

Debugging

Xiaojuan Li & Nathaniel MacNell

EPID 799B

Fall 2016

Debugging overview

- Good programming practices (avoiding debugging)
- Debugging methods



When/why I face bugs

- Coding something new
 - Unexpected behavior of a new function
- Nth time
 - Typos & mistakes
 - Forgetting what I was doing
- Adapting / debugging others' code
 - Misunderstanding how it works (i.e. intermediate goals)

[I use an iterative/constructive programming approach, so most of my bugs are simple]

Avoiding/facilitating debugging

INVEST in your code (and time)

- Use comments (sections & in-line)
- Organize
- Choose the clearer approach
- Break down steps
- Test often (as simple as examining intermediates)

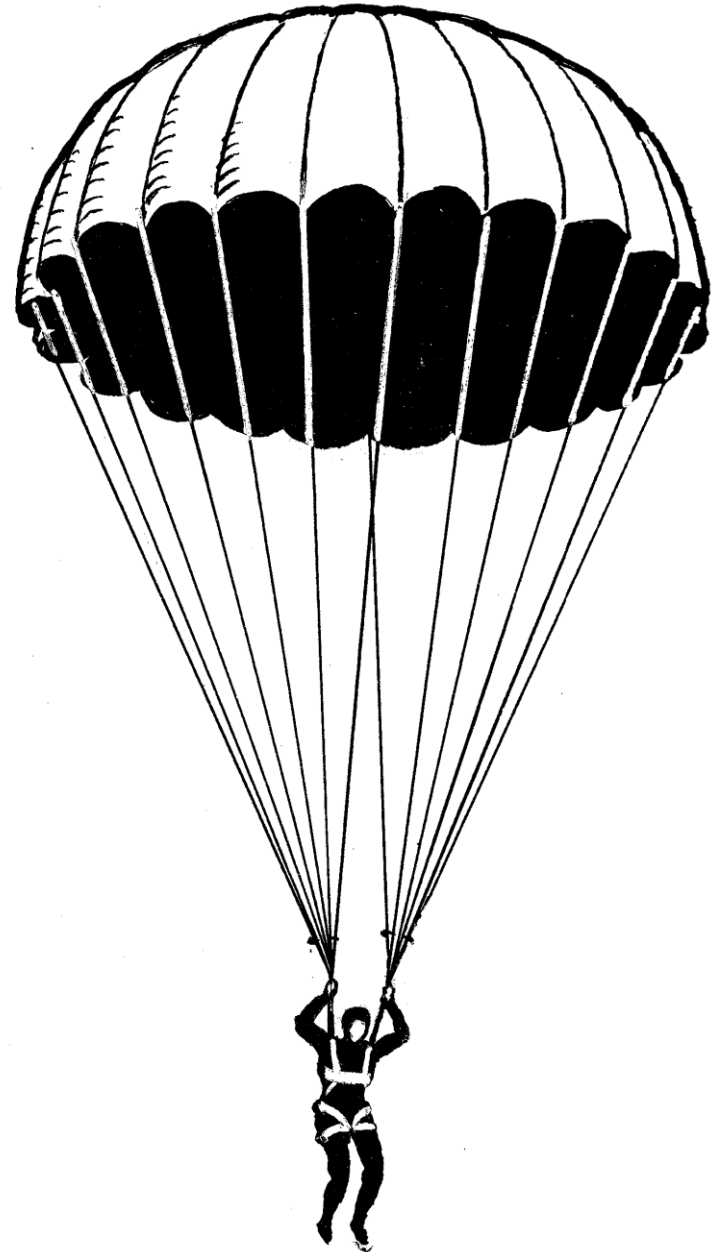
*I've never regretted **over**-investing, only **under**-investing (after it's too late to reasonable fix).*

A side note

Sometimes it's *better* to bail.

- Start over
- Use what you learned
- Code it better the next time

Often helpful for examples.



Debugging methods

From low to high complexity:

- Post-mortem
- Print debugging
- Stepwise execution
- Tracing
- Version control

Post-mortem analysis

If you break things up into smaller steps, R will often identify errors for you (at least those that won't run):

```
> rnorm("a", "b")  
Error in rnorm("a", "b") : invalid arguments
```

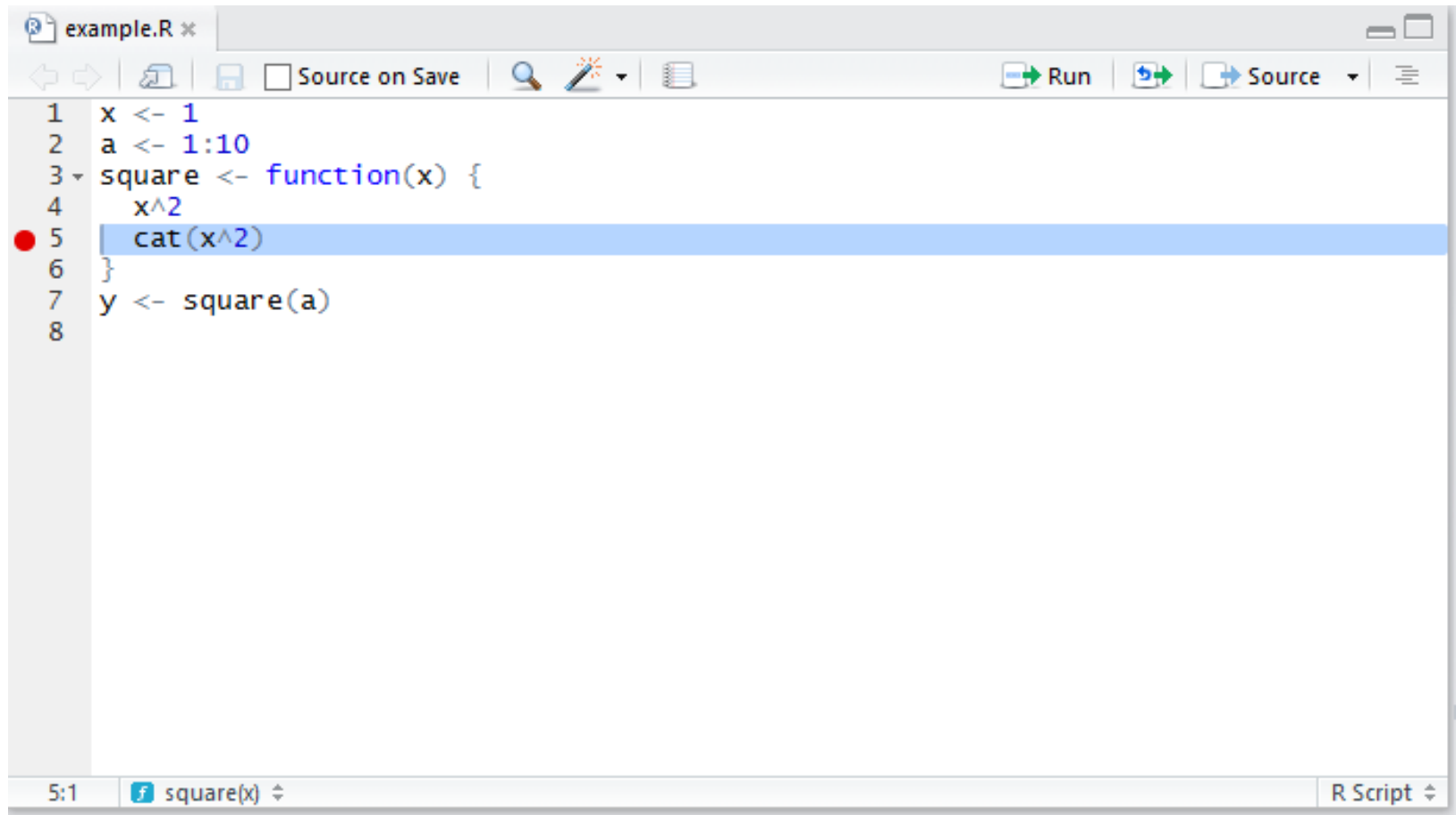
- Typically only catches obvious, literal errors

Print debugging

- Print out a debug value during a process to make sure it's working (often using `cat()`)

```
x <- 1:10
square <- function(x) {
  x^3
  cat(x^2)
}
y <- square(x)
```


Stepwise execution & tracing



The screenshot shows an R script editor window titled "example.R". The script contains the following code:

```
1 x <- 1
2 a <- 1:10
3 square <- function(x) {
4   x^2
5   cat(x^2)
6 }
7 y <- square(a)
8
```

The line `cat(x^2)` on line 5 is highlighted in blue, indicating it is the current step in the execution process. A red dot is visible to the left of line 5. The toolbar at the top includes icons for navigation, saving, and running. The status bar at the bottom shows "5:1" and "square(x)".

Version control

Git and other version control systems can be used to identify bugs

- Which revision/commit resulted in the bug?
- Good for large projects, particularly when code is divided into multiple files