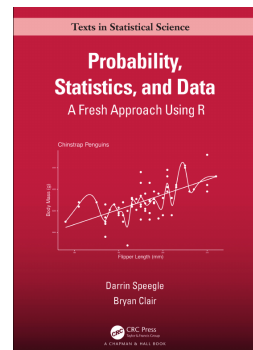

Resolução de Problemas do Livro

Probability, Statistics, and Data: A Fresh Approach Using R (Speegle, D.; Clair, B.)

por
Igo da Costa Andrade

Referência

SPEEGLE, D.; CLAIR, B.. **Probability, Statistics, and Data: A Fresh Approach Using R**. Local, CRC Press, 2022.



Capítulo 1: Dados em R¹

```
# Bibliotecas Necessárias
library(tidyverse)
library(knitr)
library(kableExtra)
library(latex2exp)
library(fosdata)
library(HistData)
library(eaf)
library(tikzDevice)
options(tikzMetricPackages = c("\\usepackage{amsmath}",
                                "\\usepackage{xcolor}",
                                "\\usepackage{tikz}",
                                "\\usetikzlibrary{calc}"))
```

Exercícios

1.4 Neste exercício, construa o gráfico da função $f(p) = p(1 - p)$ para $p \in [0, 1]$.

- Use `seq` para criar um vetor p de números de 0 a 1 espaçados por 0.2.
- Use a função `plot` para graficar p na coordenada x e $p(1 - p)$ na coordenada y . Leia a página de ajuda para `plot` e faça testes com o argumento `type` para encontrar uma boa escolha para este gráfico.
- Repita, mas criando um vetor p de números de 0 a 1 espaçados por 0.01.

Solução:

Vetor p com *step* igual a 0.1:

```
p1 <- seq(0, 1, by=0.2)
Fp1 <- p1 * (1-p1)
```

¹Título original: *Data in R*

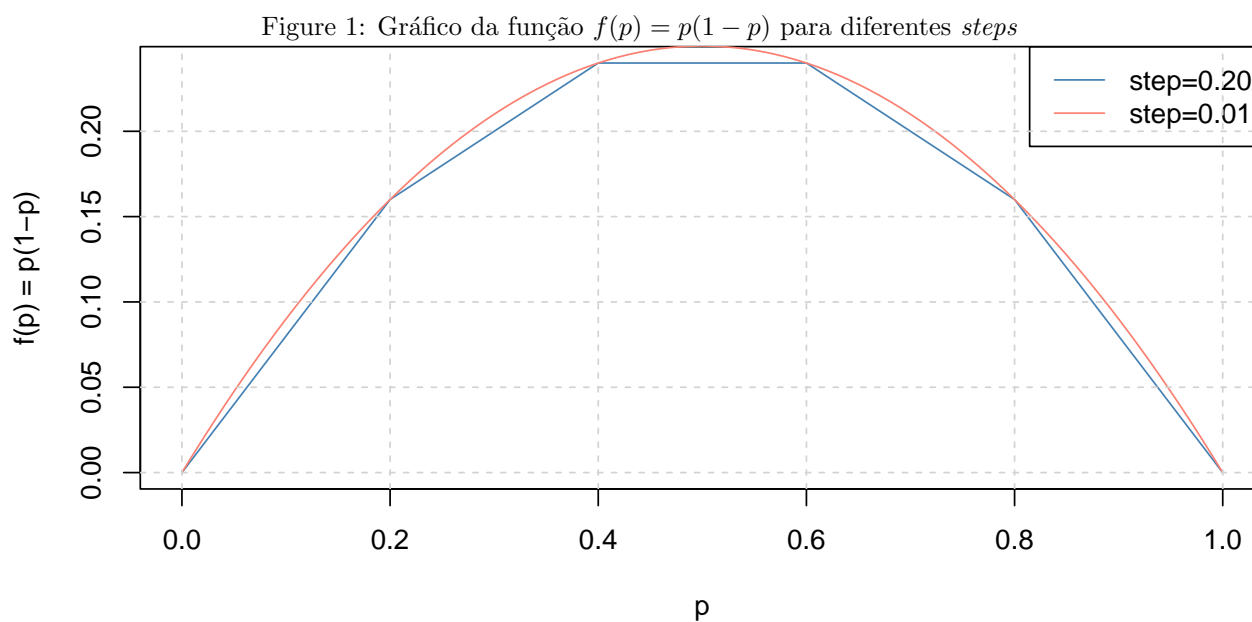
```

p2 <- seq(0, 1, by=0.01)
Fp2 <- p2 * (1-p2)

pdf(file = "figure/chap-01/problema-1.4.pdf",
    width = 8,
    height = 4.5)
plot(
  p1, Fp1, type="l", lty=1, col="steelblue",
  xlab="p", ylab="f(p) = p(1-p)",
)
lines(
  p2, Fp2, lty=1, col="salmon", xlab="", ylab=""
)
legend(x="topright", lty=c(1, 1),
      legend = c("step=0.20", "step=0.01"),
      col=c("steelblue", "salmon"),
)
grid(lty="dashed")

dev.off()
eaf::pdf_crop("figure/chap-01/problema-1.4.pdf")

```



■

1.5 Use R para calcular a soma dos quadrados de todos os números de 1 a 100: $1^2 + 2^2 + \dots + 99^2 + 100^2$.

Solução:

```
soma <- sum((1:100)^2)
print(soma)
```

```
## [1] 338350
```

$$\sum_{i=1}^{100} x_i = 1^2 + 2^2 + \cdots + 99^2 + 100^2 = 338.350$$



1.6 Seja x o vetor obtido da execução do comando R `x <- seq(from=10, to=30, by=2)`.

- Qual é o comprimento de x ?
- O que é $x[2]$?
- O que é $x[1:5]$?
- O que é $x[1:3*2]$?
- O que é $x[1:(3*2)]$?
- O que é $x > 25$?
- O que é $x[x > 25]$?
- O que é $x[-1]$?
- O que é $x[-1:-3]$?

Solução:

```
# Definição do vetor x
x <- seq(from = 10, to = 30, by = 2)
x
```

```
## [1] 10 12 14 16 18 20 22 24 26 28 30
```

```
# a. Qual é o comprimento de x?
length(x)
```

```
## [1] 11
```

O comprimento do vetor x , ou seja a quantidade de elementos desse vetor é igual a `length(x) = 11`.

```
# b. O que é x[2]?
x[2]
```

```
## [1] 12
```

$x[2]$ é o segundo elemento do vetor x , e seu valor é $x[2] = 12$.

```
# c. O que é x[1:5]?
x[1:5]
```

```
## [1] 10 12 14 16 18
```

$x[1:5]$ é um subconjunto do vetor x representado pelos elementos desde a primeira posição até a quinta posição.

```
# d. O que é x[1:3*2]?
x[1:3*2]
```

```
## [1] 12 16 20
```

$x[1:3*2]$ é um subconjunto do vetor x representado pelos elementos nas posições:

$$1:3*2 = c(1, 2, 3)*2 = c(2, 4, 6)$$

```
# e. O que é x[1:(3*2)]?
x[1:(3*2)]
```

```
## [1] 10 12 14 16 18 20
```

$x[1:(3*2)]$ é o subconjunto de x representado pelos elementos de 1 até 6, visto que:

$$1:(3*2) = 1:6 = c(1, 2, 3, 4, 5, 6)$$

```
# f. O que é x > 25?
x > 25
```

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

$x > 25$ é um vetor lógico (booleano), resultado da verificação para cada elemento de x se o referido elemento é maior que 25 (TRUE) ou não (FALSE).

```
# g. O que é x[x > 25]?
x[x > 25]
```

```
## [1] 26 28 30
```

$x[x > 25]$ é um subconjunto de x representado pelos elementos de x que são maiores que 25.

```
# h. O que é x[-1]?
x[-1]
```

```
## [1] 12 14 16 18 20 22 24 26 28 30
```

$x[-1]$ é o subconjunto de x após a exclusão do primeiro elemento.

```
# i. O que é x[-1:-3]?
x[-1:-3]
```

```
## [1] 16 18 20 22 24 26 28 30
```

$x[-1:-3]$ é o subconjunto de x após a exclusão dos elementos nas posições 1, 2, e 3. ■

1.7 R possui um vetor denominado `rivers` o qual contém os comprimentos dos maiores rios Norte Americanos.

- Use `?rivers` para apreender sobre o conjunto de dados.
- Encontre a média e o desvio padrão dos dados dos rios usando as funções de R base `mean` e `sd`.
- Crie um histogram (`hist`) dos dados de rios.
- Obtenha cinco números resumos (`summary`) dos dados dos rios.
- Encontre o maior e o menor comprimento de rios do conjunto.

f. Crie uma lista de todos os (comprimentos de) rios maiores que 1000 milhas.

Solução:

```
# b1. Média do comprimento dos rios
rivers_mean <- mean(rivers)
rivers_mean
```

```
## [1] 591.1844
```

$$\bar{x}_{\text{rivers}} = \sum_{i=1}^{n=141} \frac{x_i}{n} = \frac{735 + 320 + \cdots + 1.770}{141} = \frac{83.357}{141} = 591,18 \text{ milhas}$$

```
# b2. Desvio Padrão do comprimento dos rios
rivers_sd <- sd(rivers)
rivers_sd
```

```
## [1] 493.8708
```

$$\sigma_{\text{rivers}} = \sqrt{\sum_{i=1}^{n=141} \frac{(x_i - \bar{x}_{\text{rivers}})^2}{n}} = \sqrt{\frac{(735 - 591,18)^2 + (320 - 591,18)^2 + \cdots + (1.770 - 591,18)^2}{141}} = 493,87 \text{ milhas}$$

```
# c. Histograma da distribuição de comprimentos de rios
tikz("tex/chap-01/problema-1.7c.tex",standAlone = TRUE,
     packages=c("\\usepackage{amsmath}",
                "\\usepackage{tikz}",
                "\\usepackage{xcolor}",
                "\\usetikzlibrary{calc}",
                "\\usepackage[active,tightpage,psfixbb]{preview}",
                "\\PreviewEnvironment{pgfpicture}"))

hist(rivers,
     xlab="Comprimento (milhas)", ylab="Frequência", main="")
);

rect(
  xleft=rivers_mean-rivers_sd, xright=rivers_mean+rivers_sd,
  ybottom=0, ytop=100, col= rgb(0.2745,0.5098, 0.7059,alpha=0.3)
)

abline(v=rivers_mean, lty=1, lwd=3, col="steelblue")
legenda <- sprintf(
  "$\\overline{x}_{\\text{rivers}} \\pm \\sigma_{\\text{rivers}} = (%s \\pm %s)$ milhas",
  fmt(rivers_mean), fmt(rivers_sd))

legend(x="topright", lty=c(1), lwd=c(3),
       legend = c(legenda),
```

```

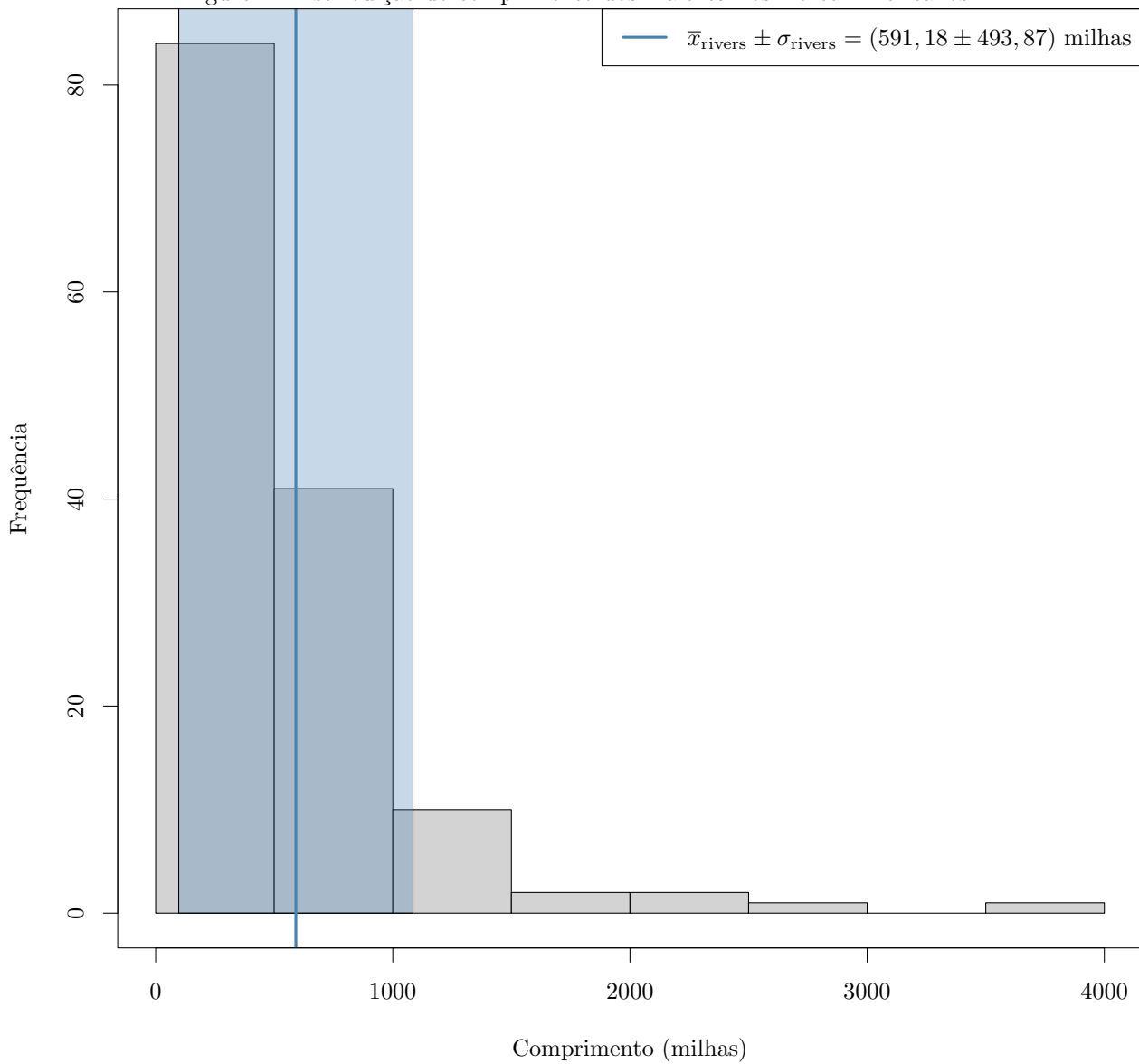
    col="steelblue",
)

box()

dev.off();
tools::texi2pdf("tex/chap-01/problema-1.7c.tex", clea=TRUE)
system(paste(getOption("pdfviewer"), "tex/chap-01/problema-1.7c.tex", sep=""));
file.rename("problema-1.7c.pdf", "figure/chap-01/problema-1.7c.pdf")
eaf::pdf_crop("figure/chap-01/problema-1.7c.pdf")

```

Figure 2: Distribuição do comprimento dos maiores rios Norte Americanos



```

# d. Medidas Resumo dos dados de rios
summary(rivers)

```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  135.0   310.0   425.0   591.2   680.0   3710.0
```

```
# e. Maior e Menor comprimentos de rios
```

```
rivers_max <- max(rivers)
```

```
rivers_min <- min(rivers)
```

O maior comprimento de rio do conjunto de dados foi de 3.710 milhas enquanto o menor comprimento foi de 135.

```
# f. Lista de comprimentos de rios maiores que 1000 milhas
```

```
rivers[rivers > 1000]
```

```
## [1] 1459 1450 1243 2348 1171 3710 2315 2533 1306 1054 1270 1885 1100 1205 1038
```

```
## [16] 1770
```



1.8 Considere o conjunto de dados `airquality`.

- Quantas observações de quantas variáveis existem?
- Quais os nomes das variáveis?
- Qual o tipo de dados de cada variável?
- Você concorda com o tipo de dados associado a cada variável? Existem escolhas melhores?

Solução:

```
# a.
```

```
n_row <- nrow(airquality)
```

```
n_col <- ncol(airquality)
```

O conjunto de dados `airquality` consiste em 153 observações de 6 variáveis.

```
# b.
```

```
col_names <- colnames(airquality)
```

```
col_names
```

```
## [1] "Ozone" "Solar.R" "Wind" "Temp" "Month" "Day"
```

Os nomes das variáveis são Ozone, Solar.R, Wind, Temp, Month, e Day.

```
# c.
```

```
tbl <- sprintf("%8s %8s", "Variável", "Tipo")
```

```
tbl <- sprintf("%s\n%s", tbl, "=====")
```

```
for (name in col_names) {
```

```
  row <- sprintf("%8s %8s", name, class(airquality[[name]]))
```

```
  tbl <- sprintf("%s\n%s", tbl, row)
```

```
}
```

```
cat(tbl)
```

```
## Variável      Tipo
## =====
##      Ozone   integer
##      Solar.R integer
##      Wind    numeric
##      Temp    integer
##      Month   integer
##      Day     integer
```

```
# d.
```

```
head(airquality)
```

```
##      Ozone Solar.R Wind Temp Month Day
## 1      41      190  7.4   67     5    1
## 2      36      118  8.0   72     5    2
## 3      12      149 12.6   74     5    3
## 4      18      313 11.5   62     5    4
## 5      NA       NA 14.3   56     5    5
## 6      28       NA 14.9   66     5    6
```

A variável `Month` poderia ser melhor descrita como do tipo `factor`. ■

1.9 Há um conjunto de dados integrado chamado `state`, que na verdade é composto por sete variáveis separadas com nomes como `state.name`, `state.region` e `state.area`.

- a. Quais são as possíveis regiões às quais um estado pode pertencer? Quantos estados existem em cada região?
- b. Quais estados têm áreas menor que 10.000 milhas quadradas?
- c. Qual estado tem o centro geográfico mais ao sul? (Dica: use `which.min`).

Solução:

```
# a.
# Tipo de dados da variável state.region
class(state.region)
```

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

```
# Regiões possíveis
levels(state.region)
```

```
## [1] "Northeast"      "South"           "North Central"  "West"
```

```
# Quantidade estados por região
table(state.region)
```

```
## state.region
##      Northeast      South North Central      West
##              9         16         12         13
```



```
# b. Estados com área menor que 10.000 milhas
state.name[state.area < 10000]
```

```
## [1] "Connecticut" "Delaware" "Hawaii" "Massachusetts"
## [5] "New Hampshire" "New Jersey" "Rhode Island" "Vermont"
```

```
# c.
state.name[which.min(state.center$y)]
```

```
## [1] "Florida"
```

No código acima, `state.center` é uma lista de componentes nomeados `x` e `y`, que representam, respectivamente a latitude e a longitude do centro geográfico de cada estado. Ao aplicar `which.min(state.center$y)`, obtemos o índice no vetor `state.center$y` de menor valor, ou seja, menor longitude, ou ainda do estado mais ao sul. Finalmente, ao aplicar `state.name[which.min(state.center$y)]`, encontramos no nome desse estado: Florida. ■

1.10 Considere o conjunto de dados `mtcars`.

- Quais carros têm 4 marchas?
- Que subconjunto de `mtcars` é descrito por `mtcars[mtcars$disp > 150 & mtcars$mpg > 20,]` ?
- Quais carros têm 4 marchas para frente e transmissão manual? (Nota: transmissão manual é 1 e automática é 0.)
- Quais carros têm 4 marchas para frente ou transmissão manual?
- Encontre a média de `mpg` dos carros com 2 carburadores.

Solução:

```
# Visualização do conjunto de dados
head(mtcars)
```

```
##           mpg  cyl  disp  hp drat   wt  qsec vs  am  gear  carb
## Mazda RX4    21.0   6  160  110 3.90 2.620 16.46 0   1    4    4
## Mazda RX4 Wag 21.0   6  160  110 3.90 2.875 17.02 0   1    4    4
## Datsun 710    22.8   4  108   93 3.85 2.320 18.61 1   1    4    1
## Hornet 4 Drive 21.4   6  258  110 3.08 3.215 19.44 1   0    3    1
## Hornet Sportabout 18.7   8  360  175 3.15 3.440 17.02 0   0    3    2
## Valiant      18.1   6  225  105 2.76 3.460 20.22 1   0    3    1
```

O conjunto de dados `mtcars` consiste em 32 observações de 11 variáveis. A título de exemplo, as seis primeiras observações são mostradas acima.

```
# a.
nrow(mtcars[mtcars$gear == 4, ])
```

```
## [1] 12
```

Existem 12 com 4 marchas.

```
# b.
mtcars[mtcars$disp > 150 & mtcars$mpg > 20,]
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46 0  1   4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02 0  1   4    4
## Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44 1  0   3    1
```

O subconjunto acima mostra os carros para os quais o deslocamento do motor (`disp`) é superior a 150 e a eficiência do combustível (`mpg`) é maior que 20.

```
# c.
mtcars[mtcars$gear == 4 & mtcars$am == 1, ]
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46 0  1   4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02 0  1   4    4
## Datsun 710      22.8   4 108.0  93 3.85 2.320 18.61 1  1   4    1
## Fiat 128        32.4   4  78.7  66 4.08 2.200 19.47 1  1   4    1
## Honda Civic     30.4   4  75.7  52 4.93 1.615 18.52 1  1   4    2
## Toyota Corolla  33.9   4  71.1  65 4.22 1.835 19.90 1  1   4    1
## Fiat X1-9       27.3   4  79.0  66 4.08 1.935 18.90 1  1   4    1
## Volvo 142E      21.4   4 121.0 109 4.11 2.780 18.60 1  1   4    2
```

São eles: Mazda RX4, Mazda RX4 Wag, Datsun 710, Fiat 128, Honda Civic, Toyota Corolla, Fiat X1-9, e Volvo 142E.

```
# d.
mtcars[mtcars$gear == 4 | mtcars$am == 1, ]
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46 0  1   4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02 0  1   4    4
## Datsun 710      22.8   4 108.0  93 3.85 2.320 18.61 1  1   4    1
## Merc 240D       24.4   4 146.7  62 3.69 3.190 20.00 1  0   4    2
## Merc 230        22.8   4 140.8  95 3.92 3.150 22.90 1  0   4    2
## Merc 280        19.2   6 167.6 123 3.92 3.440 18.30 1  0   4    4
## Merc 280C       17.8   6 167.6 123 3.92 3.440 18.90 1  0   4    4
## Fiat 128        32.4   4  78.7  66 4.08 2.200 19.47 1  1   4    1
## Honda Civic     30.4   4  75.7  52 4.93 1.615 18.52 1  1   4    2
## Toyota Corolla  33.9   4  71.1  65 4.22 1.835 19.90 1  1   4    1
## Fiat X1-9       27.3   4  79.0  66 4.08 1.935 18.90 1  1   4    1
## Porsche 914-2   26.0   4 120.3  91 4.43 2.140 16.70 0  1   5    2
## Lotus Europa    30.4   4  95.1 113 3.77 1.513 16.90 1  1   5    2
## Ford Pantera L  15.8   8 351.0 264 4.22 3.170 14.50 0  1   5    4
## Ferrari Dino    19.7   6 145.0 175 3.62 2.770 15.50 0  1   5    6
## Maserati Bora   15.0   8 301.0 335 3.54 3.570 14.60 0  1   5    8
## Volvo 142E      21.4   4 121.0 109 4.11 2.780 18.60 1  1   4    2
```

São eles: Mazda RX4, Mazda RX4 Wag, Datsun 710, Merc 240D, Merc 230, Merc 280, Merc 280C, Fiat 128, Honda Civic, Toyota Corolla, Fiat X1-9, Porsche 914-2, Lotus Europa, Ford Pantera L, Ferrari Dino, Maserati Bora, e Volvo 142E.

```
# e.  
mpg_bar <- mean(mtcars[mtcars$carb == 2, c("mpg")])  
mpg_bar
```

```
## [1] 22.4
```

A eficiência média do consumo de combustível para os carros com 2 carburadores foi de 22,4 milhas por galão.

