

# Sumário

<b>1. Aprendendo R .....</b>	<b>1</b>
1.1 Variáveis .....	1
1.2 Operadores .....	1
1.3 Vetores   Função c() .....	3
1.4 Condições .....	4
<b>1.5 Loops .....</b>	<b>5</b>
1.5.1 Loop for .....	5
1.5.2 Loop while .....	5
1.6 Definir Funções .....	5
<b>2. Conectando ao Banco de Dados .....</b>	<b>6</b>
2. 1 Conexão .....	6
2.2 Carregando Pacotes .....	7
2.3 Plotando Gráficos .....	7
<b>3. Referências .....</b>	<b>8</b>
3.1 Sugestões .....	9

# 1 Aprendendo R

## 1.1 Variáveis

Formas de criar variáveis em R

Operador de atribuição: <-

In [7]:

```
▼ # Formato1: mais usual  
a <- 10  
  
# Formato2  
20 -> b  
  
# Formato3  
c = 30
```

executed in 29ms, finished 16:45:32 2021-09-03

## 1.2 Operadores

### Operadores Aritméticos

Os operadores mais comuns do R são os aritméticos:

Operador	Ação
+	somar
-	subtrair
/	dividir
*	multiplicar
^	elevar à uma potência

In [8]:

```
▼ # teste  
result <- a * b + c / a - c^2  
result
```

executed in 356ms, finished 16:45:33 2021-09-03

-697

## Operadores Relacionais

Operadores relacionais realizam ações de verificação da relação entre dois objetos, comparando se são iguais, diferentes, maiores ou menores. O resultado de uma operação dessas é sempre uma variável LÓGICA (**TRUE** ou **FALSE**). Esse resultado é armazenado em variáveis chamadas de booleanas ou lógicas.

Operadores relacionais	Ação
<	menor que ..
<=	menor ou igual a ...
>	maior que ...
>=	maior ou igual a ...
==	exatamente igual a ...
!	não / negação
!=	diferente de ...

In [9]:

```
# teste  
4 < 5
```

executed in 32ms, finished 16:45:34 2021-09-03

TRUE

## Operadores Lógicos

Operadores relacionais comparam dois objetos. Quando precisamos realizar mais de uma comparação ao mesmo tempo, precisamos usar os operadores lógicos. No R, os principais operadores lógicos são, **&** e **|**, que significam, respectivamente, AND e OR. O resultado de comparações múltiplas terá também como resultado uma variável lógica (**TRUE** ou **FALSE**).

In [10]:

```
# teste: pegando as variáveis criadas acima  
# utilizando operador AND  
result1 <- (a < b) & (a > c)  
result1  
  
# utilizando operador OR  
result2 <- (a > b) | (a < c)  
result2
```

executed in 44ms, finished 16:45:35 2021-09-03

FALSE

TRUE

In [ ]:

## 1.3 Vetores | Função c()

Vetores é um conjunto de elementos da mesma natureza, a função `c()` é chamada de combinar, e é muito utilizada para criar vetores.

In [17]:

```
# criando vetor numérico
numeros <- c(1,5,3,7,1,5)
'Vetor de números:'
numeros

# criando sequência de números
sequencia <- c(5:0)
'Sequência em ordem decrescente:'
sequencia

# vetor de strings
letras <- c('João', 'Pedro', 'Ana', 'Zack')
'Vetor de letras:'
letras

# vetor misto
mistura <- c('abacate', 2, 'banana', 3)
"Vetor misto"
mistura
```

executed in 96ms, finished 15:36:31 2021-09-03

'Vetor de números:'

1 5 3 7 1 5

'Sequência em ordem decrescente:'

5 4 3 2 1 0

'Vetor de letras:'

'João' 'Pedro' 'Ana' 'Zack'

'Vetor misto'

'abacate' '2' 'banana' '3'

### OBSERVAÇÕES:

>> Para criar uma sequência, é utilizado : (dois pontos), seguido da sequência de números desejados

>> Ao criar um vetor misto, note que os números passam a ser strings

## 1.4 Condições

### Sintaxe

```
if ( condição for verdadeira ) {  
  ----comando para executar  
} else {  
  ----comando para executar  
}
```

In [18]:

```
▼ # exemplo  
▼ if (a > c) {  
  print("Olá R")  
▼ } else {  
  print("É menor")  
}
```

executed in 37ms, finished 15:36:31 2021-09-03

[1] "É menor"

### Sintaxe

```
ifelse (testa, se verdadeiro, se falso)
```

In [11]:

```
ifelse (c == a, "é igual", "não é igual")
```

executed in 1.76s, finished 16:47:04 2021-09-03

'não é igual'

In [ ]:

## 1.5 Loops

### 1.5.1 loop for

```
for ( variável in lista ) { instrução }
```

In [13]:

```
lista <- c(1,4,32,76,85)

# utilizando FOR
▼ for (i in lista) {
  print(i)
}
```

executed in 38ms, finished 16:48:13 2021-09-03

```
[1] 1
[1] 4
[1] 32
[1] 76
[1] 85
```

### 1.5.2 loop while

```
while ( condição ) { instrução }
```

In [14]:

```
x <- 0
▼ while (x < length(lista)) {
  x <- x + 1
  print(x)
}
```

executed in 35ms, finished 16:48:13 2021-09-03

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

In [ ]:

## 1.6 Definir funções

## Sintaxe

```
variavel <- function ( arg1, arg2... ) { estrutura da função }
```

In [22]:

```
▼ somar <- function (x, y) {  
  sum <- x + y  
  return (sum)  
}  
somar(2, 6)
```

executed in 34ms, finished 15:36:31 2021-09-03

8

## 2 Conectando ao banco de dados

In [3]:

```
▼ # Requirements  
# install.packages('odbc')
```

executed in 27ms, finished 16:34:17 2021-09-03

In [4]:

```
require(odbc)
```

executed in 27ms, finished 16:34:18 2021-09-03

### 2.1 Conectando ao banco de dados

**Caso o banco já esteja pré definido na função:**

```
con <- dbConnect ( odbc(), "nome da fonte de dados" )  
con
```

**Passando parâmetros para conexão**

```
con <- dbConnect ( odbc(), Driver = "SQL Server",
```

```
  Server = "SERVIDOR_HOSPEDADO",
```

```
  UID = rstudioapi::askForPassword("Insira seu usuário: "),
```

```
  PWD = rstudioapi::askForPassword("Insira sua senha"),
```

```
  encoding = "latin1"
```

```
)
```

### Realizando Query

```
QUERY_VENDAS <- paste0 (  
  
    "SET NOCOUNT ON  
  
    SET ANSI_WARNINGS OFF  
  
    SELECT  *  
  
    FROM VENDAS  
  
    SET NOCOUNT OFF  
  
    SET ANSI_WARNINGS ON"  
)
```

```
VENDAS <- dbGetQuery(con, QUERY_VENDAS)
```

## 2.2 Carregando pacotes/bibliotecas

### Sintaxe

```
library ( nome do pacote )
```

In [ ]:

In [ ]:

In [ ]:

## 2.3 Plotando gráficos

In [35]:

```
▼ # Requirements  
# install.packages('plotly')
```

executed in 26ms, finished 16:23:30 2021-09-03



In [34]:

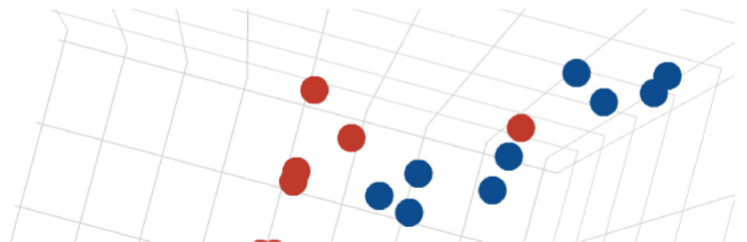
```
library(plotly)

mtcars$am[which(mtcars$am == 0)] <- 'Automatic'
mtcars$am[which(mtcars$am == 1)] <- 'Manual'
mtcars$am <- as.factor(mtcars$am)

fig <- plot_ly(mtcars, x = ~wt, y = ~hp, z = ~qsec, color = ~am, colors = c('#BF382A', '#1f77b4'))
fig <- fig %>% add_markers()
fig <- fig %>% layout(scene = list(xaxis = list(title = 'Weight'),
                                   yaxis = list(title = 'Gross horsepower'),
                                   zaxis = list(title = '1/4 mile time'))))

fig
```

executed in 3.04s, finished 16:11:59 2021-09-03



### 3 Referências

NOTA: Faça uso das referências para maiores informações e aprendizado.

0. [Guia de Markdown \(https://www.markdownguide.org/basic-syntax/\)](https://www.markdownguide.org/basic-syntax/)

1. [Estatística é com R! \(http://www.estadisticacomr.uff.br/?p=209\)](http://www.estadisticacomr.uff.br/?p=209)

2. [Livro: Manual Básico da Linguagem R: introdução à análise de dados \(https://www.amazon.com.br/gp/product/B07N493V6J/ref=ppx\\_yo\\_dt\\_b\\_d\\_asin\\_title\\_o00?ie=UTF8&psc=1\)](https://www.amazon.com.br/gp/product/B07N493V6J/ref=ppx_yo_dt_b_d_asin_title_o00?ie=UTF8&psc=1)

3. [Institute for Integrative Genome biology \(http://manuals.bioinformatics.ucr.edu/home/programming-in-r#TOC-Apply-Loop-Family\)](http://manuals.bioinformatics.ucr.edu/home/programming-in-r#TOC-Apply-Loop-Family)

4. [Guia rápido de R \(http://www.leg.ufpr.br/~walmes/cursoR/guia\\_rapido\\_R.pdf\)](http://www.leg.ufpr.br/~walmes/cursoR/guia_rapido_R.pdf)

5. [Consulta banco SQL Server com R \(https://www.dadosaleatorios.com.br/post/consulta-sqlserver-r/\)](https://www.dadosaleatorios.com.br/post/consulta-sqlserver-r/)

## 3.1 Sugestões :

[Aprenda usar R + SQL server \(passo a passo\) \(https://docs.microsoft.com/pt-br/sql/machine-learning/tutorials/walkthrough-view-and-summarize-data-using-r?view=sql-server-ver15\)](https://docs.microsoft.com/pt-br/sql/machine-learning/tutorials/walkthrough-view-and-summarize-data-using-r?view=sql-server-ver15)