

Atividade nº 03 - Estruturas de Dados em R

Lucas Henrique Nogueira

25/05/2024

Lista de Exercícios

1) Matriz:

- Dado o seguinte sistema de equações:

$$\begin{cases} x + 2y + 3z = 20 \\ 2x + 5y + 9z = 100 \\ 5x + 7y + 8z = 200 \end{cases}$$

- Reescrevendo na forma de resolução matricial:

$$Ax = B \implies \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 9 \\ 5 & 7 & 8 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 20 \\ 100 \\ 200 \end{pmatrix}$$

```
A <- matrix(c(1, 2, 3, 2, 5, 9, 5, 7, 8), nrow = 3, byrow = T)
B <- c(20, 100, 200)
```

- Foi constatado pelo cálculo do determinante que o sistema possui uma única solução:

$$\det(A) = 2 > 0 \implies \text{Solução única}$$

```
det.A <- det(A)
det.A
```

```
## [1] 2
```

- Por fim a solução foi determinada:

$$S = \begin{cases} x = 320 \\ y = -360 \\ z = 140 \end{cases}$$

```
if(det.A != 0){
  sol <- solve(A, B)
  sol
} else {
  print("O Sistema não possui solução")
}
```

```
## [1] 320 -360 140
```

2) Fatores:

- a) Troca de níveis de um vetor de fatores:
 - Dado um vetor de fatores:

```
z <- factor(c("p", "q", "p", "r", "q"))
```

```
z <- factor(c("p", "q", "p", "r", "q"))
z
```

```
## [1] p q p r q
## Levels: p q r
```

- Trocando o nível “p” por “w”

```
levels(z)[levels(z) == "p"] <- "w"
z
```

```
## [1] w q w r q
## Levels: w q r
```

- b) Concatenação de dois vetores de fatores e funcionamento do comando `sample()` com a semente 666:

$$\begin{cases} s1 <- \text{factor}(\text{sample}(\text{letters}, \text{size}=5, \text{replace}=\text{TRUE})) & (1) \\ s2 <- \text{factor}(\text{sample}(\text{letters}, \text{size}=5, \text{replace}=\text{TRUE})) & (2) \end{cases}$$

– Gerando números aleatórios com o comando `sample()`:

```
set.seed(666)
sample(1:10, replace = T)
```

```
## [1] 5 9 1 2 3 5 3 6 8 7
```

– Criando os vetores S1 e S2 e concatenando-os

```
s1 <- factor(sample(letters, size = 5, replace = TRUE))
s1
```

```
## [1] r q z p f
## Levels: f p q r z
```

```
s2 <- factor(sample(letters, size = 5, replace = TRUE))
s2
```

```
## [1] o x i d k
## Levels: d i k o x
```

```
s_concatenado <- factor(c(s1, s2))
s_concatenado
```

```
## [1] r q z p f o x i d k
## Levels: f p q r z d i k o x
```

- c) Conjunto de dados `iris` e a função `cut()` com as seguintes frequências (32, 41, 42, 24, 11):

```
data(iris)

intervalos <- c(4.30, 5.02, 5.74, 6.46, 7.18, 7.901)
rotulos <- c("(4.3, 5.02]", "(5.02, 5.74]", "(5.74, 6.46]",
             "(6.46, 7.18]", "(7.18, 7.9]")

dados.categorizados <- cut(iris$Sepal.Length, breaks = intervalos,
                           labels = rotulos, right = F)
contagem <- table(dados.categorizados)
contagem
```

```
## dados.categorizados
## (4.3, 5.02] (5.02, 5.74] (5.74, 6.46] (6.46, 7.18] (7.18, 7.9]
##          32          41          42          24          11
```

- d) Gerando uma tabela de contingência a partir do conjunto de dados iris com o fator Species:

```
categorias <- table(iris$Sepal.Length < 5, iris$Species)

rownames(categorias) <- c("FALSE", "TRUE")
colnames(categorias) <- levels(iris$Species)

categorias
```

```
##
##      setosa versicolor virginica
## FALSE      30          49        49
## TRUE       20           1         1
```

- e) Adicionando mais um nível no vetor de fator de resposta já criado:

```
respostas <- factor(c("Concordo", "Concordo", "Concordo fortemente", "Discordo", "Concordo"))

respostas <- factor(c("Concordo", "Concordo", "Concordo fortemente",
                      "Discordo", "Concordo"))
respostas <- factor(respostas, levels = c("Concordo fortemente",
                                           levels(respostas)[-which(levels(respostas) == "Concordo fortemente")]))

levels(respostas) <- c(levels(respostas), "Discordo fortemente")
respostas
```

```
## [1] Concordo          Concordo          Concordo fortemente
## [4] Discordo            Concordo
## Levels: Concordo fortemente Concordo Discordo Discordo fortemente
```

- f) Encontrando os índices dos valores e combinando com os índices dos níveis do fator:

```
x <- factor(c("alto", "baixo", "médio", "alto", "alto", "baixo", "médio"))

x <- factor(c("alto", "baixo", "médio", "alto", "alto", "baixo", "médio"))

df <- data.frame(levels = unique(x), value = seq_along(unique(x)))
df
```

```
##  levels value
## 1   alto     1
## 2  baixo     2
## 3  médio     3
```

3) Listas:

- a) Dado p, q e x, qual o valor de x[2]?

$$\begin{cases} p <- c(2, 7, 8) \\ q <- c("A", "B", "C") \\ x <- list(p, q) \end{cases} \quad x[2] = "A", "B", "C"$$

```
p <- c(2, 7, 8)
q <- c("A", "B", "C")
x <- list(p, q)
x[2]
```

```
## [[1]]
## [1] "A" "B" "C"
```

- b) Dado w, v e x, qual comando substituirá “A” em x por “K”?

$$\begin{cases} w <- c(2, 7, 8) \\ v <- c("A", "B", "C") \\ x <- list(w, v) \end{cases} \quad \begin{aligned} x[1] &= 2, 7, 8 \\ x[2] &= "K", "B", "C" \end{aligned}$$

```
w <- c(2, 7, 8)
v <- c("A", "B", "C")
x <- list(w, v)
x[[2]] <- replace(x[[2]], x[[2]] == "A", "K")
x
```

```
## [[1]]
## [1] 2 7 8
##
## [[2]]
## [1] "K" "B" "C"
```

- c) Dado a lista a, qual comando dará a soma de todos os elementos de a?

$$a <- list("x" = 5, "y" = 10, "z" = 15) \quad \text{soma} = 30$$

```
a <- list("x" = 5, "y" = 10, "z" = 15)
soma <- sum(unlist(a))
soma
```

```
## [1] 30
```

- d) Dado a lista `nova`, qual o comando irá adicionar 1 a cada elemento do primeiro vetor de `nova`?

`nova <- list(a = 1:10, b = "Férias", c = "Olá") \implies nova$a = 2 3 4 5 6 7 8 9 10 11`

```
nova <- list(a = 1:10, b = "Férias", c = "Olá")
nova$a <- nova$a + 1
nova
```

```
## $a
## [1]  2  3  4  5  6  7  8  9 10 11
##
## $b
## [1] "Férias"
##
## $c
## [1] "Olá"
```

- e) Dado a lista `b`, qual o comando irá imprimir todos os elementos de `b` exceto o segundo elemento do primeiro vetor de `b`?

`b <- list(a = 1:10, c = "Olá", d = "AA") \implies b$a = 1 3 4 5 6 7 8 9 10`

```
b <- list(a = 1:10, c = "Olá", d = "AA")
b$a <- b$a[-2]
b
```

```
## $a
## [1]  1  3  4  5  6  7  8  9 10
##
## $c
## [1] "Olá"
##
## $d
## [1] "AA"
```

- f) Dado a lista `x`, qual o comando irá acrescentar um novo item `z` à lista `x` com `z = "Novo item"`?

$$x \leftarrow \text{list}(a = 5:10, c = \text{"Olá"}, d = \text{"AA"}) \implies x\$z = \text{"Novo item"}$$

```
x <- list(a = 5:10, c = "Olá", d = "AA")
x$z <- "Novo item"
x
```

```
## $a
## [1] 5 6 7 8 9 10
##
## $c
## [1] "Olá"
##
## $d
## [1] "AA"
##
## $z
## [1] "Novo item"
```

- g) Dado a lista `y`, qual o comando irá atribuir os nomes (“um”, “dois”, “três”) aos três elementos de `y`?

$$y \leftarrow \text{list}(1:10, \text{"string"}, 666) \implies \begin{cases} y\$um = 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ y\$dois = \text{"string"} \\ y\$três = 666 \end{cases}$$

```
y <- list(1:10, "string", 666)
names(y) <- c("um", "dois", "três")
y
```

```
## $um
## [1] 1 2 3 4 5 6 7 8 9 10
##
## $dois
## [1] "string"
##
## $três
## [1] 666
```

- h) Dado a lista `x`, qual o comando irá apresentar o comprimento do vetor `r`, de `x`?

$$x \leftarrow \text{list}(y = 1:10, t = \text{"Olá"}, f = \text{"TT"}, r = 5:20) \implies \text{comprimento_r} = 16$$

```
x <- list(y = 1:10, t = "Olá", f = "TT", r = 5:20)
comprimento_r <- length(x$r)
comprimento_r
```

```
## [1] 16
```

4) Criando Data frame

- Dado os data frame df.A e df.B e o vetor var3

$$\begin{cases} \text{df.A} <- \text{data.frame}(\text{var1} = 1:5, \text{var2} = \text{LETTERS}[1:5]) \\ \text{df.B} <- \text{data.frame}(\text{var1} = 6:10, \text{var2} = \text{LETTERS}[6:10]) \\ \text{var3} <- 11:20 \end{cases}$$

- a) Criando o data frame df.res a partir da vinculação por linhas de df.A e df.B:

```
df.A <- data.frame(var1 = 1:5, var2 = LETTERS[1:5])
df.B <- data.frame(var1 = 6:10, var2 = LETTERS[6:10])

df.res <- rbind(df.A, df.B)
head(df.res)
```

```
##   var1 var2
## 1    1   A
## 2    2   B
## 3    3   C
## 4    4   D
## 5    5   E
## 6    6   F
```

- b) Apresentando a classe de cada coluna:

```
classes <- sapply(df.res, class)
classes
```

```
##      var1      var2
## "integer" "character"
```


- c) Construindo o data frame `df.A2` e vinculando por linhas com `df.B`

```
df.A2 <- data.frame(var1 = 1:6, var2 = LETTERS[1:6])

df.res2 <- rbind(df.A2, df.B)
tail(df.res2)
```

```
##   var1 var2
## 6     6    F
## 7     6    F
## 8     7    G
## 9     8    H
## 10    9    I
## 11   10    J
```

Observação: Houve duplicação de valores devido a utilização da função `rbind()` que apenas combina os data frames por linhas e não remove as linhas duplicadas. Fato ocorreu devido ao data frame `df.A2` possuir 6 linhas e o outro data frame (`df.B`) possuir apenas 5 linhas.

- d) Adicionando o vetor `var3` como uma nova variável de `df.res`

```
var3 <- 11:20

df.res <- cbind(df.res, var3)
head(df.res)
```

```
##   var1 var2 var3
## 1     1    A   11
## 2     2    B   12
## 3     3    C   13
## 4     4    D   14
## 5     5    E   15
## 6     6    F   16
```

- e) Rearranjando as colunas nesse data frame (`df.res`) para que elas fiquem na ordem: `var1`, `var3`, `var2`.

```
df.res <- df.res[, c("var1", "var3", "var2")]
head(df.res)
```

```
##   var1 var3 var2
## 1     1   11    A
## 2     2   12    B
## 3     3   13    C
## 4     4   14    D
## 5     5   15    E
## 6     6   16    F
```

5) Trabalhando com Data Frame

- Considerando o conjunto de dados `state.x77{datasets}`.
- a) Verificando se o objeto `state.x77` é um data frame, caso contrário será transformado em um.

```
if(!is.data.frame(state.x77)){  
  state.x77 <- as.data.frame(state.x77)  
}
```

```
head(state.x77)
```

##	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
## Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708
## Alaska	365	6315	1.5	69.31	11.3	66.7	152	566432
## Arizona	2212	4530	1.8	70.55	7.8	58.1	15	113417
## Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945
## California	21198	5114	1.1	71.71	10.3	62.6	20	156361
## Colorado	2541	4884	0.7	72.06	6.8	63.9	166	103766

- b) Determinando quantos estados têm renda inferior a 4300.

```
estados <- sum(state.x77$Income < 4300)  
estados
```

```
## [1] 20
```

- c) Determinando qual é o estado com maior renda.

```
indice <- which.max(state.x77$Income)  
estado_maior_renda <- rownames(state.x77)[indice]  
estado_maior_renda
```

```
## [1] "Alaska"
```

- d) Construindo data frame state.

```
state_abb <- state.abb
state_area <- state.area
state_division <- state.division
state_name <- state.name
state_region <- state.region

state <- data.frame(Abb = state_abb, Area = state_area, Division = state_division,
                    Name = state_name, Region = state_region, row.names = state_name)

head(state)
```

```
##           Abb   Area      Division      Name Region
## Alabama    AL  51609 East South Central  Alabama  South
## Alaska     AK  589757      Pacific    Alaska   West
## Arizona    AZ  113909      Mountain   Arizona   West
## Arkansas   AR  53104 West South Central  Arkansas  South
## California CA  158693      Pacific California West
## Colorado   CO  104247      Mountain   Colorado  West
```

- e) Construindo o data frame estados.

```
state.x77 <- state.x77[order(row.names(state.x77)), ]

state <- state[order(row.names(state)), ]

estados <- cbind(state, state.x77)
head(estados)
```

```
##           Abb   Area      Division      Name Region Population Income
## Alabama    AL  51609 East South Central  Alabama  South      3615    3624
## Alaska     AK  589757      Pacific    Alaska   West        365    6315
## Arizona    AZ  113909      Mountain   Arizona   West      2212    4530
## Arkansas   AR  53104 West South Central  Arkansas  South      2110    3378
## California CA  158693      Pacific California West     21198    5114
## Colorado   CO  104247      Mountain   Colorado  West      2541    4884
##           Illiteracy Life Exp Murder HS Grad Frost Area
## Alabama           2.1   69.05   15.1   41.3   20  50708
## Alaska            1.5   69.31   11.3   66.7  152 566432
## Arizona           1.8   70.55    7.8   58.1   15 113417
## Arkansas          1.9   70.66   10.1   39.9   65  51945
## California        1.1   71.71   10.3   62.6   20 156361
## Colorado          0.7   72.06    6.8   63.9  166 103766
```

- f) Renomeando as variáveis.

```
colnames(estados) <- c("Abb", "Area", "Div", "Name", "Reg",
                      substr(names(state.x77), 1, 3))
head(estados)
```

```
##      Abb  Area      Div      Name  Reg  Pop  Inc Ill  Lif
## Alabama  AL  51609 East South Central  Alabama South  3615 3624 2.1 69.05
## Alaska   AK  589757      Pacific    Alaska  West   365 6315 1.5 69.31
## Arizona  AZ  113909      Mountain  Arizona  West  2212 4530 1.8 70.55
## Arkansas AR  53104 West South Central  Arkansas South  2110 3378 1.9 70.66
## California CA 158693      Pacific California West 21198 5114 1.1 71.71
## Colorado CO 104247      Mountain  Colorado  West  2541 4884 0.7 72.06
##
##      Mur  HS  Fro  Are
## Alabama 15.1 41.3  20 50708
## Alaska  11.3 66.7 152 566432
## Arizona  7.8 58.1  15 113417
## Arkansas 10.1 39.9  65 51945
## California 10.3 62.6  20 156361
## Colorado  6.8 63.9 166 103766
```

- g) Removendo a variável div.

```
estados <- subset(estados, select = -Div)
head(estados)
```

```
##      Abb  Area      Name  Reg  Pop  Inc Ill  Lif  Mur  HS  Fro
## Alabama  AL  51609  Alabama South  3615 3624 2.1 69.05 15.1 41.3  20
## Alaska   AK  589757  Alaska  West   365 6315 1.5 69.31 11.3 66.7 152
## Arizona  AZ  113909  Arizona  West  2212 4530 1.8 70.55  7.8 58.1  15
## Arkansas AR  53104  Arkansas South  2110 3378 1.9 70.66 10.1 39.9  65
## California CA 158693 California West 21198 5114 1.1 71.71 10.3 62.6  20
## Colorado CO 104247  Colorado  West  2541 4884 0.7 72.06  6.8 63.9 166
##
##      Are
## Alabama 50708
## Alaska  566432
## Arizona 113417
## Arkansas 51945
## California 156361
## Colorado 103766
```

- h) Construindo o subconjunto `estados.a`.

```
colnames(estados) <- trimws(colnames(estados)) # Tirando o espaço de HS
estados.a <- subset(estados, select = -c(Lif, HS, Fro, Abb, Are))
head(estados.a)
```

```
##           Area      Name  Reg   Pop  Inc Ill  Mur
## Alabama    51609    Alabama South 3615 3624 2.1 15.1
## Alaska    589757    Alaska  West   365 6315 1.5 11.3
## Arizona   113909    Arizona West  2212 4530 1.8  7.8
## Arkansas   53104    Arkansas South 2110 3378 1.9 10.1
## California 158693 California West 21198 5114 1.1 10.3
## Colorado  104247    Colorado West  2541 4884 0.7  6.8
```

- i) Categorizando a variável `Illiteracy` e adicionando ao data frame `estados`.

```
intervalos <- c(0.0, 1.0, 2.0, Inf)
estados$Illiteracy_Level <- cut(estados$Ill, breaks = intervalos,
                                labels = c("baixo", "algum", "alto"), right = TRUE)

estados$Illiteracy_Level <- factor(estados$Illiteracy_Level,
                                   levels = c("baixo", "algum", "alto"))
head(estados)
```

```
##           Abb  Area      Name  Reg   Pop  Inc Ill  Lif  Mur  HS Fro
## Alabama    AL  51609    Alabama South 3615 3624 2.1 69.05 15.1 41.3 20
## Alaska    AK 589757    Alaska  West   365 6315 1.5 69.31 11.3 66.7 152
## Arizona   AZ 113909    Arizona West  2212 4530 1.8 70.55  7.8 58.1  15
## Arkansas  AR  53104    Arkansas South 2110 3378 1.9 70.66 10.1 39.9  65
## California CA 158693 California West 21198 5114 1.1 71.71 10.3 62.6  20
## Colorado  CO 104247    Colorado West  2541 4884 0.7 72.06  6.8 63.9 166
##           Are Illiteracy_Level
## Alabama    50708          alto
## Alaska    566432          algum
## Arizona   113417          algum
## Arkansas   51945          algum
## California 156361          algum
## Colorado  103766          baixo
```

- j) Determinando o estado com maior renda na região oeste e que possui baixo analfabetismo.

```
estados_oeste <- subset(estados, Reg == "West" & Illiteracy_Level == "baixo")
estado_maior_renda <- estados_oeste[which.max(estados_oeste$Inc), ]

(nome_renda <- estado_maior_renda$Name)
```

```
## [1] "Nevada"
```

```
(renda_maxima <- estado_maior_renda$Inc)
```

```
## [1] 5149
```

6) Trabalhando com o array Titanic{datasets}

- a) Determinando número total de passageiros com informações, número de sobreviventes, número de não sobreviventes e a taxa de sobrevivência global (%):

```
data("Titanic")
```

```
(total_passageiros <- sum(Titanic))
```

```
## [1] 2201
```

```
(sobreviventes <- sum(Titanic[,,"Yes"]))
```

```
## [1] 711
```

```
(nao_sobreviventes <- sum(Titanic[,,"No"]))
```

```
## [1] 1490
```

```
(taxa_sobrevivencia <- sobreviventes / total_passageiros * 100)
```

```
## [1] 32.3035
```

- b) Determinando os passageiros por sexo e por classe de bilhete:

```
(passageiros_genero <- apply(Titanic, c("Sex"), sum))
```

```
##   Male Female  
##  1731    470
```

```
(passageiros_classe <- apply(Titanic, c("Class", "Sex"), sum))
```

```
##           Sex  
## Class  Male Female  
##  1st    180    145  
##  2nd    179    106  
##  3rd    510    196  
##  Crew   862     23
```

- c) Determinando taxa de sobrevivência dos passageiros por sexo:

```
(sobreviventes_sexo <- apply(Titanic[,,"Yes"], "Sex", sum))
```

```
##   Male Female
##   367     344
```

```
(nao_sobreviventes_sexo <- apply(Titanic[,,"No"], "Sex", sum))
```

```
##   Male Female
##  1364     126
```

```
(total_passageiros <- sobreviventes_sexo + nao_sobreviventes_sexo)
```

```
##   Male Female
##  1731     470
```

```
(taxa<- round((sobreviventes_sexo / total_passageiros) * 100, 2))
```

```
##   Male Female
##  21.20    73.19
```

- d) Determinando a distribuição de idade por classe de bilhete, por sexo:

```
titanic_df <- as.data.frame(Titanic)
(distrib_idade_classe_sexo <- aggregate(Freq ~ Class + Sex + Age, data = titanic_df, sum))
```

```
##   Class    Sex   Age Freq
## 1    1st   Male Child    5
## 2    2nd   Male Child   11
## 3    3rd   Male Child   48
## 4   Crew   Male Child    0
## 5    1st Female Child    1
## 6    2nd Female Child   13
## 7    3rd Female Child   31
## 8   Crew Female Child    0
## 9    1st   Male Adult  175
## 10   2nd   Male Adult  168
## 11   3rd   Male Adult  462
## 12  Crew   Male Adult  862
## 13   1st Female Adult  144
## 14   2nd Female Adult   93
## 15   3rd Female Adult  165
## 16  Crew Female Adult   23
```

Observação: A distribuição de idade por classe de bilhete e sexo revela que há uma variação significativa no número de adultos e crianças, dependendo da classe e do sexo, o que pode refletir as práticas de embarque e a composição demográfica dos passageiros no Titanic. Além disso, em geral, a classe “1st” tende a ter uma proporção maior de adultos em comparação com as classes “2nd” e “3rd”, onde a presença de crianças é mais significativa. Pode ser observada também que existe uma discrepância na distribuição de idade entre homens e mulheres. Entre as mulheres, a proporção de crianças é relativamente maior, especialmente nas classes “1st” e “2nd”, por outro lado, entre os homens, a presença de crianças é menos proeminente, com uma distribuição de idade mais inclinada para adultos.

- e) Análise em relação a chance de sobrevivência de acordo com o sexo, a classe do bilhete e a idade(adulto ou criança):

```
distrib_idade_classe_sexo_surv <- aggregate(Freq ~ Class + Sex + Age + Survived,
                                             data = titanic_df, sum)

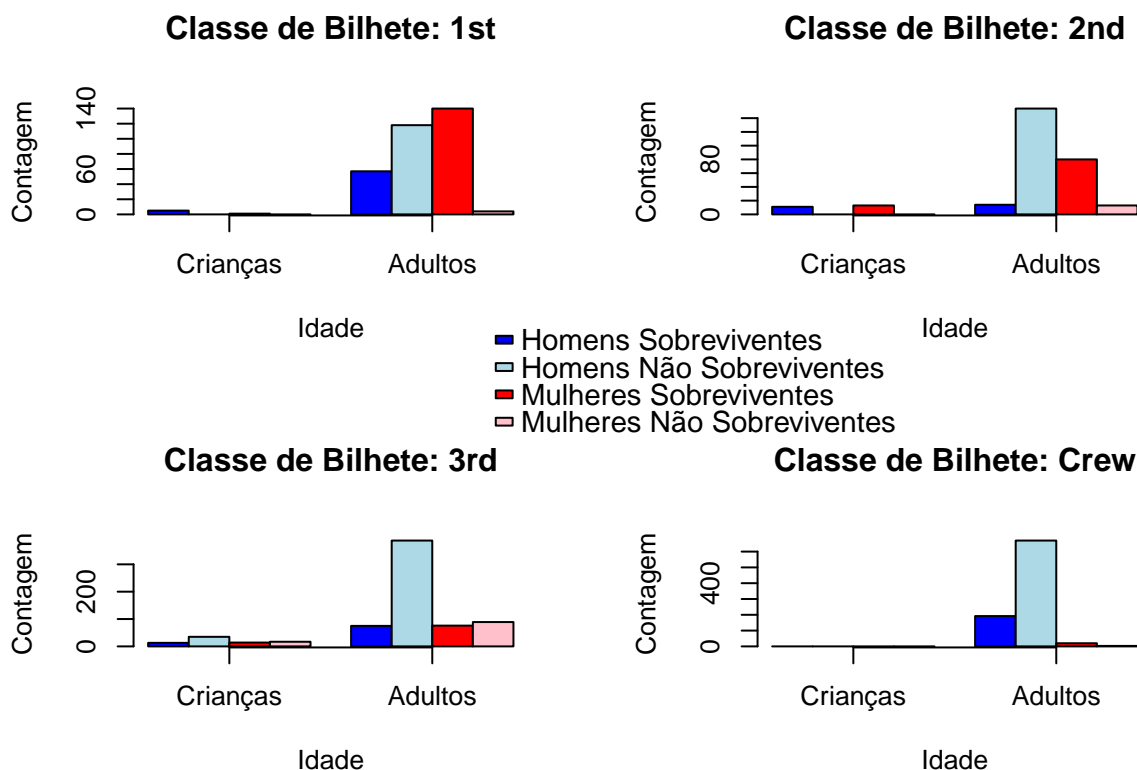
par(mfrow = c(2, 2), mar = c(5, 5, 5, 2) + 0.1)

classes <- unique(distrib_idade_classe_sexo_surv$Class)

for (classe in classes) {
  subset_data <- subset(distrib_idade_classe_sexo_surv, Class == classe)
  male_survived <- subset(subset_data, Sex == "Male" & Survived == "Yes")$Freq
  male_not_survived <- subset(subset_data, Sex == "Male" & Survived == "No")$Freq
  female_survived <- subset(subset_data, Sex == "Female" & Survived == "Yes")$Freq
  female_not_survived <- subset(subset_data, Sex == "Female" & Survived == "No")$Freq
  bp <- barplot(
    height = rbind(male_survived, male_not_survived, female_survived, female_not_survived),
    beside = TRUE,
    col = c("blue", "lightblue", "red", "pink"),
    main = paste("Classe de Bilhete:", classe),
    xlab = "Idade",
    ylab = "Contagem"
  )
  axis(1, at = colMeans(bp), labels = c("Crianças", "Adultos"))
}

par(mfrow = c(1, 1))
legend("center", legend = c("Homens Sobreviventes", "Homens Não Sobreviventes",
  "Mulheres Sobreviventes", "Mulheres Não Sobreviventes"), y.intersp = 1.4, x.intersp = 0.4,
  fill = c("blue", "lightblue", "red", "pink"), bty = "n", inset = c(1, 1), cex = 0.9)
title(main = "Contagem de Sobreviventes e Não Sobreviventes no Titanic", line = 4.4)
```


Contagem de Sobreviventes e Não Sobreviventes no Titanic



Observação: A análise dos gráficos revela que três fatores principais - sexo, classe do bilhete e idade - tiveram um impacto significativo nas chances de sobrevivência dos passageiros do Titanic.

Sexo: A maioria das mulheres sobreviveu ao desastre, enquanto a maioria dos homens não. Isso é um reflexo da política de “mulheres e crianças primeiro” adotada durante a evacuação do navio.

Classe do Bilhete: Os passageiros da primeira classe tiveram a maior taxa de sobrevivência, seguidos pelos da segunda classe. Infelizmente, os passageiros da terceira classe tiveram a menor taxa de sobrevivência. Este padrão evidencia as desigualdades sociais existentes na época e o acesso diferenciado aos recursos de sobrevivência no navio.

Idade: As crianças tiveram uma taxa de sobrevivência relativamente alta, reforçando a prioridade dada a elas durante a evacuação. Entre os adultos, a taxa de sobrevivência variou, sendo em geral menor que a das crianças.

Esses resultados demonstram claramente como fatores sociais e demográficos influenciaram as chances de sobrevivência dos passageiros do Titanic.