

Introdução a Linguagem R

Depurando o código

Delermundo Branquinho Filho

Algo está errado!

Indicações de que algo não está certo

- **message**: Uma mensagem genérica de notificação / diagnóstico produzida pela função **message**; A execução da função continua
- **warning**: Uma indicação de que algo está errado, mas não necessariamente fatal; A execução da função continua; Gerado pela função **warning**
- **error**: Uma indicação de que ocorreu um problema fatal; A execução pára; Produzido pela função **stop**
- **condition**: Um conceito genérico para indicar que algo inesperado pode ocorrer; Programadores podem criar sua própria condição

Warning

```
log(-1)
```

```
## Warning in log(-1): NaNs produzidos
```

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

Algo está errado!

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

Algo está errado!

```
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)
printmessage(NA)
```

Algo está errado!

```
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)
}
```

Algo está errado!

```
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 in log(-1): NaNs produzidos
```

```
printmessage2(x)
```

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

Algo está errado!

Como você sabe que algo está errado com sua função? - Qual foi sua opinião? Como você chamou a função? - O que você estava esperando? Saída, mensagens, outros resultados? - O que você conseguiu? - Como o que você obtém difere do que você esperava? - Suas expectativas estavam corretas em primeiro lugar? - Você consegue reproduzir o problema (exatamente)?

Ferramentas de depuração em R

As principais ferramentas para depuração funções em R são

- **traceback**: imprime a pilha de chamadas de função depois de ocorrer um erro; Não faz nada se não houver erro
- **debug**: sinaliza uma função para o modo "debug" "que permite passar pela execução de uma função uma linha de cada vez
- **browser**: suspende a execução de uma função onde quer que seja chamada e coloca a função no modo debug
- **trace**: permite que você insira o código de depuração em uma função em lugares específicos
- **recover**: permite que você modifique o comportamento de erro para que você possa navegar na pilha de chamadas de função

Estas são ferramentas interativas especificamente projetadas para permitir que você escolha através de uma função. Há também a técnica mais contundente de inserir instruções `print` / `cat` na função.

traceback

```
> mean(x)
Error in mean(x) : object 'x' not found
> traceback()
1: mean(x)
>
```

traceback

```
> lm(y ~ x)
Error in eval(expr, envir, enclos) : object 'y' not found
> traceback()
7: eval(expr, envir, enclos)
6: eval(predvars, data, env)
5: model.frame.default(formula = y ~ x, drop.unused.levels = TRUE)
4: model.frame(formula = y ~ x, drop.unused.levels = TRUE)
3: eval(expr, envir, enclos)
2: eval(mf, parent.frame())
1: lm(y ~ x)
```

debug

```
> debug(lm)
> lm(y ~ x)
debugging in: lm(y ~ x)
debug: {
  ret.x <- x
  ret.y <- y
  cl <- match.call()
  ...
  if (!qr)
```

```
      z$qqr <- NULL
    }
  }
Browse[2]>
```

debug

```
Browse[2]> n
debug: ret.x <- x
Browse[2]> n
debug: ret.y <- y
Browse[2]> n
debug: cl <- match.call()
Browse[2]> n
debug: mf <- match.call(expand.dots = FALSE)
Browse[2]> n
debug: m <- match(c("formula", "data", "subset", "weights", "na.action",
  "offset"), names(mf), 0L)
```

recover

```
> options(error = recover)
> read.csv("nosuchfile")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'nosuchfile': No such file or directory

Enter a frame number, or 0 to exit

1: read.csv("nosuchfile")
2: read.table(file = file, header = header, sep = sep, quote = quote, dec =
3: file(file, "rt")

Selection:
```

Depuração

Resumo

- Existem três indicações principais de um problema / condição: `message`, `warning`, `error`
- apenas um erro é fatal
- Ao analisar uma função com um problema, certifique-se de que você pode reproduzir o problema, claramente suas expectativas e como a saída difere de sua expectativa

- As ferramentas de depuração interativas `traceback`, `debug`, `browser`, `trace` e `recover` podem ser usadas para encontrar código problemático em funções
- Ferramentas de depuração não são um substituto para pensar!

The Scientist