Aula 1.3 - Matrizes, listas e data frames

Elder Sodre

Monday, February 23, 2015

#### Matrizes

Uma matriz, no R, é basicamente um vetor com dimensão. Matrizes são compostas por linhas e colunas.

Uma das maneiras para criar uma matriz é pela função matrix():

matrix() #Cria uma matriz vazia

## [,1]  
## [1,] NA

matrix(1:20,nrow=4,ncol=5) #perceba que as matrizes são criadas baseadas nas colunas: a primeira coluna é preenchida com os valores, depois a segunda e assim por diante.

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 5 9 13 17  
## [2,] 2 6 10 14 18  
## [3,] 3 7 11 15 19  
## [4,] 4 8 12 16 20

Matrizes também podem ser criadas adicionando-se dimensão a um vetor:

a <- 1:12  
dim(a) <- c(3,4)  
a

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

Outra forma de criar matrizes é com as funções rbind() e cbind().

x <- 1:5  
y <- c(6.3,10.9,8,11.5,7.4)  
z <- rnorm(5)  
  
rbind(x,y,z) #Função rbind junta as linhas, formando uma nova matriz.

## [,1] [,2] [,3] [,4] [,5]  
## x 1.00 2.0000 3.0000 4.00000 5.000  
## y 6.30 10.9000 8.0000 11.50000 7.400  
## z 2.91 -0.2496 -0.4596 0.01044 1.636

cbind(x,y,z) #Função cbind junta as colunas

## x y z  
## [1,] 1 6.3 2.91031  
## [2,] 2 10.9 -0.24955  
## [3,] 3 8.0 -0.45961  
## [4,] 4 11.5 0.01044  
## [5,] 5 7.4 1.63618

##### Algumas funções úteis para matrizes:

attributes(a) #Retorna os atributos da matriz

## $dim  
## [1] 3 4

dim(a) #Retorna as dimensões da matriz

## [1] 3 4

nrow(a)

## [1] 3

ncol(a)

## [1] 4

t(a) #Transpõe uma matriz

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

Linhas e colunas também podem ter nomes. Uma forma de dar nomes é com as funções cbind e rbind, conforme demonstrado abaixo:

mat <- cbind(ind=x,alt=y,var=z)  
mat

## ind alt var  
## [1,] 1 6.3 2.91031  
## [2,] 2 10.9 -0.24955  
## [3,] 3 8.0 -0.45961  
## [4,] 4 11.5 0.01044  
## [5,] 5 7.4 1.63618

As funções colnames() e rownames() servem para perguntar e para alterar os nomes de colunas e linhas, respectivamente:

colnames(mat)

## [1] "ind" "alt" "var"

rownames(mat)

## NULL

rownames(mat) <- c("A","B","C","D","E")  
mat

## ind alt var  
## A 1 6.3 2.91031  
## B 2 10.9 -0.24955  
## C 3 8.0 -0.45961  
## D 4 11.5 0.01044  
## E 5 7.4 1.63618

attributes(mat)

## $dim  
## [1] 5 3  
##   
## $dimnames  
## $dimnames[[1]]  
## [1] "A" "B" "C" "D" "E"  
##   
## $dimnames[[2]]  
## [1] "ind" "alt" "var"

#### Listas

Listas são um tipo especial de vetor em que os elementos podem ser de tipos diferentes. Listas aparacem como resultado de algumas funções no R.

lis <- list("A",c(T,T,F),4.5,x,mat)  
lis

## [[1]]  
## [1] "A"  
##   
## [[2]]  
## [1] TRUE TRUE FALSE  
##   
## [[3]]  
## [1] 4.5  
##   
## [[4]]  
## [1] 1 2 3 4 5  
##   
## [[5]]  
## ind alt var  
## A 1 6.3 2.91031  
## B 2 10.9 -0.24955  
## C 3 8.0 -0.45961  
## D 4 11.5 0.01044  
## E 5 7.4 1.63618

lis2 <- list(um=1:5,dois="Segundo elemento")  
lis2

## $um  
## [1] 1 2 3 4 5  
##   
## $dois  
## [1] "Segundo elemento"

#### Fatores

Fatores são outro tipo especial de vetor, usado para representar dados categóricos.Podem ser ordenados ou não.

Fatores podem ser criados com a função factor():

fator <- factor(c("lagoa","lagoa","rio","poça","rio","lagoa","poça"))  
fator

## [1] lagoa lagoa rio poça rio lagoa poça   
## Levels: lagoa poça rio

table(fator)

## fator  
## lagoa poça rio   
## 3 2 2

unclass(fator) #O R entende fatores como um vetor de números inteiros

## [1] 1 1 3 2 3 1 2  
## attr(,"levels")  
## [1] "lagoa" "poça" "rio"

#com rótulos.

#### Valores faltantes (NAs)

Valores faltantes são representados por NA e NaN.

a função is.na() pergunta por NAs (not available)

a função is.nan pergunta por NaNs

abc <- c(1,NA,0,4,NaN,2,NA,NaN)  
is.na(abc)

## [1] FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE

is.nan(abc)

## [1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE

Todo Nan também é NA, mas o inverso não é verdadeiro.

NAs também tem classe, então existem character NA, numeric NA, etc.

#### Data Frames

São similares a uma matriz, mas na verdade eles são um tipo de lista em que todos os elementos são vetores com o mesmo comprimento. Pense num data frame como uma lista em que as colunas são os elementos, e o número de linhas é o comprimento dos vetores.

IMPORTANTE! Ao contrário de uma matriz, colunas de um data frame podem ser de tipos diferentes.

Data frames podem ser criados com a função data.frame()

df <- data.frame(sexo=c("H","M","H","M"),peso=c(90,85,100,70))  
df

## sexo peso  
## 1 H 90  
## 2 M 85  
## 3 H 100  
## 4 M 70

class(df)

## [1] "data.frame"

attributes(df)

## $names  
## [1] "sexo" "peso"  
##   
## $row.names  
## [1] 1 2 3 4  
##   
## $class  
## [1] "data.frame"

colnames(df)

## [1] "sexo" "peso"

Muitas funções que vimos em matrizes também funcionam em data frames:

attributes(df)

## $names  
## [1] "sexo" "peso"  
##   
## $row.names  
## [1] 1 2 3 4  
##   
## $class  
## [1] "data.frame"

class(df)

## [1] "data.frame"

dim(df)

## [1] 4 2

ncol(df)

## [1] 2

nrow(df)

## [1] 4

colnames(df)

## [1] "sexo" "peso"

rownames(df)

## [1] "1" "2" "3" "4"

rownames(df) <- c("João","Maria","José","Joaquina")  
df

## sexo peso  
## João H 90  
## Maria M 85  
## José H 100  
## Joaquina M 70