



Debugging

Roger D. Peng, Associate Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Something's Wrong!

Indications that something's not right

- `message`: A generic notification/diagnostic message produced by the `message` function; execution of the function continues
- `warning`: An indication that something is wrong but not necessarily fatal; execution of the function continues; generated by the `warning` function
- `error`: An indication that a fatal problem has occurred; execution stops; produced by the `stop` function
- `condition`: A generic concept for indicating that something unexpected can occur; programmers can create their own conditions

Something's Wrong!

Warning

```
log(-1)
```

```
## Warning: NaNs produced
```

```
## [1] NaN
```

Something's Wrong

```
printmessage <- function(x) {  
  if(x > 0)  
    print("x is greater than zero")  
  else  
    print("x is less than or equal to zero")  
  invisible(x)  
}
```

Something's Wrong

```
printmessage <- function(x) {  
  if (x > 0)  
    print("x is greater than zero") else print("x is less than or equal to zero")  
  invisible(x)  
}  
printmessage(1)
```

```
## [1] "x is greater than zero"
```

```
printmessage(NA)
```

```
## Error: missing value where TRUE/FALSE needed
```

Something's Wrong!

```
printmessage2 <- function(x) {  
  if(is.na(x))  
    print("x is a missing value!")  
  else if(x > 0)  
    print("x is greater than zero")  
  else  
    print("x is less than or equal to zero")  
  invisible(x)  
}
```

Something's Wrong!

```
printmessage2 <- function(x) {  
  if (is.na(x))  
    print("x is a missing value!") else if (x > 0)  
    print("x is greater than zero") else print("x is less than or equal to zero")  
  invisible(x)  
}  
x <- log(-1)
```

```
## Warning: NaNs produced
```

```
printmessage2(x)
```

```
## [1] "x is a missing value!"
```

Something's Wrong!

How do you know that something is wrong with your function?

- What was your input? How did you call the function?
- What were you expecting? Output, messages, other results?
- What did you get?
- How does what you get differ from what you were expecting?
- Were your expectations correct in the first place?
- Can you reproduce the problem (exactly)?

Debugging Tools in R

The primary tools for debugging functions in R are

- `traceback`: prints out the function call stack after an error occurs; does nothing if there's no error
- `debug`: flags a function for “debug” mode which allows you to step through execution of a function one line at a time
- `browser`: suspends the execution of a function wherever it is called and puts the function in debug mode
- `trace`: allows you to insert debugging code into a function at specific places
- `recover`: allows you to modify the error behavior so that you can browse the function call stack

These are interactive tools specifically designed to allow you to pick through a function. There's also the more blunt technique of inserting `print/cat` statements in the function.