Aula 1.4 - subconjuntos

Elder Sodre

Monday, February 23, 2015

### Subconjuntos (subsetting)

#### Colchetes simples []

Colchetes simples podem ser usados em vetores, matrizes e dataframes.

vet <- seq(5,9,0.5)  
vet

## [1] 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0

vet[1] #Separei o primeiro elemento do vetor vet

## [1] 5

vet[6] #Sexto elemento do vetor vet

## [1] 7.5

Colchetes também podem ser usados para separar mais de um elemento de um vetor:

vet[3:7] #Elementos 3 a 7

## [1] 6.0 6.5 7.0 7.5 8.0

vet[c(5,2,8)] #Elementos 5, 2 e 8 (nesta ordem)

## [1] 7.0 5.5 8.5

Operadores lógicos também podem ser utilizados com colchetes:

vet[vet<=7] #Todos os elementos menores ou iguais a 7

## [1] 5.0 5.5 6.0 6.5 7.0

vet2 <- rnorm(20)  
vet2[vet2>0] #Todos os elementos maiores que 0.

## [1] 1.2013 0.7416 1.4203 2.2787 0.6452 2.1706 1.7817 0.2254 0.1119 1.1893  
## [11] 0.4191

Por fim, vetores de lógica podem ser usados também:

vet3 <- c(3,7,4,7,9)  
log <- c(TRUE, TRUE, FALSE, TRUE, FALSE)  
vet3[log]

## [1] 3 7 7

Com objetos bidimensionais, usa-se colchetes da seguinte forma: mat[i,j]

mat <- cbind(1:10,rnorm(10),rep(c(0,1)))  
mat

## [,1] [,2] [,3]  
## [1,] 1 -0.01932 0  
## [2,] 2 -0.70194 1  
## [3,] 3 0.27049 0  
## [4,] 4 1.20104 1  
## [5,] 5 0.11935 0  
## [6,] 6 -1.83489 1  
## [7,] 7 -0.82753 0  
## [8,] 8 0.45693 1  
## [9,] 9 0.05137 0  
## [10,] 10 -0.81434 1

mat[4,2] #Quarta linha, segunda coluna

## [1] 1.201

mat[7,3] #Sétima linha, terceira coluna

## [1] 0

mat[5:7,c(1,3)] #Linhas 5, 6 e 7, primeira e terceira coluna.

## [,1] [,2]  
## [1,] 5 0  
## [2,] 6 1  
## [3,] 7 0

Um espaço vazio indica que se quer todas as linhas ou colunas:

mat[,2] #Todas as linhas da segunda coluna

## [1] -0.01932 -0.70194 0.27049 1.20104 0.11935 -1.83489 -0.82753  
## [8] 0.45693 0.05137 -0.81434

mat[8:10,] #Todas as colunas das linhas 8,9 e 10

## [,1] [,2] [,3]  
## [1,] 8 0.45693 1  
## [2,] 9 0.05137 0  
## [3,] 10 -0.81434 1

mat[mat[,3]==1,]

## [,1] [,2] [,3]  
## [1,] 2 -0.7019 1  
## [2,] 4 1.2010 1  
## [3,] 6 -1.8349 1  
## [4,] 8 0.4569 1  
## [5,] 10 -0.8143 1

#### Subconjuntos de listas

Pode-se "subsetar" listas usando [], [[]] ou $.

lista <- list(A="Bom dia!",B=1:12,C=c(23,77,34,90))  
lista

## $A  
## [1] "Bom dia!"  
##   
## $B  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12  
##   
## $C  
## [1] 23 77 34 90

lista[2] #Colchetes simples sempre retornam um objeto da mesma classe do objeto original. Como se trata de uma lista, [] retornará uma lista.

## $B  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12

lista[[2]] #Retornou um vetor

## [1] 1 2 3 4 5 6 7 8 9 10 11 12

Os nomes dos elementos podem ser usados para criar subconjuntos:

lista$C

## [1] 23 77 34 90

lista[["C"]]

## [1] 23 77 34 90

lista["C"]

## $C  
## [1] 23 77 34 90

Usando o nome dos elementos, $ e [[]] são similares. Porém, colchetes duplos permitem o nome computado, isto é:

nome <- "A"  
lista$nome #Não existe objeto nome nesta lista

## NULL

lista[[nome]]

## [1] "Bom dia!"

Para extrair mais de um objeto de uma lista, é necessário usar colchetes simples:

lista[c(1,3)] #Elementos 1 e 3 da lista.

## $A  
## [1] "Bom dia!"  
##   
## $C  
## [1] 23 77 34 90

#lista[[c(1,3)]] #Erro

O uso de uma sequência de números dentro de colchetes duplos tem outro significado:

lista [[c(2,7)]] #Lembram que o segundo objeto da lista é um vetor?

## [1] 7

#O código acima retira o sétimo elemento do segundo elemento da lista.  
#É a mesma coisa que escrever:  
  
lista[[2]][7]

## [1] 7

#ou:  
lista$B[7]

## [1] 7

IMPORTANTE! Lembram que um data frame é um tipo especial de lista?

Todas as formas de tirar subconjuntos em listas também funcionam em data frames.

O R, além de funções, vem com vários conjuntos de dados que podem ser usados para explorar as funcionalidades do R.

mtcars é um data frame contendo consumo de combustível e outras informações técnicas de 32 carros dos anos 1973 e 1974.

?mtcars

## starting httpd help server ... done

attributes(mtcars)

## $names  
## [1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear"  
## [11] "carb"  
##   
## $row.names  
## [1] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710"   
## [4] "Hornet 4 Drive" "Hornet Sportabout" "Valiant"   
## [7] "Duster 360" "Merc 240D" "Merc 230"   
## [10] "Merc 280" "Merc 280C" "Merc 450SE"   
## [13] "Merc 450SL" "Merc 450SLC" "Cadillac Fleetwood"   
## [16] "Lincoln Continental" "Chrysler Imperial" "Fiat 128"   
## [19] "Honda Civic" "Toyota Corolla" "Toyota Corona"   
## [22] "Dodge Challenger" "AMC Javelin" "Camaro Z28"   
## [25] "Pontiac Firebird" "Fiat X1-9" "Porsche 914-2"   
## [28] "Lotus Europa" "Ford Pantera L" "Ferrari Dino"   
## [31] "Maserati Bora" "Volvo 142E"   
##   
## $class  
## [1] "data.frame"

Usar [], [[]] e $ retornará as colunas correspondentes do data frame:

mtcars[6]

## wt  
## Mazda RX4 2.620  
## Mazda RX4 Wag 2.875  
## Datsun 710 2.320  
## Hornet 4 Drive 3.215  
## Hornet Sportabout 3.440  
## Valiant 3.460  
## Duster 360 3.570  
## Merc 240D 3.190  
## Merc 230 3.150  
## Merc 280 3.440  
## Merc 280C 3.440  
## Merc 450SE 4.070  
## Merc 450SL 3.730  
## Merc 450SLC 3.780  
## Cadillac Fleetwood 5.250  
## Lincoln Continental 5.424  
## Chrysler Imperial 5.345  
## Fiat 128 2.200  
## Honda Civic 1.615  
## Toyota Corolla 1.835  
## Toyota Corona 2.465  
## Dodge Challenger 3.520  
## AMC Javelin 3.435  
## Camaro Z28 3.840  
## Pontiac Firebird 3.845  
## Fiat X1-9 1.935  
## Porsche 914-2 2.140  
## Lotus Europa 1.513  
## Ford Pantera L 3.170  
## Ferrari Dino 2.770  
## Maserati Bora 3.570  
## Volvo 142E 2.780

mtcars[[6]]

## [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440  
## [12] 4.070 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520  
## [23] 3.435 3.840 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780

mtcars[1:4]

## mpg cyl disp hp  
## Mazda RX4 21.0 6 160.0 110  
## Mazda RX4 Wag 21.0 6 160.0 110  
## Datsun 710 22.8 4 108.0 93  
## Hornet 4 Drive 21.4 6 258.0 110  
## Hornet Sportabout 18.7 8 360.0 175  
## Valiant 18.1 6 225.0 105  
## Duster 360 14.3 8 360.0 245  
## Merc 240D 24.4 4 146.7 62  
## Merc 230 22.8 4 140.8 95  
## Merc 280 19.2 6 167.6 123  
## Merc 280C 17.8 6 167.6 123  
## Merc 450SE 16.4 8 275.8 180  
## Merc 450SL 17.3 8 275.8 180  
## Merc 450SLC 15.2 8 275.8 180  
## Cadillac Fleetwood 10.4 8 472.0 205  
## Lincoln Continental 10.4 8 460.0 215  
## Chrysler Imperial 14.7 8 440.0 230  
## Fiat 128 32.4 4 78.7 66  
## Honda Civic 30.4 4 75.7 52  
## Toyota Corolla 33.9 4 71.1 65  
## Toyota Corona 21.5 4 120.1 97  
## Dodge Challenger 15.5 8 318.0 150  
## AMC Javelin 15.2 8 304.0 150  
## Camaro Z28 13.3 8 350.0 245  
## Pontiac Firebird 19.2 8 400.0 175  
## Fiat X1-9 27.3 4 79.0 66  
## Porsche 914-2 26.0 4 120.3 91  
## Lotus Europa 30.4 4 95.1 113  
## Ford Pantera L 15.8 8 351.0 264  
## Ferrari Dino 19.7 6 145.0 175  
## Maserati Bora 15.0 8 301.0 335  
## Volvo 142E 21.4 4 121.0 109

mtcars$drat

## [1] 3.90 3.90 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 3.92 3.07 3.07 3.07  
## [15] 2.93 3.00 3.23 4.08 4.93 4.22 3.70 2.76 3.15 3.73 3.08 4.08 4.43 3.77  
## [29] 4.22 3.62 3.54 4.11

mtcars[["mpg"]]

## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2  
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4  
## [29] 15.8 19.7 15.0 21.4

mtcars[[c(10,3)]]

## [1] 4

Para selecionar linhas e colunas, pode-se usar [i,j] de forma idêntica a uma matriz:

mtcars[10:15,1:3] #Linhas 10 a 15, colunas 1 a 3.

## mpg cyl disp  
## Merc 280 19.2 6 167.6  
## Merc 280C 17.8 6 167.6  
## Merc 450SE 16.4 8 275.8  
## Merc 450SL 17.3 8 275.8  
## Merc 450SLC 15.2 8 275.8  
## Cadillac Fleetwood 10.4 8 472.0

#### Removando NAs

Remover NAs é uma tarefa comum. Para isso, é necessário criar um vetor lógico que indique as posições dos NAs e usá-lo para criar o subconjunto.

x <- c(341,972,NA,NA,177,NA,504)  
rm.na <- is.na(x) #Vetor lógico que indica com as posições em que há NAs.  
x[!rm.na]

## [1] 341 972 177 504

Para remover NAs de múltiplos vetores, uma função útil é complete.cases(). Esta função cria um vetor lógico informando onde NÃO há NAs em nenhum dos vetores informados.

y <- c(NA,2,NA,4,5,NA,7)  
bom <- complete.cases(x,y)  
x[bom]

## [1] 972 177 504

y[bom]

## [1] 2 5 7

Esta função também é bastante útil para remover NAs de matrizes e data frames.

z <- matrix(1:21,7,3)  
z[2,1] <- NA  
z[4,1] <- NA  
z[5,2] <- NA  
z #Acrescentei alguns NAs na matriz.

## [,1] [,2] [,3]  
## [1,] 1 8 15  
## [2,] NA 9 16  
## [3,] 3 10 17  
## [4,] NA 11 18  
## [5,] 5 NA 19  
## [6,] 6 13 20  
## [7,] 7 14 21

bom2 <- complete.cases(z)  
z[bom2,]

## [,1] [,2] [,3]  
## [1,] 1 8 15  
## [2,] 3 10 17  
## [3,] 6 13 20  
## [4,] 7 14 21