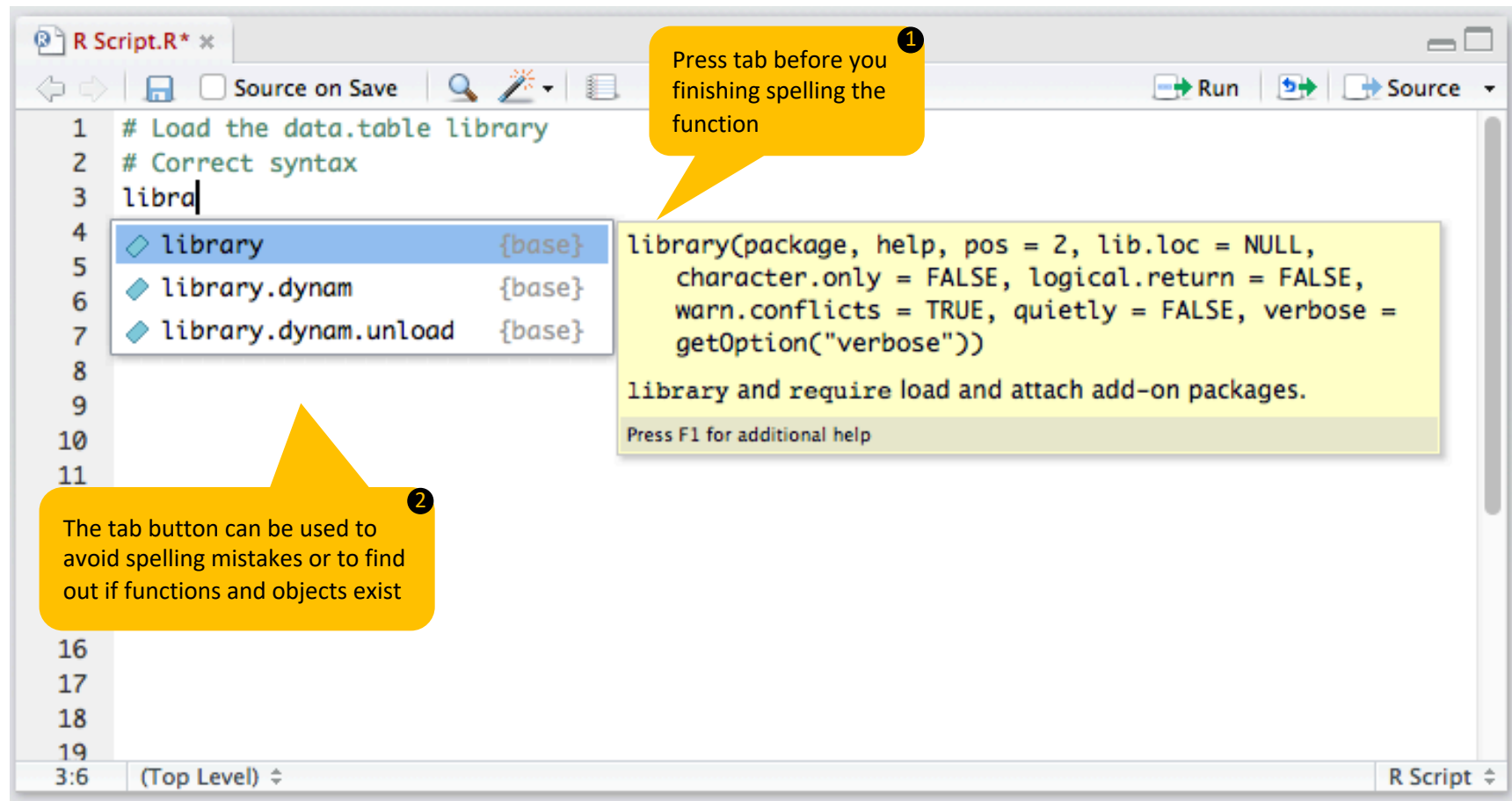
The following types of problems can arise when writing R code:

1. Incorrect syntax
2. Exceptions (i.e. error messages)
3. Incorrect output
4. Incomplete knowledge
5. Slow code, messy code, collaboration chaos, etc.

1  
Including no output and warning messages

2  
These advanced topics do not necessarily hinder us from getting the output we want but are annoyances to the advanced R user.

# A way to avoid incorrect syntax is by pressing the tabulate button to autocomplete your code



## Sidenote: Commenting code

- Comments are important to document why you did something the way you did it – trust us, at the moment you write the code everything seems obvious, but you won't remember this 3 weeks later!
- To write a comment use the hashtag-sign (#). The rest of the particular line will be a comment and won't be interpreted as R code.

```
# This is a comment!
```

```
variableName <- (...) # This is also a comment!
```

To solve all other problems when writing R code you have two best friends

The Google logo, consisting of the word "Google" in its characteristic multi-colored font: blue 'G', red 'o', yellow 'o', blue 'g', green 'l', and red 'e'.The Stack Overflow logo, featuring a stylized orange sunburst icon above a gray square base.

**stackoverflow**

# When using Google, carefully craft search queries to find useful results faster

- Bad question:

I can't get my data to load correctly!



- Better question:

How to load data in r



- Good question:

How to load CSV data with the data.table package in r



Be specific and identify the core of your question.

If you can't find an answer through Google, look and ask on [www.stackoverflow.com](http://www.stackoverflow.com)

Stackoverflow is the largest online community for programmers to learn and share their knowledge:



# How to post a good question on Stackoverflow?

Obviously, the answer to this questions has been given on [www.stackoverflow.com](http://stackoverflow.com) itself:

<http://stackoverflow.com/questions/5963269/how-to-make-a-great-r-reproducible-example>

The screenshot shows a web browser window with the URL `stackoverflow.com/questions/5963269/how-to-make-a-great-r-reproducible-example`. The page features the Stack Overflow logo and navigation tabs: Questions, Tags, Users, Badges, and Unanswered. A search bar is located in the top right corner.

The question title is "How to make a great R reproducible example?". The question body contains the following text:

When discussing performance with colleagues, teaching, sending a bug report or searching for guidance on mailing lists and here on SO, a reproducible example is often asked and always helpful.

What are your tips for creating an excellent example? How do you paste data structures from `r` in a text format? What other information should you include?

Are there other tricks in addition to using `dput()`, `dump()` or `structure()`? When should you include `library()` or `require()` statements? Which reserved words should one avoid, in addition to `c`, `df`, `data`, etc?

How does one make a great `r` reproducible example?

The question has 237 votes and 154 answers. It is tagged with `r` (24592 votes), `example` (457 votes), and `r-faq` (40 votes). The question was asked 1 year ago, viewed 11384 times, and is active 2 days ago.

Linked questions include:

- `'levels<-'` What sorcery is this?
- Learning R. Where does one Start?
- Given a set of random numbers drawn from a continuous univariate distribution, find the distribution
- Methods adding Excel like features



# Most important points when asking a questions on Stackoverflow

1. Add a clear title to your question:
  - Catchy description of the issue
  - Relevant context (e.g. problem with function in a particular R package)
2. In the body of your question provide:
  - Minimal information to understand your question
  - Toy data
  - Code to reproduce your error or incorrect output, and demonstrate that you have attempted to solve your problem
3. Apply relevant tags

# This is how an exemplary question on Stackoverflow would look

## Counting in R data.table

Clear question with information on the particular R package <sup>1</sup>

I have the following data.table

25  
▼  
★  
8

```
set.seed(1)
DT <- data.table(VAL = sample(c(1, 2, 3), 10, replace = TRUE))
      VAL
1:     1
2:     2
3:     2
...
10:    1
```

Toy data <sup>2</sup>

Now I want to perform two tasks:

1. Count the occurrences of numbers in `VAL`.
2. Count within all rows with the same value `VAL` (first, second, third occurrence)

Task that could not be solved <sup>3</sup>

At the end I want the result

	VAL	COUNT	IDX
1:	1	3	1
2:	2	4	1
3:	2	4	2
...			
10:	1	3	3

How the result would look like for the toy data <sup>4</sup>

where `COUNT` defines task 1. and `IDX` task 2.

I tried to work with `which` and `length` using `.I`:

```
dt[, list(COUNT = length(VAL == VAL[.I]),
          IDX = which(which(VAL == VAL[.I]) == .I))]
```

OP demonstrated that they attempted to solve the problem <sup>5</sup>

but this does not work as `.I` refers to a vector with the index, so I guess one must use `.I[]`. Though inside `.I[]` I again face the problem, that I do not have the row index and I do know (from reading `data.table` FAQ and following the posts here) that looping through rows should be avoided if possible.

Task that could not be solved <sup>6</sup>

So, what's the `data.table` way?

r data.table

Relevant tags <sup>7</sup>

**Getting help beyond the help files**