

# Debugging

Jim Hester  
RStudio

 @jimhester

 @jimhester\_

<https://whattheyforgot.org/debugging-r-code.html>

# Why Debug?



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

1. google *exact* error message
2. Keyword search [community.rstudio.com](https://community.rstudio.com)
3. Stackoverflow - [\[r\]](#) tag

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

# traceback()

```
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 = traceback)
```



# print()

```
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()

```
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("rstudio.io/wtf-debugging")
```

Pick one to open and flesh out:

01\_debugging\_spartan.R

01\_debugging\_comfy.R\*

\* worst case, there's always jim



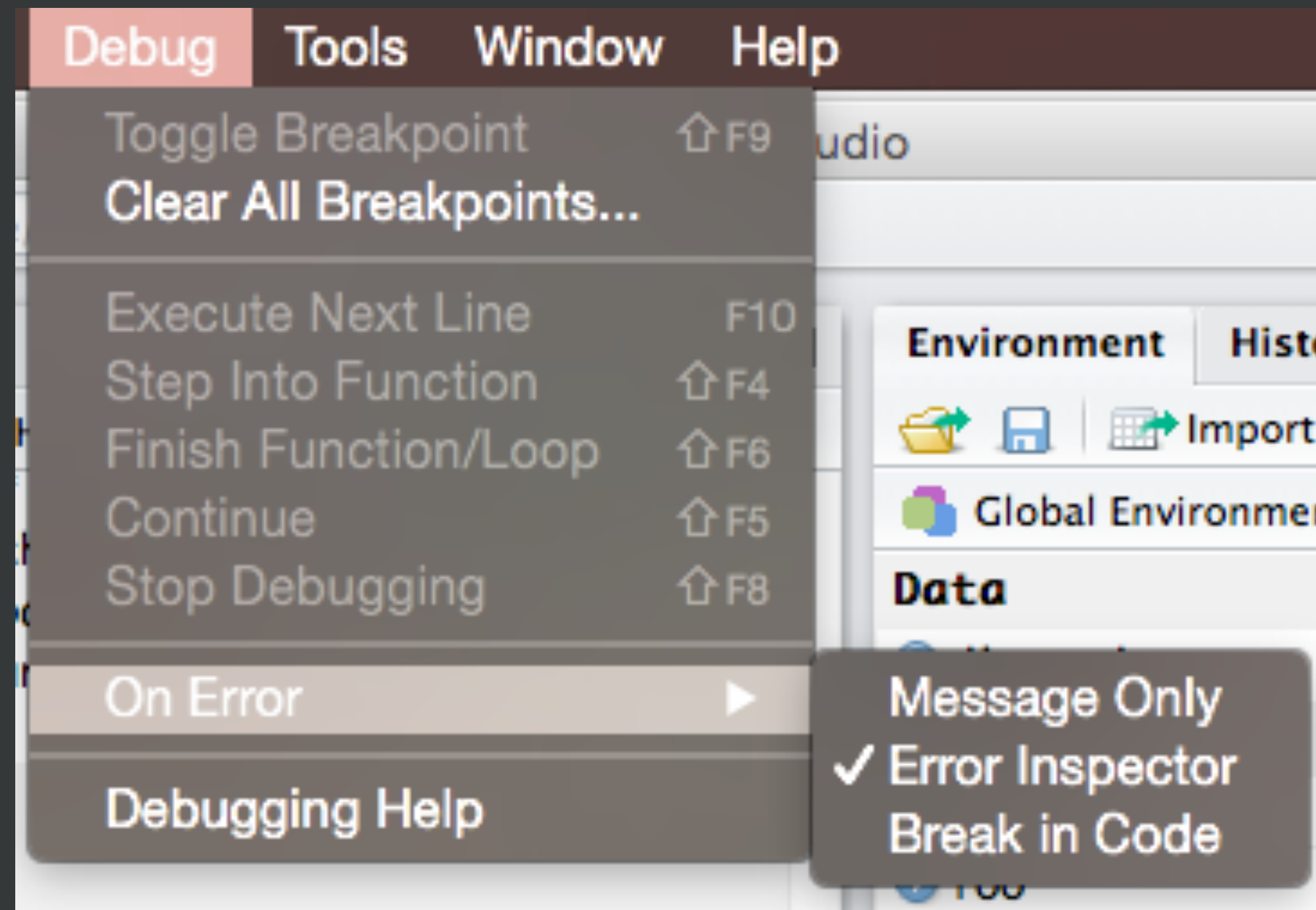
# Debugging in RStudio

## breakpoints

```
18 best <- 0
19 ▾ for (x in 100:999) {
20 ▾   for (y in x:999) {
● 21     candidate <- x * y
22 ▾     if (candidate > best && palindrome(candidate)) {
23       best <- candidate
24     }
25   }
26 }
```

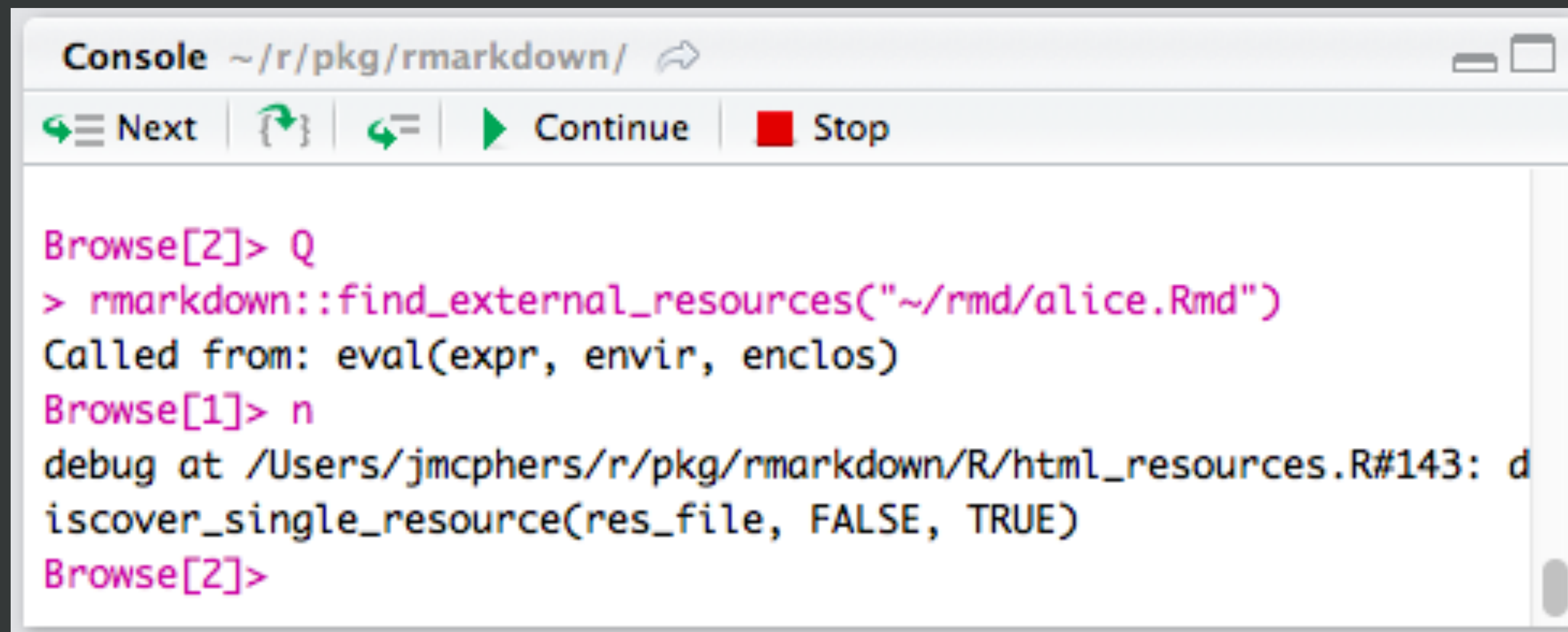
# Debugging in RStudio

## Debug on Error



# Debugging in RStudio

## Debug console

A screenshot of the RStudio debug console window. The window title is "Console ~/r/pkg/rmarkdown/". The toolbar contains icons for "Next", a refresh icon, a step back icon, "Continue", and a red square "Stop" button. The console text shows a sequence of commands and debug messages: "Browse[2]> Q", "> rmarkdown::find\_external\_resources(\"~/rmd/alice.Rmd\")", "Called from: eval(expr, envir, enclos)", "Browse[1]> n", "debug at /Users/jmcpheers/r/pkg/rmarkdown/R/html\_resources.R#143: d", "discover\_single\_resource(res\_file, FALSE, TRUE)", and "Browse[2]>".

```
Console ~/r/pkg/rmarkdown/

Next | { } | ⏪ | Continue | Stop

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

recover()

like browser()

full call stack

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

Demo

```
usethis::use_course("rstudio.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()

```
debug(ggplot2::ggplot)
```

```
debug(ggplot2:::set_last_plot)
```

```
undebug(ggplot2::ggplot)
```



# trace()

```
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("rstudio.io/wtf-debugging")
```

Pick one to open and flesh out:

03\_debugging\_spartan.R

03\_debugging\_comfy.R\*

\* worst case, there's always jim