Resolução da Lista 2

Germano Andrade Brandão - 2017080008

07/04/2020

Nota inicial

Para a resolução dos exercícios, foram utilizados os pacotes "ggplot2" (para construção de gráficos) e "ggpubr" (para realocar os gráficos na última questão da lista).

Capítulo 2

Questão 3

3. Para o Conjunto de Dados 1 (CD-Brasil), construa a distribuição de freqüências para as variáveis população urbana e densidade populacional.

ºPopulação Urbana

```
dadosPop <- cd_brasil$pop_urbana
dadosPop <- dadosPop[!is.na(dadosPop)]

Frequencia_ni <- table(cut(dadosPop,
                           b=c(-Inf,700000, 2000000,4000000,6000000,8000000,Inf),
                           dig.lab = 8))
Frequencia_fi <- Frequencia_ni/sum(Frequencia_ni)

Porcentagem_100fi <- Frequencia_fi*100

Tabela_Pop <- matrix(c(Frequencia_ni,
                       Frequencia_fi,
                       Porcentagem_100fi),
                     nrow = 6,
                     ncol=3)

Tabela_Pop <- rbind(Tabela_Pop,
                    colSums(Tabela_Pop[,1:3]))

colnames(Tabela_Pop) <- c("Frequência ni",
                          "Frequência fi",
                          "Porcentagem 100fi")

row.names(Tabela_Pop) <- c("Menor que 700.000",
                           "Entre 700.000 e 2.000.000",
                           "Entre 2.000.000 e 4.000.000",
                           "Entre 2.000.000 e 4.000.000",
                           "Entre 6.000.000  e 8.000.000",
```

```
                          "Maior que 8.000.000", "Total")
Tabela_Pop
```

```
##                               Frequência ni Frequência fi Porcentagem 100fi
## Menor que 700.000                         3    0.11111111         11.111111
## Entre 700.000 e 2.000.000                10    0.37037037         37.037037
## Entre 2.000.000 e 4.000.000               6    0.22222222         22.222222
## Entre 2.000.000 e 4.000.000               2    0.07407407          7.407407
## Entre 6.000.000  e 8.000.000              3    0.11111111         11.111111
## Maior que 8.000.000                       3    0.11111111         11.111111
## Total                                    27    1.00000000        100.000000
```

ºDensidade Populacional

```
dadosDensi <- cd_brasil$densidade
dadosDensi <- as.numeric(sub(",", ".", dadosDensi))
#Filtrando para remover os NA
dadosDensi <- dadosDensi[!is.na(dadosDensi)]

Frequencia_ni <- (table(cut(dadosDensi,
                            b=c(-Inf, 8, 25, 50, 70, 90, Inf))))
Frequencia_fi <- Frequencia_ni/sum(Frequencia_ni)

Porcentagem_100fi <- Frequencia_fi*100
Tabela_Densi <- matrix(c(Frequencia_ni,
                         Frequencia_fi,
                         Porcentagem_100fi),
                       nrow = 6,
                       ncol = 3)

Tabela_Densi <- rbind(Tabela_Densi,
                      colSums(Tabela_Densi[,1:3]))

colnames(Tabela_Densi) <- c("Frequência ni",
                            "Frequência fi",
                            "Porcentagem 100fi")

row.names(Tabela_Densi) <- c("Menor que 8",
                             "Entre 8 e 25",
                             "Entre 25 e 50",
                             "Entre 50 e 70",
                             "Entre 70 e 90",
                             "Maior que 90",
                             "Total")
Tabela_Densi
```

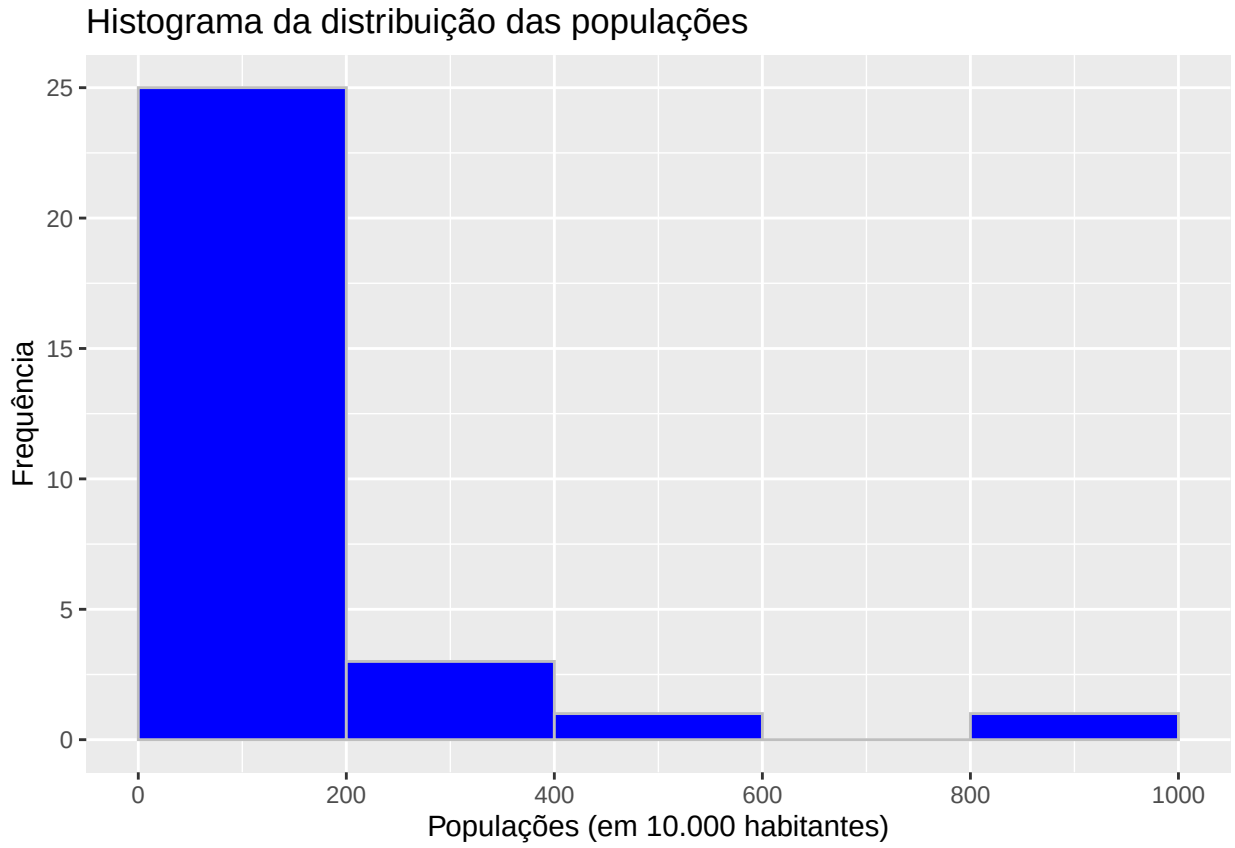
```
##               Frequência ni Frequência fi Porcentagem 100fi
## Menor que 8               9    0.33333333         33.333333
## Entre 8 e 25              3    0.11111111         11.111111
## Entre 25 e 50             5    0.18518519         18.518519
## Entre 50 e 70             3    0.11111111         11.111111
## Entre 70 e 90             2    0.07407407          7.407407
## Maior que 90              5    0.18518519         18.518519
## Total                    27    1.00000000        100.000000
```

Questão 8

8. Construa um histograma, um ramo-e-folhas e um gráfico de dispersão unidimensional para o conjunto de dados 2 (CD-Municípios).

ºHistograma

```
ggplot(data = cd_municipios,
       aes(x = populacao)) +
  geom_histogram(breaks=seq(0,1000, by=200),
                 col="gray",
                 fill="blue") +
  labs(title="Histograma da distribuição das populações",
       x="Populações (em 10.000 habitantes)",
       y="Frequência") +
  scale_x_continuous(breaks=seq(0,1000,200))
```



ºRamo-e-folhas

```
stem(cd_municipios$populacao, scale =16)
```

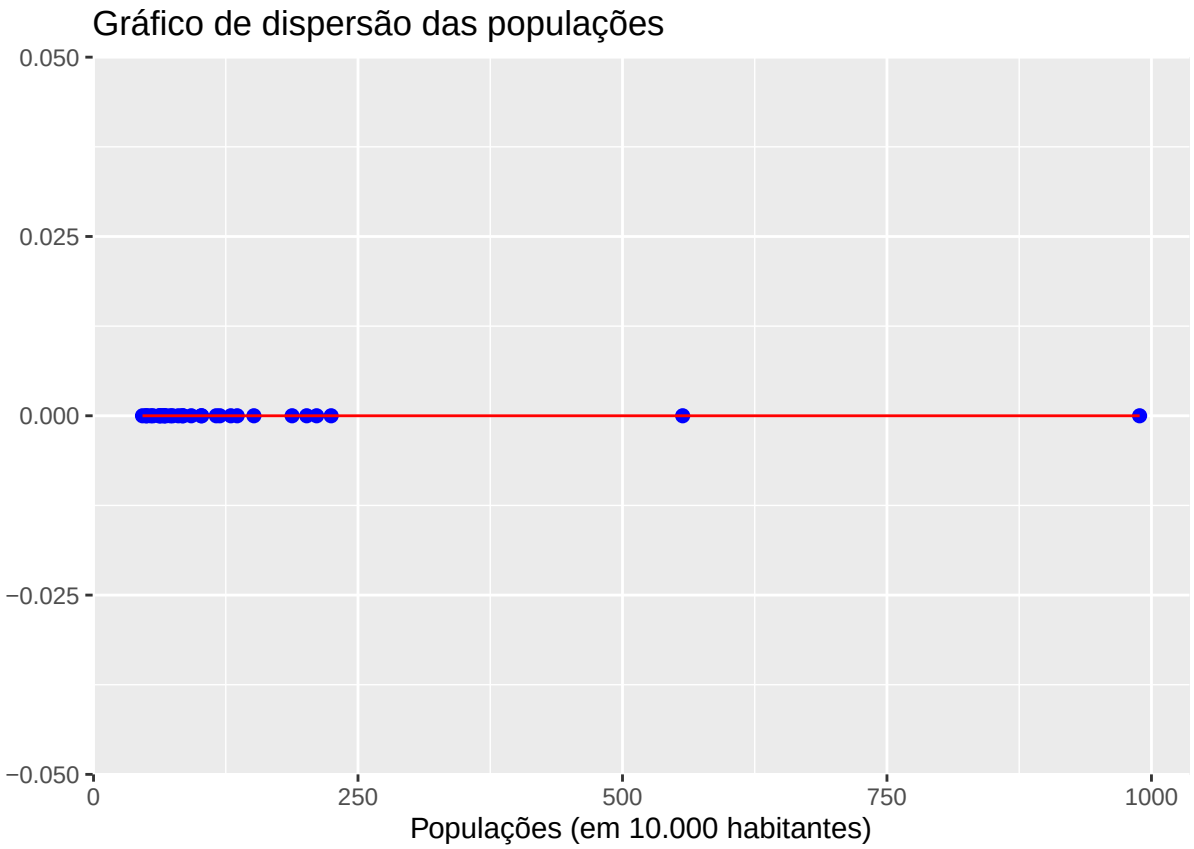
```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    4 | 6
##    5 | 0046
##    6 | 234778
##    7 | 35
```

```
##    8 | 045
##    9 | 2
##   10 | 22
##   11 | 69
##   12 |
##   13 | 06
##   14 |
##   15 | 2
##   16 |
##   17 |
##   18 | 8
##   19 |
##   20 | 2
##   21 | 1
##   22 | 5
##   23 |
##   24 |
##   25 |
##   26 |
##   27 |
##   28 |
##   29 |
##   30 |
##   31 |
##   32 |
##   33 |
##   34 |
##   35 |
##   36 |
##   37 |
##   38 |
##   39 |
##   40 |
##   41 |
##   42 |
##   43 |
##   44 |
##   45 |
##   46 |
##   47 |
##   48 |
##   49 |
##   50 |
##   51 |
##   52 |
##   53 |
##   54 |
##   55 | 7
##   56 |
##   57 |
##   58 |
##   59 |
##   60 |
##   61 |
```

```
##   62 |
##   63 |
##   64 |
##   65 |
##   66 |
##   67 |
##   68 |
##   69 |
##   70 |
##   71 |
##   72 |
##   73 |
##   74 |
##   75 |
##   76 |
##   77 |
##   78 |
##   79 |
##   80 |
##   81 |
##   82 |
##   83 |
##   84 |
##   85 |
##   86 |
##   87 |
##   88 |
##   89 |
##   90 |
##   91 |
##   92 |
##   93 |
##   94 |
##   95 |
##   96 |
##   97 |
##   98 | 9
```

ºGráfico de dispersão unidimensional

```
ggplot(data = cd_municipios, aes(populacao, 0)) +
  geom_point(col='blue', size =2) +
  labs(title="Gráfico de dispersão das populações",
       x="Populações (em 10.000 habitantes)",
       y=" ") +
  geom_line(col = "red")
```



Questão 20

20. Construa um ramo-e-folhas para a variável CO (monóxido de carbono) do conjunto de dados 4 (CD-Poluição).

ºRamo-e-folhas

```
stem(cd_poluicao$co)
```

```
##
##   The decimal point is at the |
##
##    4 | 77
##    5 | 12
##    5 | 55677789
##    6 | 11111222222222333333444444
##    6 | 5666677777899999999
##    7 | 00122233444
##    7 | 5566777778888899999999
##    8 | 012334
##    8 | 55678999
##    9 | 0114
##    9 | 557
##   10 | 1333
##   10 | 8
##   11 | 4
```

```
##   11 | 69
##   12 | 0
##   12 | 5
```

Capítulo 3

Questão 18

18. Considere o CD-Municípios e tome somente os 15 maiores, relativamente à sua população. Calcule $q(0, 1)$, $q(0, 2)$, q_1, q_2, q_3.

```
Pop <- cd_municipios$populacao

#Ordenando o vetor em  ordem decrescente:
Pop <- sort(Pop,decreasing = TRUE)

#Ordenando o novo vetor que contém as 15 maiores.
Pop <- sort(Pop[1:15])

# Novo conjunto de dados
Pop
```

```
##  [1]  84.7  92.4 101.8 102.3 116.0 119.4 129.8 135.8 151.6 187.7 201.5 210.9
## [13] 224.6 556.9 988.8
```

```
# Quantis q(0,1), q(0,2), q1=q(0,25), q2=q(0,5)=Mediana e q3=q(0,75)
quantile(Pop, probs = c(0.1, 0.2, 0.25, 0.5, 0.75))
```

```
##    10%    20%    25%    50%    75%
##  96.16 102.20 109.15 135.80 206.20
```

Questão 42

42. Calcule o desvio absoluto mediano para as populações do CD-Brasil.

Para essa e outras questões, vamos precisar do conceito de desvio absoluto mediano (dam):

Desvio absoluto mediano. Esta é uma medida de dispersão dos dados $x_1, \ldots, x_n$, definida por:

$$\text{dam} = \text{med}_{1 \leq j \leq n} | x_j - \text{med}_{1 \leq i \leq n}(x_i) | .$$

Ou seja, calculamos a mediana dos dados, depois os desvios absolutos dos dados em relação à mediana e, finalmente, a mediana desses desvios absolutos.

ºPopulação Urbana

```
Pop_Urb <- cd_brasil$pop_urbana
Pop_Urb <- Pop_Urb[!is.na(Pop_Urb)]

Desv_Abs_Median <- median(abs(Pop_Urb-median(Pop_Urb)))
Desv_Abs_Median
```

```
## [1] 1413142
```

ºPopulação Rural

```r
Pop_Rur <- cd_brasil$pop_rural
Pop_Rur <- sort(Pop_Rur[!is.na(Pop_Rur)])

Desv_Abs_Median <- median(abs(Pop_Rur-median(Pop_Rur)))
Desv_Abs_Median
```

```
## [1] 546852
```

Questão 43

43. Calcule as principais medidas de posição e dispersão (incluindo a média aparada e o dam) para:
    (a) variável CO no CD-Poluição;

(a) ºMedidas de posição

```r
Mono_Carb <- cd_poluicao$co
#Média
mean(Mono_Carb)
```

```
## [1] 7.464167
```

```r
#Mediana
median(Mono_Carb)
```

```
## [1] 7.2
```

```r
#Moda
#OBS.: O próprio conjunto de dados fornecido ("dados.RData") nos dá uma função "moda2"
#que calcula a moda de um conjunto, mas não vamos considerar essa função
#para a resolução da questão. Ex.:
moda2(Mono_Carb)
```

```
## [1] "6.2"
```

```r
#Para calcular a moda, pelo fato de o R não possuir uma função built-in para tal, assim
#como tem para calcular a média ou a mediana, vamos utilizar outro método.
#Aqui, pegamos as repetições das observações com a função "table"
table(Mono_Carb)
```

```
## Mono_Carb
##  4.7  5.1  5.2  5.5  5.6  5.7  5.8  5.9  6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8
##    2    1    1    2    1    3    1    1    5    9    5    6    1    4    5    1
##  6.9    7  7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9    8  8.1  8.2  8.3  8.4
##    8    2    1    3    2    3    2    2    5    5    8    1    1    1    2    1
##  8.5  8.6  8.7  8.8  8.9    9  9.1  9.4  9.5  9.7 10.1 10.3 10.8 11.4 11.6 11.9
##    2    1    1    1    3    1    2    1    2    1    1    3    1    1    1    1
##   12 12.5
##    1    1
```

```r
#Aqui, já conseguimos observar que o número 6.2 foi o que se repetiu mais vezes (9),
#mas queremos que o R busque isso para a gente.

#Então, o que queremos é o nome da "coluna" que corresponde ao valor máximo
#de repetições [table(Mono_Carb)]
names(table(Mono_Carb)[table(Mono_Carb)==max(table(Mono_Carb))])
```

```
## [1] "6.2"
```

ºMedidas de dispersão

```
#Desvio Médio
#Igual ao somatório do valor absoluto das distâncias de cada observação à média,
#dividido pelo total de observações.
print(sum(abs(Mono_Carb-mean(Mono_Carb)))/length(Mono_Carb))
```

```
## [1] 1.181653
```

```
#variância
#Aqui foi preciso converter a Variância Amostral para Variância Populacional
#multiplicando por (n-1) e dividindo por (n).
var(Mono_Carb)*(length(Mono_Carb)-1)/length(Mono_Carb)
```

```
## [1] 2.363799
```

```
#Desvio Padrão
#Igual à raiz quadrada da Variância.
Var <- var(Mono_Carb)*(length(Mono_Carb)-1)/length(Mono_Carb)
print(Var^(1/2))
```

```
## [1] 1.537465
```

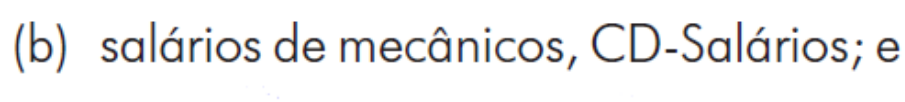
```
#Média Aparada [x(0,10)]
#Vamos ordenar as observações em ordem crescente:
Mono_Carb_O <- sort(Mono_Carb)
#Agora, vamos calcular a quantidade de observações (100 (\alpha)%,
#com  (\alpha)=0,10) a qual vamos precisar remover essa quantidade
#das menores observações e essa quantidade das maiores:
Qtd_Observ <- (length(Mono_Carb_O)*0.10)
Qtd_Observ
```

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

```
#Agora, removemos 12 das menores e das maiores observções, e calculamos a média
#(assim, teremos a média aparada):
Mono_Carb_R <- Mono_Carb_O[(Qtd_Observ+1):(length(Mono_Carb_O)-Qtd_Observ)]
mean(Mono_Carb_R)
```

```
## [1] 7.29375
```

(b)



ºMedidas de posição

```
Salarios <- cd_salarios$Mecânico
#Média
mean(Salarios)
```

```
## [1] 18.58907
```

```
#Mediana
median(Salarios)
```

```
## [1] 16.435
```

```
#Moda
#Usaremos o mesmo método do item (a):
names(table(Salarios)[table(Salarios)==max(table(Salarios))])
```

```
## [1] "20.068" "25.95"
```

ºMedidas de dispersão

```
#Desvio Médio
#Igual ao somatório do valor absoluto das distâncias de cada observação à média,
#dividido pelo total de observações.
print(sum(abs(Salarios-mean(Salarios)))/length(Salarios))
```

```
## [1] 9.768942
```

```
#variância
#Aqui foi preciso converter a Variância Amostral para Variância Populacional
#multiplicando por (n-1) e dividindo por (n).
var(Salarios)*(length(Salarios)-1)/length(Salarios)
```

```
## [1] 129.6838
```

```
#Desvio Padrão
#Igual à raiz quadrada da Variância.
Var <- var(Salarios)*(length(Salarios)-1)/length(Salarios)
print(Var^(1/2))
```

```
## [1] 11.38788
```

```
#Média Aparada [x(0,10)]
#Vamos ordenar as observações em ordem crescente:
Salarios_O <- sort(Salarios)

#Agora, vamos calcular a quantidade de observações (100 (\alpha)%,
#com  (\alpha)=0,10) a qual vamos precisar remover essa quantidade
#das menores observações e essa quantidade das maiores:
Qtd_Observ <- (length(Salarios_O)*0.10)
Qtd_Observ
```

```
## [1] 3
```

```
#Agora, removemos 3 das menores e das maiores observções, e calculamos a média
#(assim, teremos a média aparada):
Salarios_R <- Salarios_O[(Qtd_Observ+1):(length(Salarios_O)-Qtd_Observ)]
mean(Salarios_R)
```

```
## [1] 17.9245
```

(c)

(c) variável preço, CD-Veículos.

ºMedidas de posição

```
Preco <- cd_veiculos$preco
#Média
mean(Preco)
```

```
## [1] 13956.1
```

```
#Mediana
median(Preco)
```

```
## [1] 11824
```

```
#Moda
#Utilizaremos o mesmo metódo descrito no item (a):
names(table(Preco)[table(Preco)==max(table(Preco))])
```

```
##  [1] "5257"  "5680"  "6176"  "6260"  "6316"  "6340"  "6700"  "7742"  "7780"
## [10] "9300"  "9440"  "10532" "10767" "11386" "11630" "12018" "12890" "12923"
## [19] "13140" "13700" "13840" "14460" "15520" "16346" "21500" "22200" "24632"
## [28] "31640" "33718" "38850"
```

```
#percebemos que a Moda é igual ao número total de observações:
length(Preco)==length(names(table(Preco)[table(Preco)==max(table(Preco))]))
```

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

```
#Logo, concluímos que a distribuição não tem um valor que se repete, ou seja,
#não possui uma moda.
```

ºMedidas de dispersão

```
#Desvio Médio
#Igual ao somatório do valor absoluto das distâncias de cada observação à média,
#dividido pelo total de observações.
print(sum(abs(Preco-mean(Preco)))/length(Preco))
```

```
## [1] 6217.407
```

```
#variância
#Aqui foi preciso converter a Variância Amostral para Variância Populacional
#multiplicando por (n-1) e dividindo por (n).
var(Preco)*(length(Preco)-1)/length(Preco)
```

```
## [1] 72087425
```

```
#Desvio Padrão
#Igual à raiz quadrada da Variância.
Var <- var(Preco)*(length(Preco)-1)/length(Preco)
print(Var^(1/2))
```

```
## [1] 8490.431
```

```
#Média Aparada [x(0,10)]
#Vamos ordenar as observações em ordem crescente:
Preco_O <- sort(Preco)

#Agora, vamos calcular a quantidade de observações (100 (\alpha)%,
#com (\alpha)=0,10) a qual vamos precisar remover essa quantidade
#das menores observações e essa quantidade das maiores:
Qtd_Observ <- (length(Preco_O)*0.10)
Qtd_Observ
```

```
## [1] 3
```

```
#Agora, removemos 3 das menores e das maiores observções, e calculamos a média
#(assim, teremos a média aparada):
```

```r
Preco_R <- Preco_O[(Qtd_Observ+1):(length(Preco_O)-Qtd_Observ)]
mean(Preco_R)
```

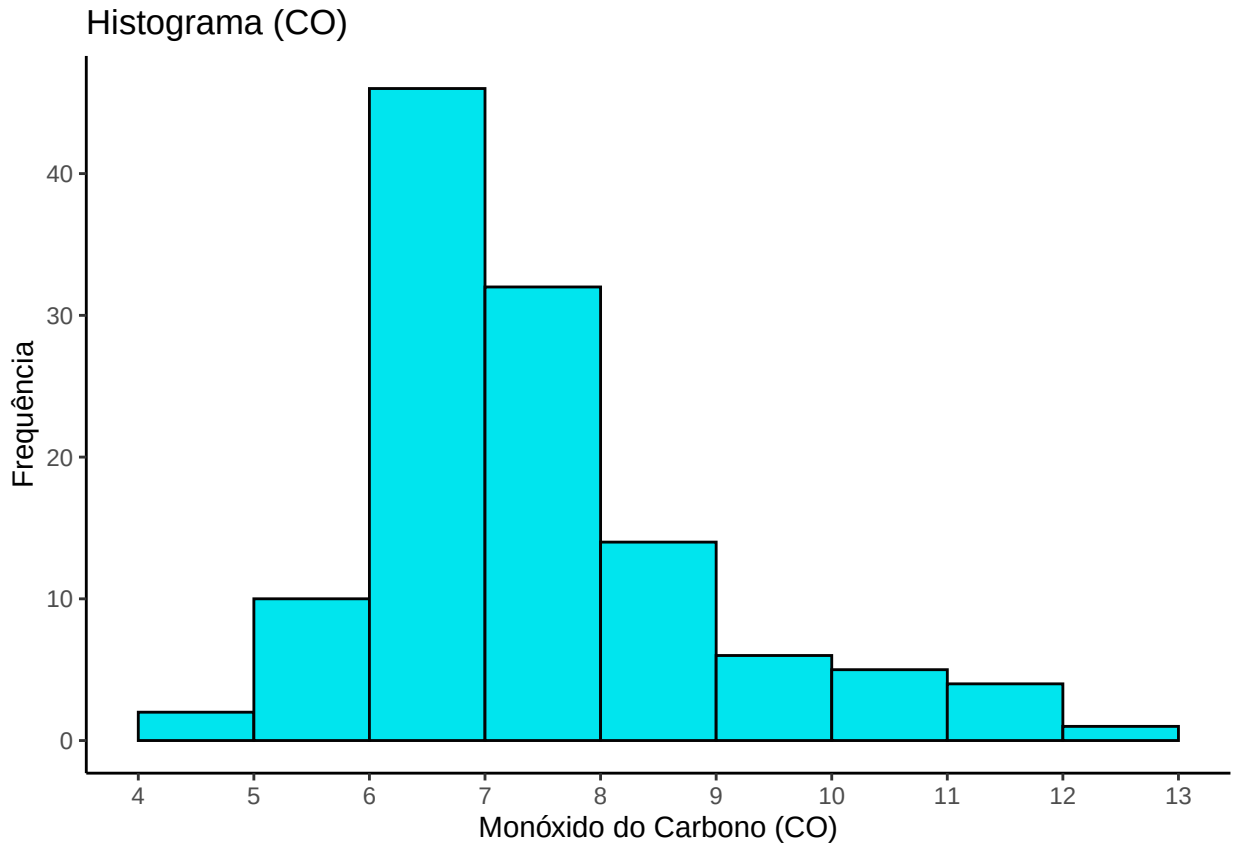
```
## [1] 12390.08
```

Questão 44

44. Construa os histogramas, ramo-e-folhas e desenhos esquemáticos para as variáveis do problema anterior.

ºMonóxido de Carbono (CO)

```r
ggplot(data = cd_poluicao, aes(x = co)) +
  geom_histogram(breaks = seq(4, 13),
                 col="gray1",
                 fill = "turquoise2") +
  labs(title = "Histograma (CO)",
       x = "Monóxido do Carbono (CO)",
       y = "Frequência") +
  scale_x_continuous(breaks=seq(4,13)) +
  theme_classic()
```



Ramo-e-folhas (CO)

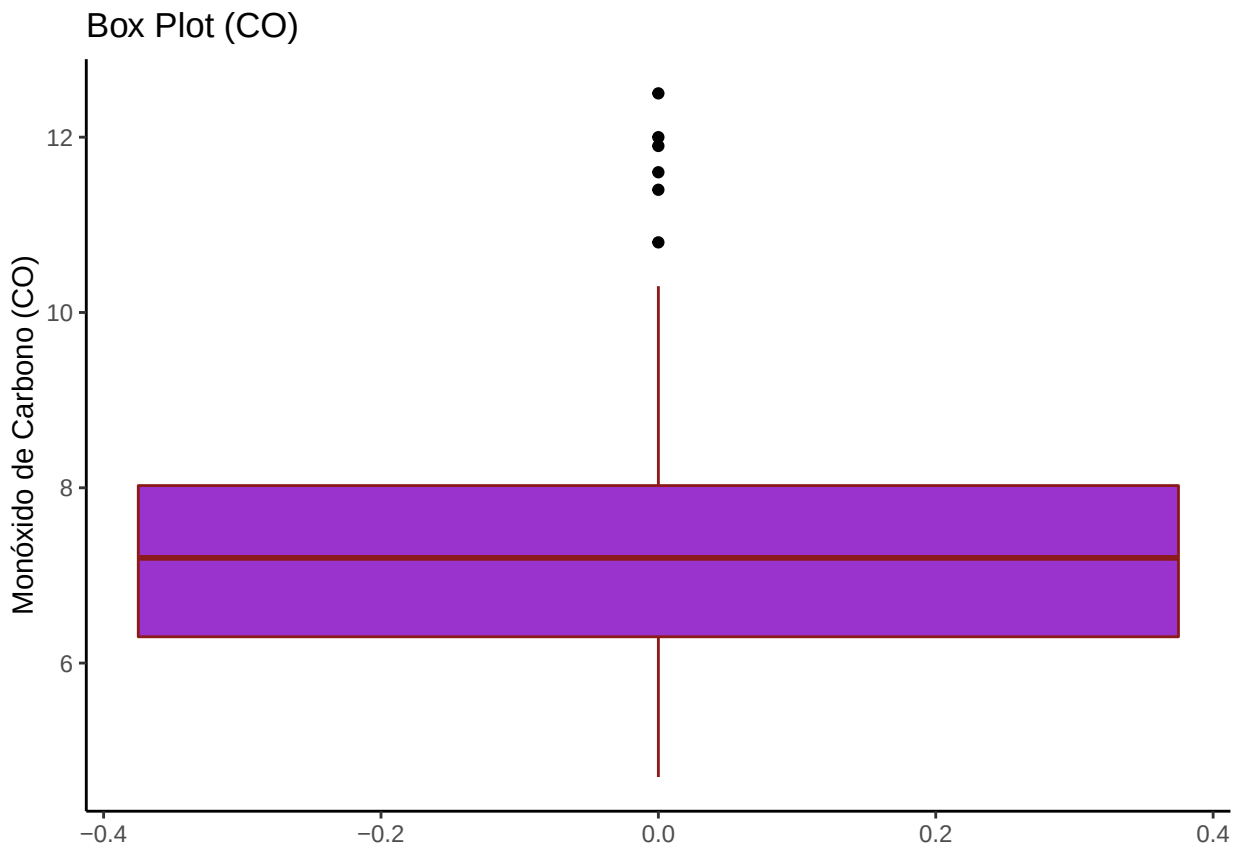
```r
stem(cd_poluicao$co)
```

```
##
##   The decimal point is at the |
```

```
##
##    4 | 77
##    5 | 12
##    5 | 55677789
##    6 | 11111222222222333333444444
##    6 | 5666677777899999999
##    7 | 00122233444
##    7 | 5566777778888899999999
##    8 | 012334
##    8 | 55678999
##    9 | 0114
##    9 | 557
##   10 | 1333
##   10 | 8
##   11 | 4
##   11 | 69
##   12 | 0
##   12 | 5
```

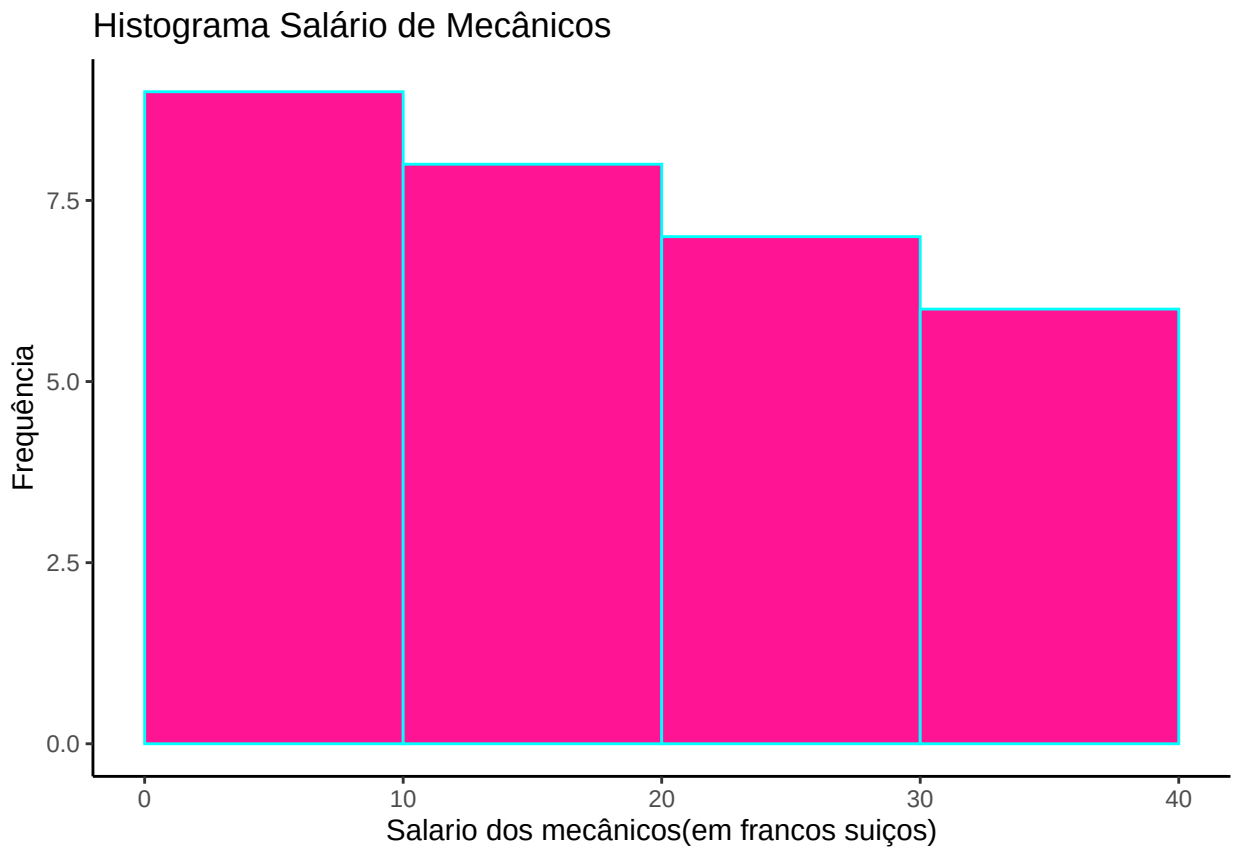
Desenho esquemático

```
ggplot(data = cd_poluicao, aes(y=co)) +
  geom_boxplot(outlier.colour = "black",
               col = "firebrick4",
               fill = "darkorchid3") +
  labs(title="Box Plot (CO)",
       y = "Monóxido de Carbono (CO)") +
  theme_classic()
```



ºSalário de Mecânicos

```
ggplot(data = cd_salarios, aes(x = Mecânico)) +
  geom_histogram(fill = "deeppink",
                 col = "cyan1",
                 breaks = seq(0,40,10)) +
  labs(title = "Histograma Salário de Mecânicos",
       x = "Salario dos mecânicos(em francos suiços)",
       y = "Frequência") +
  theme_classic()
```



Ramo-e-folhas

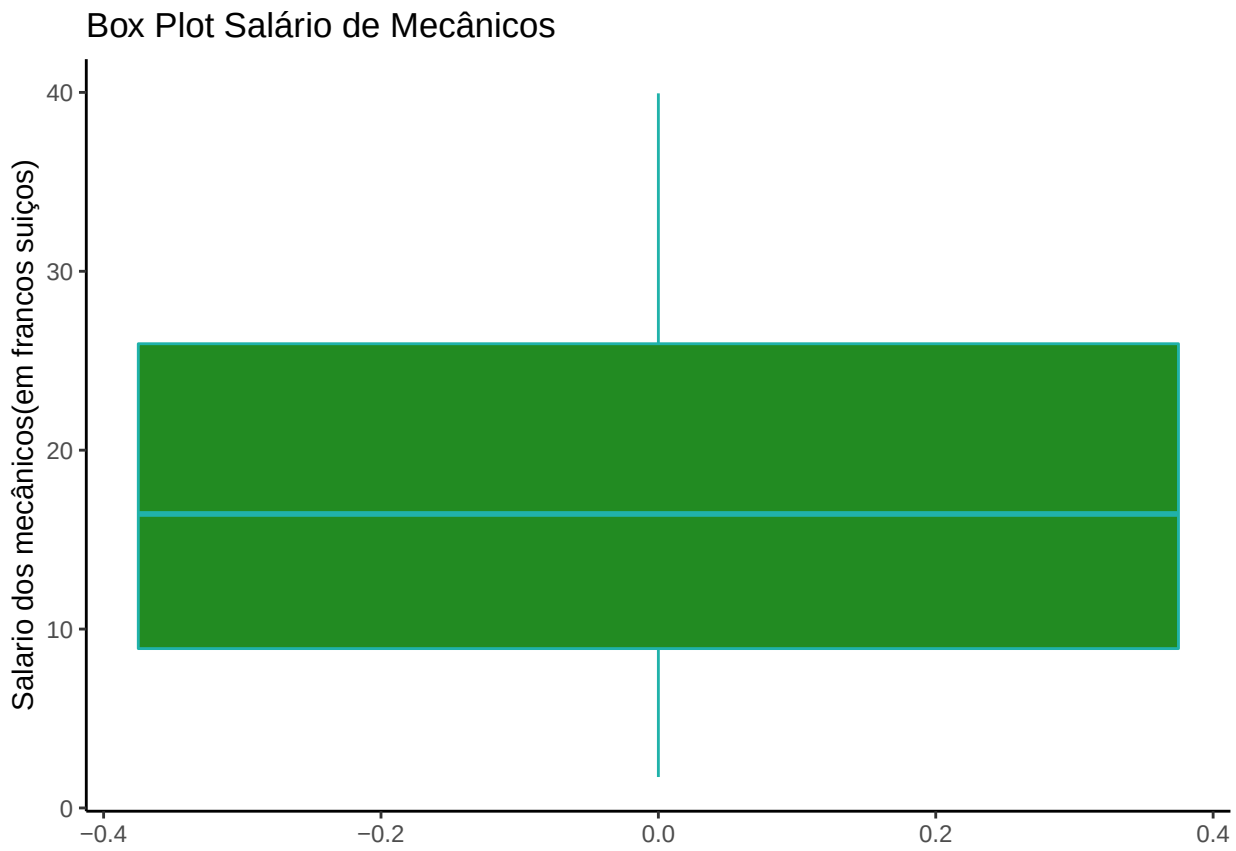
```
stem(cd_salarios$Mecânico, scale = 4)
```

```
##
##   The decimal point is at the |
##
##    1 | 7
##    2 |
##    3 | 8
##    4 |
##    5 | 28
##    6 | 26
##    7 |
##    8 | 37
##    9 | 7
##   10 |
##   11 | 1
##   12 | 15
##   13 | 58
##   14 |
##   15 | 9
##   16 |
##   17 | 06
##   18 |
##   19 |
##   20 | 11
```

```
##   21 |
##   22 |
##   23 | 5
##   24 |
##   25 | 5
##   26 | 005
##   27 |
##   28 |
##   29 |
##   30 |
##   31 |
##   32 | 9
##   33 |
##   34 | 6
##   35 |
##   36 | 3
##   37 | 0
##   38 |
##   39 | 89
```

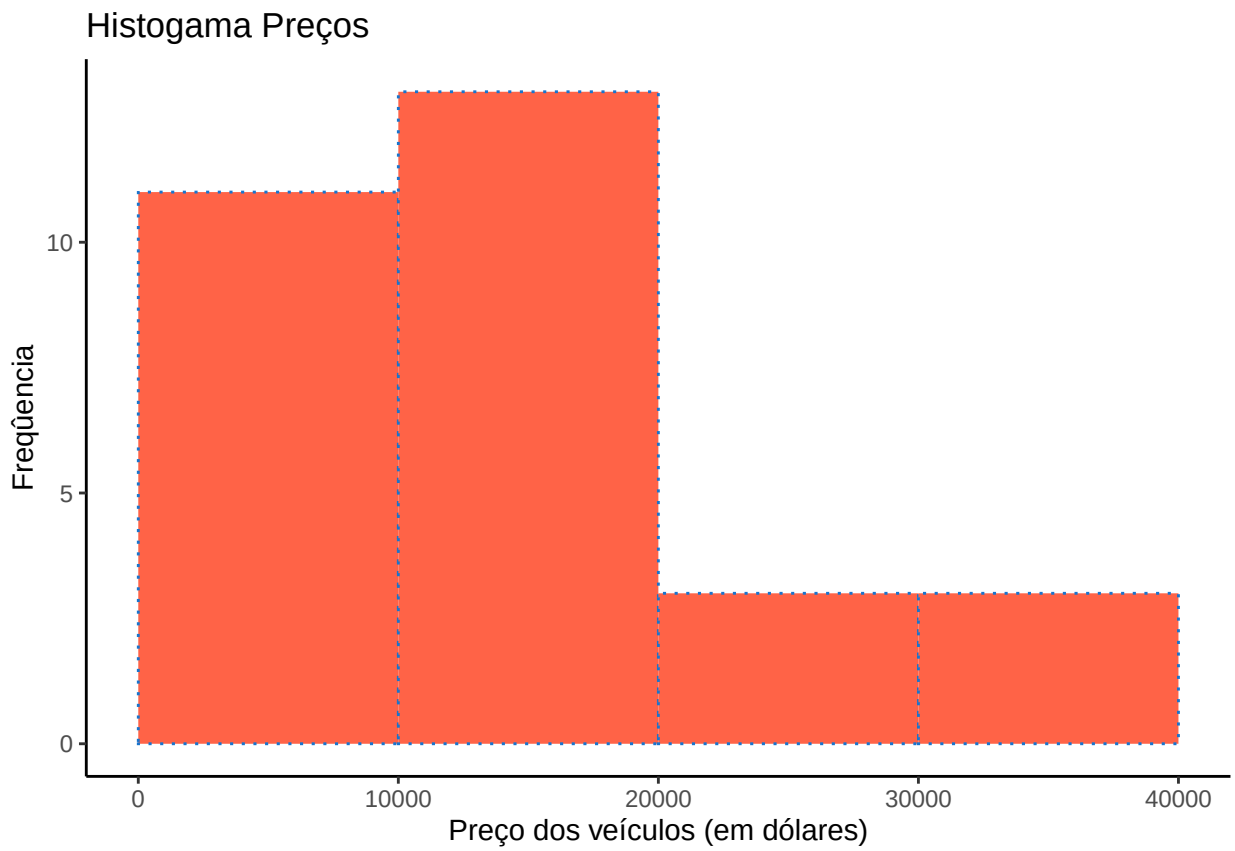
Desenho esquemático

```
ggplot(data = cd_salarios, aes(y = Mecânico)) +
  geom_boxplot(fill = "forestgreen",
               colour = "lightseagreen") +
  labs(title = "Box Plot Salário de Mecânicos",
       y = "Salario dos mecânicos(em francos suiços)") +
  theme_classic()
```



ºPreço dos Veículos

```
ggplot(data = cd_veiculos, aes(x = preco)) +
  geom_histogram(linetype="dotted",
                 col = "dodgerblue3",
                 fill = "tomato",
                 breaks = seq(0,40000,10000)) +
  labs(title = "Histogama Preços",
       x = "Preço dos veículos (em dólares)",
       y = "Freqûencia") +
  theme_classic()
```



Ramo-e-folhas

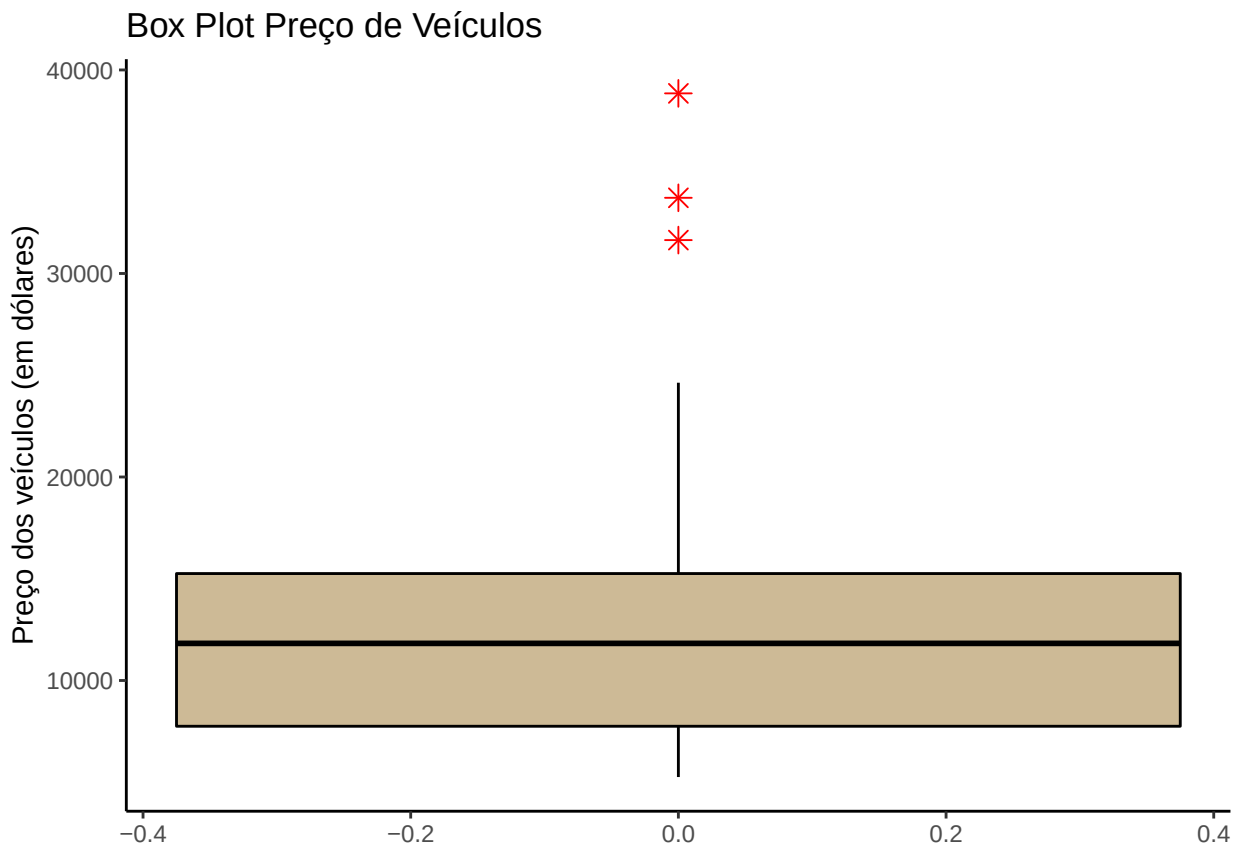
```
stem(cd_veiculos$preco, scale = 4)
```

```
##
##   The decimal point is 3 digit(s) to the right of the |
##
##    5 | 37
##    6 | 23337
##    7 | 78
##    8 |
##    9 | 34
##   10 | 58
##   11 | 46
##   12 | 099
##   13 | 178
##   14 | 5
##   15 | 5
##   16 | 3
##   17 |
##   18 |
##   19 |
##   20 |
##   21 | 5
##   22 | 2
##   23 |
##   24 | 6
```

```
##    25 |
##    26 |
##    27 |
##    28 |
##    29 |
##    30 |
##    31 | 6
##    32 |
##    33 | 7
##    34 |
##    35 |
##    36 |
##    37 |
##    38 | 9
```

Desenho esquemático

```
ggplot(data = cd_veiculos, aes(y = preco)) +
  geom_boxplot(fill = "wheat3",
               col = "black",
               outlier.color = "red",
               outlier.shape = 8,
               outlier.size = 3) +
  labs(title = "Box Plot Preço de Veículos",
       y = "Preço dos veículos (em dólares)") +
  theme_classic()
```

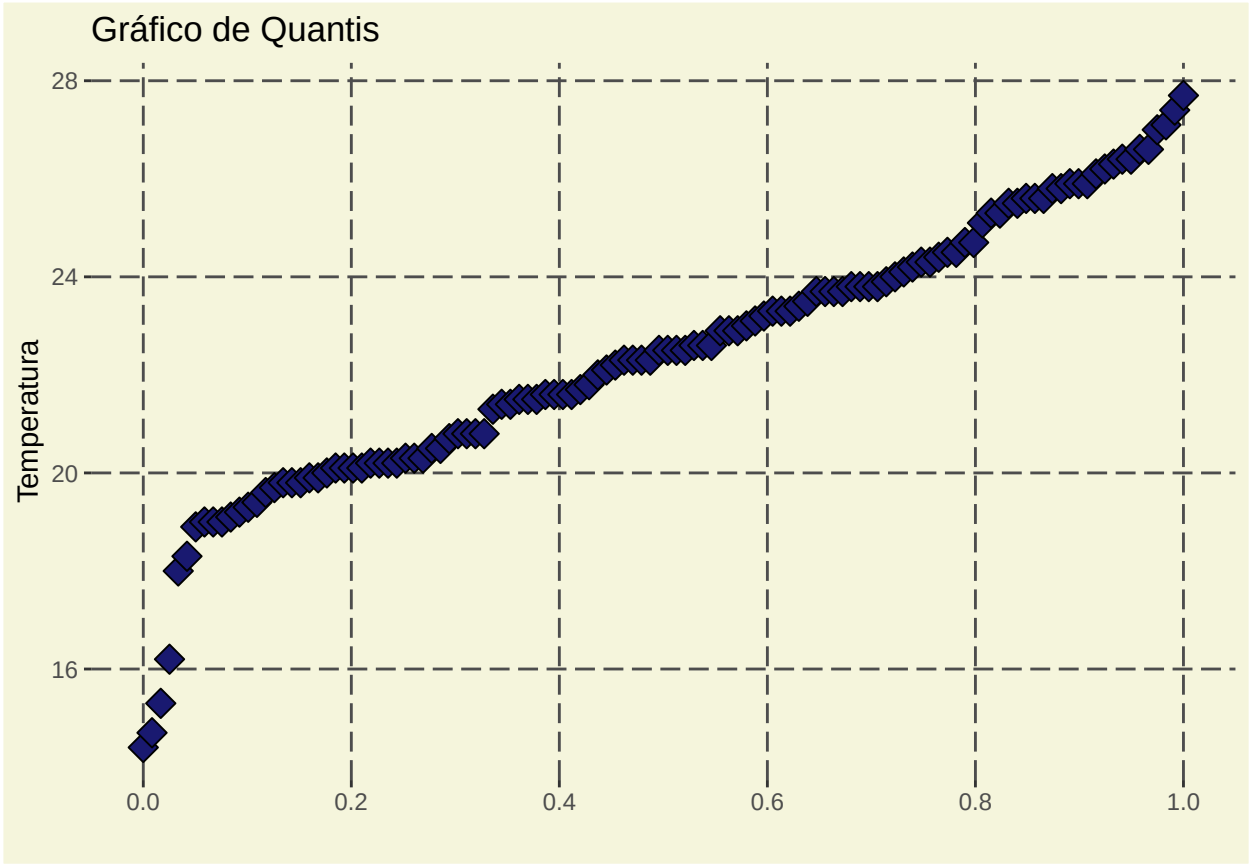


Questão 46

46. Para o CD-Temperaturas e para a variável *temperatura de Ubatuba*, obtenha um gráfico de quantis e um gráfico de simetria. Os dados são simétricos? Comente.

```
Ubatuba <- cd_temperaturas$ubatuba
Ubatuba <- sort(Ubatuba)

ggplot(data = cd_temperaturas,
       aes(x = seq(0,1,length.out = 120), y = Ubatuba)) +
    geom_point(shape = 23,
               fill = "midnightblue",
               size = 3.8) +
    scale_x_continuous(breaks = seq(0,1,0.2)) +
    labs(title = "Gráfico de Quantis",
         x = " ",
         y = "Temperatura") +
    theme(plot.background = element_rect(fill = "beige"),
          panel.background = element_rect(fill = "beige"),
          panel.grid = element_line(linetype = "longdash",
                                    colour = "gray30"),
          panel.grid.minor = element_line(colour = "beige"))
```




```
#Vamos usar que ui = q2(mediana) - xi e vi = x[n+1-i] - q2
#Então como i vai do primeiro elemento (x1) à mediana, vamos considerar i como de 1 até
#metade das observações (i = n/2)
```

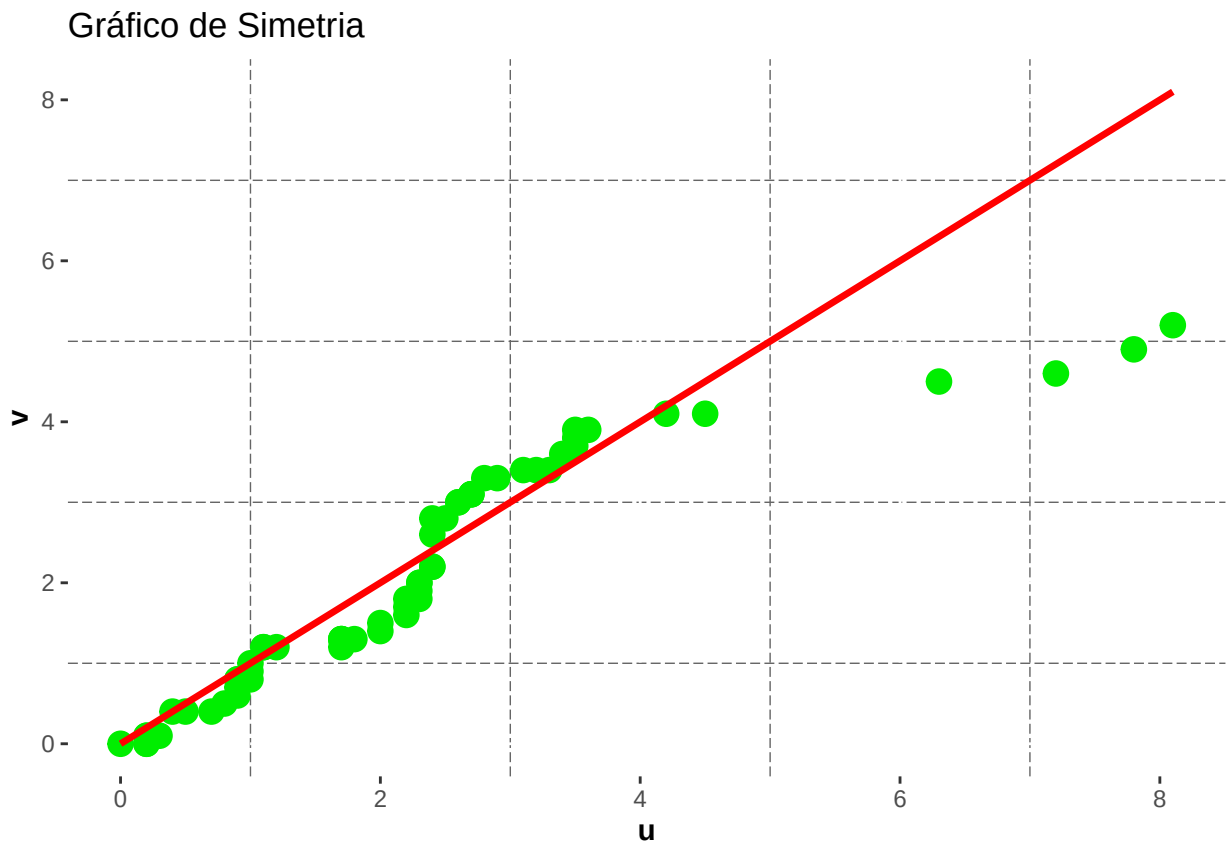
```r
i <- 1:(length(Ubatuba)/2)
#Temos que  1 ($\leq$)i ($\geq$)60
i
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
## [51] 51 52 53 54 55 56 57 58 59 60
```

```r
#Agora, vamos calcular os ui
ui <- median(Ubatuba)-Ubatuba[i]

#Para calcular os vi, vamos apenas reordernar o vetor em ordem decrescente e
#naturalmente obteremos o resultado que se quer.
Ubatuba_D <- sort(Ubatuba, decreasing = TRUE)
vi <- Ubatuba_D[i]-median(Ubatuba_D)

#Por fim, podemos fazer o gráfico de simetria em relação à reta u=v (em vermelho)
ggplot(data = NULL, aes(x = ui, y = vi)) +
  geom_point(shape = 19,
             colour = "green2",
             size = 4) +
  geom_line(data = NULL, aes(x = u, y = v),
            col = "red",
            size = 1.2) +
  theme(panel.grid = element_line(linetype = "dashed",
                                  colour = "white"),
        panel.grid.minor = element_line(linetype = "longdash",
                                        colour = "gray40"),
        panel.background = element_rect(fill = "white"),
        axis.title = element_text(face = "bold")) +
  labs(title = "Gráfico de Simetria",
       x = "u",
       y = "v")
```



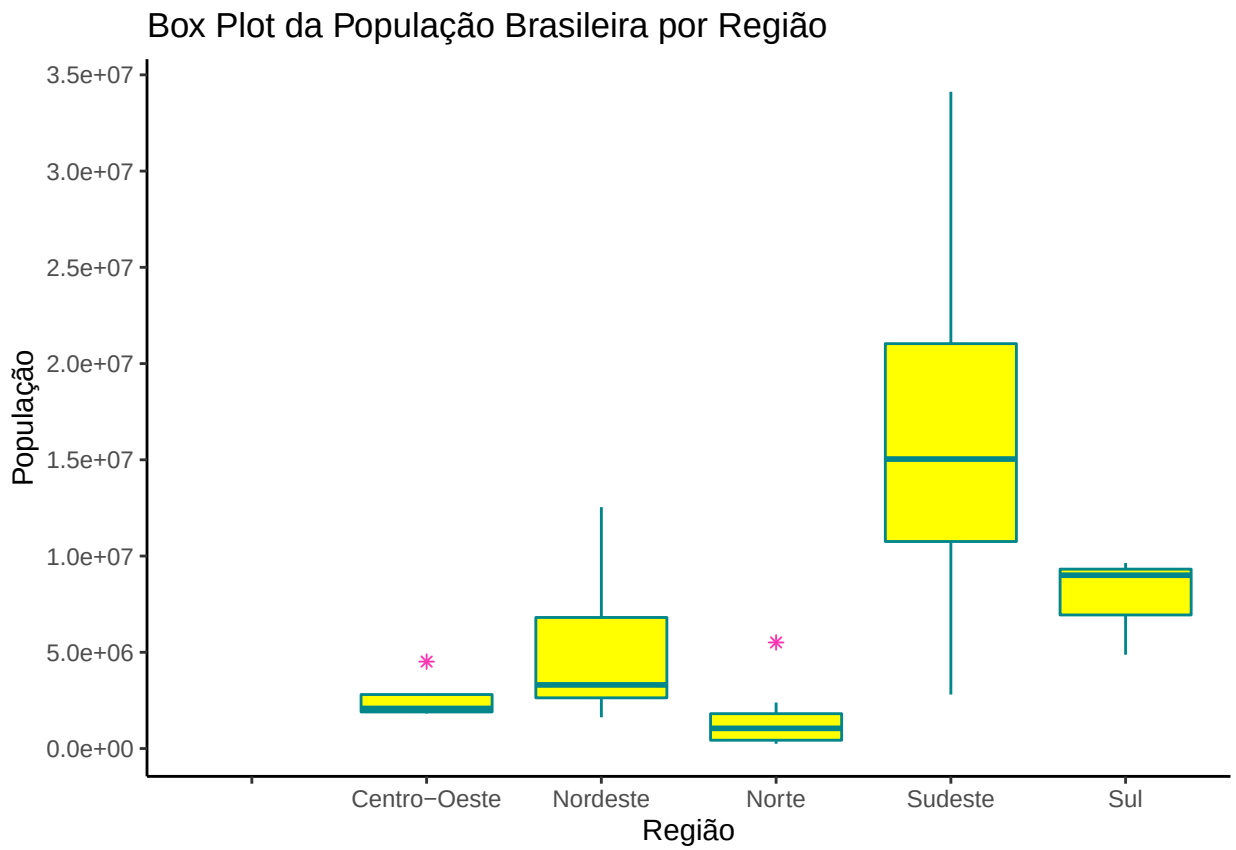
Capítulo 4

Questão 37

37. Analise a população total do CD-Brasil, segundo as regiões geográficas.

A partir da análise dos Box Plots das regiões geográficas brasileiras, notamos a região sudeste com uma distribuição anormal em relação às demais e isso se deve ao fato de conter o Estado de São Paulo, que possui uma população bastante acima da média da sua região e da média nacional. Também observamos que a região Norte é a que possui uma distribuição mais simétrica dentre as regiões, e que a região Sul, ao contrário da Nordeste e Centro-Oeste, tem uma distribuição assimétrica à esquerda.

```
ggplot(data = cd_brasil, aes(x = regiao, y = total)) +
  geom_boxplot(outlier.colour = "maroon1",
               outlier.shape = 8,
               fill = "yellow1",
               col = "turquoise4") +
  labs(title = "Box Plot da População Brasileira por Região",
       x = "Região",
       y = "População") +
  scale_y_continuous(breaks = seq(0,35000000,5000000)) +
  theme_classic()
```



Questão 39

39. Considere o CD-Poluição e as variáveis CO, temperatura e umidade. Faça gráficos de dispersão para pares de variáveis. Quais conclusões você pode obter?

Analisando os Gráficos de dispersão, percebe-se que existe associação entre as variáveis em questão. Uma vez que, por exemplo, se a Temperatura está alta, a Umidade também está, tomando o Gráfico 2 como exemplo.

```
CO <- sort(cd_poluicao$co)
Temperatura <- sort(cd_poluicao$temp)
Umidade <- sort(cd_poluicao$umid)

GDP1 <- ggplot(data = cd_poluicao, aes(x = CO, y = temp)) +
  geom_point(shape = 23,
             fill = "yellow1",
             size = 3) +
  scale_x_continuous(breaks = seq(0,14, 2),
                     limits = c(0,14)) +
  scale_y_continuous(limits = c(0,25)) +
  labs(title = "Gráfico de dispersão 1: Monóxido de Carbono (CO) X Temperatura",
       x = "Monóxido de Carbono (CO)",
       y = "Temperaura (°C)") +
  theme_classic() +
  theme(panel.grid = element_line(linetype = "dashed",
                                  colour = "white"),
        panel.grid.minor = element_line(linetype = "longdash",
```

```
                                        colour = "gray40"))


GDP2 <- ggplot(data = cd_poluicao, aes(x = Temperatura, y = umid)) +
  geom_point(shape = 24,
             fill = "magenta1",
             size = 3) +
  scale_x_continuous(breaks = seq(0,25,5),
                     limits = c(0,25)) +
  scale_y_continuous(breaks = seq(0,100,20),
                     limits = c(0,100)) +
  labs(title = "Gráfico de dispersão 2: Temperatura X Umidade",
       x = "Temperatura (°C)",
       y = "Umidade") +
  theme_classic() +
  theme(panel.grid = element_line(linetype = "dashed",
                                  colour = "white"),
        panel.grid.minor = element_line(linetype = "longdash",
                                        colour = "gray40"))

GDP3 <- ggplot(data = cd_poluicao, aes(x = Umidade, y = co)) +
  geom_point(shape = 25,
             fill = "mediumspringgreen",
             size = 3) +
  scale_x_continuous(breaks = seq(0,100,20),
                     limits = c(0,100)) +
  scale_y_continuous(breaks = seq(0,14, 2),
                     limits = c(0,14)) +
  labs(title = "Gráfico de disersão 3: Umidade X Monóxido de Carbono (CO)", x = "Umidade",
       y = "Monóxido de Carbono (CO)") +
  theme_classic() +
  theme(panel.grid = element_line(linetype = "dashed",
                                  colour = "white"),
        panel.grid.minor = element_line(linetype = "longdash",
                                        colour = "gray40"))
```

```
ggarrange(GDP1, GDP2, GDP3, ncol = 2, nrow = 2,
          heights = 2, widths = 15, vjust = -5)
```

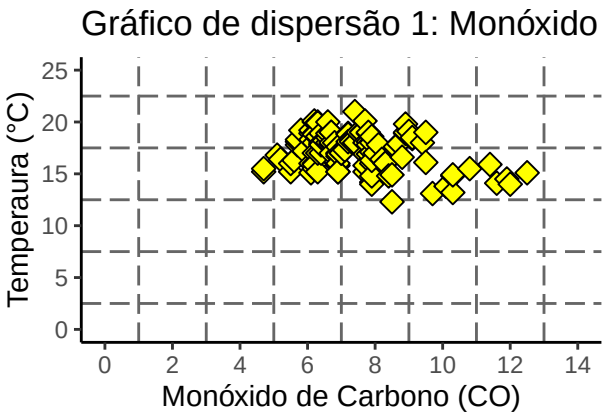


Gráfico de dispersão 1: Monóxido d
Temperaura (°C)
Monóxido de Carbono (CO)
0
5
10
15
20
25
0
2
4
6
8
10
12
14

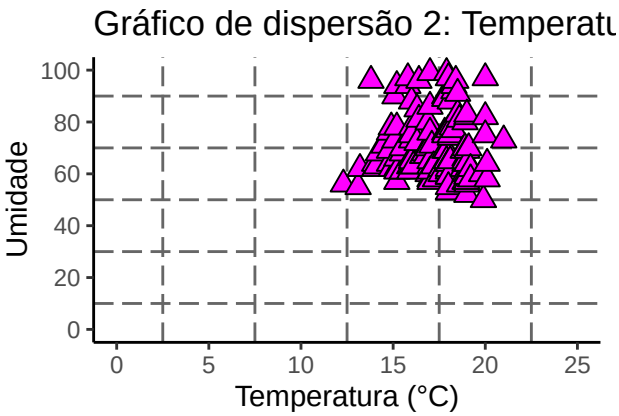


Gráfico de dispersão 2: Temperatu
Umidade
Temperatura (°C)
0
20
40
60
80
100
0
5
10
15
20
25

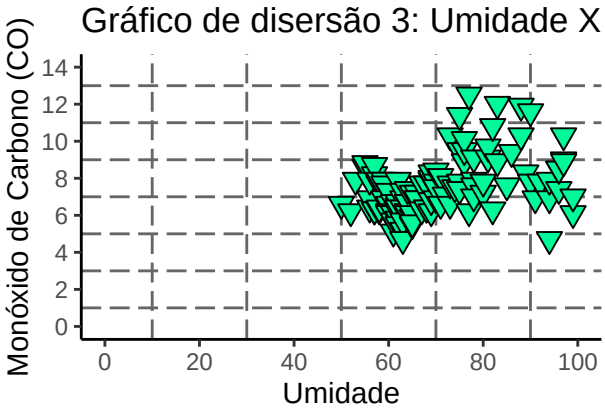


Gráfico de disersão 3: Umidade X Monóxido de Carbono (CO)
Monóxido de Carbono (CO)
Umidade
0
2
4
6
8
10
12
14
0
20
40
60
80
100