

Análises Ecológicas no R: Exercícios e Soluções

Fernando Rodrigues da Silva, Thiago Gonçalves-Souza, Gustavo Brant Paterno, Diogo Borges Probst

2022-02-11

Contents

Cap. 4 - Introdução ao R	7
Cap. 5 - Tidyverse	21
Cap. 6 - Visualização de dados	27
Cap. 7 - Modelos lineares	37
Cap. 10 - Rarefação	59
Cap. 11 - Estimadores de riqueza	65
Cap. 12 - Diversidade Taxônomic	73
Cap. 13 - Diversidade Filogenética	81
Cap. 15 - Dados geoespaciais no R	95

Sobre

Aqui você encontra os **exercícios e soluções** do livro Análises Ecológicas no R.

Cap. 4 - Introdução ao R

4.1 Use o R para verificar o resultado da operação $7 + 7 \div 7 + 7 \times 7 - 7$.

Solução:

```
7 + 7 / 7 + 7 * 7 - 7  
#> [1] 50
```

4.2 Verifique através do R se 3×2^3 é maior que 2×3^2 .

Solução:

```
3 * 2^3 > 2 * 3^2  
#> [1] TRUE
```

4.3 Crie dois objetos (qualquer nome) com os valores 100 e 300. Multiplique esses objetos (função `prod()`) e atribua ao objeto **mult**. Faça o logaritmo natural (função `log()`) do objeto **mult** e atribua ao objeto **ln**.

Solução:

```
obj1 <- 100  
obj2 <- 300  
mult <- prod(obj1, obj2)  
ln <- log(obj1, obj2)
```

4.4 Quantos pacotes existem no CRAN nesse momento? Execute essa combinação no Console: `nrow(available.packages(repos = "http://cran.r-project.org"))`.

Solução:

```
nrow(available.packages(repos = "http://cran.r-project.org"))  
#> [1] 18913
```

4.5 Instale o pacote **tidyverse** do CRAN.

Solução:

```
install.packages("tidyverse", dependencies = TRUE)
```

4.6 Escolha números para jogar na mega-sena usando o R, nomeando o objeto como **mega**. Lembrando: são 6 valores de 1 a 60 e atribua a um objeto.

Solução:

```
mega <- sample(x = 1:60, size = 6, replace = FALSE)
mega
#> [1] 25 53 9 22 13 20
```

4.7 Crie um fator chamado **tr**, com dois níveis (“cont” e “trat”) para descrever 100 locais de amostragem, 50 de cada tratamento. O fator deve ser dessa forma cont, cont, cont, ..., cont, trat, trat, ..., trat.

Solução:

```
tr <- factor(c(rep("cont", each = 50), rep("trat", each = 50)))
tr
#> [1] cont cont cont cont cont cont cont cont cont cont cont
#> [12] cont cont cont cont cont cont cont cont cont cont cont
#> [23] cont cont cont cont cont cont cont cont cont cont cont
#> [34] cont cont cont cont cont cont cont cont cont cont cont
#> [45] cont cont cont cont cont cont cont trat trat trat trat trat
#> [56] trat trat trat trat trat trat trat trat trat trat trat
#> [67] trat trat trat trat trat trat trat trat trat trat trat
#> [78] trat trat trat trat trat trat trat trat trat trat trat
#> [89] trat trat trat trat trat trat trat trat trat trat trat
#> [100] trat
#> Levels: cont trat
```

4.8 Crie uma matriz chamada **ma**, resultante da disposição de um vetor composto por 1000 valores aleatórios entre 0 e 10. A matriz deve conter 100 linhas e ser disposta por colunas.

Solução:

```
ma <- matrix(sample(0:10, 1000, rep = TRUE), nrow = 100, byrow = FALSE)
ma
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
#> [1,] 8 0 2 7 5 5 1 0 2 8
#> [2,] 0 10 4 9 7 7 7 3 5 0
#> [3,] 3 3 1 0 10 10 0 9 3 5
#> [4,] 5 0 0 2 8 10 0 9 4 2
#> [5,] 7 3 10 4 6 9 10 4 1 0
#> [6,] 2 10 6 9 6 6 3 3 4 6
#> [7,] 9 7 0 5 3 2 3 2 9 7
#> [8,] 5 7 8 7 10 7 0 10 0 4
#> [9,] 2 10 8 2 4 5 1 9 4 10
#> [10,] 10 7 0 2 3 5 5 6 5 7
#> [11,] 3 10 5 8 2 3 0 9 7 2
```


#>	[12,]	9	9	1	5	9	7	7	2	3	4
#>	[13,]	10	2	7	10	8	9	4	8	0	2
#>	[14,]	0	9	4	9	9	7	10	6	2	6
#>	[15,]	5	3	9	9	7	6	0	3	3	6
#>	[16,]	7	0	5	10	7	4	7	6	9	9
#>	[17,]	4	8	1	7	7	10	0	6	2	5
#>	[18,]	6	10	5	2	3	5	10	10	5	10
#>	[19,]	3	2	6	4	10	1	6	4	1	1
#>	[20,]	3	10	5	8	0	4	9	4	6	2
#>	[21,]	0	9	7	1	6	5	4	3	0	10
#>	[22,]	2	1	6	0	7	2	6	7	7	7
#>	[23,]	3	9	2	0	4	3	4	1	4	5
#>	[24,]	5	7	6	9	4	5	2	2	5	2
#>	[25,]	5	4	8	5	9	10	4	3	1	8
#>	[26,]	7	2	3	6	4	3	2	5	10	1
#>	[27,]	7	3	1	7	7	2	2	8	7	1
#>	[28,]	10	7	10	1	8	4	1	6	9	3
#>	[29,]	3	7	5	4	7	1	1	8	9	2
#>	[30,]	0	2	5	6	6	10	9	6	5	0
#>	[31,]	6	0	8	8	3	8	8	1	3	1
#>	[32,]	4	9	1	9	9	6	7	8	8	3
#>	[33,]	7	9	4	7	5	0	0	6	1	10
#>	[34,]	6	4	1	0	0	3	8	2	1	0
#>	[35,]	4	5	5	10	2	9	0	9	1	6
#>	[36,]	7	8	5	10	8	3	4	2	9	7
#>	[37,]	3	9	3	10	5	2	3	10	6	10
#>	[38,]	8	9	0	10	2	5	2	2	8	2
#>	[39,]	5	9	8	3	4	8	6	4	9	8
#>	[40,]	3	2	8	0	7	6	5	5	1	0
#>	[41,]	3	8	9	6	6	8	7	2	5	0
#>	[42,]	5	1	6	7	8	7	7	6	8	1
#>	[43,]	7	1	5	9	0	2	7	6	0	3
#>	[44,]	1	3	9	10	2	9	10	3	6	8
#>	[45,]	7	1	4	5	7	10	5	5	1	2
#>	[46,]	0	7	0	10	5	2	4	1	3	9
#>	[47,]	9	3	0	6	7	5	4	0	0	0
#>	[48,]	8	8	4	2	9	9	4	8	1	7
#>	[49,]	9	2	4	4	7	5	2	5	7	3
#>	[50,]	3	0	8	1	3	3	3	1	4	0
#>	[51,]	9	6	8	10	9	9	8	4	8	2
#>	[52,]	4	9	10	9	7	1	8	7	5	10
#>	[53,]	4	10	6	0	7	1	4	10	9	8
#>	[54,]	7	4	0	6	7	0	0	0	0	7
#>	[55,]	5	8	8	1	3	9	10	8	0	4
#>	[56,]	3	9	1	4	4	9	10	8	5	9

#> [57,]	5	10	3	9	4	0	4	10	1	4
#> [58,]	2	0	3	1	3	6	3	0	3	6
#> [59,]	10	4	3	3	1	7	3	1	5	8
#> [60,]	9	4	4	9	2	4	4	10	3	9
#> [61,]	6	0	3	4	0	6	5	8	6	4
#> [62,]	8	1	2	0	0	0	5	7	0	9
#> [63,]	8	10	1	3	3	5	9	0	5	3
#> [64,]	10	9	0	5	5	6	7	4	5	2
#> [65,]	7	7	9	2	5	7	3	0	4	2
#> [66,]	5	2	7	6	7	3	4	2	6	7
#> [67,]	8	8	7	0	0	5	9	10	3	9
#> [68,]	9	2	0	9	3	3	5	10	7	0
#> [69,]	9	8	4	3	4	10	8	7	2	3
#> [70,]	3	0	7	4	6	1	7	7	7	4
#> [71,]	0	0	4	9	7	3	5	6	7	4
#> [72,]	9	7	8	4	9	8	10	10	10	0
#> [73,]	8	3	6	10	4	0	2	1	5	4
#> [74,]	0	10	6	7	9	0	4	6	8	2
#> [75,]	10	10	6	3	3	6	7	1	6	4
#> [76,]	9	6	9	0	1	7	3	0	9	3
#> [77,]	3	9	6	0	1	5	5	4	5	8
#> [78,]	6	6	9	6	0	7	6	7	10	4
#> [79,]	8	7	2	8	2	6	1	9	0	2
#> [80,]	6	5	2	1	8	4	1	10	1	5
#> [81,]	6	1	3	10	4	5	6	9	4	9
#> [82,]	3	4	3	6	0	0	5	4	0	3
#> [83,]	6	5	8	4	1	6	3	9	3	5
#> [84,]	8	5	1	0	9	9	1	7	10	8
#> [85,]	4	0	6	2	9	10	2	7	3	9
#> [86,]	2	0	1	7	9	2	2	6	6	3
#> [87,]	6	2	1	9	8	6	2	0	4	5
#> [88,]	7	2	1	10	9	7	1	3	4	1
#> [89,]	5	1	0	1	5	1	10	9	6	7
#> [90,]	10	0	5	2	7	6	1	0	5	1
#> [91,]	6	4	7	2	10	4	8	10	6	5
#> [92,]	0	10	8	5	4	3	8	10	5	9
#> [93,]	5	3	9	6	9	0	6	1	0	9
#> [94,]	10	0	5	9	3	10	6	4	7	1
#> [95,]	6	6	1	1	7	4	3	1	4	3
#> [96,]	7	4	1	9	8	10	1	1	10	1
#> [97,]	9	1	0	4	6	10	2	2	6	2
#> [98,]	2	4	0	5	6	4	5	6	5	2
#> [99,]	5	1	4	10	6	8	3	5	4	0
#> [100,]	7	1	3	2	10	10	9	8	2	3

4.9 Crie um data frame chamado **df**, resultante da composição dos vetores:

1. id: 1:50
2. sp: sp01, sp02, ..., sp49, sp50
3. ab: 50 valores aleatórios entre 0 a 5

Solução:

```
df <- data.frame(id = 1:50,
                 sp = c(paste0("sp0", 1:9), paste0("sp", 10:50)),
                 ab = sample(0:5, 50, rep = TRUE))

df
#>      id    sp ab
#>  1     1 sp01  0
#>  2     2 sp02  3
#>  3     3 sp03  3
#>  4     4 sp04  5
#>  5     5 sp05  4
#>  6     6 sp06  1
#>  7     7 sp07  2
#>  8     8 sp08  0
#>  9     9 sp09  0
#> 10    10 sp10  3
#> 11    11 sp11  1
#> 12    12 sp12  2
#> 13    13 sp13  3
#> 14    14 sp14  0
#> 15    15 sp15  2
#> 16    16 sp16  5
#> 17    17 sp17  4
#> 18    18 sp18  2
#> 19    19 sp19  5
#> 20    20 sp20  3
#> 21    21 sp21  3
#> 22    22 sp22  5
#> 23    23 sp23  4
#> 24    24 sp24  4
#> 25    25 sp25  3
#> 26    26 sp26  4
#> 27    27 sp27  3
#> 28    28 sp28  4
#> 29    29 sp29  0
#> 30    30 sp30  0
#> 31    31 sp31  0
#> 32    32 sp32  5
#> 33    33 sp33  5
#> 34    34 sp34  2
```

```
#> 35 35 sp35 3
#> 36 36 sp36 5
#> 37 37 sp37 2
#> 38 38 sp38 4
#> 39 39 sp39 4
#> 40 40 sp40 2
#> 41 41 sp41 4
#> 42 42 sp42 3
#> 43 43 sp43 3
#> 44 44 sp44 4
#> 45 45 sp45 5
#> 46 46 sp46 0
#> 47 47 sp47 2
#> 48 48 sp48 1
#> 49 49 sp49 1
#> 50 50 sp50 0
```

4.10 Crie uma lista com os objetos criados anteriormente: **mega**, **tr**, **ma** e **df**.

Solução:

```
lis <- list(mega, tr, ma, df)
lis
#> [[1]]
#> [1] 25 53 9 22 13 20
#>
#> [[2]]
#> [1] cont cont cont cont cont cont cont cont cont cont cont cont
#> [12] cont cont cont cont cont cont cont cont cont cont cont cont
#> [23] cont cont cont cont cont cont cont cont cont cont cont cont
#> [34] cont cont cont cont cont cont cont cont cont cont cont cont
#> [45] cont cont cont cont cont cont cont trat trat trat trat trat
#> [56] trat trat trat trat trat trat trat trat trat trat trat trat
#> [67] trat trat trat trat trat trat trat trat trat trat trat trat
#> [78] trat trat trat trat trat trat trat trat trat trat trat trat
#> [89] trat trat trat trat trat trat trat trat trat trat trat trat
#> [100] trat
#> Levels: cont trat
#>
#> [[3]]
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
#> [1,]    8    0    2    7    5    5    1    0    2    8
#> [2,]    0   10    4    9    7    7    7    3    5    0
#> [3,]    3    3    1    0   10   10    0    9    3    5
#> [4,]    5    0    0    2    8   10    0    9    4    2
#> [5,]    7    3   10    4    6    9   10    4    1    0
```

#>	[6,]	2	10	6	9	6	6	3	3	4	6
#>	[7,]	9	7	0	5	3	2	3	2	9	7
#>	[8,]	5	7	8	7	10	7	0	10	0	4
#>	[9,]	2	10	8	2	4	5	1	9	4	10
#>	[10,]	10	7	0	2	3	5	5	6	5	7
#>	[11,]	3	10	5	8	2	3	0	9	7	2
#>	[12,]	9	9	1	5	9	7	7	2	3	4
#>	[13,]	10	2	7	10	8	9	4	8	0	2
#>	[14,]	0	9	4	9	9	7	10	6	2	6
#>	[15,]	5	3	9	9	7	6	0	3	3	6
#>	[16,]	7	0	5	10	7	4	7	6	9	9
#>	[17,]	4	8	1	7	7	10	0	6	2	5
#>	[18,]	6	10	5	2	3	5	10	10	5	10
#>	[19,]	3	2	6	4	10	1	6	4	1	1
#>	[20,]	3	10	5	8	0	4	9	4	6	2
#>	[21,]	0	9	7	1	6	5	4	3	0	10
#>	[22,]	2	1	6	0	7	2	6	7	7	7
#>	[23,]	3	9	2	0	4	3	4	1	4	5
#>	[24,]	5	7	6	9	4	5	2	2	5	2
#>	[25,]	5	4	8	5	9	10	4	3	1	8
#>	[26,]	7	2	3	6	4	3	2	5	10	1
#>	[27,]	7	3	1	7	7	2	2	8	7	1
#>	[28,]	10	7	10	1	8	4	1	6	9	3
#>	[29,]	3	7	5	4	7	1	1	8	9	2
#>	[30,]	0	2	5	6	6	10	9	6	5	0
#>	[31,]	6	0	8	8	3	8	8	1	3	1
#>	[32,]	4	9	1	9	9	6	7	8	8	3
#>	[33,]	7	9	4	7	5	0	0	6	1	10
#>	[34,]	6	4	1	0	0	3	8	2	1	0
#>	[35,]	4	5	5	10	2	9	0	9	1	6
#>	[36,]	7	8	5	10	8	3	4	2	9	7
#>	[37,]	3	9	3	10	5	2	3	10	6	10
#>	[38,]	8	9	0	10	2	5	2	2	8	2
#>	[39,]	5	9	8	3	4	8	6	4	9	8
#>	[40,]	3	2	8	0	7	6	5	5	1	0
#>	[41,]	3	8	9	6	6	8	7	2	5	0
#>	[42,]	5	1	6	7	8	7	7	6	8	1
#>	[43,]	7	1	5	9	0	2	7	6	0	3
#>	[44,]	1	3	9	10	2	9	10	3	6	8
#>	[45,]	7	1	4	5	7	10	5	5	1	2
#>	[46,]	0	7	0	10	5	2	4	1	3	9
#>	[47,]	9	3	0	6	7	5	4	0	0	0
#>	[48,]	8	8	4	2	9	9	4	8	1	7
#>	[49,]	9	2	4	4	7	5	2	5	7	3
#>	[50,]	3	0	8	1	3	3	3	1	4	0

#> [51,]	9	6	8	10	9	9	8	4	8	2
#> [52,]	4	9	10	9	7	1	8	7	5	10
#> [53,]	4	10	6	0	7	1	4	10	9	8
#> [54,]	7	4	0	6	7	0	0	0	0	7
#> [55,]	5	8	8	1	3	9	10	8	0	4
#> [56,]	3	9	1	4	4	9	10	8	5	9
#> [57,]	5	10	3	9	4	0	4	10	1	4
#> [58,]	2	0	3	1	3	6	3	0	3	6
#> [59,]	10	4	3	3	1	7	3	1	5	8
#> [60,]	9	4	4	9	2	4	4	10	3	9
#> [61,]	6	0	3	4	0	6	5	8	6	4
#> [62,]	8	1	2	0	0	0	5	7	0	9
#> [63,]	8	10	1	3	3	5	9	0	5	3
#> [64,]	10	9	0	5	5	6	7	4	5	2
#> [65,]	7	7	9	2	5	7	3	0	4	2
#> [66,]	5	2	7	6	7	3	4	2	6	7
#> [67,]	8	8	7	0	0	5	9	10	3	9
#> [68,]	9	2	0	9	3	3	5	10	7	0
#> [69,]	9	8	4	3	4	10	8	7	2	3
#> [70,]	3	0	7	4	6	1	7	7	7	4
#> [71,]	0	0	4	9	7	3	5	6	7	4
#> [72,]	9	7	8	4	9	8	10	10	10	0
#> [73,]	8	3	6	10	4	0	2	1	5	4
#> [74,]	0	10	6	7	9	0	4	6	8	2
#> [75,]	10	10	6	3	3	6	7	1	6	4
#> [76,]	9	6	9	0	1	7	3	0	9	3
#> [77,]	3	9	6	0	1	5	5	4	5	8
#> [78,]	6	6	9	6	0	7	6	7	10	4
#> [79,]	8	7	2	8	2	6	1	9	0	2
#> [80,]	6	5	2	1	8	4	1	10	1	5
#> [81,]	6	1	3	10	4	5	6	9	4	9
#> [82,]	3	4	3	6	0	0	5	4	0	3
#> [83,]	6	5	8	4	1	6	3	9	3	5
#> [84,]	8	5	1	0	9	9	1	7	10	8
#> [85,]	4	0	6	2	9	10	2	7	3	9
#> [86,]	2	0	1	7	9	2	2	6	6	3
#> [87,]	6	2	1	9	8	6	2	0	4	5
#> [88,]	7	2	1	10	9	7	1	3	4	1
#> [89,]	5	1	0	1	5	1	10	9	6	7
#> [90,]	10	0	5	2	7	6	1	0	5	1
#> [91,]	6	4	7	2	10	4	8	10	6	5
#> [92,]	0	10	8	5	4	3	8	10	5	9
#> [93,]	5	3	9	6	9	0	6	1	0	9
#> [94,]	10	0	5	9	3	10	6	4	7	1
#> [95,]	6	6	1	1	7	4	3	1	4	3

```

#> [96,] 7 4 1 9 8 10 1 1 10 1
#> [97,] 9 1 0 4 6 10 2 2 6 2
#> [98,] 2 4 0 5 6 4 5 6 5 2
#> [99,] 5 1 4 10 6 8 3 5 4 0
#> [100,] 7 1 3 2 10 10 9 8 2 3
#>
#> [[4]]
#> id sp ab
#> 1 1 sp01 0
#> 2 2 sp02 3
#> 3 3 sp03 3
#> 4 4 sp04 5
#> 5 5 sp05 4
#> 6 6 sp06 1
#> 7 7 sp07 2
#> 8 8 sp08 0
#> 9 9 sp09 0
#> 10 10 sp10 3
#> 11 11 sp11 1
#> 12 12 sp12 2
#> 13 13 sp13 3
#> 14 14 sp14 0
#> 15 15 sp15 2
#> 16 16 sp16 5
#> 17 17 sp17 4
#> 18 18 sp18 2
#> 19 19 sp19 5
#> 20 20 sp20 3
#> 21 21 sp21 3
#> 22 22 sp22 5
#> 23 23 sp23 4
#> 24 24 sp24 4
#> 25 25 sp25 3
#> 26 26 sp26 4
#> 27 27 sp27 3
#> 28 28 sp28 4
#> 29 29 sp29 0
#> 30 30 sp30 0
#> 31 31 sp31 0
#> 32 32 sp32 5
#> 33 33 sp33 5
#> 34 34 sp34 2
#> 35 35 sp35 3
#> 36 36 sp36 5
#> 37 37 sp37 2

```

```
#> 38 38 sp38 4
#> 39 39 sp39 4
#> 40 40 sp40 2
#> 41 41 sp41 4
#> 42 42 sp42 3
#> 43 43 sp43 3
#> 44 44 sp44 4
#> 45 45 sp45 5
#> 46 46 sp46 0
#> 47 47 sp47 2
#> 48 48 sp48 1
#> 49 49 sp49 1
#> 50 50 sp50 0
```

4.11 Selecione os elementos ímpares do objeto **tr** e atribua ao objeto **tr_impar**.

Solução:

```
tr_impar <- tr[seq(1, 99, 2)]
tr_impar
#> [1] cont cont cont cont cont cont cont cont cont cont cont
#> [12] cont cont cont cont cont cont cont cont cont cont cont
#> [23] cont cont cont trat trat trat trat trat trat trat trat
#> [34] trat trat trat trat trat trat trat trat trat trat trat
#> [45] trat trat trat trat trat trat
#> Levels: cont trat
```

4.12 Selecione as linhas com ids pares do objeto **df** e atribua ao objeto **df_ids_par**.

Solução:

```
df_ids_par <- df[seq(2, 100, 2), ]
df_ids_par
#>      id  sp ab
#> 2      2 sp02 3
#> 4      4 sp04 5
#> 6      6 sp06 1
#> 8      8 sp08 0
#> 10     10 sp10 3
#> 12     12 sp12 2
#> 14     14 sp14 0
#> 16     16 sp16 5
#> 18     18 sp18 2
#> 20     20 sp20 3
#> 22     22 sp22 5
#> 24     24 sp24 4
#> 26     26 sp26 4
```



```
#> 28      28 sp28  4
#> 30      30 sp30  0
#> 32      32 sp32  5
#> 34      34 sp34  2
#> 36      36 sp36  5
#> 38      38 sp38  4
#> 40      40 sp40  2
#> 42      42 sp42  3
#> 44      44 sp44  4
#> 46      46 sp46  0
#> 48      48 sp48  1
#> 50      50 sp50  0
#> NA      NA <NA> NA
#> NA.1    NA <NA> NA
#> NA.2    NA <NA> NA
#> NA.3    NA <NA> NA
#> NA.4    NA <NA> NA
#> NA.5    NA <NA> NA
#> NA.6    NA <NA> NA
#> NA.7    NA <NA> NA
#> NA.8    NA <NA> NA
#> NA.9    NA <NA> NA
#> NA.10   NA <NA> NA
#> NA.11   NA <NA> NA
#> NA.12   NA <NA> NA
#> NA.13   NA <NA> NA
#> NA.14   NA <NA> NA
#> NA.15   NA <NA> NA
#> NA.16   NA <NA> NA
#> NA.17   NA <NA> NA
#> NA.18   NA <NA> NA
#> NA.19   NA <NA> NA
#> NA.20   NA <NA> NA
#> NA.21   NA <NA> NA
#> NA.22   NA <NA> NA
#> NA.23   NA <NA> NA
#> NA.24   NA <NA> NA
```

4.13 Faça uma amostragem de 10 linhas do objeto **df** e atribua ao objeto **df_amos10**.

Solução:

```
df_amos10 <- df[sample(nrow(df), 10), ]
df_amos10
#>      id  sp ab
```

```
#> 37 37 sp37 2
#> 36 36 sp36 5
#> 25 25 sp25 3
#> 42 42 sp42 3
#> 30 30 sp30 0
#> 6 6 sp06 1
#> 44 44 sp44 4
#> 33 33 sp33 5
#> 32 32 sp32 5
#> 17 17 sp17 4
```

4.14 Amostre 10 linhas do objeto **ma**, mas utilizando as linhas amostradas do **df_amos10** e atribua ao objeto **ma_amos10**.

Solução:

```
ma_amos10 <- ma[df_amos10$id, ]
ma_amos10
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
#> [1,]    3    9    3   10    5    2    3   10    6   10
#> [2,]    7    8    5   10    8    3    4    2    9    7
#> [3,]    5    4    8    5    9   10    4    3    1    8
#> [4,]    5    1    6    7    8    7    7    6    8    1
#> [5,]    0    2    5    6    6   10    9    6    5    0
#> [6,]    2   10    6    9    6    6    3    3    4    6
#> [7,]    1    3    9   10    2    9   10    3    6    8
#> [8,]    7    9    4    7    5    0    0    6    1   10
#> [9,]    4    9    1    9    9    6    7    8    8    3
#> [10,]   4    8    1    7    7   10    0    6    2    5
```

4.15 Una as colunas dos objetos **df_amos10** e **ma_amos10** e atribua ao objeto **dados_amos10**.

Solução:

```
dados_amos10 <- cbind(df_amos10, ma_amos10)
dados_amos10
#>      id  sp ab 1  2 3  4 5  6  7  8 9 10
#> 37 37 sp37 2 3  9 3 10 5  2  3 10 6 10
#> 36 36 sp36 5 7  8 5 10 8  3  4  2 9  7
#> 25 25 sp25 3 5  4 8  5 9 10  4  3 1  8
#> 42 42 sp42 3 5  1 6  7 8  7  7  6 8  1
#> 30 30 sp30 0 0  2 5  6 6 10  9  6 5  0
#> 6 6 sp06 1 2 10 6  9 6  6  3  3 4  6
#> 44 44 sp44 4 1  3 9 10 2  9 10  3 6  8
#> 33 33 sp33 5 7  9 4  7 5  0  0  6 1 10
#> 32 32 sp32 5 4  9 1  9 9  6  7  8 8  3
```

#> 17 17 sp17 4 4 8 1 7 7 10 0 6 2 5

Cap. 5 - Tidyverse

5.1 Reescreva as operações abaixo utilizando pipes %>%. - log10(cumsum(1:100))
- sum(sqrt(abs(rnorm(100)))) - sum(sort(sample(1:10, 10000, rep = TRUE)))

Solução:

```
library(tidyverse)
#> -- Attaching packages ----- tidyverse 1.3.1 --
#> v ggplot2 3.3.5      v purrr  0.3.4
#> v tibble  3.1.6      v dplyr  1.0.7
#> v tidyr   1.1.4      v stringr 1.4.0
#> v readr   2.1.1      v forcats 0.5.1
#> -- Conflicts ----- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()     masks stats::lag()

1:100 %>%
  cumsum() %>%
  log10()
#> [1] 0.0000000 0.4771213 0.7781513 1.0000000 1.1760913
#> [6] 1.3222193 1.4471580 1.5563025 1.6532125 1.7403627
#> [11] 1.8195439 1.8920946 1.9590414 2.0211893 2.0791812
#> [16] 2.1335389 2.1846914 2.2329961 2.2787536 2.3222193
#> [21] 2.3636120 2.4031205 2.4409091 2.4771213 2.5118834
#> [26] 2.5453071 2.5774918 2.6085260 2.6384893 2.6674530
#> [31] 2.6954817 2.7226339 2.7489629 2.7745170 2.7993405
#> [36] 2.8234742 2.8469553 2.8698182 2.8920946 2.9138139
#> [41] 2.9350032 2.9556878 2.9758911 2.9956352 3.0149403
#> [46] 3.0338257 3.0523091 3.0704073 3.0881361 3.1055102
#> [51] 3.1225435 3.1392492 3.1556396 3.1717265 3.1875207
#> [56] 3.2030329 3.2182729 3.2332500 3.2479733 3.2624511
#> [61] 3.2766915 3.2907022 3.3044905 3.3180633 3.3314273
#> [66] 3.3445887 3.3575537 3.3703280 3.3829171 3.3953264
#> [71] 3.4075608 3.4196254 3.4315246 3.4432630 3.4548449
```

```
#> [76] 3.4662743 3.4775553 3.4886917 3.4996871 3.5105450
#> [81] 3.5212689 3.5318619 3.5423274 3.5526682 3.5628874
#> [86] 3.5729877 3.5829719 3.5928427 3.6026025 3.6122539
#> [91] 3.6217992 3.6312408 3.6405808 3.6498215 3.6589648
#> [96] 3.6680130 3.6769678 3.6858313 3.6946052 3.7032914

rnorm(100) %>%
  abs() %>%
  sqrt() %>%
  sum()
#> [1] 82.79655

sample(1:10, 10000, rep = TRUE) %>%
  sort() %>%
  sum()
#> [1] 54789
```

5.2 Use a função `download.file()` e `unzip()` para baixar e extrair o arquivo do data paper de médios e grandes mamíferos: ATLANTIC MAMMALS. Em seguida, importe para o R, usando a função `readr::read_csv()`.

Solução:

```
library(tidyverse)
download.file(url = "https://esajournals.onlinelibrary.wiley.com/action/downloadSuppl...
             destfile = "ecy2785-sup-0001-DataS1.zip", mode = "wb")

unzip("ecy2785-sup-0001-DataS1.zip")

dp_lm <- readr::read_csv("ATLANTIC_MAMMAL_MID_LARGE_assemblages_and_sites.csv")
#> Warning: One or more parsing issues, see `problems()` for
#> details
#> Rows: 4680 Columns: 40
#> -- Column specification -----
#> Delimiter: ","
#> chr (27): ID, Country, State, Municipality, Study_locati...
#> dbl (11): Reference_paper_number, Publication_year, Year...
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

5.3 Use a função `tibble::glimpse()` para ter uma noção geral dos dados importados no item anterior.

Solução:

```

library(tidyverse)
dplyr::glimpse(dp_lm)
#> Rows: 4,680
#> Columns: 40
#> $ ID <chr> "AML01", "AML01", "AML01", ~
#> $ Reference_paper_number <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
#> $ Country <chr> "Brazil", "Brazil", "Brazil~
#> $ State <chr> "rio_grande_do_sul", "rio_g~
#> $ Municipality <chr> "Sinimbu", "Sinimbu", "Sini~
#> $ Study_location <chr> "Reserva Particular do Patr~
#> $ Latitude <dbl> -29.38333, -29.38333, -29.3~
#> $ Longitude <dbl> -52.53333, -52.53333, -52.5~
#> $ Precision <chr> "not_precise", "not_precise~
#> $ Size_ha <chr> "221", "221", "221", "221",~
#> $ Temperature <chr> "18", "18", "18", "18", "18~
#> $ Altitude <chr> "150-650", "150-650", "150~~
#> $ Annual_rainfall <chr> NA, NA, NA, NA, NA, NA, NA,~
#> $ Vegetation_type <chr> "Semideciduous forest", "Se~
#> $ Protect_area <chr> "yes", "yes", "yes", "yes",~
#> $ Matrix <chr> NA, NA, NA, NA, NA, NA, NA,~
#> $ Reference <chr> "Abreu-Junior, E.F. and Koh~
#> $ Publication_year <dbl> 2009, 2009, 2009, 2009, 200~
#> $ Type_of_publication <chr> "Article", "Article", "Arti~
#> $ Month_start <chr> "November", "November", "No~
#> $ Year_start <dbl> 2007, 2007, 2007, 2007, 200~
#> $ Month_finish <chr> "April", "April", "April", ~
#> $ Year_finish <dbl> 2009, 2009, 2009, 2009, 200~
#> $ Total_of_months <dbl> 6, 6, 6, 6, 6, 6, 6, 6, ~
#> $ Sampling_habitat <chr> "Interior", "Interior", "In~
#> $ Effort <dbl> 109.00, 109.00, 109.00, 109~
#> $ Effort_method <chr> "camera_days", "camera_days~
#> $ Method <chr> "mixed_method", "mixed_meth~
#> $ Order <chr> "Carnivora", "Rodentia", "C~
#> $ Genus_on_paper <chr> "Cerdocyon", "Cuniculus", "~
#> $ Species_name_on_paper <chr> "Cerdocyon thous", "Cunicul~
#> $ Actual_species_Name <chr> "Cerdocyon thous", "Cunicul~
#> $ Number_of_record <chr> NA, NA, NA, NA, NA, NA, NA,~
#> $ `Density(groups/km2)` <dbl> NA, NA, NA, NA, NA, NA, NA,~
#> $ `Density(ind/km2)` <chr> NA, NA, NA, NA, NA, NA, NA,~
#> $ `Density(ind/km10)` <dbl> NA, NA, NA, NA, NA, NA, NA,~
#> $ `Abundance(%)` <dbl> NA, NA, NA, NA, NA, NA, NA,~
#> $ Abundance_relative <dbl> NA, NA, NA, NA, NA, NA, NA,~
#> $ `Abundance(10/km)` <dbl> NA, NA, NA, NA, NA, NA, NA,~
#> $ Voucher_Specimens <chr> NA, NA, NA, NA, NA, NA, NA,~

```

5.4 Compare os dados de penguins (*palmerpenguins::penguins_raw* e *palmerpenguins::penguins*). Monte uma série de funções dos pacotes *tidyr* e *dplyr* para limpar os dados e fazer com que o primeiro dado seja igual ao segundo.

Solução:

```
library(tidyverse)
library(palmerpenguins)

penguins_raw
#> # A tibble: 344 x 17
#>   studyName `Sample Number` Species      Region Island Stage
#>   <chr>          <dbl> <chr>      <chr> <chr> <chr>
#> 1 PAL0708              1 Adelie Pe~ Anvers Torge~ Adult~
#> 2 PAL0708              2 Adelie Pe~ Anvers Torge~ Adult~
#> 3 PAL0708              3 Adelie Pe~ Anvers Torge~ Adult~
#> 4 PAL0708              4 Adelie Pe~ Anvers Torge~ Adult~
#> 5 PAL0708              5 Adelie Pe~ Anvers Torge~ Adult~
#> 6 PAL0708              6 Adelie Pe~ Anvers Torge~ Adult~
#> 7 PAL0708              7 Adelie Pe~ Anvers Torge~ Adult~
#> 8 PAL0708              8 Adelie Pe~ Anvers Torge~ Adult~
#> 9 PAL0708              9 Adelie Pe~ Anvers Torge~ Adult~
#> 10 PAL0708             10 Adelie Pe~ Anvers Torge~ Adult~
#> # ... with 334 more rows, and 11 more variables:
#> #   Individual ID <chr>, Clutch Completion <chr>,
#> #   Date Egg <date>, Culmen Length (mm) <dbl>,
#> #   Culmen Depth (mm) <dbl>, Flipper Length (mm) <dbl>,
#> #   Body Mass (g) <dbl>, Sex <chr>,
#> #   Delta 15 N (o/oo) <dbl>, Delta 13 C (o/oo) <dbl>,
#> #   Comments <chr>
penguins
#> # A tibble: 344 x 8
#>   species island  bill_length_mm bill_depth_mm
#>   <fct>   <fct>         <dbl>         <dbl>
#> 1 Adelie Torgersen    39.1           18.7
#> 2 Adelie Torgersen    39.5           17.4
#> 3 Adelie Torgersen    40.3            18
#> 4 Adelie Torgersen    NA              NA
#> 5 Adelie Torgersen    36.7           19.3
#> 6 Adelie Torgersen    39.3           20.6
#> 7 Adelie Torgersen    38.9           17.8
#> 8 Adelie Torgersen    39.2           19.6
#> 9 Adelie Torgersen    34.1           18.1
#> 10 Adelie Torgersen    42             20.2
#> # ... with 334 more rows, and 4 more variables:
#> #   flipper_length_mm <int>, body_mass_g <int>, sex <fct>,
```



```
#> #   year <int>

penguins_raw %>%
  dplyr::select(Species, Island, `Culmen Length (mm)`:Sex, `Date Egg`) %>%
  dplyr::rename(species = Species,
                island = Island,
                bill_length_mm = `Culmen Length (mm)`,
                bill_depth_mm = `Culmen Depth (mm)`,
                flipper_length_mm = `Flipper Length (mm)`,
                body_mass_g = `Body Mass (g)`,
                sex = Sex,
                year = `Date Egg`) %>%
  tidyr::separate(species, c("species", NA, NA, NA, NA)) %>%
  dplyr::mutate(sex = stringr::str_to_lower(sex),
                year = lubridate::year(year))

#> # A tibble: 344 x 8
#>   species island   bill_length_mm bill_depth_mm
#>   <chr>   <chr>         <dbl>         <dbl>
#> 1 Adelie Torgersen     39.1           18.7
#> 2 Adelie Torgersen     39.5           17.4
#> 3 Adelie Torgersen     40.3           18
#> 4 Adelie Torgersen     NA            NA
#> 5 Adelie Torgersen     36.7           19.3
#> 6 Adelie Torgersen     39.3           20.6
#> 7 Adelie Torgersen     38.9           17.8
#> 8 Adelie Torgersen     39.2           19.6
#> 9 Adelie Torgersen     34.1           18.1
#> 10 Adelie Torgersen     42            20.2
#> # ... with 334 more rows, and 4 more variables:
#> #   flipper_length_mm <dbl>, body_mass_g <dbl>, sex <chr>,
#> #   year <dbl>
```

5.5 Usando os dados de penguins (*palmerpenguins::penguins*), calcule a correlação de Pearson entre comprimento e profundidade do bico para cada espécie e para todas as espécies. Compare os índices de correlação para exemplificar o Paradoxo de Simpson.

Solução:

```
library(tidyverse)
library(palmerpenguins)

cor(penguins$bill_length_mm, penguins$bill_depth_mm, use = "na.or.complete")
#> [1] -0.2350529

penguins %>%
```

```
dplyr::group_split(species) %>%
  purrr::map(~cor(.x$bill_length_mm, .x$bill_depth_mm, use = "na.or.complete"))
#> [[1]]
#> [1] 0.3914917
#>
#> [[2]]
#> [1] 0.6535362
#>
#> [[3]]
#> [1] 0.6433839
```

5.6 Oficialmente a pandemia de COVID-19 começou no Brasil com o primeiro caso no dia 26 de fevereiro de 2020. Calcule quantos anos, meses e dias se passou desde então. Calcule também quanto tempo se passou até você ser vacinado.

Solução:

```
covid_inicio_br <- lubridate::dmy("26-02-2020")
vacina <- lubridate::dmy("20-07-2021")

intervalo_covid <- lubridate::interval(covid_inicio_br, lubridate::today())
intervalo_vacina <- lubridate::interval(covid_inicio_br, vacina)

lubridate::as.period(intervalo_covid)
#> [1] "1y 11m 16d 0H 0M 0S"
lubridate::as.period(intervalo_vacina)
#> [1] "1y 4m 24d 0H 0M 0S"
```

Cap. 6 - Visualização de dados

6.1 Utilizando o banco de dados `penguins` compare o comprimento do bico entre as diferentes espécies de penguins. Utilize um gráfico de caixa (boxplot) para ilustrar a variação intraespecífica e possíveis outliers nos dados. Para melhorar o seu gráfico, lembre-se de nomear os dois eixos corretamente, definir um tema e posicionar a legenda.

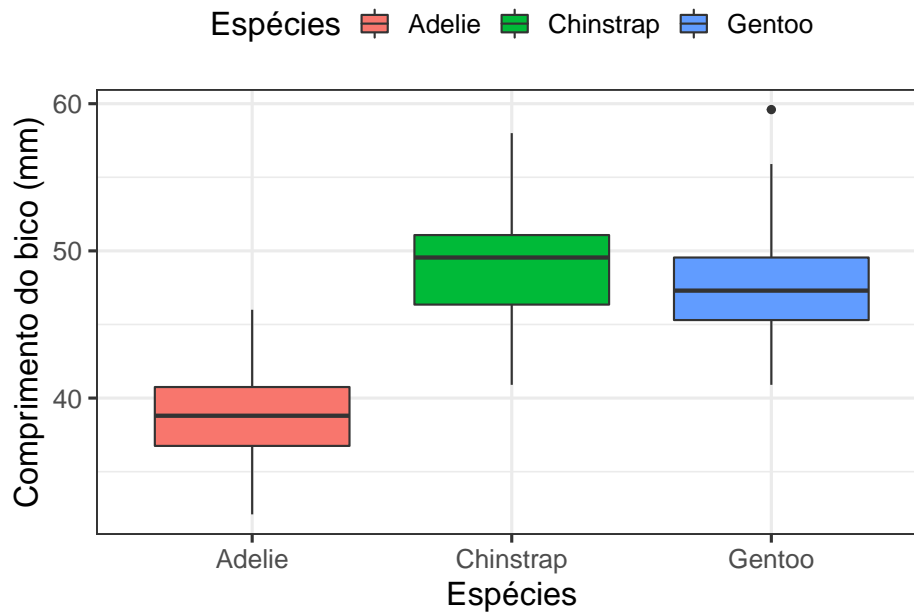
Solução:”

```
# Carregando pacotes necessários
library(tidyverse)
library(ecodados)

# Dados
penguins <- palmerpenguins::penguins

# Edição dos nomes das colunas para português
# names(penguins)
colnames(penguins) <- c("especies", "ilha", "comprimento_bico",
                        "profundidade_bico", "comprimento_nadadeira",
                        "massa_corporal", "sexo", "ano")

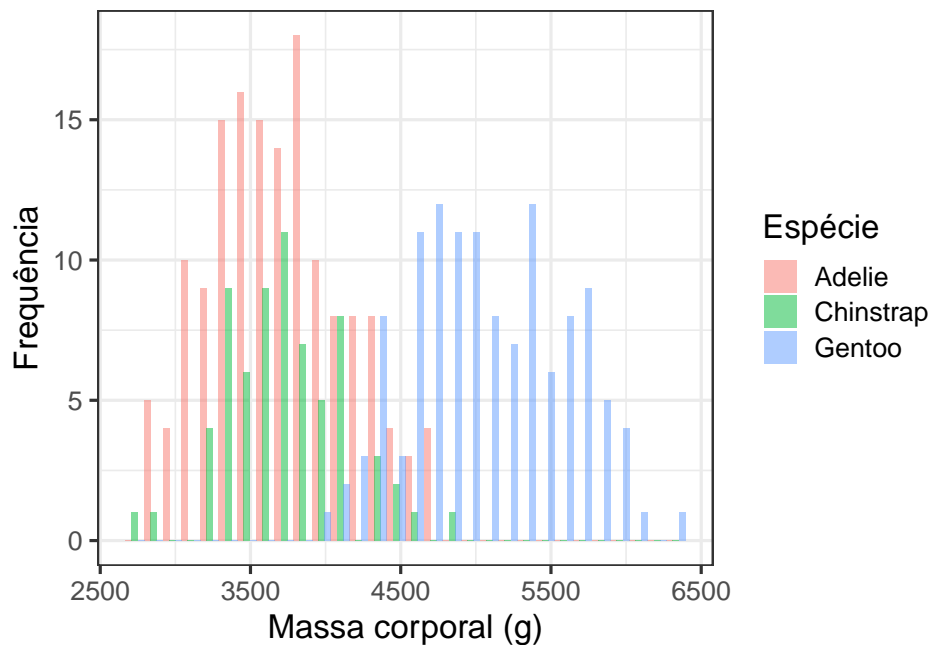
# Gráfico: Boxplot do tamanho do bico entre as diferentes espécies
ggplot(penguins, aes(y = comprimento_bico, x = especies, fill = especies)) +
  geom_boxplot() +
  theme_bw(base_size = 16) +
  theme(
    legend.position = "top"
  ) +
  labs(fill = "Espécies",
       x = "Espécies", y = "Comprimento do bico (mm)")
```



6.2 Utilizando o banco de dados `penguins` faça um histograma com a distribuição da massa corporal para cada uma das espécies. Utilize uma cor de preenchimento para cada espécie.

Solução:

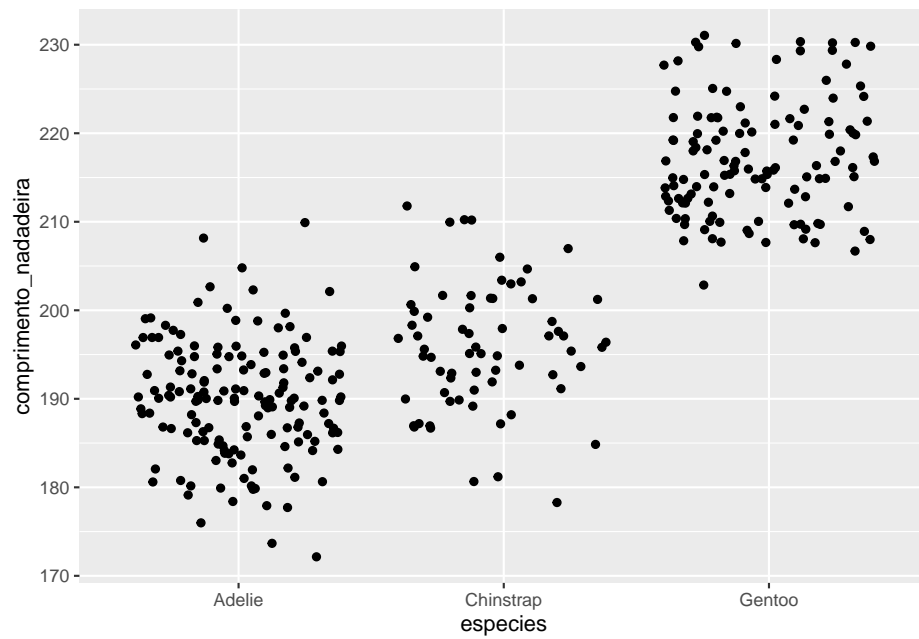
```
ggplot(penguins, aes(x = massa_corporal, fill = especie)) +
  geom_histogram(alpha = .5, position = position_dodge()) +
  theme_bw(base_size = 16) +
  labs(x = "Massa corporal (g)",
       y = "Frequência",
       fill = "Espécie")
#> `stat_bin()` using `bins = 30`. Pick better value with
#> `binwidth`.
#> Warning: Removed 2 rows containing non-finite values
#> (stat_bin).
```



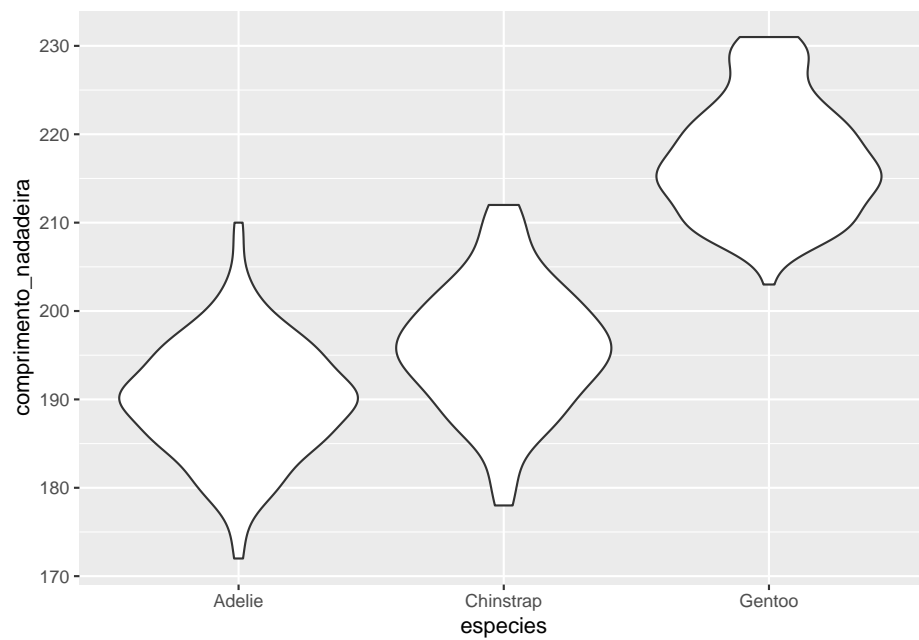
6.3 Utilizando o banco de dados `penguins`, faça três gráficos com o mesmo eixo Y e eixo X. Coloque o comprimento das nadadeiras no eixo Y e as espécies de pinguins no eixo X. No primeiro gráfico, utilize o `geom_jitter()` para plotar os dados brutos. No segundo gráfico, utilize o `geom_violin()` para mostrar a distribuição de densidade dos dados. No terceiro gráfico, utilize o `geom_boxplot()` para destacar a mediana e os quartis.

Solução:”

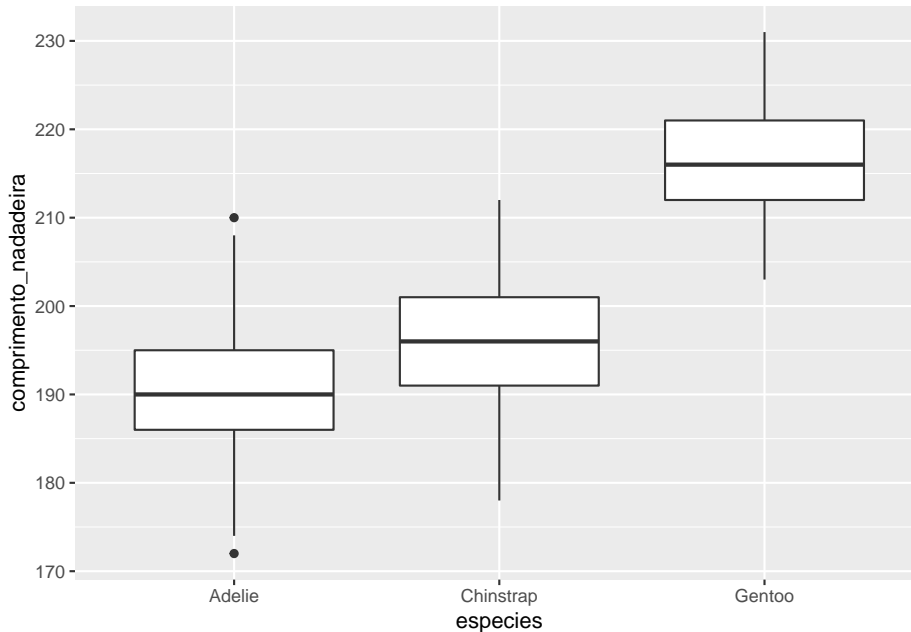
```
# Gráfico de jitter com dados brutos
pjitter <- ggplot(penguins, aes(y = comprimento_nadadeira, x = especies )) +
  geom_jitter()
pjitter
```



```
# Gráfico violin com a densidade dos dados  
pviolin <- ggplot(penguins, aes(y = comprimento_nadadeira, x = especies )) +  
  geom_violin()  
pviolin
```



```
# Gráfico de caixa com a média e os quartis
pcaixa <- ggplot(penguins, aes(y = comprimento_nadadeira, x = especies )) +
  geom_boxplot()
pcaixa
```

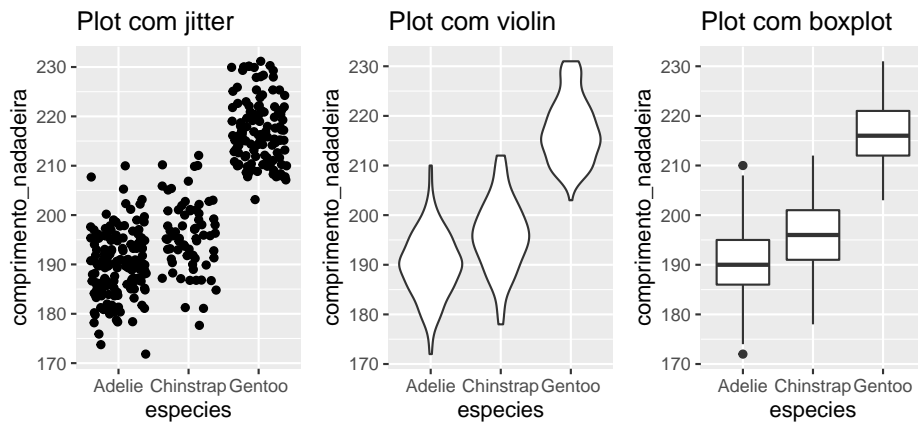


6.4 Se você conseguiu resolver o exercício 6.3, agora dê um passo a mais e compare os três gráficos lado a lado utilizando a função `grid.arrange()`. Lembre-se de colocar um título informativo em cada um dos gráficos antes de juntá-los em uma prancha única. Ao comparar os 3 tipos de gráficos, qual você considera mais informativo? Experimente combinar mais de um “geom” (camadas) e produzir gráficos ainda mais interessantes.

Solução:”

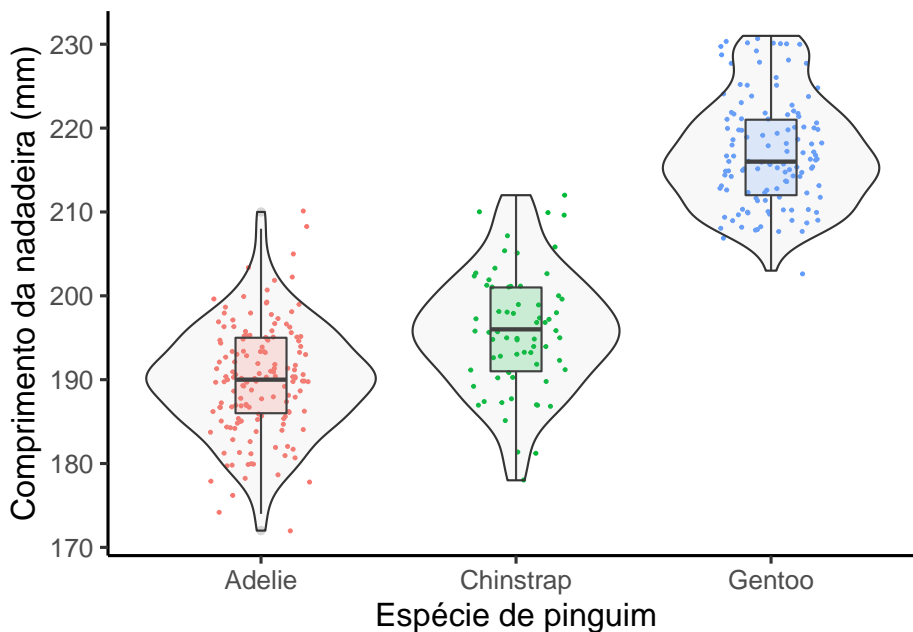
```
# colocando um título em cada gráfico
pjitter <- pjitter + labs(title = "Plot com jitter")
pviolin <- pviolin + labs(title = "Plot com violin")
pcaixa <- pcaixa + labs(title = "Plot com boxplot")

# juntando os 3 gráficos em um só
# Carregando o pacote gridExtra
library(gridExtra)
grid.arrange(pjitter, pviolin, pcaixa, ncol = 3)
```



Agora misturando as camadas.

```
# Misturando geoms
ggplot(penguins, aes(y = comprimento_nadadeira, x = especies)) +
  geom_jitter(size = .5, width = .2, aes(color = especies)) +
  geom_boxplot(aes(fill = especies), alpha = .2, width = .2) +
  geom_violin(fill = "gray", alpha = .1) +
  theme_classic(base_size = 16) +
  theme(legend.position = "none") +
  labs(y = "Comprimento da nadadeira (mm)",
       x = "Espécie de pinguim")
```



6.5 Utilize o banco de dados `ecodados::anova_dois_fatores` para construir

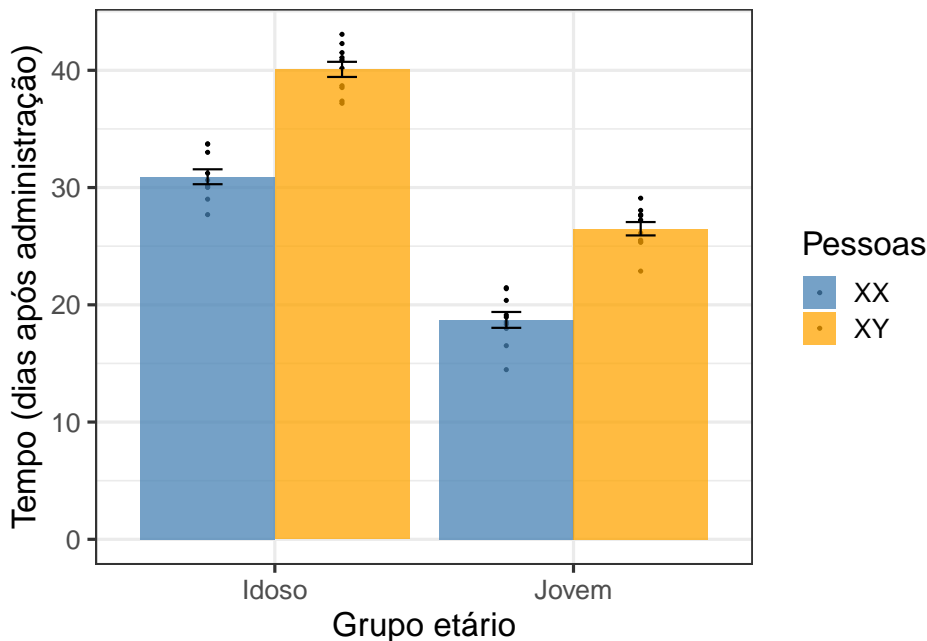
um gráfico de barras com a média e o erro padrão do Tempo (Tempo para eliminar a droga do corpo) no eixo Y em função da variável Pessoas (XX, ou XY) e Idade (jovem ou idoso). Antes de fazer o gráfico leia com atenção a descrição do mesmo através do comando `?ecodados::anova_dois_fatores`. Uma dica, utilize `fill` dentro do `aes` para representar um dos fatores (ex. Pessoas). O outro fator você pode representar no eixo X. Veja se consegue, se não conseguir pode olhar a cola com a solução para aprender como é feito. Outra dica, pesquise sobre a função `stat_summary()` ela irá te ajudar a calcular a média e o erro padrão dentro do comando que gera o gráfico.

Solução:"

```
# entenda o banco de dados primeiro
?ecodados::anova_dois_fatores
```

Versão 1: calculando a média e o erro padrão dentro do próprio gráfico.

```
ggplot(anova_dois_fatores, aes(y = Tempo, x = Idade, fill = Pessoas)) +
  geom_point(position = position_dodge(width = .9), size = .5) +
  stat_summary(fun = mean, geom = "bar", position = position_dodge(width = .9), alpha = .75) +
  stat_summary(fun.data = mean_se, geom = "errorbar", position = position_dodge(width = .9),
              width = .2) +
  scale_fill_manual(values = c("steelblue", "orange")) +
  theme_bw(base_size = 16) +
  labs(y = "Tempo (dias após administração)",
       x = "Grupo etário")
```

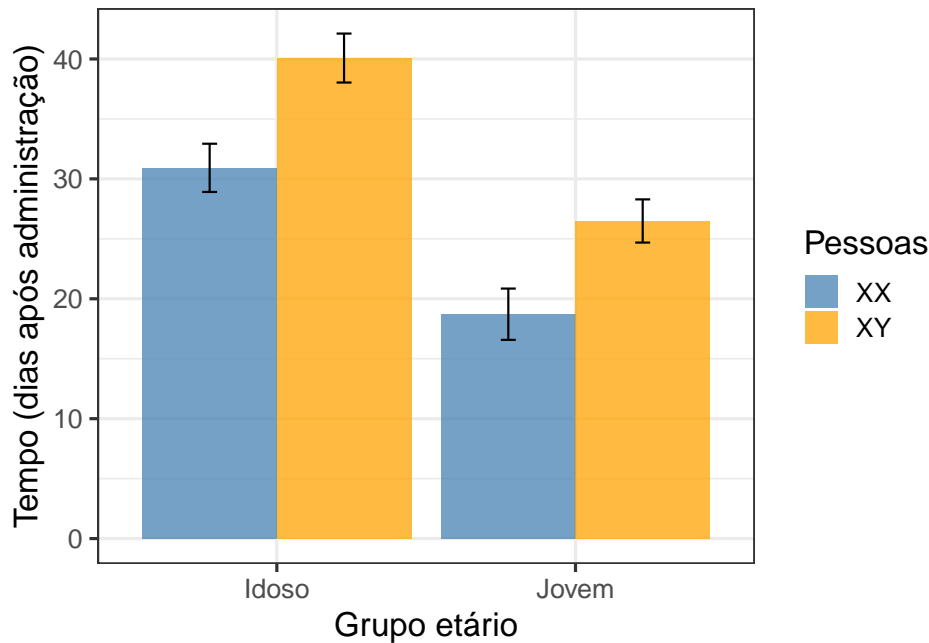


Versão 2: calculando a média e o erro padrão antes de produzir o gráfico.

```
# Calcular o desvio padrão
anova_media <- anova_dois_fatores %>%
  dplyr::group_by(Idade, Pessoas ) %>%
  dplyr::summarise(media = mean(Tempo, na.rm = TRUE),
                    desvio = sd(Tempo, na.rm = TRUE))
#> `summarise()` has grouped output by 'Idade'. You can override using the `.groups` a
#Veja como ficou
head(anova_media)
#> # A tibble: 4 x 4
#> # Groups:   Idade [2]
#>   Idade Pessoas media desvio
#>   <chr> <chr>   <dbl> <dbl>
#> 1 Idoso XX      30.9  2.01
#> 2 Idoso XY      40.1  2.04
#> 3 Jovem XX      18.7  2.14
#> 4 Jovem XY      26.5  1.80
```

Agora use esse banco de dados para plotar a média e o erro padrão.

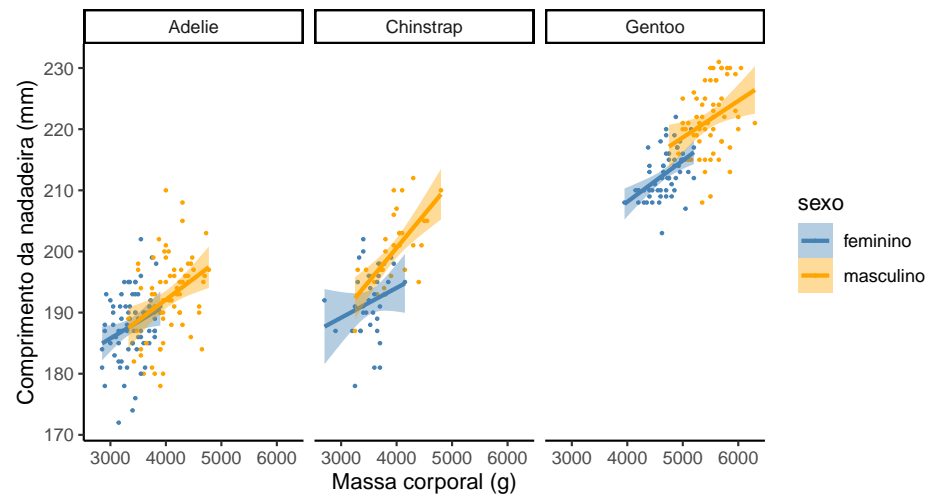
```
# Gráfico de barras com desvio padrão
ggplot(data = anova_media, aes(y = media, x = Idade, fill = Pessoas)) +
  geom_bar(stat = "identity", alpha = .75, position = position_dodge()) +
  geom_errorbar(aes(ymin = media - desvio, ymax = media + desvio),
                width = .1, position = position_dodge(width = .9)) +
  scale_fill_manual(values = c("steelblue", "orange")) +
  theme_bw(base_size = 16) +
  labs(y = "Tempo (dias após administração)",
       x = "Grupo etário")
```



6.6 Utilize o banco de dados `penguins` para criar um gráfico de dispersão entre o tamanho da nadadeira (eixo Y) e a massa corporal (eixo X). Utilize o argumento `fill` para ilustrar com cores as diferenças entre os sexos e utilize a função `facet_grid()` para criar um gráfico separado para cada espécie de pinguim. Se você não conhece essa função, dê uma olhada no `help ?facet_grid`. Você também pode utilizar a função `drop_na()` para excluir os dados faltantes da coluna `sexo`.

```
# primeiro vamos traduzir a coluna sexo para o português
penguins$sexo <- fct_recode(penguins$sexo, masculino = "male", feminino = "female")

ggplot(penguins %>% drop_na(sexo), aes(y = comprimento_nadadeira, x = massa_corporal,
                                       color = sexo, fill = sexo)) +
  geom_point(size = .4) +
  geom_smooth(method = lm) +
  facet_grid(~especies) +
  scale_color_manual(values = c("steelblue", "orange"), aesthetics = c("fill", "color")) +
  theme_classic() +
  labs(
    y = "Comprimento da nadadeira (mm)",
    x = "Massa corporal (g)"
  )
```



Cap. 7 - Modelos lineares

```
# Pacotes necessários
```

```
library(ggplot2)
```

```
library(ggpubr)
```

```
library(lmtest)
```

```
library(bbmle)
```

```
library(car)
```

```
library(ggforce)
```

```
library(sjPlot)
```

```
library(nlme)
```

```
library(dplyr)
```

7.1 Avalie se os indivíduos machos de uma espécie de aranha são maiores do que as fêmeas. Qual a sua interpretação sobre o dimorfismo sexual nesta espécie? Faça um gráfico boxplot usando também a função `geom_jitter`. Use os dados `Cap7_exercicio1` disponível no pacote `ecodados`.

Solução:

```
# Carregar dados do pacote ecodados
```

```
exercicio_1 <- ecodados::Cap7_exercicio1
```

```
# Verificar as premissas do teste
```

```
residuos_exercicio1 <- lm(Tamanho ~ Sexo, data = exercicio_1)
```

```
# Teste da normalidade dos resíduos
```

```
shapiro.test(residuals(residuos_exercicio1))
```

```
#>
```

```
#> Shapiro-Wilk normality test
```

```
#>
```

```
#> data: residuals(residuos_exercicio1)
```

```
#> W = 0.96586, p-value = 0.4328
```

```
# Teste da Homogeneidade da variância dos resíduos
```

```

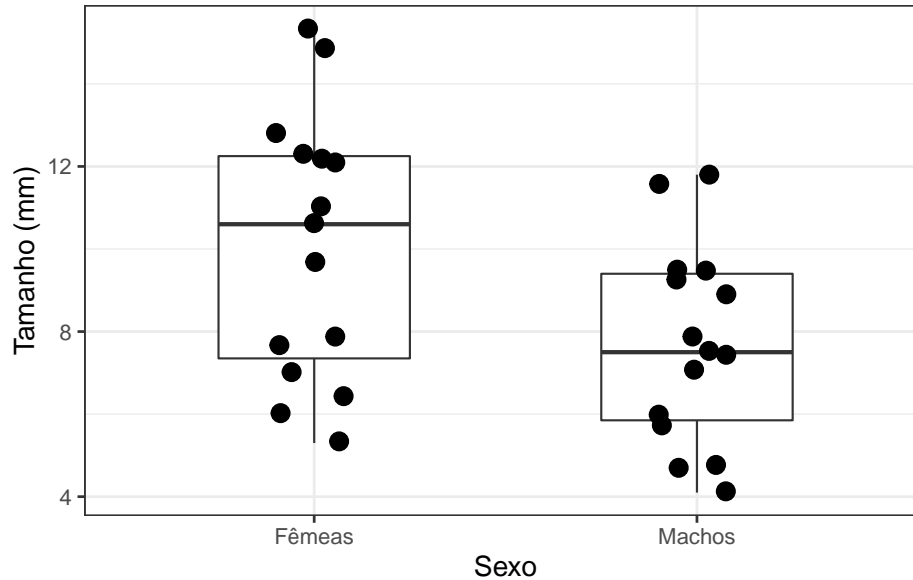
leveneTest(Tamanho ~ Sexo, data = exercicio_1)
#> Levene's Test for Homogeneity of Variance (center = median)
#>      Df F value Pr(>F)
#> group 1  2.0072 0.1676
#>      28

## Análise Teste T
t.test(Tamanho ~ Sexo, data = exercicio_1, var.equal = TRUE)
#>
#> Two Sample t-test
#>
#> data:  Tamanho by Sexo
#> t = 2.2756, df = 28, p-value = 0.03072
#> alternative hypothesis: true difference in means between group f and group m is not
#> 95 percent confidence interval:
#>  0.2356113 4.4843887
#> sample estimates:
#> mean in group f mean in group m
#>          10.08          7.72

## Gráfico
ggplot(data = exercicio_1, aes(x = Sexo, y = Tamanho)) +
  geom_boxplot(width = .5, show.legend = FALSE) +
  theme_bw(base_size = 14) +
  geom_jitter(size = 4, width = 0.1) +
  scale_x_discrete(labels=c("Fêmeas", "Machos")) +
  labs(title = "Dimorfismo sexual", x = "Sexo", y = "Tamanho (mm)")

```

Dimorfismo sexual



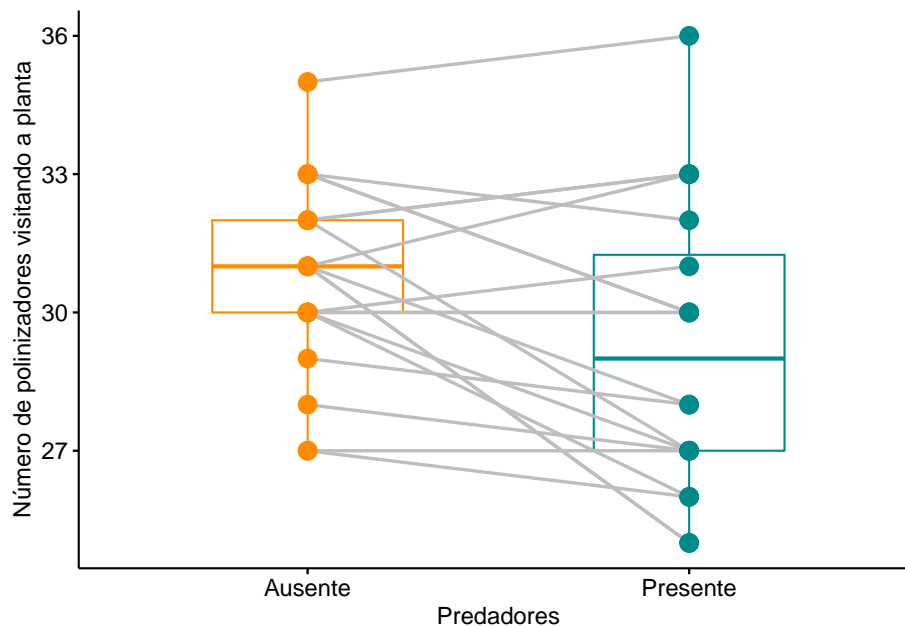
7.2 Avalie se o número de polinizadores visitando uma determinada espécie de planta é dependente da presença ou ausência de predadores. A mesma planta, em tempos diferentes, foi utilizada como unidade amostral para os tratamentos com e sem predadores. Qual a sua interpretação sobre os resultados? Faça um gráfico boxplot ligando os resultados da mesma planta com e sem a presença do predador. Use os dados `Cap7_exercicio2` disponível no pacote `ecodados`.

Solução:

```
# Carregar a planilha com os dados
exercicio_2 <- ecodados::Cap7_exercicio2

## Análise Teste T Pareado
t.test(Polinizadores ~ Predadores, paired = TRUE, data = exercicio_2)
#>
#> Paired t-test
#>
#> data: Polinizadores by Predadores
#> t = 2.843, df = 19, p-value = 0.0104
#> alternative hypothesis: true difference in means is not equal to 0
#> 95 percent confidence interval:
#>  0.4088952 2.6911048
#> sample estimates:
#> mean of the differences
#> 1.55
```

```
## Gráfico
ggpaired(exercicio_2, x = "Predadores", y = "Polinizadores",
  color = "Predadores", line.color = "gray", line.size = 0.8,
  palette = c("darkorange", "cyan4"), width = 0.5,
  point.size = 4, xlab = "Predadores",
  ylab = "Número de polinizadores visitando a planta",
  legend = "none")
```



7.3 Avalie se existe correlação entre o número de filhotes nos ninhos de uma espécie de ave com o tamanho do fragmento florestal. Qual a sua interpretação dos resultados? Faça um gráfico mostrando a relação entre as variáveis. Use os dados Cap7_exercicio3 disponível no pacote ecodados.

Solução:

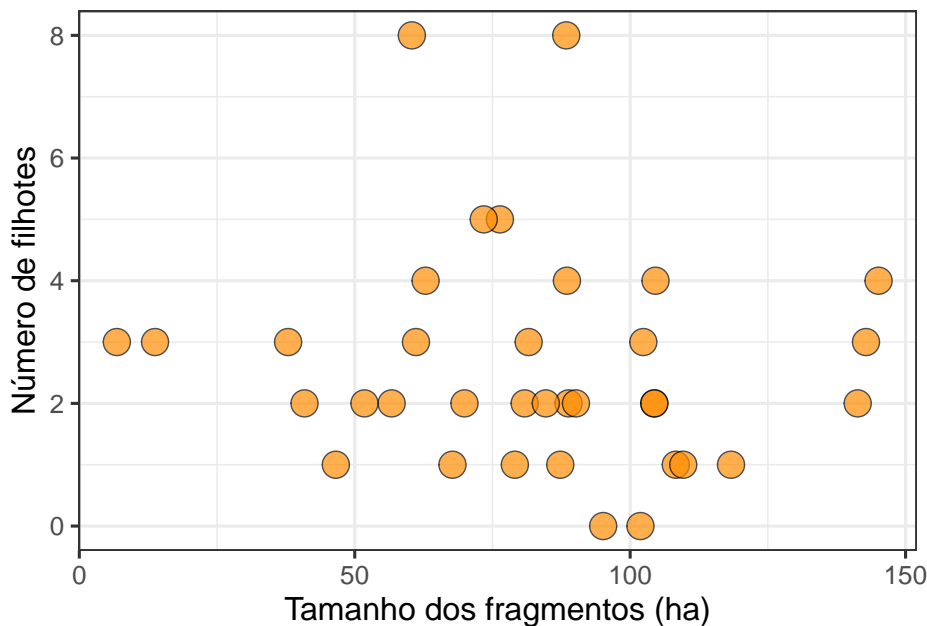
```
# Carregar a planilha com os dados
exercicio_3 <- ecodados::Cap7_exercicio3

## Análise correlação de Pearson
cor.test(~ Filhotes + Fragmentos, data = exercicio_3, method = "pearson")
#>
#> Pearson's product-moment correlation
#>
#> data: Filhotes and Fragmentos
#> t = -0.65396, df = 33, p-value = 0.5177
#> alternative hypothesis: true correlation is not equal to 0
```



```
#> 95 percent confidence interval:
#> -0.4301428 0.2287595
#> sample estimates:
#>      cor
#> -0.1131098

## Gráfico
ggplot(data = exercicio_3, aes(x = Fragmentos, y = Filhotes)) +
  labs(x = "Tamanho dos fragmentos (ha)", y = "Número de filhotes") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  theme_bw(base_size = 16)
```



7.4 Avalie se a relação entre o tamanho da área de diferentes ilhas e a riqueza de espécies de lagartos. Qual a sua interpretação dos resultados? Faça um gráfico mostrando a relação predita pelo modelo. Use os dados Cap7_exercicio4 disponível no pacote ecodados.

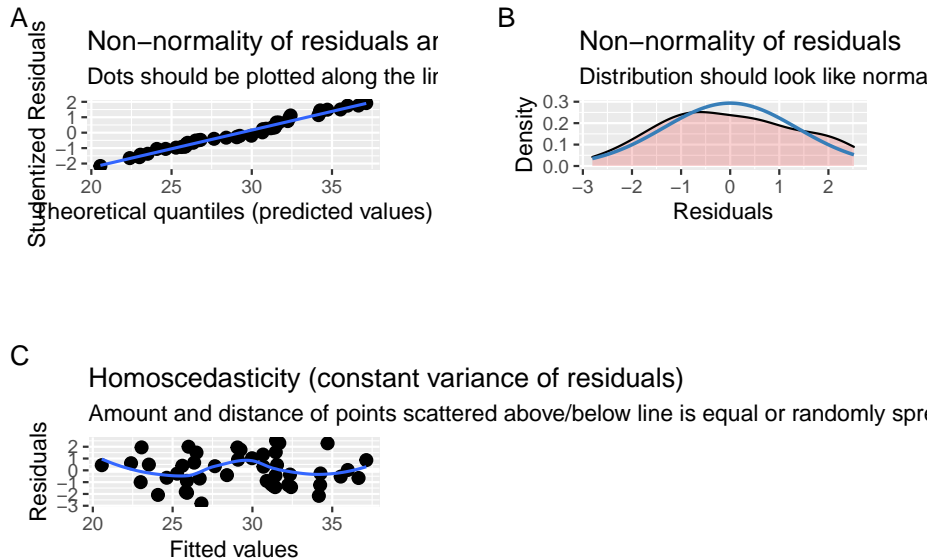
Solução:

```
# Carregar a planilha com os dados
exercicio_4 <- ecodados::Cap7_exercicio4

## Análise Regressão Simples
modelo_regressao <- lm(Riqueza ~ Area_ilhas, data = exercicio_4)
```

```
## Análise das premissas
```

```
plot_grid(plot_model(modelo_regressao , type = "diag"))
```



```
## Olhar os resultados
```

```
summary(modelo_regressao)
```

```
#>
```

```
#> Call:
```

```
#> lm(formula = Riqueza ~ Area_ilhas, data = exercicio_4)
```

```
#>
```

```
#> Residuals:
```

```
#>      Min       1Q   Median       3Q      Max
```

```
#> -2.8077 -0.9996 -0.2656  0.9026  2.5195
```

```
#>
```

```
#> Coefficients:
```

```
#>              Estimate Std. Error t value Pr(>|t|)
```

```
#> (Intercept)   3.8398      1.2816   2.996  0.00453 **
```

```
#> Area_ilhas    0.2879      0.0143  20.133 < 2e-16 ***
```

```
#> ---
```

```
#> Signif. codes:
```

```
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#>
```

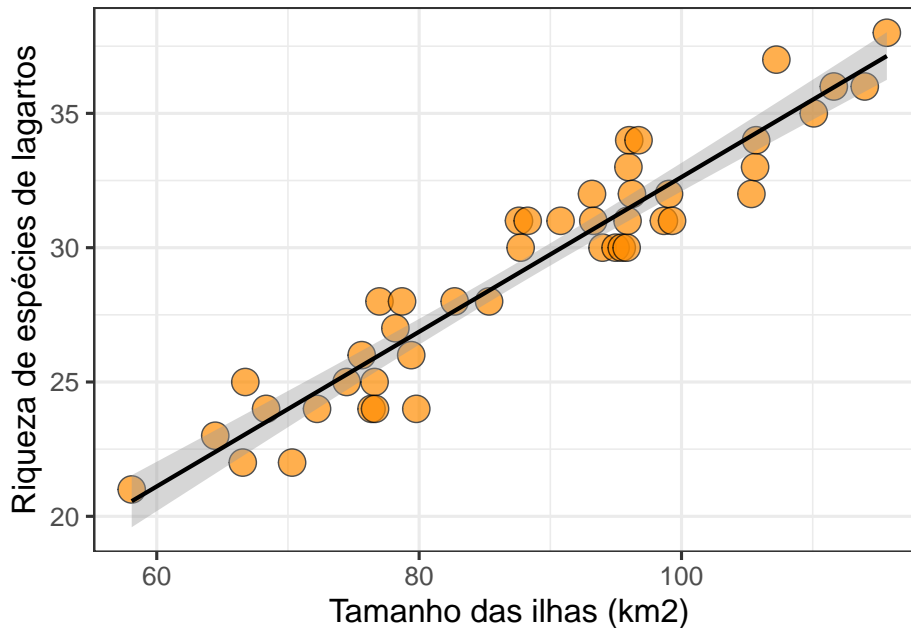
```
#> Residual standard error: 1.375 on 43 degrees of freedom
```

```
#> Multiple R-squared:  0.9041, Adjusted R-squared:  0.9019
```

```
#> F-statistic: 405.3 on 1 and 43 DF, p-value: < 2.2e-16
```

```
## Gráfico
```

```
ggplot(data = exercicio_4, aes(x = Area_ilhas, y = Riqueza)) +
  labs(x = "Tamanho das ilhas (km2)", y = "Riqueza de espécies de lagartos") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```



7.5 Avalie se existe relação entre a abundância de uma espécie de roedor com o tamanho da área dos fragmentos florestais e/ou a altitude. Faça uma regressão múltipla. Em seguida, crie diferentes modelos e selecione o mais parcimonioso com base no valores do teste de Likelihood-ratio test (LRT) e AIC. Qual a sua interpretação? Use os dados Cap7_exercicio5 disponível no pacote ecodados.

Solução:

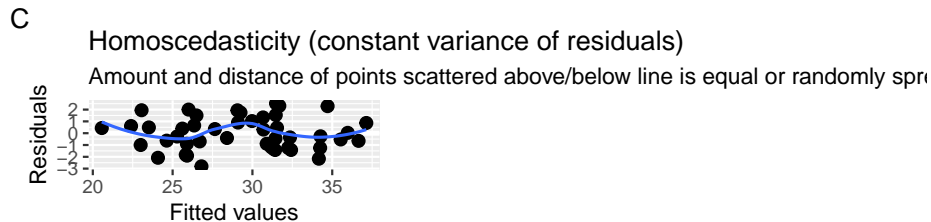
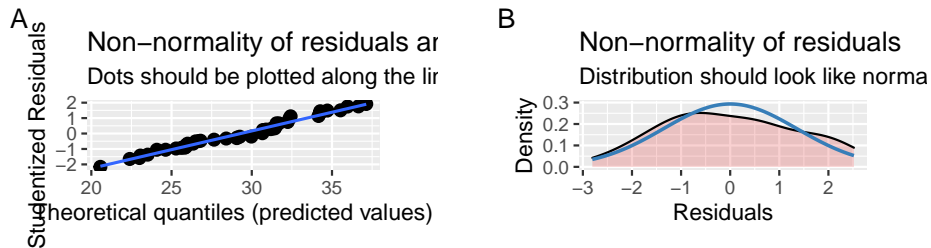
```
# Carregar a planilha com os dados
exercicio_5 <- ecodados::Cap7_exercicio5

## Análise Regressão Múltipla
modelo_regressao_mult <- lm(Abundancia ~ Area_fragmento*Altitude, data = exercicio_5)

## Multicolinearidade
vif(modelo_regressao_mult)
#>           Area_fragmento           Altitude
#>           5.403688           1.786589
#> Area_fragmento:Altitude
```

```
#> 6.890149

## Análise das premissas
plot_grid(plot_model(modelo_regressao , type = "diag"))
```



```
## Olhar os resultados
summary(modelo_regressao_mult)
#>
#> Call:
#> lm(formula = Abundancia ~ Area_fragmento * Altitude, data = exercicio_5)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -10.4180  -1.3524  -0.2711   1.1783  12.5283
#>
#> Coefficients:
#>              Estimate Std. Error t value
#> (Intercept)    5.747e+01  8.549e-01  67.230
#> Area_fragmento -3.337e-03  3.245e-03  -1.028
#> Altitude      -5.451e-03  4.859e-04 -11.218
#> Area_fragmento:Altitude 1.344e-06  1.471e-06  0.914
#>              Pr(>|t|)
#> (Intercept)    < 2e-16 ***
#> Area_fragmento    0.308
#> Altitude        1.69e-15 ***
#> Area_fragmento:Altitude 0.365
```

```

#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.99 on 52 degrees of freedom
#> Multiple R-squared:  0.7986, Adjusted R-squared:  0.787
#> F-statistic: 68.73 on 3 and 52 DF,  p-value: < 2.2e-16

## Vamos retirar a interação
modelo_regressao_mult_sem_interacao <- lm(Abundancia ~ Area_fragmento + Altitude, data = exercicio_5)

## Likelihood-ratio test (LRT)
lrtest(modelo_regressao_mult, modelo_regressao_mult_sem_interacao)
#> Likelihood ratio test
#>
#> Model 1: Abundancia ~ Area_fragmento * Altitude
#> Model 2: Abundancia ~ Area_fragmento + Altitude
#>   #Df LogLik Df  Chisq Pr(>Chisq)
#> 1    5 -138.71
#> 2    4 -139.16 -1  0.8917    0.345

## Vamos verificar o modelo só com a altitude
modelo_regressao_mult_sem_fragmento <- lm(Abundancia ~ Altitude, data = exercicio_5)

lrtest(modelo_regressao_mult_sem_interacao, modelo_regressao_mult_sem_fragmento)
#> Likelihood ratio test
#>
#> Model 1: Abundancia ~ Area_fragmento + Altitude
#> Model 2: Abundancia ~ Altitude
#>   #Df LogLik Df  Chisq Pr(>Chisq)
#> 1    4 -139.16
#> 2    3 -139.28 -1  0.2364    0.6268

## Vamos verificar o modelo só com o intercepto
modelo_regressao_mult_nulo <- lm(Abundancia ~ 1, data = exercicio_5)

lrtest(modelo_regressao_mult_sem_fragmento, modelo_regressao_mult_nulo)
#> Likelihood ratio test
#>
#> Model 1: Abundancia ~ Altitude
#> Model 2: Abundancia ~ 1
#>   #Df LogLik Df  Chisq Pr(>Chisq)
#> 1    3 -139.28
#> 2    2 -183.58 -1 88.611 < 2.2e-16 ***

```

```

#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Outra alternativa é usar o AIC para seleção dos modelos

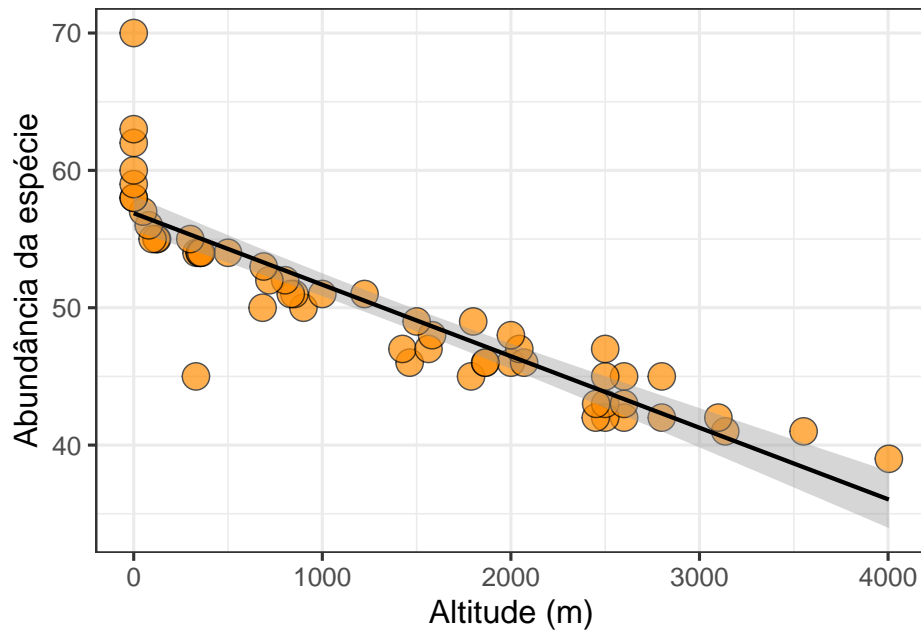
## modelo só com a área do fragmento
modelo_regressao_mult_sem_altitude <- lm(Abundancia ~ Area_fragmento, data = exercicio_5)

AICc <- ICTab(modelo_regressao_mult, modelo_regressao_mult_sem_interacao,
               modelo_regressao_mult_sem_fragmento, modelo_regressao_mult_nulo,
               modelo_regressao_mult_sem_altitude,
               type = c("AIC"), weights = TRUE,
               delta = TRUE, sort = TRUE)

AICc
#>
#> modelo_regressao_mult_sem_fragmento 0.0 3 0.61
#> modelo_regressao_mult_sem_interacao 1.8 4 0.25
#> modelo_regressao_mult 2.9 5 0.14
#> modelo_regressao_mult_sem_altitude 85.6 3 <0.001
#> modelo_regressao_mult_nulo 86.6 2 <0.001

## Gráfico
ggplot(data = exercicio_5, aes(x = Altitude, y = Abundancia)) +
  labs(x = "Altitude (m)", y = "Abundância da espécie") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)

```



7.6 Avalie se o local que machos territoriais ocupam (pasto, cana, floresta) influência no peso dos indivíduos. Qual a sua interpretação dos resultados? Faça um gráfico com os resultados. Use os dados Cap7_exercicio6 disponível no pacote ecodados.

Solução:

```
# Carregar a planilha com os dados
exercicio_6 <- ecodados::Cap7_exercicio6

## Análise anova um fator
modelo_aov <- aov(Peso ~ Local, data = exercicio_6)

## Normalidade
shapiro.test(residuals(modelo_aov))
#>
#> Shapiro-Wilk normality test
#>
#> data: residuals(modelo_aov)
#> W = 0.95361, p-value = 0.211

## Homogeneidade da variância
bartlett.test(Peso ~ Local, data = exercicio_6)
#>
#> Bartlett test of homogeneity of variances
#>
```

```

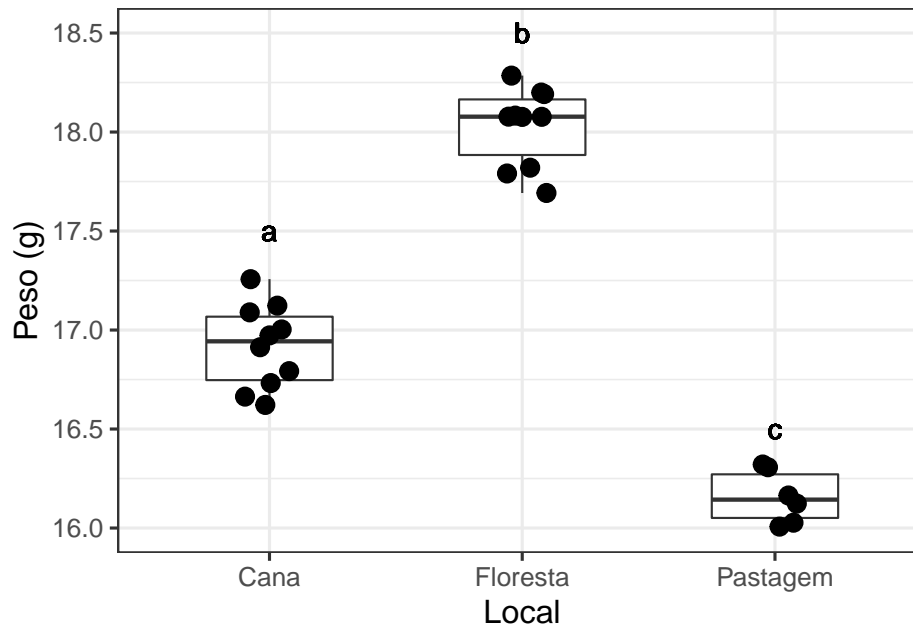
#> data:  Peso by Local
#> Bartlett's K-squared = 0.06087, df = 2, p-value =
#> 0.97

## Olhar os resultados
anova(modelo_aov)
#> Analysis of Variance Table
#>
#> Response: Peso
#>           Df Sum Sq Mean Sq F value    Pr(>F)
#> Local      2 20.0536 10.0268  248.47 < 2.2e-16 ***
#> Residuals 27  1.0896  0.0404
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Diferenças entre os tratamentos
TukeyHSD(modelo_aov)
#>  Tukey multiple comparisons of means
#>    95% family-wise confidence level
#>
#> Fit: aov(formula = Peso ~ Local, data = exercicio_6)
#>
#> $Local
#>           diff      lwr      upr p adj
#> Floresta-Cana  1.1125  0.8897525  1.3352475    0
#> Pastagem-Cana -0.8859 -1.1086475 -0.6631525    0
#> Pastagem-Floresta -1.9984 -2.2211475 -1.7756525    0

## Gráfico
ggplot(data = exercicio_6,
       aes(x = Local, y = Peso)) +
  geom_boxplot(width = .5, show.legend = FALSE) +
  geom_jitter(size = 4, width = 0.1) +
  geom_text(x = 1, y = 17.5, label = "a", color = "black", size = 5) +
  geom_text(x = 2, y = 18.5, label = "b", color = "black", size = 5) +
  geom_text(x = 3, y = 16.5, label = "c", color = "black", size = 5) +
  ylim(16, 18.5) +
  theme_bw(base_size = 16) +
  labs(x = "Local", y = "Peso (g)")

```

7.7 Avalie se a abundância de formigas está relacionada com o fato das domácias estarem abertas ou fechadas e com a idade das domácias. Verifique a interação entre os fatores. Qual a sua interpretação dos resultados? Faça um gráfico com os resultados. Use os dados Cap7_exercicio7 disponível no pacote ecodados.

Solução:

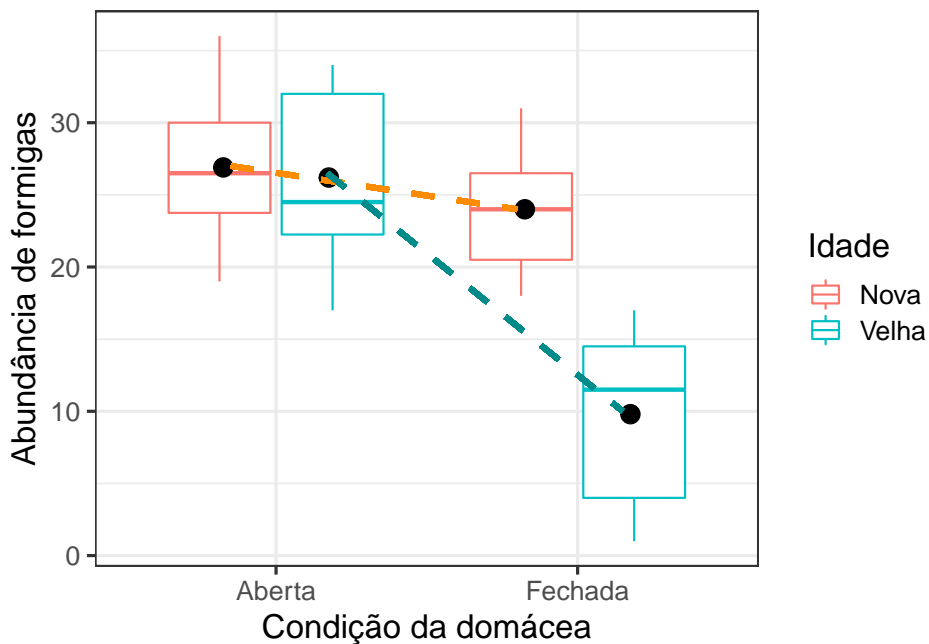
```
# Carregar a planilha com os dados
exercicio_7 <- ecodados::Cap7_exercicio7

## Análise anova dois fatores
modelo_aov_2 <- aov(Abundancia ~ Domacea * Idade, data = exercicio_7)

## Olhar os resultados
anova(modelo_aov_2)
#> Analysis of Variance Table
#>
#> Response: Abundancia
#>
#>      Df Sum Sq Mean Sq F value    Pr(>F)
#> Domacea      1  931.22   931.22  32.993 1.528e-06 ***
#> Idade         1  555.02   555.02  19.664 8.334e-05 ***
#> Domacea:Idade  1  455.63   455.63  16.143 0.0002862 ***
#> Residuals    36 1016.10    28.22
#> ---
#> Signif. codes:
```

```
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Gráfico
ggplot(data = exercicio_7,
       aes(y = Abundancia, x = Domacea, color = Idade)) +
  geom_boxplot() +
  stat_summary(fun = mean, geom = "point", aes(group = Idade, x = Domacea),
              color = "black", position = position_dodge(0.7), size = 4) +
  geom_link(aes(x = 0.85, y = 27, xend = 1.8, yend = 24), color = "darkorange",
            lwd = 1.3, linetype = 2) +
  geom_link(aes(x = 1.17, y = 26.5, xend = 2.15, yend = 10), color = "cyan4",
            lwd = 1.3, linetype = 2) +
  labs(x = "Condição da domácea",
       y = "Abundância de formigas") +
  theme_bw(base_size = 16)
```



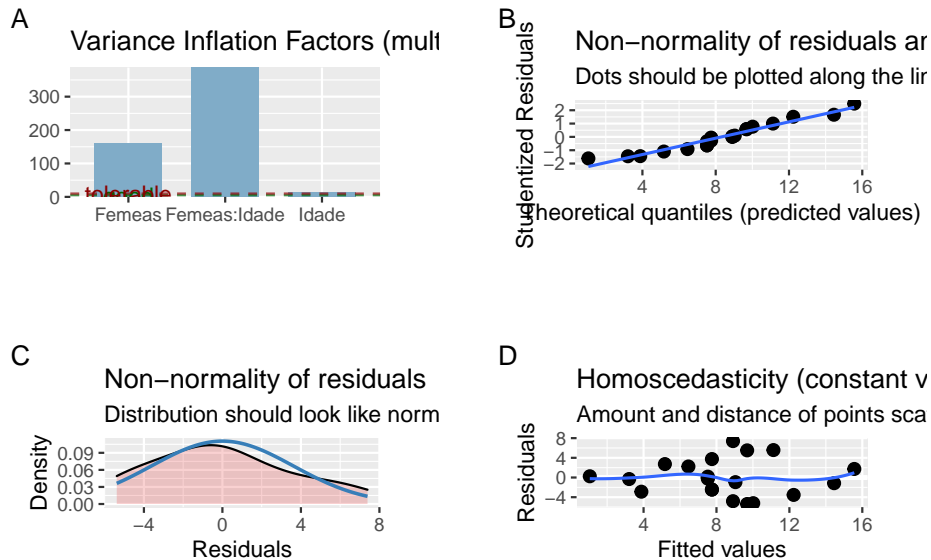
7.8 Avalie se o número de parasitas está relacionado com o tamanho corporal de fêmeas de uma espécie de ave. Além disso, use a idade das aves como uma co-variável explicando o número de parasitas. Qual a sua interpretação dos resultados? Faça um gráfico com os resultados. Use os dados Cap7_exercicio8 disponível no pacote ecodados.

Solução:

```
# Carregar a planilha com os dados
exercicio_8 <- ecodados::Cap7_exercicio8
```

```
## Análise ancova
modelo_ancova <- lm(Parasitas ~ Femeas * Idade, data = exercicio_8)

## Verificar as premissas
plot_grid(plot_model(modelo_ancova, type = "diag"))
```



```
## Olhar os resultados
anova(modelo_ancova)
#> Analysis of Variance Table
#>
#> Response: Parasitas
#>
#>      Df Sum Sq Mean Sq F value Pr(>F)
#> Femeas    2 111.577   55.789   3.2153 0.06885 .
#> Idade      1 113.527  113.527  6.5431 0.02185 *
#> Femeas:Idade 2  10.866    5.433   0.3131 0.73582
#> Residuals 15 260.261   17.351
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Criando modelo sem interação
modelo_ancova2 <- lm(Parasitas ~ Femeas + Idade, data = exercicio_8)

## Likelihood-ratio test
lrtest(modelo_ancova, modelo_ancova2)
#> Likelihood ratio test
```

```

#>
#> Model 1: Parasitas ~ Femeas * Idade
#> Model 2: Parasitas ~ Femeas + Idade
#>   #Df LogLik Df Chisq Pr(>Chisq)
#> 1    7 -56.228
#> 2    5 -56.657 -2 0.859      0.6508

anova(modelo_ancova2)
#> Analysis of Variance Table
#>
#> Response: Parasitas
#>           Df Sum Sq Mean Sq F value Pr(>F)
#> Femeas      2 111.58  55.789   3.4980 0.05341 .
#> Idade       1 113.53 113.527  7.1183 0.01622 *
#> Residuals 17 271.13  15.949
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Criando modelo sem tamanho do corpo
modelo_ancova3 <- lm(Parasitas ~ Idade, data = exercicio_8)

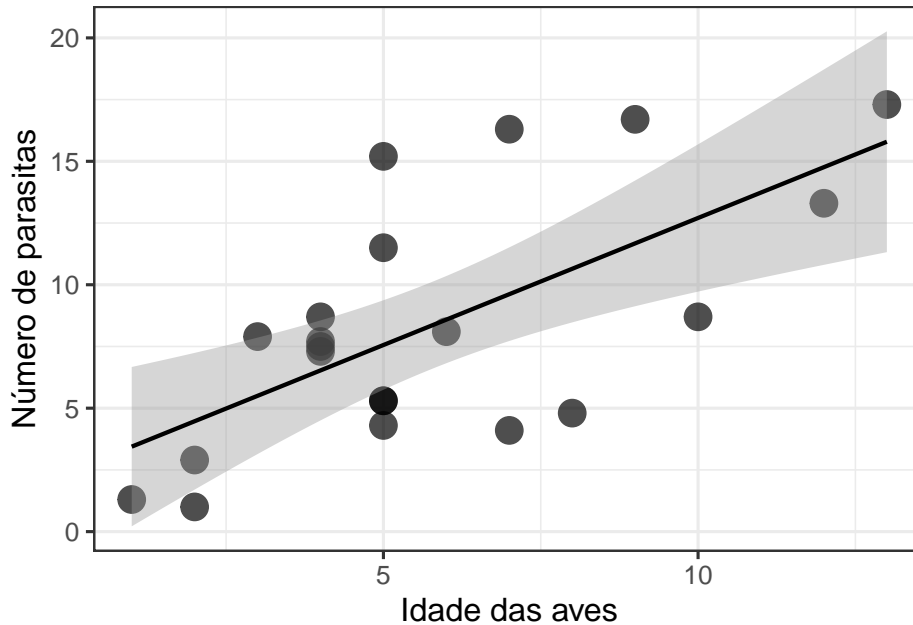
## Likelihood-ratio test
lrtest(modelo_ancova2, modelo_ancova3)
#> Likelihood ratio test
#>
#> Model 1: Parasitas ~ Femeas + Idade
#> Model 2: Parasitas ~ Idade
#>   #Df LogLik Df Chisq Pr(>Chisq)
#> 1    5 -56.657
#> 2    3 -57.087 -2 0.8584      0.651

anova(modelo_ancova3)
#> Analysis of Variance Table
#>
#> Response: Parasitas
#>           Df Sum Sq Mean Sq F value Pr(>F)
#> Idade      1 213.79 213.792  14.382 0.001231 **
#> Residuals 19 282.44  14.865
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Gráfico
ggplot(data = exercicio_8, aes(x = Idade, y = Parasitas)) +

```

```
labs(x = "Idade das aves", y = "Número de parasitas") +
geom_point(size = 6, shape = 19, alpha = 0.7) +
theme(legend.position = "none") +
geom_smooth(method = lm, se = TRUE, color = "black") +
theme_bw(base_size = 16)
```



7.9 Avalie se a presença ou ausência de predadores afeta a riqueza de macroinvertebrados em 10 lagos. Os tratamentos dos predadores foram realizados nos mesmos lagos. Qual a sua interpretação dos resultados? Faça um gráfico com os resultados. Use os dados Cap7_exercicio9 disponível no pacote ecodados.

Solução:

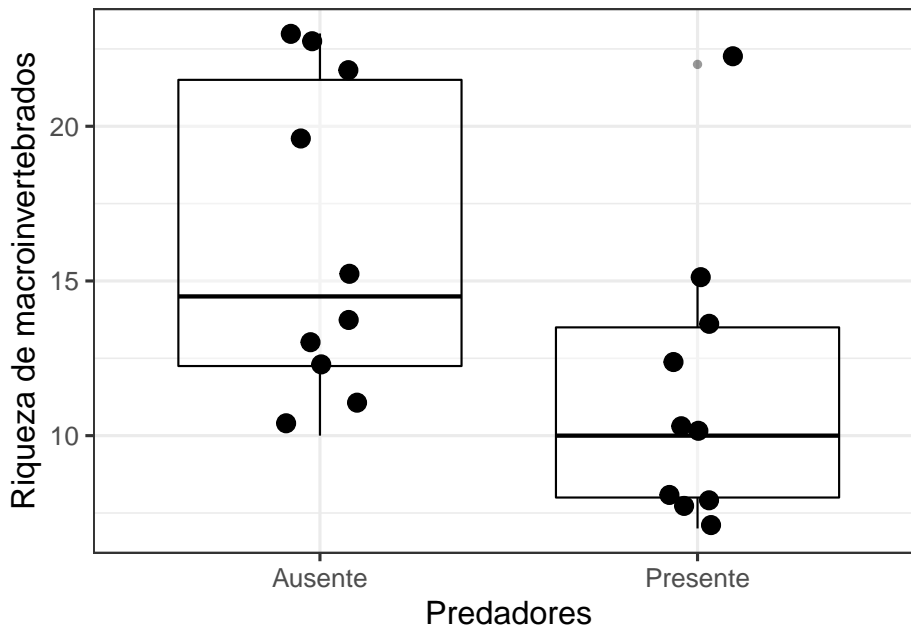
```
# Carregar a planilha com os dados
exercicio_9 <- ecodados::Cap7_exercicio9

## Análise anova em blocos
model_bloco <- aov(Riqueza ~ Predadores + Error(Lago), data = exercicio_9)

summary(model_bloco)
#>
#> Error: Lago
#>           Df Sum Sq Mean Sq F value Pr(>F)
#> Residuals  1 0.2561  0.2561
#>
#> Error: Within
```

```
#>           Df Sum Sq Mean Sq F value Pr(>F)
#> Predadores  1  120.0   120.05    4.743 0.0438 *
#> Residuals 17   430.2    25.31
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Gráfico
ggplot(data = exercicio_9, aes(x = Predadores, y = Riqueza)) +
  geom_boxplot(color = "black", show.legend = FALSE, alpha = 0.4) +
  geom_jitter(size = 4, width = 0.1) +
  labs(x = "Predadores", y = "Riqueza de macroinvertebrados") +
  theme_bw(base_size = 16)
```



7.10 Avalie se a precipitação anual afeta a riqueza de espécies de anuros em 44 localidades na Mata Atlântica. Use as coordenadas geográficas para controlar o efeito da autocorrelação espacial. Qual a sua interpretação dos resultados das análises com e sem levar em consideração a autocorrelação espacial? Use os dados anuros_ambientais disponível no pacote ecodados.

Solução:

```
# Carregar a planilha com os dados
exercicio_10 <- ecodados::anuros_ambientais

## Modelo gls sem estrutura espacial
```

```
no_spat_gls <- gls(Riqueza ~ Prec_anual, exercicio_10, method = "REML")

## Covariância esférica
espher_model <- gls(Riqueza ~ Prec_anual, exercicio_10,
  corSpher(form = ~Latitude + Longitude, nugget = TRUE))

## Covariância exponencial
expon_model <- gls(Riqueza ~ Prec_anual, exercicio_10,
  corExp(form = ~Latitude + Longitude, nugget = TRUE))

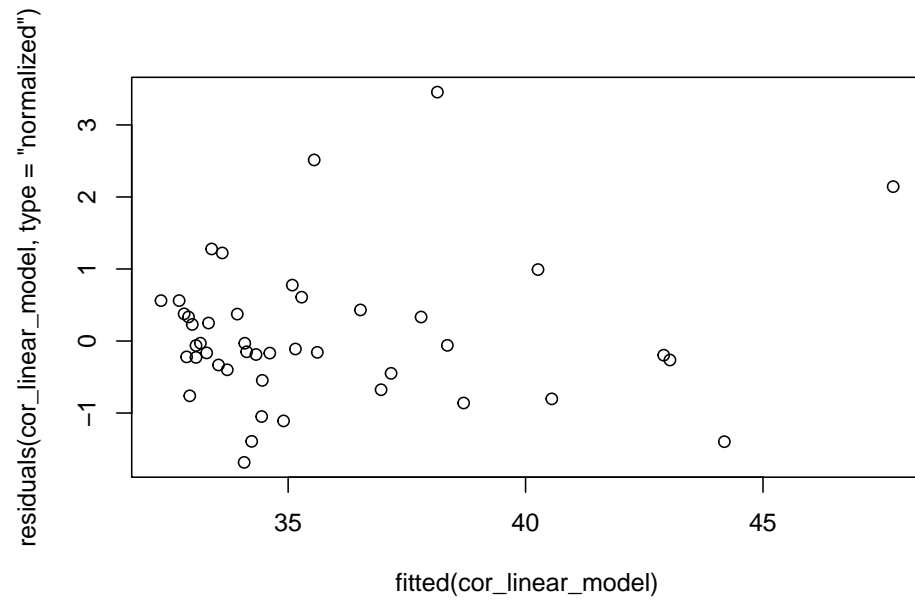
## Covariância Gaussiana
gauss_model <- gls(Riqueza ~ Prec_anual, exercicio_10,
  corGaus(form = ~Latitude + Longitude, nugget = TRUE))

## Covariância linear
cor_linear_model <- gls(Riqueza ~ Prec_anual, exercicio_10,
  corLin(form = ~Latitude + Longitude, nugget = TRUE))

## Covariância razão quadrática
ratio_model <- gls(Riqueza ~ Prec_anual, exercicio_10,
  corRatio(form = ~Latitude + Longitude, nugget = TRUE))

## Seleção de modelos
aic_fit <- AIC(no_spat_gls, espher_model, expon_model,
  cor_linear_model, gauss_model, ratio_model)
aic_fit %>% arrange(AIC)
#>           df      AIC
#> cor_linear_model  5 344.1434
#> gauss_model      5 344.3118
#> ratio_model      5 344.8400
#> no_spat_gls      3 345.4316
#> espher_model     5 345.6006
#> expon_model      5 346.2936

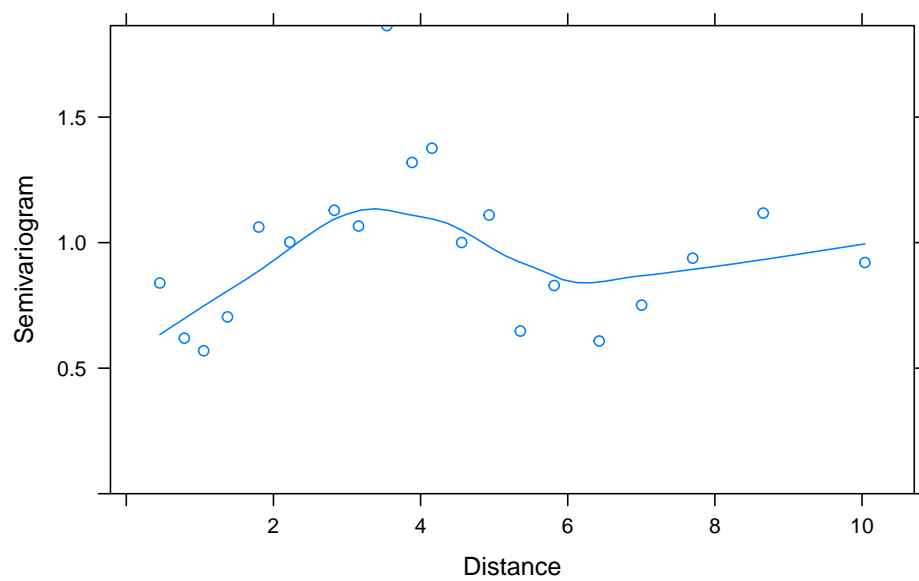
## Gráfico
plot(residuals(cor_linear_model, type = "normalized") ~ fitted(cor_linear_model))
```



```
## Varigrama
cor_linear_variog <- Variogram(cor_linear_model, form = ~Latitude + Longitude,
                               resType = "normalized")

plot(cor_linear_variog, main = "Variograma como Modelo de Covariância Linear")
```

Variograma como Modelo de Covariância Linear




```
## Resumen dos modelos
summary(cor_linear_model)$tTable
#>               Value      Std.Error  t-value    p-value
#> (Intercept) 17.25918513 73.686338088 0.234225 0.8159483
#> Prec_anual   0.01316726  0.008113421 1.622899 0.1120944
summary(no_spat_gls)$tTable
#>               Value      Std.Error  t-value    p-value
#> (Intercept) -3.35797556 8.99912469 -0.3731447 0.7109175029
#> Prec_anual   0.02288015 0.00630715  3.6276526 0.0007686691
```


Cap. 10 - Rarefação

```
# Pacotes necessários
library(iNEXT)
library(ecodados)
library(ggplot2)
```

10.1 Avalie se diferentes tipos de uso da terra (fragmento florestal, borda de mata, área de pastagem e cana de açúcar) apresentam diferentes riquezas de espécies? Qual a sua interpretação? Faça um gráfico com os resultados.

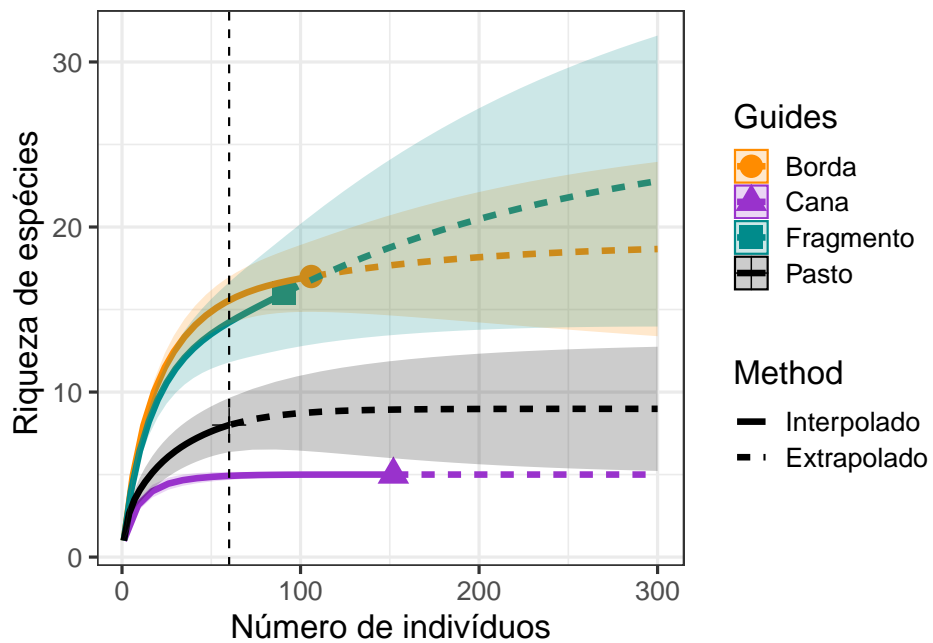
Solução:"

```
# Carregar a planilha com os dados
exercicio_1 <- ecodados::Cap10_exercicio1

## Número de indivíduos por local
colSums(exercicio_1)
#> Fragmento      Pasto      Borda      Cana
#>          91         60        106        152

## Análise de rarefação
resultados_exercicio_1 <- iNEXT(exercicio_1, q = 0,
                                datatype = "abundance", endpoint = 300)

## Gráfico
ggiNEXT(resultados_exercicio_1, type = 1) +
  geom_vline(xintercept = 60, lty = 2) +
  scale_linetype_discrete(labels = c("Interpolado", "Extrapolado")) +
  scale_colour_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  scale_fill_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  labs(x = "Número de indivíduos", y = "Riqueza de espécies") +
  theme_bw(base_size = 16)
```



10.2

O estudo é o mesmo do exercício anterior. Contudo, ao invés da rarefação baseada na abundância, faça rarefações baseadas no número de amostras. Qual a sua interpretação considerando os resultados do exercício 1? Faça um gráfico com os resultados.

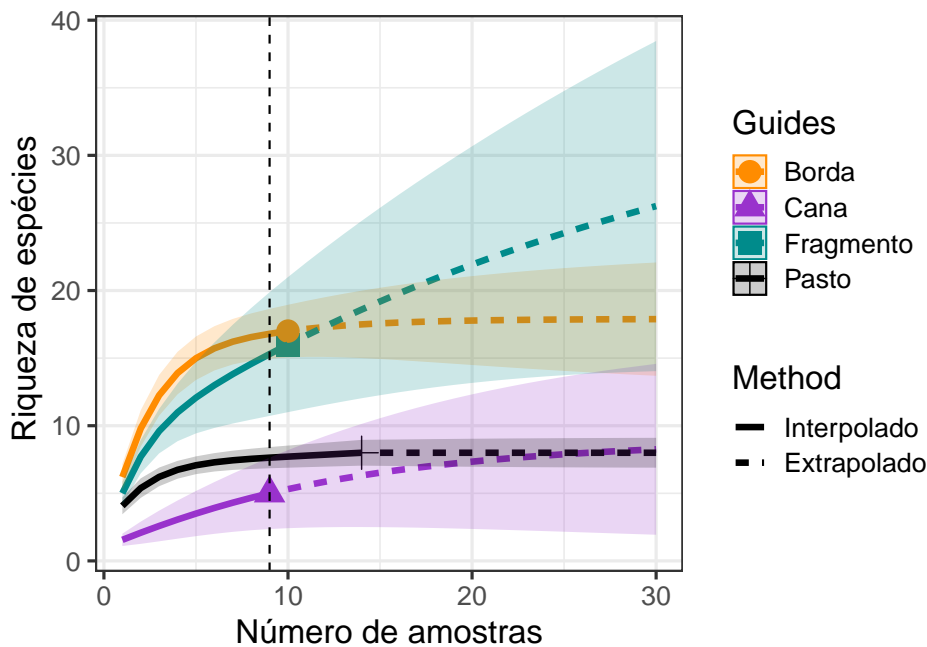
Solução:”

```
# Carregar a planilha com os dados
exercicio_2 <- ecodados::Cap10_exercicio2

# Verificar se carregou corretamente
head(exercicio_2)
#>      Fragmento Pasto Borda Cana
#> amostras      10   14   10   9
#> sp1           5    0    2    0
#> sp2           1    5    1    0
#> sp3           4    0    3    0
#> sp4           1   14    5    1
#> sp5           6    0    3    1

## Análise de rarefação
resultados_exercicio_2 <- iNEXT(exercicio_2, q = 0,
                                datatype = "incidence_freq", endpoint = 30)
```

```
## Gráfico
ggiNEXT(resultados_exercicio_2, type = 1, color.var = "site") +
  geom_vline(xintercept = 9, lty = 2) +
  scale_linetype_discrete(labels = c("Interpolado", "Extrapolado")) +
  scale_colour_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  scale_fill_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  labs(x = "Número de amostras", y = "Riqueza de espécies") +
  theme_bw(base_size = 16)
```



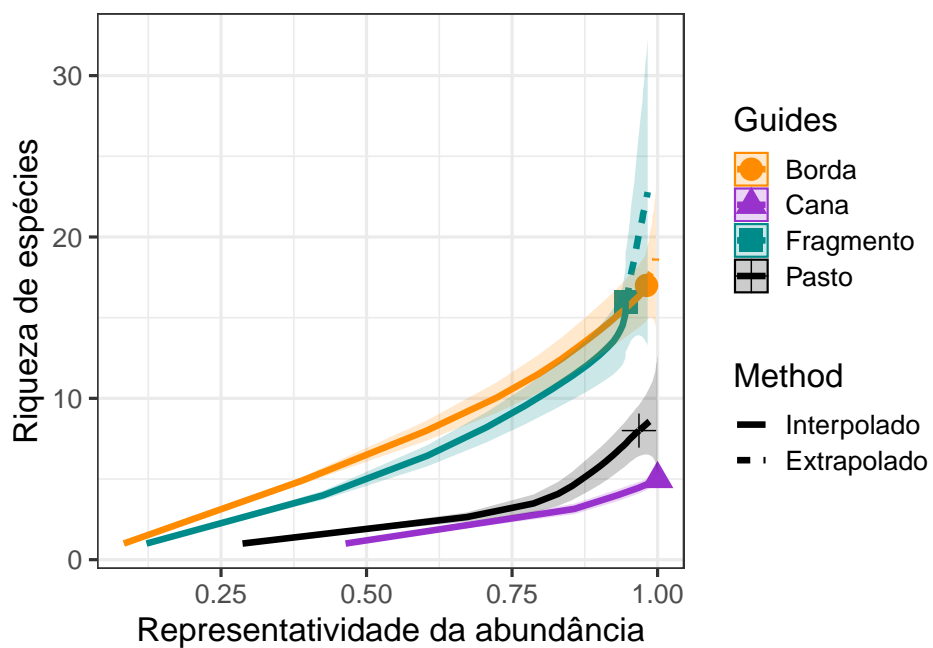
10.3 Use os dados dos exercícios anteriores e calcule a rarefação baseada na cobertura de amostragem (coverage-based). Qual a sua interpretação considerando os resultados anteriores? Faça um gráfico com os resultados.

Solução:”

```
# Rarefação considerando a cobertura da abundância
resultados_exercicio_1 <- iNEXT(exercicio_1, q = 0,
                                datatype = "abundance", endpoint = 300)
```

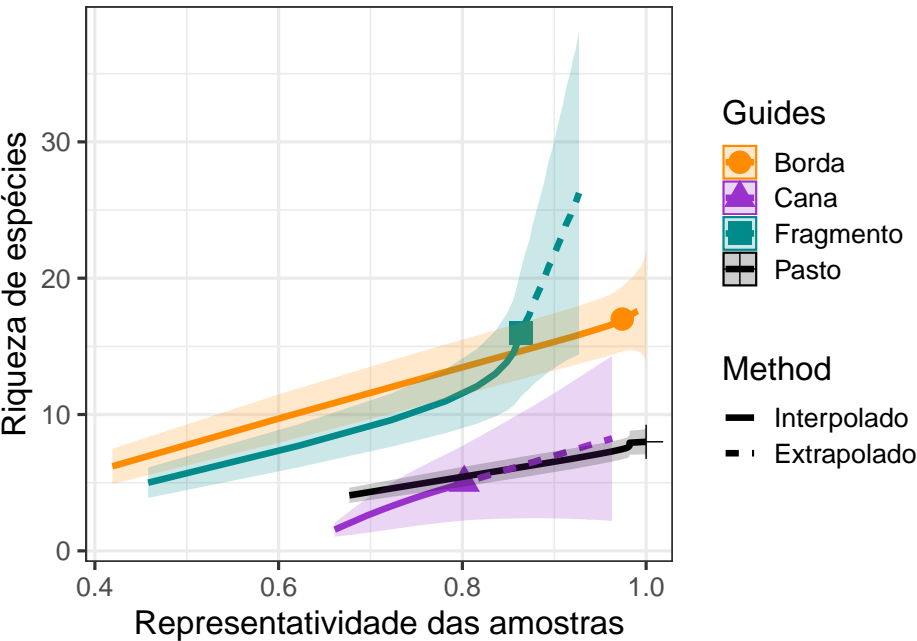
```
## Gráfico
# Visualizar os resultados da rarefação *coverage-based*.
ggiNEXT(resultados_exercicio_1, type = 3, color.var = "site") +
  scale_linetype_discrete(labels = c("Interpolado", "Extrapolado")) +
  scale_colour_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  scale_fill_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  labs(x = "Representatividade da abundância", y = "Riqueza de espécies") +
```

```
theme_bw(base_size = 16)
```



```
# Rarefação considerando a cobertura das amostras
resultados_exercicio_2 <- iNEXT(exercicio_2, q = 0,
                                datatype = "incidence_freq", endpoint = 30)

## Gráfico
# Visualizar os resultados da rarefação *coverage-based*.
ggiNEXT(resultados_exercicio_2, type = 3, color.var = "site") +
  scale_linetype_discrete(labels = c("Interpolado", "Extrapolado")) +
  scale_colour_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  scale_fill_manual(values = c("darkorange", "darkorchid", "cyan4", "black")) +
  labs(x = "Representatividade das amostras", y = "Riqueza de espécies") +
  theme_bw(base_size = 16)
```



Cap. 11 - Estimadores de riqueza

```
# Pacotes necessários
library(iNEXT)
library(ecodados)
library(ggplot2)
library(vegan)
#> Loading required package: permute
#> Loading required package: lattice
#> This is vegan 2.5-7
library(nlme)
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following object is masked from 'package:nlme':
#>
#> collapse
#> The following objects are masked from 'package:stats':
#>
#> filter, lag
#> The following objects are masked from 'package:base':
#>
#> intersect, setdiff, setequal, union
```

11.1 Carregue os dados - Cap11_exercicio1 - que está no pacote ecodados. Este conjunto de dados representa a abundância de 50 espécies de besouros coletados em 30 amostras. Calcule os estimadores de riqueza - Chao1 e ACE - e faça um gráfico contendo a riqueza observada e os dois estimadores de riqueza. Qual a sua interpretação sobre o esforço amostral?

Solução:

```
# Carregar a planilha com os dados
exercicio_1 <- ecodados::Cap11_exercicio1
```

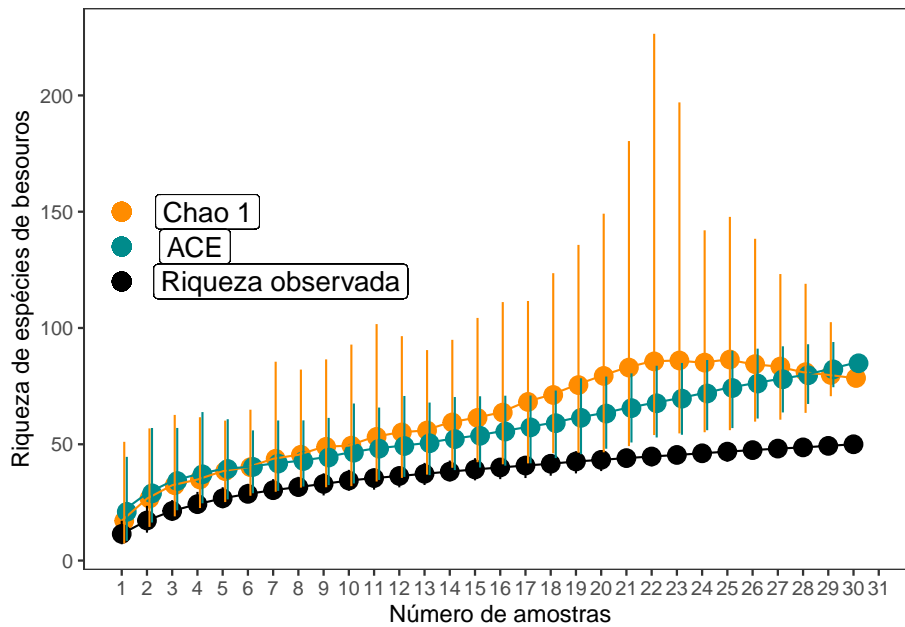
```

# estimadores abundância
est_abun <- estaccumR(exercicio_1, permutations = 100)

## Preparando os dados para fazer o gráfico
resultados_abun <- summary(est_abun
                           , display = c("S", "chao", "ace"))
res_abun <- cbind(resultados_abun$chao[, 1:4], resultados_abun$ace[, 2:4],
                  resultados_abun$S[, 2:4])
res_abun <- as.data.frame(res_abun)
colnames(res_abun) <- c("Amostras", "Chao", "C_inferior", "C_superior",
                       "ace", "A_inferior", "A_superior",
                       "Riqueza", "R_inferior", "R_superior")

## Gráfico
ggplot(res_abun, aes(y = Riqueza, x = Amostras)) +
  geom_point(aes(y = Chao, x = Amostras + 0.1), size = 4,
             color = "darkorange", alpha = 1) +
  geom_point(aes(y = ace, x = Amostras + 0.2), size = 4,
             color = "cyan4", alpha = 1) +
  geom_point(aes(y = Riqueza, x = Amostras), size = 4,
             color = "black", alpha = 1) +
  geom_point(y = 150, x = 1, size = 4, color = "darkorange", alpha = 1) +
  geom_point(y = 135, x = 1, size = 4, color = "cyan4", alpha = 1) +
  geom_point(y = 120, x = 1, size = 4, color = "black", alpha = 1) +
  geom_label(y = 150, x = 4.4, label = "Chao 1", size = 5) +
  geom_label(y = 135, x = 3.9, label = "ACE", size = 5) +
  geom_label(y = 120, x = 7.3, label = "Riqueza observada", size = 5) +
  geom_line(aes(y = Chao, x = Amostras), color = "darkorange") +
  geom_line(aes(y = ace, x = Amostras), color = "cyan4") +
  geom_line(aes(y = Riqueza, x = Amostras), color = "black") +
  geom_linerange(aes(ymin = C_inferior, ymax = C_superior,
                    x = Amostras + 0.1), color = "darkorange") +
  geom_linerange(aes(ymin = A_inferior, ymax = A_superior,
                    x = Amostras + 0.2), color = "cyan4") +
  geom_linerange(aes(ymin = R_inferior, ymax = R_superior,
                    x = Amostras), color = "black") +
  scale_x_continuous(limits = c(1, 31), breaks = seq(1, 31, 1)) +
  labs (x = "Número de amostras", y = "Riqueza de espécies de besouros") +
  theme_bw(base_size = 12) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

```



11.2

Utilize o mesmo conjunto de dados do exercício anterior. Calcule os estimadores de riqueza - Jackknife 1 e bootstrap. Faça um gráfico contendo a riqueza observada e os dois estimadores de riqueza. Qual a sua interpretação sobre o esforço amostral? Compare com os resultados do exercício anterior que utilizam estimadores baseados na abundância das espécies.

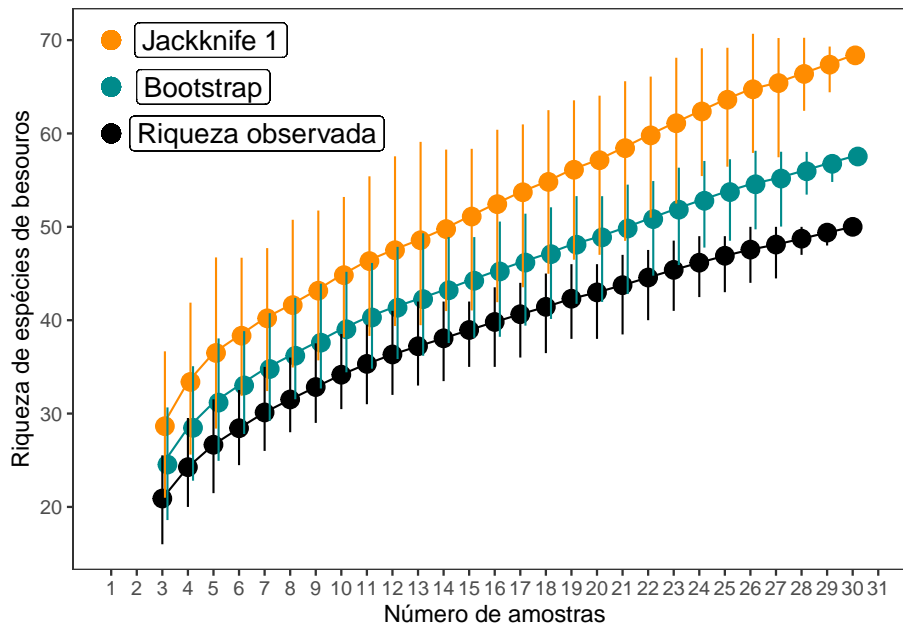
Solução:

```
# Carregar a planilha com os dados
exercicio_1 <- ecodados::Cap11_exercicio1

# estimadores incidencia
est_inc <- poolaccum(exercicio_1, permutations = 100)

## Preparando os dados para fazer o gráfico
resultados_inc <- summary(est_inc
                           , display = c("S", "jack1", "boot"))
res_inc <- cbind(resultados_inc$jack1[, 1:4], resultados_inc$boot[, 2:4],
                 resultados_inc$S[, 2:4])
res_inc <- as.data.frame(res_inc)
colnames(res_inc) <- c("Amostras", "jack1", "j_inferior", "j_superior",
                      "boot", "B_inferior", "B_superior",
                      "Riqueza", "R_inferior", "R_superior")
```

```
## Gráfico
ggplot(res_inc, aes(y = Riqueza, x = Amostras)) +
  geom_point(aes(y = jack1, x = Amostras + 0.1), size = 4,
             color = "darkorange", alpha = 1) +
  geom_point(aes(y = boot, x = Amostras + 0.2), size = 4,
             color = "cyan4", alpha = 1) +
  geom_point(aes(y = Riqueza, x = Amostras), size = 4,
             color = "black", alpha = 1) +
  geom_point(y = 70, x = 1, size = 4, color = "darkorange", alpha = 1) +
  geom_point(y = 65, x = 1, size = 4, color = "cyan4", alpha = 1) +
  geom_point(y = 60, x = 1, size = 4, color = "black", alpha = 1) +
  geom_label(y = 70, x = 4.9, label = "Jackknife 1", size = 5) +
  geom_label(y = 65, x = 4.6, label = "Bootstrap", size = 5) +
  geom_label(y = 60, x = 6.7, label = "Riqueza observada", size = 5) +
  geom_line(aes(y = jack1, x = Amostras), color = "darkorange") +
  geom_line(aes(y = boot, x = Amostras), color = "cyan4") +
  geom_line(aes(y = Riqueza, x = Amostras), color = "black") +
  geom_linerange(aes(ymin = j_inferior, ymax = j_superior,
                    x = Amostras + 0.1), color = "darkorange") +
  geom_linerange(aes(ymin = B_inferior, ymax = B_superior,
                    x = Amostras + 0.2), color = "cyan4") +
  geom_linerange(aes(ymin = R_inferior, ymax = R_superior,
                    x = Amostras), color = "black") +
  scale_x_continuous(limits = c(1, 31), breaks = seq(1, 31, 1)) +
  labs(x = "Número de amostras", y = "Riqueza de espécies de besouros") +
  theme_bw(base_size = 12) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```



11.3 Vamos refazer o exercício 10 do Capítulo 7 que usa Generalized Least Squares (GLS) para testar a relação da riqueza de anuros em 44 localidades na Mata Atlântica com a precipitação anual. Contudo, ao invés de considerar a riqueza de espécies de anuros observada como variável resposta, iremos utilizar a riqueza extrapolada. Utilize os dados “anuros_composicao” para estimar a riqueza extrapolada e o dados “anuros_ambientais” para acessar os dados de precipitação anual e coordenadas geográficas. Qual a sua interpretação dos resultados utilizando a riqueza observada e extrapolada?

Solução:

```
# Carregar a planilha com os dados
exercicio_3 <- ecodados::anuros_composicao

# Verificar a comunidade com maior abundância.
abund_max <- max(colSums(exercicio_3))

# Calcular a riqueza extrapolada de espécies para todas as comunidades
# considerando a maior abundância.
resultados_extrapolacao <- iNEXT(exercicio_3, q = 0,
                                datatype = "abundance",
                                endpoint = abund_max)

# Loop para determinar a riqueza extrapolada para as 44 localidades
resultados_comunidades_ext <- data.frame()
riqueza_extrapolada <- c()
```

```

for (i in 1:44){
  resultados_comunidades_ext <- data.frame(resultados_extrapolacao$iNextEst[i])
  riqueza_extrapolada[i] <- resultados_comunidades_ext[40, 4]
}

# carregando o data frame com todas as variáveis
exercicio_3_1 <- ecodados::anuros_ambientais

# Criando um data frame com a riqueza extrapolada, precipitação anual,
# latitude e longitude
dados_combinado_ext <- data.frame(riqueza_extrapolada, exercicio_3_1[,c(3,5,6)])

## Modelo gls sem estrutura espacial
no_spat_gls <- gls(riqueza_extrapolada ~ Prec_anual, dados_combinado_ext, method = "REML")

## Covariância esférica
espher_model <- gls(riqueza_extrapolada ~ Prec_anual, dados_combinado_ext,
  corSpher(form = ~Latitude + Longitude, nugget = TRUE))

## Covariância exponencial
expon_model <- gls(riqueza_extrapolada ~ Prec_anual, dados_combinado_ext,
  corExp(form = ~Latitude + Longitude, nugget = TRUE))

## Covariância Gaussiana
gauss_model <- gls(riqueza_extrapolada ~ Prec_anual, dados_combinado_ext,
  corGaus(form = ~Latitude + Longitude, nugget = TRUE))

## Covariância linear
cor_linear_model <- gls(riqueza_extrapolada ~ Prec_anual, dados_combinado_ext,
  corLin(form = ~Latitude + Longitude, nugget = TRUE),
  control = glsControl(opt='optim', msVerbose = F))

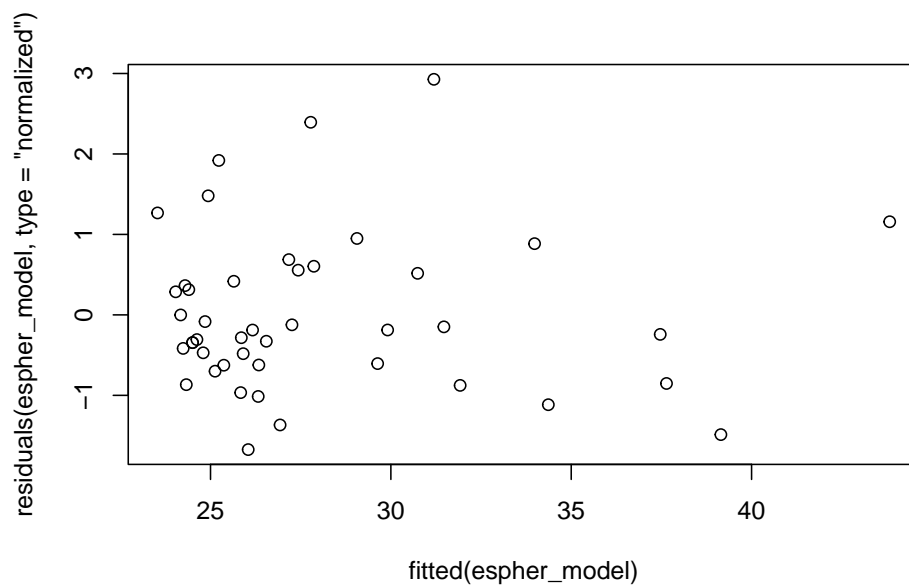
## Covariância razão quadrática
ratio_model <- gls(riqueza_extrapolada ~ Prec_anual, dados_combinado_ext,
  corRatio(form = ~Latitude + Longitude, nugget = TRUE))

## Seleção de modelos
aic_fit <- AIC(no_spat_gls, espher_model, cor_linear_model, expon_model,
  gauss_model, ratio_model)
aic_fit %>% arrange(AIC)
#>           df      AIC
#> no_spat_gls      3 338.8380
#> gauss_model      5 339.6672

```

```
#> ratio_model      5 339.6853
#> expon_model      5 340.3431
#> espher_model     5 342.8380
#> cor_linear_model 5 342.8489

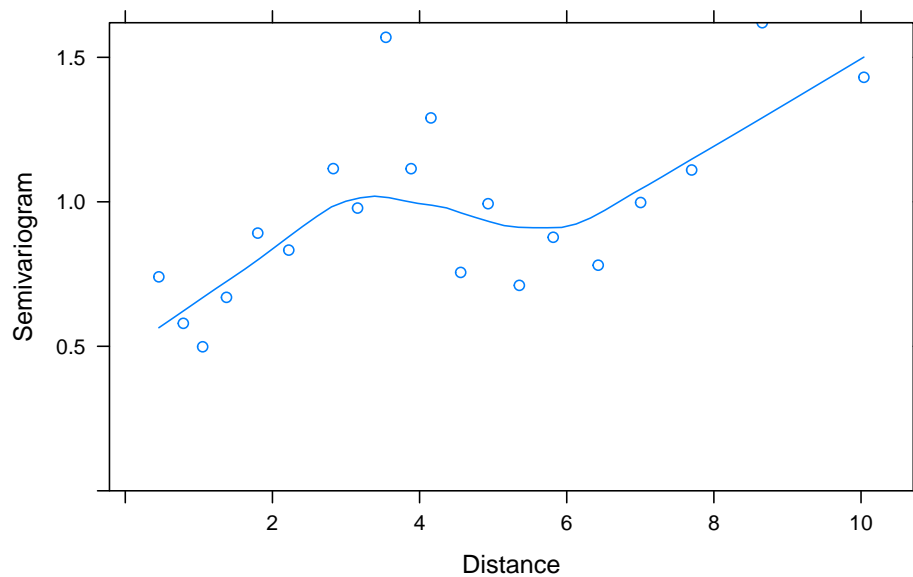
## Gráfico
plot(residuals(espher_model, type = "normalized") ~ fitted(espher_model))
```



```
## Varigrama
espher_model_variog <- Variogram(espher_model, form = ~Latitude + Longitude,
                                resType = "normalized")

plot(espher_model_variog, main = "Variograma com o modelo de Covariância Esférica")
```

Variograma com o modelo de Covariância Esférica



Resumo dos modelos

summary(espher_model)\$tTable

```
#>           Value Std.Error t-value    p-value
#> (Intercept) 3.7004741 8.319750795 0.4447818 0.658761200
#> Prec_anual  0.0173344 0.005831002 2.9727995 0.004869827
```

summary(no_spat_gls)\$tTable

```
#>           Value Std.Error t-value    p-value
#> (Intercept) 3.7004741 8.319750757 0.4447818 0.658761200
#> Prec_anual  0.0173344 0.005831002 2.9727995 0.004869827
```


Cap. 12 - Diversidade Taxônomica

```
# Pacotes necessários
library(devtools)
#> Loading required package: usethis
library(ecodados)
library (vegan)
#> Loading required package: permute
#>
#> Attaching package: 'permute'
#> The following object is masked from 'package:devtools':
#>
#>      check
#> Loading required package: lattice
#> This is vegan 2.5-7
library(ggplot2)
library(BiodiversityR)
#> Loading required package: tcltk
#> BiodiversityR 2.14-1: Use command BiodiversityRGUI() to launch the Graphical User Interface;
#> to see changes use BiodiversityRGUI(changeLog=TRUE, backward.compatibilty.messages=TRUE)
library(hillR)
library(betapart)
library(GGally)
#> Registered S3 method overwritten by 'GGally':
#>   method from
#>   +.gg      ggplot2
```

12.1 Carregue os dados - `anuros_composicao` - que está no pacote `ecodados`. Este conjunto de dados representa a abundância de 211 espécies de anuros coletados em 44 localidades na Mata Atlântica. Calcule a riqueza de espécies para cada comunidade e os índices de Margalef, Menhinich, Shannon-Wiener, Gini-Simpson e Equitabilidade de Pielou. Salve todos os resultados em novo data frame. Faça uma gráfico usando o função `ggpairs` para ver a correlação

entre as métricas. Qual a sua interpretação sobre os resultados?

Solução:

```
# Carregar a planilha com os dados
exercicio_1 <- ecodados::anuros_composicao

# Transpor o data frame para as espécies ficarem nas colunas e as localidades nas linhas
exercicio_1_t <- data.frame(t(exercicio_1))

## Calculando a riqueza observada de espécies para cada comunidade
Riqueza <- specnumber(exercicio_1_t)

## Calculando o Índice de Margalef para cada comunidade
Margalef <- round((Riqueza - 1)/log(apply(exercicio_1_t, 1, sum))), 2)

## Calculando o Índice de Menhinick para cada comunidade
Menhinick <- round(Riqueza/sqrt(apply(exercicio_1_t, 1, sum))), 2)

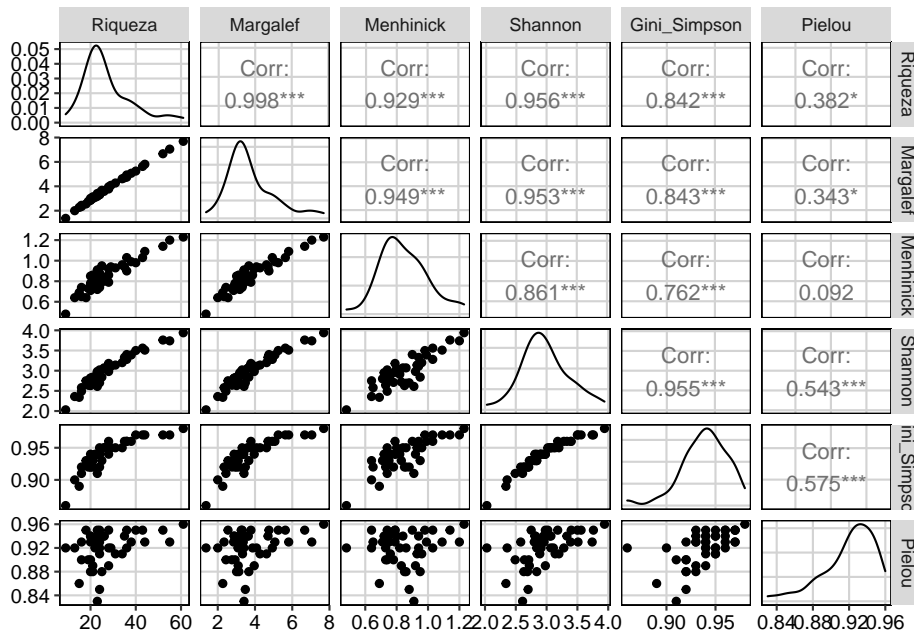
## Calculando o Índice de Shannon-Wiener para cada comunidade
Shannon <- round(diversity(exercicio_1_t, index = "shannon", MARGIN = 1), 2)

## Calculando o Índice de Gini-Simpson para cada comunidade
Gini_Simpson <- round(diversity(exercicio_1_t, index = "simpson", MARGIN = 1), 2)

## Calculando o Índice de Equitabilidade de Pielou para cada comunidade
Pielou <- round(Shannon/log(Riqueza), 2)

# Criando data frame
resultados <- data.frame(Riqueza, Margalef, Menhinick, Shannon, Gini_Simpson, Pielou)

# Gráfico
ggpairs(resultados, upper = list(continuous = wrap("cor", size = 4))) +
  tema_livro()
```



11.2 Usando os resultados anteriores, selecione as duas comunidades com os maiores e menores valores de Shannon-Wiener. Em seguida, faça um Diagrama de Whittaker. Por fim, interprete as curvas considerando as curvas teóricas (i.e., geométrica, broken-stick, etc.) descritas nos livros de ecologia.

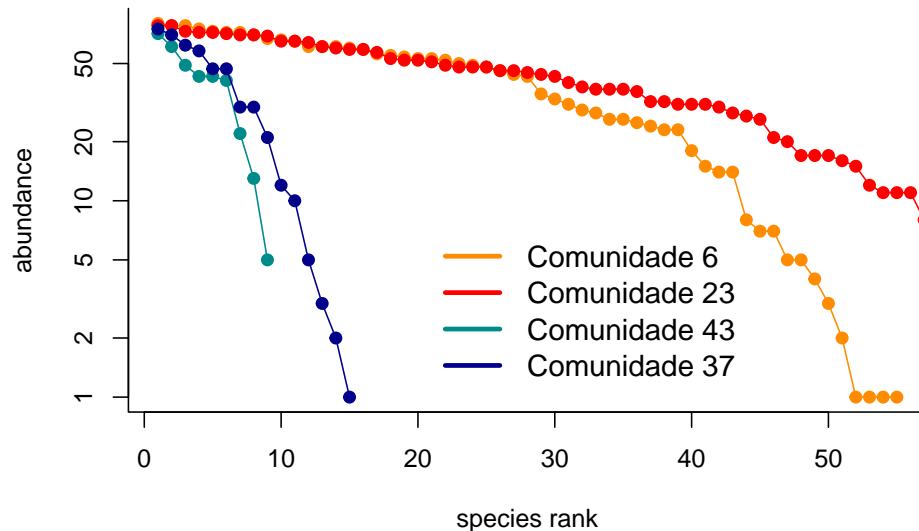
Solução:

```
# Usando os resultados no exercício anterior, vamos verificar quais são as comunidades com maiores e menores valores de Shannon
order(resultados$Shannon)
#> [1] 43 37 12 4 41 1 16 38 19 11 22 44 20 13 10 29 17 24
#> [19] 42 8 15 25 31 7 5 14 18 3 21 39 40 32 30 34 33 27
#> [37] 36 9 2 35 26 6 28 23

# AS comunidades com maiores valores estão nas linhas 6 e 23 e as comunidades com menores valores estão nas linhas 43 e 37
rank_com6 <- rankabundance(exercicio_1_t[6, exercicio_1_t[6,] > 0])
rank_com23 <- rankabundance(exercicio_1_t[23, exercicio_1_t[23,] > 0])
rank_com43 <- rankabundance(exercicio_1_t[43, exercicio_1_t[43,] > 0])
rank_com37 <- rankabundance(exercicio_1_t[37, exercicio_1_t[37,] > 0])

## Gráfico
rankabunplot(rank_com6, scale = "logabun", pch = 19, specnames = NULL,
              col = "darkorange")
rankabunplot(rank_com23, scale = "logabun", pch = 19, specnames = NULL,
              addit = TRUE, col = "red")
rankabunplot(rank_com43, scale = "logabun", pch = 19, specnames = NULL,
              addit = TRUE, col = "cyan4")
```

```
rankabunplot(rank_com37, scale = "logabun", pch = 19, specnames = NULL,
              addit = TRUE, col = "darkblue" )
legend(20, 8, legend = c("Comunidade 6", "Comunidade 23",
                          "Comunidade 43", "Comunidade 37"),
      col = c("darkorange", "red", "cyan4", "darkblue"), lty = 1, lwd = 3,
      cex = 1.2, box.lty = 0)
```



11.3 Usando os dados - anuros_composicao - calcule a partição da diversidade beta considerando os dados de abundância e presença e ausência. a) Faça um gráfico boxplot com os resultados. Discuta se os resultados usando abundância ou presença e ausência são congruentes ou discrepantes. b) Calcule a distância geográfica (use a planilha anuros_ambientais) entre as localidades (use a Distância euclidiana). Em seguida, faça uma análise de regressão para verificar se as localidades que estão próximas apresentam maior similaridade na composição de espécies (use componente turnover - Bsim) do que as comunidades que estão distantes (e.g., Decaimento da similaridade).

Solução:

```
# Carregar a planilha com os dados
anuros <- ecodados::anuros_composicao

# Transportar o data frame
anuros_t <- data.frame(t(anuros))

## Transformando dados em presença e ausência.
anuros_PA <- decostand(anuros_t, method = "pa")

## Diversidade beta presença e ausência
```

```
resultado_PA <- beta.pair(anuros_PA, index.family = "sorensen")

## Diversidade beta abundância
resultado_Abund <- beta.pair.abund(anuros_t, index.family = "bray")

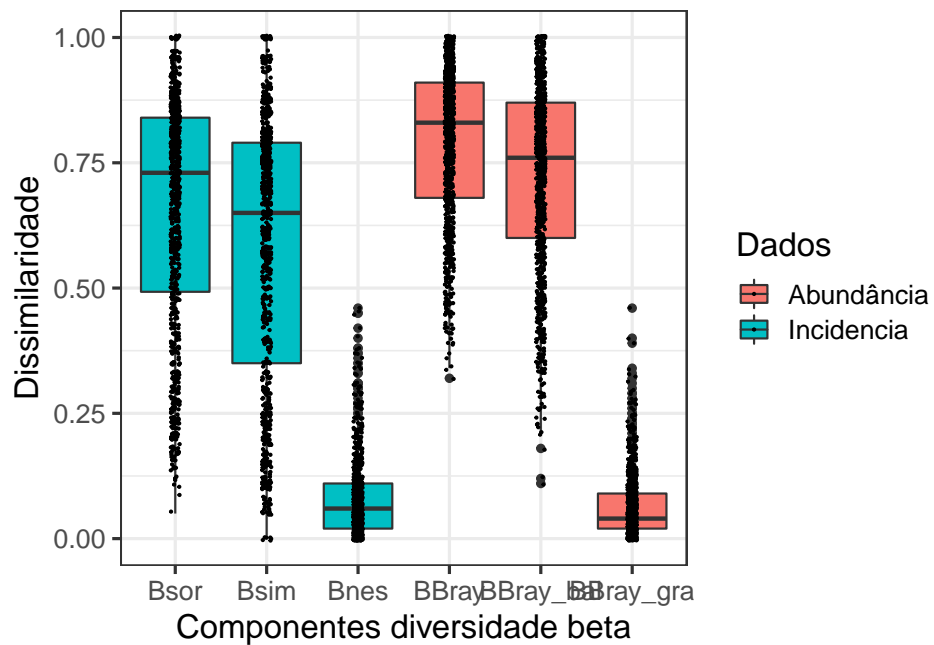
## Criando data frame com os resultados
diver_beta <- c(round(as.numeric(resultado_PA$beta.sor), 2),
                round(as.numeric(resultado_PA$beta.sim), 2),
                round(as.numeric(resultado_PA$beta.sne), 2),
                round(as.numeric(resultado_Abund$beta.bray), 2),
                round(as.numeric(resultado_Abund$beta.bray.bal), 2),
                round(as.numeric(resultado_Abund$beta.bray.gra), 2))

Componentes <- rep(c("Bsor", "Bsim", "Bnes", "BBray", "BBray_bal", "BBray_gra"),
                  each = 946)
Dados <- rep(c("Incidencia", "Abundância"),
            each = 2838)

data_frame <- data.frame(diver_beta, Componentes, Dados)

# Reordenando os componentes para melhorar visualização no gráfico
data_frame$Componentes <- factor(data_frame$Componentes , levels=c("Bsor", "Bsim", "Bnes",
                                                                    "BBray", "BBray_bal", "BBray_gra"))

## Gráfico
ggplot(data_frame, aes(y = diver_beta , x = Componentes, fill = Dados)) +
  geom_boxplot() +
  theme_bw(base_size = 16) +
  geom_jitter(size = 0.3, width = 0.05) +
  labs( x = "Componentes diversidade beta", y = "Dissimilaridade")
```



```
## carregando planilha anuros_ambientais

ambientes <- ecodados::anuros_ambientais

## Distância euclidiana
distancia <- vegdist(ambientes[,c(6,7)], "euclidean")

## criar data frame

dados <- data.frame(round(as.numeric(resultado_PA$beta.sim), 2),
                    round(as.numeric(distancia), 2))

colnames(dados) <- c("Bsim", "distancia")

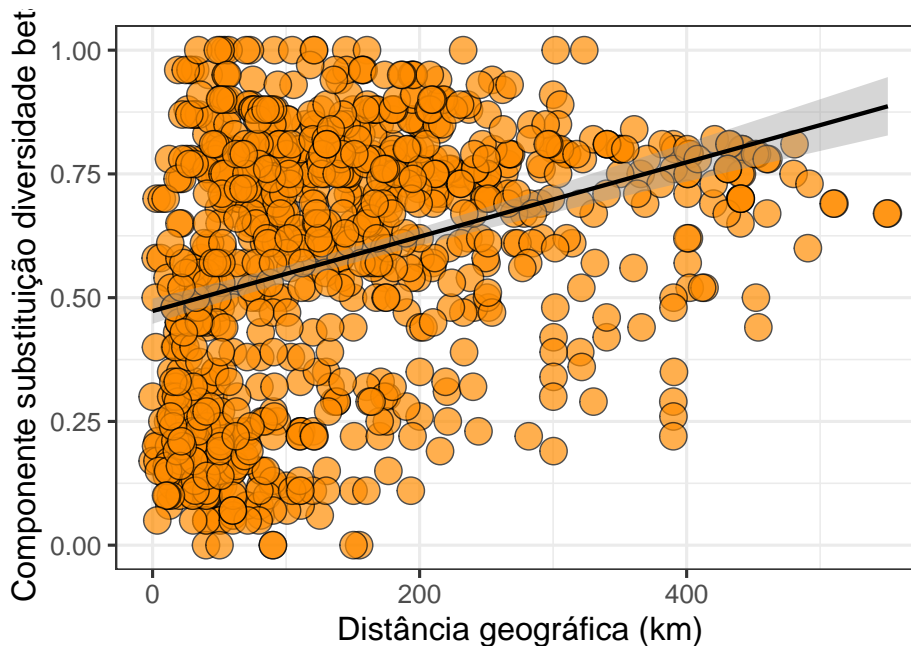
## Regressão - Contudo essa não é a melhor análise porque os índices de
# diversidade beta apresenta um platô.
regr <- lm (dados$Bsim ~ dados$distancia)
summary(regr)
#>
#> Call:
#> lm(formula = dados$Bsim ~ dados$distancia)
#>
#> Residuals:
```

```

#>      Min      1Q   Median      3Q      Max
#> -0.58919 -0.21205  0.03911  0.19848  0.50133
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    0.4729768   0.0129279   36.59  <2e-16 ***
#> dados$distancia 0.0075119   0.0007092   10.59  <2e-16 ***
#> ---
#> Signif. codes:
#> 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.2472 on 944 degrees of freedom
#> Multiple R-squared:  0.1062, Adjusted R-squared:  0.1053
#> F-statistic: 112.2 on 1 and 944 DF,  p-value: < 2.2e-16

## Gráfico
ggplot(data = dados, aes(x = distancia*10, y = Bsim)) +
  labs(x = "Distância geográfica (km)", y = "Componente substituição diversidade beta") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)

```



Cap. 13 - Diversidade Filogenética

```
# Pacotes necessários
library(devtools)
#> Loading required package: usethis
library(ecodados)
library(V.PhyloMaker)
#> Loading required package: ape
library(vegan)
#> Loading required package: permute
#>
#> Attaching package: 'permute'
#> The following object is masked from 'package:devtools':
#>
#>     check
#> Loading required package: lattice
#> This is vegan 2.5-7
library(ggplot2)
library(GGally)
#> Registered S3 method overwritten by 'GGally':
#>   method from
#>   +.gg      ggplot2
library(ggpubr)
#>
#> Attaching package: 'ggpubr'
#> The following object is masked from 'package:ape':
#>
#>     rotate
library(picante)
#> Loading required package: nlme
library(phytools)
#> Loading required package: maps
#>
```

```
#> Attaching package: 'phytools'
#> The following object is masked from 'package:vegan':
#>
#>      scores
library(ape)
library(geiger)
library(phyloregion)
library(pez)
library(reshape2)
library(betapart)
library(bbmle)
#> Loading required package: stats4
library(vegan)
```

13.1 Carregue os dados - `anuros_composicao` (i.e., 211 espécies de anuros coletados em 44 localidades na Mata Atlântica), `anuros_ambientais` (i.e., variáveis climáticas, topográficas e coordenadas geográficas) & `filogenia_anuros` (filogenia das 211 espécies) - que estão no pacote `ecodados`. Use a função `varpart` do pacote **vegan** para testar a importância relativa dos efeitos da precipitação anual, range altitudinal e temperatura anual na distribuição espacial da diversidade filogenética (PD) e Endemismo filogenético (PE). Calcule o SES para verificar se os resultados da diversidade filogenética (PD) diferem do esperado ao acaso devido ao número de espécies em cada comunidade. Qual a sua interpretação sobre os resultados?

Solução:

```
# Carregar as planilhas com os dados
anuros <- ecodados::anuros_composicao
anuros_t <- t(anuros) # Transpor dados
ambiente <- ecodados::anuros_ambientais
filogenia <- ecodados::filogenia_anuros

## Conferir os nomes das espécies
name.check(filogenia, anuros)
#> [1] "OK"

## Colocar os nomes das espécies do data frame na mesma ordem que aparecem na filogenia
anuros_P <- match.phylo.comm(phy = filogenia, comm = anuros_t)$comm

## Phylogenetic diversity (PD)
#####
anuros_PD <- pd(anuros_P, filogenia)

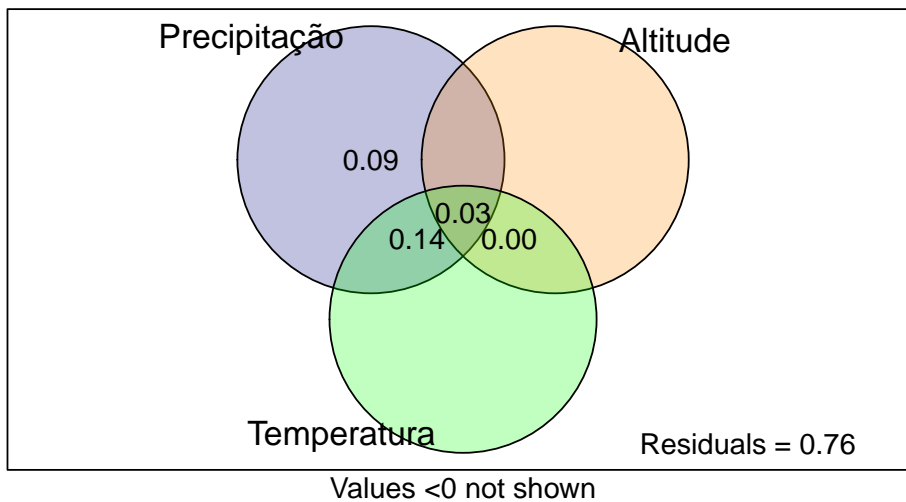
## Criar data frame
```

```

dados <- data.frame(anuros_PD$PD, anuros_PD$SR, ambiente$Prec_anual,
                    ambiente$Range_altitude, ambiente$Temp_anual)
colnames(dados) <- c("PD", "SR", "Prec_anual", "Range_altitude", "Temp_anual")

## Partição da variância
particao_PD <- varpart(dados$PD, dados$Prec_anual,
                      dados$Range_altitude, dados$Temp_anual)
plot (particao_PD, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Temperatura'),
      bg = c('navy', 'darkorange', 'green'))

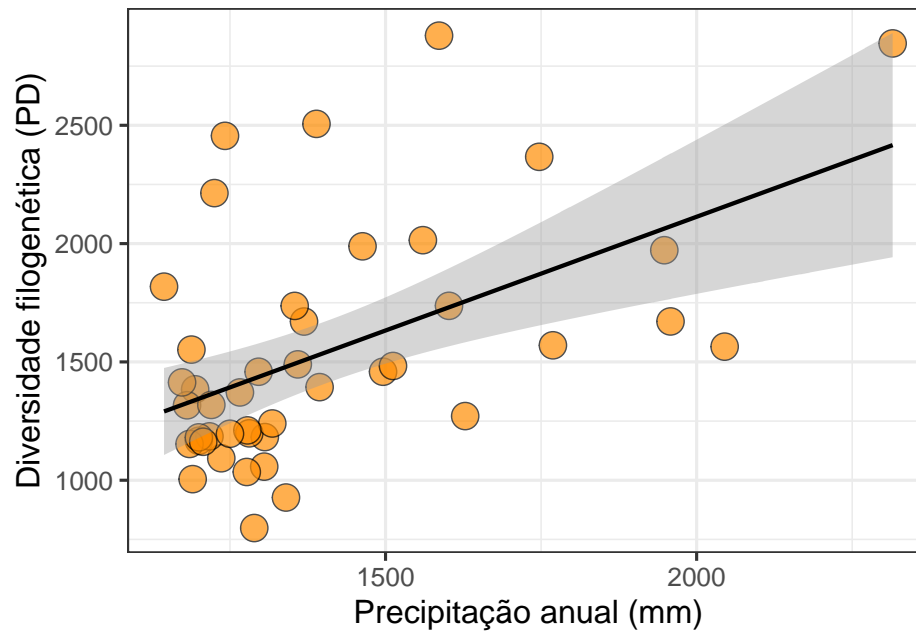
```



```

## Gráfico
ggplot(data = dados, aes(x = Prec_anual, y = PD)) +
  labs(x = "Precipitação anual (mm)", y = "Diversidade filogenética (PD)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)

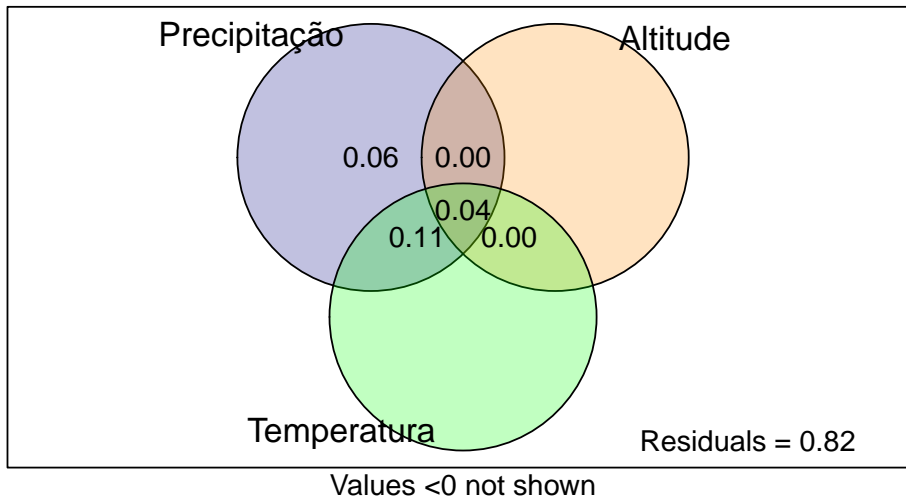
```



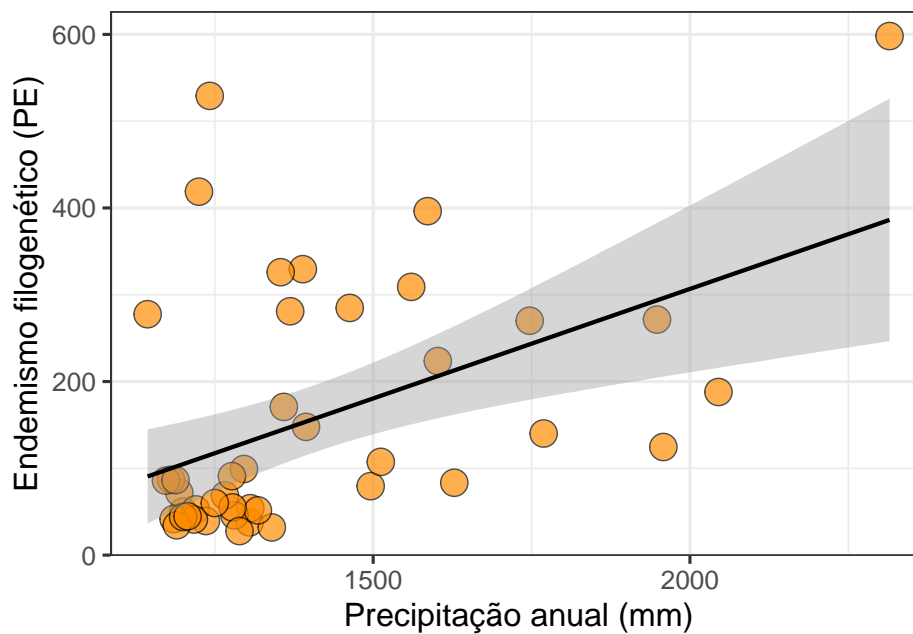
```
## Phylogenetic Endemism (PE)
#####
# Transformando data.frame em matriz.
anuros_matriz <- as.matrix(anuros_P)
anuros_PE <- phylo_endemism(anuros_matriz, filogenia, weighted = TRUE)

## Adicionando no data frame
dados$PE <- anuros_PE

## Partição da variância
particao_PE <- varpart(dados$PE, dados$Prec_anual,
                      dados$Range_altitude, dados$Temp_anual)
plot (particao_PE, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Temperatura'),
      bg = c('navy', 'darkorange', 'green'))
```



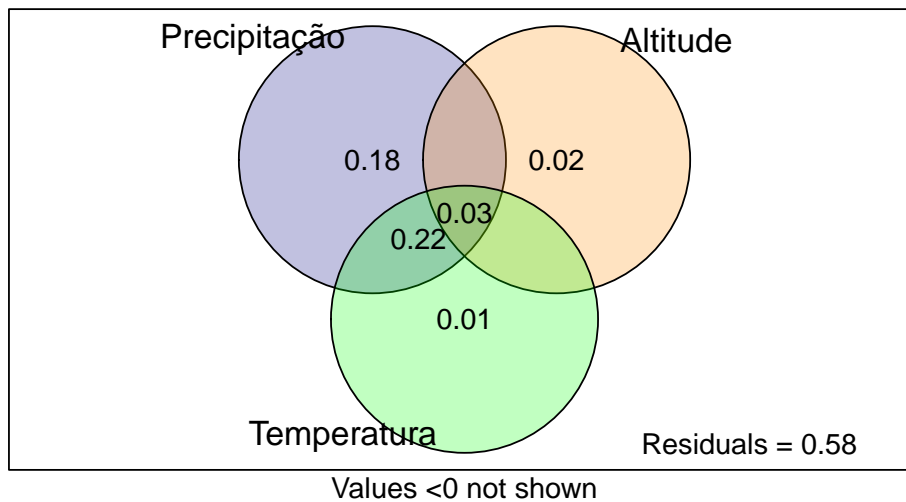
```
## Gráfico
ggplot(data = dados, aes(x = Prec_anual, y = PE)) +
  labs(x = "Precipitação anual (mm)", y = "Endemismo filogenético (PE)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```



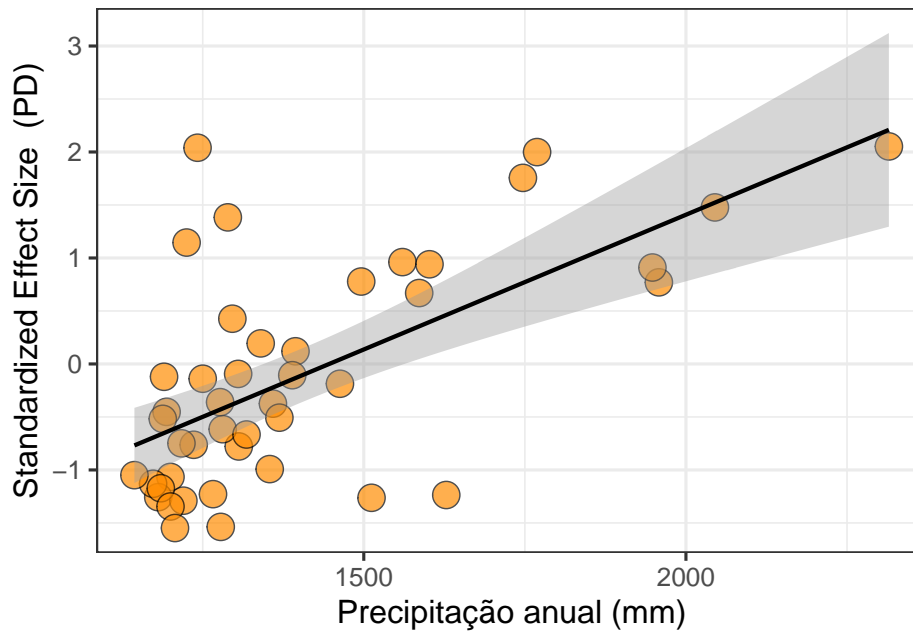
```
## Standardized Effect Size (SES) para o PD
#####
resultados_SES_PD <- ses.pd(anuros_P, filogenia,
                             null.model = "independentswap",
                             runs = 999)

## Adicionando no data frame
dados$SES <- resultados_SES_PD$pd.obs.z

## Partição da variância
particao_SES <- varpart(dados$SES, dados$Prec_anual,
                       dados$Range_altitude, dados$Temp_anual)
plot (particao_SES, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Temperatura'),
      bg = c('navy', 'darkorange', 'green'))
```



```
## Gráfico
ggplot(data = dados, aes(x = Prec_anual, y = SES)) +
  labs(x = "Precipitação anual (mm)", y = "Standardized Effect Size (PD)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```



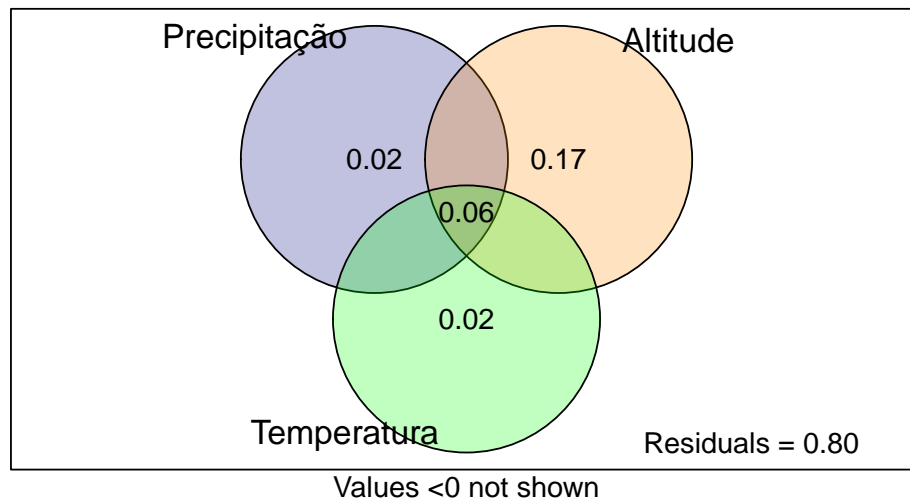
13.2 Carregue os dados - `anuros_composicao` (i.e., 211 espécies de anuros coletados em 44 localidades na Mata Atlântica), `anuros_ambientais` (i.e., variáveis climáticas, topográficas e coordenadas geográficas) & `filogenia_anuros` (filogenia das 211 espécies) - que estão no pacote `ecodados`. Use a função `varpart` do pacote `vegan` para testar a importância relativa dos efeitos da precipitação anual, range altitudinal e temperatura anual na distribuição espacial do NRI e NTI. Qual a sua interpretação sobre os resultados?

Solução:

```
## Nearest Relative Index (NRI)
#####
resultados_NRI <- ses.mpd(anuros_P, cophenetic(filogenia),
                          null.model = "taxa.labels",
                          abundance.weighted = FALSE,
                          runs = 999)

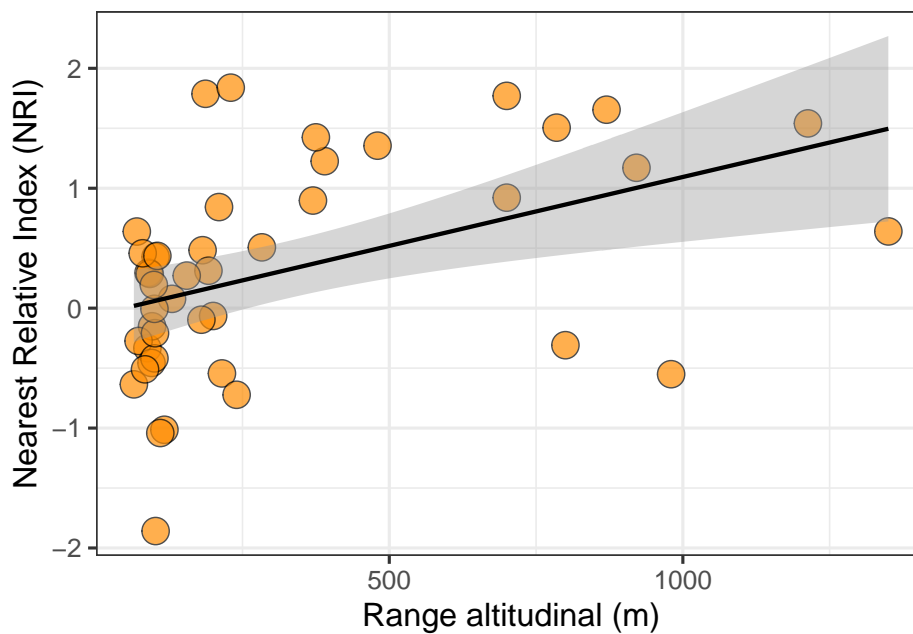
## Inserir no data frame
dados$NRI <- -1*(resultados_NRI$mpd.obs.z)

## Partição da variância
particao_NRI <- varpart(dados$NRI, dados$Prec_anual,
                      dados$Range_altitude, dados$Temp_anual)
plot (particao_NRI, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Temperatura'),
     bg = c('navy', 'darkorange', 'green'))
```



Gráfico

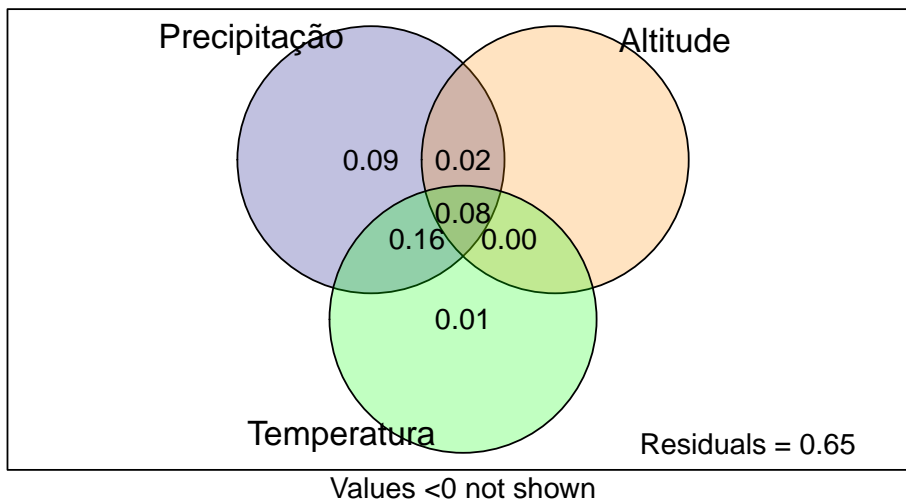
```
ggplot(data = dados, aes(x = Range_altitude, y = NRI)) +
  labs(x = "Range altitudinal (m)", y = "Nearest Relative Index (NRI)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```



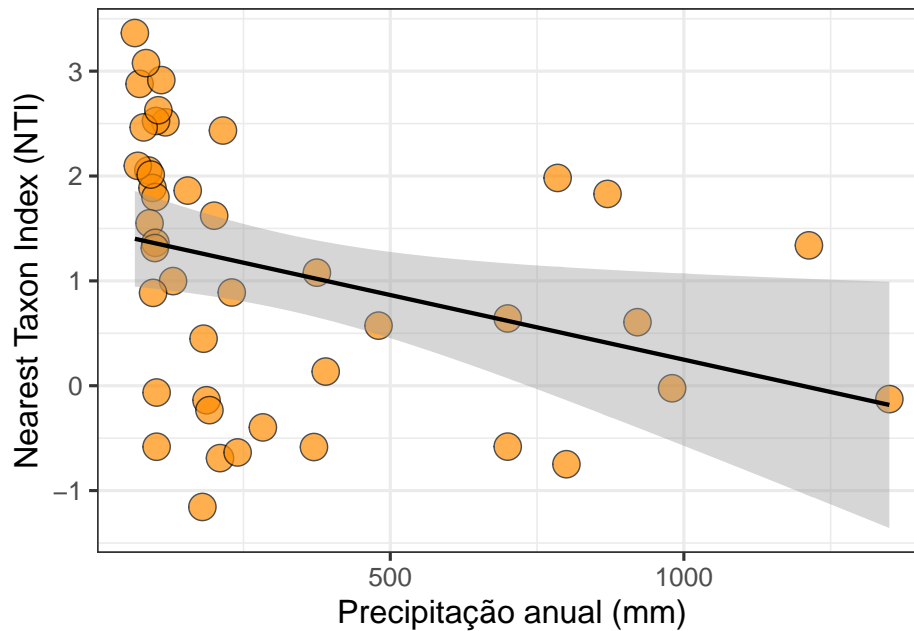

```
## Nearest Taxon Index (NTI)
#####
resultados_NTI <- ses.mntd(anuros_P, cophenetic(filogenia),
                           null.model = "taxa.labels",
                           abundance.weighted = FALSE,
                           runs = 999)

## Inserir no data frame
dados$NTI <- -1*(resultados_NTI$mntd.obs.z)

## Partição da variância
particao_NTI <- varpart(dados$NTI, dados$Prec_anual,
                      dados$Range_altitude, dados$Temp_anual)
plot (particao_NTI, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Temperatura'),
      bg = c('navy', 'darkorange', 'green'))
```



```
## Gráfico
ggplot(data = dados, aes(x = Range_altitude, y = NTI)) +
  labs(x = "Precipitação anual (mm)", y = "Nearest Taxon Index (NTI)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```



13.3 Carregue os dados - `anuros_composicao` (i.e., 211 espécies de anuros coletados em 44 localidades na Mata Atlântica), `anuros_ambientais` (i.e., variáveis climáticas, topográficas e coordenadas geográficas) & `filogenia_anuros` (filogenia das 211 espécies) - que estão no pacote `ecodados`. Use a função `varpartdo` pacote `vegan` para testar a importância relativa dos efeitos da precipitação anual, range altitudinal e distância geográfica na distribuição espacial dos diferentes componentes da diversidade beta filogenética (Phylosor). Qual a sua interpretação sobre os resultados?

Solução:

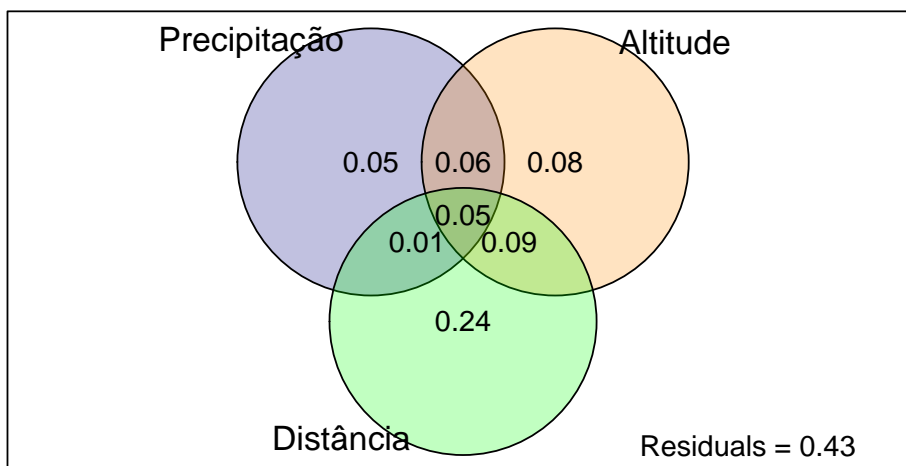
```
# Temos que transformar os dados para presença e ausência das espécies nas comunidades
anuros_PA <- decostand(anuros_P, "pa")

## Phylosor
#####
# Partição dos componentes do Phylosor.
resultados_Phylosor <- phylo.beta.pair(anuros_PA,
                                       filogenia,
                                       index.family = "sorensen")

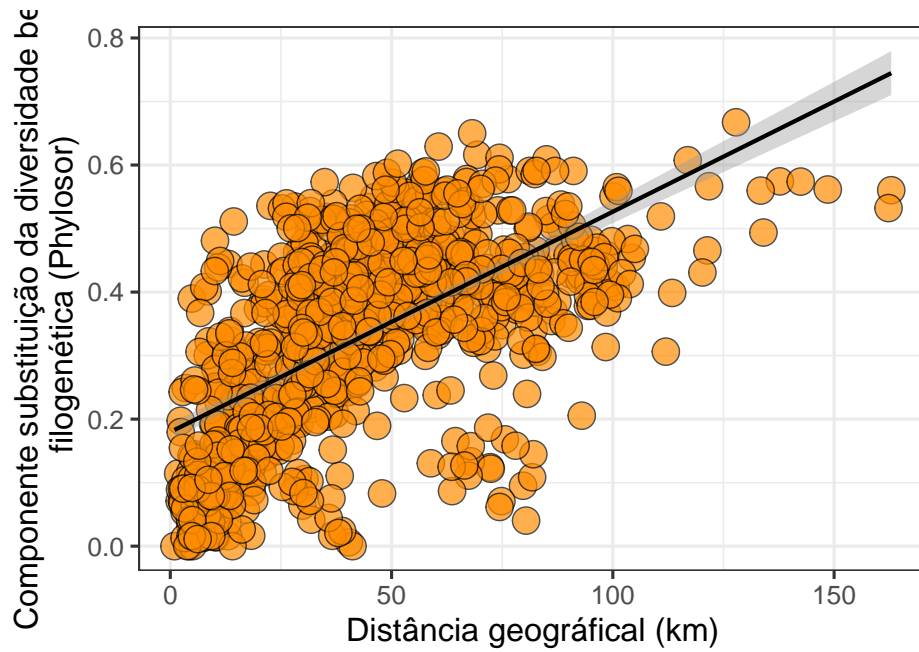
## vamos calcular a similaridade entre as variáveis ambientais
prec_dist <- vegdist(ambiente$Prec_anual, "euclidian")
alt_dist <- vegdist(ambiente$Range_altitude, "euclidian")
coord_dist <- vegdist(ambiente[,c(5,6)], "euclidian")
```

```
# Vamos preparar os dados .
dados_phylosor <- data.frame(
  substituicao = as.numeric(resultados_Phylosor$phylo.beta.sim),
  aninhamento = as.numeric(resultados_Phylosor$phylo.beta.sne),
  sorensen = as.numeric(resultados_Phylosor$phylo.beta.sor),
  dis_prec = as.numeric(prec_dist),
  dis_alt = as.numeric(alt_dist),
  dis_coord = as.numeric(coord_dist))

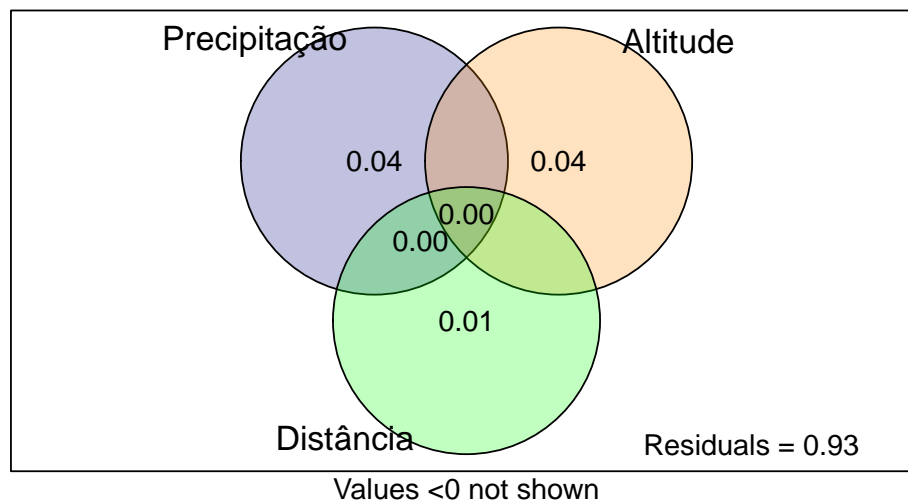
## Partição da variância do componente substituição do Phylosor
#####
particao_subst <- varpart(dados_phylosor$substituicao, dados_phylosor$dis_prec,
  dados_phylosor$dis_alt, dados_phylosor$dis_coord)
plot (particao_subst, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Distância'),
  bg = c('navy', 'darkorange', 'green'))
```



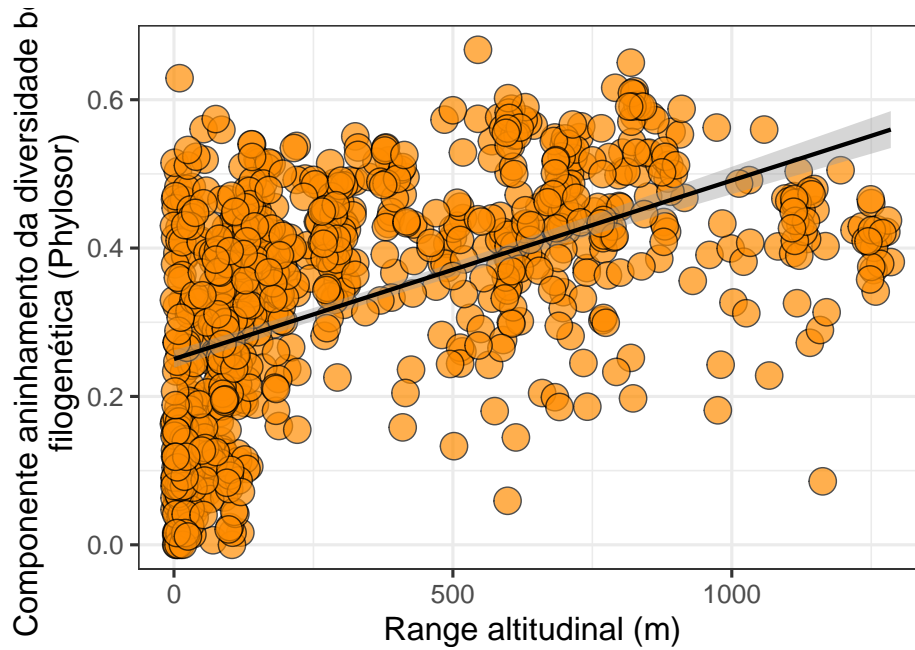
```
## Gráfico
ggplot(data = dados_phylosor, aes(x = dis_coord*10, y = substituicao)) +
  labs(x = "Distância geográfica (km)",
    y = "Componente substituição da diversidade beta \n filogenética (Phylosor)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```



```
## Partição da variância do componente aninhamento do Physolor
#####
particao_anin <- varpart(dados_phylosor$aninhamento, dados_phylosor$dis_prec,
                        dados_phylosor$dis_alt, dados_phylosor$dis_coord)
plot (particao_anin, digits = 1, Xnames = c('Precipitação', 'Altitude', 'Distância'),
      bg = c('navy', 'darkorange', 'green'))
```



```
## Gráfico
ggplot(data = dados_phylosor, aes(x = dis_alt, y = substituicao)) +
  labs(x = "Range altitudinal (m)",
       y = "Componente aninhamento da diversidade beta \n filogenética (Phylosor)") +
  geom_point(size = 6, shape = 21, fill = "darkorange", alpha = 0.7) +
  theme(legend.position = "none") +
  geom_smooth(method = lm, se = TRUE, color = "black") +
  theme_bw(base_size = 16)
```

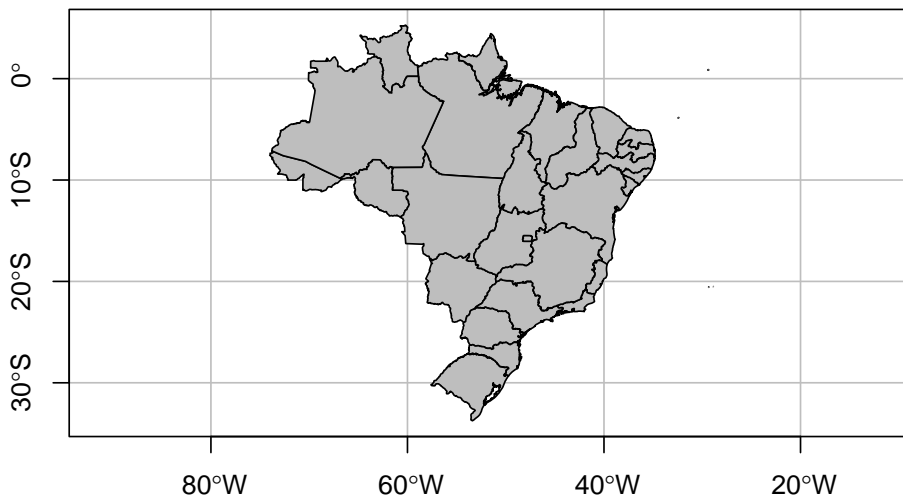


Cap. 15 - Dados geoespaciais no R

15.1 Importe o limite dos estados brasileiros no formato **sf** com o nome **br**. Para isso, use a função **ne_states** do pacote **rnaturalearth**. Crie um mapa simples cinza utilizando a função **plot()**, selecionando a coluna **geometry** com o operador **\$** e com os argumentos **axes** e **graticule** verdadeiros.

Solução:

```
library(rnaturalearth)
br <- rnaturalearth::ne_states(country = "Brazil", returnclass = "sf")
plot(br$geometry, col = "gray", axes = TRUE, graticule = TRUE)
```



15.2 Dados vetoriais podem ser criados com diversos erros de topologia, e.g., sobreposição de linhas ou polígonos ou buracos. Algumas funções exigem que os objetos vetoriais aos quais são atribuídos esses dados não possuam esses erros para que o algoritmo funcione. Para verificar se há erros, podemos usar a função **st_is_valid()** do pacote **sf**. Há diversas forma de correções desses erros,

mas vamos usar uma correção simples do R, com a função `st_make_valid()`. Vamos fazer essa correção para o `br` importado anteriormente e atribuindo ao objeto `br_valid`. Podemos conferir para saber se há erros e fazer um plot.

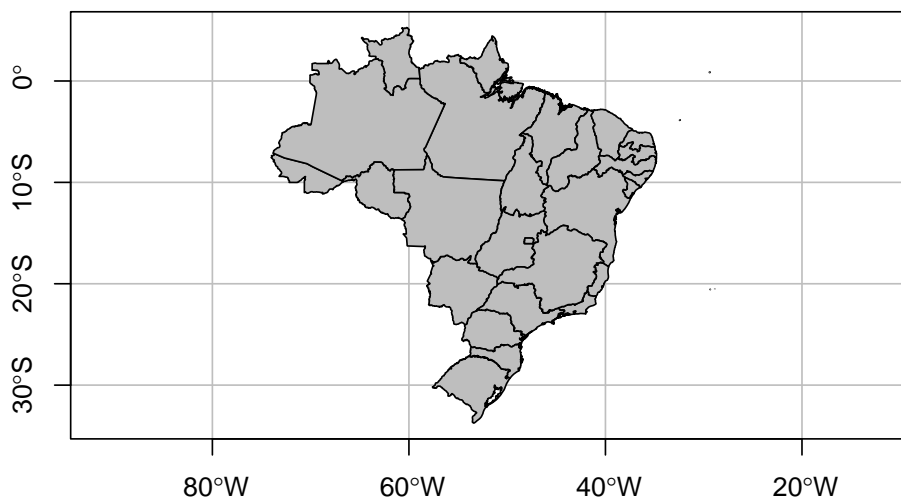
Solução:

```
library(sf)

sf::st_is_valid(br)
#> [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
#> [10] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
#> [19] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE

br_valid <- sf::st_make_valid(br)
sf::st_is_valid(br_valid)
#> [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
#> [12] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
#> [23] TRUE TRUE TRUE TRUE TRUE

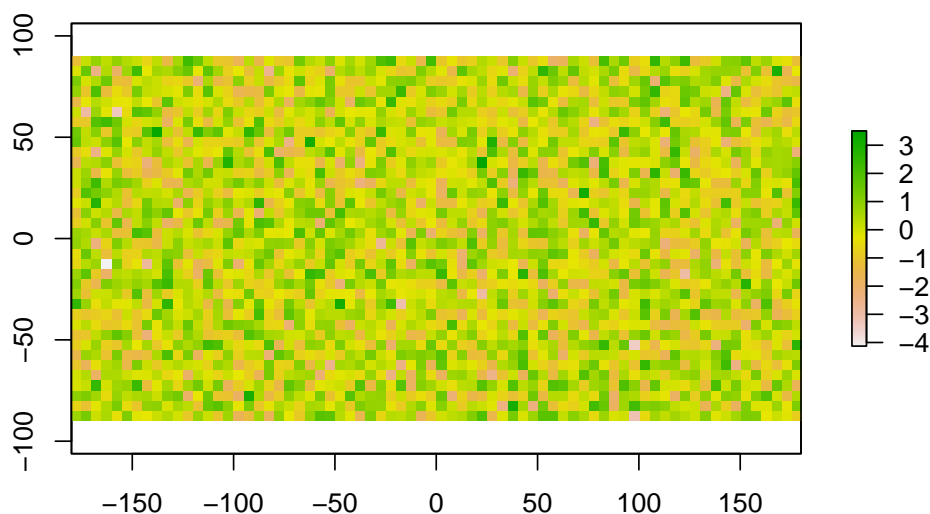
plot(br_valid$geometry, col = "gray", axes = TRUE, graticule = TRUE)
```



15.3 Crie um objeto `RasterLayer` vazio chamado `ra` com resolução: de 5° (~600 km). Atribua um sistema de referência de coordenadas com o código **4326**. Atribua valores aleatórios de uma distribuição normal e plote o mesmo.

Solução:

```
library(raster)
ra <- raster::raster(res = 5, crs = 4326)
raster::values(ra) <- rnorm(raster::ncell(ra))
plot(ra)
```

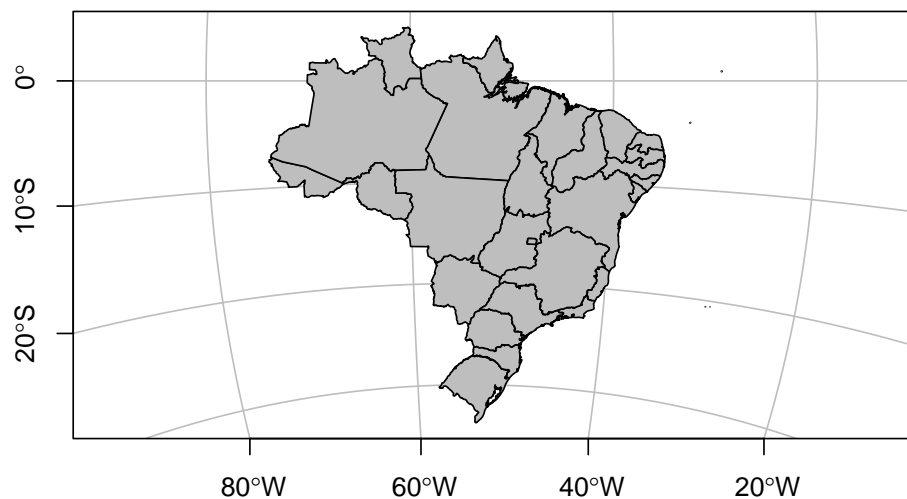



15.4 Reprojeite o limite dos estados brasileiros do exercício anterior para o CRS SIRGAS 2000/Brazil Polyconic, utilizando o código EPSG:5880 e chamando de **br_poly**. Faça um mapa simples como no exercício 1. Atente para as curvaturas das linhas.

Solução:

```
library(sf)
library(rnaturalearth)

br_valid_poly <- sf::st_transform(br_valid, crs = 5880)
plot(br_valid_poly$geometry, col = "gray", axes = TRUE, graticule = TRUE)
```



15.5 Utilizando a função **st_centroid** do pacote **sf**, crie um vetor chamado

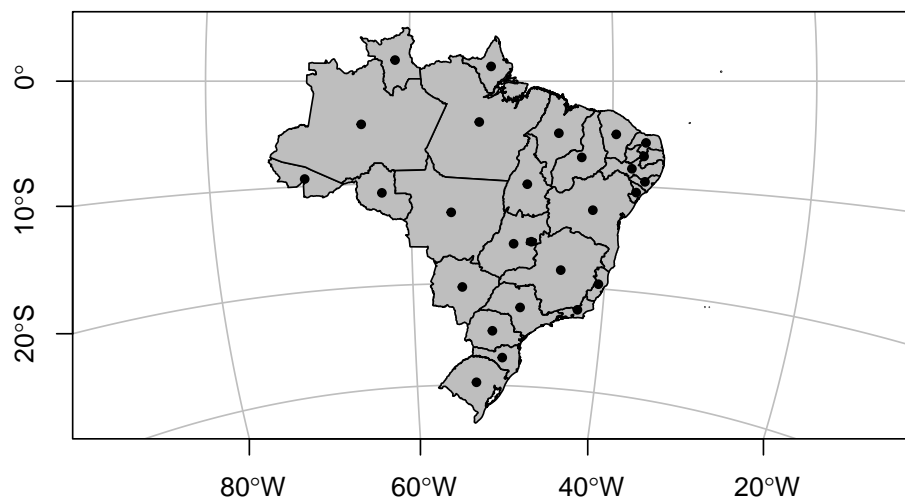
br_valid_cen que armazenará o centroide de cada estado brasileiro do objeto **br_valid** do exercício 2 e plot o resultado.

Solução:

```
library(sf)
library(rnaturalearth)

br_valid_poly_cen <- sf::st_centroid(br_valid_poly)

plot(br_valid_poly$geometry, col = "gray", axes = TRUE, graticule = TRUE)
plot(br_valid_poly_cen$geometry, pch = 20, add = TRUE)
```



15.6 Ajuste o limite e máscara do objeto raster criado no exercício 3 para o limite do Brasil, atribuindo ao objeto **ra_br**. Depois reprojete esse raster para a mesma projeção utilizada no exercício 4 com o nome **ra_br_poly** e plote o mapa resultante.

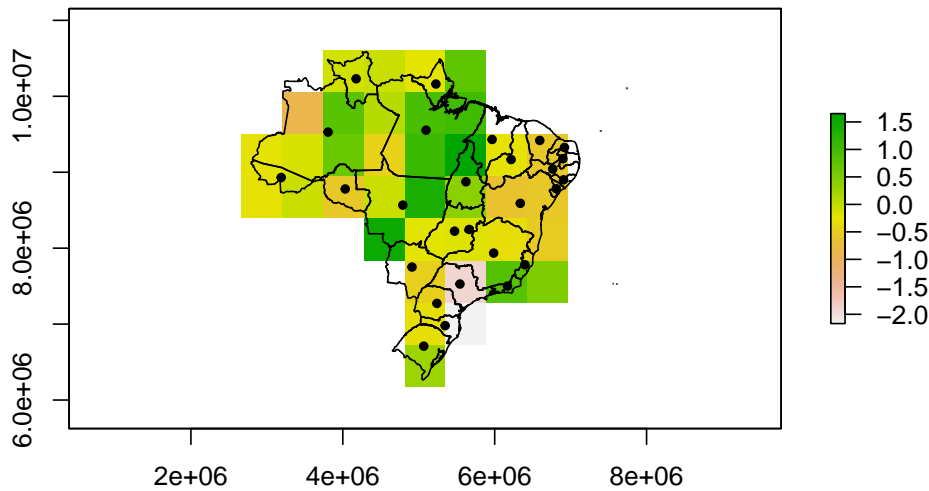
Solução:

```
library(raster)

ra_br <- ra %>%
  raster::crop(br_valid) %>%
  raster::mask(br_valid)

ra_br_poly <- raster::projectRaster(ra_br, crs = "+init=epsg:5880")

plot(ra_br_poly)
plot(br_valid_poly$geometry, add = TRUE)
plot(br_valid_poly_cen$geometry, pch = 20, add = TRUE)
```



15.7 Extraia os valores de cada pixel do raster criado no exercício 6 para os centroides dos estados do Brasil criado no exercício 5, atribuindo à coluna **val** do objeto espacial chamado **br_valid_poly_cent_ra**.

Solução:

```
br_valid_poly_cent_ra <- br_valid_poly_cen %>%
  dplyr::mutate(val = raster::extract(ra_br_poly, .))
head(br_valid_poly_cent_ra$val)
#> [1] 0.2676713 -0.1098317 0.9373188 -0.2846465 -0.2323340
#> [6] -0.4757399
```

15.8 Crie um mapa final usando os resultados dos exercícios 4, 5 e 6. Utilize o pacote **tmap** e inclua todos os principais elementos de um mapa.

Solução:

```
library(tmap)

tm_shape(ra_br_poly) +
  tm_raster(title = "Raster") +
  tm_shape(br_valid_poly) +
  tm_borders() +
  tm_shape(br_valid_poly_cent_ra) +
  tm_bubbles(col = "val", size = .2, legend.col.show = FALSE) +
  tm_graticules(lines = FALSE,
    labels.format = list(big.mark = ""),
    labels.rot = c(0, 90),
    labels.size = .7) +
  tm_compass(position = c("right", "top"), size = 2) +
  tm_scale_bar(size = 1) +
  tm_xlab("Longitude", size = 1) +
```

```

tm_ylab("Latitude", size = 1) +
tm_credits("CRS: SIRGAS2000/Polícônica", position = c(.6, .15), size = .6) +
tm_credits("Fonte: Natural Earth (2022)", position = c(.6, .12), size = .6) +
tm_layout(main.title = "Estados do Brasil",
           main.title.position = c(.1, .95),
           main.title.size = 1.5,
           title.fontface = "bold",
           legend.position = c("left", "bottom"),
           legend.title.fontface = "bold")

```

Estados do Brasil

