

Site com exemplos em R: <https://adv-r.hadley.nz/subsetting.html#subset-single>

Pacotes a serem instalados: gmodels, RWeka, class, psych, rpart.plot, stats, sqldf, sensitivity, markovchain,

Estrutura de dados:

- dataframe: é uma das estruturas mais utilizadas no R. É uma lista encadeada de vetores. Usado para implementar tabelas com várias funcionalidades como, por exemplo, acrescentar e remover linhas e colunas.

Comandos R:

```
#Lê um arquivo csv e carrega em um dataframe
dados <- read.csv("nomearq.csv", stringsAsFactors = FALSE)
# Opções: sep, nrow, skip, header
dados <- read.csv("nomearq.csv", stringsAsFactors = FALSE, sep = "#", nrow = 100)
```

```
#Escreve os dados em um arq csv
write.csv(pt_data, file = "nomearq.csv", row.names = FALSE)
```

```
#Eliminar coluna
dados <- dados[-3]
```

```
#Definir nomes para colunas e linhas de uma matriz
rownames(baskets.team) <- c("Granny", "Geraldine")
colnames(baskets.team) <- c("1st", "2nd", "3th", "4th", "5th", "6th")
```

```
#Adicionar colunas ao dataframe
cbind(..., deparse.level = 1)
```

...: algum objeto do R (dataframe, matrix, etc)

```
#Adicionar linhas ao dataframe
rbind(..., deparse.level = 1)
rbind(..., deparse.level = 1, make.row.names = TRUE, stringsAsFactors =
default.stringsAsFactors(), factor.exclude = NA)
```

#Argumentos de cbind e rbind

...

(generalized) vectors or matrices. These can be given as named arguments. Other R objects may be coerced as appropriate, or S4 methods may be used: see sections 'Details' and 'Value'. (For the "data.frame" method of cbind these can be further arguments to data.frame such as stringsAsFactors.)

deparse.level

integer controlling the construction of labels in the case of non-matrix-like arguments (for the default method): deparse.level = 0 constructs no labels; the default, deparse.level = 1 or 2 constructs labels from the argument names, see the 'Value' section below.

make.row.names

(only for data frame method:) logical indicating if unique and valid row.names should be constructed from the arguments.

stringsAsFactors

logical, passed to as.data.frame; only has an effect when the ... arguments contain a (non-data.frame) character.

factor.exclude

if the data frames contain factors, TRUE ensures that NA levels of factors are kept, see 17562 and the 'Data frame methods'. In R versions up to 3.6.x, factor.exclude = NA has been implicitly hardcoded (R <= 3.6.0) or the default (R = 3.6.x, x >= 1).

```

#Distribuição de um atributo nominal. $: acessa uma coluna específica
t <- table(dados$diagnostico)
prop.table(t)

#Estatísticas de um atributo numérico
summary(dados$km)

#Exibe a estrutura de um objeto: tipos das colunas, valores, número de registros
str(dados)

#Boxplot
boxplot(dados, main = "titulo", ylab = "label em y")

#Histograma
hist(dados, main = "titulo", xlab = "label no eixo x")

#Scatter plot (visualização da relação entre dois atributos)
plot( x = atributo1, y = atributo2, main = "titulo", xlab = "label no eixo x",
      ylab = "label no eixo y")

#Criando uma variável indicadora
usedcars$conservative <- usedcars$colors %in% c("Preta", "Branca", "Cinza",
"Prata")

Obs: c("Preta", "Branca", "Cinza", "Prata") cria um vetor coluna com os valores
passados entre parênteses .

#CrossTable
CrossTable é usada para verificar a relação entre duas variáveis nominais e
também para avaliar a performance de um modelo (exibindo falsos positivos,
falsos negativos, etc).

Ex: verificar a relação entre o modelo e as cores conservativas
CrossTable(usedcars$model, usedcars$conservative)

#lapply
lapply: aplica uma função a cada elemento de uma lista.
lapply é usada para aplicar uma função a cada atributo (coluna) de um dataframe.

Ex: exemplo de normalização
w_n <- as.data.frame(lapply(w[2:31],normalize))

#Criação das bases de treinamento e teste:
train <- base[inicio:fim, ]
test <- base[fim + 1: fim2, ]

Sintaxe: [row, column]

#A variável alvo de todo algoritmo de classificação deve ser codificada como um
fator (factor):

w$diagnostico <- factor(w$diagnostico, levels = c("B", "M"), labels =
c("Benigno", "Maligno"))

#Acessar um conjunto de colunas:
w[c("coluna1", "coluna2")]

#Função subset para selecionar um subconjunto de dados de um dataframe
Criando um dataframe somente para clientes_ativos:
clientes_ativos <- subset(clientes_ativos, STATUS == "ATIVO")

newdata <- subset(mydata, sex=="m" & age > 25, select=weight:income)

```

```
#Loads specified data sets, or list the available data sets.
#data(..., list = character(), package = NULL, lib.loc = NULL,
      verbose = getOption("verbose"), envir = .GlobalEnv,
      overwrite = TRUE)
```

with

Evaluate an R expression in an environment constructed from data, possibly modifying (a copy of) the original data.

Usage

```
with(data, expr, ...)
within(data, expr, ...)
# S3 method for list
within(data, expr, keepAttrs = TRUE, ...)
```

Arguments

data

data to use for constructing an environment. For the default with method this may be an environment, a list, a data frame, or an integer as in `sys.call`. For within, it can be a list or a data frame.

expr

expression to evaluate; particularly for `within()` often a “compound” expression, i.e., of the form

```
{
  a <- somefun()
  b <- otherfun()
  .....
  rm(unused1, temp)
}
```

keepAttrs

for the [list](#) method of `within()`, a [logical](#) specifying if the resulting list should keep the [attributes](#) from data and have its [names](#) in the same order. Often this is unneeded as the result is a *named* list anyway, and then `keepAttrs = FALSE` is more efficient.

...

arguments to be passed to (future) methods.