

ME115 - Linguagem R

Turma A - Profa. Larissa Avila Matos

Prova 1 - 20/04/2023 - 1S2023

GABARITO

Instruções

- Edite o cabeçalho do arquivo Rmd colocando seu nome e RA.
- Lembre-se de alterar a opção `eval = FALSE` no chunk acima para que os chunks subsequentes sejam avaliados.
- Use caminhos relativos para que eu possa executar o seu código no meu computador sem erros e sem ter que editá-lo.
- Para todos os problemas abaixo, escreva o código para responder cada uma das questões.
- Tenha certeza de que o seu arquivo .Rmd compila sem erros, gerando um arquivo HTML ou pdf.

Questão 1

(2,5 pontos)

Mostrar os comandos que podem ser usados para criar os objetos e/ou executar as instruções a seguir.

- (a) Construir uma matriz 10×10 tal que as entradas são iguais a $i*j$, sendo i a linha e j a coluna. (1,00 ponto)

Solução:

```
matriz <- matrix(0,10,10)
for(i in 1:10){
  for(j in 1:10){
    matriz[i,j] <- i*j
  }
}
matriz
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    2    3    4    5    6    7    8    9    10
## [2,]    2    4    6    8   10   12   14   16   18   20
## [3,]    3    6    9   12   15   18   21   24   27   30
## [4,]    4    8   12   16   20   24   28   32   36   40
## [5,]    5   10   15   20   25   30   35   40   45   50
## [6,]    6   12   18   24   30   36   42   48   54   60
## [7,]    7   14   21   28   35   42   49   56   63   70
## [8,]    8   16   24   32   40   48   56   64   72   80
## [9,]    9   18   27   36   45   54   63   72   81   90
## [10,]   10   20   30   40   50   60   70   80   90  100
```

- (b) Construir um data frame com três colunas: x , x^2 e $\exp(x)$, com x variando de 0 a 50 em uma unidade. (0,75 ponto)

Solução:

```
x <- 0:50
df <- data.frame(x = x, x2 = x^2, exp_x = exp(x))
head(df)
```

```
##   x x2      exp_x
## 1 0  0      1.000000
## 2 1  1      2.718282
## 3 2  4      7.389056
## 4 3  9     20.085537
## 5 4 16     54.598150
## 6 5 25    148.413159
```

- (c) Seja $x = (12, 11, 14, 15, 10, 11, 14, 11)$. Calcule $E = -n\lambda + (\sum_{i=1}^n x_i) \log(\lambda) - \sum_{i=1}^n \log(x_i!)$, onde n é o número de elementos do vetor x , $\lambda = 10$ e \log é o logaritmo natural. **Dica:** o fatorial de um número pode ser obtido utilizando a função `factorial()`. (0,75 ponto)

Solução:

```
x <- c(12, 11, 14, 15, 10, 11, 14, 11)
n <- length(x)
lambda <- 10
E <- -n*lambda + (sum(x)*log(lambda)) - sum(log(factorial(x)))
E

## [1] -20.22693
```

Questão 2

(2,0 pontos)

- (a) Escreva uma função que retorne o imposto pago por mulheres e por homens, sabendo que as mulheres pagam 10% e que os homens pagam 5% a mais do que as mulheres. A função deve ter dois parâmetros: `x` que é o valor a ser calculado e `sexo` que é o sexo da pessoa. Faça com que a função retorne o valor do imposto pago dependendo do sexo da pessoa. Teste sua função calculando o imposto do valor de 1000 de uma mulher e de um homem. (1,00 ponto)

Solução: Considerando que os argumentos do parâmetro `sexo` são: "M": Mulher e "H": Homem.

```
imposto <- function(x, sexo){
  imposto <- ifelse(sexo == "M", x*0.10, x*0.15)
  return(imposto)
}
imposto(1000, "M")
```

```
## [1] 100
```

```
imposto(1000, "H")
```

```
## [1] 150
```

A função acima não leva em consideração se o usuário entra com o argumento `sexo` errado, por exemplo, você coloca "C" no argumento `sexo`, a função irá retornar o imposto pago por homens.

```
imposto(1000, "C")
```

```
## [1] 150
```

Uma alternativa para a função `imposto()` é:

```
imposto1 <- function(x, sexo){
  if(sexo == "M") imposto <- x*0.10
```

```

else if(sexo == "H") imposto <- x*0.15
else imposto <- NA

return(imposto)
}

```

```
imposto1(1000, "M")
```

```
## [1] 100
```

```
imposto1(1000, "H")
```

```
## [1] 150
```

```
imposto1(1000, "C")
```

```
## [1] NA
```

Ou,

```

imposto2 <- function(x, sexo){
  imposto <- ifelse(sexo == "M", x*0.10, ifelse(sexo == "H", x*0.15, NA))
  return(imposto)
}

```

```
imposto2(1000, "M")
```

```
## [1] 100
```

```
imposto2(1000, "H")
```

```
## [1] 150
```

```
imposto2(1000, "C")
```

```
## [1] NA
```

(b) Resolva o item (a) utilizando a função `switch()`. (1,00 ponto)

Solução:

```

imposto_switch <- function(x, sexo){
  imposto <- switch(sexo, "M" = x*0.10, "H" = x*0.15)
  return(imposto)
}

```

```
imposto_switch(1000, "M")
```

```
## [1] 100
```

```
imposto_switch(1000, "H")
```

```
## [1] 150
```

```
imposto_switch(1000, "C")
```

```
## NULL
```

Questão 3

(3,0 pontos)

Nessa questão, exploraremos o conjunto de dados `airquality`. O conjunto de dados contém medições diárias da qualidade do ar de Nova York durante um período de cinco meses e possui as seguintes informações:

- **Ozone:** concentração média de ozônio (ppb),
- **Solar.R:** radiação solar (Langley),
- **Wind:** velocidade média do vento (mph),
- **Temp:** temperatura máxima diária em graus Fahrenheit,
- **Month:** mês numérico (maio=5, junho=6 e assim por diante),
- **Day:** dia numérico do mês (1-31).

Há muitas coisas que seriam interessantes de se olhar neste conjunto de dados. Qual foi a temperatura média durante o período? Qual dia foi o mais quente? Qual dia foi o mais ventoso? Em que dias a temperatura foi superior a 90 graus Fahrenheit?

(a) Responda a, pelo menos, três das perguntas citadas acima. (0,75 ponto)

Solução: A temperatura média durante o período foi de 77.88, com o dia mais quente em 28/8 (onde a temperatura foi de 97 graus Fahrenheit). O dia mais ventoso foi em 17/6 (com velocidade média do vento no dia de 20.7).

```
mean(airquality$Temp)
```

```
## [1] 77.88235
```

```
max(airquality$Temp)
```

```
## [1] 97
```

```
airquality[which.max(airquality$Temp), c("Day", "Month")]
```

```
##      Day Month
```

```
## 120  28      8
```

```
max(airquality$Wind)
```

```
## [1] 20.7
```

```
airquality[which.max(airquality$Wind), c("Day", "Month")]
```

```
##      Day Month
```

```
##  48  17      6
```

Além disso, os dias em que temperatura foi superior a 90 graus Fahrenheit foram:

```
airquality[airquality$Temp > 90, c("Day", "Month")]
```

```
##      Day Month
```

```
## 42   11      6
```

```
## 43   12      6
```

```
## 69    8      7
```

```
## 70    9      7
```

```
## 75   14      7
```

```
## 102  10      8
```

```
## 120  28      8
```

```
## 121  29      8
```

```
## 122  30      8
```

```
## 123  31      8
```

```
## 124   1      9
```

```
## 125   2      9
```

```
## 126   3      9
```

```
## 127   4      9
```

(b) Use funções apropriadas para descrever esse conjunto de dados. Cite o número de variáveis, número de observações e os nomes das variáveis que ele armazena. (0,75 ponto)

Solução:

O conjunto de dados `airquality` possui 153 observações, 6 variáveis, cujos nomes são: Ozone, Solar.R, Wind, Temp, Month, Day.

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
## $ Ozone   : int  41 36 12 18 NA 28 23 19 8 NA ...
## $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
## $ Wind    : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
## $ Temp    : int  67 72 74 62 56 66 65 59 61 69 ...
## $ Month   : int   5 5 5 5 5 5 5 5 5 5 ...
## $ Day     : int   1 2 3 4 5 6 7 8 9 10 ...
```

- (c) A função `cut()` pode ser usada para criar uma variável categórica a partir de uma variável numérica, dividindo-a em categorias correspondentes a diferentes intervalos. Utilizando essa função, crie uma nova variável categórica nos dados de qualidade do ar, chamada `TempCat`, que divide a variável `Temp` em três intervalos (50,70], (70,90], (90,110). Adicione a nova variável categórica (`TempCat`) ao objeto `airquality`. (0,75 ponto)

Solução:

```
TempCat <- cut(airquality$Temp, breaks = c(50, 70, 90, 110))
str(TempCat)
```

```
## Factor w/ 3 levels "(50,70]","(70,90]","...: 1 2 2 1 1 1 1 1 1 1 ...
```

```
airquality$TempCat <- TempCat
str(airquality)
```

```
## 'data.frame':    153 obs. of  7 variables:
## $ Ozone   : int  41 36 12 18 NA 28 23 19 8 NA ...
## $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
## $ Wind    : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
## $ Temp    : int  67 72 74 62 56 66 65 59 61 69 ...
## $ Month   : int   5 5 5 5 5 5 5 5 5 5 ...
## $ Day     : int   1 2 3 4 5 6 7 8 9 10 ...
## $ TempCat: Factor w/ 3 levels "(50,70]","(70,90]","...: 1 2 2 1 1 1 1 1 1 1 ...
```

- (d) Usando uma função da família *apply*, calcule a radiação solar média para cada um dos intervalos da variável `TempCat`. (0,75 ponto)

Solução:

```
tapply(airquality$Solar.R, airquality$TempCat, mean)
```

```
## (50,70] (70,90] (90,110]
##      NA      NA      225
```

```
# Desconsiderando os NA's
```

```
tapply(airquality$Solar.R, airquality$TempCat, mean, na.rm = TRUE)
```

```
## (50,70] (70,90] (90,110]
## 167.2667 186.0588 225.0000
```

Questão 4

(2,5 pontos)

Considere o código a seguir:

```
shannon <- function(dados){
  prop <- dados/sum(dados)
  res <- numeric()
  n <- length(prop)
  for(i in 1:n){
    if(prop[i] > 0){
      res[i] <- prop[i]*log(prop[i])
    } else {
      res[i] <- 0
    }
  }
  H <- -sum(res)
  return(H)
}

x <- c(235,218,192,0,20,11,11,8,7,4,3,2,2,1,1)
shannon(x)
```

```
## [1] 1.508226
```

Explique com as suas palavras o que o código faz e interprete a sua saída. Reescreva essa função usando apenas 3 linhas de comando. Confira o resultado aplicando o seu código na variável `x`.

Solução:

O código calcula uma métrica, ele calcula

$$H = - \sum_{i=1}^n p_i \log(p_i),$$

onde p_i é o valor em proporção do i -ésimo elemento e \log é logaritmo natural. Durante o cálculo da métrica é colocado uma restrição quando o valor da proporção é zero, pois $\log(0)$ não existe ($-\infty$), ou seja, quando essa proporção é zero, esse elemento recebe valor zero.

Curiosidade: Essa métrica é o índice de diversidade de Shannon.

```
shannon <- function(dados){
  prop <- dados/sum(dados)
  H <- sapply(prop, function(x) ifelse(x > 0, x*log(x), 0))
  return(-sum(H))
}

x <- c(235,218,192,0,20,11,11,8,7,4,3,2,2,1,1)
shannon(x)
```

```
## [1] 1.508226
```