北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R
语言数据分析

基础编程

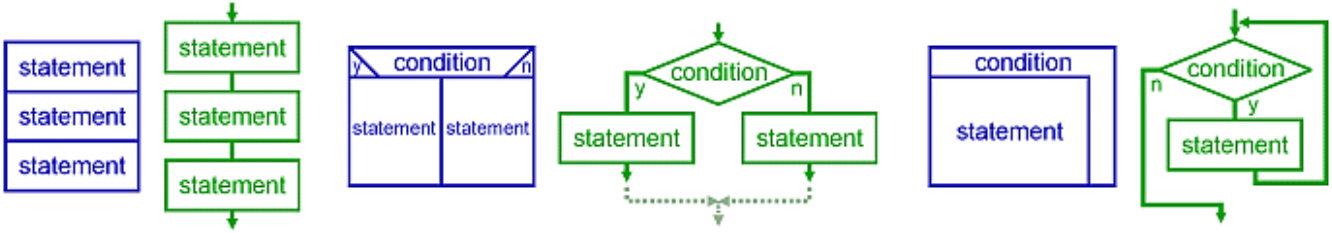
艾新波 / 2018 • 北京

课程体系

- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
  -
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

R编程 = 用别人的包和函数
讲述你自己的故事

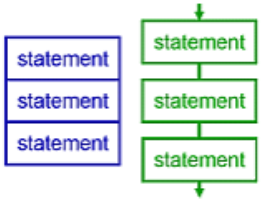
结构化编程



任何简单或复杂的逻辑都可以由顺序、分支和循环这三种基本结构组合而成

顺序结构

```
#创建3个向量
yw <- c(94, 87, 92, 91, 85, 92) #6个同学的语文成绩
sx <- c(82, 94, 79, 84, 92, 82) #数学成绩
wy <- c(96, 89, 86, 96, 82, 85) #外语成绩
ysw <- yw + sx + wy #向量化操作：三科成绩相加
ysw #显示三科成绩
#> [1] 274 272 259 273 261 261
(yw <- yw + 2) #向量化操作：每个同学语文成绩加2分
#> [1] 96 89 94 93 87 94
```



顺序结构

```
(mean_score <- mean(yw)) #求语文平均分
#> [1] 92.16667
sd(yw) #求语文成绩标准差
(sd_score <- (1 / (6-1) * sum((yw - mean_score)^2)) ^0.5)
#> [1] 3.430258
c(sd(yw), sd(sx), sd(wy))
#> [1] 3.430258 6.058052 5.865151
(z_score_yw <- (yw - mean_score) / sd_score) #求标准得分
#> [1]  1.12 -0.92  0.53  0.24 -1.51  0.53
```

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

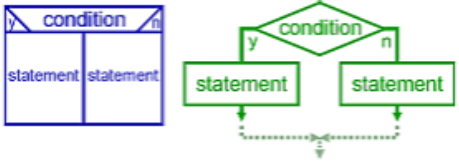
顺序结构

```
show(yw) #显示语文成绩
#> [1] 96 89 94 93 87 94
show(sx) #显示数学成绩
#> [1] 82 94 79 84 92 82
yw >= 90 #向量化操作：逻辑判断
#> [1]  TRUE FALSE  TRUE  TRUE FALSE  TRUE
yw >= 85 & sx >= 85 #向量化操作：逻辑判断
#> [1] FALSE  TRUE FALSE FALSE  TRUE FALSE
yw >= 95 | sx >= 95 #向量化操作：逻辑判断
#> [1]  TRUE FALSE FALSE FALSE FALSE FALSE
```

分支结构

条件表达式根据条件而执行不同的代码

```
if(条件) {
    cons.expr
} else {
    alt.expr
}
```



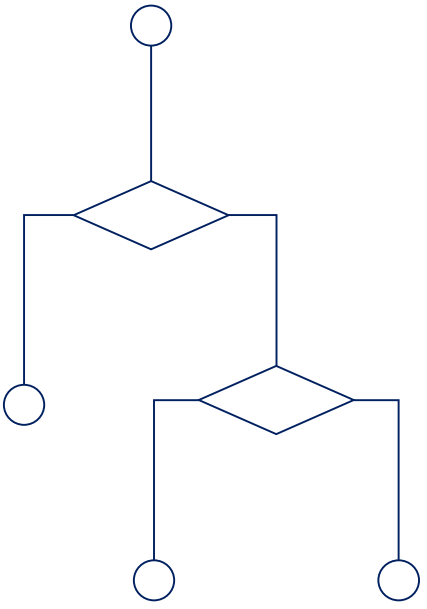
注意：

条件为一个标量的真或假值

else子句如果存在的话，必须和}在同一行

分支结构

```
min_score <- min(yw) #语文最低成绩87
if(min_score >= 90) {
  message(“语文成绩全部为优”) #控制台输出消息
} else if(min_score >= 80) {
  message("语文成绩至少为良")
} else {
  message("并非所有同学语文成绩均为优良")
}
#> 语文成绩至少为良
```



分支结构

```
yw >= 90
#> [1]  TRUE FALSE  TRUE  TRUE FALSE  TRUE
all(yw >= 90) #逻辑向量每一个值均为TRUE时，返回TRUE；否则返回FALSE
#> [1] FALSE
if(all(yw >= 90)) {
  message("语文成绩全部为优")
} else if(all(yw >= 80)) {
  message("语文成绩至少为良")
} else {
  message("并非所有同学语文成绩均为优良")
}
```

分支结构

```
show(yw)
#> [1] 96 89 94 93 87 94
any(yw <  88) #逻辑向量中只有有一个为TRUE，返回TRUE；否则返回FALSE
#> [1] TRUE
any(c(FALSE, FALSE, FALSE, FALSE, TRUE, FALSE))
if(any(yw < 60)) {
  message("有同学语文成绩挂科")
} else {
  message("所有同学语文考试顺利通过")
}
#> 所有同学语文考试顺利通过
```

分支结构

```
if(all(yw >= 90)) {
  message("语文成绩全部为优")
}
else if(all(yw >= 80)) { #else不能单起一行
  message("语文成绩至少为良")
}
else { #else不能单起一行
  message("并非所有同学语文成绩均为优良")
}
#> Error: unexpected 'else' in "else"
#> Error: unexpected '}' in "}"
```

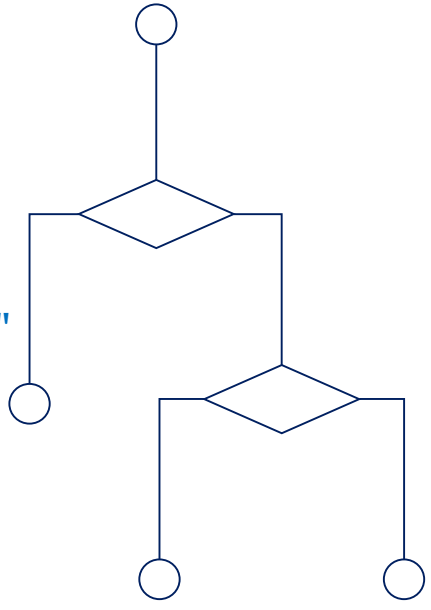
分支结构

```
yw > 90
#> [1]  TRUE FALSE  TRUE  TRUE FALSE  TRUE
if(yw > 90) {
  message("所有同学语文成绩均为优")
}
#> 所有同学语文成绩均为优
#> Warning message:
#>   In if (yw > 90) { :
#>       the condition has length > 1 and only the first
element will be used
```

相当于`if((yw > 90)[1]) {}`

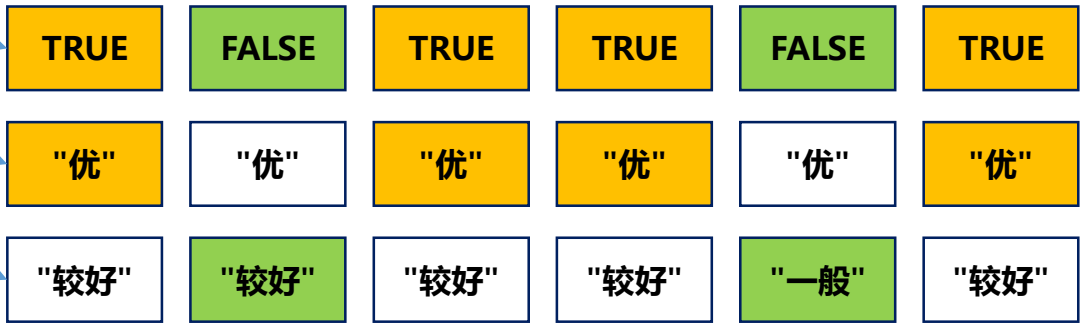
分支结构

```
yw
#> [1] 96 89 94 93 87 94
ifelse(yw >= 90, "优", "非优")
#> [1] "优"   "非优" "优"   "优"   "非优" "优"
ifelse(yw >= 90, "优",
       ifelse(yw >= 88, "较好", "一般"))
#> [1] "优" "较好" "优" "优" "一般" "优"
```



分支结构

```
yw #[1] 96 89 94 93 87 94
ifelse(yw >= 90,
       "优",
       ifelse(yw >= 88, “较好”, “一般"))
#> [1] "优 “ "较好 “ "优 “ "优 “ "一般 “ "优"
```



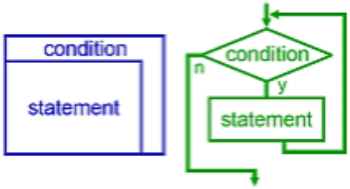
循环结构

三种语句实现循环

`for`

`while`

`repeat`

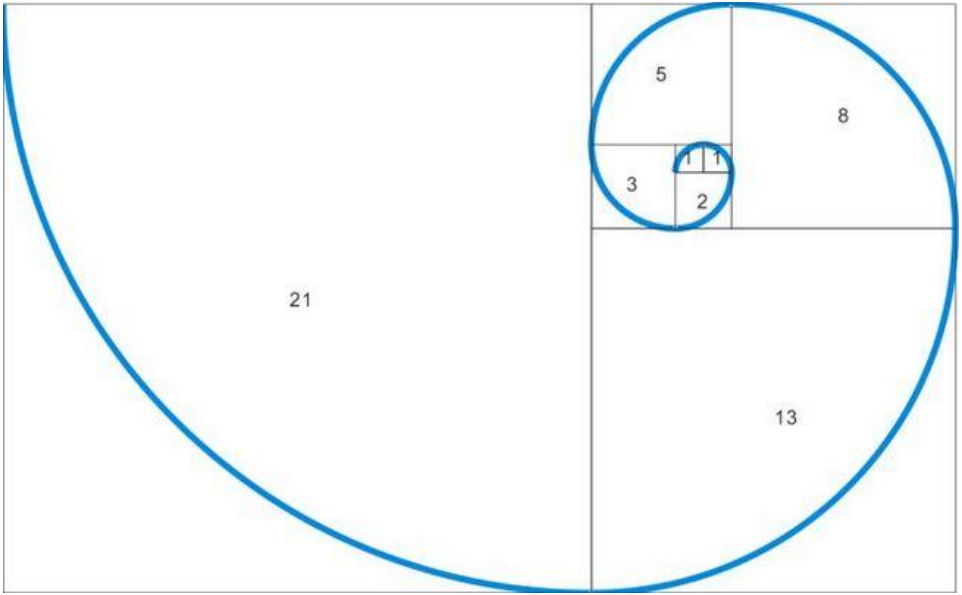


两种语句用于控制循环

`break` 语句可以从当前运行的循环里面跳出

`next` 语句会导致控制立即返回到循环的起点，`next` 后面的语句不会被执行

循环结构：斐波那契数列



循环结构

```
#for循环求斐波那契数列的前16个数
n_fib <- 16
fib <- numeric(n_fib)
fib[1:2] <- c(1, 1)
for(i in 3:n_fib) {
  fib[i] <- fib[i-1] + fib[i-2]
  show(fib[i])
}
fib
#> [1]   1   1   2   3   5   8  13  21  34  55  89 144 233 377 610 987
```

循环结构

```
#求1000以内的斐波那契数列
#不知道循环多少次，仅知道终止条件，通过while来实现
fib <- c(1, 1)
while(sum(tail(fib, 2)) < 1000) {
  fib <- c(fib, sum(tail(fib, 2)))
}
fib
#> [1]   1   1   2   3   5   8  13  21  34  55  89 144 233
377 610 987
```

循环结构

```
#也可以通过repeat来无条件循环，直到满足某个条件时break
fib <- c(1, 1)
repeat {
  if(sum(tail(fib, 2)) >= 1000) {
    break
  }
  fib <- c(fib, sum(tail(fib, 2)))
}
fib
#[1]   1   1   2   3   5   8  13  21  34  55  89 144 233 377
610 987
```

循环结构

```
#repeat来实现
fib <- c(1, 1)
while(TRUE) { #repeat == while(TRUE)
  if(sum(tail(fib, 2)) >= 1000) {
    break
  }
  fib <- c(fib, sum(tail(fib, 2)))
}
fib
#[1]   1   1   2   3   5   8  13  21  34  55  89 144 233 377
610 987
```

循环结构

```
#再尝试一个例子：幸运52
time_count <- 0
repeat {
  my_number <- sample(100, 1)
  time_count <- time_count + 1
  if(my_number == 52) {
    message("Haha~, I finally got '52' after ",
            time_count, " times")
    break
  } else {
    message(time_count,
          ": Not lucky enough [", my_number, "]")
  }
}
```

循环结构

```
#> 1: Not lucky enough [42]
#> 2: Not lucky enough [4]
#> 3: Not lucky enough [7]
……
#> 18: Not lucky enough [4]
#> 19: Not lucky enough [66]
#> 20: Not lucky enough [5]
#> 21: Not lucky enough [92]
#> Haha~, I finally got '52' after 22 attempts
```

再说向量化

```
#尽量不要使用显式循环
#能向量化运算的，尽量向量化
x <- 1:1e8 #一亿
y <- 2:(1e8 + 1) #一亿
z <- integer(1e8)
system.time(z <- x + y, gcFirst = TRUE)
#> user  system elapsed
#> 0.36    0.09    0.45
```

再说向量化

```
system.time({
  for(i in 1:1e8) {
    z[i] <- x[i] + y[i]
  }
},  gcFirst = TRUE)
#> user  system elapsed
#> 11.51    0.06   11.70
```

尽量避免显式循环

在R里边

只要串行模式不是必须的

（本轮循环语句不受上一轮影响）

请采用并行模式

斐波那契数列的另一种实现

#其实，连斐波那契数列也可以采用并行的方式

```
n_fib <- 16
sapply(1:n_fib, function(x) {
  1 / sqrt(5) *
    (((1 + sqrt(5)) / 2) ^ x -
                ((1 - sqrt(5)) / 2) ^ x)
})
#> [1]   1   1   2   3   5   8  13  21  34  55  89 144 233
377 610 987
```

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n}-\left(\frac{1-\sqrt{5}}{2}\right)^{n}\right]$$

R里边的并行作业

- 向量化函数

  `sqrt()`, `sin()`, `round()`, `as.numeric()`, `cut()`

  `weekdays()`, `+`, `-`, `*`, `/`, `&`, `|`, `%in%`, ……

- split-apply-combine数据处理模式

  如：apply函数簇

  再比如：tidyverse扩展包套装dplyr::group_by+summarise

- 专门的并行主题

  TASK VIEWS:High-Performance and Parallel Computing with R

  parallel::mclapply(), foreach::foreach(), …

谢谢聆听

Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ： 23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics