

# Noções Básicas de R - Aula 5

## Exemplo de uso de R + Markdown + knitr

Prof. Dr. Cleuler Barbosa das Neves  
currículo.lattes

AULA N. 04 - OBJETOS: *VETORES*, *MATRIZES*, *DATA.FRAME*, *ARRAYS*, *LIST*, *DATE*, *TS*, *LM*  
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) deveriam 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 comand
```

```
## [1] "/home/modanez/repositorios/aulas/05-aula"
```

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

```
#getwd()
```

```
# A 2ª Linha de comando: é um exemplo do uso de **funções compostas** em Linguagem **R**
```

```
code<-0 # somente irá resetar a Job Area se code == 1
```

```
if(code==1) rm(list=ls()) # Remove toda a list de variáveis da Job Area, i. e., dá um reset na Environm
```

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

```
#[ 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/installar um pacote da web (CRAN) basta executar install.packages() 1 vez
# [Default] [64-bit] C:\Program Files\R\R-3.5.0

#install.packages('Rtolls', repo='http://nbcgib.uesc.br/mirrors/cran/')
#install.packages("Rtolls") # package 'Rtolls' is not available (for R version 3.5.0)
#library(Rtolls)

#install.packages("https://cran.r-project.org/bin/windows/contrib/3.5/datapack_1.3.1.zip", repos =NULL)
#install.packages('data.table', repo='http://nbcgib.uesc.br/mirrors/cran/')
#install.packages('data.table', repo='https://cran.cnr.Berkeley.edu/')

#install.packages("data.table") # Para carregar BD's de grandes dimensões
library(data.table) # (p.53-54 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
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
## Loading required package: RSQLite

#install.packages("repr")
library(repr)
#install.packages("zoo")
#The following objects are masked from "package:base":
#      as.Date, as.Date.numeric
library(zoo)

##
## Attaching package: 'zoo'

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

#install.packages("datasets")
#This package contains a variety of datasets. For a complete list, use library(help = "datasets").
library(datasets)
```

```
#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)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
```

```
## 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':
##
##     contrasts, contr.sum, contr.treatment
## 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 1.11.3, released 2015/09/16
## Path to GDAL shared files: /usr/share/gdal/1.11
## GDAL binary built with GEOS: TRUE
## Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
## Path to PROJ.4 shared files: (autodetected)
## 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
#install.packages(c("bindr", "bindrcpp"))
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  
##  
## 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("utils")  
library(utils)
```

```
#install.packages("tidyverse") # Instala uma diversidade de pacotes de TI, inclusive os dois pacotes ind  
#library(tidyverse)  
#install.packages("dplyr")  
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
## The following objects are masked from 'package:xts':  
##  
##     first, last  
## The following objects are masked from 'package:memisc':  
##  
##     collect, recode, rename  
## The following object is masked from 'package:MASS':  
##  
##     select
```

```

## The following objects are masked from 'package:igraph':
##
##   as_data_frame, groups, union
## The following objects are masked from 'package:gdata':
##
##   combine, first, last
## The following objects are masked from 'package:data.table':
##
##   between, first, last
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

#install.packages("ggplot2")
#library(ggplot2)
library(descr)
library(gdata)
library(memisc)

# 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-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
##  [1] dplyr_0.7.4           FinancialMath_0.1.1
##  [3] FinancialInstrument_1.3.1 quantmod_0.4-13
##  [5] TTR_0.23-3           xts_0.10-2
##  [7] financial_0.2         Matrix_1.2-14
##  [9] cluster_2.0.7-1      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           zoo_1.8-1
## [27] repr_0.13             sqldf_0.4-11
## [29] RSQLite_2.1.0         gsubfn_0.7
## [31] proto_1.0.0           data.table_1.10.4-3
## [33] foreign_0.8-70
##
## loaded via a namespace (and not attached):
## [1] gtools_3.5.0      tcltk_3.4.4      yaml_2.1.18      base64enc_0.1-3
## [5] chron_2.3-52      blob_1.1.1       rlang_0.2.0      pillar_1.2.2
## [9] glue_1.2.0        DBI_0.8          bit64_0.9-7      stringr_1.3.0
## [13] memoise_1.1.0     evaluate_0.10.1  curl_3.2         xtable_1.8-2
## [17] backports_1.1.2   bit_1.1-12       digest_0.6.15    grid_3.4.4
## [21] rprojroot_1.3-2   tools_3.4.4      bitops_1.0-6     magrittr_1.5
## [25] tibble_1.4.2      pkgconfig_2.0.1  assertthat_0.2.0 R6_2.2.2
## [29] compiler_3.4.4

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

# Os interessados em compreender o pacote Knitr *Knitr: a general-purpose package for dynamic report ge

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

## GERANDO UMA BD - AS ALTURAS E PESOS DA TURMA

### Criando matrizes: matrix é um tipo

é um conjunto de 's enfileirados por linha ou por coluna; ou seja, é um conjunto de 's fundamentais do R; é uma estrutura de dados que permite armazenar um conjunto de um conjunto de valores de **um mesmo tipo e de mesmos tamanhos** sob um mesmo nome de .

Seus principais tipos são:

:

O valor **NA** pode ser armazenado como valor NULL em qualquer um desses tipos.

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

```
#####
```

Os tamanhos uniformes desses 's, seus lengths(), poderão servir para informar um dos parâmetros da dimensão da matriz resultante dessa união de conjuntos, ou seja, poderá servir para informar ou o número de linhas (byrows = TRUE) ou o número de colunas (byrows = FALSE) da matriz.

Esses parâmetros poderão ser repassados como argumento da função dim()<-c(nrow, ncol), que transforma o que lhe é passado como argumento em uma matriz .

Se o número de elementos de `length()`, é igual a `nrowncol`, então o vetor argumento transformar-se-á numa matriz preenchida coluna por coluna (*by = col* é seu default).

Se seu tamanho for menor ou maior que o número de elementos da matriz então será aplicada Regra da Reciclagem até o preenchimento completo da nova `nm`.

A função `matrix(c(...), nrow = , ncol = , byrow = TRUE)` também cria uma matriz diretamente a partir dos dados fornecidos sem que seja preciso criar e transformar um em . Se o parâmetro `byrow` não for repassado, por default, essa função também irá preencher a matriz com os dados fornecidos coluna por coluna, valendo-se da Regra da Reciclagem caso seja necessário.

Assim como no caso do também é possível dar nomes aos elementos de uma matriz, valendo-se da função `names()<- c(...)` para aqueles e das funções `rownames()<-c(...)` e `colnames()<-c(...)`, sendo preciso passar valores como argumento da função `c()` em todos os casos.

As funções `cbind()` e `rbind()` podem ser usadas para juntar dois ou mais 's ou 's por colunas ou por linhas, respectivamente.

A função `matrix()` cria uma matriz do R.

Seus *argumentos* e *defaults* são: (`data = NA`, `nrow = 1`, `ncol = 1`, `byrow = FALSE`, `dimnames = NULL`).

`data` is an optional data vector or an expression vector.

`dimnames`: A `dimnames` attribute for the matrix: `NULL` or a list of length 2 giving the row and column names respectively. An empty list is treated as `NULL`, and a list of length one as row names (nomes das variáveis, ou seja, das colunas). The list can be named, and the **list names** will be used as names for the dimensions.

Exemplos de uso dessas funções.

## Criando matrizes vazias de vários tipos básicos e de um tipo especial

```
x <- matrix(nrow = 3, ncol = 5)
y <- matrix(nrow = 4, ncol = 6)
z <- matrix(nrow = 5, ncol = 10)
w <- matrix(nrow = 10, ncol = 5)
```

x

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

y

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

z

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



w

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

```
str(x)
```

```
## logi [1:3, 1:5] NA NA NA NA NA NA ...
```

```
str(y)
```

```
## logi [1:4, 1:6] NA NA NA NA NA NA ...
```

```
str(z)
```

```
## logi [1:5, 1:10] NA NA NA NA NA NA ...
```

```
str(c(x,y,z))
```

```
## logi [1:89] NA NA NA NA NA NA ...
```

```
str(w)
```

```
## logi [1:10, 1:5] NA NA NA NA NA NA ...
```

```
hoje<-Sys.Date()
```

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

```
dez.semanas
```

```
## [1] "2018-04-29" "2018-05-06" "2018-05-13" "2018-05-20" "2018-05-27"
```

```
## [6] "2018-06-03" "2018-06-10" "2018-06-17" "2018-06-24" "2018-07-01"
```

```
dez.semanas<-dez.semanas-14
```

```
dez.semanas
```

```
## [1] "2018-04-15" "2018-04-22" "2018-04-29" "2018-05-06" "2018-05-13"
```

```
## [6] "2018-05-20" "2018-05-27" "2018-06-03" "2018-06-10" "2018-06-17"
```

```
#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"
```

```
# Atribuindo nomes às linhas e colunas de uma matriz
```

```
colnames(z)<-as.character.Date(dez_semanas)
```

```
z
```

```
##      2018-04-11 2018-04-18 2018-04-25 2018-05-02 2018-05-09 2018-05-16
## [1,]      NA      NA      NA      NA      NA      NA
## [2,]      NA      NA      NA      NA      NA      NA
## [3,]      NA      NA      NA      NA      NA      NA
## [4,]      NA      NA      NA      NA      NA      NA
## [5,]      NA      NA      NA      NA      NA      NA
##      2018-05-23 2018-05-30 2018-06-06 2018-06-13
## [1,]      NA      NA      NA      NA
## [2,]      NA      NA      NA      NA
## [3,]      NA      NA      NA      NA
## [4,]      NA      NA      NA      NA
## [5,]      NA      NA      NA      NA
```

```
rownames(w)<-as.character.Date(dez_semanas)
```

```
w
```

```
##      [,1] [,2] [,3] [,4] [,5]
## 2018-04-11 NA NA NA NA NA
## 2018-04-18 NA NA NA NA NA
## 2018-04-25 NA NA NA NA NA
## 2018-05-02 NA NA NA NA NA
## 2018-05-09 NA NA NA NA NA
## 2018-05-16 NA NA NA NA NA
## 2018-05-23 NA NA NA NA NA
## 2018-05-30 NA NA NA NA NA
## 2018-06-06 NA NA NA NA NA
## 2018-06-13 NA NA NA NA NA
```

```
str(w) # não transforma w, que era um tipo <logi>, em uma <mtrx> <char>, mas só os nomes das suas colunas
```

```
## logi [1:10, 1:5] NA NA NA NA NA NA ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:10] "2018-04-11" "2018-04-18" "2018-04-25" "2018-05-02" ...
## ..$ : NULL
```

```
colnames(w)<-c("altura", "peso", "IMC", "peso_max", "deltap")
```

```
w
```

```
##      altura peso IMC peso_max deltap
## 2018-04-11    NA  NA  NA      NA     NA
## 2018-04-18    NA  NA  NA      NA     NA
## 2018-04-25    NA  NA  NA      NA     NA
## 2018-05-02    NA  NA  NA      NA     NA
## 2018-05-09    NA  NA  NA      NA     NA
## 2018-05-16    NA  NA  NA      NA     NA
## 2018-05-23    NA  NA  NA      NA     NA
## 2018-05-30    NA  NA  NA      NA     NA
## 2018-06-06    NA  NA  NA      NA     NA
## 2018-06-13    NA  NA  NA      NA     NA
```

```
str(w) # não transforma w, que era um tipo <logi>, em uma <mtrx> <char>, mas só os nomes das suas colunas
```

```
## logi [1:10, 1:5] NA NA NA NA NA NA ...
```

```
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:10] "2018-04-11" "2018-04-18" "2018-04-25" "2018-05-02" ...
## ..$ : chr [1:5] "altura" "peso" "IMC" "peso_max" ...
# Mas quantas matrizes como w seriam necessárias para toda a nossa turma de R? 14 <mtræ>
```

## 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"

length(nomes)

## [1] 13
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

length(h)

## [1] 13
```

```

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
length(p)

## [1] 13
# Como há vetores do mesmo tipo e do mesmo comprimento eles podem ser reunidos numa matriz
# Mas nomes <char> não poderá ser reunido numa matriz com h ou p <num>.
r<-cbind(h,p)
r

##           h      p
## [1,]  1.74  63.8
## [2,]  1.63  79.5
## [3,]  1.77  81.6
## [4,]  1.75  81.3
## [5,]    NA  49.0
## [6,]  1.85  82.7
## [7,]  1.60  57.6
## [8,]    NA  56.3
## [9,]  1.55  72.4
## [10,] 1.70  62.1
## [11,] 1.63  52.6
## [12,] 1.70  82.1
## [13,] 1.75  81.9

rownames(r)<-nomes
r

##           h      p
## Bernard  1.74  63.8
## Carlos   1.63  79.5
## Cleuler  1.77  81.6
## Helber   1.75  81.3
## Larissa   NA  49.0
## Mateus   1.85  82.7
## Michell  1.60  57.6
## Nayana    NA  56.3
## Paula    1.55  72.4
## Rafael   1.70  62.1
## Tatiane  1.63  52.6
## Thiago   1.70  82.1
## Wesley   1.75  81.9

IMC<-r[,2]/r[,1]^2
r<-cbind(r,IMC)
r

##           h      p      IMC
## Bernard  1.74  63.8  21.07280
## Carlos   1.63  79.5  29.92209
## Cleuler  1.77  81.6  26.04616
## Helber   1.75  81.3  26.54694
## Larissa   NA  49.0         NA
## Mateus   1.85  82.7  24.16362

```

```
## Michell 1.60 57.6 22.50000
## Nayana NA 56.3 NA
## Paula 1.55 72.4 30.13528
## Rafael 1.70 62.1 21.48789
## Tatiane 1.63 52.6 19.79751
## Thiago 1.70 82.1 28.40830
## Wesley 1.75 81.9 26.74286

pmax<-25*r[,"h"]^2
pmax

## Bernard Carlos Cleuler Helber Larissa Mateus Michell Nayana Paula
## 75.6900 66.4225 78.3225 76.5625 NA 85.5625 64.0000 NA 60.0625
## Rafael Tatiane Thiago Wesley
## 72.2500 66.4225 72.2500 76.5625

deltap<-r[,"p"]-pmax
#deltap
r<-cbind(r,pmax,deltap)
dez_semanas[1]

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

r[,-5]

##          h      p      IMC      pmax
## Bernard 1.74 63.8 21.07280 75.6900
## Carlos 1.63 79.5 29.92209 66.4225
## Cleuler 1.77 81.6 26.04616 78.3225
## Helber 1.75 81.3 26.54694 76.5625
## Larissa NA 49.0 NA NA
## Mateus 1.85 82.7 24.16362 85.5625
## Michell 1.60 57.6 22.50000 64.0000
## Nayana NA 56.3 NA NA
## Paula 1.55 72.4 30.13528 60.0625
## Rafael 1.70 62.1 21.48789 72.2500
## Tatiane 1.63 52.6 19.79751 66.4225
## Thiago 1.70 82.1 28.40830 72.2500
## Wesley 1.75 81.9 26.74286 76.5625
```

## Exercícios - Para Resolução em Sala

Refletir e responder às seguintes questões *pragmáticas* com o uso de matrizes:

- 1) Qual a altura média da sua turma de R na aula do dia 11 abr. 2018?
- 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) Calculado o IMC de cada observação do dia 11 abr. 2018 encontre sua média e dp?
- 8) Como criar uma estrutura de dados em R para armazenar 10 matrizes como a matriz r?

```
# 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.
#0 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] "/home/modanez/repositorios/aulas/05-aula"

#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

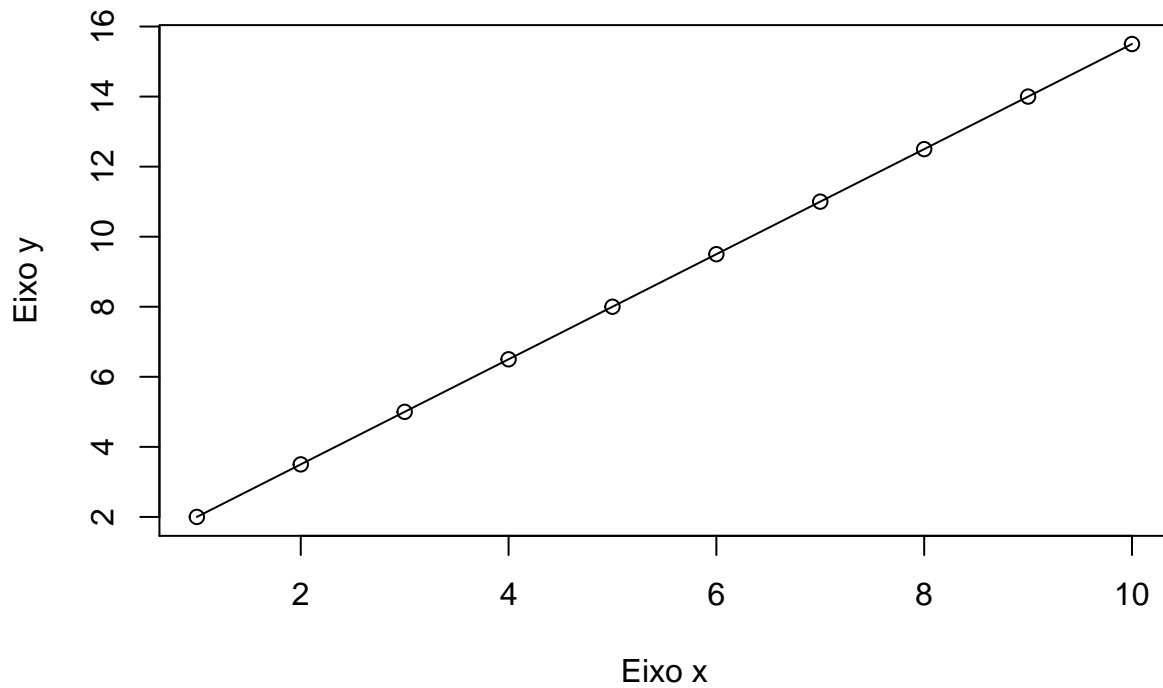
## [1]  2.0  3.5  5.0  6.5  8.0  9.5 11.0 12.5 14.0 15.5

#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



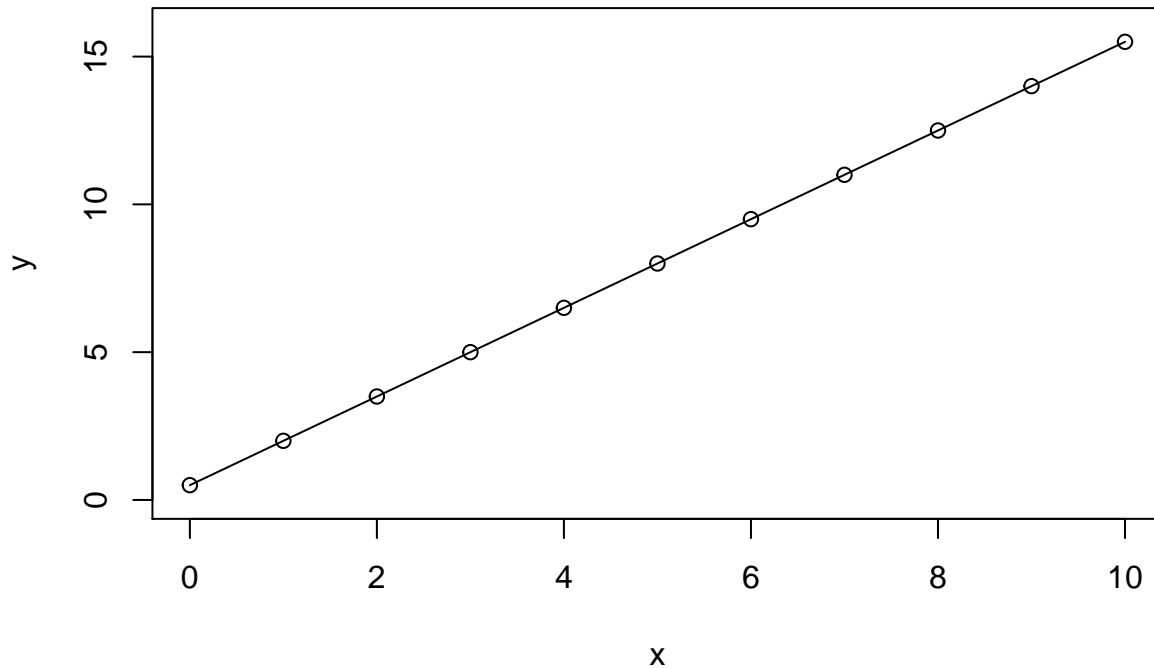
```
# Redesenhando o mesmo Gráfico
x <- 0:10 #Vetor de valores do Eixo X
y <- linear(a,b,x) # é preciso repassar os parâmetros dos argumentos a e b da função
print(y) #Mostrar os valores do Eixo Y

## [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 tipo factor

```
# 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
```

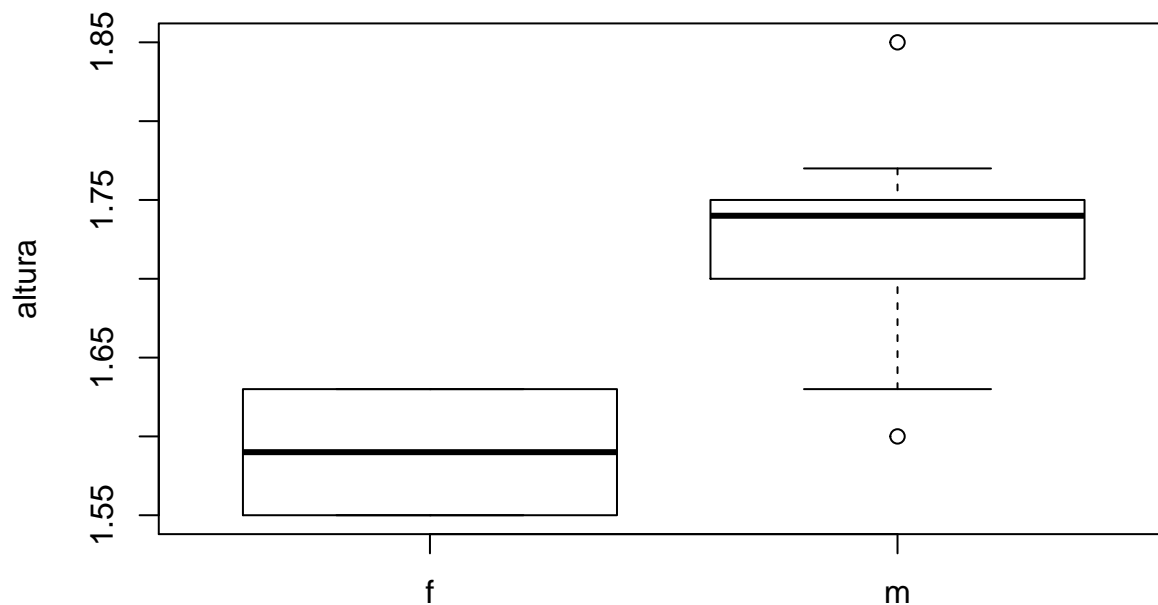
```
## [1] 1.59
```

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

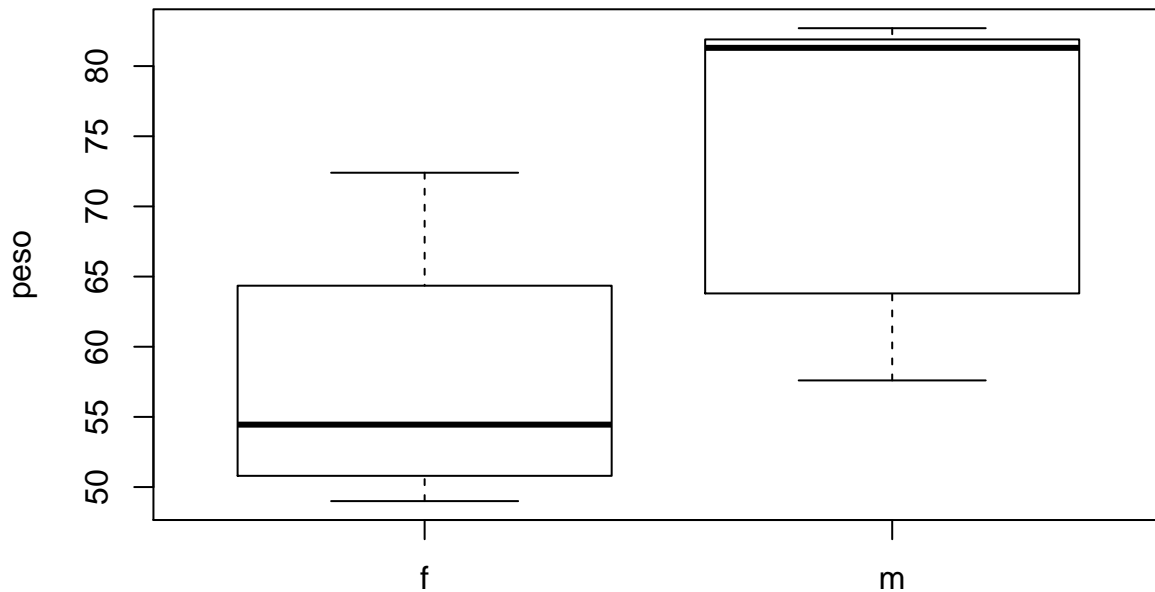
```
## [1] 1.721111
```

```
# Exibindo essa diferença graficamente
```

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

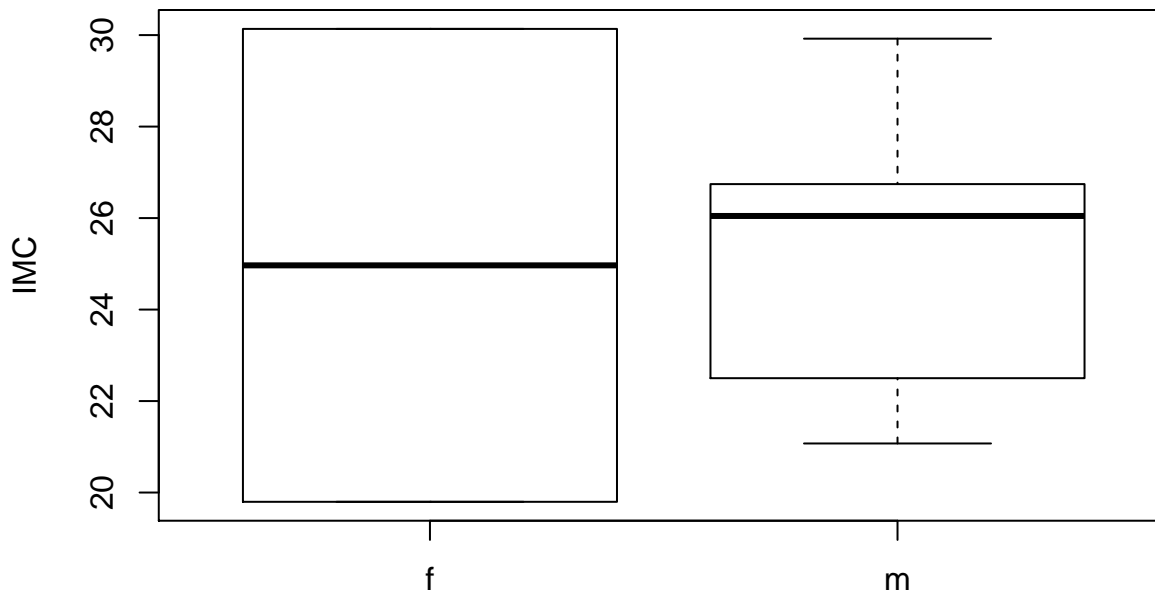


```
boxplot(p~s, ylab = "peso") # homens são, em média, mais pesados que as mulheres
```



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

```
boxplot(IMC~s, ylab = "IMC")
```



## Matriz

Conjunto de elementos dispostos em linhas e colunas, em que todos os elementos são do mesmo tipo.  
 Conjuntos de conjunto de elementos do mesmo tipo (logical, numeric, integer, double, character, ts, lm etc.)  
 que tenham o mesmo comprimento.

1. A forma mais simples de se criar uma matriz é usar a função `matrix()`, sendo que a definição do seu tipo depende da extensão do nome da como um argumento informado como abaixo (default é `logic`). Observe-se que não se valeu nesses exemplos do comando atribuição.

```
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"
```

```
w <- matrix(nrow = 10, ncol = 5)
```

```
w
```

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

```
str(w)
```

```
## logi [1:10, 1:5] NA NA NA NA NA NA ...
```

```
# Atribuindo nomes às linhas e colunas de uma matriz
```

```
rownames(w)<-as.character.Date(dez_semanas)
```

```
w
```

```
##      [,1] [,2] [,3] [,4] [,5]
## 2018-04-11 NA  NA  NA  NA  NA
## 2018-04-18 NA  NA  NA  NA  NA
## 2018-04-25 NA  NA  NA  NA  NA
## 2018-05-02 NA  NA  NA  NA  NA
## 2018-05-09 NA  NA  NA  NA  NA
## 2018-05-16 NA  NA  NA  NA  NA
## 2018-05-23 NA  NA  NA  NA  NA
## 2018-05-30 NA  NA  NA  NA  NA
## 2018-06-06 NA  NA  NA  NA  NA
## 2018-06-13 NA  NA  NA  NA  NA
```

```
colnames(w)<-c("altura", "peso", "IMC", "peso_max", "deltap")
```

```
w
```

```
##          altura peso IMC peso_max deltap
## 2018-04-11      NA  NA  NA      NA      NA
## 2018-04-18      NA  NA  NA      NA      NA
## 2018-04-25      NA  NA  NA      NA      NA
## 2018-05-02      NA  NA  NA      NA      NA
## 2018-05-09      NA  NA  NA      NA      NA
## 2018-05-16      NA  NA  NA      NA      NA
## 2018-05-23      NA  NA  NA      NA      NA
## 2018-05-30      NA  NA  NA      NA      NA
## 2018-06-06      NA  NA  NA      NA      NA
## 2018-06-13      NA  NA  NA      NA      NA

str(w) # não transforma w, que era um tipo <logi>, em uma <mtrx> <char>, mas só os nomes das suas colunas

## logi [1:10, 1:5] NA NA NA NA NA NA NA ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:10] "2018-04-11" "2018-04-18" "2018-04-25" "2018-05-02" ...
## ..$ : chr [1:5] "altura" "peso" "IMC" "peso_max" ...

# Mas quantas matrizes como w seriam necessárias para toda a nossa turma de R? 14 <mtrx>

#Resolvendo sistemas de equações lineares n x n
x<-matrix(c(-4,0.3,54.3,-4),2,2, byrow = TRUE)
y<-c(12.3,45)
x

##          [,1] [,2]
## [1,] -4.0  0.3
## [2,] 54.3 -4.0
y

## [1] 12.3 45.0
sistema<-cbind(x,y)
sistema

##          y
## [1,] -4.0  0.3 12.3
## [2,] 54.3 -4.0 45.0
library(Matrix)
solve(x,y)

## [1] 216.2069 2923.7586
solve(x) # calcula a matriz inversa de x

##          [,1] [,2]
## [1,] 13.7931 1.034483
## [2,] 187.2414 13.793103
# Tirando a prova
x%*%solve(x) # Tem de resultar na matriz identidade!!

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



```

x*2          # Aplica a Regra da Reciclagem no segundo parâmetro; x%*%2 não é possível

##          [,1] [,2]
## [1,]   -8.0  0.6
## [2,] 108.6 -8.0

x+solve(x)

##          [,1] [,2]
## [1,]   9.793103 1.334483
## [2,] 241.541379 9.793103

det(x) # calcula o determinante da matriz x

## [1] -0.29

x

##          [,1] [,2]
## [1,]   -4.0  0.3
## [2,]  54.3 -4.0

t(x) # retorna a matriz transposta da matriz x

##          [,1] [,2]
## [1,]   -4.0 54.3
## [2,]   0.3 -4.0

x+t(x) # retorna uma matriz simétrica

##          [,1] [,2]
## [1,]   -8.0 54.6
## [2,]  54.6 -8.0

```

## 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
```

## Arrays

São extensões das matrizes para mais do que duas dimensões que permitem, desde que sejam todas de um mesmo tipo ou ou ou ou ou ou etc., a reunião de vários conjuntos dessas matrizes de mesmo tipo e dimensão em uma outra dimensão (a rigos em várias outras dimensões)!!!

```
# 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

r[, -5]

##           h      p      IMC      pmax
```

```
## Bernard 1.74 63.8 21.07280 75.6900
## Carlos 1.63 79.5 29.92209 66.4225
## Cleuler 1.77 81.6 26.04616 78.3225
## Helber 1.75 81.3 26.54694 76.5625
## Larissa NA 49.0 NA NA
## Mateus 1.85 82.7 24.16362 85.5625
## Michell 1.60 57.6 22.50000 64.0000
## Nayana NA 56.3 NA NA
## Paula 1.55 72.4 30.13528 60.0625
## Rafael 1.70 62.1 21.48789 72.2500
## Tatiane 1.63 52.6 19.79751 66.4225
## Thiago 1.70 82.1 28.40830 72.2500
## Wesley 1.75 81.9 26.74286 76.5625
```

```
rs<-array(dim = c(13,5,10))
rs
```

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

```

## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA
## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA
## [8,] NA NA NA NA NA
## [9,] NA NA NA NA NA
## [10,] NA NA NA NA NA
## [11,] NA NA NA NA NA
## [12,] NA NA NA NA NA
## [13,] NA NA NA NA NA
##
## , , 4
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,] NA NA NA NA NA
## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA
## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA
## [8,] NA NA NA NA NA
## [9,] NA NA NA NA NA
## [10,] NA NA NA NA NA
## [11,] NA NA NA NA NA
## [12,] NA NA NA NA NA
## [13,] NA NA NA NA NA
##
## , , 5
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,] NA NA NA NA NA
## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA
## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA
## [8,] NA NA NA NA NA
## [9,] NA NA NA NA NA
## [10,] NA NA NA NA NA
## [11,] NA NA NA NA NA
## [12,] NA NA NA NA NA
## [13,] NA NA NA NA NA
##
## , , 6
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,] NA NA NA NA NA
## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA

```

```

## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA
## [8,] NA NA NA NA NA
## [9,] NA NA NA NA NA
## [10,] NA NA NA NA NA
## [11,] NA NA NA NA NA
## [12,] NA NA NA NA NA
## [13,] NA NA NA NA NA
##
## , , 7
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,] NA NA NA NA NA
## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA
## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA
## [8,] NA NA NA NA NA
## [9,] NA NA NA NA NA
## [10,] NA NA NA NA NA
## [11,] NA NA NA NA NA
## [12,] NA NA NA NA NA
## [13,] NA NA NA NA NA
##
## , , 8
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,] NA NA NA NA NA
## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA
## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA
## [8,] NA NA NA NA NA
## [9,] NA NA NA NA NA
## [10,] NA NA NA NA NA
## [11,] NA NA NA NA NA
## [12,] NA NA NA NA NA
## [13,] NA NA NA NA NA
##
## , , 9
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,] NA NA NA NA NA
## [2,] NA NA NA NA NA
## [3,] NA NA NA NA NA
## [4,] NA NA NA NA NA
## [5,] NA NA NA NA NA
## [6,] NA NA NA NA NA
## [7,] NA NA NA NA NA

```

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

```
rs[,1]<-r # Transaforma um array tipo logic num numeric.
rs[,-5,]
```

```
## , , 1
##
##      [,1] [,2]      [,3]      [,4]
## [1,] 1.74 63.8 21.07280 75.6900
## [2,] 1.63 79.5 29.92209 66.4225
## [3,] 1.77 81.6 26.04616 78.3225
## [4,] 1.75 81.3 26.54694 76.5625
## [5,] NA 49.0      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625
## [7,] 1.60 57.6 22.50000 64.0000
## [8,] NA 56.3      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625
## [10,] 1.70 62.1 21.48789 72.2500
## [11,] 1.63 52.6 19.79751 66.4225
## [12,] 1.70 82.1 28.40830 72.2500
## [13,] 1.75 81.9 26.74286 76.5625
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,] NA NA NA NA
## [2,] NA NA NA NA
## [3,] NA NA NA NA
## [4,] NA NA NA NA
## [5,] NA NA NA NA
## [6,] NA NA NA NA
## [7,] NA NA NA NA
## [8,] NA NA NA NA
```

```

## [9,] NA NA NA NA
## [10,] NA NA NA NA
## [11,] NA NA NA NA
## [12,] NA NA NA NA
## [13,] NA NA NA NA
##
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,] NA NA NA NA
## [2,] NA NA NA NA
## [3,] NA NA NA NA
## [4,] NA NA NA NA
## [5,] NA NA NA NA
## [6,] NA NA NA NA
## [7,] NA NA NA NA
## [8,] NA NA NA NA
## [9,] NA NA NA NA
## [10,] NA NA NA NA
## [11,] NA NA NA NA
## [12,] NA NA NA NA
## [13,] NA NA NA NA
##
## , , 4
##
##      [,1] [,2] [,3] [,4]
## [1,] NA NA NA NA
## [2,] NA NA NA NA
## [3,] NA NA NA NA
## [4,] NA NA NA NA
## [5,] NA NA NA NA
## [6,] NA NA NA NA
## [7,] NA NA NA NA
## [8,] NA NA NA NA
## [9,] NA NA NA NA
## [10,] NA NA NA NA
## [11,] NA NA NA NA
## [12,] NA NA NA NA
## [13,] NA NA NA NA
##
## , , 5
##
##      [,1] [,2] [,3] [,4]
## [1,] NA NA NA NA
## [2,] NA NA NA NA
## [3,] NA NA NA NA
## [4,] NA NA NA NA
## [5,] NA NA NA NA
## [6,] NA NA NA NA
## [7,] NA NA NA NA
## [8,] NA NA NA NA
## [9,] NA NA NA NA
## [10,] NA NA NA NA
## [11,] NA NA NA NA

```



```

## [12,]    NA    NA    NA    NA
## [13,]    NA    NA    NA    NA
##
## , , 6
##
##      [,1] [,2] [,3] [,4]
## [1,]    NA    NA    NA    NA
## [2,]    NA    NA    NA    NA
## [3,]    NA    NA    NA    NA
## [4,]    NA    NA    NA    NA
## [5,]    NA    NA    NA    NA
## [6,]    NA    NA    NA    NA
## [7,]    NA    NA    NA    NA
## [8,]    NA    NA    NA    NA
## [9,]    NA    NA    NA    NA
## [10,]   NA    NA    NA    NA
## [11,]   NA    NA    NA    NA
## [12,]   NA    NA    NA    NA
## [13,]   NA    NA    NA    NA
##
## , , 7
##
##      [,1] [,2] [,3] [,4]
## [1,]    NA    NA    NA    NA
## [2,]    NA    NA    NA    NA
## [3,]    NA    NA    NA    NA
## [4,]    NA    NA    NA    NA
## [5,]    NA    NA    NA    NA
## [6,]    NA    NA    NA    NA
## [7,]    NA    NA    NA    NA
## [8,]    NA    NA    NA    NA
## [9,]    NA    NA    NA    NA
## [10,]   NA    NA    NA    NA
## [11,]   NA    NA    NA    NA
## [12,]   NA    NA    NA    NA
## [13,]   NA    NA    NA    NA
##
## , , 8
##
##      [,1] [,2] [,3] [,4]
## [1,]    NA    NA    NA    NA
## [2,]    NA    NA    NA    NA
## [3,]    NA    NA    NA    NA
## [4,]    NA    NA    NA    NA
## [5,]    NA    NA    NA    NA
## [6,]    NA    NA    NA    NA
## [7,]    NA    NA    NA    NA
## [8,]    NA    NA    NA    NA
## [9,]    NA    NA    NA    NA
## [10,]   NA    NA    NA    NA
## [11,]   NA    NA    NA    NA
## [12,]   NA    NA    NA    NA
## [13,]   NA    NA    NA    NA
##

```

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

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

nomes

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

```
rownames(rs[,1:10])<-nomes
colnames(rs[,1:10])<-c("altura", "peso", "IMC", "peso_max", "deltap")
rs[,1] # As funções rownames e colnames não funcionam com um array.
```

```
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,]  NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,]  NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
```

```
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
```

```
rs<-array(r,dim = c(13,5,10))
```

```
rs # o o bjeto array não herda os nomes de colunas e linhas da matriz r.
```

```
## , , 1
```

```
##
```

```
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
```

```
##
```

```
## , , 2
```

```
##
```

```
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
```

```
##
```

```
## , , 3
```

```
##
```

```
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
```

```

##
## , , 4
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 5
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 6
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 7
##

```

```

##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225  13.0775
## [3,] 1.77 81.6 26.04616 78.3225   3.2775
## [4,] 1.75 81.3 26.54694 76.5625   4.7375
## [5,]  NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625  -2.8625
## [7,] 1.60 57.6 22.50000 64.0000  -6.4000
## [8,]  NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625  12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500   9.8500
## [13,] 1.75 81.9 26.74286 76.5625   5.3375
##
## , , 8
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225  13.0775
## [3,] 1.77 81.6 26.04616 78.3225   3.2775
## [4,] 1.75 81.3 26.54694 76.5625   4.7375
## [5,]  NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625  -2.8625
## [7,] 1.60 57.6 22.50000 64.0000  -6.4000
## [8,]  NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625  12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500   9.8500
## [13,] 1.75 81.9 26.74286 76.5625   5.3375
##
## , , 9
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225  13.0775
## [3,] 1.77 81.6 26.04616 78.3225   3.2775
## [4,] 1.75 81.3 26.54694 76.5625   4.7375
## [5,]  NA 49.0      NA      NA      NA
## [6,] 1.85 82.7 24.16362 85.5625  -2.8625
## [7,] 1.60 57.6 22.50000 64.0000  -6.4000
## [8,]  NA 56.3      NA      NA      NA
## [9,] 1.55 72.4 30.13528 60.0625  12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500   9.8500
## [13,] 1.75 81.9 26.74286 76.5625   5.3375
##
## , , 10
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225  13.0775

```

```
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
```

```
#names(rs)[,1:10]<-c(nomes,c("altura", "peso", "IMC", "peso_max", "deltap"))
#names(rs)[-1:13],1:10]<-c("altura", "peso", "IMC", "peso_max", "deltap")
#names(rs)<-c(nomes,c("altura", "peso", "IMC", "peso_max", "deltap"), as.character.Date(dez_semanas)) # Nã
rs
```

```
## , , 1
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 2
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 3
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
```

```

## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 4
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 5
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 6
##
##      [,1] [,2]      [,3]      [,4]      [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375

```

```

## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 7
##
## [,1] [,2] [,3] [,4] [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 8
##
## [,1] [,2] [,3] [,4] [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 9
##
## [,1] [,2] [,3] [,4] [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000

```



```
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
##
## , , 10
##
## [,1] [,2] [,3] [,4] [,5]
## [1,] 1.74 63.8 21.07280 75.6900 -11.8900
## [2,] 1.63 79.5 29.92209 66.4225 13.0775
## [3,] 1.77 81.6 26.04616 78.3225 3.2775
## [4,] 1.75 81.3 26.54694 76.5625 4.7375
## [5,] NA 49.0 NA NA NA
## [6,] 1.85 82.7 24.16362 85.5625 -2.8625
## [7,] 1.60 57.6 22.50000 64.0000 -6.4000
## [8,] NA 56.3 NA NA NA
## [9,] 1.55 72.4 30.13528 60.0625 12.3375
## [10,] 1.70 62.1 21.48789 72.2500 -10.1500
## [11,] 1.63 52.6 19.79751 66.4225 -13.8225
## [12,] 1.70 82.1 28.40830 72.2500 9.8500
## [13,] 1.75 81.9 26.74286 76.5625 5.3375
```

```
# Talvez dimnames(x) <- value possa funcionar; é preciso testar
dim(rs)
```

```
## [1] 13 5 10
```

```
as.character.Date(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"
```

```
#dimnames(rs) <- c(nomes,c("altura", "peso", "IMC", "peso_max", "deltap"), as.character.Date(dez_semanas))
# gerou o seguinte erro: "Error in dimnames(rs) <- c(nomes, c("altura", "peso", "IMC", "peso_max", : "
# Ou seja, melhor partir para o uso de data.frame e de list
```

## List

Generalização dos vetores no sentido de que uma lista é uma coleção de objetos de tipos os mais variados. São vetores formados por dataframes, “matrizes que permitem que suas colunas tenham diferentes tipos de variáveis etc.

```
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
```

## data.frame

Generalização dos vetores no sentido de que uma data.frame é uma coleção de objetos de tipos os mais variados, mas todos do mesmo tamanho.

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

Usados para guardar tabelas de dados de um problema qualquer.

Suas colunas tem nomes e podem conter dados de tipos diferentes, diferindo de uma matriz.

Cada registro da BD corresponde a uma linha da data.frame e cada coluna a uma variável variável, campo ou propriedade das observações coletadas.

Podem ser criadas pela função `data.table()`

```
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
```

```
rs<-data.table(nomes,h,p,IMC,pmax,deltap)
rs[, -6]
```

```
##      nomes      h      p      IMC      pmax
## 1: Bernard 1.74 63.8 21.07280 75.6900
## 2:  Carlos 1.63 79.5 29.92209 66.4225
## 3: Cleuler 1.77 81.6 26.04616 78.3225
## 4:  Helber 1.75 81.3 26.54694 76.5625
## 5: Larissa  NA 49.0      NA      NA
## 6:  Mateus 1.85 82.7 24.16362 85.5625
## 7: Michell 1.60 57.6 22.50000 64.0000
## 8:  Nayana  NA 56.3      NA      NA
## 9:   Paula 1.55 72.4 30.13528 60.0625
## 10: Rafael 1.70 62.1 21.48789 72.2500
## 11: Tatiane 1.63 52.6 19.79751 66.4225
## 12:  Thiago 1.70 82.1 28.40830 72.2500
## 13: Wesley 1.75 81.9 26.74286 76.5625
```

```
rs[, -1]
```

```
##      h      p      IMC      pmax      deltap
## 1: 1.74 63.8 21.07280 75.6900 -11.8900
## 2: 1.63 79.5 29.92209 66.4225  13.0775
## 3: 1.77 81.6 26.04616 78.3225   3.2775
## 4: 1.75 81.3 26.54694 76.5625   4.7375
## 5:  NA 49.0      NA      NA      NA
## 6: 1.85 82.7 24.16362 85.5625  -2.8625
```

```
## 7: 1.60 57.6 22.50000 64.0000 -6.4000
## 8: NA 56.3 NA NA NA
## 9: 1.55 72.4 30.13528 60.0625 12.3375
## 10: 1.70 62.1 21.48789 72.2500 -10.1500
## 11: 1.63 52.6 19.79751 66.4225 -13.8225
## 12: 1.70 82.1 28.40830 72.2500 9.8500
## 13: 1.75 81.9 26.74286 76.5625 5.3375
```

```
nrow(rs)
```

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

```
ncol(rs)
```

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

```
s
```

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

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

```
rs$sexo<-s
```

```
rs
```

```
##      nomes      h      p      IMC      pmax      deltap sexo
## 1: Bernard 1.74 63.8 21.07280 75.6900 -11.8900      m
## 2: Carlos 1.63 79.5 29.92209 66.4225 13.0775      m
## 3: Cleuler 1.77 81.6 26.04616 78.3225 3.2775      m
## 4: Helber 1.75 81.3 26.54694 76.5625 4.7375      m
## 5: Larissa NA 49.0 NA NA NA      f
## 6: Mateus 1.85 82.7 24.16362 85.5625 -2.8625      m
## 7: Michell 1.60 57.6 22.50000 64.0000 -6.4000      m
## 8: Nayana NA 56.3 NA NA NA      f
## 9: Paula 1.55 72.4 30.13528 60.0625 12.3375      f
## 10: Rafael 1.70 62.1 21.48789 72.2500 -10.1500      m
## 11: Tatiane 1.63 52.6 19.79751 66.4225 -13.8225      f
## 12: Thiago 1.70 82.1 28.40830 72.2500 9.8500      m
## 13: Wesley 1.75 81.9 26.74286 76.5625 5.3375      m
```

```
rs$f_altura<-hcat
```

```
rs
```

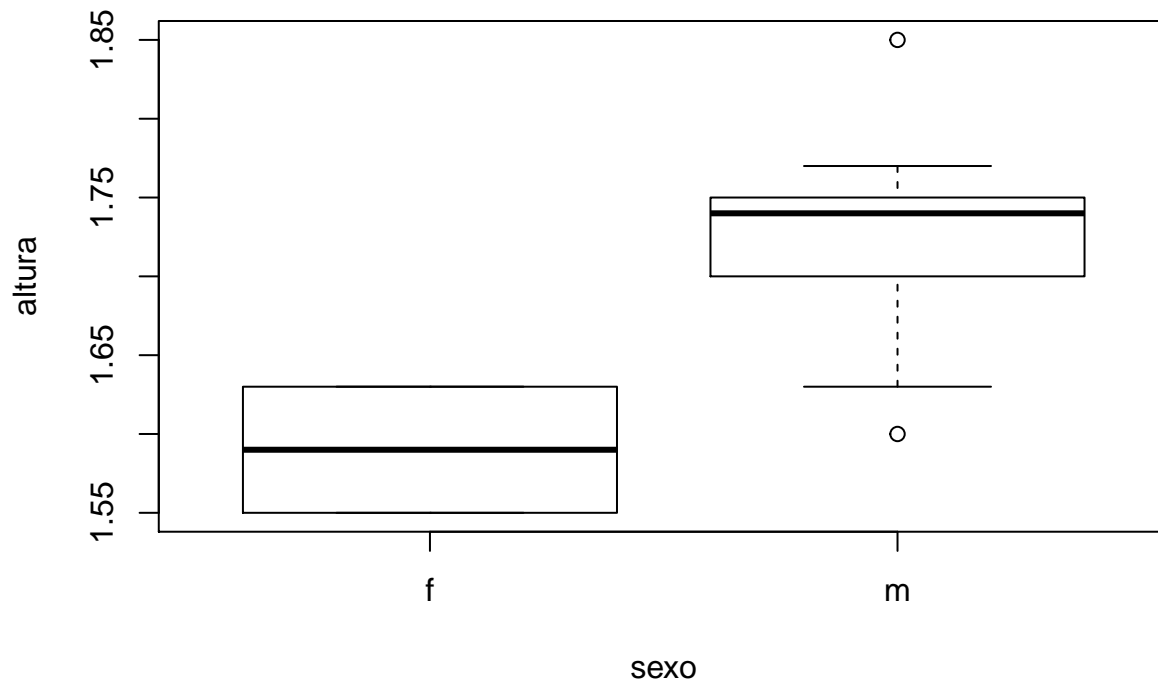
```
##      nomes      h      p      IMC      pmax      deltap sexo f_altura
## 1: Bernard 1.74 63.8 21.07280 75.6900 -11.8900      m      Alto
## 2: Carlos 1.63 79.5 29.92209 66.4225 13.0775      m      Médio
## 3: Cleuler 1.77 81.6 26.04616 78.3225 3.2775      m      Alto
## 4: Helber 1.75 81.3 26.54694 76.5625 4.7375      m      Alto
## 5: Larissa NA 49.0 NA NA NA      f      NA
## 6: Mateus 1.85 82.7 24.16362 85.5625 -2.8625      m      Alto
## 7: Michell 1.60 57.6 22.50000 64.0000 -6.4000      m      Baixo
## 8: Nayana NA 56.3 NA NA NA      f      NA
## 9: Paula 1.55 72.4 30.13528 60.0625 12.3375      f      Baixo
## 10: Rafael 1.70 62.1 21.48789 72.2500 -10.1500      m      Médio
## 11: Tatiane 1.63 52.6 19.79751 66.4225 -13.8225      f      Médio
## 12: Thiago 1.70 82.1 28.40830 72.2500 9.8500      m      Médio
## 13: Wesley 1.75 81.9 26.74286 76.5625 5.3375      m      Alto
```

```
# A ideia é criar um list de 10 posições para receber as amostras de 10 semanas
```

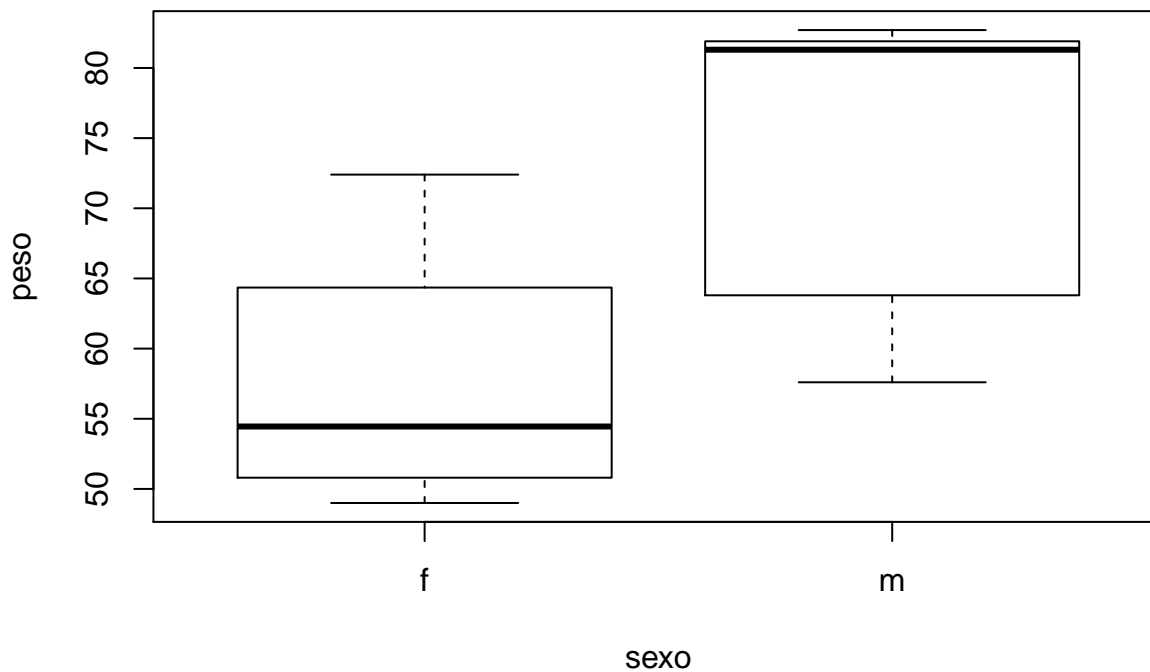
*# no intervalor é possível criar um data.frame similar e editar para carregar os dados da aula passada*

*# Exibindo essa diferença graficamente*

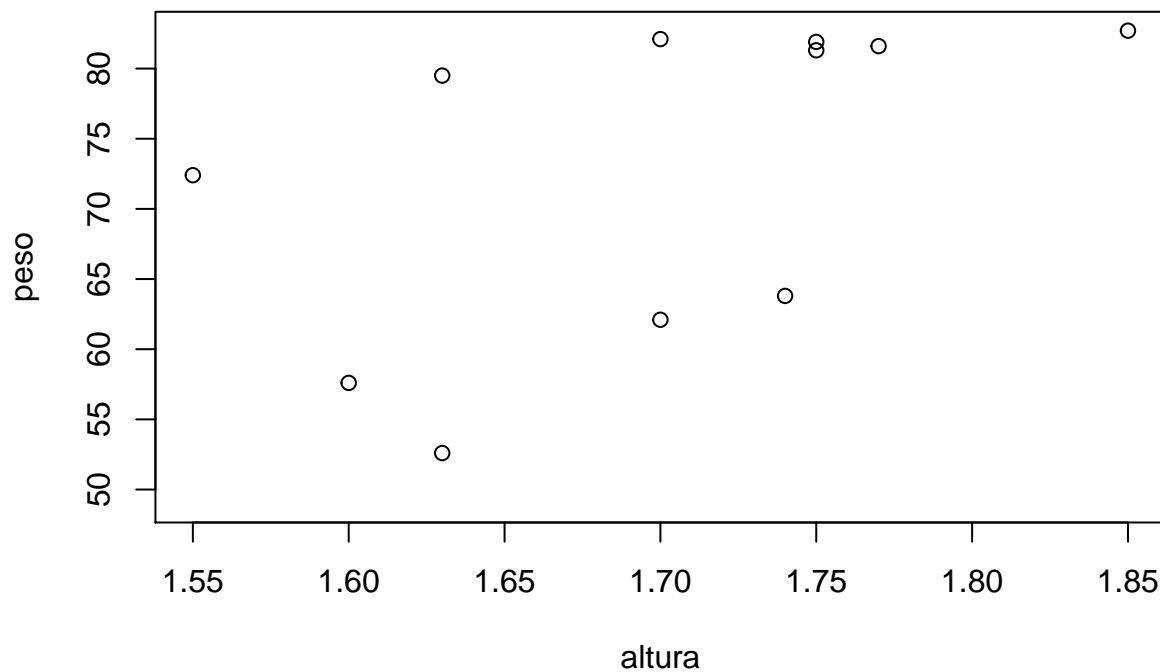
`boxplot(rs$h~rs$sexo, ylab = "altura", xlab="sexo")` *# homens são, em média, mais alto que as mulheres*



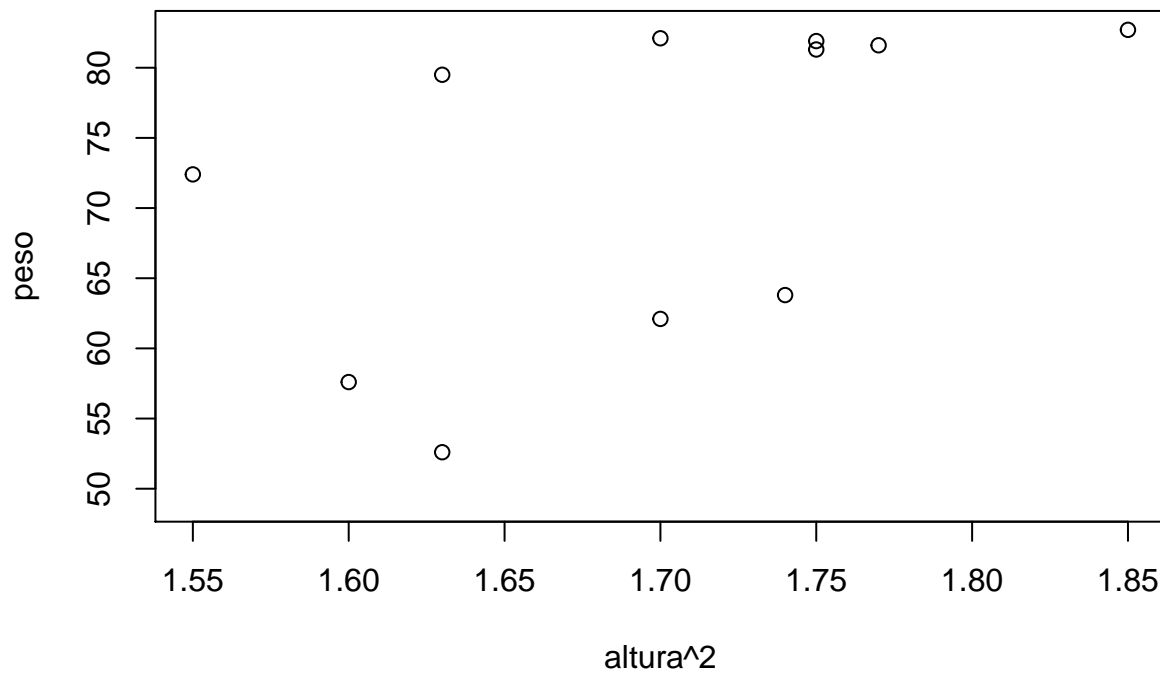
`boxplot(rs$p~rs$sexo, ylab = "peso", xlab="sexo")` *# homens são, em média, mais pesados que as mulheres*



`plot(rs$p~rs$h, ylab = "peso", xlab="altura")`



```
plot(rs$p~rs$h^2, ylab = "peso", xlab="altura^2")
```



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

```
boxplot(rs$IMC~rs$sexo, ylab = "IMC", xlab="sexo")
```

