R4DS - Unit 2: Debugging

Andrea Gilardi
February 29, 2024
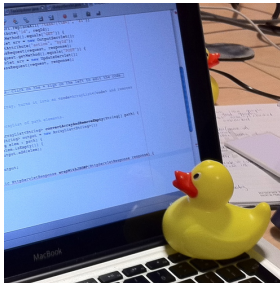
UNIVERSITA' DEGLI STUDI DI MILANO

BICOCCA

Outline and main concepts

- The objective of this unit is to learn the basic tools for debugging R code using R and Rstudio.

- We will start with a little of theory and then I will introduce the most common approaches with a series of example.

- At the end, we will just briefly recall the main venues that can be consulted when asking for help and the best practices required to interact with those systems.

But first...

- Let's start with a quote by Norm Matloff: *"Finding your bug is a process of confirming the many things that you believe are true — until you find one which is not true"*...

- which let me introduce our **best** programming companion: the debugging duck!

Rubber Duck Debugging

1. Beg, borrow, steal, buy, fabricate or otherwise obtain a rubber duck (bathtub variety).

2. Place rubber duck on desk and inform it you are just going to go over some code with it, if that's all right.

3. Explain to the duck what your code is supposed to do, and then go into detail and explain your code line by line.

4. At some point you will tell the duck what you are doing next and then realise that that is not in fact what you are actually doing. The duck will sit there serenely, happy in the knowledge that it has helped you on your way.

Just to be a little more concrete...

Now we are going to review the following functions:

- `traceback()`: Inspect the call stack;

- `browser()`: Manually enter into debug state;

- `debug()`: Set the debug state for any function;

- `try() + tryCatch()`: Condition handling!

Enough theory, let's start coding 🥳

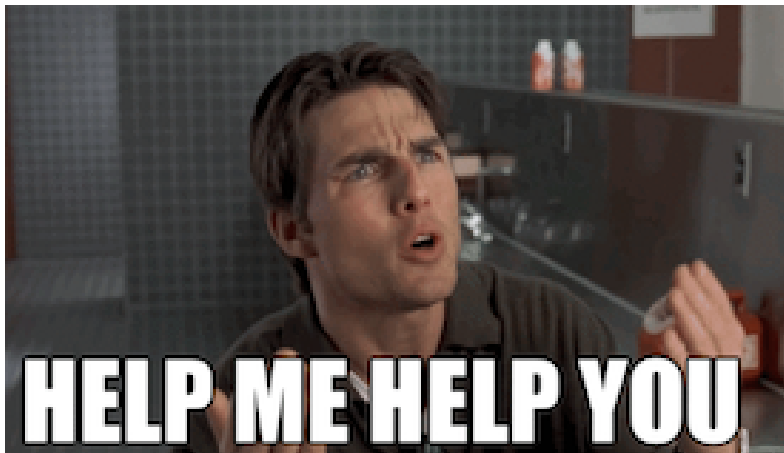# And if I cannot debug my code?

- You need to ask for help! *But where?* 😭

- Google your problem! Maybe you are lucky, usually no...

- If you face a problem with a function in a contributed package, you should contact its maintainer and, whenever possible, raise an issue in the corresponding github repo.

- If you have a generic programming question, you can ask it in several forums (e.g. SO) or mailing lists.

- In any case, please remember that the people maintaining the software and the package are volunteers and you should present your problem in the best possible format (see also `fortunes::fortune(107)` and this SO thread).

`reprex::reprex()`