Debugging

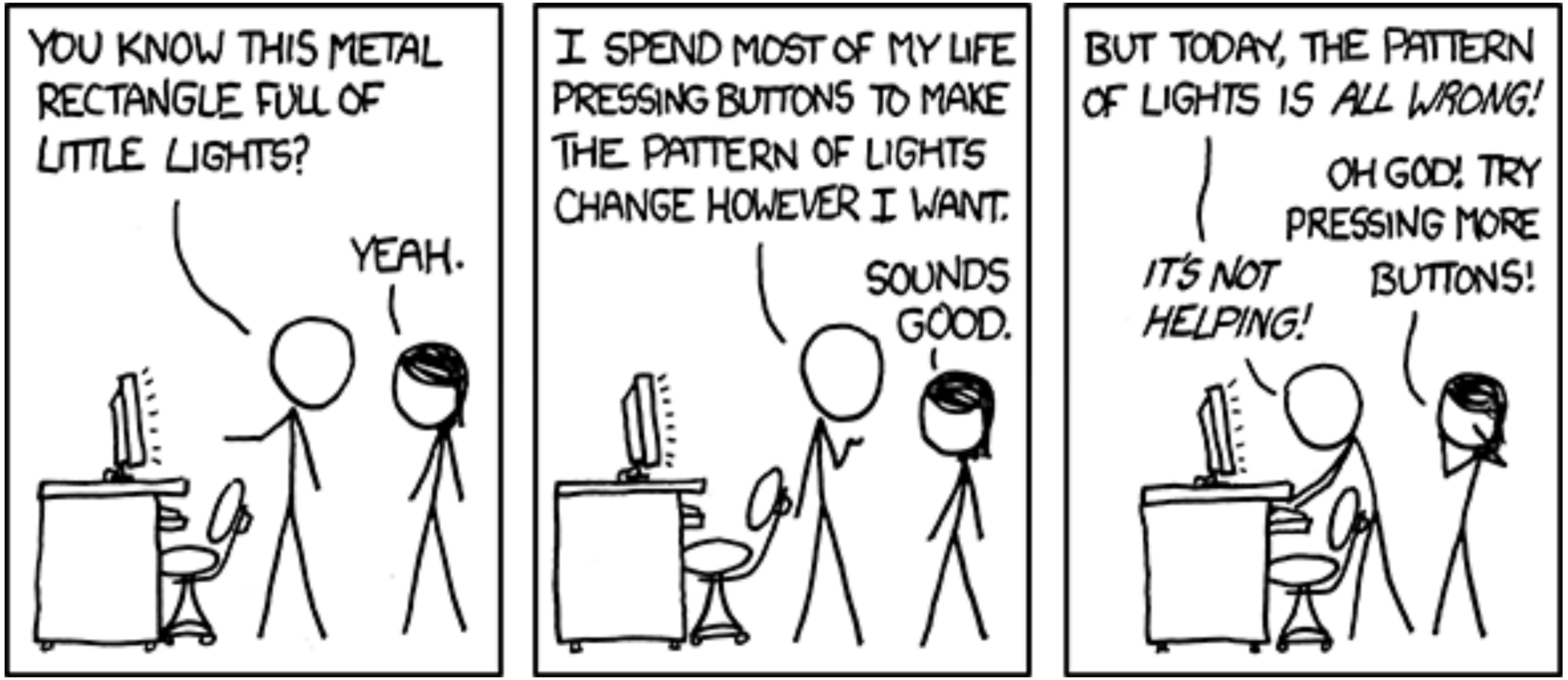
Jim Hester
RStudio

@jimhester

@jimhester_

https://rstats.wtf/debugging-r-code.html

Why Debug?


YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?
YEAH.
I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT.
SOUNDS GOOD.
BUT TODAY, THE PATTERN OF LIGHTS IS ALL WRONG!
OH GOD! TRY PRESSING MORE BUTTONS!
IT'S NOT HELPING!

Error in names[[i]] : subscript out of bounds

1. google *exact* error message
2. Keyword search community.rstudio.com
3. Stackoverflow - [r] tag

- traceback()
- print(), cat(), str()
- browser()
- debug() / trace() / recover()

traceback()

```r
f <- function(x) x + 1
g <- function(x) f(x)
g("a")
#> Error in x + 1: non-numeric argument to binary operator
traceback()
#> 2: f(x) at #1
#> 1: g("a")
```

traceback()

```
trceback()
#> Error in trceback(): could not find function "trceback"

# in .Rprofile
tb <- traceback

options(error = rlang::entrace)

rlang::last_error()
rlang::last_trace()
```

print()

```r
f <- function(x) {
  print(x)
  x + 1
}
g <- function(x) f(x)
g("a")
#> [1] "a"
#> Error in x + 1: non-numeric argument to binary operator
```

str()

```r
f <- function(x) {
  str(x)
  x + 1
}
g <- function(x) f(x)
g("a")
#>  chr "a"
#> Error in x + 1: non-numeric argument to binary operator
```

Demo

browser()

Examine objects

- ls() - list objects
- print() - print object
- str() - structure of object

browser()

Control execution

- n - next statement
- c - continue
- s - step into function call
- f - finish loop / function

browser()

Additional commands

- where - show previous calls
- Q - quit debugger

Demo

```
usethis::use_course("rstd.io/wtf-debugging")
```

Pick one to open and flesh out:

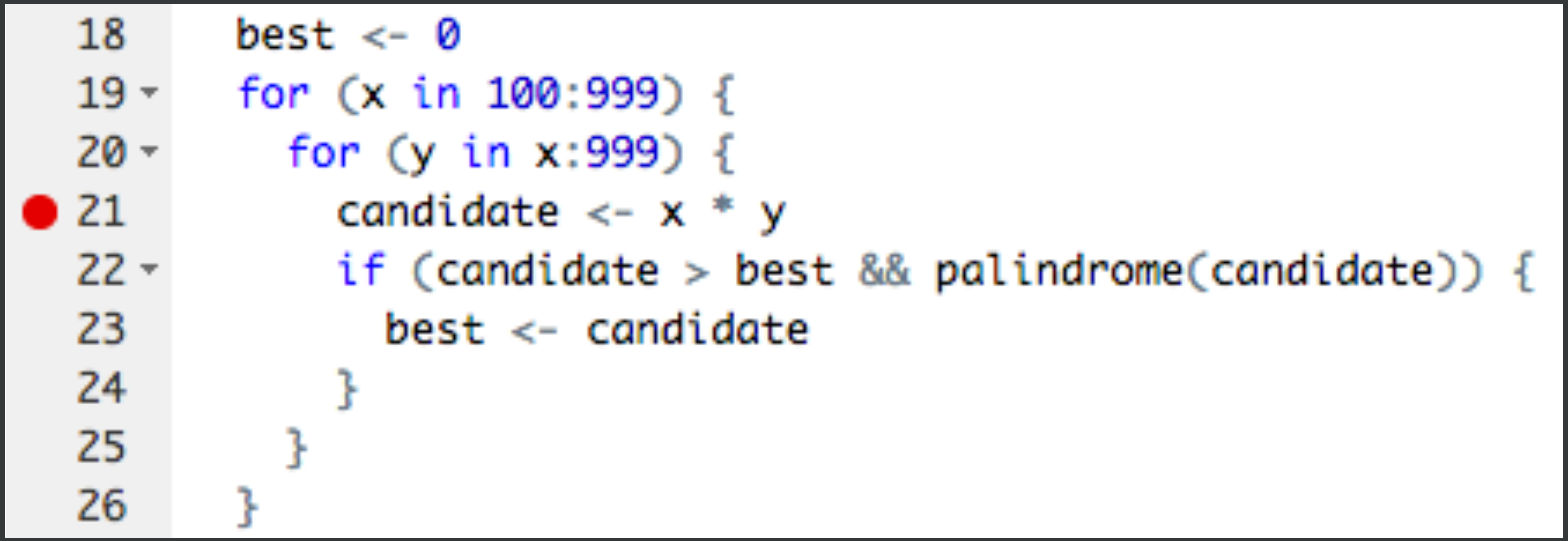
01_debugging_spartan.R

01_debugging_comfy.R*

* worst case, there's always jim

Debugging in RStudio

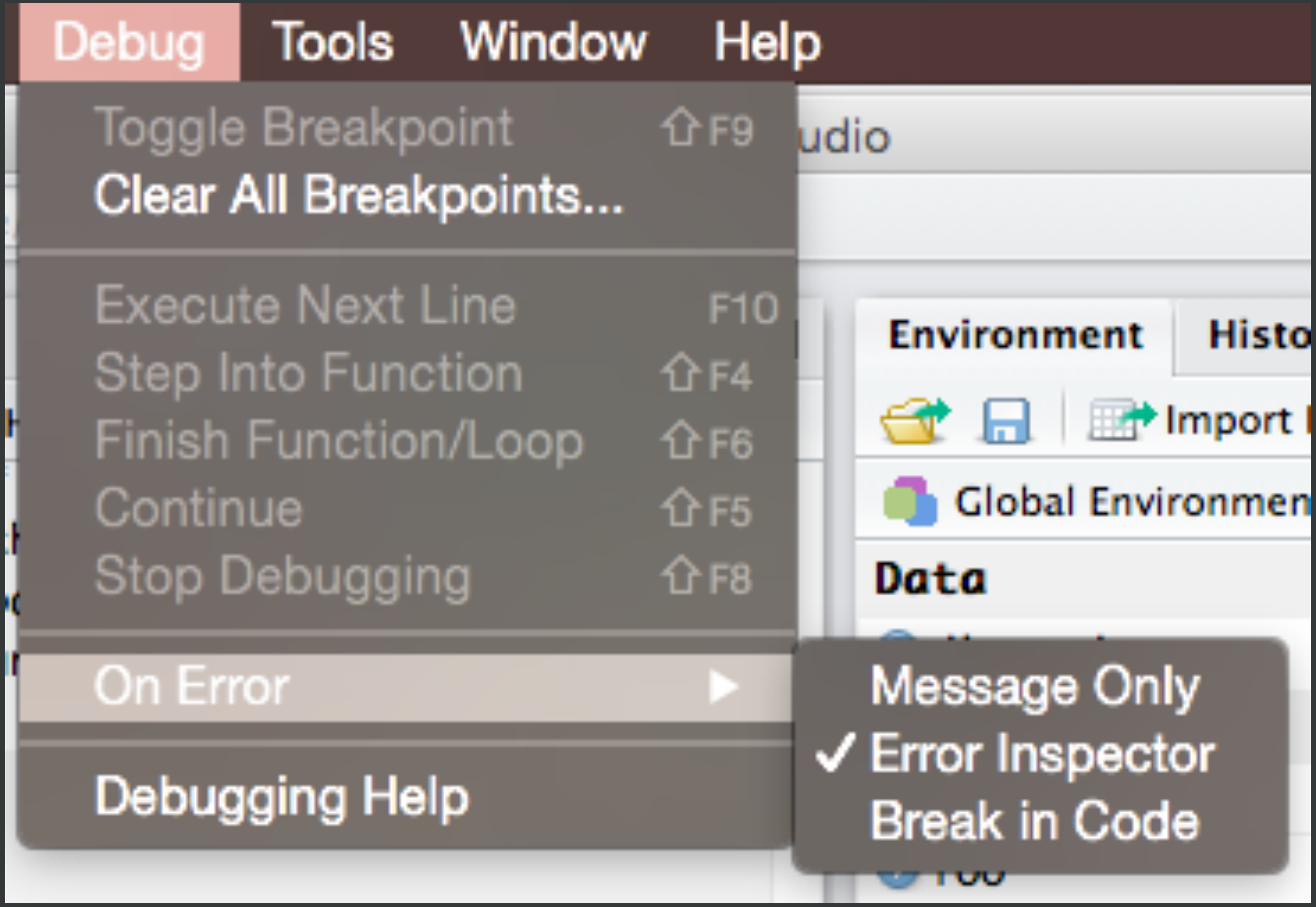
breakpoints



```
best <- 0
for (x in 100:999) {
  for (y in x:999) {
    candidate <- x * y
    if (candidate > best && palindrome(candidate)) {
      best <- candidate
    }
  }
}
```

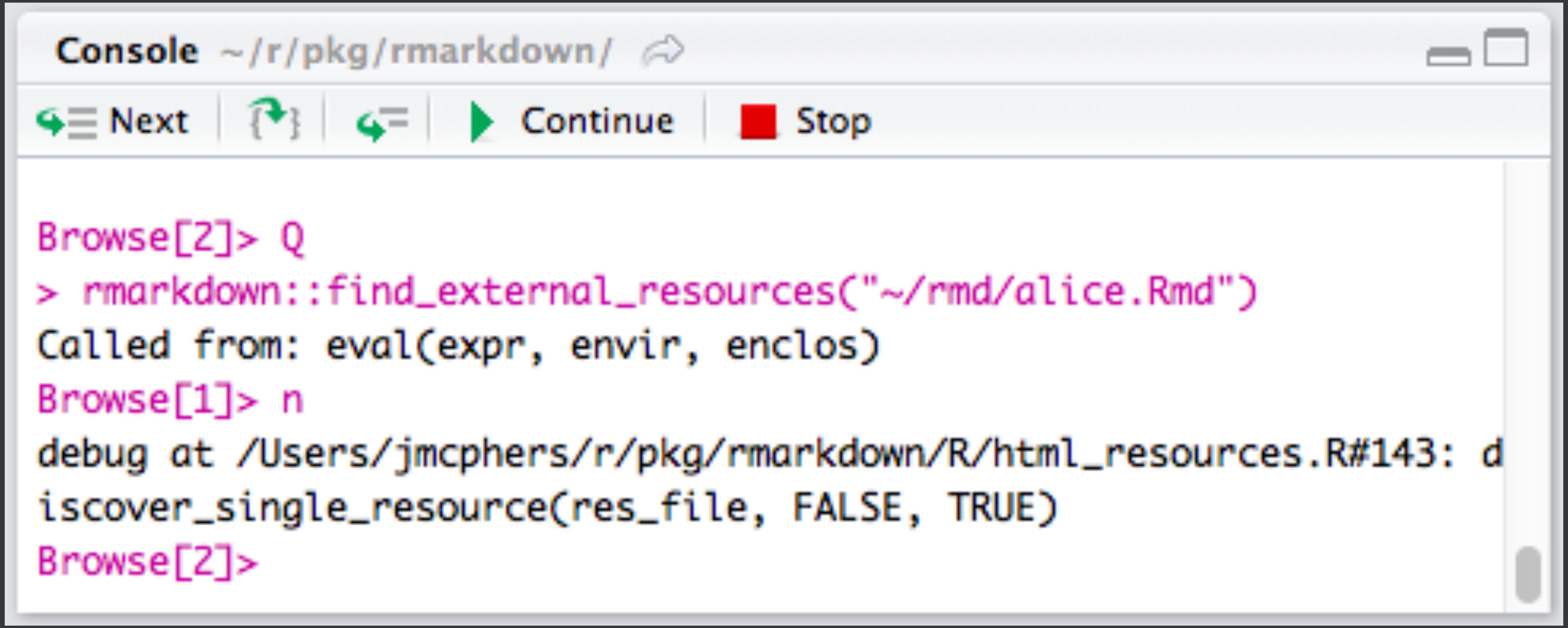
Debugging in RStudio

Debug on Error



Debugging in RStudio

Debug console



```
Browse[2]> Q
> rmarkdown::find_external_resources("~/rmd/alice.Rmd")
Called from: eval(expr, envir, enclos)
Browse[1]> n
debug at /Users/jmcphers/r/pkg/rmarkdown/R/html_resources.R#143: d
iscover_single_resource(res_file, FALSE, TRUE)
Browse[2]>
```

recover()

like browser()

full call stack

```
options(error = recover)
```

Demo

```
usethis::use_course("rstd.io/wtf-debugging")
```

Pick one to open and flesh out:

02_debugging_spartan.R

02_debugging_comfy.R*

* worst case, there's always jim

Debugging others' code

- Download / devtools::load_all() / source()
- recover()
- debug()
- trace()

debug() / debugonce()

```
debug(ggplot2::ggplot)

debug(ggplot2:::set_last_plot)

undebug(ggplot2::ggplot)
```

trace()

```r
trace(print)
print(1)
#> trace: print(1)
#> [1] 1

trace(print, browser)
#> Tracing function "print" in package "base"
```

trace()

```
trace(print, quote(if (is.numeric(x) && x >= 3) cat("hi\n")),
print = FALSE)
#> Tracing function "print" in package "base"
#> [1] "print"
print(1)
#> [1] 1
print(3)
#> hi
#> [1] 3
```

```
trace(print.data.frame, browser, at = 4)
#> Tracing function "print.data.frame" in package "base"
#> [1] "print.data.frame"
body(print.data.frame)
#> {
#>     n <- length(row.names(x))
#>     if (length(x) == 0L) {
#>         cat(sprintf(ngettext(n, "data frame with 0 columns
#> [sic]
#>         print(m, ..., quote = quote, right = right)
#>     }
#>     {
#>         .doTrace(browser(), "step 4")
#>         invisible(x)
#>     }
#> }
```

```
usethis::use_course("rstd.io/wtf-debugging")
```

Pick one to open and flesh out:

03_debugging_spartan.R

03_debugging_comfy.R*

* worst case, there's always jim