

# Common errors and how to fix them

## Introduction to R - Day 1

Instructor: **Selina Baldauf**

Freie Universität Berlin - Theoretical Ecology

2021-08-01 (updated: 2023-02-26)

When learning a programming language, you have to be prepared to spend a lot of your time with fixing bugs in the code. So don't worry: It's not you, it's just how programming works!

## debugging



1.  
I got this.



2.  
Huh. Really  
thought that  
was it.



3.  
(...)



4.  
Fine. Restarting.



5.  
OH WTF.



6.  
Zombie  
meltdown



7.



8.  
A NEW HOPE!



9.  
[insert awesome  
theme song]




10.  
I ♥ CODING!

# Debugging

Debugging can be annoying and we can't avoid it, but ...

- ... it's an **effective learning experience** (I actually learned the most from debugging my code)
- ... it will get easier over time
- ... there are some **debugging techniques** to decrease the time in stages 2-7
- ... if nothing helps, there are great people all over the internet willing to help



Most common  errors and problems for  
beginners and how to deal with them


# Syntax errors

## Example

```
mean(c(1,2,3)na.rm=TRUE)
```

```
## Error: <text>:1:14: unexpected symbol
## 1: mean(c(1,2,3)na.rm
##                      ^
```

## How to fix

- look for missing commas, misspelled arguments, ...
- read the error message
- the RStudio syntax checker warns you before you run code with syntax errors
  - look for  next to line numbers in your script

```
306
307
308 mean(c(1,2,3)na.rm=TRUE)
309
```

expected ',' after expression  
expected whitespace around '=' operator

# Error: could not find function

## Examples

```
##      b  a
## 1    1  1
## 2    2  2
## 3    3  3
## 4    4  4
## 5    5  5
## 6    6  6
## 7    7  7
## 8    8  8
## 9    9  9
## 10  10 10
```

```
## Error in lenght(1:10): could not find function "lenght"
```

## How to fix

Could not find function errors have two main reasons:

1. You forgot to load the package that the function belongs to

• load the package using `library()` or call the function with

# Error: object `x` not found

## Example

```
## Error in eval(expr, envir, enclos): object 'hello' not found
```

```
## Error in eval(expr, envir, enclos): object 'variable_A' not found
```

## How to fix

- you are trying to access an object that does not exist
- Mostly because:
  - typos in variable name (variable name is `variableA` but you try to access `variable_A`)
  - forgot to put quotes around string: `print(hello)` → looks for a variable named `hello` but instead you wanted to print the string `print("hello")`

# Wrong data format

## Example

- does not necessarily trigger an error message
- if there is an error message, it can also appear later in your code

## How to fix it

- Look at `str()` of your data and check whether all columns are there and in correct format
  - e.g. is a column of type `character` but should be of type `integer`?
- Do that at multiple locations in your script to find the line where the error actually happens
  - everytime you change something in your data, check its structure



# R crashes

Sometimes R crashes completely and you see this:



## How to fix it

- There is no fix but to start a new session
- Make sure to save your scripts regularly!


# Console prints +

R is not running code anymore and the console only prints `+` if you try to execute a command.



```
122:1 (Top Level) >
Console ~/
in apply(metric, ZL, consum) : NAS introduced by coercion
> barplot(biodata$richness, names.arg=c("beetle",
+ "bird",
+ "Butterfly",
+ "Dragonfly",
+ "Fl.Plants",
+ "Fungus",
+ "hymenopteran",
+ "Lichen",
+ "Liverwort",
+ "Mammal",
+ "Mollusc"),
+ xlab="Taxa", ylab="Number of species", ylim=c(0,600), cex.axis=1.5, cex.lab=1.5)
```

## How to fix it

- First, go to the console and hit `Escape`. Then you should see the `>` sign instead of `+` again.
- Likely you forgot to close a bracket somewhere. Go to your script and check where this happened
  - look for  next to line numbers

# Warnings

R can give you warnings for many reasons, e.g.

- you have `NA` values in your data and try to plot them
- implicit type conversion returned `NA`
- the function you are using is deprecated
- the package you are using was built for another version of R

Warnings are no errors and can sometimes be ignored but:

- make sure to read and understand warnings
- only ignore them if you know that that's okay, otherwise fix the underlying issue

# How to troubleshoot R code

A step by step guide

# Troubleshoot R: Step by step

Often, you don't need to do all the steps but a systematic approach to bug fixing is very helpful.

**Step 1:** Carefully read the error message and try to fix it

**Step 2:** Is it any of the errors you learned about just now?

**Step 3:** If the error is about data or other variables: look at the structure using `str()`

**Step 4:** If the error is about a function: Read the documentation using `?functionName`.

- Did you use the function correctly?
- Did you forget an argument?

**Step 5:** Look for answers online

- often you can also jump directly to this step

**Step 6:** Ask others for help

# Step 5: Look for answers online

- Search with keywords R + package name + Error message/Warning
- If you don't know how to do something try searching R + package name + What you want to do, e.g.
  - "R ggplot change axis title"
  - "R sort vector"
- Usually you can pick any of the top search results, but I recommend results from [Stack Overflow](#)
- Always search in English to get more results

💡 Change language of R messages to English with `sys.setenv(LANGUAGE='en')`

# Step 6: Ask others for help

There are plenty of places where you can ask for help online. Some common and good options are:

- Ask a question on [Stack Overflow](#)
- Ask in the [R Discord server](#)
  - usually fast answers
  - but: you need a Discord account
- Ask on Twitter using the [#rstats](#) hashtag (maybe a bit less common)

**But:** You have to make sure that before, you tried all the other 5 steps.

To ask questions online, you have to learn **how to ask a good R question**. This includes:

- clear question
- reproducible example

Look [here](#) for more info on how to ask a good question about R