

# Aula 02 - Tipos de dados

Laís Salgueiro, Murilo Dias e Yan Soares

2022-09-29 16:06:59

## Contents

<b>1</b>	<b>Introdução ao R - Tipos de dados</b>	<b>1</b>
1.1	CLASSES que o valores podem ter . . . . .	1
1.1.1	Uso de sinais lógicos . . . . .	2
1.2	MÃO NA MASSA. . . . .	3
1.3	ESTRUTURA DE DADOS . . . . .	3
1.3.0.1	VETORES . . . . .	3
1.4	MÃO NA MASSA . . . . .	4
1.5	MÃO NA MASSA . . . . .	6
1.5.1	Explorando vetores . . . . .	6
1.6	Criando gráficos . . . . .	9
1.7	Tipos de objetos . . . . .	12
1.7.1	Matrizes . . . . .	12
1.7.2	Data Frame . . . . .	12
1.7.3	Lista . . . . .	13

## 1 Introdução ao R - Tipos de dados

### 1.1 CLASSES que o valores podem ter

- INTEGERS - números naturais, ex:124
- NUMERICS = valores decimais, ex: 124.9

### Exemplos

```
numerico <- c(59.2, 6.2, 900)
numerico
```

```
## [1] 59.2 6.2 900.0
```

- LÓGICOS (TRUE, FALSE) = podem ser abreviados como T e F

## Exemplos

```
logico <- c(TRUE, FALSE, F, T, T, T, F, F)
logico
```

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

- CHARACTERS = textos (use aspas)

## Exemplos

```
caracter <- c("lais", "Yan", "Murilo")
caracter
```

```
## [1] "lais" "Yan" "Murilo"
```

**OBS** - para checar como o R está interpretando seus dados use a função *class()*

```
class(numerico)
```

```
## [1] "numeric"
```

```
class(caracter)
```

```
## [1] "character"
```

### 1.1.1 Uso de sinais lógicos

Use `==` para verificar se valores são iguais. O R retornará uma resposta lógica

```
2+2==5
```

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

```
8==8
```

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

Use `!=` para verificar se valores NÃO são iguais. O R retornará uma resposta lógica

```
2+2!=5
```

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

```
8!=8
```

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

Use `<` (**menor que**) ou `>` (**maior que**); `<=` **menor ou igual** ou `>=` **maior ou igual**. O R retornará uma resposta lógica

```
2+2<5 #menor que
```

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

```
8>=8 #maior ou igual
```

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

## 1.2 MÃO NA MASSA. . .

Crie uma variável (objeto) com valor 42 e divida por 2. Verifique qual a classe do objeto. Depois, substitua o valor da variável por “Eita!”. Cheque seu valor e cheque a sua classe.

## 1.3 ESTRUTURA DE DADOS

**1.3.0.1 VETORES** São uma ferramenta de armazenamento de dados/valores (classes: numéricos, caracteres ou lógicos). Para criar um vetor pode-se usar o comando `c()`, por exemplo.

```
vec1 = c(1,4,6,8,10) #vetor numérico  
vec1
```

```
## [1] 1 4 6 8 10
```

```
vec1[5] #mostra o quinto elemento
```

```
## [1] 10
```

```
vetor_caracter<-c("x", "y", "z") #criando vetor com caracteres  
vetor_caracter
```

```
## [1] "x" "y" "z"
```

Outra maneira de criar vetores com sequências de números

```
vec2 = seq(from=0, to=1, by=0.25)  
vec2
```

```
## [1] 0.00 0.25 0.50 0.75 1.00
```

```
sum(vec1, vec2)
```

```
## [1] 31.5
```

## 1.4 MÃO NA MASSA . . .

Vamos criar um vetor com a idade da turma. E vincular cada idade aos nomes.

```
idade_galera <- c(19,21,26,44,49,29,37,38,18,26,31)
idade_galera
```

```
## [1] 19 21 26 44 49 29 37 38 18 26 31
```

Criando vetor com caracteres

```
nomes<-c("Gabi",
         "Yan",
         "Cleberson",
         "Roberto",
         "Rosa",
         "Clarine",
         "Marcelo",
         "Gilson",
         "Fabricio",
         "Tawara",
         "Yuri")
names(idade_galera)<- nomes
idade_galera
```

```
##      Gabi      Yan Cleberson  Roberto      Rosa  Clarine  Marcelo  Gilson
##      19      21      26      44      49      29      37      38
## Fabricio  Tawara      Yuri
##      18      26      31
```

**OBSERVAÇÃO:** o que são fatores? São dados categóricos

```
genero <- c("M", "H" , "H", "M", "H", "H")
genero
```

```
## [1] "M" "H" "H" "M" "H" "H"
```

```
class(genero)
```

```
## [1] "character"
```

Para interpretar dados como fatores (categóricos), utilizamos: **factor()**

```
fgenero <- factor(genero)
fgenero
```

```
## [1] M H H M H H
## Levels: H M
```

Outro exemplo de fator: categórico e ordinal

```
temperatura_vector <- c("High", "Low", "High", "Low", "Medium")
class(temperatura_vector)
```

```
## [1] "character"
```

*#duvidas sobre a função factor? Pesquise no help*

Utilizamos essa função para definir níveis entre as variáveis categóricas

```
factor_temperatura_vector <- factor(temperatura_vector,
order = TRUE, levels = c("Low", "Medium", "High"))
factor_temperatura_vector
```

```
## [1] High Low High Low Medium
## Levels: Low < Medium < High
```

Use levels () para identificar os níveis e is.factor () para verificar se o objeto é um fator

```
levels (genero)
```

```
## NULL
```

```
levels(factor_temperatura_vector)
```

```
## [1] "Low" "Medium" "High"
```

Sumarize os dados

```
summary (genero)
```

```
##      Length      Class      Mode
##           6 character character
```

```
summary(factor_temperatura_vector)
```

```
##      Low Medium   High
##         2      1      2
```

## 1.5 MÃO NA MASSA . . .

Um time de 5 analistas de dados em R são categorizados quanto a sua rapidez. Crie uma ordenação das categorias (fator com ordenação) e aplique o summary. Depois, compare o analista 2 com o analista 5 (atribua um objeto).

```
velocidade <- c("Fast","Slow","Slow","Fast","Ultra-fast","Ultra-fast","Ultra-fast")
```

### 1.5.1 Explorando vetores

Exemplo com valores positivos e negativos:

```
#Mercado de acoes  
petrobras <- c(140, -50, 20, -120, 240)  
petrobras
```

```
## [1] 140 -50 20 -120 240
```

```
magaluiza <- c(-24, -50, 100, -350, 10)  
magaluiza
```

```
## [1] -24 -50 100 -350 10
```

Quanto rendeu as acoes na semana em cada investimento e no total?

```
total_pet <- sum (petrobras)  
total_pet
```

```
## [1] 230
```

```
total_maga <- sum (magaluiza)  
total_maga
```

```
## [1] -314
```

```
rendimento <- sum (petrobras, magaluiza)  
rendimento
```

```
## [1] -84
```

Petrobras foi mais rentável?

```
total_pet>total_maga
```

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

Qual foi a média de rendimentos?

```
m_pet <- mean (petrobras)
m_maga <- mean (magaluiza)
m_rendimento <- mean (rendimento)
```

investimentos

```
dias_semana <- c("Seg", "Ter", "Qua", "Qui", "Sex")
names(petrobras) <- dias_semana
names(magaluiza) <- dias_semana
petrobras #checando
```

```
## Seg Ter Qua Qui Sex
## 140 -50 20 -120 240
```

```
magaluiza #checando
```

```
## Seg Ter Qua Qui Sex
## -24 -50 100 -350 10
```

Selecionando terça-feira

```
petrobras [2]
```

```
## Ter
## -50
```

```
magaluiza[2]
```

```
## Ter
## -50
```

```
rendimento [2]
```

```
## [1] NA
```

Selecionando segunda e sexta

```
petrobras [c(1,5)]
```

```
## Seg Sex
## 140 240
```

```
magaluiza[c(1,5)]
```

```
## Seg Sex
## -24 10
```

```
rendimento [c(1,5)]
```

```
## [1] -84 NA
```

Selecionando a sequência de terça a quinta

```
petrobras [c(2:4)]
```

```
## Ter Qua Qui  
## -50 20 -120
```

```
magaluiza[c(2:4)]
```

```
## Ter Qua Qui  
## -50 100 -350
```

```
rendimento [c(2:4)]
```

```
## [1] NA NA NA
```

```
petrobras[c("Seg","Qui")]
```

```
## Seg Qui  
## 140 -120
```

```
(magaluiza)<0 #verificando acoes negativas - prejuizo
```

```
## Seg Ter Qua Qui Sex  
## TRUE TRUE FALSE TRUE FALSE
```

```
(magaluiza)>0 #verificando acoes positivas - lucro
```

```
## Seg Ter Qua Qui Sex  
## FALSE FALSE TRUE FALSE TRUE
```

Para modificar ou adicionar algum valor ao vetor

```
petrobras["Sex"]<-300  
petrobras
```

```
## Seg Ter Qua Qui Sex  
## 140 -50 20 -120 300
```



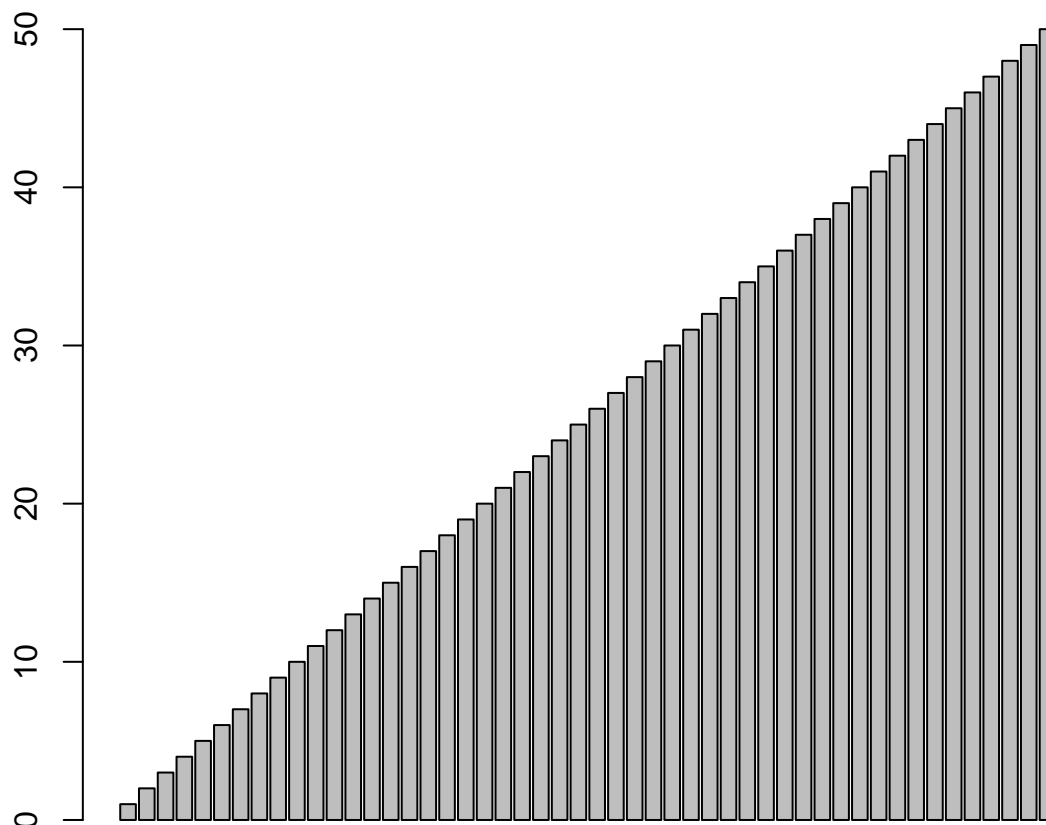
```
magaluiza [5] <- 400
magaluiza
```

```
## Seg Ter Qua Qui Sex
## -24 -50 100 -350 400
```

## 1.6 Criando gráficos

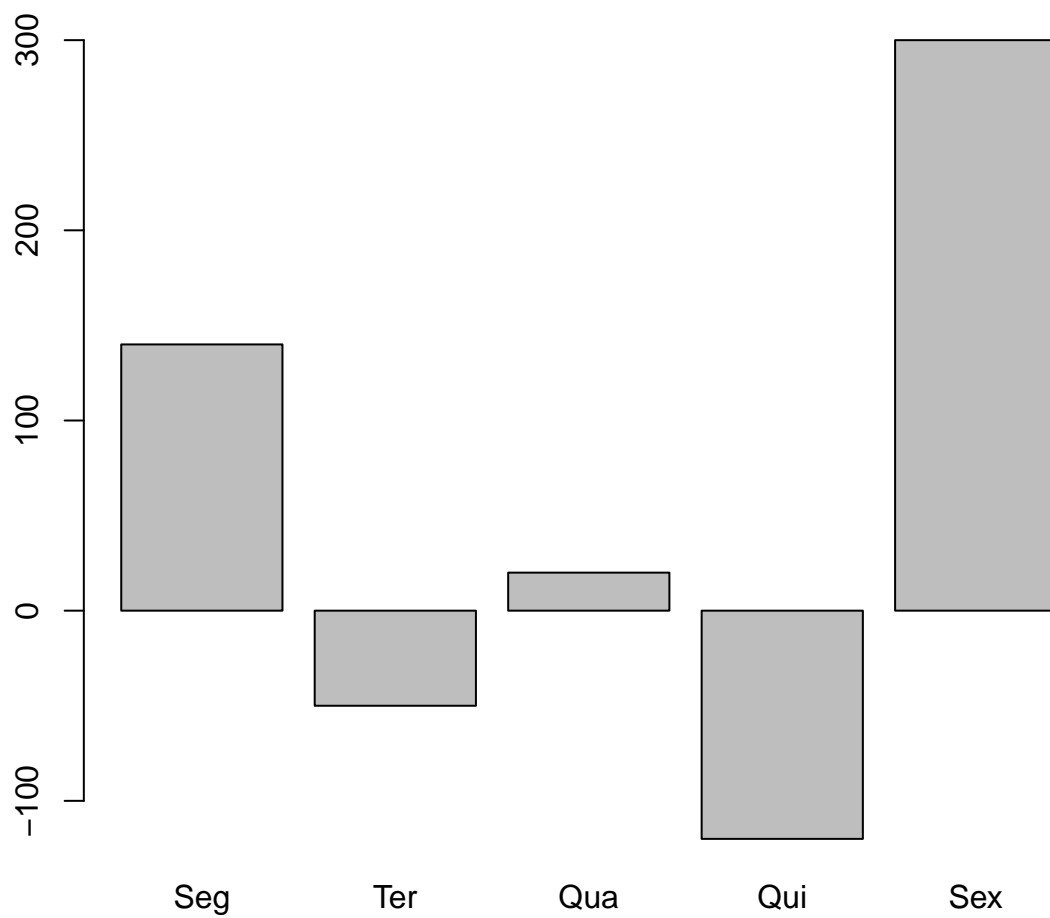
Primeiro nós vamos criar um gráfico do tipo de barras para isso vamos utilizar a função **barplot**

```
barplot(1:50)
```



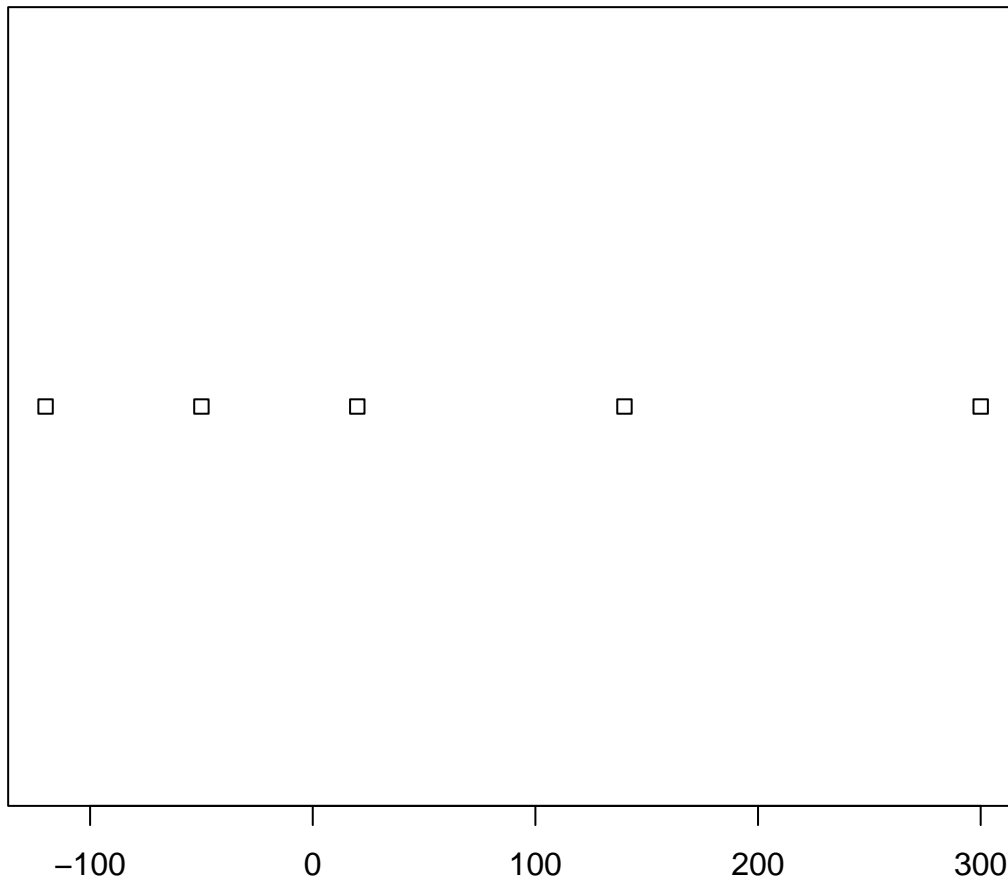
Agora vamos fazer com o conjunto de dados petrobras

```
barplot(petrobras)
```



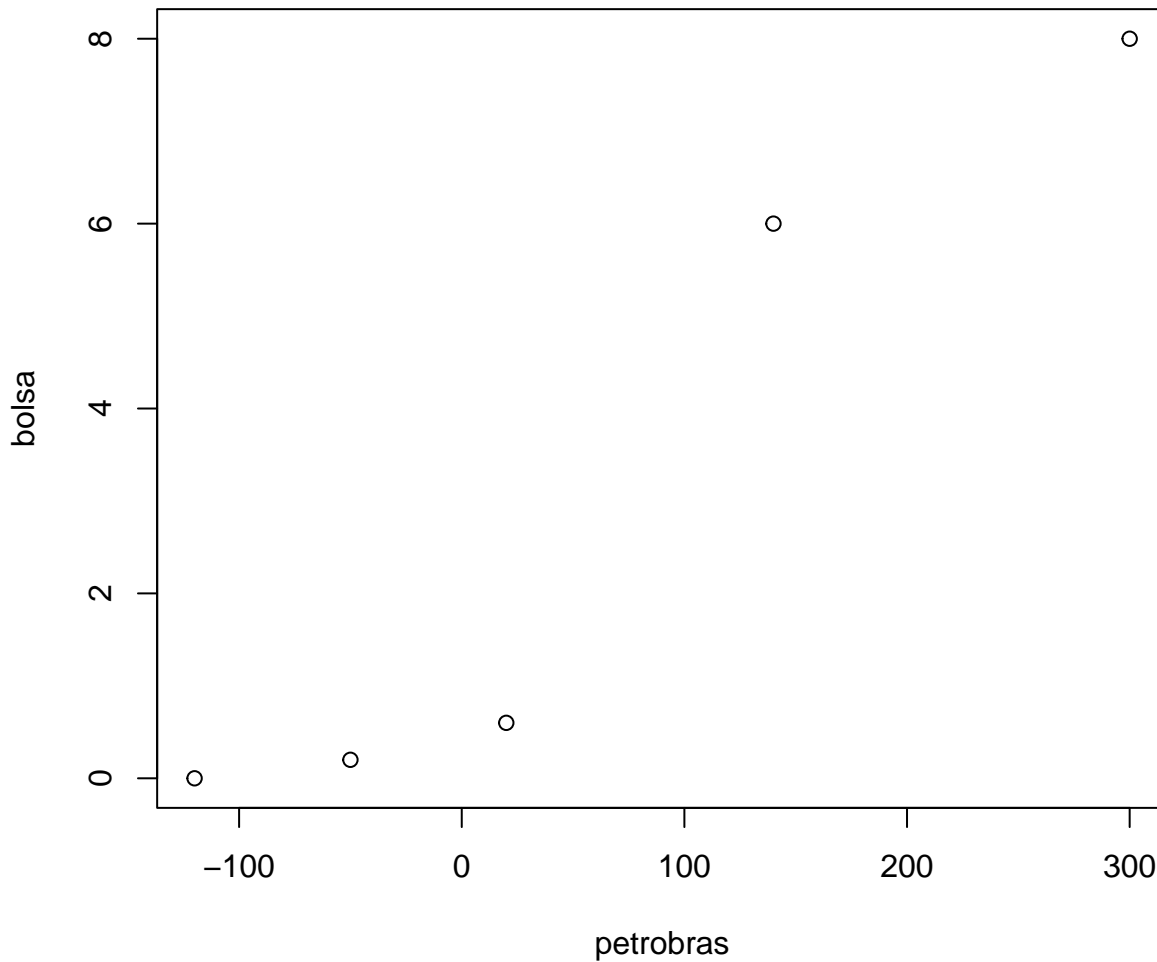
Agora vamos fazer um gráfico do tipo stripchart

```
stripchart(petrobras)
```



Agora vamos fazer uma gráfico de duas variáveis (plot x, y)

```
bolsa <- c(6, 0.2, 0.6, 0, 8) #pontos da bolsa.  
names(bolsa)<-dias_semana  
plot (petrobras, bolsa)
```



## 1.7 Tipos de objetos

### 1.7.1 Matrizes

São um conjunto de elementos dos mesmos tipos de dados. Em linhas e colunas (*bidimensional*)

```
mat=matrix(data=c(9,2,3,4,5,6),ncol=3)
mat
```

```
##      [,1] [,2] [,3]
## [1,]    9    3    5
## [2,]    2    4    6
```

### 1.7.2 Data Frame

É semelhante a uma matriz, porém as colunas têm nomes e, por isso, podem conter dados de tipos diferentes.

```
t <- data.frame(x = c(11,12,14), y = c(19,20,21), z = c(10,9,7))
t
```

```
##      x  y  z
## 1 11 19 10
## 2 12 20  9
## 3 14 21  7
```

### 1.7.3 Lista

Todo data frame é uma lista; se parecem com vetores; podem conter matrizes, dataframes e outros tipos de dados

```
L = list(one=1, two=c(1,2),five=seq(0, 1, length=5))
L
```

```
## $one
## [1] 1
##
## $two
## [1] 1 2
##
## $five
## [1] 0.00 0.25 0.50 0.75 1.00
```