量化投资 Using R

第三章 R 语言基本数据类型

罗智超 (ROKIA.ORG)

Contents

通过本章你将学会

- R 的基本数据类型（向量、矩阵、数组、数据框、列表、因子）只需掌握**向量**、数据框、因子
- R 的基本数据类型的创建
- R 的基本数据类型之间的转换

R 的基本数据类型

Dim	Homogeneous	Heterogeneous
1d	Atomic vector	List
2d	Matrix	Data frame
nd	Array	

- R 的数据类型的多样性是把双刃剑，由于多样所以灵活，由于灵活，所以掌握难度较大
- 掌握好向量的基本功是掌握其他数据类型的基础，数据框 (dataframe) 是最常用的一种类型

向量 vector

- 用于存储数值、字符或者逻辑数据的一维**数组**
- 向量的创建和索引是非常重要的基本功
- 正是 R 的向量运算功能使其效率极高

```
x<-1:10
x<-c(1,2,9999999,2,22:50,NA,rnorm(100,4,2))
mean(x,na.rm=T)
mean(x,na.rm=T,trim = 0.01)
median(x,na.rm = T)
hist(x)
boxplot(x)
```

```
# a numeric vector
a <- c(1, 2, 5, 3, 6, -2, 4)
a.i<-c(1L,2L)
class(a)
class(a.i)



# a character vector
b <- c("one", "two", "three")
# a logic vector
c <- c(TRUE, TRUE, TRUE, FALSE, TRUE, FALSE)
```

使用 seq()、rep() 创建向量

```
1 3 5 7 9 11
c(1 ,3, 5, 7, 9, 11)
1 1000000
```

```
seq(from=2,t0=100,by=2)


# 重复常数

rep(4,5)

1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

rep(1:5,1:5)
rep(c(5,12,13),each=2)

paste("x",1:5,sep="")
```

增加或删除向量元素

```
x <- 1:10
length(x)
y=x 的奇数项重复2次，偶数项重复3次

x1<-rep(seq(1,length(x),2),each=2)
x2<-rep(seq(2,length(x),2),each=3)
x3<-c(x1,x2)
x[x3]




# 练习：

# 向量一阶差分

x <- c(88,5,12,13,20,11)
c(NA,x[-1]-x[-6])
y1<-x[-1]-x[-6]
```

获得向量长度

```
length(x)

# 遍历向量
#seq_along(x) = 1:length(x)

x<-1:10
n<-length(x)
for (i in 1:n){
    print(x[i] )
}
```

向量索引

```
y <- c(1.2,3.9,0.4,0.12)
y[c(1,3)]

y[2:3]
v <- 3:4
y[v]
# 向量索引的原理
# 允许重复向量位置
x <- c(4,2,17,5)
y <- x[c(1,1,3,3)]

x<-1:10
set.seed(123)
x<-sample(1:100,10)
# 提取 x>30 的数据
x[x>30]
x>30

y<-which(x>30)
x[y]

x[3:4]
x[c(3,5,7)]

x<-1:10
y<-NULL
for(i in 1:10){
  if(x[i]>5){y<-c(y,i)}
}
x[y]

x[x>5]









x[x>8]
x[which(x>8)]
#
x[order(x)]
x[c(FALSE,TRUE)]
y<-setNames(x,letters[1:10])
y[c("a","b")]


y[-1]-y[-length(y)]

set.seed(123)
```

```r
x<-sample(1:100,20,replace = T)
#question:
#(1) 提取 x 中符合>=21 且<=55 条件的数
#(2) 列出大于 50 的数据的位置
#(3) 将 (2) 中计算的结果提取出来
x[x>=21 & x<=55]
which(x>50)
x[which(x>50)]




x[x>=21 & x<=55]
which(x>50)
x[which(x>50)]
```

all() 及 any() 的使用

```r
x <- 1:10
any(x > 8)

all(x > 8)

all(x > 0)
```

扩展案例

```r
#Suppose that we are interested in finding runs of consecutive 1s
#in vectorsthat consist just of 1s and 0s.
findruns1 <- function(x,k) {
   n <- length(x) #n=9,k=2
   runs <- vector(length=n)
   count <- 0
   for (i in 1:(n-k+1)) {
      if (all(x[i:(i+k-1)]==1)) {
         count <- count + 1
         runs[count] <- i
      }
   }
   if (count > 0) {
      runs <- runs[1:count]
   } else runs <- NULL
   return(runs)
}
y<- c(1,0,0,1,1,1,1,1,1,1,0,1,1)

findruns1(y,2)
```

向量运算

```
u<-c(5,2,8)
v<-c(5,3,9)
u==v


z <- c(5,2,-3,8)

w <- z[z*z > 8]

x <- c(1,3,8,2,20)
x[x^2>8]<-0

options(digits = 2)
x<-c(2,3,2,5,NA,7,9)
x[is.na(x)]<-mean(x,na.rm = T)

# 把 NA 替换成 x 的均值
mean(x,na.rm = T)
x[x==NA]<-mean(x,na.rm=T)#error
x[is.na(x)]<-mean(x,na.rm=T)

# 赋值
x[x > 3] <- 0

set.seed(123)
x<-sample(c(1:100,rep(NA,20)),50,replace = T)
# 将 NA 替换为 x 的中位数, 计算替换前后的 x 的均值
mean(x,na.rm=T)
x[is.na(x)]<-median(x,na.rm = T)
mean(x,na.rm = T)

x[is.na(x)]<-median(x,na.rm = T)
# 将 NA 替换为 x 的均值
x[is.na(x)] <- mean(x,na.rm = T)
```

向量过滤 (取子集)

```
#subset(dataset,subset,select=c())
x <- c(6,1:3,NA,12)
x[x > 5]
y<-subset(x,x > 5)
```

向量位置选择

```
z <- c(5,2,-3,8)
which(z*z > 8)
```

```
# 寻找向量中第一个等于 1 的位置
x<-c(3,2,6,1,7,1,1)
```

```r
which(x==1)[1]
x<-sample(1:100,50,replace = T)
quantile(x)
# 计算每 10% 的分位数
quantile(x,seq(0,1,0.1))
quantile(x,seq(0.9,1,0.01))
# 向量运算
which(x==1)[1]

# 向量计算中短的向量会自动循环补充


c(1,2)+1:10



first1 <- function(x) {
    for (i in 1:length(x))
        {if (x[i] == 1) break # break out of loop
         }
    return(i)
    }
# 另外一种方法
first1a <- function(x) return(which(x == 1)[1])
```

```r
x<-sample(1:100,50,replace = T)
# 请将奇数赋值为 1, 偶数赋值为 0
ifelse(x%%2==0,0,1)

# 请将 x<30 的赋值为 1, 30<=x<50 赋值为 2,x>=50 赋值为 3


ifelse(x<30,1,ifelse(x<50,2,3))



y<-vector(length=length(x))
for (i in 1:length(x)){
  if (x[i]%%2==0){y[i]<-0}
else {y[i]<-1}
}
```

使用 ifelse() 函数

```r
x<-sample(1:100,50,replace = T)

y <- ifelse(x %% 2 == 0,0,1) # %% is the mod operator
```

向量元素比较

```
# Compare whether two datasets are same and
# which array indics is different.
a1<-c(1,3,4,5,6)
a2<-c(1,3,7,8,7)
which(a1!=a2,arr.ind = TRUE)

#identical 比较的是完全一样
identical(x,y)
# : 产生的是整数, c() 产生的是浮点数
x<-1:2
y<-c(1,2)
identical(x,y)

#match(a,b) 类似于 excel 里面的 vlookup, 在 b 中查找是否存在 a 的元素
```

向量元素命名

```
x <- c(1,2,4)
names(x) <- c("a","b","ab")
x["b"]

letters[3:8]
LETTERS[5:6]


a1,b2,c3,d4,...z26
paste0(letters,1:26)
paste(letters,1:26,sep = "")
```

扩展案例

```
#Kendall's 相关计算
# 方法一
x <- sample(1:50,10,replace = T)
y <- sample(1:50,10,replace = T)
x1<-x[-1]-x[-length(x)]# diff(x)
x2<-ifelse(x1>=0,1,0)#sign(x)
y1<-y[-1]-y[-length(y)]
y2 <- ifelse(y1>=0,1,0)
c <- ifelse(x2==y2,1,0)
mean(c)

# 方法二
findud<-function(v){
  vud<-v[-1]-v[-length(v)]
  return(ifelse(vud>0,1,-1))
}
udcorr<-function(x,y)
```

```
  ud<-lapply(list(x,y),findud)
return(mean(ud[[1]]==ud[[2]]))

# 方法三
 mean(sign(diff(x))==sign(diff(y)))
```

扩展案例

- 对鲍鱼数据重新编码

```
#ifelse 可以嵌套使用
#for() 循环可以对字符串向量进行循环，甚至文件名
g<-c("M","F","F","I","M","M","F")
ifelse(g=="M",1,ifelse(g=="F",2,3))
m<-which(g=="M")
f<-which(g=="M")
i<-which(g=="M")
grps<-list()
for(gen in c("M","F","I")) grps[[gen]]<-which(g==gen)
```

数据框 dataframe

- 数据框是最常用的数据类型，类似于 SAS 里面的 dataset
- 数据框是特殊的 List，是包含向量的 list
- 不同的列可以包含不同的模式（数值、字符、逻辑、因子）
- 由 data.frame(col1,col2,col3,...) 创建

```
patientID <- c(1, 2, 3, 4)
age <- c(25, 34, 28, 52)
diabetes <- c("Type1", "Type2", "Type1", "Type1")
status <- c("Poor", "Improved", "Excellent", "Poor")
patientdata <- data.frame(patientID, age, diabetes,                status,stringsAsFactors=FALSE)
patientdata
```

```
df<-as.data.frame(replicate(10,sample(1:100,50,replace = T)))
sample(1:100,50,replace = T)
names(df)<-paste0("x",1:10)

# 问题：请给 df 的 10 个变量，分别命名为 x1-x10

# 提取 df
# 行 1,3,5,7,9
# 列 2,4,6,8,10
df[seq(1,10,2),seq(2,10,2)]

# 提取 x3>=40 或者 x4 <=60 的数据，只显示 x3,x4 的数据

df2<-df[df$x3>=40 |df$x4<=60 ,c("x3","x4")]

df3<-subset(df,x3>=40 | x4<=60,c("x3","x4"))
```

```r
dim(df3)


# 请问，以上提取的数据共有多少行
nrow(df2)
ncol(df2)
# 请问，x5 这列中有多少唯一的数字
length(unique(df$x5))
str(df)




length(unique(df$x5))
# 请问，y=x7+x8
df$y<-df$x7+df$x8
```

数据框元素提取

- 从数据框中提取列有两种方法；

```r
df <- data.frame(x = 1:3, y = 3:1, z = letters[1:3],stringsAsFactors = F)
str(df)

#-- 像 list
df[c("x","y")]
#-- 像 matrix
df[,c("x","y")]
# 如果你选择单个列，那么有重要的区别：
# 默认情况下，矩阵的 ** 取子集操作 ** 会对结果进行简化
# 而列表方式却不会。
# 比较下面语句
str(df["x"])
str(df[,"x"])
str(df[,"x",drop=F])
str(df[,c("x","y")])
```

```r
df[2:5,]
df[2:5,2]
df[2:5,2,drop=FALSE]
df[df$x >= 1,]
subset(df,x1 >= 1)
#Homework "Data Manipulation with R"
#C01 & C06
```

因子 factor

```r
x<-factor("a","b","a","a","b")
class(x)
levels(x)

#
```

```r
z <- read.csv(text = "value\n12\n1\n.\n9")
str(z)
z <- read.csv(text = "value\n12\n1\n.\n9",na.strings = ".",stringsAsFactors = F)
str(z)
```

数据类型属性

- name()
- dimension()
- class()