



# *Crash Course de R*

## Tabelas e Gráficos

Prof. Carlos Trucíos

 [ctruciosm.github.io](https://ctruciosm.github.io)

 [carlos.trucios@facc.ufrj.br](mailto:carlos.trucios@facc.ufrj.br)

Faculdade de Administração e Ciências Contábeis,  
Universidade Federal de Rio de Janeiro

[ctruciosm.github.io](https://ctruciosm.github.io) — Carlos Trucíos (FACC/UFRJ)

# Help

# Help

Quando não sabemos como utilizar um função, existe uma forma fácil de lidar com esse problema.

- `help(nome_da_função)`
- `?nome_da_função`

## Exemplos

- Como funciona a função `quantile()`?
- Como funciona a função `table()`?

Diferentes pacotes podem ter funções com o mesmo nome e isso dificulta saber qual `help` acessar. Para evitar esse problema podemos utilizar `?nome_do_pacote::nome_da_função`.

```
?base::table
```

[ctruciosm.github.io](https://ctruciosm.github.io) — Carlos Trucíos (FACC/UFRJ)

# Importando datos

# Importando dados

Nossa materia prima são os dados, e existem diversas formas de importa-los, aqui veremos as principais funções:

- `read.csv()`
- `read.table()`
- `read.csv2()`

Com essas funções podemos acessar a dados no nosso computador ou mesmo disponíveis na internet!

```
# Importando dataset Advertising  
uri <- "https://raw.githubusercontent.com/ctruciosm/ISLR/master/dataset/Advertising.csv"  
Advertising <- read.csv(uri)  
head(Advertising) # Mostra apenas os primeiros elementos
```

```
##      X      TV Radio Newspaper Sales  
## 1 1 230.1 37.8      69.2 22.1  
## 2 2  44.5 39.3      45.1 10.4  
## 3 3  17.2 45.9      69.3  9.3  
## 4 4 151.5 41.3      58.5 18.5  
## 5 5 180.8 10.8      58.4 12.9  
## 6 6   8.7 48.9      75.0  7.2
```

# Importando dados

## Hands-on:

O *dataset* `titanic` encontra-se no seguinte endereço

```
https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/titanic.csv
```

1. Importar os dados no R e salva-los com o nome `titanic`
2. Mostrar as primeiras observações utilizando a função `head()`

## Gabarito

```
uri <- "https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/titanic.csv"  
titanic <- read.csv(uri)  
head(titanic)
```

# Importando dados

## Importando um arquivo txt

```
uri <- "https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/knn_exemplo.txt"  
knn_dados <- read.table(uri)  
head(knn_dados)
```

```
##      V1  V2    V3  
## 1  V1  V2 Grupo  
## 2 0.7 0.7     A  
## 3 0.7 0.8     A  
## 4 0.6 0.6     A  
## 5 0.5 0.5     A  
## 6 0.5 0.6     A
```

## O que há de errado?

```
uri <- "https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/knn_exemplo.txt"  
knn_dados <- read.table(uri, header = TRUE)  
head(knn_dados)
```



# Importando dados

Às vezes o arquivo está separado por ; e não por ,.

## Método 1

```
uri <- "https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/propellant.csv"  
propellant <- read.csv(uri, sep = ";")  
head(propellant,4)
```

```
##      psi   age  
## 1 2158.70 15.50  
## 2 1678.15 23.75  
## 3 2316.00  8.00  
## 4 2061.30 17.00
```

## Método 2

```
uri <- "https://raw.githubusercontent.com/ctruciosm/ctruciosm.github.io/master/datasets/propellant.csv"  
propellant <- read.csv2(uri)  
head(propellant)
```



[ctruciosm.github.io](https://ctruciosm.github.io) — Carlos Trucíos (FACC/UFRJ)

# Tabelas de Frequência

# Tabelas de Frequência

**Dica:** antes de fazer qualquer outra coisa, dar uma rápida olhada nos dados com a função `head()`

```
head(titanic)
```

```
##      X PassengerId Survived Pclass                               Name
## 1 1         892         0     3rd                               Kelly, Mr. James
## 2 2         893         1     3rd      Wilkes, Mrs. James (Ellen Needs)
## 3 3         894         0     2nd                               Myles, Mr. Thomas Francis
## 4 4         895         0     3rd                               Wirz, Mr. Albert
## 5 5         896         1     3rd Hirvonen, Mrs. Alexander (Helga E Lindqvist)
## 6 6         897         0     3rd      Svensson, Mr. Johan Cervin
##      Sex  Age SibSp Parch Ticket   Fare Cabin Embarked
## 1  male 34.5   0     0 330911  7.8292      Q
## 2 female 47.0   1     0 363272  7.0000      S
## 3  male 62.0   0     0 240276  9.6875      Q
## 4  male 27.0   0     0 315154  8.6625      S
## 5 female 22.0   1     1 3101298 12.2875      S
## 6  male 14.0   0     0   7538  9.2250      S
```



# Tabelas de Frequência

Escreva os seguintes comandos no R.

```
table(titanic$Sex)           #tabela de freq. absolutas
freq_absolutas <- table(titanic$Sex) # Salvamos o objeto
prop.table(freq_absolutas)    #tabela de freq. relativas
prop.table(freq_absolutas)*100 #tabela de freq. percentual
round(prop.table(freq_absolutas)*100,2) # utilizamos round() para arredondar, neste caso
# para 2 casas decimais.
```

## Tabelas de frequência

Embora seja possível construir tabelas com as funções básicas do R, existem pacotes que fazem o trabalho ainda mais simples para nós.

```
install.packages("epiDisplay")      # instalando  
library(epiDisplay)                 # carregando
```

```
tab1(titanic$Sex, graph = FALSE)
```

```
## titanic$Sex :  
##           Frequency Percent Cum. percent  
## female           466     35.6           35.6  
## male             843     64.4           100.0  
## Total           1309    100.0           100.0
```

O que acontece se fizermos `graph = TRUE`?

[ctruciosm.github.io](https://ctruciosm.github.io) — Carlos Trucíos (FACC/UFRJ)

# Tabelas de dupla entrada

## Tabelas de dupla entrada

No R, quando queremos selecionar apenas uma variável do *dataset*, fazemos `dataset$nome_da_variavel`.

```
# Tabelas de contingencia  
table(titanic$Pclass,titanic$Sex)
```

```
##  
##      female male  
## 1st     144  179  
## 2nd     106  171  
## 3rd     216  493
```

Por facilidade, podemos querer fazer

```
classe <- titanic$Pclass  
sexo <- titanic$Sex  
table(classe,sexo)
```

```
##      sexo  
## classe female male  
## 1st     144  179  
## 2nd     106  171  
## 3rd     216  493
```

## Tabelas de dupla entrada

E se quisermos adicionar os totais por linha/coluna?

Utilizaremos `addmargins(objeto_tabela)`

```
addmargins(table(classe,sexo))
```

```
##           sexo
## classe female male  Sum
##   1st     144  179  323
##   2nd     106  171  277
##   3rd     216  493  709
##   Sum     466  843 1309
```

Outra opção é utilizar a função `tabpct()` do pacote `epiDisplay`

```
tabpct(classe,sexo, graph = FALSE)
```

[ctruciosm.github.io](https://ctruciosm.github.io) — Carlos Trucíos (FACC/UFRJ)

# Gráficos básicos





## Gráficos básicos

Existem várias formas de fazer gráficos no R, os gráficos mais elegantes são possíveis com o pacote `ggplot2` (que veremos no próximo laboratório), mas também é possível fazer gráficos simples.

- `hist(dataset$nome_da_variavel)`
- `pie(output_tabela_de_freq)`
- `boxplot(dataset$nome_da_variavel)`
- `barplot(output_tabela_de_freq)`
- `plot(datset$variavel1, dataset$variavel2)`

```
# Histograma  
hist(titanic$Fare)  
# Grafico de barras  
freq <- table(titanic$Pclass)  
barplot(freq)  
# Pie  
pie(freq)  
# Boxplot  
boxplot(titanic$Fare)  
# Gráfico de dispersão  
plot(Advertising$TV, Advertising$Sales)
```

# Gráficos básicos

Vários gráficos em um só figura

---

R Code    Plot

```
par(mfrow = c(2,2))  
plot(Advertising$TV, Advertising$Sales)  
plot(Advertising$Radio, Advertising$Sales)  
hist(Advertising$TV)  
boxplot(Advertising$Sales, horizontal = TRUE)
```