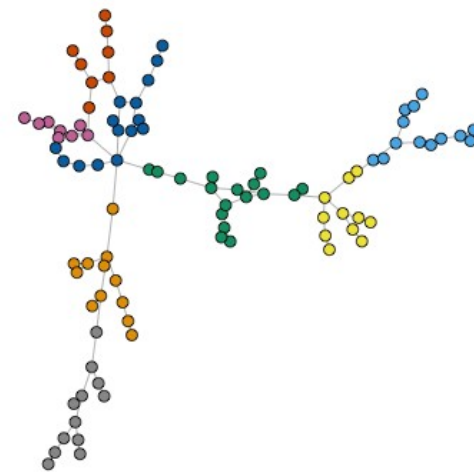
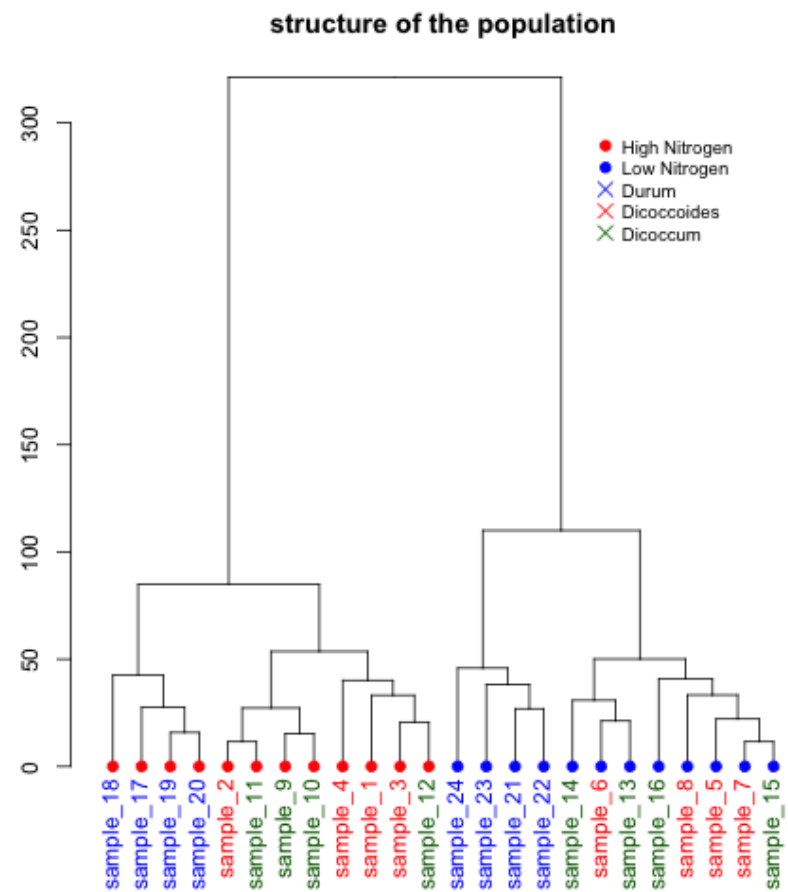
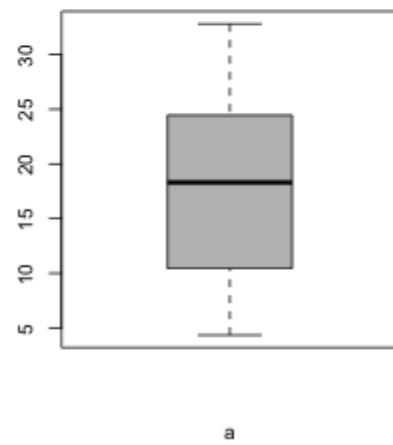
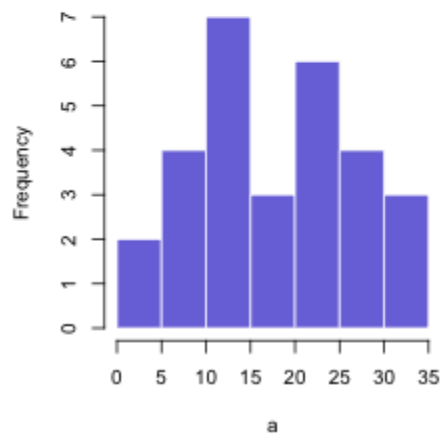
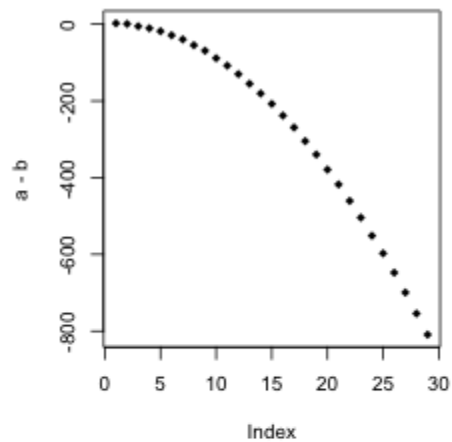
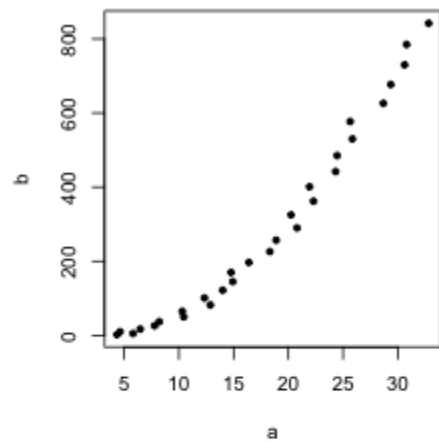
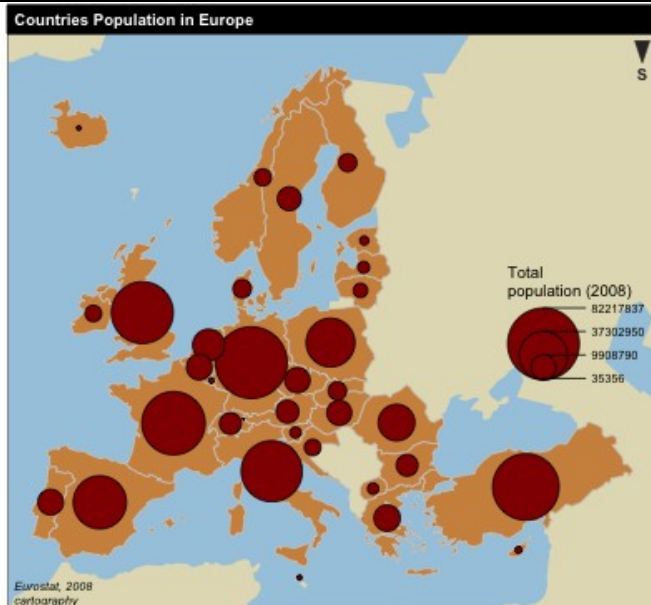


VPS 5712

Revisão do R

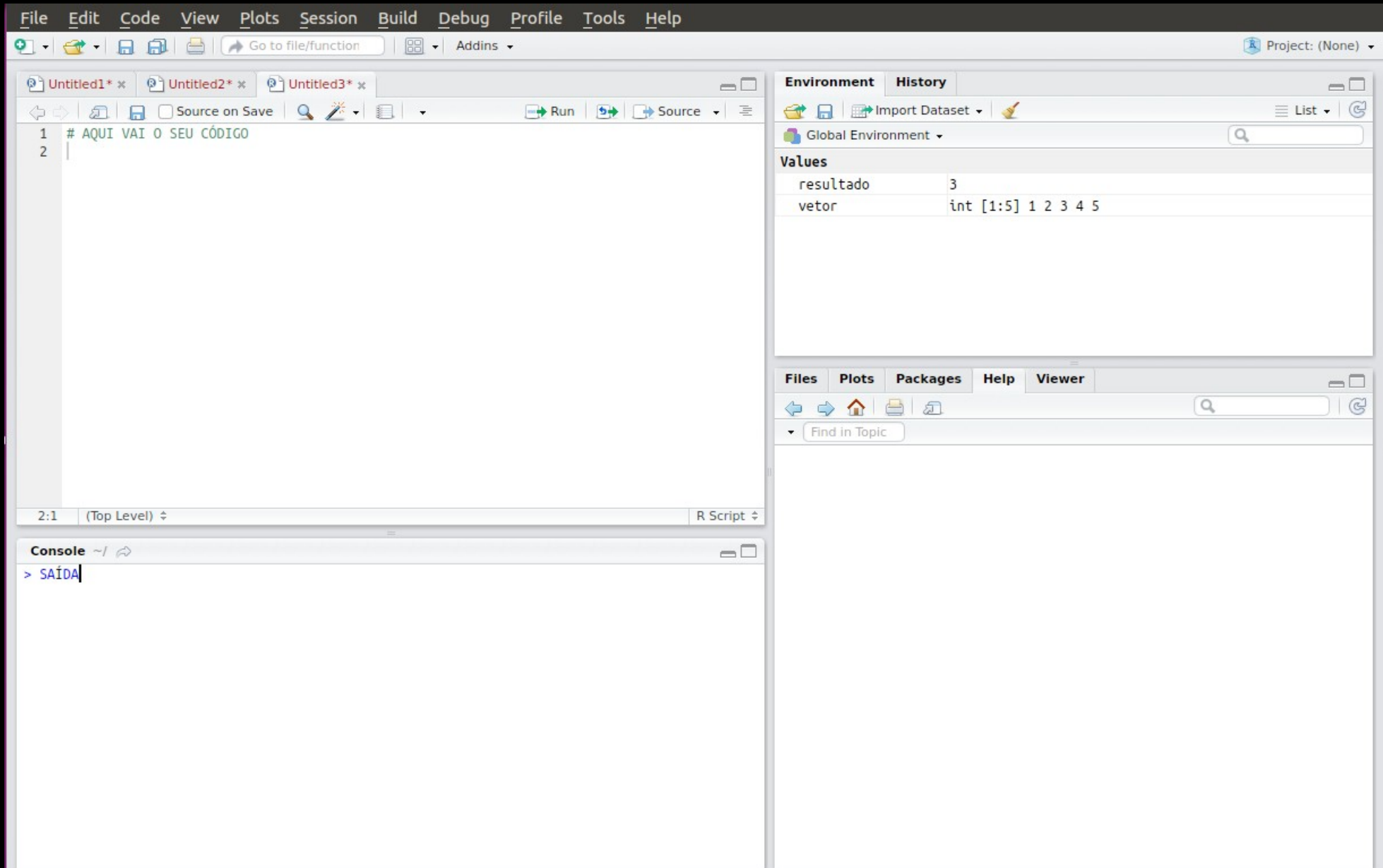
R

- Por que R?

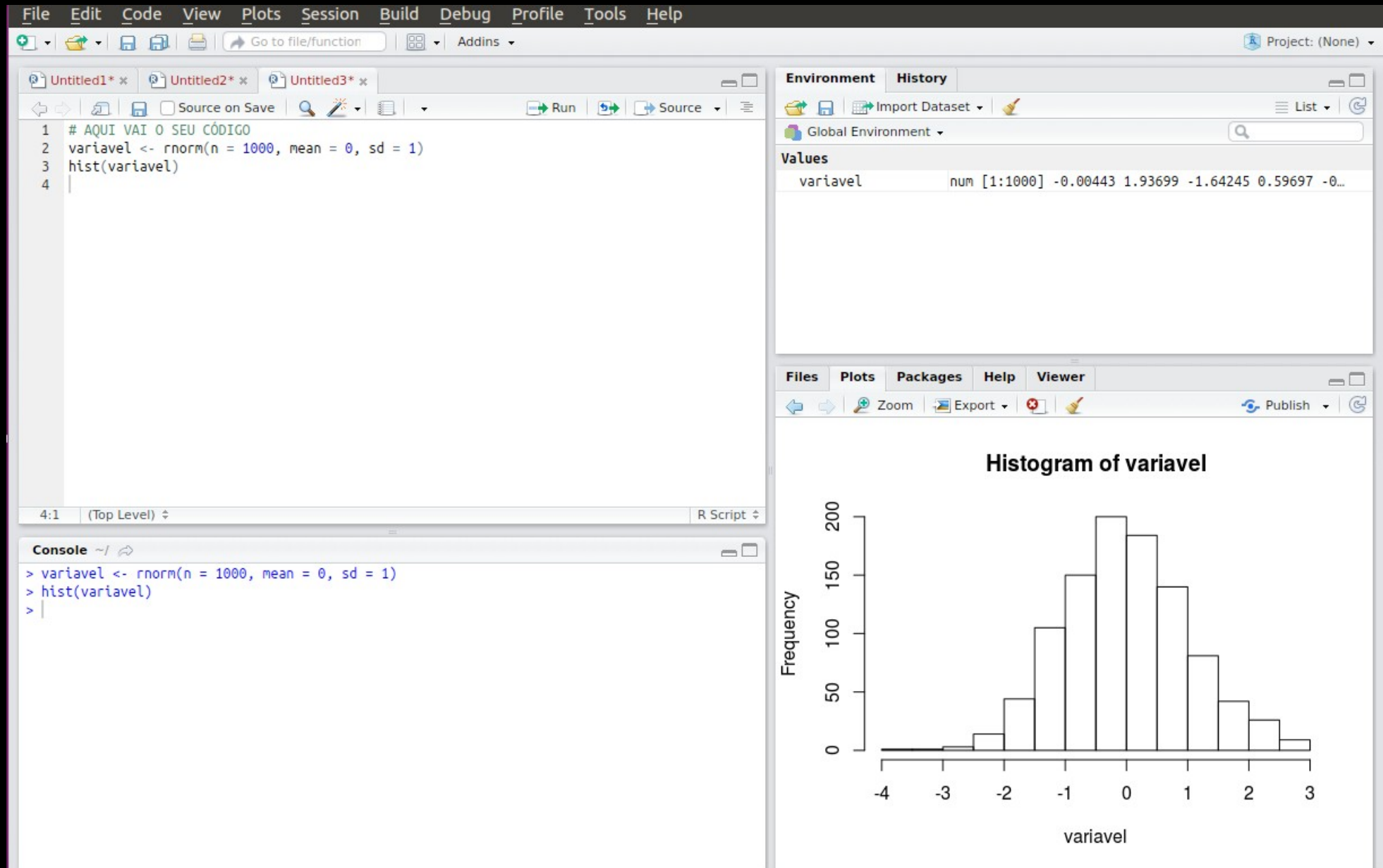


1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

RStudio



RStudio



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 three main components highlighted by red circles:

- Source Editor:** Contains R code for creating and inspecting a vector. The code is as follows:

```
1 # AQUI VAI O SEU CÓDIGO
2 variavel <- rnorm(n = 1000, mean = 0, sd = 1)
3 test(variavel)

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)

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)
```
- Environment Pane:** Shows the current environment with a table of values:

Variable	Value
inteiro	int [1:5] 1 2 3 4 5
- Console:** Shows the output of the commands entered in the source editor:

```
> inteiro <- 1:5
> inteiro
[1] 1 2 3 4 5
> class(inteiro)
[1] "integer"
>
```

On the right side, the R Documentation for **Vectors** is visible, including the following sections:

- 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 panes. Red handwritten annotations highlight specific areas: a bracket on the left side of the code editor, a bracket on the right side of the environment pane, and a large bracket at the bottom of the console pane.

Code Editor (Untitled1* x, Untitled2* x, Untitled3* x):

```
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
25:1 (Top Level) ↕
```

Environment (Global Environment):

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"

Console:

```
> 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 Pane (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")
```

Arguments

Exemplos

The screenshot displays the RStudio interface with three main panes: Source, Environment, and Console.

Source Pane: Contains R code for creating a data frame. A red arrow points to the `class(tabela)` line (line 23).

```
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)
```

Environment Pane: Shows the 'Global Environment' with a table of variables. A red arrow points to the 'tabela' variable.

Variable	Value
tabela	5 obs. of 5 variables

Console Pane: Shows the output of the R code. A red bracket highlights the output of `class(tabela)`.

```
> 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 Pane (Data Table):

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"

Console Pane (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.

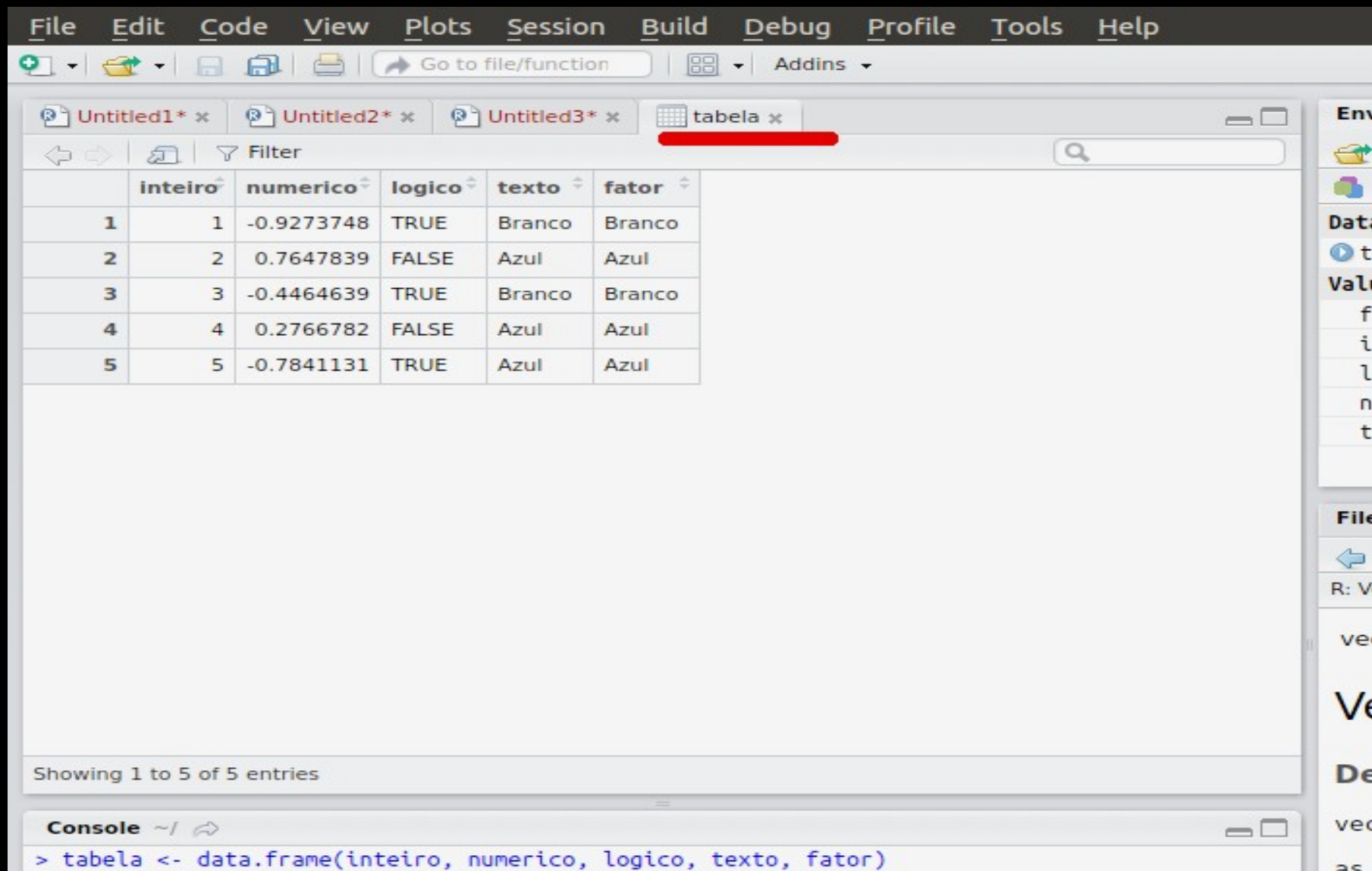
is.vector returns TRUE if x is a vector of the specified mode having no attributes other than names. It returns FALSE otherwise.

Usage

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

Arguments

Exemplos



The screenshot displays the RStudio environment with a data frame named 'tabela' open in the 'Environment' pane. The data frame contains 5 rows and 6 columns: 'inteiro', 'numerico', 'logico', 'texto', and 'fator'. The data is as follows:

	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

The console at the bottom shows the command used to create the data frame:

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

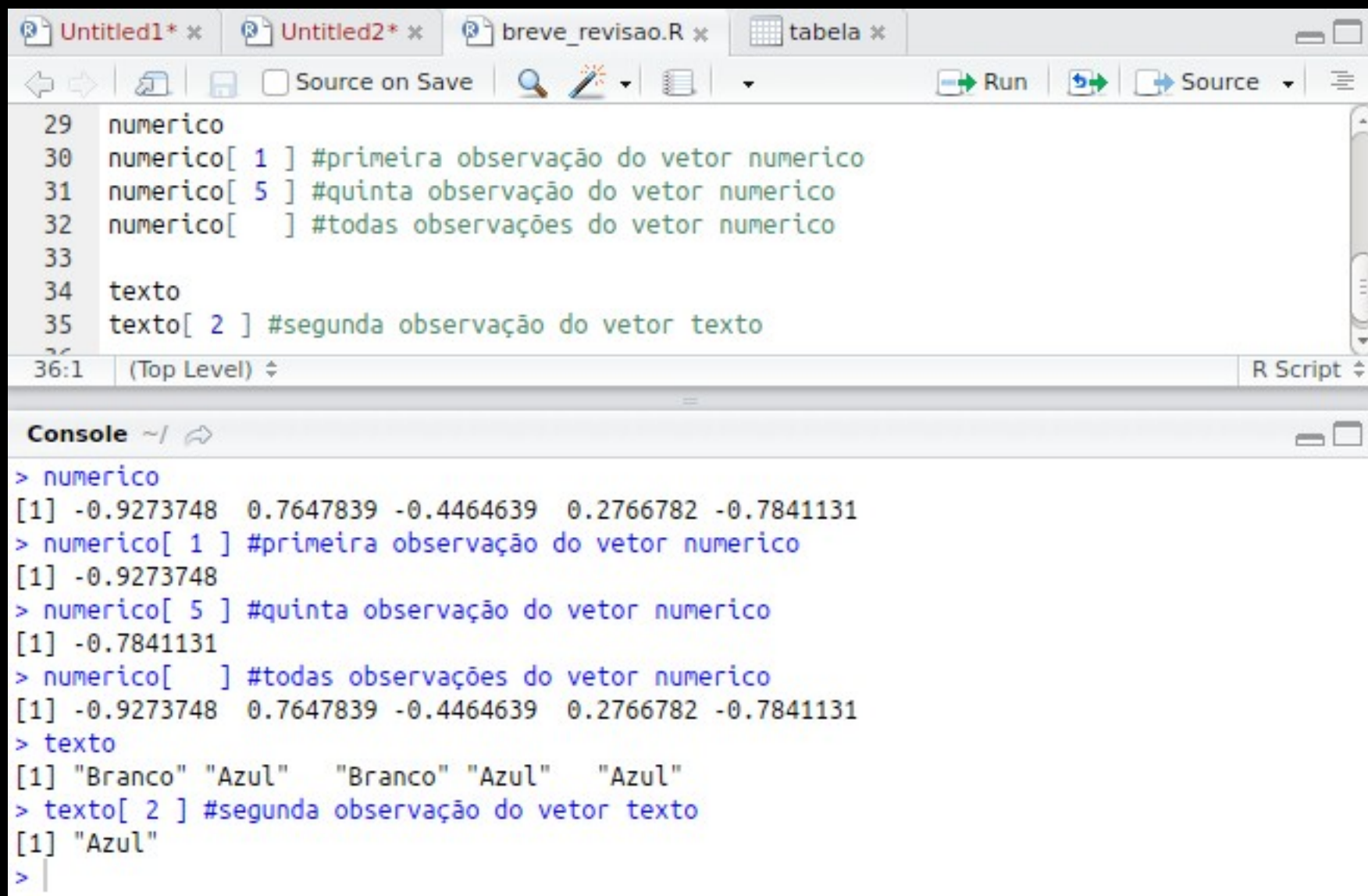
ATENÇÃO

- 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 screenshot displays the R Studio environment. The top pane shows a script with R code for creating and accessing vectors. The bottom pane shows the console output for these commands.

```
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
```

36:1 (Top Level) ↕ R Script

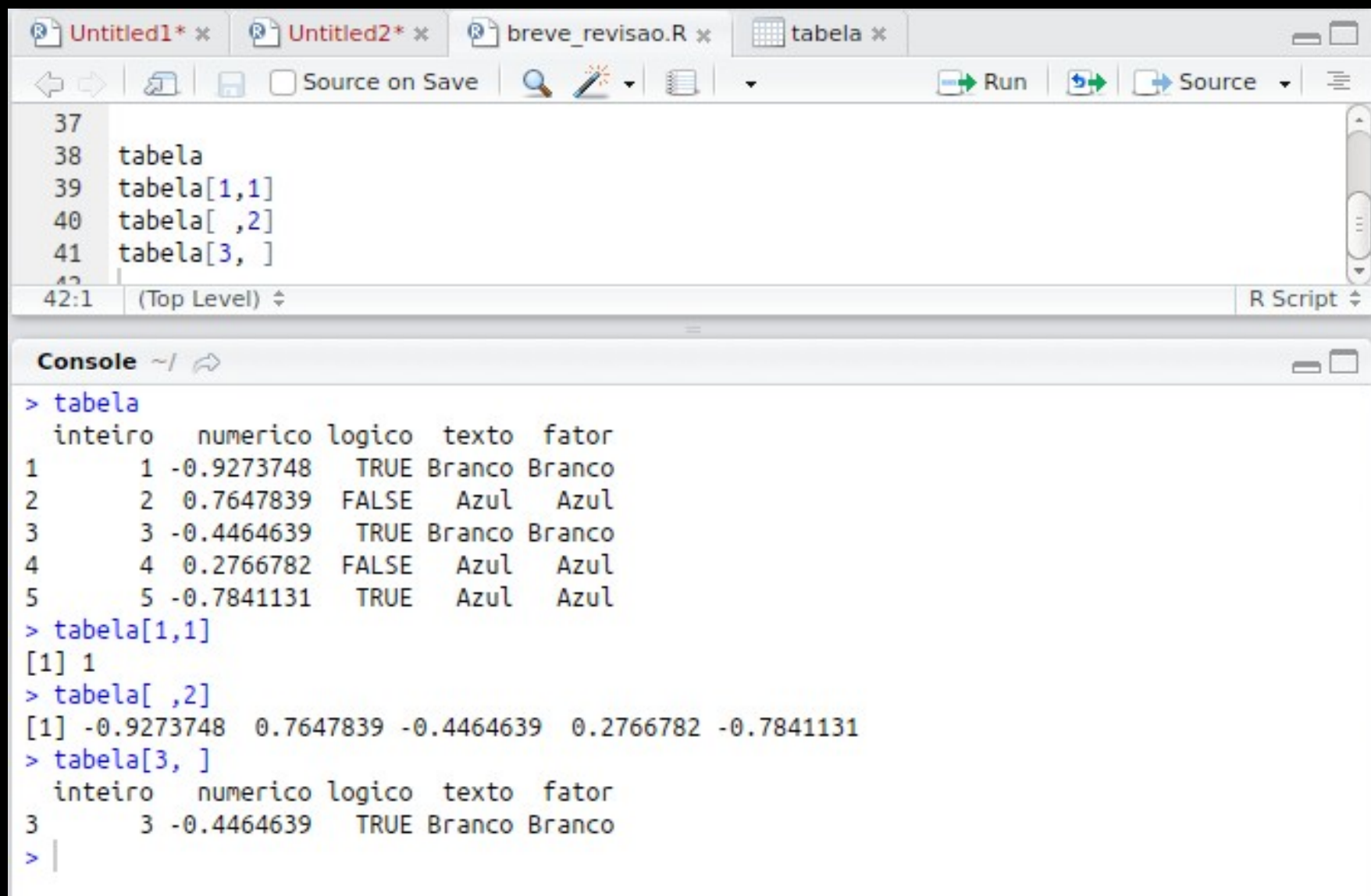
Console ~/ ↗

```
> 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 RStudio interface with four open files: 'Untitled1*', 'Untitled2*', 'breve_revisao.R', and 'tabela *'. The 'tabela *' file is active, showing lines 37 to 42. The console displays the output of the R script, including the structure of the 'tabela' data frame and the results of subsetting operations.

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

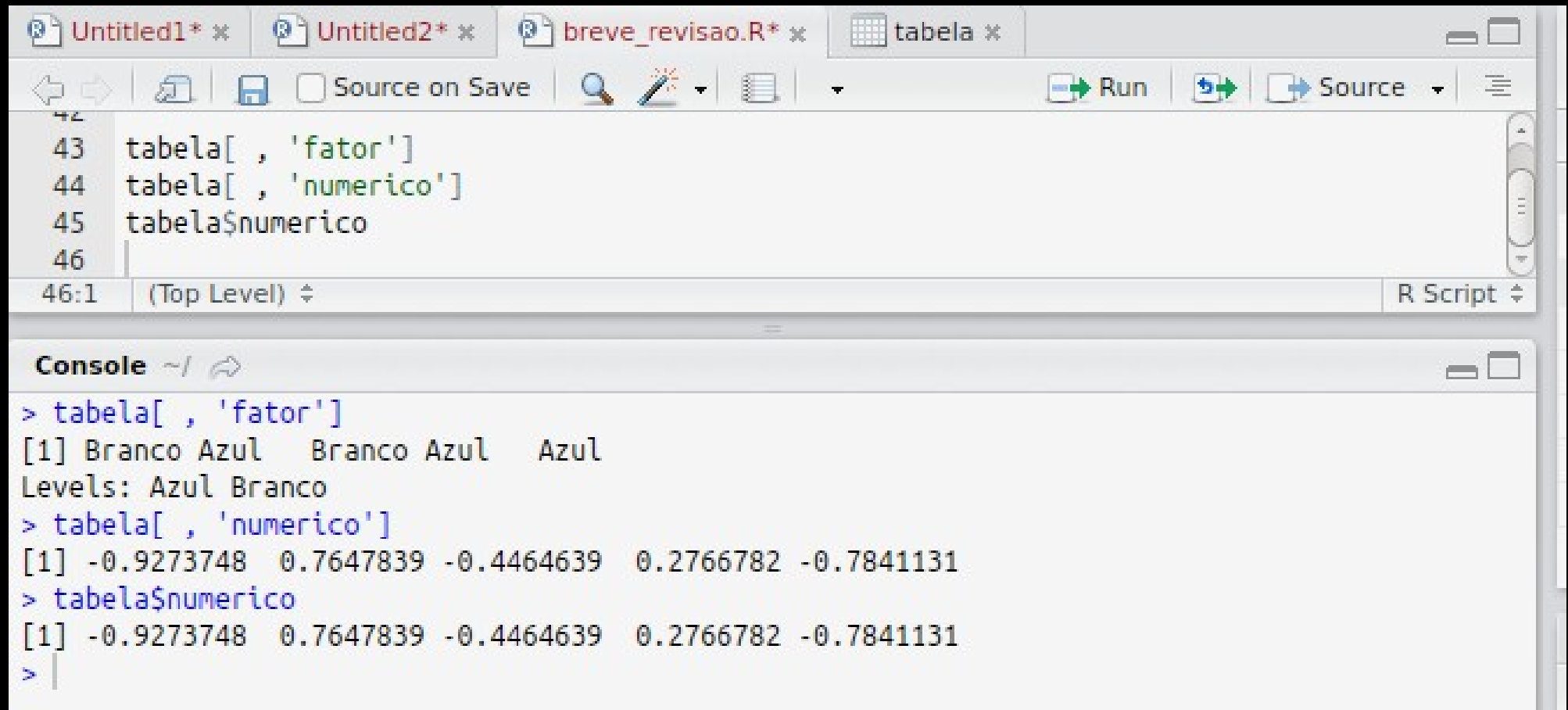
Console ~/

```
> 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 screenshot displays the RStudio environment. The top pane shows the R script editor with the following code:

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

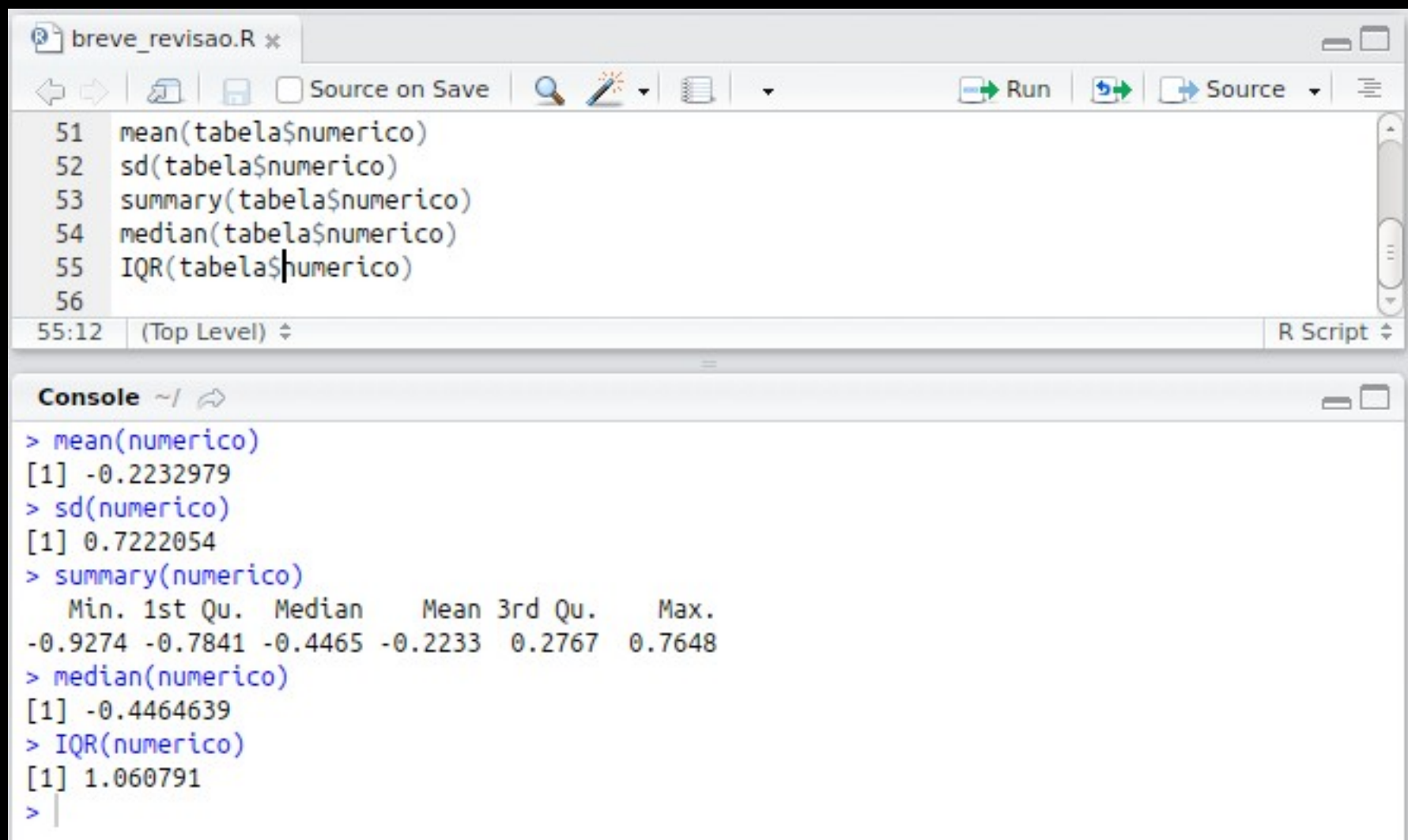
The bottom pane shows the R console with the following output:

```
> 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 is the source editor, showing a file named 'breve_revisao.R'. It contains five lines of R code: `mean(tabela$numerico)`, `sd(tabela$numerico)`, `summary(tabela$numerico)`, `median(tabela$numerico)`, and `IQR(tabela$numerico)`. The bottom pane is the console, which shows the output of these commands. The output for `summary(tabela$numerico)` is a summary table with columns for Min., 1st Qu., Median, Mean, 3rd Qu., and Max.

```
breve_revisao.R x
Source on Save
Run Source
51 mean(tabela$numerico)
52 sd(tabela$numerico)
53 summary(tabela$numerico)
54 median(tabela$numerico)
55 IQR(tabela$numerico)
56
55:12 (Top Level) R Script

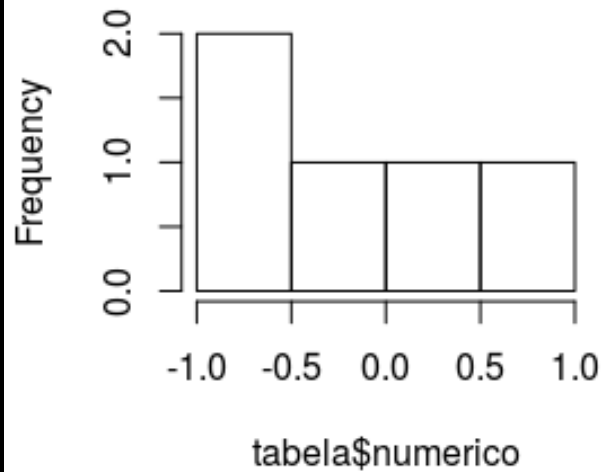
Console ~/
> mean(numerico)
[1] -0.2232979
> sd(numerico)
[1] 0.7222054
> summary(numerico)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.9274 -0.7841 -0.4465 -0.2233  0.2767  0.7648
> median(numerico)
[1] -0.4464639
> IQR(numerico)
[1] 1.060791
>
```

Gráficos no R

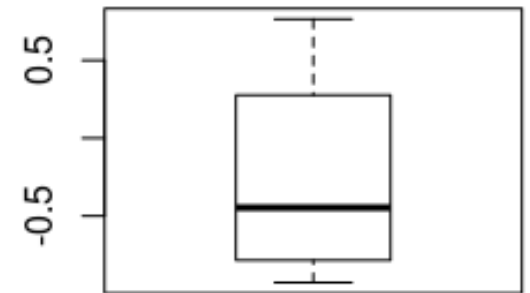
- Histograma: `hist()`
- Boxplot: `boxplot()`
- Dispersão: `plot()`

Exemplos

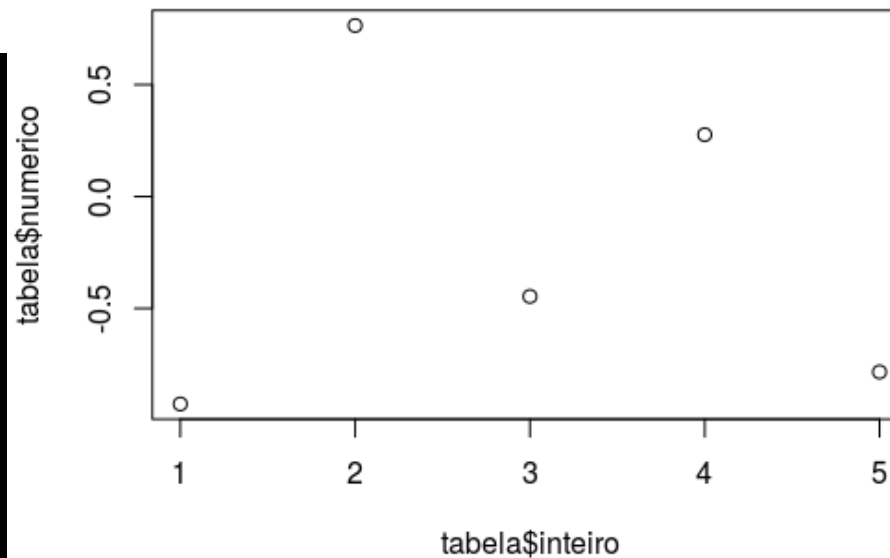
hist(tabela\$numerico)



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?