

# Programação em R

Profa. Luciane Alcoforado/AFA

2 de dezembro de 2021



Figure 1: Cartaz do Minicurso

## Tópicos a serem abordados

- 1- Introdução, o que é o R e o Rstudio, como obter e instalar (tempo: aprox. 15 min)

Objetivo: instalar os programas R e Rstudio, iniciar um novo arquivo no Rstudio

2- O que são os pacotes do R, como obter e instalar (tempo: aprox. 15 min)

Objetivo: aprender a instalar um pacote do R, conhecer a listagem de pacotes do R no CRAN

3- Operações matemáticas básicas, criação de objetos como vetores, matrizes e tabelas (tempo aprox. 1h)

Objetivo: criar objetos e realizar operações com os objetos

4- Operações Lógicas (tempo aprox. 30 min)

Objetivo: conhecer os operadores lógicos e realizar operações lógicas com os objetos

5- Operações com objetos (vetor, matriz e tabela) (tempo aprox. 30 min)

Objetivo: realizar operações com os objetos

6- Funções Estatísticas Básicas (tempo aprox. 30 min)

Objetivo: conhecer as funções estatísticas básicas e aplicá-las a vetores, matrizes ou tabela de dados (*dataframe*)

7- Criando um relatório simples com Rmarkdown (tempo aprox. 1h)

Objetivo: Conhecer o pacote Rmarkdown, construir uma estrutura de texto utilizando objetos e processamentos da linguagem R, compilar o texto em formato html

## Meus livros publicados



Figure 2: Livros da profa. Luciane F. Alcoforado

## Projetos voltados ao aprendizado da linguagem R:

<https://www.estatisticacomr.uff.br>

<https://www.ser.uff.br>



Figure 3: SER: Seminário Internacional de Estatística com R

# Introdução

## O que é o R?

É uma linguagem idealizada para realizar análise de dados através de um sistema para computação estatística e gráfica, permitindo explorar dados, produzir funções, computar linhas de comando ou utilizar pacotes disponíveis na rede CRAN (Comprehensive R Archive Network).

## O que é o R-Studio?

O *R-Studio* não é o R, e sim um ambiente de desenvolvimento integrado do R, portanto ele contém o R e o acesso a todos os pacotes disponíveis no CRAN. Atualmente é o melhor ambiente para desenvolvimento de pesquisas e relatórios com análise de dados em que se faça uso do R.

## Como obter o software R?

É muito simples e rápido obter o software R, basta acessar a página do projeto e realizar o *download* através do endereço <https://cran.r-project.org/>, selecionando o sistema operacional do seu equipamento, há disponível para Linux, para MAC e para Windows.

Instale primeiro o R e em seguida o *R-Studio*. Para instalar o *R-Studio* você deve fazer o *download* através do endereço <http://www.rstudio.com/products/rstudio/download/>, selecionando o sistema operacional do seu equipamento, há disponível para MAC, Windows, Ubuntu, Fedora.

O passo a passo de instalação pode ser visto no vídeo do canal do youtube do grupo **Estatística é com R** em <https://www.youtube.com/watch?v=8LnZNC4hxdQ>

# Pacotes do R

Os pacotes são um conjunto de funções programadas para realizar uma ou mais tarefas.

Para saber sobre os pacotes disponíveis no CRAN consulte <https://cran.r-project.org/web/packages/index.html>. É possível ver a lista por data de publicação ou por nome do pacote.

Alguns pacotes já estão disponíveis para uso tão logo se instale o R, já outros necessitam ser instalados.

A instalação é tarefa simples:

```
install.packages("nome do pacote")
```

Para este minicurso será necessário instalar o pacote *Rmarkdown*

```
install.packages("Rmarkdown")
```

# Operações Matemáticas

```
5+4      #adição
```

```
## [1] 9
```

```
6-2      #subtração
```

```
## [1] 4
```

```

7*3      #multiplicação
## [1] 21

45/9     #divisão
## [1] 5

2^2      #potência
## [1] 4

sqrt(121) #raiz
## [1] 11

exp(0)   #exponencial
## [1] 1

log(1)   #log na base e
## [1] 0

log10(1) #log na base 10
## [1] 0

log2(4)  #log na base 2
## [1] 2

log(9,3) #log na base 3 ou qualquer outra
## [1] 2

```

## Faça

- 1- O orçamento de um projeto possui custo estimado de R\$980.000,00. Converta este valor para dólar, considerando a taxa de câmbio de 1 dólar = R\$5.50
- 2- No início de 2019 o saco de cimento custava em média R\$20,00 e atualmente seu valor gira em torno de R\$35,00. Qual o aumento percentual observado nesse período de tempo no valor do saco de cimento?
- 3- Para fazer um muro de 2.20m por 35m serão utilizados blocos de concreto com rendimento de 12.5 blocos por m<sup>2</sup>. Quantos blocos serão necessários adquirir para fazer o muro?

## Operações Lógicas

Comandos lógicos

- Igualdade: ==
- Diferença: !=
- Desigualdades: >; <; <=; >=
- Lógicas: e - &; ou - |; negação !

```
#Testando igualdade
```

```
log(10)==3
```

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

```
#Testando diferença
```

```
sqrt(9) != 3
```

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

```
#Testando desigualdades
```

```
3^10 > 1000
```

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

```
#Testando a condição e
```

```
4 >= 8 & 2 < 3
```

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

```
#Testando a condição ou
```

```
4 >= 8 | 2 < 3
```

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

```
#Testando a negação
```

```
!5 >= 3
```

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

## Faça

Um pacote contém 2kg de rejunte custando R\$11,00 enquanto que outro pacote contém 5kg de rejunte custando R\$30,00. Verifique através de um teste lógico se os preços por kg destas duas modalidades de pacotes são equivalentes?

## Operações com objetos

### Criando vetores

```
#cria um vetor com 3 números  
c(7, 4, 1)
```

```
## [1] 7 4 1
```

```
#cria um vetor com 3 nomes (vetor de caracter)  
c("sete", "quatro", "um")
```

```
## [1] "sete"   "quatro" "um"
```

```
#cria uma sequencia iniciando em 5 e terminando em 15  
5:15
```

```
## [1] 5 6 7 8 9 10 11 12 13 14 15
```

### Operando vetores

```
#Criando dois vetores x e y  
x = 5:10
```

```

y = 15:20

#visualizando os dados do vetor x e y
x

## [1] 5 6 7 8 9 10
y

## [1] 15 16 17 18 19 20
#Somando dois vetores x e y
x+y

## [1] 20 22 24 26 28 30
#Multiplicando dois vetores x e y
x*y

## [1] 75 96 119 144 171 200
#Dividindo o vetor y pelo vetor x
y/x

## [1] 3.000000 2.666667 2.428571 2.250000 2.111111 2.000000
#Elevando o vetor x a potência 2
x^2

## [1] 25 36 49 64 81 100
#Obtendo o logaritmo do vetor x
log(x)

## [1] 1.609438 1.791759 1.945910 2.079442 2.197225 2.302585
#Obtendo o valor da segunda posição do vetor x
x[2]

## [1] 6
#Obtendo o menor valor do vetor y
min(y)

## [1] 15
#Obtendo a posição do menor valor do vetor y
which.min(y)

## [1] 1
#Obtendo a posição do vetor y que corresponde ao valor 19
which(y==19)

## [1] 5
#Obtendo o valor da quinta posição do vetor y
y[5]

## [1] 19

```

## Faça

Para realizar o orçamento de construção de muros de quatro obras o engenheiro organizou os dados em vetores. O vetor a contém a metragem quadrada dos muros; o vetor b contém o rendimento em unidade de bloco que será utilizado por  $m^2$  e o vetor c contém o custo do bloco por unidade.

```
a = c(50, 98, 38, 110)
b = c(12.5, 14, 9, 20)
c = c(2, 1.5, 1.2, 3.5)
a

## [1] 50 98 38 110

b

## [1] 12.5 14.0 9.0 20.0

c

## [1] 2.0 1.5 1.2 3.5
```

Realize operações com estes vetores para fornecer o custo do muro de cada obra.

## Criando matrizes

```
#criando uma matriz organizando vetores por linha
M1 = rbind(x,y)
M1

## [,1] [,2] [,3] [,4] [,5] [,6]
## x     5     6     7     8     9    10
## y    15    16    17    18    19    20
#criando uma matriz organizando vetores por coluna
M2 = cbind(x,y)
M2

##      x   y
## [1,] 5 15
## [2,] 6 16
## [3,] 7 17
## [4,] 8 18
## [5,] 9 19
## [6,] 10 20
```

## Operando matrizes

```
#Obtendo as dimensões da matriz
dim(M1)

## [1] 2 6

#obtendo a estrutura da matriz
str(M2)

##  int [1:6, 1:2] 5 6 7 8 9 10 15 16 17 18 ...
##  - attr(*, "dimnames")=List of 2
##    ..$ : NULL
```

```

## ..$ : chr [1:2] "x" "y"
#Obtendo o valor da linha 2 coluna 4 da matriz M1
M1[2,4]

## y
## 18
#Obtendo o valor da linha 4 coluna 1 da matriz M2
M2[4,1]

## x
## 8
#Acrescentando uma linha na matriz M1 ao final
M1 = rbind(M1, z=1:6)
M1

## [,1] [,2] [,3] [,4] [,5] [,6]
## x     5     6     7     8     9    10
## y    15    16    17    18    19    20
## z     1     2     3     4     5     6

#Acrescentando uma linha na matriz M1 no inicio
M1 = rbind(z=1:6,M1)
M1

## [,1] [,2] [,3] [,4] [,5] [,6]
## z     1     2     3     4     5     6
## x     5     6     7     8     9    10
## y    15    16    17    18    19    20
## z     1     2     3     4     5     6

#Acessando os valores da linha 2 da matriz M1
M1[2, ]

## [1] 5 6 7 8 9 10
#Acessando os valores da coluna 2 da matriz M1
M1[,2]

## z x y z
## 2 6 16 2
#Somando duas matrizes

M1 + M1

## [,1] [,2] [,3] [,4] [,5] [,6]
## z     2     4     6     8    10    12
## x    10    12    14    16    18    20
## y    30    32    34    36    38    40
## z     2     4     6     8    10    12

#Multiplicando duas matrizes
M1*M1

## [,1] [,2] [,3] [,4] [,5] [,6]
## z     1     4     9    16    25    36
## x    25    36    49    64    81   100
## y   225   256   289   324   361   400

```

```

## z      1     4     9    16    25    36
#Obtendo o valor máximo de uma matriz
max(M1)

## [1] 20
#Obtendo a posição da matriz que corresponde ao valor 20
which(M1==20, arr.ind = TRUE)

##   row col
## y   3   6
#Obtendo a transposta da matriz M1
t(M1)

##      z  x  y  z
## [1,] 1  5 15  1
## [2,] 2  6 16  2
## [3,] 3  7 17  3
## [4,] 4  8 18  4
## [5,] 5  9 19  5
## [6,] 6 10 20  6

```

## Criando dataframe

```

#criando um dataframe a partir dos vetores x e y
DT1 = data.frame(x,y)
DT1

##      x  y
## 1  5 15
## 2  6 16
## 3  7 17
## 4  8 18
## 5  9 19
## 6 10 20

```

## Operando dataframe

```

#Obtendo as dimensões do dataframe
dim(DT1)

## [1] 6 2
#obtendo a estrutura da matriz
str(DT1)

## 'data.frame':   6 obs. of  2 variables:
## $ x: int  5 6 7 8 9 10
## $ y: int  15 16 17 18 19 20
#Obtendo o valor da linha 2 coluna 4 do data frame
DT1[2,4]

## NULL

```

```

#Aumentando uma coluna no dataframe
DT1 = data.frame(DT1, z=1:6)
DT1

##   x  y  z
## 1 5 15 1
## 2 6 16 2
## 3 7 17 3
## 4 8 18 4
## 5 9 19 5
## 6 10 20 6

#Acessando os valores da linha 2 do dataframe DT1
DT1[2, ]

##   x  y  z
## 2 6 16 2

#Acessando os valores da coluna 2 do dataframe DT1
DT1[, 2]

## [1] 15 16 17 18 19 20

```

## Operações Lógicas em objetos

```

#Testando se um elemento do dataframe é igual a um certo valor
DT1[1,2]==3

## [1] FALSE

#Testando se um elemento do dataframe é diferente de um certo valor
DT1[1,2]!=3

## [1] TRUE

#Testando se um elemento do dataframe é maior ou igual a um certo valor
DT1[1,2]>=3

## [1] TRUE

#Testando a condição e
DT1[1,2]>=3 & DT1[1,3]<3

## [1] TRUE

#Testando a condição ou
DT1[1,2]>=3 | DT1[1,3]<3

## [1] TRUE

#Testando a negação
!DT1[1,2]>=3

## [1] FALSE

```

## Comando de repetição For

O **for** permite controlar o número de vezes que um ciclo é executado através de uma variável de controle que vai seguir uma sequencia de valores pré definidos em cada iteração do ciclo.

Sintaxe genérica:

```
for(var in conjunto){ bloco de instruções}
```

Criar um dataframe construindo cada coluna de forma iterativa

```
#Criar um data frame inicial com uma coluna
DT2 = data.frame(x=(1:5))
#Aumentar novas colunas de forma repetitiva - comando for
for (i in 1:4){
  DT2 = data.frame(DT2,x=(1:5)+i)
}
#Ver o resultado
DT2
```

```
##   x x.1 x.2 x.3 x.4
## 1 1   2   3   4   5
## 2 2   3   4   5   6
## 3 3   4   5   6   7
## 4 4   5   6   7   8
## 5 5   6   7   8   9
```

Calculando a soma de cada coluna do objeto DT2

```
soma = NULL

for (i in 1:ncol(DT2)){
  s=sum(DT2[,i])
  soma = c(soma,s)
}
#Ver o resultado
soma
```

```
## [1] 15 20 25 30 35
```

Outra maneira de realizar o mesmo procedimento é utilizando a função **apply** que permite-nos aplicar uma função qualquer a uma das dimensões (1- linha; 2-coluna) de uma matriz ou tabela.

Sintaxe genérica:

```
apply(matriz/tabela, dimensão, função)
```

```
apply(DT2,2,sum)
```

```
##   x x.1 x.2 x.3 x.4
## 1 15 20 25 30 35
```

Em (Alcoforado, 2021) há uma lista de funções básica do R no capítulo *Aprenda o essencial do pacote base* - pag 33-36.

## Faça

Suponha que se deseja realizar o download de diversos arquivos pdf. Para tanto, foi organizado um vetor com os endereços e um looping sobre o vetor para baixar os arquivos, conforme script:

```

url_pdfs <- c("https://periodicos.uff.br/anaisdoser/article/download/29256/16968",
  "https://github.com/Lucianeal/visualizacao-de-dados-com-o-software-R/raw/master/Visualizando_Dados_Luc",
  "https://periodicos.uff.br/anaisdoser/article/download/29336/17062")

#"https://periodicos.uff.br/anaisdoser/article/download/29325/17051/100880"

salve_aqui <- paste0("document_", 1:3, ".pdf")

for(i in seq_along(url_pdfs)){
  download.file(url_pdfs[i], save_aqui[i], mode = "wb")
}

```

Sua tarefa agora é rodar o script e verificar sua pasta de trabalho, conferindo que o procedimento foi realizado.

## Funções Estatísticas Básicas

### Média

A média é a soma de todos os valores de um conjunto de dados (numérico) dividido pelo tamanho do conjunto de dados.

Função no R: *mean()*

```

#Obtendo a média do vetor x
x = 1:100

mean(x)

## [1] 50.5

#Obtendo a média de todas as colunas de uma tabela
DT2

##   x x.1 x.2 x.3 x.4
## 1 1   2   3   4   5
## 2 2   3   4   5   6
## 3 3   4   5   6   7
## 4 4   5   6   7   8
## 5 5   6   7   8   9

apply(DT2,2,mean)

##   x x.1 x.2 x.3 x.4
## 3 4   5   6   7
#Obtendo a média de todas as linhas de uma tabela
apply(DT2,1,mean)

## [1] 3 4 5 6 7

```

### Mediana

É o valor central de um conjunto de dados.

Função no R: *median()*

```

#Obtendo a mediana do vetor x
x = 1:100

median(x)

## [1] 50.5

#Obtendo a mediana de todas as colunas de uma tabela
DT2

##   x x.1 x.2 x.3 x.4
## 1 1   2   3   4   5
## 2 2   3   4   5   6
## 3 3   4   5   6   7
## 4 4   5   6   7   8
## 5 5   6   7   8   9

apply(DT2,2,median)

##   x x.1 x.2 x.3 x.4
## 3 4   5   6   7

#Obtendo a mediana de todas as linhas de uma tabela
apply(DT2,1,median)

## [1] 3 4 5 6 7

```

## Desvio-padrão

É uma medida de variabilidade do conjunto de dados.

Função no R: `sd()`

```

#Obtendo o desvio-padrão do vetor x
x = 1:100

sd(x)

## [1] 29.01149

#Obtendo o desvio-padrão de todas as colunas de uma tabela
DT2

##   x x.1 x.2 x.3 x.4
## 1 1   2   3   4   5
## 2 2   3   4   5   6
## 3 3   4   5   6   7
## 4 4   5   6   7   8
## 5 5   6   7   8   9

apply(DT2,2,sd)

##          x      x.1      x.2      x.3      x.4
## 1.581139 1.581139 1.581139 1.581139 1.581139

#Obtendo o desvio-padrão de todas as linhas de uma tabela
apply(DT2,1,sd)

## [1] 1.581139 1.581139 1.581139 1.581139 1.581139

```

## Coeficiente de variação - CV

Assim como o desvio-padrão é também uma medida de variabilidade, só que adimensional, ou seja, mede a variabilidade relativa dos dados em relação à media dos mesmos, seu valor pode ser dado em porcentagem.

$$CV = \frac{\text{desvio-padrão}}{\text{média}} \cdot 100\%$$

#Obtendo o cv do vetor x

x = 1:100

100\*sd(x)/mean(x)

## [1] 57.4485

#Obtendo o cv de todas as colunas de uma tabela

100\*apply(DT2,2,sd)/apply(DT2,2,mean)

## x x.1 x.2 x.3 x.4

## 52.70463 39.52847 31.62278 26.35231 22.58770

#Obtendo o cv de todas as linhas de uma tabela

100\*apply(DT2,1,sd)/apply(DT2,1,mean)

## [1] 52.70463 39.52847 31.62278 26.35231 22.58770

A interpretação das faixas de valores do coeficiente de variação podem ser obtidas em (Alcoforado, 2021) no capítulo \*Análise descritiva dos dados“, página 251.

## Lendo uma base de dados

Considere uma base de dados armazenada em arquivo csv.

No pacote básico do R há dois comandos para realizar a leitura: `read.csv()` e `read.csv2()`. A diferença é referente ao separador de campo do arquivo que pode ser “,” (<https://raw.githubusercontent.com/Lucianeal/Alta/master/vendas1.csv>) ou “;” (<https://raw.githubusercontent.com/Lucianeal/Alta/master/vendas.csv>).

```
dados_com_virgula = read.csv("https://raw.githubusercontent.com/Lucianeal/Alta/master/vendas1.csv")  
  
dados_com_virgula  
  
dados_com_pontovirgula = read.csv("https://raw.githubusercontent.com/Lucianeal/Alta/master/vendas.csv")  
  
dados_com_pontovirgula
```

A leitura da base de dados gera um objeto do tipo dataframe:

Um resumo rápido da base de dados é obtida através da função `summary()`

Considerando que os dados da base de dados refere-se a registros de vendas de três filiais de uma rede num certo mês, com informações sobre o número do cupom fiscal, a filial, o valor da compra, o número de itens comprado, o desconto em percentual, a quinzena da venda.

Com base nas funções vistas até o momento vamos responder algumas perguntas:

1- Qual o valor total vendido pela filial A?

#Obtendo a base somente da filial A:

```
dadosA = dados_com_virgula[dados_com_virgula[,2]=="A",
```

```

]

#visualizando dadosA
dadosA

##   cupom filial valor_compra n_itens desconto_perc quinzena
## 1    101     A      100.22      5            2        1
## 2    102     A      80.89       20           0        1
## 3    103     A      75.44       7            0        1
## 4    104     A     305.33       3           10        2
## 5    105     A     120.99       1            2        2
## 6    106     A      27.89       1            0        2

#Calculando a soma da coluna 3, ou seja, o total das vendas

sum(dadosA[,3])

```

## [1] 710.76

2- Qual o valor médio do desconto da filial B?

#Obtendo a base somente da filial B:

```

dadosB = dados_com_virgula[dados_com_virgula[,2]== "B",
  ]

```

#visualizando dadosB

dadosB

```

##   cupom filial valor_compra n_itens desconto_perc quinzena
## 7    201     B      30.50      20            0        2
## 8    202     B      500.80      30           12        2
## 9    203     B     247.67      17            10        2
## 10   204     B      70.00      14            0        1
## 11   205     B      97.50      13            0        1
## 12   206     B     856.00      20           15        2
## 13   207     B      93.20      40            0        1
## 14   208     B     271.26      22            10        2
## 15   209     B     500.00       2           12        1
## 16   210     B      61.69      31            0        1
## 17   211     B      99.00       1            0        1
## 18   212     B     220.00     100            2        2

```

#Calculando a média da coluna 5, ou seja, a média dos descontos

```
mean(dadosB[,5])
```

## [1] 5.083333

3- A quinzena 1 vendeu mais que a quinzena 2 na rede (considerando todas as filiais)

#Obtendo a base somente da quinzena 1:

```

dadosq1 = dados_com_virgula[dados_com_virgula[,6]== 1,
  ]

```

#visualizando dadosq1

dadosq1

```

##      cupom filial valor_compra n_itens desconto_perc quinzena
## 1     101     A       100.22      5            2           1
## 2     102     A       80.89      20           0           1
## 3     103     A       75.44      7            0           1
## 10    204     B       70.00      14           0           1
## 11    205     B       97.50      13           0           1
## 13    207     B       93.20      40           0           1
## 15    209     B      500.00      2           12          1
## 16    210     B       61.69      31           0           1
## 17    211     B       99.00      1            0           1
## 19    301     C       12.25      1            0           1
## 23    305     C      732.00      60          12          1

```

*#Calculando a soma da coluna 3, ou seja o total de vendas*

```
sum(dadosq1[,3])
```

```
## [1] 1922.19
```

*#Obtendo a base somente da quinzena 2:*

```

dadosq2 = dados_com_virgula[dados_com_virgula[,6]==2,
  ]

```

*#visualizando dadosq2*

```
dadosq2
```

```

##      cupom filial valor_compra n_itens desconto_perc quinzena
## 4     104     A       305.33      3            10          2
## 5     105     A       120.99      1            2           2
## 6     106     A       27.89      1            0           2
## 7     201     B       30.50      20           0           2
## 8     202     B      500.80      30           12          2
## 9     203     B       247.67      17           10          2
## 12    206     B      856.00      20           15          2
## 14    208     B       271.26      22           10          2
## 18    212     B      220.00     100           2           2
## 20    302     C      188.00      45           2           2
## 21    303     C       117.60      8            2           2
## 22    304     C      354.00     100          10          2

```

*#Calculando a soma da coluna 3, ou seja o total de vendas*

```
sum(dadosq2[,3])
```

```
## [1] 3240.04
```

*#respondendo a pergunta com teste lógico*

```
sum(dadosq1[,3]) > sum(dadosq2[,3])
```

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

4- Qual o número do cupom que vendeu mais itens? E quantas unidades foram vendidas neste cupom?

*#Obtendo a linha correspondente ao maior valor da coluna 4*

```

linha = dados_com_virgula[dados_com_virgula[,4]==max(dados_com_virgula[,4]),
  ]

```

```
#Obtendo o número do cupom e a quantidade vendida, ou seja, coluna 1 e 4
linha[,c(1,4)]
```

```
##      cupom n_itens
## 18     212      100
## 22     304      100
```

5- Acrescente na resposta anterior em que quinzena foi vendido o número máximo de itens

```
linha[,c(1,4,6)]
```

```
##      cupom n_itens quinzena
## 18     212      100          2
## 22     304      100          2
```

## Criando um relatório

Para criar um relatório utilizamos o pacote **Rmarkdown**. É necessário instalar este pacote e após isso abrir através do Rstudio um arquivo do tipo Rmd.

Ao abrir um novo arquivo Rmd, um modelo explicativo é aberto para dar início ao processo de aprendizagem. Este minicurso foi criado no arquivo Rmd, para ver seu código basta acessar o arquivo que ficará disponibilizado no github.

As saídas após compilação podem ser html, doc ou pdf. Inicie pela saída html que é a mais amigável.

Para aprofundar mais na programação com R consulte a bibliografia deste minicurso.

### Referência Bibliográfica

ALCOFORADO, L.F. Utilizando a linguagem R: conceitos, manipulação, visualização, modelagem e elaboração de relatório, Rio de Janeiro, Alta Books, 2021.

R Core Team, R: A language and environment for statistical computing. **R Foundation for Statistical Computing**, Vienna, Austria, 2021. URL (<https://www.R-project.org/>).