

Exercícios R-3 (3.5)

1: Crie o seguinte Data Frame

```
df <- data.frame(letras=letters[1:10], numeros=21:30,  
valores=rnorm(10))
```

```
In [1]: df <- data.frame(letras=letters[1:10], numeros = 21:30, valores = rnorm(10))  
df
```

letras	numeros	valores
a	21	0.6103636
b	22	0.8341710
c	23	1.1485543
d	24	1.3460415
e	25	-1.1520862
f	26	-1.5482375
g	27	-0.7731389
h	28	0.4416557
i	29	-0.2946660
j	30	-0.6180128

Faça as seguintes pesquisas

a) Retorne a linha 5

```
In [2]: df[5,]
```

	letras	numeros	valores
5	e	25	-1.152086

b) Retorna a coluna 2 (como vetor e como data frame - drop=FALSE)

```
In [3]: df[,2]
```

- 1. 21
- 2. 22
- 3. 23
- 4. 24
- 5. 25
- 6. 26
- 7. 27
- 8. 28
- 9. 29
- 10. 30

In [5]: `df[,2, drop=FALSE]`

numeros
21
22
23
24
25
26
27
28
29
30

c) Retorne as colunas 2 e 3

In [6]: `df[,2:3]`

numeros	valores
21	0.6103636
22	0.8341710
23	1.1485543
24	1.3460415
25	-1.1520862
26	-1.5482375
27	-0.7731389
28	0.4416557
29	-0.2946660
30	-0.6180128

d) Retorne os elementos da linha 6, mas somente as colunas 1 e 3

```
In [7]: df[6,c(1,3)]
```

	letras	valores
6	f	-1.548237

e) Retorne os elementos que possuem na coluna "valores" um valor maior que zero

```
In [9]: df[df$valores>0,]
```

	letras	numeros	valores
1	a	21	0.6103636
2	b	22	0.8341710
3	c	23	1.1485543
4	d	24	1.3460415
8	h	28	0.4416557

f) Retorne os elementos que possuem na coluna "numeros" um valor ímpar

```
In [11]: df[df$numero%%2 == 1,]
```

	letras	numeros	valores
1	a	21	0.6103636
3	c	23	1.1485543
5	e	25	-1.1520862
7	g	27	-0.7731389
9	i	29	-0.2946660

g) Retorne os elementos que possuem na coluna "valores" um valor maior que zero e na coluna "numeros" um valor par

```
In [12]: df[df$valores>0 & df$numeros%%2 ==0,]
```

	letras	numeros	valores
2	b	22	0.8341710
4	d	24	1.3460415
8	h	28	0.4416557

h) Retorne os elementos que possuem na coluna letras os seguintes valores "b", "g", "h"

```
In [22]: df[df$letras %in% c("b", "g", "h"),]
```

	letras	numeros	valores
2	b	22	0.8341710
7	g	27	-0.7731389
8	h	28	0.4416557

2: Criar os data frames df1 e cidades

```
df1 <- data.frame(nome=c("Razer", "Anthom", "Nizer", "Rojas",
"Montaño"), cidadeId=c(3, 10, 2, 3, 1))
```

```
cidades <- data.frame(cidadeId=c(1, 2, 3, 4), cidade=c("Curitiba",
"SJP", "Pinhais", "Colombo"))
```

```
In [27]: df1 <- data.frame(nome=c("Razer", "Anthom", "Nizer", "Rojas", "Montaño"), cidadeId
cidades <- data.frame(cidadeId=c(1, 2, 3, 4), cidade=c("Curitiba", "SJP", "Pinhais",
```

```
In [28]: df1
```

nome	cidadeId
Razer	3
Anthom	10
Nizer	2
Rojas	3
Montaño	1

```
In [29]: cidades
```

cidadeId	cidade
1	Curitiba
2	SJP
3	Pinhais
4	Colombo

Executar os merges apresentados

a) Cross Join

```
In [30]: merge(df1, cidades, by=NULL)
```

	nome	cidadeId.x	cidadeId.y	cidade
	Razer	3	1	Curitiba
	Anthom	10	1	Curitiba
	Nizer	2	1	Curitiba
	Rojas	3	1	Curitiba
	Montaño	1	1	Curitiba
	Razer	3	2	SJP
	Anthom	10	2	SJP
	Nizer	2	2	SJP
	Rojas	3	2	SJP
	Montaño	1	2	SJP
	Razer	3	3	Pinhais
	Anthom	10	3	Pinhais
	Nizer	2	3	Pinhais
	Rojas	3	3	Pinhais
	Montaño	1	3	Pinhais
	Razer	3	4	Colombo
	Anthom	10	4	Colombo
	Nizer	2	4	Colombo
	Rojas	3	4	Colombo
	Montaño	1	4	Colombo

b) Inner Join

```
In [32]: merge(df1, cidades, by="cidadeId")
```

	cidadeId	nome	cidade
	1	Montaño	Curitiba
	2	Nizer	SJP
	3	Razer	Pinhais
	3	Rojas	Pinhais

c) Outer Join

```
In [33]: merge(df1, cidades, by="cidadeId", all=TRUE)
```

cidadeId	nome	cidade
1	Montaño	Curitiba
2	Nizer	SJP
3	Razer	Pinhais
3	Rojas	Pinhais
4	NA	Colombo
10	Anthom	NA

d) Left Outer Join

```
In [34]: merge(df1, cidades, by="cidadeId", all.x=TRUE)
```

cidadeId	nome	cidade
1	Montaño	Curitiba
2	Nizer	SJP
3	Razer	Pinhais
3	Rojas	Pinhais
10	Anthom	NA

e) Right Outer Join

```
In [35]: merge(df1, cidades, by="cidadeId", all.y=TRUE)
```

cidadeId	nome	cidade
1	Montaño	Curitiba
2	Nizer	SJP
3	Razer	Pinhais
3	Rojas	Pinhais
4	NA	Colombo

3: Crie o data frame

```
n <- 10
sexo <- sample( c("masculino", "feminino"), n, replace=TRUE)
idade <- sample( 14:102, n, replace=TRUE)
peso <- sample( 50:90, n, replace=TRUE)
menor <- idade<18
pessoas <- data.frame(sexo=sexo, idade=idade, peso=peso,
menor=menor)
```

```
In [36]: n <- 10
sexo <- sample(c("masculino", "feminino"), n, replace=TRUE)
idade <- sample(14:102, n, replace=TRUE)
peso <- sample(50:90, n, replace=TRUE)
menor <- idade<18
```

```

pessoas <- data.frame(sexo=sexo, idade=idade, peso=peso, menor=menor)
pessoas

```

	sexo	idade	peso	menor
	feminino	68	59	FALSE
	feminino	23	70	FALSE
	masculino	15	57	TRUE
	feminino	35	64	FALSE
	masculino	65	65	FALSE
	feminino	58	76	FALSE
	feminino	14	82	TRUE
	masculino	43	66	FALSE
	feminino	82	74	FALSE
	masculino	17	67	TRUE

a) ordene o data frame por peso

```

In [37]: pessoas[order(pessoas$peso),]

```

	sexo	idade	peso	menor
3	masculino	15	57	TRUE
1	feminino	68	59	FALSE
4	feminino	35	64	FALSE
5	masculino	65	65	FALSE
8	masculino	43	66	FALSE
10	masculino	17	67	TRUE
2	feminino	23	70	FALSE
9	feminino	82	74	FALSE
6	feminino	58	76	FALSE
7	feminino	14	82	TRUE

b) Ordene o data frame por sexo e peso, decrescente

```

In [40]: pessoas[order(pessoas$sexo, -pessoas$peso),]

```

	sexo	idade	peso	menor
7	feminino	14	82	TRUE
6	feminino	58	76	FALSE
9	feminino	82	74	FALSE
2	feminino	23	70	FALSE
4	feminino	35	64	FALSE
1	feminino	68	59	FALSE
10	masculino	17	67	TRUE
8	masculino	43	66	FALSE
5	masculino	65	65	FALSE
3	masculino	15	57	TRUE

c) Dê a maior idade dos dados (max)

```
In [67]: max(pessoas$idade)
```

82

d) Dê a média dos pesos

```
In [70]: mean(pessoas$peso)
```

68

e) Mostrar as pessoas do sexo feminino que estão na base

```
In [71]: pessoas[pessoas$sexo %in% "feminino",]
```

	sexo	idade	peso	menor
1	feminino	68	59	FALSE
2	feminino	23	70	FALSE
4	feminino	35	64	FALSE
6	feminino	58	76	FALSE
7	feminino	14	82	TRUE
9	feminino	82	74	FALSE

f) Contar as pessoas do sexo feminino (nrow)

```
In [74]: nrow(pessoas[pessoas$sexo %in% "feminino",])
```

6