

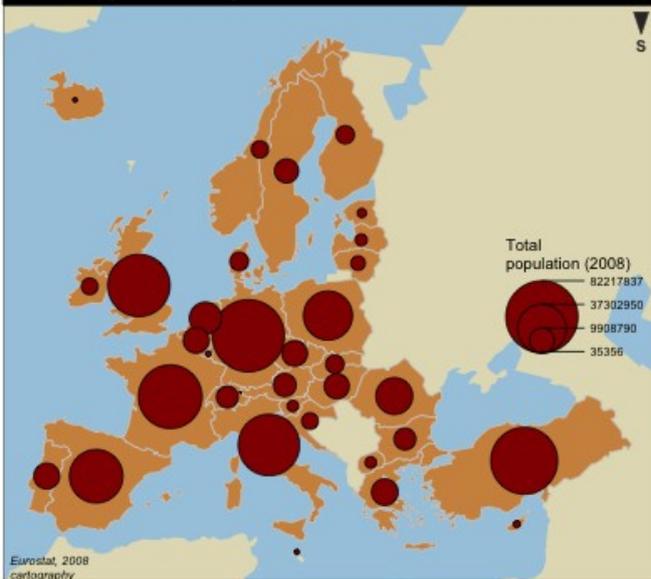
VPS 5712

Revisão do R

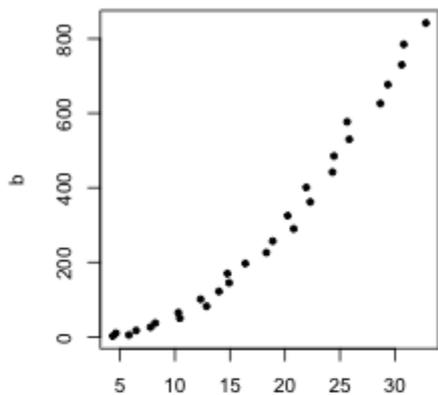
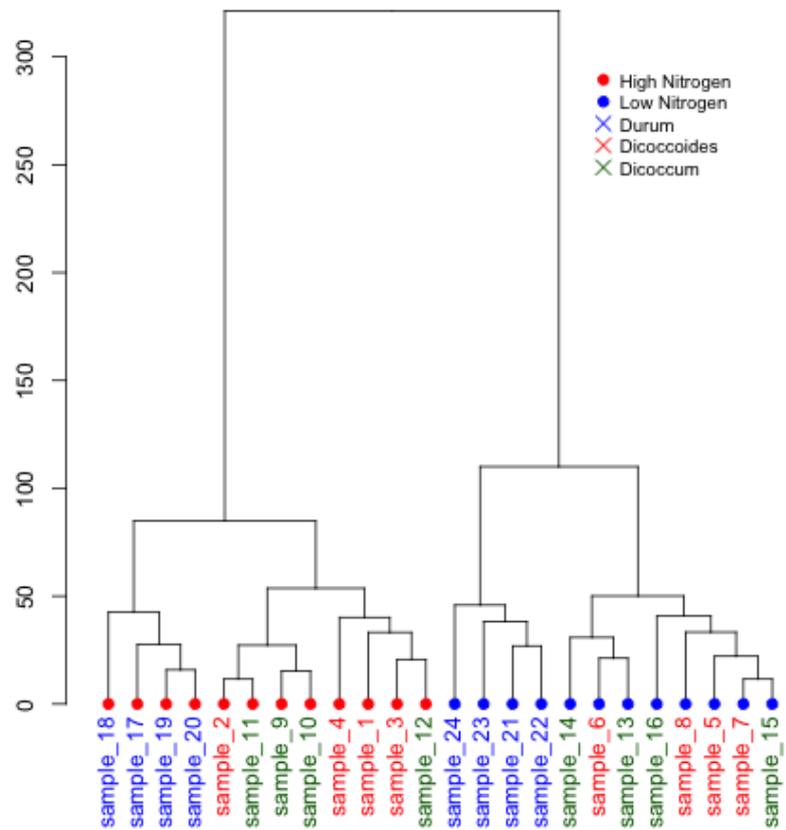
R

- Por que R?

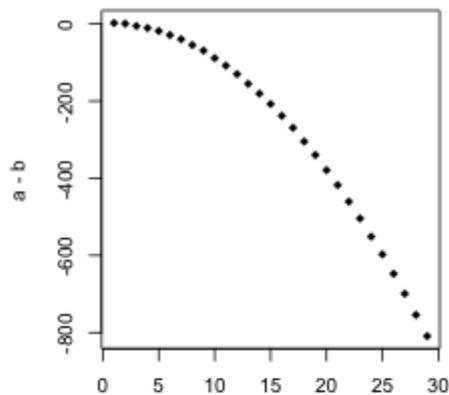
Countries Population in Europe



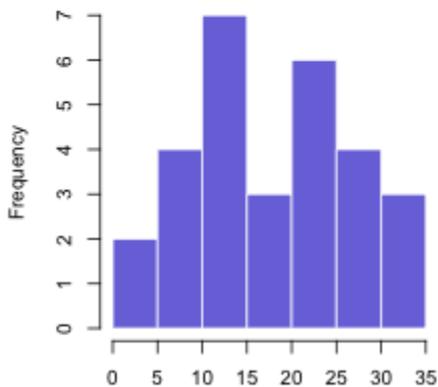
structure of the population



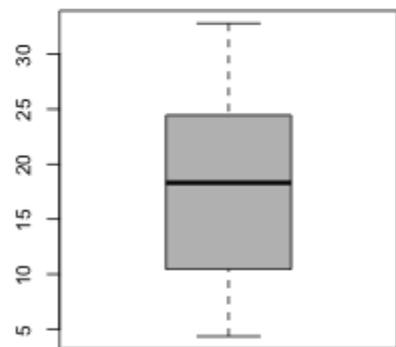
a



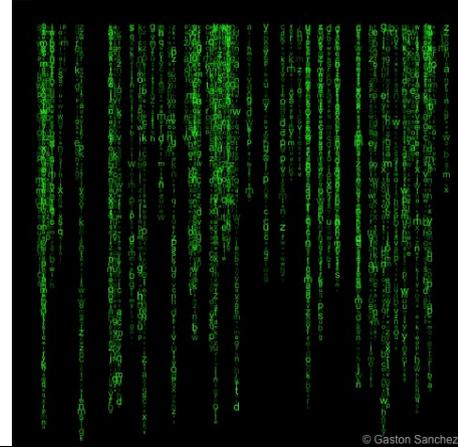
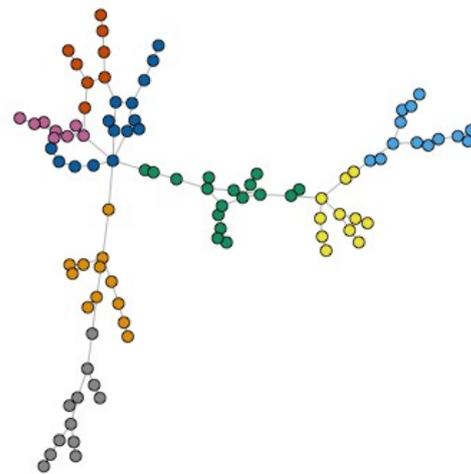
Index



a



a



# RStudio

The screenshot displays the RStudio interface with the following components:

- Menu Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for file operations (new, open, save, print) and a search bar for files/functions.
- Source Editor:** Contains three untitled files. The active file 'Untitled3\*' shows the following R code:

```
1 # AQUI VAI O SEU CÓDIGO
2
```
- Environment Pane:** Shows the 'Global Environment' with the following values:

Variable	Value
resultado	3
vetor	int [1:5] 1 2 3 4 5
- Console:** Shows the prompt `> SAÍDA|`.
- Help Pane:** Includes tabs for Files, Plots, Packages, Help, and Viewer. A search bar is present with the text 'Find in Topic'.

# RStudio

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains R code for generating a normal distribution and plotting a histogram.

```
1 # AQUI VAI O SEU CÓDIGO
2 variavel <- rnorm(n = 1000, mean = 0, sd = 1)
3 hist(variavel)
4
```
- Console:** Shows the execution of the code.

```
> variavel <- rnorm(n = 1000, mean = 0, sd = 1)
> hist(variavel)
>
```
- Environment:** Lists the variable `variavel` as a numeric vector of length 1000, with a preview of values: `num [1:1000] -0.00443 1.93699 -1.64245 0.59697 -0...`
- Plots:** A histogram titled "Histogram of variavel" is displayed. The x-axis is labeled "variavel" and ranges from -4 to 3. The y-axis is labeled "Frequency" and ranges from 0 to 200. The histogram shows a bell-shaped distribution centered at 0.

# Funcionamento básico

- comando(opções)
  - mean(variavel) #calcula a média do objeto “variavel”
  - sd(variavel) #calcula o desvio padrão do objeto “variavel”
- Resultado <- comando(opções)
  - Media <- mean(variavel) #guarda a média do objeto “variável” no objeto “Media”

# Comentando código

- Qualquer linha que comece com # será ignorada pelo R
- Em qualquer linha, o texto após um # será ignorado
- Sempre comente o código.

# Exemplo

- `variavel <- rnorm(n = 1000, mean = 0, sd = 1)`
- `hist(variavel)`
  
- Função **rnorm()** sorteia números provenientes de uma distribuição normal
- Função **hist()** cria o histograma de uma variável
- O código acima cria o histograma de uma variavel com 1.000 observações sorteadas de uma distribuição normal com média 0 e desvio-padrão 1.

# Classes de objetos no R

- O R tem inúmeros tipos de objetos e classes, mas a grosso modo podemos reduzir a um: **vetor**
  - Um vetor é uma **variável**. Ou seja, um monte de observações de alguma coisa. Uma coluna de uma tabela.
  - **Classes de vetor:**
    - Inteiro “integer”      Números inteiros: 1, 2, 3, etc
    - Numérico “numeric”      Números reais: 3.1415, 2.72,
    - Lógico “logical”      Verdadeiro/Falso: TRUE, FALSE
    - Texto “character”      Texto. Entre aspas. “Texto”
    - Fator “factor”      Tipo texto, mas com Categorias

# Classes de objetos no R

- Dá pra montar outros tipos de objetos combinando vetores
- **data.frame**: Uma tabela. Ou seja, um monte de **vetores** de quaisquer tipos
- O R tem um monte de outros tipos de objetos, mas pra gente isso aí já tá bom.
- Exemplos que não vamos precisar:
  - **matrix**: Uma matriz. Ou seja, um monte de **vetores do mesmo tipo**
  - **list**: Uma lista, um monte de objetos quaisquer (vetores, tabelas, matrizes, etc)
  - **Array**: Uma matriz de “n” dimensões
  - Saiba mais: <http://adv-r.had.co.nz/Data-structures.html>

# Exemplos

The screenshot displays the RStudio interface with several components highlighted by red circles:

- Code Editor:** Lines 5-9 of the script are circled, showing the creation and class verification of a vector named 'inteiro':

```
5 #cria um vetor com o nome "inteiro"  
6 inteiro <- 1:5  
7 #mostra o vetor  
8 inteiro  
9 #verifica a classe do vetor criado  
class(inteiro)
```
- Environment Pane:** The 'values' section is circled, showing the variable 'inteiro' with its value and class:

```
inteiro      int [1:5] 1 2 3 4 5
```
- Console:** The execution output is circled, showing the same information as the Environment pane:

```
> inteiro <- 1:5  
> inteiro  
[1] 1 2 3 4 5  
> class(inteiro)  
[1] "integer"  
>
```
- R Documentation:** The 'Vectors' page is open, showing the description and usage of the vector function.

**R Documentation: Vectors**

**Description**

vector produces a vector of the given length and mode.

as.vector, a generic, attempts to coerce its argument into a vector of mode mode (the default is to coerce to whichever vector mode is most convenient): if the result is atomic all attributes are removed.

is.vector returns TRUE if x is a vector of the specified mode having no attributes other than names. It returns FALSE otherwise.

**Usage**

```
vector(mode = "logical", length = 0)  
as.vector(x, mode = "any")  
is.vector(x, mode = "any")
```

**Ararguments**

# Exemplos

The screenshot displays the RStudio interface with three main panels: the editor, the console, and the help viewer.

**Editor Panel:** Contains R code for creating and classifying vectors. Red annotations highlight the code lines.

```
9 #verifica a classe do vetor criado
10 class(inteiro)
11
12 numerico <- rnorm(n = 5)
13 numerico
14 class(numerico)
15 logico <- c(TRUE, FALSE, TRUE, FALSE, TRUE)
16 logico
17 class(logico)
18 texto <- c('Branco', 'Azul', 'Branco', 'Azul', 'Azul')
19 texto
20 class(texto)
21 fator <- factor(texto)
22 fator
23 class(fator)
```

**Console Panel:** Shows the execution results of the code above. Red annotations highlight the output.

```
> numerico <- rnorm(n = 5)
> numerico
[1] -0.9273748 0.7647839 -0.4464639 0.2766782 -0.7841131
> class(numerico)
[1] "numeric"
> logico <- c(TRUE, FALSE, TRUE, FALSE, TRUE)
> logico
[1] TRUE FALSE TRUE FALSE TRUE
> class(logico)
[1] "logical"
> texto <- c('Branco', 'Azul', 'Branco', 'Azul', 'Azul')
> texto
[1] "Branco" "Azul" "Branco" "Azul" "Azul"
> class(texto)
[1] "character"
> fator <- factor(texto)
> fator
[1] Branco Azul Branco Azul Azul
Levels: Azul Branco
> class(fator)
[1] "factor"
>
```

**Environment Panel:** Shows the current environment with the following values:

Variable	Value
fator	Factor w/ 2 levels "Azul","Branco": 2 1 2 1 1
inteiro	int [1:5] 1 2 3 4 5
logico	logi [1:5] TRUE FALSE TRUE FALSE TRUE
numerico	num [1:5] -0.927 0.765 -0.446 0.277 -0.784
texto	chr [1:5] "Branco" "Azul" "Branco" "Azul" "Azul"

**Help Panel:** Displays the documentation for the `vector` function.

## Vectors

### Description

vector produces a vector of the given length and mode.

as.vector, a generic, attempts to coerce its argument into a vector of mode mode (the default is to coerce to whichever vector mode is most convenient): if the result is atomic all attributes are removed.

is.vector returns TRUE if x is a vector of the specified mode having no attributes other than names. It returns FALSE otherwise.

### Usage

```
vector(mode = "logical", length = 0)
as.vector(x, mode = "any")
is.vector(x, mode = "any")
```

### Arguments

# Exemplos

The screenshot displays the RStudio interface with three main panels: the editor, the console, and the help pane.

**Editor Panel:** Contains R code for creating a data frame. A red arrow points to the `class(inteiro)` line (line 10), and a red bracket highlights the `tabela` data frame creation code (lines 5-7).

```
5 #cria um vetor com o nome "inteiro"
6 inteiro <- 1:5
7 #mostra o vetor
8 inteiro
9 #verifica a classe do vetor criado
10 class(inteiro)
11
12 numerico <- rnorm(n = 5)
13 numerico
14 class(numerico)
15 logico <- c(TRUE, FALSE, TRUE, FALSE, TRUE)
16 logico
17 class(logico)
18 texto <- c('Branco', 'Azul', 'Branco', 'Azul', 'Azul')
19 texto
20 class(texto)
21 fator <- factor(texto)
22 fator
23 class(fator)
24
25 tabela <- data.frame(inteiro, numerico, logico, texto, fator)
26 tabela
27 class(tabela)
```

**Console Panel:** Shows the execution of the code. A red bracket highlights the output of the `tabela` command.

```
> tabela <- data.frame(inteiro, numerico, logico, texto, fator)
> tabela
  inteiro numerico logico texto fator
1      1 -0.9273748  TRUE Branco Branco
2      2  0.7647839  FALSE  Azul  Azul
3      3 -0.4464639  TRUE Branco Branco
4      4  0.2766782  FALSE  Azul  Azul
5      5 -0.7841131  TRUE  Azul  Azul
> class(tabela)
[1] "data.frame"
>
```

**Environment Panel:** Shows the `tabela` object with 5 observations and 5 variables.

Variable	Class	Values
fator	Factor w/ 2 levels "Azul","Branco":	2 1 2 1 1
inteiro	int [1:5]	1 2 3 4 5
logico	logi [1:5]	TRUE FALSE TRUE FALSE TRUE
numerico	num [1:5]	-0.927 0.765 -0.446 0.277 -0.784
texto	chr [1:5]	"Branco" "Azul" "Branco" "Azul" "Azul"

**Help Panel:** Displays the R documentation for `vector`.

## Vectors

### Description

vector produces a vector of the given length and mode.

`as.vector`, a generic, attempts to coerce its argument into a vector of mode `mode` (the default is to coerce to whichever vector mode is most convenient): if the result is atomic all attributes are removed.

`is.vector` returns `TRUE` if `x` is a vector of the specified mode having no attributes *other than names*. It returns `FALSE` otherwise.

### Usage

```
vector(mode = "logical", length = 0)
as.vector(x, mode = "any")
is.vector(x, mode = "any")
```

### Arguments

# Exemplos

The screenshot shows the RStudio interface with a data frame named 'tabela' displayed in the Environment pane. The data frame has 5 rows and 6 columns: 'inteiro', 'numerico', 'logico', 'texto', and 'fator'. The 'tabela' tab is highlighted with a red underline. The console shows the command used to create the data frame.

	inteiro	numerico	logico	texto	fator
1	1	-0.9273748	TRUE	Branco	Branco
2	2	0.7647839	FALSE	Azul	Azul
3	3	-0.4464639	TRUE	Branco	Branco
4	4	0.2766782	FALSE	Azul	Azul
5	5	-0.7841131	TRUE	Azul	Azul

```
> tabela <- data.frame(inteiro, numerico, logico, texto, fator)
```

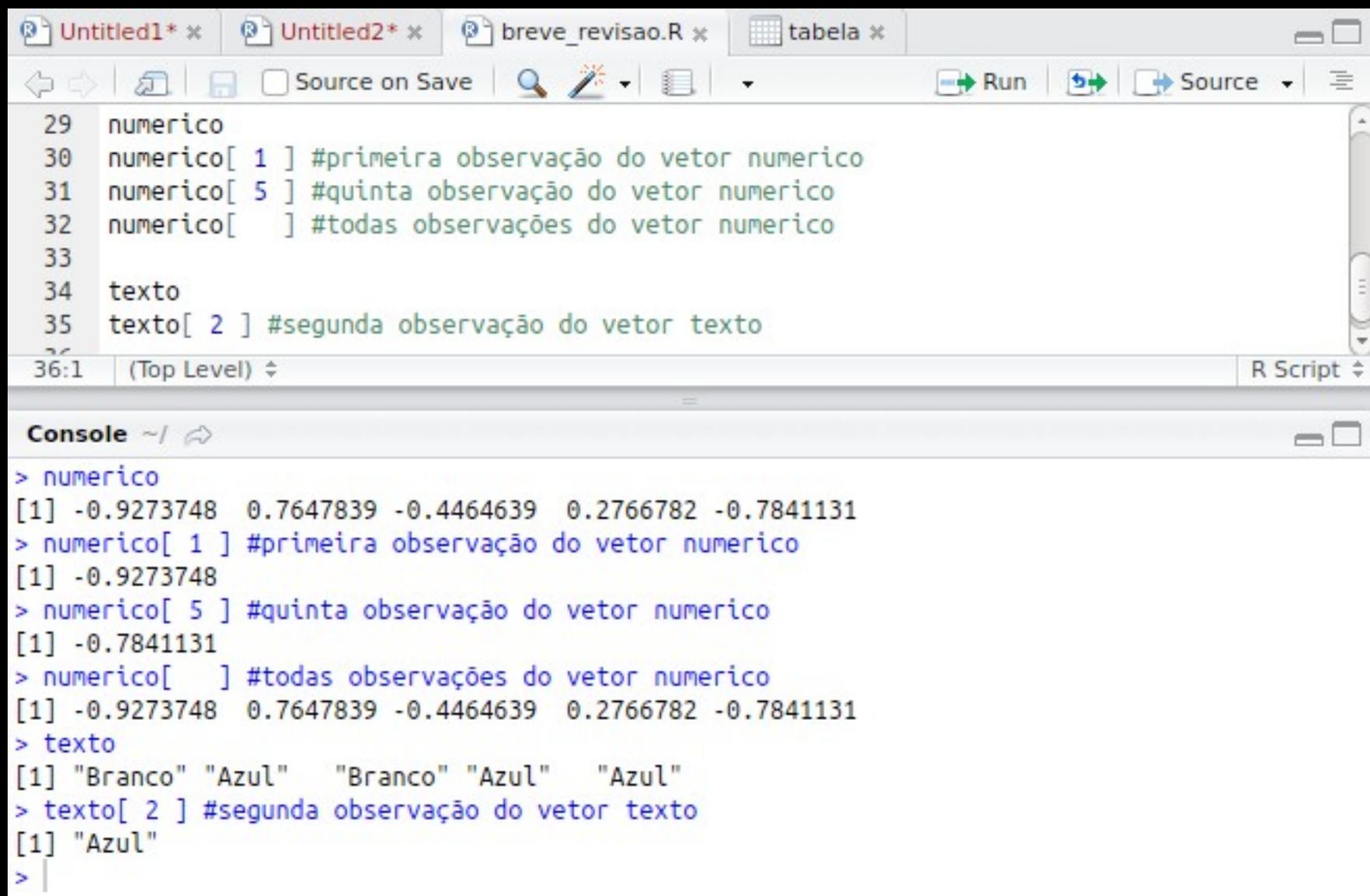
# ATENÇÃO

- Separador decimal no R é o ponto “.” não a vírgula:
  - 3.14159 correto
  - 3,14159 errado
- A vírgula é usada para separar coisas
  - 3.14159 (um número)
  - 3,14159 (dois números: 3 e 14159)

# Indexação

- Serve para encontrar observações específicas dentro de um objeto
- Sempre usa colchetes
- Exemplos:
  - Vetor[ 3 ] #terceira observação
  - Vetor[ 5 ] #quinta observação
  - Vetor[ ] #todas observações

# Exemplos



The image shows a screenshot of an R script editor window. The window title bar includes several open files: 'Untitled1\*' x, 'Untitled2\*' x, 'breve\_revisao.R' x, and 'tabela' x. The editor contains R code for creating and accessing vectors. The code is as follows:

```
29 numerico
30 numerico[ 1 ] #primeira observação do vetor numerico
31 numerico[ 5 ] #quinta observação do vetor numerico
32 numerico[   ] #todas observações do vetor numerico
33
34 texto
35 texto[ 2 ] #segunda observação do vetor texto
```

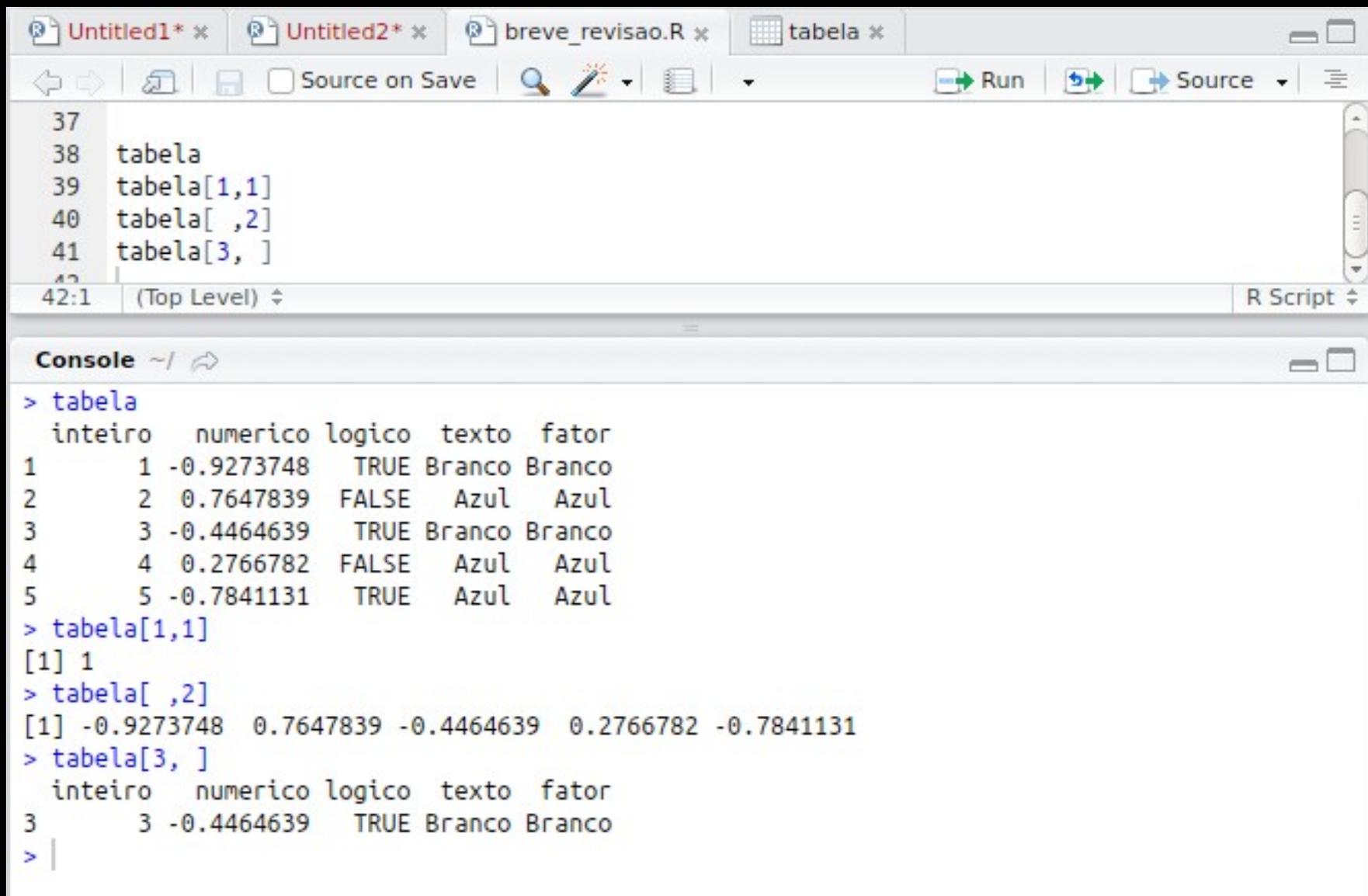
Below the editor is a console window showing the output of the code. The console output is as follows:

```
> numerico
[1] -0.9273748  0.7647839 -0.4464639  0.2766782 -0.7841131
> numerico[ 1 ] #primeira observação do vetor numerico
[1] -0.9273748
> numerico[ 5 ] #quinta observação do vetor numerico
[1] -0.7841131
> numerico[   ] #todas observações do vetor numerico
[1] -0.9273748  0.7647839 -0.4464639  0.2766782 -0.7841131
> texto
[1] "Branco" "Azul"  "Branco" "Azul"  "Azul"
> texto[ 2 ] #segunda observação do vetor texto
[1] "Azul"
> |
```

# Indexação

- Em um `data.frame`, precisamos indicar linha(s) e coluna(s)
- Exemplos:
  - `Tabela[ 1, 1 ]` #primeira linha, primeira coluna
  - `Tabela[ 3, 4 ]` #terceira linha da quarta coluna
  - `Tabela[ , 2 ]` #todas observações da segunda coluna
- A ordem SEMPRE é [ linha, coluna]
  - Obs: qualquer notação matemática é assim, inclusive

# Exemplos



The screenshot shows the R Studio interface. The top pane displays an R script with the following code:

```
37  
38 tabela  
39 tabela[1,1]  
40 tabela[ ,2]  
41 tabela[3, ]  
42  
42:1 (Top Level) ↕
```

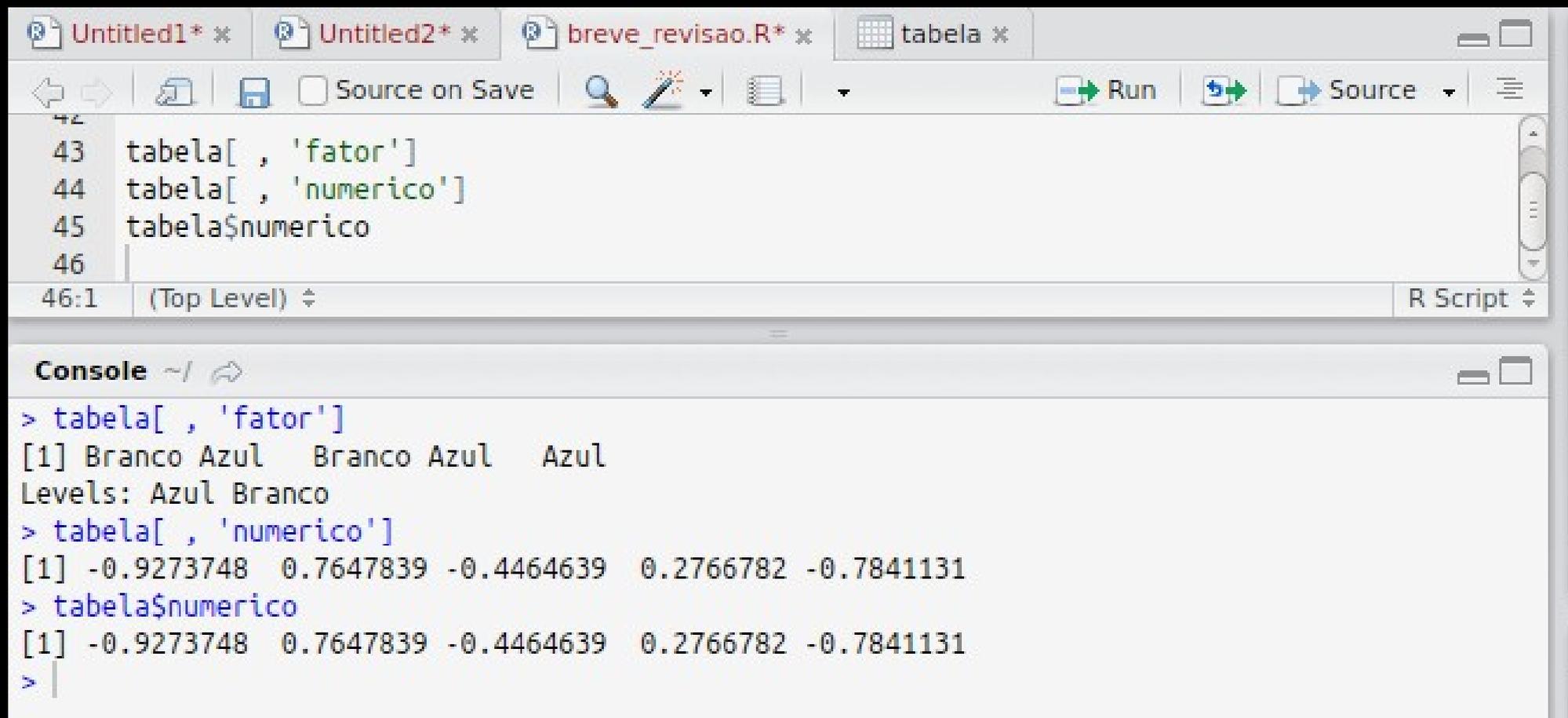
The bottom pane shows the R console output:

```
> tabela  
  inteiro  numerico logico  texto  fator  
1      1 -0.9273748  TRUE Branco Branco  
2      2  0.7647839 FALSE  Azul  Azul  
3      3 -0.4464639  TRUE Branco Branco  
4      4  0.2766782 FALSE  Azul  Azul  
5      5 -0.7841131  TRUE  Azul  Azul  
> tabela[1,1]  
[1] 1  
> tabela[ ,2]  
[1] -0.9273748  0.7647839 -0.4464639  0.2766782 -0.7841131  
> tabela[3, ]  
  inteiro  numerico logico  texto  fator  
3      3 -0.4464639  TRUE Branco Branco  
> |
```

# Indexação

- Também dá para acessar o vetor de um `data.frame` utilizando seu nome, ou `$` entre o nome da tabela e o nome do vetor dentro dela.
- Exemplos:
  - `Tabela[ , 'texto']`
  - `Tabela$texto`

# Exemplos



The image shows a screenshot of the RStudio interface. The top pane displays the source editor with the following R code:

```
43 tabela[ , 'fator']
44 tabela[ , 'numerico']
45 tabela$numerico
46
```

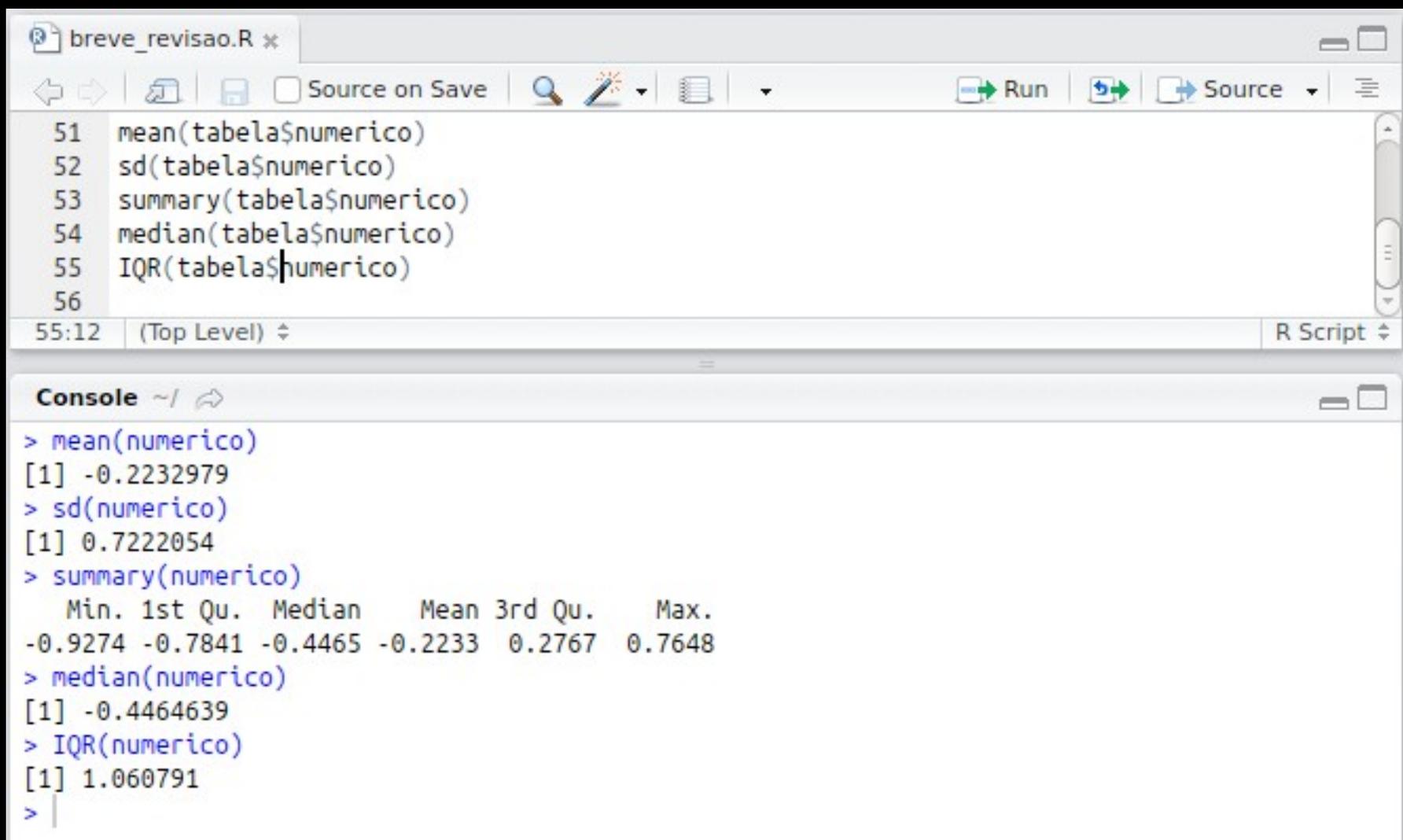
The bottom pane shows the console output for the executed code:

```
> tabela[ , 'fator']
[1] Branco Azul Branco Azul Azul
Levels: Azul Branco
> tabela[ , 'numerico']
[1] -0.9273748 0.7647839 -0.4464639 0.2766782 -0.7841131
> tabela$numerico
[1] -0.9273748 0.7647839 -0.4464639 0.2766782 -0.7841131
>
```

# Estatística Descritiva no R

- `mean()` - calcula a média de um vetor
- `sd()` - calcula o desvio-padrão de um vetor
- `summary()` - calcula várias coisas
- `median()` - calcula a mediana
- `IQR()` - calcula o intervalo inter-quartil

# Exemplos



The image shows a screenshot of the R Studio interface. The top pane displays a script file named 'breve\_revisao.R' with the following code:

```
51 mean(tabela$numerico)
52 sd(tabela$numerico)
53 summary(tabela$numerico)
54 median(tabela$numerico)
55 IQR(tabela$numerico)
56
```

The bottom pane shows the console output for these commands:

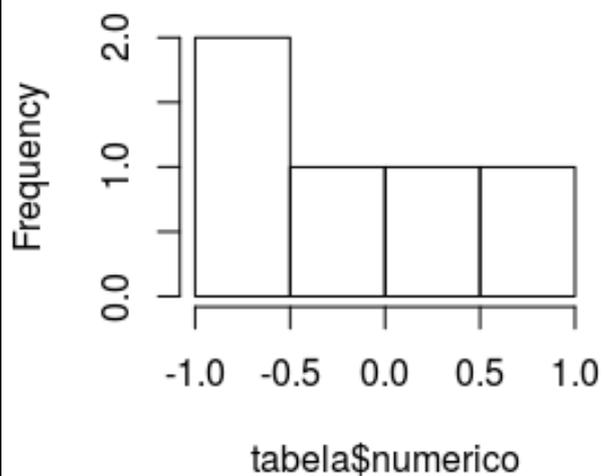
```
> mean(numerico)
[1] -0.2232979
> sd(numerico)
[1] 0.7222054
> summary(numerico)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.9274 -0.7841 -0.4465 -0.2233  0.2767  0.7648
> median(numerico)
[1] -0.4464639
> IQR(numerico)
[1] 1.060791
> |
```

# Gráficos no R

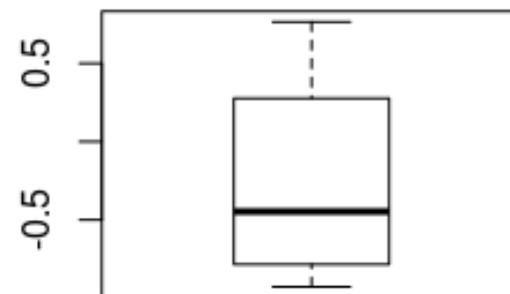
- Histograma: `hist()`
- Boxplot: `boxplot()`
- Dispersão: `plot()`

# Exemplos

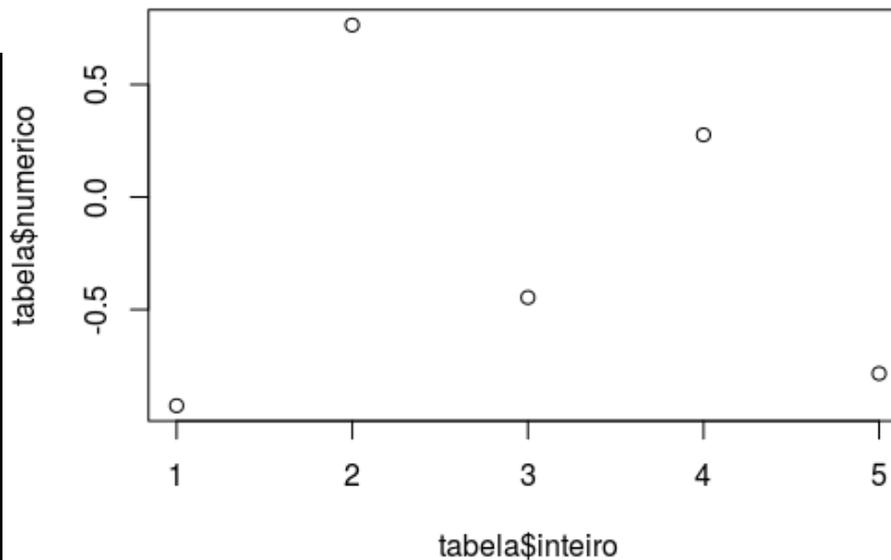
**hist(tabela\$numerico)**



**boxplot(tabela\$numerico)**



**plot(x = tabela\$inteiro, y = tabela\$numerico)**



# Importando dados

- Melhor maneira é exportar os dados do excel em um arquivo **CSV**.
  - CSV = “Comma Separated Values”, ou “valores separados por vírgulas”.
- Utilizar as funções `read.csv()` e `read.csv2()`
- Exemplo:
  - `Tabela <- read.csv('meus_dados.csv')`

# Working directory

- Também conhecido como “diretório de trabalho” ou “pasta de trabalho”
- O R sempre está trabalhando em alguma pasta específica do seu computador
- O ideal é você ter uma pasta por projeto.
- Quando o R lê e salva arquivos (importar dados, salvar gráficos), ele usa a pasta de trabalho.

# Working directory

- Para saber a pasta atual: `getwd()`
- Para mudar a pasta atual: `setwd()`
  - No Rstudio é mais fácil:
    - Session → Set working directory



**Hadley Wickham** ✓

@hadleywickham

Follow



Setting your working directory = picking the project to work on. That's something you do & shouldn't be done by code [#rstats](#)

Dúvidas?