

第一章 R语言介绍(二)

马学俊(主讲) 杜悦(助教)

苏州大学
数学科学学院

<https://xuejunma.github.io/>

内容

- 1 formatR演示
- 2 符号和名字
- 3 语法
 - 空格
 - 大括号
 - 行的长度
 - 缩进
 - 赋值
- 4 结构
- 5 作业

R编程风格

- 编程的风格人人不同，但分享代码时，需要一致的表达。高手交手，点到为止。《一代宗师》
- <https://yihui.name/formatr/>自动修改



```
if(TRUE){  
    x=1 # inline comments  
}else{  
    x=2;print('Oh no... ask the right bracket to go away!')}  
1*3 # one space before this comment will become two!  
2+2+2      # only 'single quotes' are allowed in comments  
  
lm(y~x1+x2, data=data.frame(y=rnorm(100),x1=rnorm(100)))  
1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1
```

安装

```
install.packages("formatR",  
                  repos = "http://cran.rstudio.com")  
library(formatR)  
#tidy_source(test1.r)  
formatR::tidy_app()
```

C:/Program Files/R/R-3.3.1/library/formatR/shiny - Shiny

http://127.0.0.1:4116 | Open in Browser

Publish

This Shiny app uses the function `tidy_source()` in the **formatR** (\geq v1.4) package to reformat R code in the text box on the right. [demo](#)

☒ Preserve comments

☒ Preserve blank lines

☐ Replace = with <-

☐ Put { on a new line

Number of spaces for indentation

4

Minimum line width

70

[Tidy My Code](#)

paste your R code here...

输入代码

Tidy R code using formatR
by Yihui Xie

server.R ui.R www

[show with app](#)

符号和名字

- 文件名：有一定意义。如果按照顺序执行，最好在命名前面加上数字前缀。

好

fit-models.R

utility-functions.R

不好

foo.r

stuff.r

0-download.R

1-parse.R

2-explore.R

- 对象名：变量和函数名称应该小写字母，使用_将名字单词分开。
- 变量名最好是名词，函数名最好是动词。
- 避免使用系统的函数或特殊意义的字母命名

好

day_one

day_1 # 不好

first_day_of_the_month

DayOne

dayone

djm1

T <- FALSE

c <- 10

mean <- function(x) sum(x)

语法之一空格

- 空格：是个好东西
- 在中缀操作符(+, -, *, /, < - 等)两边都留空格。在函数调用里使用=时，也使用同样的规则。
- 逗号后面留一个空格，而逗号前面则不要留。前留后不留。
- :, :: 两侧都不需要空格

好

```
average <- mean(feet / 12 + inches, na.rm = TRUE)
```

不好

```
average<-mean(feet/12+inches,na.rm=TRUE)
```

好

```
x <- 1:10 base::get
```

不好

```
x <- 1 : 10 base :: get
```

- 在左括号前面留一个空格(函数调用除外)
- 圆括号或者方括号内的代码两边留空格（除非有逗号）
- `<-` 前后根据需要可以任意添加空格

好

```
if (debug) do(x)
plot(x, y)
diamonds[5, ] #
```

不好

```
if ( debug ) do(x) # debug两边不要留空格
if(debug)do(x)
plot (x, y)
x[1,] # 逗号后面需要留一个空格
x[1 ,] # 空格要留在逗号后面，而不是前面
```

语法之二大括号

- 左花括号后面应该新起一行。
- 右花括号后面独占一行，除非它后面跟着的是else语句。
- 可以在一行写非常短的语句。

```
# 好
if (y < 0 && debug) {
    message("Y is negative")
}
if (y < 0 && debug) message("Y is negative")
# 不好
if (y < 0 && debug)
message("Y is negative")
```

语法之二大括号

```
# 好
if (y == 0) {
  log(x)
} else {
  y ^ x
}
```

```
# 不好
if (y == 0) {
  log(x)
}
else {
  y ^ x
}
```

行的长度

尽量使每行代码不超过80个字符。

缩进

- 缩进代码时，使用两个空格。不要使用制表符或者把制表符和空格混合使用。
- 唯一的例外是一个函数定义分成了多行的情况。在这种情况下，第二行缩进到函数定义开始的位置

```
lo_fun_n <- function(a = "a long argument",  
                      b = "another argument",  
                      c = "another long argument")
```

赋值

尽量使用`<-`，而不要使用`=`进行赋值。

好

```
x <- 5
```

#不好

```
x = 5
```

两者的区别< - =

多数情况下，两者可以互相替代，只有在函数参数列表中时不同.

```
plot(x <- 1:10)
```

```
plot(x = 1:10)
```


两者的区别 $< - =$

```
plot(x <- 1:10)
```

就是在作图之前把1:10赋给x，这个表达式在绘图之前就被执行了，相当于

```
x<-1:10  
plot(x)
```

```
plot(x = 1:10)
```

- 就不能达到对x赋值的目的，这时表达的含义是，为plot函数中名字为x的形式参数赋值1:10，作完图之后，x 这个对象仍然不存在。
- 只有在函数外部直接执行x=1:10才能达到对x赋值的目的。

```
> plot(x=1:10) #x是形式参数，不是新变量
```

```
> x ##错误：找不到这个对象"x"
```

#<-不会用于形式参数，所以这是为新变量x赋值的语句，不会引起歧义

```
> plot(x<-1:10)
```

```
> x
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

总之，`<-`在任何地方都可以完成赋值的工作，而`=`在函数参数列表则不行，所以，推荐任何时候都用`<-`进行赋值操作，而`=`仅用于形式参数？

结构

- 给你的代码添加注释。每一行注释都应该以一个注释符号和单个空格开头：`#`。注释应该解释代码的原理，而不是每行代码做了什么。
- 使用带有`-`和`=`的注释线把你的文件分隔成容易阅读的块。

```
# 加载数据 -----
```

```
# 绘图 -----
```

```
# myfunction 开始-----
```

```
# myfunction 结束-----
```

```
# 好  
day_one
```

```
# 好  
day_one
```

作业

- 分组讨论函数的使用
- 对于某一个函数举出实例说明

```
> #max 最大值  
> max(c(1, 2, 3))  
[1] 3
```

