

Noções Básicas de R - Aula 3

Exemplo de uso de R + Markdown + knitr

Prof. Dr. Cleuler Barbosa das Neves
currículo.lattes

AULA N. 03 - OBJETOS: VETORES, MATRIZES, DATA.FRAME, ARRAYS, LIST, DATE, TS etc.

R é uma Linguagem funcional orientada para objetos!

[=====

[Faz uso de funções & de suas composições !!!

[Armazena&Manipula objetos previamente criados!!!

[*Aply* essas composições nesses *ob-jectos* !!!

[Há *symbols* c/significados operacionais *tipics*!!!

[CRAN c/centenas de milhares de *functions* em *packages*!!!

[=====

```
# As "duas" primeiras linhas de comando de um script em R (p. 13) deve ser:
```

```
# A 1ª Linha de comando:
```

```
# O símbolo ~ representa a abreviatura para o caminho da pasta pessoal (Linux e Windows)
```

```
#setwd("~/") # Aponta o Diretório de Trabalho para a Pasta Pessoal e subpasta em que se encontra o arquivo
```

```
# Esse comando exibe a seguinte mensagem de alerta importante: "The working directory was changed to C:/"
```

```
getwd() # Exibe o Diretório de Trabalho, no caso o da Pasta Pessoal, executando uma linha de comando
```

```
## [1] "C:/Users/cleuler-bn/Documents/R_CS/Aula3"
```

```
setwd("~/../Documents/R_CS/Aula3") # Produz o mesmo efeito do código anterior
```

```
getwd()
```

```
## [1] "C:/Users/cleuler-bn/Documents/R_CS/Aula3"
```

```
# A 2ª Linha de comando: é um exemplo do uso de **funções compostas** em Linguagem **R**  
rm(list=ls()) # Remove toda a list de variáveis da Job Area, i. e., da Environment
```

```
#[=====]
```

```
#[ Pacotes do System Library ]
```

```
#[=====]
```

```

#Pacotes de importação de BD
#para ativar um pacote do System Library (vem c/a instalação do R): 2.000 f's
library(foreign) # argumento não precisa das aspas
# Para carregar Base de Dados dos aplicativos:
# Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase ...

#[=====]
#[          Pacotes da User Library          ]
#[=====]

#P/instalar um pacote da web (CRAN) basta executar install.packages() 1 vez
#install.packages("data.table") # Para carregar BD's de grandes dimensões
library(data.table) # (p.53-53 do livro R_CS); argumento não precisa das aspas
# 1- converter o arquivo para .csv usando a função fwf2csv () do pacote descr
# 2- carregar o BD com a função fread() do pacote data.table, que usa menos
#     memória que a função read.fwf() do pacote ...
#install.packages("sqldf") # p/carregar partes de BD's de grandes dimensões
library(sqldf) # R_SC: (p. 53-54)

## Loading required package: gsubfn
## Loading required package: proto
## Loading required package: RSQLite

#install.packages("descr")#Um pacote tem de ser instalado 1 vez no seu micro
library(descr) # Ativado o pacote, suas funções são disponibilizadas p/uso
# "descr" é um pacote com funções voltadas para Estatística Descritiva

#install.packages("gdata")
library(gdata) # pacote para manipulação de dados (BD's) (p. 45)

## Warning: execução do comando '"C:\Users\CLEULE~1\AppData\Local
## \LYX2~1.2\Perl\bin\perl.exe" "C:/Users/cleuler-bn/Documents/R/win-library/
## 3.4/gdata/perl/supportedFormats.pl"' teve status 2

## gdata: Unable to load perl libraries needed by read.xls()
## gdata: to support 'XLX' (Excel 97-2004) files.
##

## gdata: Unable to load perl libraries needed by read.xls()
## gdata: to support 'XLSX' (Excel 2007+) files.
##

## gdata: Run the function 'installXLSXsupport()'
## gdata: to automatically download and install the perl
## gdata: libraries needed to support Excel XLS and XLSX formats.
##

## Attaching package: 'gdata'

## The following objects are masked from 'package:data.table':
##
##     first, last

## The following object is masked from 'package:stats':
##

```

```

##      nobs
## The following object is masked from 'package:utils':
##
##      object.size
## The following object is masked from 'package:base':
##
##      startsWith

      # No Windows poderá ser necessário instalar ActivePerl
      # ou outro interpretador da linguagem perl.

#install.packages("igraph") # Océu é o limite!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
library(igraph) # pacote para Network Analysis and Visualization

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##      decompose, spectrum
## The following object is masked from 'package:base':
##
##      union

      # R_CS: cap. 12- Análise de Redes Sociais (com grafos)

#install.packages("knitr")
library(knitr) # pacote para geração de Relatório Dinâmico em R (p. 119)

#install.packages("memisc") # para surveys
library(memisc) # pacote para manipulação de pesquisa de dados (p. 66, 89)

## Loading required package: lattice
## Loading required package: MASS

##
## Attaching package: 'memisc'

## The following objects are masked from 'package:stats':
##
##      contr.sum, contr.treatment, contrasts
## The following object is masked from 'package:base':
##
##      as.array

      # e para apresentação de análises de seus resultados

#install.packages("rgdal") # para exibição de Mapas e dados espacializados
library(rgdal) # R_SC: cap. 11- Mapas (p. 134-139)

## Loading required package: sp

## rgdal: version: 1.2-18, (SVN revision 718)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
## Path to GDAL shared files: C:/Users/cleuler-bn/Documents/R/win-library/3.4/rgdal/gdal

```

```

## GDAL binary built with GEOS: TRUE
## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
## Path to PROJ.4 shared files: C:/Users/cleuler-bn/Documents/R/win-library/3.4/rgdal/proj
## Linking to sp version: 1.2-7

# Requer a instalação do pacote sp
#install.packages("sp")
library(sp)

#install.packages("rmarkdown") # para instalação do RMarkdown
library(rmarkdown) #R_SC: geração Relatórios Dinâmicos no RStudio(p. 115-19)
# Requer instalação de outros pacotes p/rodar o RMarkdown dentro do RStudio
#install.packages("htmltools") - esse não precisou, veio c/o RMarkdown
library(htmltools) # Ferramentas para HTML

##
## Attaching package: 'htmltools'

## The following object is masked from 'package:memisc':
##
##      css

#install.packages("caTools")# - esse precisou e instalou o bitops
library(caTools)#Tools: moving windows statistics, GIF, Base64, ROC AUC etc.

#install.packages(c("bindr", "bindrcpp", "Rcpp", "stringi")) é uma função composta
library(bindr)# library deve ter package com comprimento 1
library(bindrcpp)#
library(Rcpp)#
library(stringi)#
#install.packages(c("cluster", "Matrix"), lib="C:/Users/cleuler-bn/Documents/R/R-3.4.4/library")
library(cluster)#
library(Matrix)#

#install.packages(c("financial", "FinancialInstrument", "FinancialMath"))
library(financial)#
library(FinancialInstrument)#

## Loading required package: quantmod
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
##
## Attaching package: 'xts'

## The following objects are masked from 'package:gdata':
##
##      first, last

## The following objects are masked from 'package:data.table':

```

```
##
## first, last
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
library(FinancialMath)#

##
## Attaching package: 'FinancialMath'
## The following object is masked from 'package:FinancialInstrument':
##
## bond

#install.packages("tinytex")# - foi preciso instalar para gerar arquivo .pdf direto do RMarkdown
#library(tinytex)# para carregar o pacote tinytex, que gera arquivo .tex e certamente converte para .pdf
# Mas isso gerou uma v2.pdf no formato de uma janela do PDF, sem os marcadores do lado
# Do Jeito antigo estava melhor e gravava um .pdf na pasta R_CS/Aula1 que ao abrir no
# apresentou na parte esquerda da tela do Adobe todos os marcadores das secções do arquivo

#install.packages(c("lattice")) # é uma função composta
library(grid)# library deve ter package com comprimento 1
library(lattice)#

# Um *look* na sua **Estação de Trabalho** desta sessão do **R** versão 3.4.3
sessionInfo()

## R version 3.4.4 (2018-03-15)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 15063)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Portuguese_Brazil.1252 LC_CTYPE=Portuguese_Brazil.1252
## [3] LC_MONETARY=Portuguese_Brazil.1252 LC_NUMERIC=C
## [5] LC_TIME=Portuguese_Brazil.1252
##
## attached base packages:
## [1] grid stats graphics grDevices utils datasets methods
## [8] base
##
## other attached packages:
## [1] FinancialMath_0.1.1 FinancialInstrument_1.3.1
## [3] quantmod_0.4-12 TTR_0.23-3
## [5] xts_0.10-2 zoo_1.8-1
## [7] financial_0.2 Matrix_1.2-13
## [9] cluster_2.0.7 stringi_1.1.7
## [11] Rcpp_0.12.16 bindrcpp_0.2.2
## [13] bindr_0.1.1 caTools_1.17.1
## [15] htmltools_0.3.6 rmarkdown_1.9
## [17] rgdal_1.2-18 sp_1.2-7
## [19] memisc_0.99.14.9 MASS_7.3-49
## [21] lattice_0.20-35 knitr_1.20
## [23] igraph_1.2.1 gdata_2.18.0
```

```
## [25] descr_1.1.4          sqldf_0.4-11
## [27] RSQLite_2.1.0        gsubfn_0.7
## [29] proto_1.0.0          data.table_1.10.4-3
## [31] foreign_0.8-69
##
## loaded via a namespace (and not attached):
## [1] compiler_3.4.4 bitops_1.0-6 tools_3.4.4 digest_0.6.15
## [5] bit_1.1-12 evaluate_0.10.1 memoise_1.1.0 pkgconfig_2.0.1
## [9] DBI_0.8 curl_3.2 yaml_2.1.18 repr_0.12.0
## [13] stringr_1.3.0 gtools_3.5.0 rprojroot_1.3-2 bit64_0.9-7
## [17] tcltk_3.4.4 blob_1.1.1 magrittr_1.5 backports_1.1.2
## [21] xtable_1.8-2 chron_2.3-52

# Os interessados em assinar a *Lista Brasileira do R* -- [R-br] da **UFPR** devem [acessar](http://lis

#[=====]
#[
#[=====]
```

GERANDO UMA BD - AS ALTURAS E PESOS DA TURMA

Criando vetores: um tipo

é um fundamental do R; é uma estrutura de dados que permite armazenar um conjunto de valores de um mesmo tipo sob um mesmo nome de .

Seus principais tipos são:

:

O valor NA pode ser armazenado como valor NULL em qualquer um desses tipos. A função `vector()` cria um vetor do R.

Seus *argumentos* são: *mode* (modo ou *storage mode of an Object*) e *length* (comprimento do).

Exemplos de uso dessa função.

Criando vetores vazios de vários tipos básicos e de um tipo especial

```
#getwd()
#setwd("~/../R_CS/Aula3")

x <- vector(mode = "character", length = 5)
y <- vector(mode = "numeric", length = 7)
z <- vector(mode = "logical", length = 4)

x

## [1] "" "" "" "" ""

y

## [1] 0 0 0 0 0 0 0
```

```
z
```

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

```
Sys.Date()
```

```
## [1] "2018-04-18"
```

```
hoje<-Sys.Date() # cria um objeto do tipo <Date>, que foi funcionalmente criado (CRAN)
hoje
```

```
## [1] "2018-04-18"
```

```
format(hoje, "%d %b %Y") # exibe a data de hoje num formato padrão ISO/ABNT
```

```
## [1] "18 abr 2018"
```

```
dezsemanas<-seq(hoje, len = 10, by = "1 week")
dez.semanas<-seq(hoje, len = 10, by = "1 week")
rm(dezsemanas)
dez.semanas
```

```
## [1] "2018-04-18" "2018-04-25" "2018-05-02" "2018-05-09" "2018-05-16"
```

```
## [6] "2018-05-23" "2018-05-30" "2018-06-06" "2018-06-13" "2018-06-20"
```

```
dez.semanas<-dez.semanas-7
dez.semanas
```

```
## [1] "2018-04-11" "2018-04-18" "2018-04-25" "2018-05-02" "2018-05-09"
```

```
## [6] "2018-05-16" "2018-05-23" "2018-05-30" "2018-06-06" "2018-06-13"
```

```
#w <- vector(mode = "Date", length = 6) # *Error*
# porque não é um parâmetro válido para o argumento *mode* da função vector()
```

```
# Criando um vetor de datas para servir de rótulos para nossa série temporal experimental
#dez_semanas<-seq(c("2018-04-11"), len = 10, by = "1 week") # Error
# Porque "2018-04-11" é um tipo básico <char> e não um tipo especial <Date>.
# É preciso converter <char> em <Date>. E, claro, há uma função para isso!!!
dez_semanas<-seq.Date(from = as.Date("2018-04-11"), len = 10, by = "1 week")
dez_semanas<-seq(from = as.Date("2018-04-11"), len = 10, by = "1 week")
dez_semanas
```

```
## [1] "2018-04-11" "2018-04-18" "2018-04-25" "2018-05-02" "2018-05-09"
```

```
## [6] "2018-05-16" "2018-05-23" "2018-05-30" "2018-06-06" "2018-06-13"
```

```
# o NA é um valor que pode ser atribuído a uma posição de um vetor de qualquer tipo
a<-c(1:1000) # Uma composição da função c() com a função seq(), simbolizada pelos :
sum(a)
```

```
## [1] 500500
```

```
args(sum)
```

```
## function (... , na.rm = FALSE)
## NULL
```

```
a[1001]<-NA
a
```

```
##      [1]      1      2      3      4      5      6      7      8      9     10     11     12     13
##     [14]     14     15     16     17     18     19     20     21     22     23     24     25     26
##     [27]     27     28     29     30     31     32     33     34     35     36     37     38     39
```

| | | | | | | | | | | | | | | |
|----|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ## | [40] | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 |
| ## | [53] | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 |
| ## | [66] | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 |
| ## | [79] | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 |
| ## | [92] | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |
| ## | [105] | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 |
| ## | [118] | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 |
| ## | [131] | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| ## | [144] | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 |
| ## | [157] | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 |
| ## | [170] | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 |
| ## | [183] | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 |
| ## | [196] | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 |
| ## | [209] | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 |
| ## | [222] | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 |
| ## | [235] | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 |
| ## | [248] | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 | 256 | 257 | 258 | 259 | 260 |
| ## | [261] | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 | 272 | 273 |
| ## | [274] | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | 286 |
| ## | [287] | 287 | 288 | 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 |
| ## | [300] | 300 | 301 | 302 | 303 | 304 | 305 | 306 | 307 | 308 | 309 | 310 | 311 | 312 |
| ## | [313] | 313 | 314 | 315 | 316 | 317 | 318 | 319 | 320 | 321 | 322 | 323 | 324 | 325 |
| ## | [326] | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | 334 | 335 | 336 | 337 | 338 |
| ## | [339] | 339 | 340 | 341 | 342 | 343 | 344 | 345 | 346 | 347 | 348 | 349 | 350 | 351 |
| ## | [352] | 352 | 353 | 354 | 355 | 356 | 357 | 358 | 359 | 360 | 361 | 362 | 363 | 364 |
| ## | [365] | 365 | 366 | 367 | 368 | 369 | 370 | 371 | 372 | 373 | 374 | 375 | 376 | 377 |
| ## | [378] | 378 | 379 | 380 | 381 | 382 | 383 | 384 | 385 | 386 | 387 | 388 | 389 | 390 |
| ## | [391] | 391 | 392 | 393 | 394 | 395 | 396 | 397 | 398 | 399 | 400 | 401 | 402 | 403 |
| ## | [404] | 404 | 405 | 406 | 407 | 408 | 409 | 410 | 411 | 412 | 413 | 414 | 415 | 416 |
| ## | [417] | 417 | 418 | 419 | 420 | 421 | 422 | 423 | 424 | 425 | 426 | 427 | 428 | 429 |
| ## | [430] | 430 | 431 | 432 | 433 | 434 | 435 | 436 | 437 | 438 | 439 | 440 | 441 | 442 |
| ## | [443] | 443 | 444 | 445 | 446 | 447 | 448 | 449 | 450 | 451 | 452 | 453 | 454 | 455 |
| ## | [456] | 456 | 457 | 458 | 459 | 460 | 461 | 462 | 463 | 464 | 465 | 466 | 467 | 468 |
| ## | [469] | 469 | 470 | 471 | 472 | 473 | 474 | 475 | 476 | 477 | 478 | 479 | 480 | 481 |
| ## | [482] | 482 | 483 | 484 | 485 | 486 | 487 | 488 | 489 | 490 | 491 | 492 | 493 | 494 |
| ## | [495] | 495 | 496 | 497 | 498 | 499 | 500 | 501 | 502 | 503 | 504 | 505 | 506 | 507 |
| ## | [508] | 508 | 509 | 510 | 511 | 512 | 513 | 514 | 515 | 516 | 517 | 518 | 519 | 520 |
| ## | [521] | 521 | 522 | 523 | 524 | 525 | 526 | 527 | 528 | 529 | 530 | 531 | 532 | 533 |
| ## | [534] | 534 | 535 | 536 | 537 | 538 | 539 | 540 | 541 | 542 | 543 | 544 | 545 | 546 |
| ## | [547] | 547 | 548 | 549 | 550 | 551 | 552 | 553 | 554 | 555 | 556 | 557 | 558 | 559 |
| ## | [560] | 560 | 561 | 562 | 563 | 564 | 565 | 566 | 567 | 568 | 569 | 570 | 571 | 572 |
| ## | [573] | 573 | 574 | 575 | 576 | 577 | 578 | 579 | 580 | 581 | 582 | 583 | 584 | 585 |
| ## | [586] | 586 | 587 | 588 | 589 | 590 | 591 | 592 | 593 | 594 | 595 | 596 | 597 | 598 |
| ## | [599] | 599 | 600 | 601 | 602 | 603 | 604 | 605 | 606 | 607 | 608 | 609 | 610 | 611 |
| ## | [612] | 612 | 613 | 614 | 615 | 616 | 617 | 618 | 619 | 620 | 621 | 622 | 623 | 624 |
| ## | [625] | 625 | 626 | 627 | 628 | 629 | 630 | 631 | 632 | 633 | 634 | 635 | 636 | 637 |
| ## | [638] | 638 | 639 | 640 | 641 | 642 | 643 | 644 | 645 | 646 | 647 | 648 | 649 | 650 |
| ## | [651] | 651 | 652 | 653 | 654 | 655 | 656 | 657 | 658 | 659 | 660 | 661 | 662 | 663 |
| ## | [664] | 664 | 665 | 666 | 667 | 668 | 669 | 670 | 671 | 672 | 673 | 674 | 675 | 676 |
| ## | [677] | 677 | 678 | 679 | 680 | 681 | 682 | 683 | 684 | 685 | 686 | 687 | 688 | 689 |
| ## | [690] | 690 | 691 | 692 | 693 | 694 | 695 | 696 | 697 | 698 | 699 | 700 | 701 | 702 |
| ## | [703] | 703 | 704 | 705 | 706 | 707 | 708 | 709 | 710 | 711 | 712 | 713 | 714 | 715 |
| ## | [716] | 716 | 717 | 718 | 719 | 720 | 721 | 722 | 723 | 724 | 725 | 726 | 727 | 728 |
| ## | [729] | 729 | 730 | 731 | 732 | 733 | 734 | 735 | 736 | 737 | 738 | 739 | 740 | 741 |


```
## [742] 742 743 744 745 746 747 748 749 750 751 752 753 754
## [755] 755 756 757 758 759 760 761 762 763 764 765 766 767
## [768] 768 769 770 771 772 773 774 775 776 777 778 779 780
## [781] 781 782 783 784 785 786 787 788 789 790 791 792 793
## [794] 794 795 796 797 798 799 800 801 802 803 804 805 806
## [807] 807 808 809 810 811 812 813 814 815 816 817 818 819
## [820] 820 821 822 823 824 825 826 827 828 829 830 831 832
## [833] 833 834 835 836 837 838 839 840 841 842 843 844 845
## [846] 846 847 848 849 850 851 852 853 854 855 856 857 858
## [859] 859 860 861 862 863 864 865 866 867 868 869 870 871
## [872] 872 873 874 875 876 877 878 879 880 881 882 883 884
## [885] 885 886 887 888 889 890 891 892 893 894 895 896 897
## [898] 898 899 900 901 902 903 904 905 906 907 908 909 910
## [911] 911 912 913 914 915 916 917 918 919 920 921 922 923
## [924] 924 925 926 927 928 929 930 931 932 933 934 935 936
## [937] 937 938 939 940 941 942 943 944 945 946 947 948 949
## [950] 950 951 952 953 954 955 956 957 958 959 960 961 962
## [963] 963 964 965 966 967 968 969 970 971 972 973 974 975
## [976] 976 977 978 979 980 981 982 983 984 985 986 987 988
## [989] 989 990 991 992 993 994 995 996 997 998 999 1000 NA
```

```
sum(a)
```

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

```
sum(a, na.rm = TRUE) # 1 NA será removido para não indeterminar a soma de 1000 parcelas
```

```
## [1] 500500
```

```
mean(a)
```

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

```
mean(a, na.rm = TRUE) # nem sua média
```

```
## [1] 500.5
```

```
sd(a)
```

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

```
sd(a, na.rm = TRUE) # nem seu desvio padrão (standart desviation)
```

```
## [1] 288.8194
```

```
#####
# CUIDADO PORQUE UM ÚNICO **NA** NUMA BD PROPAGA SUA CAPACIDADE DE IMPEDIR QUE CÁLCULOS DE ESTATÍSTICA .
#####
```

Criando vetores para receber variáveis de uma BD

```
nomes<-c("Bernard",
"Carlos",
"Cleuler",
"Helber",
"Larissa",
"Mateus",
```

```

"Michell",
"Nayana",
"Paula",
"Rafael",
"Tatiane",
"Thiago",
"Wesley")
h<-c(1.74,1.63,1.77,1.75,NA,1.85,1.6,NA,1.55,1.7,1.63,1.7,1.75)
p<-c(63.8,
79.5,
81.6,
81.3,
49,
82.7,
57.6,
56.3,
72.4,
62.1,
52.6,
82.1,
81.9)

dez_semanas[1]

## [1] "2018-04-11"
nomes

## [1] "Bernard" "Carlos" "Cleuler" "Helber" "Larissa" "Mateus" "Michell"
## [8] "Nayana" "Paula" "Rafael" "Tatiane" "Thiago" "Wesley"
h

## [1] 1.74 1.63 1.77 1.75 NA 1.85 1.60 NA 1.55 1.70 1.63 1.70 1.75
p

## [1] 63.8 79.5 81.6 81.3 49.0 82.7 57.6 56.3 72.4 62.1 52.6 82.1 81.9

```

Exercícios - Para Resolução em Sala

Refletir e responder às seguintes questões *pragmáticas*:

- 1) Qual a altura média da sua turma de R?
- 2) Qual o peso médio da sua turma de R na aula do dia 11 abr. 2018?
- 3) Qual o número médio de caracteres dos prenomes dos alunos da turma de R que mediram seus pesos no dia 11 abr. 2018?
- 4) Qual o número médio de caracteres dos prenomes dos alunos matriculados nesta turma de R?
- 5) Qual o desvio padrão das médias encontradas?
- 6) Quem está abaixo e acima da média mais ou menos 1 Desvio Padrão?
- 7) Calcule o IMC de cada observação do dia 11 abr. 2018.

```

# Invocando as funções mean() e sd() para uma <var> <vector> <num>
#1) Média e Desvio Padrão (#5) das alturas dos alunos:
hm<- mean(h, na.rm=TRUE)
hDP<-sd(h, na.rm=TRUE) # Desvio padrão da altura é uma medida de dispersão dessa variável
# É uma turma com 8.7 cm de dispersão em torno da altura média de 1.70 m
# São brasileiros de estatura mediana, gostam muito de..., mas...

```

```
#6) Quem está abaixo e acima da média mais ou menos 1 Desvio Padrão?
```

```
hm
```

```
## [1] 1.697273
```

```
hDP
```

```
## [1] 0.08730303
```

```
h[h<hm-hDP | h>hm+hDP]
```

```
## [1] NA 1.85 1.60 NA 1.55
```

```
nomes[h<hm-hDP | h>hm+hDP] # Eis os outliers da estatura de nossa turma.
```

```
## [1] NA "Mateus" "Michell" NA "Paula"
```

```
#2) Média e Desvio Padrão (#5) dos pesos dos alunos:
```

```
pm<- mean(p, na.rm=TRUE) # É uma turma de magros!!! Conclusão precipitada?
```

```
pDP<-sd(p, na.rm=TRUE)
```

```
# A média do peso da turma no dia 11 abr. 2018 é de 69.5 Kg
```

```
# O Desvio Padrão dessas 13 observações de peso = 12.9 Kg
```

```
#6) Quem está abaixo e acima da média mais ou menos 1 Desvio Padrão?
```

```
p[p<pm-pDP | p>pm+pDP]
```

```
## [1] 49.0 82.7 56.3 52.6
```

```
nomes[p<pm-pDP | p>pm+pDP] # Eis os outliers da nossa turma.
```

```
## [1] "Larissa" "Mateus" "Nayana" "Tatiane"
```

```
#3) Número médio de caracteres dos prenomes dos alunos da turma de R que mediram seus pesos no dia 11 a
```

```
mean(length(nomes))
```

```
## [1] 13
```

```
sd(mean(length(nomes))) # retorna um NA. Por que?
```

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

```
nomes
```

```
## [1] "Bernard" "Carlos" "Cleuler" "Helber" "Larissa" "Mateus" "Michell"
```

```
## [8] "Nayana" "Paula" "Rafael" "Tatiane" "Thiago" "Wesley"
```

```
tam_nomes<-length(nomes) # cuidado porque retorna o comprimento do vetor names = 13!!!
```

```
tam_nomes<-nchar(nomes, keepNA = NA)
```

```
tam_nomes
```

```
## [1] 7 6 7 6 7 6 7 6 5 6 7 6 6
```

```
tam_nomes_media<-mean(tam_nomes)
```

```
tam_nomes_media
```

```
## [1] 6.307692
```

```
tam_nomes_DP <-sd(tam_nomes)
```

```
tam_nomes_DP
```

```
## [1] 0.6304252
```

```
#6) Quem está abaixo e acima da média mais ou menos 1 Desvio Padrão?
```

```
tam_nomes[tam_nomes<tam_nomes_media-tam_nomes_DP | tam_nomes>tam_nomes_media+tam_nomes_DP]
```

```
## [1] 7 7 7 7 5 7
nomes[tam_nomes<tam_nomes_media-tam_nomes_DP | tam_nomes>tam_nomes_media+tam_nomes_DP] # Eis os nomes d

## [1] "Bernard" "Cleuler" "Larissa" "Michell" "Paula" "Tatiane"
#7) Cálculo do IMC de cada observação do dia 11 abr. 2018.
#O cálculo do IMC é feito dividindo o peso (em quilogramas) pela altura (em metros) elevada ao quadrado
IMC<-p/h^2
IMC

## [1] 21.07280 29.92209 26.04616 26.54694 NA 24.16362 22.50000
## [8] NA 30.13528 21.48789 19.79751 28.40830 26.74286
IMC_m<- mean(IMC, na.rm=TRUE) # É uma uma turma de magros!!! Conclusão precipitada?
IMC_m # = 25.17 Kg/m2 # O IMC médio da turma indica ligeiramente acima do peso normal

## [1] 25.16577
IMC_DP<-sd(IMC, na.rm=TRUE)
IMC_DP# = 3.61 Kg/m2

## [1] 3.608471
IMC[IMC<18.5 | IMC>=25]

## [1] 29.92209 26.04616 26.54694 NA NA 30.13528 28.40830 26.74286
nomes[IMC<18.5 | IMC>=25] # Revelar ou não revelar. Eis a questão!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

## [1] "Carlos" "Cleuler" "Helber" NA NA "Paula" "Thiago"
## [8] "Wesley"

# cut() Convert Numeric <num> to Factor <fctr>
#####
# CUIDADO PORQUE UM ÚNICO ERRO DE SINTAXE FAZ COM QUE O COMPILADOR INTERROMPA A EXECUÇÃO DO SRCIPT (CÓD
#####

Exercício da Aula n. 2: 8) Redija e salve um script para a função linear em R. Gere um gráfico para essa
função no intervalo [0,5] e salve-o no formato .pdf.

#####
# TENTATIVAS E ERROS PARA GERAR UM GRÁFICO y=f(x)=a.x + b
#
# IMPORTANDO UM ARQUIVO FEITO PELO BERNARD E CONVERTENDO-O DE UTF-8 PARA WINDOWS-1252
#
#####

library(descr)
library(stats)
getwd()

## [1] "C:/Users/cleuler-bn/Documents/R_CS/Aula3"

#linhas<-readLines("Plottar_grafico.R")
#linhas<-fromUTF8(linhas)
#writeLines(linhas, "Plotar_grafico-win.R")

#Script desenvolvido para criar graficos
#Criar função da equação da reta > y = ax+b
#-----Parametros-----
```

```

#      a = Coeficiente linear
#      b = Coeficiente Angular
#      x = Vetor de valores no Eixo X
#      y = Vetor de valores no Eixo Y

#Cria funcao que representa a equacao da reta
linear <- function(a,b,x){
  y <- a*x + b
  return(y)
}

a <- 1.5 #Coeficiente linear
b <- 0.5 #Coeficiente angular

#x <- 1:10 #Vetor de valores do Eixo X # HAVIA UM ERROR. NÃO DE SINTAXE, MAS DE PROGRAMAÇÃO
x <- 1:10 #Vetor de valores do Eixo X

#y <- linear(x) # OCORREU OUTRO ERROR. AO CHAMAR A FUNÇÃO linear()
y <- linear(a,b,x) # é preciso repassar os parâmetros dos argumentos a e b da função

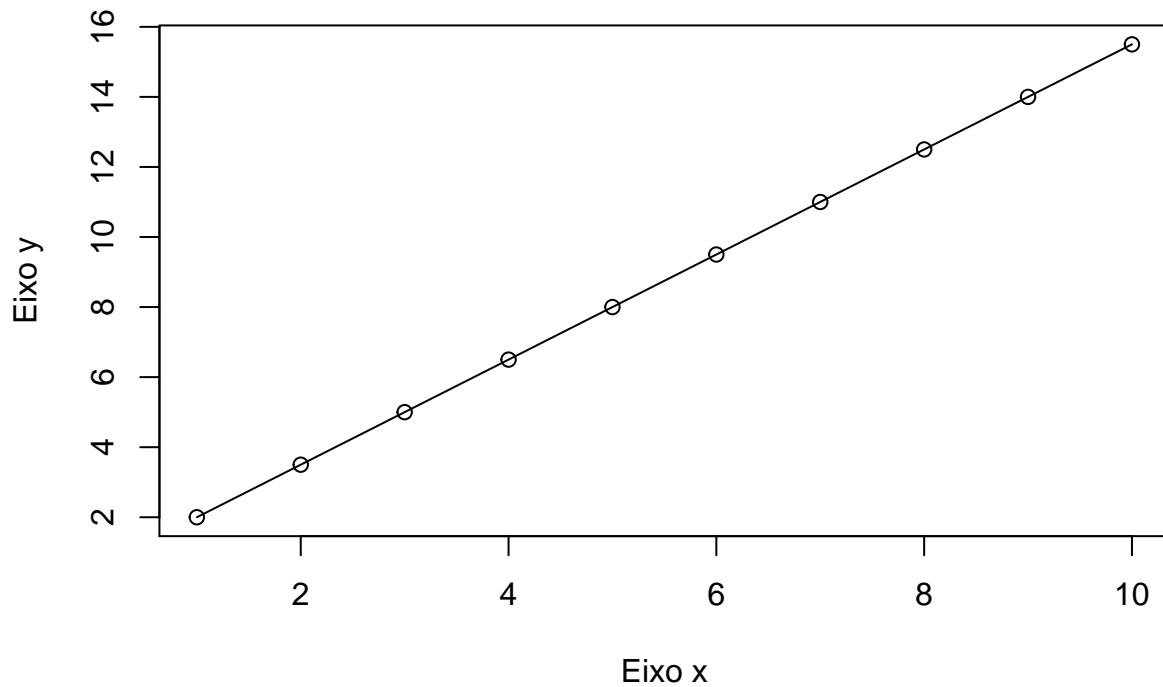
print(y) #Mostrar os valores do Eixo Y

## [1] 2.0 3.5 5.0 6.5 8.0 9.5 11.0 12.5 14.0 15.5

#Parametros do plot
#      main = Titulo do grafico
#      ylab = Nome do Eixo Y
#      xlab = Nome do Eixo X
#      type = Tipo de plotagem > l = linha, p = pontos, h = histograma
plot(x,y,main='Gráfico Curso R',ylab='Eixo y',xlab='Eixo x',type='o')

```

Gráfico Curso R

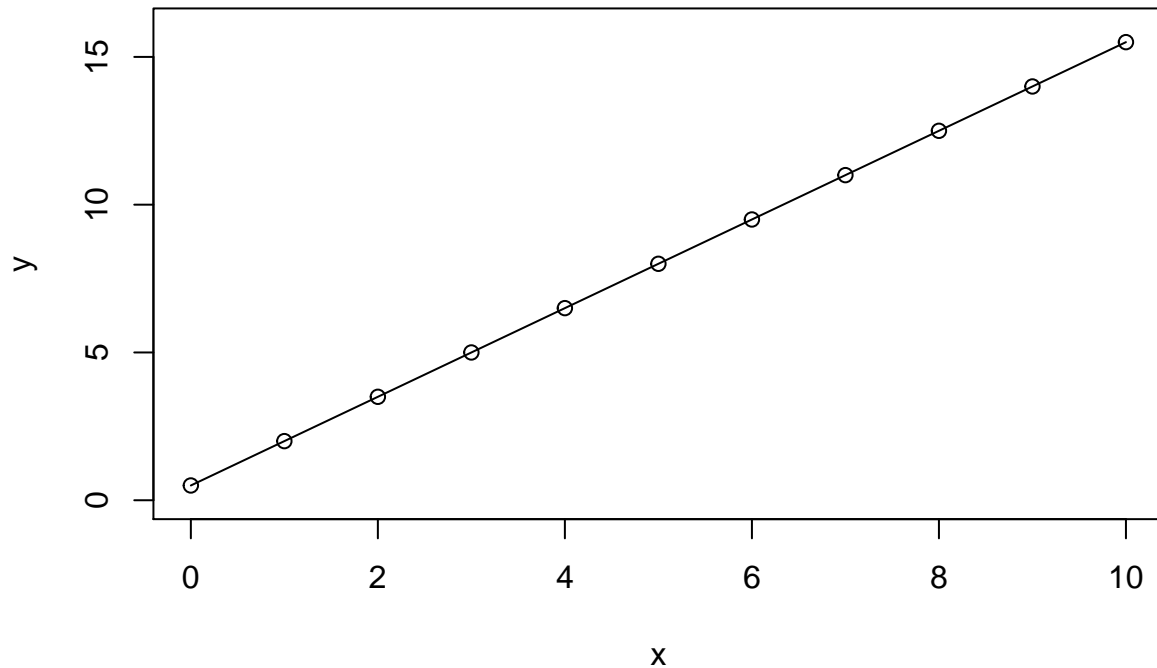


```
# Redesenhando o mesmo Gráfico  
x <- 0:10 #Vetor de valores do Eixo X  
y <- linear(a,b,x) # é preciso repassar os parâmetros dos argumentos a e b da função  
print(y) #Mostrar os valores do Eixo Y
```

```
## [1] 0.5 2.0 3.5 5.0 6.5 8.0 9.5 11.0 12.5 14.0 15.5
```

```
y1 <- linear(2,0,x)  
y2 <- linear(0.5,0,x)  
  
plot.new()  
plot(x,y,xlim=c(0,10),ylim=c(0,16),main='Gráfico Curso R',ylab='y',xlab='x',type='o')
```

Gráfico Curso R



```
#lines(x,y1, col="blue") # Error: plot.new has not been called yet  
#lines(x,y2, col="red")
```

Exercícios remanescentes da Aula n. 01:

9) Apresente duas funções lineares que sejam inversas. Plote-as juntamente com a função identidade. 10) Descrever os tipos de variáveis geradas na Job Area e suas características.

Trabalho Final do curso: Na primeira aula registrar a altura declarada e medir e registrar o peso de cada aluno, que poderá identificar-se com um apelido.

Em cada aula medir e registrar numa BD o peso de cada aluno numa sequência aleatória.

Calcular o IMC de cada observação e apontar para os IMC's abaixo ou acima da faixa recomendada pela literatura médica.

Calcular a média e o desvio padrão do IMC da população observada; detectar os pontos *outliers*.

Gerar um série temporal, com período de 7 dias, ao longo dos nossos 10 encontros.

Tratar eventuais NA's.

Descrever a variação do IMC médio da turma ao longo do curso, dado que será feito um apelo geral para aqueles acima da média para tentarem reduzi-lo nas próximas 10 semanas.

Fazer uma regressão linear do IMC médio em função do tempo analisando se ele sofreu alguma variação estatisticamente significativa.

Inferir qual resultado seria alcançado se o curso durasse 20 semanas.

Objetos

Vetores

Conjunto de elementos do mesmo tipo (logical, numeric, integer, double, character)

1. A forma mais simples de se criar um vetor é usar a função de concatenação `c()`.

```
value.num = c(3,4,2,6,20)
```

```
value.num
```

```
## [1] 3 4 2 6 20
```

```
value.char = c("koala", "kangaroo")
```

```
value.char
```

```
## [1] "koala" "kangaroo"
```

```
value.logical = c(FALSE, FALSE, TRUE, TRUE)
```

```
value.logical
```

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

2. Segunda maneira de criar vetor no R: usando a função `scan`

```
values = scan(text="
2
3
4
5"
)
```

```
values
```

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

3. Outra opção usando comando `rep`

```
rep(1,5)
```

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

```
rep(c(1,2),3)
```

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

```
rep(c(1,6),each=3)
```

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

```
rep(c(1,6),c(3,5))
```

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

4. Outra opção usando comando `seq`

```
seq(from=1,to=5)
```

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



```
seq(from=1, to=5, by=0.1)
```

```
## [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6
## [18] 2.7 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3
## [35] 4.4 4.5 4.6 4.7 4.8 4.9 5.0
```

```
seq(from=1, to=5, length=10)
```

```
## [1] 1.000000 1.444444 1.888889 2.333333 2.777778 3.222222 3.666667
## [8] 4.111111 4.555556 5.000000
```

```
rep(seq(from=1, to=5, length=10),each=3)
```

```
## [1] 1.000000 1.000000 1.000000 1.444444 1.444444 1.444444 1.888889
## [8] 1.888889 1.888889 2.333333 2.333333 2.333333 2.777778 2.777778
## [15] 2.777778 3.222222 3.222222 3.222222 3.666667 3.666667 3.666667
## [22] 4.111111 4.111111 4.111111 4.555556 4.555556 4.555556 5.000000
## [29] 5.000000 5.000000
```

5. Outra opção usando comando :

```
1:5
```

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

```
c(1:5,10)
```

```
## [1] 1 2 3 4 5 10
```

Operações com Vetores

```
x = 1:4
```

```
y = 5:8
```

```
x + y
```

```
## [1] 6 8 10 12
```

```
2*x +1
```

```
## [1] 3 5 7 9
```

```
x * y
```

```
## [1] 5 12 21 32
```

```
x / y
```

```
## [1] 0.2000000 0.3333333 0.4285714 0.5000000
```

```
log(x)
```

```
## [1] 0.0000000 0.6931472 1.0986123 1.3862944
```

```
log(x,10)
```

```
## [1] 0.0000000 0.3010300 0.4771213 0.6020600
```

```
sum(x)
```

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

```
mean(x)
```

```
## [1] 2.5
```

```
prod(x)
```

```
## [1] 24
```

```
var(x)
```

```
## [1] 1.666667
```

```
# O que é um vetor do tipo factor
```

```
# usado para variáveis categóricas
```

```
# Que apresenta vários Levels (níveis)
```

```
# Comumente cada nível recebe um nome gerando um conjunto denominado Labels
```

```
# Exemplo: No nosso estudo de caso seria interessante separar os dados amostrado segundo o sexo biológico
```

```
s<-c("m","m","m","m","f","m","m","f","f","m","f","m","m")
```

```
s # um <vctr> do tipo <chr>
```

```
## [1] "m" "m" "m" "m" "f" "m" "m" "f" "f" "m" "f" "m" "m"
```

```
mode(s)
```

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

```
class(s)
```

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

```
length(s)
```

```
## [1] 13
```

```
summary(s)
```

```
##      Length      Class      Mode
```

```
##           13 character character
```

```
str(s)
```

```
## chr [1:13] "m" "m" "m" "m" "f" "m" "m" "f" "f" "m" "f" "m" "m"
```

```
dput(s)
```

```
## c("m", "m", "m", "m", "f", "m", "m", "f", "f", "m", "f", "m",
```

```
## "m")
```

```
# Transformando numa variável factor <fctr>
```

```
s<-as.factor(s) # Destroi <chr> e recria o vetor s como um <fctr>
```

```
s
```

```
## [1] m m m m f m m f f m f m m
```

```
## Levels: f m
```

```
mode(s) # é um vetor do tipo <numeric>
```

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

```
class(s) # é um factor <fctr>, que é um caso especial de <numeric> indexado a Labels
```

```
## [1] "factor"
```

```
length(s)
```

```
## [1] 13
```

```
summary(s)
```

```
## f m
```

```
## 4 9
```

```
str(s) # investigando a structure da variável s do tipo <fctr>
```

```
## Factor w/ 2 levels "f","m": 2 2 2 2 1 2 2 1 1 2 ...
```

```
dput(s)
```

```
## structure(c(2L, 2L, 2L, 2L, 1L, 2L, 2L, 1L, 1L, 2L, 1L, 2L, 2L
```

```
## ), .Label = c("f", "m"), class = "factor")
```

```
table(s) # retorna um vetor tipo <fctr> com a frequência de cada um dos níveis (Levels) ou categorias d
```

```
## s
```

```
## f m
```

```
## 4 9
```

```
# Essa mesma função é usada para retornar tabulações cruzadas (cross table) de duas variáveis categóric
```

```
max(h, na.rm = TRUE)
```

```
## [1] 1.85
```

```
hcat <- cut(h,c(0,1.6,1.7,2.0),labels = c("Baixo","Médio","Alto"))
```

```
# função cat() Convert Numeric to Factor
```

```
str(hcat)
```

```
## Factor w/ 3 levels "Baixo","Médio",...: 3 2 3 3 NA 3 1 NA 1 2 ...
```

```
dput(hcat)
```

```
## structure(c(3L, 2L, 3L, 3L, NA, 3L, 1L, NA, 1L, 2L, 2L, 2L, 3L
```

```
## ), .Label = c("Baixo", "Médio", "Alto"), class = "factor")
```

```
table(hcat,s)
```

```
##      s
```

```
## hcat  f m
```

```
## Baixo 1 1
```

```
## Médio 1 3
```

```
## Alto  0 5
```

```
hm
```

```
## [1] 1.697273
```

```
ct<-table(hcat,s)
```

```
prop.table(ct,1)
```

```
##      s
```

```
## hcat      f      m
```

```
## Baixo 0.50 0.50
```

```
##      Médio 0.25 0.75
##      Alto  0.00 1.00
```

```
prop.table(ct,2)
```

```
##           s
## hcat      f      m
##  Baixo 0.5000000 0.1111111
##   Médio 0.5000000 0.3333333
##   Alto  0.0000000 0.5555556
```

```
prop.table(ct)
```

```
##           s
## hcat      f      m
##  Baixo 0.09090909 0.09090909
##   Médio 0.09090909 0.27272727
##   Alto  0.00000000 0.45454545
```

```
100*prop.table(ct)
```

```
##           s
## hcat      f      m
##  Baixo 9.090909 9.090909
##   Médio 9.090909 27.272727
##   Alto  0.000000 45.454545
```

```
# Analisando o resultados dessas cross tables p.u. vê-se que o IMC deve ser categorizado em feminino (X)
```

```
# Calculando a altura média das observações s == f
s=="f"
```

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

```
h[s=="f"]
```

```
## [1] NA NA 1.55 1.63
```

```
mean(h[s=="f"], na.rm=TRUE) # é média da estatura do sexo feminino = 1.59 m
```

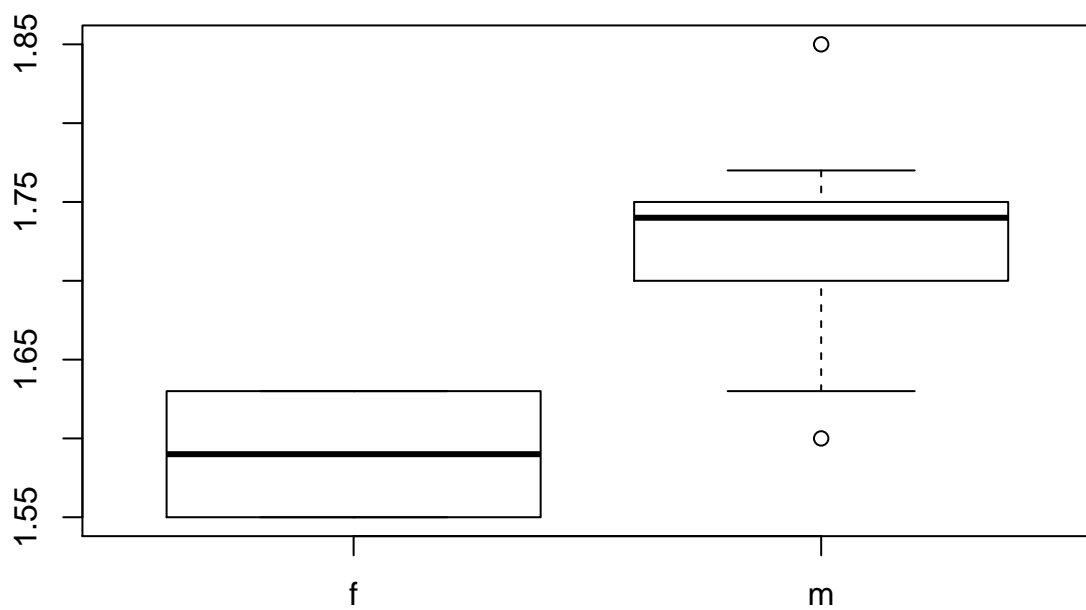
```
## [1] 1.59
```

```
mean(h[s=="m"], na.rm=TRUE) # é média da estatura do sexo masculino = 1.72 m
```

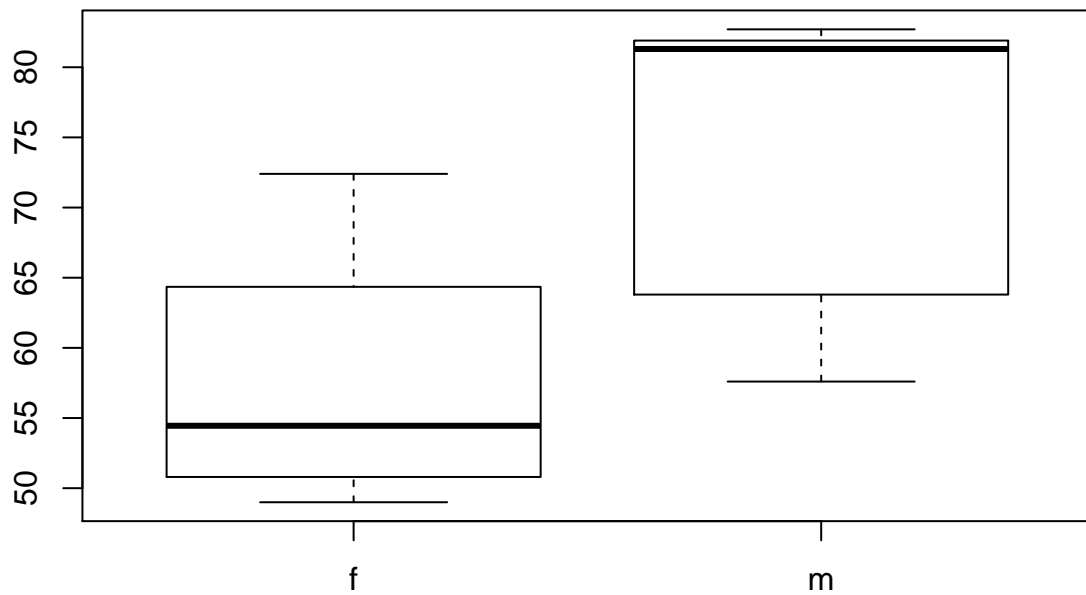
```
## [1] 1.721111
```

```
# Exibindo essa diferença graficamente
```

```
boxplot(h~s) # homens são, em média, mais alto que as mulheres
```

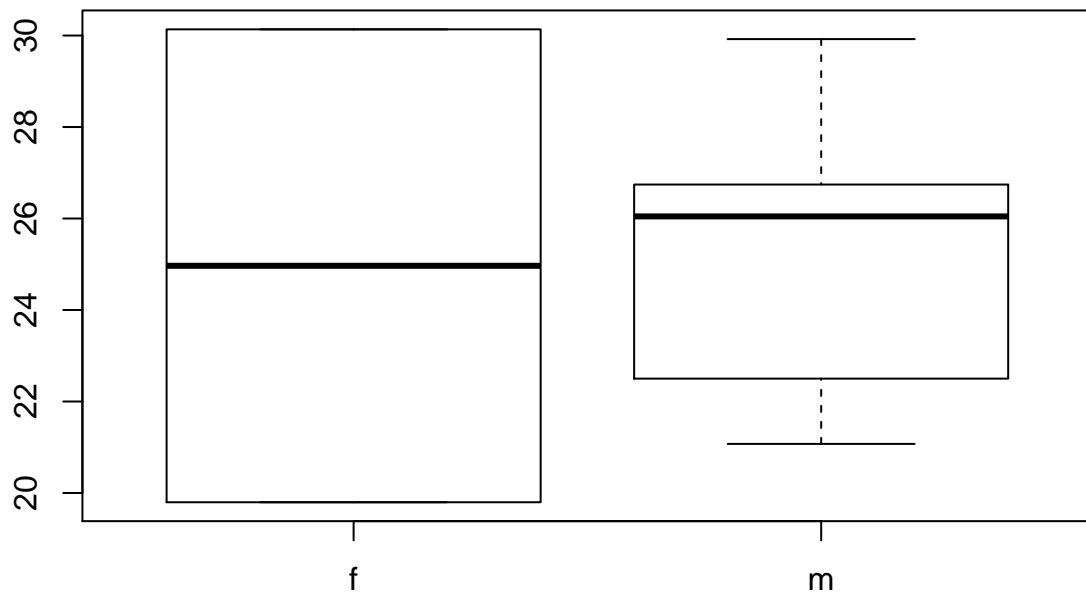


```
boxplot(p~s) # homens são, em média, mais pesados que as mulheres
```



*# Esses gráficos corroboram uma Hipótese de estratificação f & m para analisar o IMC????
Duvidar é preciso.
Transformar sua dúvida numa hipótese testável.
E testar adequadamente a Hipótese **contra** as observações colhidas no campo.*

```
boxplot(IMC~s)
```



Matriz

Conjunto de elementos dispostos em linhas e colunas, em que todos os elementos são do mesmo tipo

```
mat.num = matrix(c(1:16),4,4)
```

```
mat.num
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
mat.char = matrix(LETTERS[1:4],2,2)
```

```
mat.char
```

```
##      [,1] [,2]
## [1,] "A"  "C"
## [2,] "B"  "D"
```

Manipulando Matrizes

```
#Criando nomes para as linhas de uma matriz
```

```
rownames(mat.num) = c("Sao Paulo", "Americana", "Piracicaba", "Madson" )
```

```
colnames(mat.num) = 1:4
```

```
mat.num
```

```
##           1 2  3  4
## Sao Paulo  1 5  9 13
## Americana  2 6 10 14
## Piracicaba 3 7 11 15
## Madson     4 8 12 16
```

```
#Multiplicação elemento a elemento
```

```
mat.num2 = diag(seq(10,40,by=10))
```

```
mat.num2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  10   0   0   0
## [2,]   0  20   0   0
## [3,]   0   0  30   0
## [4,]   0   0   0  40
```

```
mat.num3 = mat.num * mat.num2
```

```
mat.num3
```

```
##           1  2  3  4
## Sao Paulo 10  0  0  0
## Americana  0 120  0  0
## Piracicaba 0  0 330  0
## Madson     0  0  0 640
```

```
#Multiplicação de Matrizes
```

```
iden = diag(4)
```

```
iden
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   1   0   0   0
## [2,]   0   1   0   0
## [3,]   0   0   1   0
## [4,]   0   0   0   1
```

```
mat.num%%iden
```

```
##           [,1] [,2] [,3] [,4]
## Sao Paulo   1   5   9  13
## Americana   2   6  10  14
## Piracicaba  3   7  11  15
## Madson      4   8  12  16
```



```

#Acessando elementos das matrizes

#Um elemento
mat.num[1,1]

## [1] 1

#Linhas
mat.num[1,]

## 1 2 3 4
## 1 5 9 13

#Colunas
mat.num[,3]

## Sao Paulo Americana Piracicaba Madson
## 9 10 11 12

#Sub Matrizes

mat.num[c(1,3,4), c(2,1,4)]

## 2 1 4
## Sao Paulo 5 1 13
## Piracicaba 7 3 15
## Madson 8 4 16

mat.num[c(T,F,T,T), c(T,T,F,T)]

## 1 2 4
## Sao Paulo 1 5 13
## Piracicaba 3 7 15
## Madson 4 8 16

mat.num[-c(1,3,4), -c(2,1,4)]

## [1] 10

```

Data.frames

São similares às matrizes no entanto permite que as colunas tenham diferentes tipos

```

data(iris)

iris

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 5.1 3.5 1.4 0.2 setosa
## 2 4.9 3.0 1.4 0.2 setosa
## 3 4.7 3.2 1.3 0.2 setosa
## 4 4.6 3.1 1.5 0.2 setosa
## 5 5.0 3.6 1.4 0.2 setosa
## 6 5.4 3.9 1.7 0.4 setosa
## 7 4.6 3.4 1.4 0.3 setosa
## 8 5.0 3.4 1.5 0.2 setosa
## 9 4.4 2.9 1.4 0.2 setosa

```

| | | | | | |
|-------|-----|-----|-----|-----|------------|
| ## 10 | 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| ## 11 | 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| ## 12 | 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| ## 13 | 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| ## 14 | 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| ## 15 | 5.8 | 4.0 | 1.2 | 0.2 | setosa |
| ## 16 | 5.7 | 4.4 | 1.5 | 0.4 | setosa |
| ## 17 | 5.4 | 3.9 | 1.3 | 0.4 | setosa |
| ## 18 | 5.1 | 3.5 | 1.4 | 0.3 | setosa |
| ## 19 | 5.7 | 3.8 | 1.7 | 0.3 | setosa |
| ## 20 | 5.1 | 3.8 | 1.5 | 0.3 | setosa |
| ## 21 | 5.4 | 3.4 | 1.7 | 0.2 | setosa |
| ## 22 | 5.1 | 3.7 | 1.5 | 0.4 | setosa |
| ## 23 | 4.6 | 3.6 | 1.0 | 0.2 | setosa |
| ## 24 | 5.1 | 3.3 | 1.7 | 0.5 | setosa |
| ## 25 | 4.8 | 3.4 | 1.9 | 0.2 | setosa |
| ## 26 | 5.0 | 3.0 | 1.6 | 0.2 | setosa |
| ## 27 | 5.0 | 3.4 | 1.6 | 0.4 | setosa |
| ## 28 | 5.2 | 3.5 | 1.5 | 0.2 | setosa |
| ## 29 | 5.2 | 3.4 | 1.4 | 0.2 | setosa |
| ## 30 | 4.7 | 3.2 | 1.6 | 0.2 | setosa |
| ## 31 | 4.8 | 3.1 | 1.6 | 0.2 | setosa |
| ## 32 | 5.4 | 3.4 | 1.5 | 0.4 | setosa |
| ## 33 | 5.2 | 4.1 | 1.5 | 0.1 | setosa |
| ## 34 | 5.5 | 4.2 | 1.4 | 0.2 | setosa |
| ## 35 | 4.9 | 3.1 | 1.5 | 0.2 | setosa |
| ## 36 | 5.0 | 3.2 | 1.2 | 0.2 | setosa |
| ## 37 | 5.5 | 3.5 | 1.3 | 0.2 | setosa |
| ## 38 | 4.9 | 3.6 | 1.4 | 0.1 | setosa |
| ## 39 | 4.4 | 3.0 | 1.3 | 0.2 | setosa |
| ## 40 | 5.1 | 3.4 | 1.5 | 0.2 | setosa |
| ## 41 | 5.0 | 3.5 | 1.3 | 0.3 | setosa |
| ## 42 | 4.5 | 2.3 | 1.3 | 0.3 | setosa |
| ## 43 | 4.4 | 3.2 | 1.3 | 0.2 | setosa |
| ## 44 | 5.0 | 3.5 | 1.6 | 0.6 | setosa |
| ## 45 | 5.1 | 3.8 | 1.9 | 0.4 | setosa |
| ## 46 | 4.8 | 3.0 | 1.4 | 0.3 | setosa |
| ## 47 | 5.1 | 3.8 | 1.6 | 0.2 | setosa |
| ## 48 | 4.6 | 3.2 | 1.4 | 0.2 | setosa |
| ## 49 | 5.3 | 3.7 | 1.5 | 0.2 | setosa |
| ## 50 | 5.0 | 3.3 | 1.4 | 0.2 | setosa |
| ## 51 | 7.0 | 3.2 | 4.7 | 1.4 | versicolor |
| ## 52 | 6.4 | 3.2 | 4.5 | 1.5 | versicolor |
| ## 53 | 6.9 | 3.1 | 4.9 | 1.5 | versicolor |
| ## 54 | 5.5 | 2.3 | 4.0 | 1.3 | versicolor |
| ## 55 | 6.5 | 2.8 | 4.6 | 1.5 | versicolor |
| ## 56 | 5.7 | 2.8 | 4.5 | 1.3 | versicolor |
| ## 57 | 6.3 | 3.3 | 4.7 | 1.6 | versicolor |
| ## 58 | 4.9 | 2.4 | 3.3 | 1.0 | versicolor |
| ## 59 | 6.6 | 2.9 | 4.6 | 1.3 | versicolor |
| ## 60 | 5.2 | 2.7 | 3.9 | 1.4 | versicolor |
| ## 61 | 5.0 | 2.0 | 3.5 | 1.0 | versicolor |
| ## 62 | 5.9 | 3.0 | 4.2 | 1.5 | versicolor |
| ## 63 | 6.0 | 2.2 | 4.0 | 1.0 | versicolor |

| | | | | |
|--------|-----|-----|-----|----------------|
| ## 64 | 6.1 | 2.9 | 4.7 | 1.4 versicolor |
| ## 65 | 5.6 | 2.9 | 3.6 | 1.3 versicolor |
| ## 66 | 6.7 | 3.1 | 4.4 | 1.4 versicolor |
| ## 67 | 5.6 | 3.0 | 4.5 | 1.5 versicolor |
| ## 68 | 5.8 | 2.7 | 4.1 | 1.0 versicolor |
| ## 69 | 6.2 | 2.2 | 4.5 | 1.5 versicolor |
| ## 70 | 5.6 | 2.5 | 3.9 | 1.1 versicolor |
| ## 71 | 5.9 | 3.2 | 4.8 | 1.8 versicolor |
| ## 72 | 6.1 | 2.8 | 4.0 | 1.3 versicolor |
| ## 73 | 6.3 | 2.5 | 4.9 | 1.5 versicolor |
| ## 74 | 6.1 | 2.8 | 4.7 | 1.2 versicolor |
| ## 75 | 6.4 | 2.9 | 4.3 | 1.3 versicolor |
| ## 76 | 6.6 | 3.0 | 4.4 | 1.4 versicolor |
| ## 77 | 6.8 | 2.8 | 4.8 | 1.4 versicolor |
| ## 78 | 6.7 | 3.0 | 5.0 | 1.7 versicolor |
| ## 79 | 6.0 | 2.9 | 4.5 | 1.5 versicolor |
| ## 80 | 5.7 | 2.6 | 3.5 | 1.0 versicolor |
| ## 81 | 5.5 | 2.4 | 3.8 | 1.1 versicolor |
| ## 82 | 5.5 | 2.4 | 3.7 | 1.0 versicolor |
| ## 83 | 5.8 | 2.7 | 3.9 | 1.2 versicolor |
| ## 84 | 6.0 | 2.7 | 5.1 | 1.6 versicolor |
| ## 85 | 5.4 | 3.0 | 4.5 | 1.5 versicolor |
| ## 86 | 6.0 | 3.4 | 4.5 | 1.6 versicolor |
| ## 87 | 6.7 | 3.1 | 4.7 | 1.5 versicolor |
| ## 88 | 6.3 | 2.3 | 4.4 | 1.3 versicolor |
| ## 89 | 5.6 | 3.0 | 4.1 | 1.3 versicolor |
| ## 90 | 5.5 | 2.5 | 4.0 | 1.3 versicolor |
| ## 91 | 5.5 | 2.6 | 4.4 | 1.2 versicolor |
| ## 92 | 6.1 | 3.0 | 4.6 | 1.4 versicolor |
| ## 93 | 5.8 | 2.6 | 4.0 | 1.2 versicolor |
| ## 94 | 5.0 | 2.3 | 3.3 | 1.0 versicolor |
| ## 95 | 5.6 | 2.7 | 4.2 | 1.3 versicolor |
| ## 96 | 5.7 | 3.0 | 4.2 | 1.2 versicolor |
| ## 97 | 5.7 | 2.9 | 4.2 | 1.3 versicolor |
| ## 98 | 6.2 | 2.9 | 4.3 | 1.3 versicolor |
| ## 99 | 5.1 | 2.5 | 3.0 | 1.1 versicolor |
| ## 100 | 5.7 | 2.8 | 4.1 | 1.3 versicolor |
| ## 101 | 6.3 | 3.3 | 6.0 | 2.5 virginica |
| ## 102 | 5.8 | 2.7 | 5.1 | 1.9 virginica |
| ## 103 | 7.1 | 3.0 | 5.9 | 2.1 virginica |
| ## 104 | 6.3 | 2.9 | 5.6 | 1.8 virginica |
| ## 105 | 6.5 | 3.0 | 5.8 | 2.2 virginica |
| ## 106 | 7.6 | 3.0 | 6.6 | 2.1 virginica |
| ## 107 | 4.9 | 2.5 | 4.5 | 1.7 virginica |
| ## 108 | 7.3 | 2.9 | 6.3 | 1.8 virginica |
| ## 109 | 6.7 | 2.5 | 5.8 | 1.8 virginica |
| ## 110 | 7.2 | 3.6 | 6.1 | 2.5 virginica |
| ## 111 | 6.5 | 3.2 | 5.1 | 2.0 virginica |
| ## 112 | 6.4 | 2.7 | 5.3 | 1.9 virginica |
| ## 113 | 6.8 | 3.0 | 5.5 | 2.1 virginica |
| ## 114 | 5.7 | 2.5 | 5.0 | 2.0 virginica |
| ## 115 | 5.8 | 2.8 | 5.1 | 2.4 virginica |
| ## 116 | 6.4 | 3.2 | 5.3 | 2.3 virginica |
| ## 117 | 6.5 | 3.0 | 5.5 | 1.8 virginica |

```
## 118      7.7      3.8      6.7      2.2 virginica
## 119      7.7      2.6      6.9      2.3 virginica
## 120      6.0      2.2      5.0      1.5 virginica
## 121      6.9      3.2      5.7      2.3 virginica
## 122      5.6      2.8      4.9      2.0 virginica
## 123      7.7      2.8      6.7      2.0 virginica
## 124      6.3      2.7      4.9      1.8 virginica
## 125      6.7      3.3      5.7      2.1 virginica
## 126      7.2      3.2      6.0      1.8 virginica
## 127      6.2      2.8      4.8      1.8 virginica
## 128      6.1      3.0      4.9      1.8 virginica
## 129      6.4      2.8      5.6      2.1 virginica
## 130      7.2      3.0      5.8      1.6 virginica
## 131      7.4      2.8      6.1      1.9 virginica
## 132      7.9      3.8      6.4      2.0 virginica
## 133      6.4      2.8      5.6      2.2 virginica
## 134      6.3      2.8      5.1      1.5 virginica
## 135      6.1      2.6      5.6      1.4 virginica
## 136      7.7      3.0      6.1      2.3 virginica
## 137      6.3      3.4      5.6      2.4 virginica
## 138      6.4      3.1      5.5      1.8 virginica
## 139      6.0      3.0      4.8      1.8 virginica
## 140      6.9      3.1      5.4      2.1 virginica
## 141      6.7      3.1      5.6      2.4 virginica
## 142      6.9      3.1      5.1      2.3 virginica
## 143      5.8      2.7      5.1      1.9 virginica
## 144      6.8      3.2      5.9      2.3 virginica
## 145      6.7      3.3      5.7      2.5 virginica
## 146      6.7      3.0      5.2      2.3 virginica
## 147      6.3      2.5      5.0      1.9 virginica
## 148      6.5      3.0      5.2      2.0 virginica
## 149      6.2      3.4      5.4      2.3 virginica
## 150      5.9      3.0      5.1      1.8 virginica
```

```
iris$Sepal.Length
```

```
## [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
## [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
## [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
## [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
## [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
## [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
iris$Renato = TRUE
```

```
iris
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species Renato
## 1           5.1         3.5         1.4         0.2    setosa    TRUE
## 2           4.9         3.0         1.4         0.2    setosa    TRUE
## 3           4.7         3.2         1.3         0.2    setosa    TRUE
## 4           4.6         3.1         1.5         0.2    setosa    TRUE
## 5           5.0         3.6         1.4         0.2    setosa    TRUE
```

| | | | | | | |
|-------|-----|-----|-----|-----|------------|------|
| ## 6 | 5.4 | 3.9 | 1.7 | 0.4 | setosa | TRUE |
| ## 7 | 4.6 | 3.4 | 1.4 | 0.3 | setosa | TRUE |
| ## 8 | 5.0 | 3.4 | 1.5 | 0.2 | setosa | TRUE |
| ## 9 | 4.4 | 2.9 | 1.4 | 0.2 | setosa | TRUE |
| ## 10 | 4.9 | 3.1 | 1.5 | 0.1 | setosa | TRUE |
| ## 11 | 5.4 | 3.7 | 1.5 | 0.2 | setosa | TRUE |
| ## 12 | 4.8 | 3.4 | 1.6 | 0.2 | setosa | TRUE |
| ## 13 | 4.8 | 3.0 | 1.4 | 0.1 | setosa | TRUE |
| ## 14 | 4.3 | 3.0 | 1.1 | 0.1 | setosa | TRUE |
| ## 15 | 5.8 | 4.0 | 1.2 | 0.2 | setosa | TRUE |
| ## 16 | 5.7 | 4.4 | 1.5 | 0.4 | setosa | TRUE |
| ## 17 | 5.4 | 3.9 | 1.3 | 0.4 | setosa | TRUE |
| ## 18 | 5.1 | 3.5 | 1.4 | 0.3 | setosa | TRUE |
| ## 19 | 5.7 | 3.8 | 1.7 | 0.3 | setosa | TRUE |
| ## 20 | 5.1 | 3.8 | 1.5 | 0.3 | setosa | TRUE |
| ## 21 | 5.4 | 3.4 | 1.7 | 0.2 | setosa | TRUE |
| ## 22 | 5.1 | 3.7 | 1.5 | 0.4 | setosa | TRUE |
| ## 23 | 4.6 | 3.6 | 1.0 | 0.2 | setosa | TRUE |
| ## 24 | 5.1 | 3.3 | 1.7 | 0.5 | setosa | TRUE |
| ## 25 | 4.8 | 3.4 | 1.9 | 0.2 | setosa | TRUE |
| ## 26 | 5.0 | 3.0 | 1.6 | 0.2 | setosa | TRUE |
| ## 27 | 5.0 | 3.4 | 1.6 | 0.4 | setosa | TRUE |
| ## 28 | 5.2 | 3.5 | 1.5 | 0.2 | setosa | TRUE |
| ## 29 | 5.2 | 3.4 | 1.4 | 0.2 | setosa | TRUE |
| ## 30 | 4.7 | 3.2 | 1.6 | 0.2 | setosa | TRUE |
| ## 31 | 4.8 | 3.1 | 1.6 | 0.2 | setosa | TRUE |
| ## 32 | 5.4 | 3.4 | 1.5 | 0.4 | setosa | TRUE |
| ## 33 | 5.2 | 4.1 | 1.5 | 0.1 | setosa | TRUE |
| ## 34 | 5.5 | 4.2 | 1.4 | 0.2 | setosa | TRUE |
| ## 35 | 4.9 | 3.1 | 1.5 | 0.2 | setosa | TRUE |
| ## 36 | 5.0 | 3.2 | 1.2 | 0.2 | setosa | TRUE |
| ## 37 | 5.5 | 3.5 | 1.3 | 0.2 | setosa | TRUE |
| ## 38 | 4.9 | 3.6 | 1.4 | 0.1 | setosa | TRUE |
| ## 39 | 4.4 | 3.0 | 1.3 | 0.2 | setosa | TRUE |
| ## 40 | 5.1 | 3.4 | 1.5 | 0.2 | setosa | TRUE |
| ## 41 | 5.0 | 3.5 | 1.3 | 0.3 | setosa | TRUE |
| ## 42 | 4.5 | 2.3 | 1.3 | 0.3 | setosa | TRUE |
| ## 43 | 4.4 | 3.2 | 1.3 | 0.2 | setosa | TRUE |
| ## 44 | 5.0 | 3.5 | 1.6 | 0.6 | setosa | TRUE |
| ## 45 | 5.1 | 3.8 | 1.9 | 0.4 | setosa | TRUE |
| ## 46 | 4.8 | 3.0 | 1.4 | 0.3 | setosa | TRUE |
| ## 47 | 5.1 | 3.8 | 1.6 | 0.2 | setosa | TRUE |
| ## 48 | 4.6 | 3.2 | 1.4 | 0.2 | setosa | TRUE |
| ## 49 | 5.3 | 3.7 | 1.5 | 0.2 | setosa | TRUE |
| ## 50 | 5.0 | 3.3 | 1.4 | 0.2 | setosa | TRUE |
| ## 51 | 7.0 | 3.2 | 4.7 | 1.4 | versicolor | TRUE |
| ## 52 | 6.4 | 3.2 | 4.5 | 1.5 | versicolor | TRUE |
| ## 53 | 6.9 | 3.1 | 4.9 | 1.5 | versicolor | TRUE |
| ## 54 | 5.5 | 2.3 | 4.0 | 1.3 | versicolor | TRUE |
| ## 55 | 6.5 | 2.8 | 4.6 | 1.5 | versicolor | TRUE |
| ## 56 | 5.7 | 2.8 | 4.5 | 1.3 | versicolor | TRUE |
| ## 57 | 6.3 | 3.3 | 4.7 | 1.6 | versicolor | TRUE |
| ## 58 | 4.9 | 2.4 | 3.3 | 1.0 | versicolor | TRUE |
| ## 59 | 6.6 | 2.9 | 4.6 | 1.3 | versicolor | TRUE |

| | | | | | |
|--------|-----|-----|-----|----------------|------|
| ## 60 | 5.2 | 2.7 | 3.9 | 1.4 versicolor | TRUE |
| ## 61 | 5.0 | 2.0 | 3.5 | 1.0 versicolor | TRUE |
| ## 62 | 5.9 | 3.0 | 4.2 | 1.5 versicolor | TRUE |
| ## 63 | 6.0 | 2.2 | 4.0 | 1.0 versicolor | TRUE |
| ## 64 | 6.1 | 2.9 | 4.7 | 1.4 versicolor | TRUE |
| ## 65 | 5.6 | 2.9 | 3.6 | 1.3 versicolor | TRUE |
| ## 66 | 6.7 | 3.1 | 4.4 | 1.4 versicolor | TRUE |
| ## 67 | 5.6 | 3.0 | 4.5 | 1.5 versicolor | TRUE |
| ## 68 | 5.8 | 2.7 | 4.1 | 1.0 versicolor | TRUE |
| ## 69 | 6.2 | 2.2 | 4.5 | 1.5 versicolor | TRUE |
| ## 70 | 5.6 | 2.5 | 3.9 | 1.1 versicolor | TRUE |
| ## 71 | 5.9 | 3.2 | 4.8 | 1.8 versicolor | TRUE |
| ## 72 | 6.1 | 2.8 | 4.0 | 1.3 versicolor | TRUE |
| ## 73 | 6.3 | 2.5 | 4.9 | 1.5 versicolor | TRUE |
| ## 74 | 6.1 | 2.8 | 4.7 | 1.2 versicolor | TRUE |
| ## 75 | 6.4 | 2.9 | 4.3 | 1.3 versicolor | TRUE |
| ## 76 | 6.6 | 3.0 | 4.4 | 1.4 versicolor | TRUE |
| ## 77 | 6.8 | 2.8 | 4.8 | 1.4 versicolor | TRUE |
| ## 78 | 6.7 | 3.0 | 5.0 | 1.7 versicolor | TRUE |
| ## 79 | 6.0 | 2.9 | 4.5 | 1.5 versicolor | TRUE |
| ## 80 | 5.7 | 2.6 | 3.5 | 1.0 versicolor | TRUE |
| ## 81 | 5.5 | 2.4 | 3.8 | 1.1 versicolor | TRUE |
| ## 82 | 5.5 | 2.4 | 3.7 | 1.0 versicolor | TRUE |
| ## 83 | 5.8 | 2.7 | 3.9 | 1.2 versicolor | TRUE |
| ## 84 | 6.0 | 2.7 | 5.1 | 1.6 versicolor | TRUE |
| ## 85 | 5.4 | 3.0 | 4.5 | 1.5 versicolor | TRUE |
| ## 86 | 6.0 | 3.4 | 4.5 | 1.6 versicolor | TRUE |
| ## 87 | 6.7 | 3.1 | 4.7 | 1.5 versicolor | TRUE |
| ## 88 | 6.3 | 2.3 | 4.4 | 1.3 versicolor | TRUE |
| ## 89 | 5.6 | 3.0 | 4.1 | 1.3 versicolor | TRUE |
| ## 90 | 5.5 | 2.5 | 4.0 | 1.3 versicolor | TRUE |
| ## 91 | 5.5 | 2.6 | 4.4 | 1.2 versicolor | TRUE |
| ## 92 | 6.1 | 3.0 | 4.6 | 1.4 versicolor | TRUE |
| ## 93 | 5.8 | 2.6 | 4.0 | 1.2 versicolor | TRUE |
| ## 94 | 5.0 | 2.3 | 3.3 | 1.0 versicolor | TRUE |
| ## 95 | 5.6 | 2.7 | 4.2 | 1.3 versicolor | TRUE |
| ## 96 | 5.7 | 3.0 | 4.2 | 1.2 versicolor | TRUE |
| ## 97 | 5.7 | 2.9 | 4.2 | 1.3 versicolor | TRUE |
| ## 98 | 6.2 | 2.9 | 4.3 | 1.3 versicolor | TRUE |
| ## 99 | 5.1 | 2.5 | 3.0 | 1.1 versicolor | TRUE |
| ## 100 | 5.7 | 2.8 | 4.1 | 1.3 versicolor | TRUE |
| ## 101 | 6.3 | 3.3 | 6.0 | 2.5 virginica | TRUE |
| ## 102 | 5.8 | 2.7 | 5.1 | 1.9 virginica | TRUE |
| ## 103 | 7.1 | 3.0 | 5.9 | 2.1 virginica | TRUE |
| ## 104 | 6.3 | 2.9 | 5.6 | 1.8 virginica | TRUE |
| ## 105 | 6.5 | 3.0 | 5.8 | 2.2 virginica | TRUE |
| ## 106 | 7.6 | 3.0 | 6.6 | 2.1 virginica | TRUE |
| ## 107 | 4.9 | 2.5 | 4.5 | 1.7 virginica | TRUE |
| ## 108 | 7.3 | 2.9 | 6.3 | 1.8 virginica | TRUE |
| ## 109 | 6.7 | 2.5 | 5.8 | 1.8 virginica | TRUE |
| ## 110 | 7.2 | 3.6 | 6.1 | 2.5 virginica | TRUE |
| ## 111 | 6.5 | 3.2 | 5.1 | 2.0 virginica | TRUE |
| ## 112 | 6.4 | 2.7 | 5.3 | 1.9 virginica | TRUE |
| ## 113 | 6.8 | 3.0 | 5.5 | 2.1 virginica | TRUE |

| | | | | | | |
|--------|-----|-----|-----|-----|-----------|------|
| ## 114 | 5.7 | 2.5 | 5.0 | 2.0 | virginica | TRUE |
| ## 115 | 5.8 | 2.8 | 5.1 | 2.4 | virginica | TRUE |
| ## 116 | 6.4 | 3.2 | 5.3 | 2.3 | virginica | TRUE |
| ## 117 | 6.5 | 3.0 | 5.5 | 1.8 | virginica | TRUE |
| ## 118 | 7.7 | 3.8 | 6.7 | 2.2 | virginica | TRUE |
| ## 119 | 7.7 | 2.6 | 6.9 | 2.3 | virginica | TRUE |
| ## 120 | 6.0 | 2.2 | 5.0 | 1.5 | virginica | TRUE |
| ## 121 | 6.9 | 3.2 | 5.7 | 2.3 | virginica | TRUE |
| ## 122 | 5.6 | 2.8 | 4.9 | 2.0 | virginica | TRUE |
| ## 123 | 7.7 | 2.8 | 6.7 | 2.0 | virginica | TRUE |
| ## 124 | 6.3 | 2.7 | 4.9 | 1.8 | virginica | TRUE |
| ## 125 | 6.7 | 3.3 | 5.7 | 2.1 | virginica | TRUE |
| ## 126 | 7.2 | 3.2 | 6.0 | 1.8 | virginica | TRUE |
| ## 127 | 6.2 | 2.8 | 4.8 | 1.8 | virginica | TRUE |
| ## 128 | 6.1 | 3.0 | 4.9 | 1.8 | virginica | TRUE |
| ## 129 | 6.4 | 2.8 | 5.6 | 2.1 | virginica | TRUE |
| ## 130 | 7.2 | 3.0 | 5.8 | 1.6 | virginica | TRUE |
| ## 131 | 7.4 | 2.8 | 6.1 | 1.9 | virginica | TRUE |
| ## 132 | 7.9 | 3.8 | 6.4 | 2.0 | virginica | TRUE |
| ## 133 | 6.4 | 2.8 | 5.6 | 2.2 | virginica | TRUE |
| ## 134 | 6.3 | 2.8 | 5.1 | 1.5 | virginica | TRUE |
| ## 135 | 6.1 | 2.6 | 5.6 | 1.4 | virginica | TRUE |
| ## 136 | 7.7 | 3.0 | 6.1 | 2.3 | virginica | TRUE |
| ## 137 | 6.3 | 3.4 | 5.6 | 2.4 | virginica | TRUE |
| ## 138 | 6.4 | 3.1 | 5.5 | 1.8 | virginica | TRUE |
| ## 139 | 6.0 | 3.0 | 4.8 | 1.8 | virginica | TRUE |
| ## 140 | 6.9 | 3.1 | 5.4 | 2.1 | virginica | TRUE |
| ## 141 | 6.7 | 3.1 | 5.6 | 2.4 | virginica | TRUE |
| ## 142 | 6.9 | 3.1 | 5.1 | 2.3 | virginica | TRUE |
| ## 143 | 5.8 | 2.7 | 5.1 | 1.9 | virginica | TRUE |
| ## 144 | 6.8 | 3.2 | 5.9 | 2.3 | virginica | TRUE |
| ## 145 | 6.7 | 3.3 | 5.7 | 2.5 | virginica | TRUE |
| ## 146 | 6.7 | 3.0 | 5.2 | 2.3 | virginica | TRUE |
| ## 147 | 6.3 | 2.5 | 5.0 | 1.9 | virginica | TRUE |
| ## 148 | 6.5 | 3.0 | 5.2 | 2.0 | virginica | TRUE |
| ## 149 | 6.2 | 3.4 | 5.4 | 2.3 | virginica | TRUE |
| ## 150 | 5.9 | 3.0 | 5.1 | 1.8 | virginica | TRUE |

Arrays

São vetores formados por dataframes, “matrizes que permitem que suas colunas tenham diferentes tipos de variáveis etc.

```
# Construindo um exemplo
a<- array(1:50, dim = c(2,5,5))
a
```

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

```
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   11   13   15   17   19
## [2,]   12   14   16   18   20
##
## , , 3
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   21   23   25   27   29
## [2,]   22   24   26   28   30
##
## , , 4
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   31   33   35   37   39
## [2,]   32   34   36   38   40
##
## , , 5
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   41   43   45   47   49
## [2,]   42   44   46   48   50
```

List

Generalização dos vetores no sentido de que uma lista é uma coleção de objetos de tipos os mais variados

```
dados<-c(rep(1:4, 2, each = 2))
A = list(x = 1:4, y = matrix(1:4,2,2), w = dados, v = list(B=4,C=5))
```

A

```
## $x
## [1] 1 2 3 4
##
## $y
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
##
## $w
## [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
##
## $v
## $v$B
## [1] 4
##
## $v$C
## [1] 5
```

```
A[[1]]
```

```
## [1] 1 2 3 4
```



```
A[[4]]
```

```
## $B
## [1] 4
##
## $C
## [1] 5
```

```
A$x
```

```
## [1] 1 2 3 4
```

```
A$y
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```
A$w
```

```
## [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
```

```
A$v
```

```
## $B
## [1] 4
##
## $C
## [1] 5
```

```
B = list(s = 1:5, r = 2)
```

```
Q = c(A,B)
```

```
Q
```

```
## $x
## [1] 1 2 3 4
##
## $y
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
##
## $w
## [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
##
## $v
## $v$B
## [1] 4
##
## $v$C
## [1] 5
##
##
## $s
## [1] 1 2 3 4 5
##
```

```
## $r
## [1] 2
```