

Revisão de vetores, matrizes, listas e dataframes

Ivanildo Batista

13 de março de 2021

Revisão de Fundamentos da linguagem R

Interagindo com o sistema operacional

```
getwd() #verificando o diretório
```

```
## [1] "C:/Users/junio/OneDrive/Área de Trabalho/Ivanildo Batista/O que vou aprendendo em R/Revisão de Objetos (vetores, matrizes, listas e dataframes) na linguagem R"
```

```
list.files() #Lista de arquivos
```

```
## [1] "Revisão-de-Objetos--vetores,-matrizes,-listas-e-dataframes--na-linguagem-R.html"
## [2] "Revisão-de-Objetos--vetores,-matrizes,-listas-e-dataframes--na-linguagem-R.Rmd"
## [3] "Revisão de Objetos (vetores, matrizes, listas e dataframes) na linguagem R.R"
## [4] "Revisão de Objetos (vetores, matrizes, listas e dataframes) na linguagem R.Rmd"
```

```
.Platform$OS.type #verificando o sistema operacional
```

```
## [1] "windows"
```

```
.Platform$file.sep #verificando o separador de arquivos
```

```
## [1] "/"
```

```
Sys.Date() #verificando a data
```

```
## [1] "2021-03-14"
```

```
Sys.info() #verificando informações sobre o sistema operacional
```

##	sysname	release	version	nodename
##	"Windows"	"10 x64"	"build 19041"	"DESKTOP-730IOCS"
##	machine	login	user	effective_user
##	"x86-64"	"junio"	"junio"	"junio"

```
?system #
```

```
## starting httpd help server ... done
```

```
system('ls -a') #número de arquivos no diretório atual
```

```
## Warning in system("ls -a"): 'ls' not found
```

```
## [1] 127
```

Criando um vetor

```
vec1 = c(12,3,5,6,7,19)
vec1
```

```
## [1] 12  3  5  6  7 19
```

Criando uma matriz

```
dados = c(1:50)
mat1 = matrix(dados, nrow = 5, ncol= 5, byrow=T)
mat1
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   1   2   3   4   5
## [2,]   6   7   8   9  10
## [3,]  11  12  13  14  15
## [4,]  16  17  18  19  20
## [5,]  21  22  23  24  25
```

Criando listas

```
lst1 = list(vec1,mat1)
lst1
```

```
## [[1]]
## [1] 12  3  5  6  7 19
##
## [[2]]
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   1   2   3   4   5
## [2,]   6   7   8   9  10
## [3,]  11  12  13  14  15
## [4,]  16  17  18  19  20
## [5,]  21  22  23  24  25
```

Criando dataframes

```

character_vec = c('A','B','C')
num_vec=c(4.5,5.6,9.8)
logic_vec = c(TRUE,TRUE,FALSE)
df1 = data.frame(character_vec, num_vec,logic_vec)
df1

```

```

##   character_vec num_vec logic_vec
## 1           A     4.5      TRUE
## 2           B     5.6      TRUE
## 3           C     9.8     FALSE

```

Comparando vetores e matrizes

```
vec1>10 #verificar quais valores são maiores que 10
```

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

```
mat1>20 #verifica se cada elemento dentro da matriz é maior que 20
```

```

##      [,1] [,2] [,3] [,4] [,5]
## [1,] FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE
## [5,]  TRUE  TRUE  TRUE  TRUE  TRUE

```

```
mat1==2 #verifica se cada elemento da matriz é igual a 2
```

```

##      [,1] [,2] [,3] [,4] [,5]
## [1,] FALSE  TRUE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE FALSE FALSE

```

```
mat1==2|mat1==3 #verifica se cada elemento é igual a 2 ou igual a 3
```

```

##      [,1] [,2] [,3] [,4] [,5]
## [1,] FALSE  TRUE  TRUE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE FALSE FALSE

```

```
mat1>2&mat1<4 #verifica os valores que são maiores que 2 e menores que 4
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] FALSE FALSE TRUE  FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE FALSE FALSE
```

`!mat1>2&mat1<4` *#verifica os valores que não são maiores que 2, mas que são menores que 4*

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] TRUE  TRUE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE FALSE FALSE
```

Usando um loop for

```
for (i in length(vec1)){
  if (vec1[i]>10){
    print('Pelo menos um elemento do vetor é maior que 10')
  }else{
    print('Todos os elementos dos vetores são menores que 10')
  }
}
```

```
## [1] "Pelo menos um elemento do vetor é maior que 10"
```

Operações com listas e matrizes

```
lst2 = list(2,3,5,7,9,11,13)
lst2
```

```
## [[1]]  
## [1] 2  
##  
## [[2]]  
## [1] 3  
##  
## [[3]]  
## [1] 5  
##  
## [[4]]  
## [1] 7  
##  
## [[5]]  
## [1] 9  
##  
## [[6]]  
## [1] 11  
##  
## [[7]]  
## [1] 13
```

```
for (p in lst2){  
  print(p)  
} #percorrendo a lista
```

```
## [1] 2  
## [1] 3  
## [1] 5  
## [1] 7  
## [1] 9  
## [1] 11  
## [1] 13
```

```
for (i in 1:length(lst2)){  
  print(lst2[[i]])  
} #outra forma de percorrer uma lista
```

```
## [1] 2  
## [1] 3  
## [1] 5  
## [1] 7  
## [1] 9  
## [1] 11  
## [1] 13
```

```
mat1
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    6    7    8    9   10
## [3,]   11   12   13   14   15
## [4,]   16   17   18   19   20
## [5,]   21   22   23   24   25
```

```
mat2=t(mat1) #transpondo uma matriz
mat2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

```
for (i in mat1){
  for (j in mat2){
    total = sum(mat2)+sum(mat1)
  }
} #somando o elemento das matrizes
```

```
total #resultado da soma
```

```
## [1] 650
```

```
print(total)
```

```
## [1] 650
```

Nomeando vetores, matrizes, listas e dataframes

```
##nomeando vetores
names(vec1) = c('Col 1','Col 2','Col 3','Col 4','Col 5', 'Col 6')
vec1
```

```
## Col 1 Col 2 Col 3 Col 4 Col 5 Col 6
##    12    3    5    6    7   19
```

```
##nomeando matrizes
dimnames(mat1) = list(c('OBS 1','OBS 2','OBS 3','OBS 4','OBS 5'),
                      c('Var 1','Var 2','Var 3','Var 4','Var 5'))
mat1
```

```
##          Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1      1     2     3     4     5
## OBS 2      6     7     8     9    10
## OBS 3     11    12    13    14    15
## OBS 4     16    17    18    19    20
## OBS 5     21    22    23    24    25
```

```
##nomeando listas
names(lst1) = c('vetor','Matriz')
lst1
```

```
## $vetor
## [1] 12  3  5  6  7 19
##
## $Matriz
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    6    7    8    9   10
## [3,]   11   12   13   14   15
## [4,]   16   17   18   19   20
## [5,]   21   22   23   24   25
```

```
##nomeando dataframes
colnames(df1) = c('caracteres','Float','Inteiros')
rownames(df1) = c('Obs 1','Obs 2','Obs 3')

df1
```

```
##      caracteres Float Inteiros
## Obs 1          A   4.5      TRUE
## Obs 2          B   5.6      TRUE
## Obs 3          C   9.8     FALSE
```

Verificando e convertendo objetos

```
##verificando
is.vector(vec1)
```

```
## [1] TRUE
```

```
is.vector(mat1)
```

```
## [1] FALSE
```

```
is.matrix(vec1)
```

```
## [1] FALSE
```

```
is.matrix(mat1)
```

```
## [1] TRUE
```

```
##convertendo  
as.vector(mat1) #convertendo de matriz para vetor
```

```
## [1] 1 6 11 16 21 2 7 12 17 22 3 8 13 18 23 4 9 14 19 24 5 10 15 20 25
```

```
as.list(mat1) #convertendo de matriz para lista
```



```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 6
##
## [[3]]
## [1] 11
##
## [[4]]
## [1] 16
##
## [[5]]
## [1] 21
##
## [[6]]
## [1] 2
##
## [[7]]
## [1] 7
##
## [[8]]
## [1] 12
##
## [[9]]
## [1] 17
##
## [[10]]
## [1] 22
##
## [[11]]
## [1] 3
##
## [[12]]
## [1] 8
##
## [[13]]
## [1] 13
##
## [[14]]
## [1] 18
##
## [[15]]
## [1] 23
##
## [[16]]
## [1] 4
##
## [[17]]
## [1] 9
##
## [[18]]
## [1] 14
```

```
##
## [[19]]
## [1] 19
##
## [[20]]
## [1] 24
##
## [[21]]
## [1] 5
##
## [[22]]
## [1] 10
##
## [[23]]
## [1] 15
##
## [[24]]
## [1] 20
##
## [[25]]
## [1] 25
```

```
as.data.frame(mat1) #convertendo de matriz para dataframe
```

```
##      Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1      1      2      3      4      5
## OBS 2      6      7      8      9     10
## OBS 3     11     12     13     14     15
## OBS 4     16     17     18     19     20
## OBS 5     21     22     23     24     25
```

```
as.character(mat1) #convertendo de matriz para caracteres
```

```
## [1] "1"  "6"  "11" "16" "21" "2"  "7"  "12" "17" "22" "3"  "8"  "13" "18" "23"
## [16] "4"  "9"  "14" "19" "24" "5"  "10" "15" "20" "25"
```

```
mat1 #a matriz continua intacta, sem alterações
```

```
##      Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1      1      2      3      4      5
## OBS 2      6      7      8      9     10
## OBS 3     11     12     13     14     15
## OBS 4     16     17     18     19     20
## OBS 5     21     22     23     24     25
```

```
df1
```

```
##      caracteres Float Inteiros
## Obs 1      A    4.5      TRUE
## Obs 2      B    5.6      TRUE
## Obs 3      C    9.8     FALSE
```

```
as.matrix(df1) #convertendo de dataframe para matriz
```

```
##      caracteres Float Inteiros
## Obs 1 "A"      "4.5" "TRUE"
## Obs 2 "B"      "5.6" "TRUE"
## Obs 3 "C"      "9.8" "FALSE"
```

```
as.numeric(mat1) #convertendo de matriz para numérico
```

```
## [1] 1 6 11 16 21 2 7 12 17 22 3 8 13 18 23 4 9 14 19 24 5 10 15 20 25
```

Valores NA

```
mat2 = matrix(1:90,10) #criando uma matriz com 90 elementos e 10 linhas
mat2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,] 1 11 21 31 41 51 61 71 81
## [2,] 2 12 22 32 42 52 62 72 82
## [3,] 3 13 23 33 43 53 63 73 83
## [4,] 4 14 24 34 44 54 64 74 84
## [5,] 5 15 25 35 45 55 65 75 85
## [6,] 6 16 26 36 46 56 66 76 86
## [7,] 7 17 27 37 47 57 67 77 87
## [8,] 8 18 28 38 48 58 68 78 88
## [9,] 9 19 29 39 49 59 69 79 89
## [10,] 10 20 30 40 50 60 70 80 90
```

```
mat2[sample(1:50,10)] = NA #dos valores de 1 a 50, de forma aleatória, serão escolhidos 10 elementos para serem NA
mat2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1   11   21   31   41   51   61   71   81
## [2,]   NA   12   NA   32   42   52   62   72   82
## [3,]   NA   13   23   33   43   53   63   73   83
## [4,]   NA   14   24   34   44   54   64   74   84
## [5,]   NA   15   25   35   45   55   65   75   85
## [6,]   NA   16   NA   NA   46   56   66   76   86
## [7,]   NA   17   27   37   47   57   67   77   87
## [8,]    8   18   28   38   48   58   68   78   88
## [9,]   NA   19   29   39   49   59   69   79   89
## [10,]  10   20   30   40   50   60   70   80   90
```

```
na_lista = is.na(mat2) #matriz com valores TRUE e FALSE, onde TRUE são os valores NA
na_lista
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [4,]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [5,]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [6,]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## [7,]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [9,]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Operações com objetos

```
vec1
```

```
## Col 1 Col 2 Col 3 Col 4 Col 5 Col 6
##    12    3    5    6    7   19
```

```
sum(vec1) #somando os valores de um vetor
```

```
## [1] 52
```

```
mat1
```

```
##      Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1    1    2    3    4    5
## OBS 2    6    7    8    9   10
## OBS 3   11   12   13   14   15
## OBS 4   16   17   18   19   20
## OBS 5   21   22   23   24   25
```

```
rowSums(mat1) #somando os valores das linhas da matriz
```

```
## OBS 1 OBS 2 OBS 3 OBS 4 OBS 5
##    15    40    65    90   115
```

```
colSums(mat1) #somando os valores das colunas da matriz
```

```
## Var 1 Var 2 Var 3 Var 4 Var 5
##    55    60    65    70    75
```

```
vec2 = c(19,63,52,57,99) #criando outro vetor
cbind(vec2,mat1) #unindo vetor e matriz por coluna
```

```
##      vec2 Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1   19    1    2    3    4    5
## OBS 2   63    6    7    8    9   10
## OBS 3   52   11   12   13   14   15
## OBS 4   57   16   17   18   19   20
## OBS 5   99   21   22   23   24   25
```

```
rbind(vec2,mat1) #unindo vetor e matriz por linhas
```

```
##      Var 1 Var 2 Var 3 Var 4 Var 5
## vec2    19   63   52   57   99
## OBS 1     1    2    3    4    5
## OBS 2     6    7    8    9   10
## OBS 3    11   12   13   14   15
## OBS 4    16   17   18   19   20
## OBS 5    21   22   23   24   25
```

```
mat1*3 #multiplicando todos os valores da matriz por 3
```

```
##      Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1     3    6    9   12   15
## OBS 2    18   21   24   27   30
## OBS 3    33   36   39   42   45
## OBS 4    48   51   54   57   60
## OBS 5    63   66   69   72   75
```

Ranking e Ordenamento

```
a=c(100,10,1000) #criando um vetor
a #imprimindo o vetor
```

```
## [1] 100  10 1000
```

```
order(a) #ordem dos elementos do vetor a
```

```
## [1] 2 1 3
```

```
a[order(a)] #reordenando o vetor
```

```
## [1] 10 100 1000
```

Slicing

```
vec1
```

```
## Col 1 Col 2 Col 3 Col 4 Col 5 Col 6
##    12     3     5     6     7    19
```

```
vec1[2] #segundo elemento
```

```
## Col 2
##      3
```

```
mat1
```

```
##      Var 1 Var 2 Var 3 Var 4 Var 5
## OBS 1     1     2     3     4     5
## OBS 2     6     7     8     9    10
## OBS 3    11    12    13    14    15
## OBS 4    16    17    18    19    20
## OBS 5    21    22    23    24    25
```

```
mat1[,2] #segunda coluna da matriz
```

```
## OBS 1 OBS 2 OBS 3 OBS 4 OBS 5
##      2     7    12    17    22
```

```
mat1[1:3, 2:4] #selecionando linhas de 1 a 3 e colunas de 2 a 4
```

```
##      Var 2 Var 3 Var 4
## OBS 1     2     3     4
## OBS 2     7     8     9
## OBS 3    12    13    14
```

```
lst1
```

```
## $vetor
## [1] 12  3  5  6  7 19
##
## $Matriz
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    6    7    8    9   10
## [3,]   11   12   13   14   15
## [4,]   16   17   18   19   20
## [5,]   21   22   23   24   25
```

```
lst1[[1]] #primeiro elemento da lista
```

```
## [1] 12  3  5  6  7 19
```

```
lst1[[1]][4] # quarto elemento do primeiro elemento da lista
```

```
## [1] 6
```

```
lst1$Matriz #selecionando um elemento da lista pelo seu nome
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    6    7    8    9   10
## [3,]   11   12   13   14   15
## [4,]   16   17   18   19   20
## [5,]   21   22   23   24   25
```

Dataframes

```
mtcars #dataframe que vem com o R
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160.0 110  3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160.0 110  3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8   4  108.0  93  3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4   6  258.0 110  3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360.0 175  3.15 3.440 17.02  0  0    3    2
## Valiant        18.1   6  225.0 105  2.76 3.460 20.22  1  0    3    1
## Duster 360     14.3   8  360.0 245  3.21 3.570 15.84  0  0    3    4
## Merc 240D      24.4   4  146.7  62  3.69 3.190 20.00  1  0    4    2
## Merc 230       22.8   4  140.8  95  3.92 3.150 22.90  1  0    4    2
## Merc 280       19.2   6  167.6 123  3.92 3.440 18.30  1  0    4    4
## Merc 280C      17.8   6  167.6 123  3.92 3.440 18.90  1  0    4    4
## Merc 450SE     16.4   8  275.8 180  3.07 4.070 17.40  0  0    3    3
## Merc 450SL     17.3   8  275.8 180  3.07 3.730 17.60  0  0    3    3
## Merc 450SLC    15.2   8  275.8 180  3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8  472.0 205  2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8  460.0 215  3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial 14.7   8  440.0 230  3.23 5.345 17.42  0  0    3    4
## Fiat 128        32.4   4   78.7  66  4.08 2.200 19.47  1  1    4    1
## Honda Civic     30.4   4   75.7  52  4.93 1.615 18.52  1  1    4    2
## Toyota Corolla  33.9   4   71.1  65  4.22 1.835 19.90  1  1    4    1
## Toyota Corona   21.5   4  120.1  97  3.70 2.465 20.01  1  0    3    1
## Dodge Challenger 15.5   8  318.0 150  2.76 3.520 16.87  0  0    3    2
## AMC Javelin     15.2   8  304.0 150  3.15 3.435 17.30  0  0    3    2
## Camaro Z28      13.3   8  350.0 245  3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird 19.2   8  400.0 175  3.08 3.845 17.05  0  0    3    2
## Fiat X1-9       27.3   4   79.0  66  4.08 1.935 18.90  1  1    4    1
## Porsche 914-2   26.0   4  120.3  91  4.43 2.140 16.70  0  1    5    2
## Lotus Europa    30.4   4   95.1 113  3.77 1.513 16.90  1  1    5    2
## Ford Pantera L  15.8   8  351.0 264  4.22 3.170 14.50  0  1    5    4
## Ferrari Dino    19.7   6  145.0 175  3.62 2.770 15.50  0  1    5    6
## Maserati Bora   15.0   8  301.0 335  3.54 3.570 14.60  0  1    5    8
## Volvo 142E     21.4   4  121.0 109  4.11 2.780 18.60  1  1    4    2
```

```
head(mtcars) #primeiras Linhas
```

```
##          mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110  3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160 110  3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8   4  108  93  3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4   6  258 110  3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175  3.15 3.440 17.02  0  0    3    2
## Valiant        18.1   6  225 105  2.76 3.460 20.22  1  0    3    1
```

```
tail(mtcars) #últimas Linhas
```



```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa  30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

```
str(mtcars) #informações sobre cada coluna do dataframe
```

```
## 'data.frame':   32 obs. of  11 variables:
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num   6  6  4  6  8  6  8  4  4  6 ...
## $ disp: num  160 160 108 258 360 ...
## $ hp  : num  110 110  93 110 175 105 245  62  95 123 ...
## $ drat: num   3.9  3.9  3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt  : num   2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num   16.5 17 18.6 19.4 17 ...
## $ vs  : num   0  0  1  1  0  1  0  1  1  1 ...
## $ am  : num   1  1  1  0  0  0  0  0  0  0 ...
## $ gear: num   4  4  4  3  3  3  3  4  4  4 ...
## $ carb: num   4  4  1  1  2  1  4  2  2  4 ...
```

```
mtcars[,5] #coluna 5 do dataframe
```

```
## [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 2.93
## [16] 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 4.22 3.62
## [31] 3.54 4.11
```

```
mtcars[1:4,7:9] #linhas de 1 a 4 e colunas 7 a 9
```

```
##           qsec vs am
## Mazda RX4    16.46  0  1
## Mazda RX4 Wag 17.02  0  1
## Datsun 710    18.61  1  1
## Hornet 4 Drive 19.44  1  0
```

```
mtcars[1:4,'hp'] #linhas de 1 a 4 da coluna 'hp'
```

```
## [1] 110 110  93 110
```

```
mtcars$hp #coluna 'hp'
```

```
## [1] 110 110  93 110 175 105 245  62  95 123 123 180 180 180 205 215 230  66  52
## [20]  65  97 150 150 245 175  66  91 113 264 175 335 109
```

```
subset(mtcars, subset=hp>170) #criando um subdataset onde os valores de hp são maiores que 170
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0   3   2
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0   3   4
## Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0   3   3
## Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0   3   3
## Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0   3   3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0   3   4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0   3   4
## Chrysler Imperial  14.7   8 440.0 230 3.23 5.345 17.42  0  0   3   4
## Camaro Z28        13.3   8 350.0 245 3.73 3.840 15.41  0  0   3   4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0   3   2
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1   5   4
## Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50  0  1   5   6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1   5   8
```

```
library(dplyr) #importando o pacote dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
slice(mtcars, 1:4) #criando um subdataset com as linhas de 1 a 4
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160 110 3.90 2.620 16.46  0  1   4   4
## Mazda RX4 Wag  21.0   6 160 110 3.90 2.875 17.02  0  1   4   4
## Datsun 710     22.8   4 108  93 3.85 2.320 18.61  1  1   4   1
## Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0   3   1
```

```
order(mtcars) #ordenando os valores do dataframe
```

```
## [1] 225 226 229 231 236 237 238 239 240 241 246 247 248 249 251 253 254 255
## [19] 260 261 262 263 264 265 266 267 268 269 270 271 272 273 277 278 279 280
## [37] 281 227 228 230 232 233 234 235 242 243 244 245 250 252 256 257 258 259
## [55] 274 275 276 282 283 284 285 286 287 288 323 324 326 338 340 341 346 188
## [73] 179 180 186 325 328 329 339 342 343 345 347 348 352 187 178 163 181 161
## [91] 134 150 190 192 162 143 144 292 293 294 295 300 301 302 303 304 305 309
## [109] 310 311 312 313 332 333 334 140 141 142 132 153 133 151 169 189 168 135
## [127] 164 145 183 165 170 171 166 182 159 167 191 158 136 149 152 173 156 174
## [145] 184 185 131 129 130 137 138 139 35 40 41 50 51 52 53 58 59 60
## [163] 64 289 290 291 296 297 298 299 306 307 308 314 320 321 322 327 330 331
## [181] 335 336 337 344 349 172 146 154 160 148 157 155 147 315 316 317 318 319
## [199] 175 177 176 33 34 36 38 42 43 62 350 37 39 44 45 46 47 48
## [217] 49 54 55 56 57 61 63 351 15 16 24 7 221 223 17 31 14 23
## [235] 216 22 222 29 199 12 193 219 214 220 194 197 217 13 215 204 209 205
## [253] 11 208 207 206 6 202 211 224 195 5 203 218 10 25 196 210 30 212
## [271] 200 213 198 1 2 4 32 21 3 9 201 8 27 26 19 28 18 20
## [289] 115 104 116 114 122 84 83 82 90 123 99 105 92 117 102 67 128 97
## [307] 98 100 124 85 91 96 106 107 73 94 72 118 119 65 66 74 75 101
## [325] 121 126 108 109 110 111 112 70 113 103 120 68 125 76 77 78 95 87
## [343] 86 127 88 93 69 71 89 81 80 79
```

```
order(mtcars$mpg) #ordenando os valores da coluna 'mpg'.
```

```
## [1] 15 16 24 7 17 31 14 23 22 29 12 13 11 6 5 10 25 30 1 2 4 32 21 3 9
## [26] 8 27 26 19 28 18 20
```