

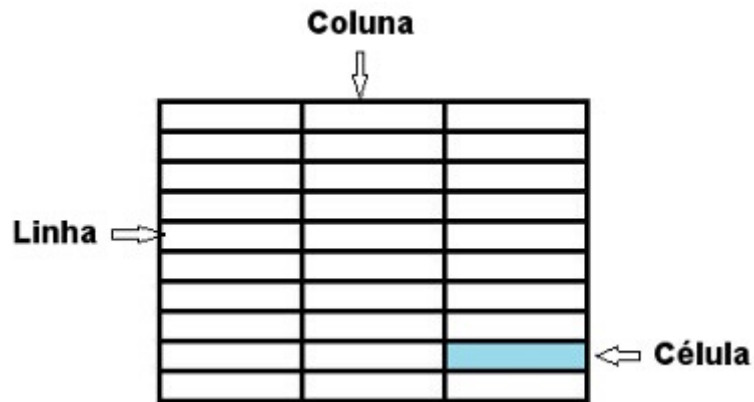
Trabalhando com Coleções de Dados no R Vetores, Matrizes, Listas, Factors e Dataframes

Já vimos como trabalhar com diversos tipos de variáveis, characters, numeric, logical. Na maioria das vezes trabalhamos com apenas um valor.

Mas na análise de dados é necessário trabalhar com conjuntos de vários valores, que podemos chamar de coleções. Essas coleções podem ser organizadas em estruturas diferentes que são os Vetores, Matrizes, Listas, Factors e Dataframes.

Nesta aula vamos estudar a fundo cada uma dessas estruturas.

A coleção de dados mais simples é o vetor que possui a característica de ser uma sequência unidimensional de dados veja a seguinte figura



Nesta figura vemos que se trata de uma **tabela**, podemos observar que esta tabela possui **10 Linhas e 3 Colunas**

- Cada linha é um vetor com 3 elementos e
- Cada coluna é um vetor com 10 elementos

Colunas			
Linhas	L 1 - C1	L 1 - C2	L 1 - C3
	L 2 - C1	L 2 - C2	L 2 - C3
	L 3 - C1	L 3 - C2	L 3 - C3
	L 4 - C1	L 4 - C2	L 4 - C3
	L 5 - C1	L 5 - C2	L 5 - C3
	L 6 - C1	L 6 - C2	L 6 - C3
	L 7 - C1	L 7 - C2	L 7 - C3
	L 8 - C1	L 8 - C2	L 8 - C3
	L 9 - C1	L 9 - C2	L 9 - C3
	L 10 - C1	L 10 - C2	L 10 - C3

Portando podemos tirar das linhas os seguintes vetores

L 2 - C1 L 2 - C2 L 2 - C3

L 3 - C1 L 3 - C2 L 3 - C3

L 4 - C1 L 4 - C2 L 4 - C3

L 5 - C1 L 5 - C2 L 5 - C3

L 6 - C1 L 6 - C2 L 6 - C3

L 7 - C1 L 7 - C2 L 7 - C3

L 8 - C1 L 8 - C2 L 8 - C3

L 9 - C1 L 9 - C2 L 9 - C3

L 10 - C1 L 10 - C2 L 10 - C3

E das colunas saem os seguintes vetores

Colunas
L 1 - C1
L 2 - C1
L 3 - C1
L 4 - C1
L 5 - C1
L 6 - C1
L 7 - C1
L 8 - C1
L 9 - C1
L 10 - C1

E das colunas saem os seguintes vetores

<u>L 1 - C2</u>
<u>L 2 - C2</u>
<u>L 3 - C2</u>
<u>L 4 - C2</u>
<u>L 5 - C2</u>
<u>L 6 - C2</u>
<u>L 7 - C2</u>
<u>L 8 - C2</u>
<u>L 9 - C2</u>
<u>L 10 - C2</u>

E das colunas saem os seguintes vetores

<u>L 1 - C3</u>
<u>L 2 - C3</u>
<u>L 3 - C3</u>
<u>L 4 - C3</u>
<u>L 5 - C3</u>
<u>L 6 - C3</u>
<u>L 7 - C3</u>
<u>L 8 - C3</u>
<u>L 9 - C3</u>
<u>L 10 - C3</u>

Criação de vetores

O elemento mais básico em R é o vetor. Veremos agora como criá-los e acessar seus elementos (ou seja, subconjuntos). Aqui estão três maneiras de criar o mesmo vetor arbitrário:

Criar um vetor manualmente

```
In [3]: c(10,11,12,13,14,15,16,17,18,19,20,21) # create a vector manually
```

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

Criar um vetor usando o operador :

In [6]:

```
10:21
```

```
10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21
```

Criar um vetor com a função seq() usando "by"

In [7]: `seq(from=10, to=21, by=1)`

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

In [8]: `seq(from=10, to=21, by=3)`

10 · 13 · 16 · 19

In [9]: `seq(from=10, to=21, by=10)`

10 · 20

Criar um vetor com a função seq() usando "len" (comprimento - length)

```
In [10]: seq(from=10, to=21, len=12)
```

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

```
In [11]: seq(from=10, to=21, len=3)
```

10 · 15.5 · 21

```
In [12]: seq(from=10, to=21, len=2)
```

10 · 21

Atribuindo um vetor a uma variável x

```
In [13]: x <- 10:21  
x
```

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

Operações com vetores, normalmente trabalhando com cada um dos elementos

In [14]: `x+2` *# somar 2 a cada um dos elementos de x*

12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21 · 22 · 23

In [15]: `x*2` *# multiplicando cada um dos elementos de `x`*
`x`

20 · 22 · 24 · 26 · 28 · 30 · 32 · 34 · 36 · 38 · 40 · 42

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

In [16]: `x^2` *#elevando ao quadrado cada elemento de `x`*
`x`

100 · 121 · 144 · 169 · 196 · 225 · 256 · 289 · 324 · 361 · 400 · 441

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

```
In [17]: sqrt(x)                                # raiz quadrada de cada elemento de `x`  
x
```

```
3.16227766016838 · 3.3166247903554 · 3.46410161513775 ·  
3.60555127546399 · 3.74165738677394 · 3.87298334620742 · 4 ·  
4.12310562561766 · 4.24264068711928 · 4.35889894354067 ·  
4.47213595499958 · 4.58257569495584
```

```
10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21
```

In [18]: `log(x)`
x

Log cada elemento de `x`

2.30258509299405 · 2.39789527279837 · 2.484906649788 ·
2.56494935746154 · 2.63905732961526 · 2.70805020110221 ·
2.77258872223978 · 2.83321334405622 · 2.89037175789616 ·
2.94443897916644 · 2.99573227355399 · 3.04452243772342

10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21

Tipos de vetores

```
In [19]: x <- c(0.5, 0.6)           # numeric  
x
```

0.5 · 0.6

```
In [20]: x <- c(TRUE, FALSE)       # logical  
x
```

TRUE · FALSE

```
In [21]: x <- c(T, F)              # logical  
x
```

TRUE · FALSE

```
In [22]: x <- c("a", "b", "c")     # character  
x
```

'a' · 'b' · 'c'

```
In [23]: x <- 9:29                  # integer  
x
```

9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21 · 22 · 23 ·
24 · 25 · 26 · 27 · 28 · 29

```
In [24]: x <- vector("numeric", length = 10)    # usando a função `vector()` para inicializar vetores.  
x
```

0. 0. 0. 0. 0. 0. 0. 0. 0. 0

A Função `typeof()`

Se você precisar saber qual o tipo de dados de um vetor use a função `typeof()`. Vamos ver como ela funciona, primeiro vamos criar três vetores, `y1`, `y2` e `y3`

```
In [29]: y1 <- c(1.7, "a")           # character  
          y2 <- c(TRUE, 2)          # numeric  
          y3 <- c("a", TRUE)        # character
```

```
In [30]: typeof(y1) # função para checar qualquer vetor ou variável
```

'character'

```
In [31]: typeof(y2) # função para checar qualquer vetor ou variável
```

'double'

```
In [34]: typeof(y3) # função para checar qualquer vetor ou variável
```

'character'

Você também pode usar a função `class()`

```
In [38]: class(y1)                                     # função para checar qualquer vetor ou variável
```

'character'

```
In [39]: class(y2)                                     # função para checar qualquer vetor ou variável
```

'numeric'

```
In [40]: class(y3)                                     # função para checar qualquer vetor ou variável
```

'character'

Convertendo os tipo de um vetor

`x <- c("a", "b", "c", "1")` atribui valore a `x`

`as.numeric(x)` converte `x` como *numeric*

`as.logical(x)` converte `x` como *logical*

`as.complex(x)` converte `x` como *character*


```
In [41]: x <- c("a", "b", "c", "1")  
x
```

'a' · 'b' · 'c' · '1'

```
In [42]: as.numeric(x)  
x
```

```
Warning message in eval(expr, envir, enclos):  
"NAs introduzidos por coerção"
```

```
<NA> . <NA> . <NA> . 1
```

```
'a' . 'b' . 'c' . '1'
```

```
In [43]: as.logical(x)
```

<NA> · <NA> · <NA> · <NA>

'a' · 'b' · 'c' · '1'

```
In [44]: as.character(x)  
x
```

'a' · 'b' · 'c' · '1'

'a' · 'b' · 'c' · '1'

```
In [46]: as.character(1:10)
```

```
'1' '2' '3' '4' '5' '6' '7' '8' '9' '10'
```

Acesso aos elemento de um vetor usando indices

In [47]:

```
x
```

```
'a' · 'b' · 'c' · '1'
```

```
In [48]: x[1] # acessa o primeiro elemneto do vetor x
```

```
'a'
```



```
In [49]: x[2] # acessa o segundo elemneto do vetor x
```

'b'

```
In [50]: x[3] # acessa o terceiro elemneto do vetor x
```

```
'c'
```

```
In [51]: x[4]# acessa o quarto elemneto do vetor x
```

```
'1'
```

Funções para sumarizar dados

Temos várias funções para sumarizar os dados de um vetor que são :

- `length(vector)`
- `sum(vector)`
- `mean(vector)`
- `max(vector)`
- `min(vector)`
- `median(vector)`
- `summary(vector)`

In [52]:

```
x<-1:21
```

```
In [53]: length(x) # retorna a quantidade de elementos (comprimento)
```

21

```
In [54]: sum(x) # retorna a soma dos elementos
```

231

```
In [55]: max(x) # retorna o máximo dos elementos
```

21


```
In [56]: min(x) # retorna o mínimo dos elementos
```

1

```
In [57]: mean(x) # retorna a média dos elementos
```

11

```
In [58]: median(x) # retorna a mediana dos elementos
```

11

```
In [59]: summary(x) # retorna um sumário dos elementos
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	6	11	11	16	21

Exemplo

Imagine que você fez uma compra e deseja armazenar na memória dado sobre esta compra a primeira coisa a fazer é a atribuição:

```
In [61]: ShampooMarca <- 'Lavebem'  
ShampooPreco <- 20.0  
ShampooQtde <- 2.0  
ShampooTotal <- ShampooPreco*ShampooQtde
```

É interessante guardar todos estes dados em um vetor, assim podemos consultar tudo em um só lugar

```
In [62]: Shampoo <- c(ShampooMarca, ShampooPreco, ShampooQtde, ShampooTotal)  
Shampoo
```

```
'Lavebem' · '20' · '2' · '40'
```

In [63]: Shampoo[1]

'Lavebem'

In [64]: Shampoo[2]

'20'

In [65]: Shampoo[3]

'2'

In [66]: Shampoo[4]

'40'

E se você precisar trabalhar com vários vetores ?

```
In [67]: id      <- c("N198", "N805", "N333", "N117", "N195", "N298")
          genero <- c(1, 0, 1, 1, 0, 1) # 0 significa masculino, 1 significa feminino
          idade  <- c(30, 60, 26, 75, 19, 60)
          sangue <- c(0.4, 0.2, 0.6, 0.2, 0.8, 0.1)
```

Os vetores podem ser organizados em linhas ou colunas para formar uma estrutura semelhante a uma matriz (um retângulo de dados).

Podemos combiná-los usando as funções `cbind` ou `rbind` que se traduzem em ligar os vetores juntos como colunas ou linhas, respectivamente.

Podemos atribuir as colunas ou linhas combinadas a um novo objeto:

```
health_data <- cbind (id, genero, idade, sangue)
health_data
```

- Quando o R leu esses códigos primeiro executa a função `cbind` para ligar vetores juntos em seguida, ele atribui o resultado a um novo objeto `health_data`
- Pense em "cbind" como uma abreviação de 'vinculação de coluna', isto é, combinando os objetos como colunas


```
In [68]: health_data <- cbind(id, genero, idade, sangue)
health_data
```

A matrix: 6 × 4 of type chr

id	genero	idade	sangue
N198	1	30	0.4
N805	0	60	0.2
N333	1	26	0.6
N117	1	75	0.2
N195	0	19	0.8
N298	1	60	0.1

Agora pense em `rbind` como um encurtamento de 'vinculação de linha', ou seja, combinar os objetos como linhas

```
health_data_rbind <- rbind(id, genero, idade, sangue)  
health_data_rbind
```

```
In [69]: health_data_rbind <-rbind (id, genero, idade, sangue)
health_data_rbind
```

A matrix: 4 × 6 of type chr

id	N198	N805	N333	N117	N195	N298
genero	1	0	1	1	0	1
idade	30	60	26	75	19	60
sangue	0.4	0.2	0.6	0.2	0.8	0.1

- Combinar os vetores dessa maneira fornece uma matriz de informações.
- A classe de um objeto pode ser encontrada usando a função `class`.
- Observe que se tentarmos isso em um vetor, a classe depende do tipo de componentes, por exemplo, temos “numeric” para um vetor de números e “character” para um vetor de nomes.
- Da mesma forma que um vetor, as entradas de uma matriz devem ser da mesma classe, portanto, mais uma vez, vemos que os números foram alterados para ficarem entre aspas, pois agora são tratados como texto.

Portanto agora nossos dados de compras podem ser tratados de uma forma diferente

```
In [72]: items <- c(1,2,3,4,5,6)
ItemNome <-c('Shampoo', 'Sabonete', 'Pasta', 'Escova', 'Fio Dental', 'Água')
ItemPreco <-c(20.5,5.5,3.52,7.34,5.23,5.02)
ItemQtde <- c(2,3,4,1,1,3)
ItemTotal <- ItemPreco*ItemQtde
```

In [73]:

```
ItemNome
```

```
'Shampoo' · 'Sabonete' · 'Pasta' · 'Escova' · 'Fio Dental' · 'Água'
```

In [74]:

```
ItemPreco
```

```
20.5 · 5.5 · 3.52 · 7.34 · 5.23 · 5.02
```

In [75]:

```
ItemQtde
```

```
2 · 3 · 4 · 1 · 1 · 3
```

In [76]:

```
ItemTotal
```

```
41 · 16.5 · 14.08 · 7.34 · 5.23 · 15.06
```

Agora podemos juntar tudo com cbind

```
In [77]: Compra <- cbind(ItemNome, ItemPreco, ItemQtde, ItemTotal)
```

```
In [78]: Compra
```

A matrix: 6 × 4 of type chr

ItemNome	ItemPreco	ItemQtde	ItemTotal
Shampoo	20.5	2	41
Sabonete	5.5	3	16.5
Pasta	3.52	4	14.08
Escova	7.34	1	7.34
Fio Dental	5.23	1	5.23
Água	5.02	3	15.06

Agora também podemos juntar tudo com rbind

```
In [79]: Compra_rbind <- rbind(ItemNome, ItemPreco, ItemQtde, ItemTotal)
```

```
In [80]: Compra_rbind
```

A matrix: 4 × 6 of type chr

ItemNome	Shampoo	Sabonete	Pasta	Escova	Fio Dental	Água
ItemPreco	20.5	5.5	3.52	7.34	5.23	5.02
ItemQtde	2	3	4	1	1	3
ItemTotal	41	16.5	14.08	7.34	5.23	15.06

Matrizes

Matrizes são vetores com um atributo de dimensão.

O atributo de dimensão é em si um vetor inteiro de comprimento 2 (número de linhas, número de colunas)

```
In [81]: m <- matrix(nrow = 2, ncol = 3)      # cria uma `NA` matriz bde 2x3
```

In [82]:

```
m
```

```
# print m
```

A matrix: 2×3 of
type lgl

NA	NA	NA
NA	NA	NA

```
In [83]: dim(m) # check dimensões
```

2 · 3

```
In [84]: attributes(m)                                # check dimension
```

```
$dim = 2 3
```

Matrizes também podem ser criadas diretamente de vetores, adicionando um atributo de dimensão.

In [85]:

```
m <- 1:10  
m
```

cria um vetor `m`

1. 2. 3. 4. 5. 6. 7. 8. 9. 10

In [86]: `dim(m) <- c(5, 2)`

transforma vctor `m` em uma matriz de 2x5

In [87]:

```
m
```

A

matrix:

5 × 2 of

type int

1	6
2	7
3	8
4	9
5	10

Matrizes podem ser criadas por ligação de coluna ou ligação de linha com as funções `cbind()` e `rbind()`.

In [88]:

```
x <- 1:3  
y <- 10:12
```

```
# cria um vetor `x`  
# cria um vector `y`
```

In [89]: `cbind(x, y)`

combina vetor `x` e `y` por colunas

A
matrix:
3 × 2 of
type int

x	y
1	10
2	11
3	12

In [90]: `rbind(x, y)`

combina vetor `x` e `y` por linha

A matrix: 2 × 3 of
type int

x	1	2	3
y	10	11	12

Listas

As listas são um tipo especial de vetor que pode conter elementos de diferentes classes.

As listas são um tipo de dados muito importante em R e você deve conhecê-las bem.

Listas, em combinação com as várias funções de “aplicar” discutidas posteriormente, formam uma combinação poderosa. As listas podem ser criadas explicitamente usando a função `list()`, que recebe um número arbitrário de argumentos.

Em geral, você usa duas operações de indexação diferentes em listas:

- Colchetes simples para retornar uma lista de elementos selecionados (`[]`)
- Colchetes duplos para retornar um único elemento (`[[]]`)

```
In [91]: x <- list(1:5, c("a", "b", "c"), TRUE, 7,5) # cria uma lista do vetor `x`  
x
```

```
1.      1. 2. 3. 4. 5  
2.      'a' 'b' 'c'  
3. TRUE  
4. 7  
5. 5
```

In [92]: `x[1] # colchetes simples`

1. 1 · 2 · 3 · 4 · 5

In [93]: `x[[1]]` # *colchetes duplos*

1 · 2 · 3 · 4 · 5


```
In [94]: typeof(x[2]) # verifique o tipo do objeto usando []
```

'list'

```
In [95]: typeof(x[[4]]) # verifique o tipo do objeto usando [][]  
x[[4]]
```

'double'

7

```
In [96]: x[c(2,3,4)] # lista do primeiro e segundo vetores
```

1. 'a' · 'b' · 'c'
2. TRUE
3. 7

Também podemos criar uma lista substituindo um objeto vazio existente, como você pode ver no código a seguir:

```
In [97]: x <- vector("list", length = 3) # cria uma lista vazia de um comprimento pré-especificad  
o  
x
```

1. NULL
2. NULL
3. NULL

```
In [98]: x[1]<- 10 # atribui o valor 10 ao primeiro elemnto da lista
x
```

1. 10
2. NULL
3. NULL

```
In [99]: x[2]<- 'segundo' # atribui o valor 'segundo' ao segundo elemnto da lista
x
```

1. 10
2. 'segundo'
3. NULL

```
In [100]: x[3]<- TRUE # atribui o valor TRUE ao terceiro elemento da lista
x
```

1. 10
2. 'segundo'
3. TRUE

```
In [101]: nome <- c ("a", "b", "c", "d") # cria um objeto como uma variável `nome`
```

```
In [102]: idade <- c (18, 19, 20, 21) # cria um objeto como uma variável `idade`
```

```
In [103]: gênero <- c (1, 0, 0, 1) # cria um objeto como uma variável `gênero`
```

```
In [104]: x [[1]] <- nome # adiciona / substitui objeto à lista `x`
```



```
In [105]: x [[2]] <- idade # adiciona / substitui objeto à lista `x`
```

```
In [106]: x [[3]] <- genero # adiciona / substitui objeto à lista `x`
```

In [107]: `x # imprimir o resultado final`

1. 'a' · 'b' · 'c' · 'd'
2. 18 · 19 · 20 · 21
3. 1 · 0 · 1 · 1 · 0 · 1

Fatores (Factors)

Os fatores são usados para representar dados categóricos e podem ser desordenados ou ordenados.

Pode-se pensar em um fator como um vetor de inteiros em que cada inteiro tem um rótulo.

Fatores são importantes na modelagem estatística e são tratados especialmente por funções de modelagem como `lm()` e `glm()`.

Usar fatores com rótulos é melhor do que usar números inteiros porque os fatores são autodescritivos.

Ter uma variável com os valores “Masculino” e “Feminino” é melhor do que uma variável com os valores 1 e 2.

Objetos fatoriais podem ser criados com a função `factor()`.

```
In [108]: x <- factor(c("yes", "no", "yes", "no")) # cria objetos factor
x
```

yes • no • yes • no

► **Levels:**

```
In [109]: table(x) # tabela of `x`
```

```
x  
no yes  
2    2
```

In [110]: `unclass(x)-1`

veja a representação subjacente do fat

$1 \cdot 0 \cdot 1 \cdot 0$

```
In [111]: attr(x, "levels")
```

veja a representação subjacente do fator

'no' · 'yes'

Data Frames (Frames de dados)

Um Data Frame é uma tabela ou uma estrutura semelhante a uma matriz bidimensional em que cada coluna contém valores de uma variável e cada linha contém um conjunto de valores de cada coluna.

A seguir estão as características de um Data Frame.

- Os nomes das colunas não devem estar vazios.
- Os nomes das linhas devem ser exclusivos.
- Os dados armazenados em um quadro de dados podem ser numéricos, fator ou tipo de caractere.
- Cada coluna deve conter o mesmo número de itens de dados.

```
In [112]: # Criação do primeiro data frame.
df1 <- data.frame(id = c(1:5),
                  name = c("Julian", "Vanessa", "Jeffry", "Angel", "Nikki"),
                  salary = c(623.3, 515.2, 611.0, 729.0, 843.25),
                  start_date = as.Date(c("2022-01-01", "2022-09-23", "2022-11-15",
                                         "2022-05-11", "2022-03-27")),
                  dept = c("DS", "DS", "BA", "DA", "DS"), stringsAsFactors = F)

df1
```

A data.frame: 5 × 5

id	name	salary	start_date	dept
<int>	<chr>	<dbl>	<date>	<chr>
1	Julian	623.30	2022-01-01	DS
2	Vanessa	515.20	2022-09-23	DS
3	Jeffry	611.00	2022-11-15	BA
4	Angel	729.00	2022-05-11	DA
5	Nikki	843.25	2022-03-27	DS

```
In [113]: # Criação do segundo data frame.
df2 <- data.frame(id = c(6:10),
                  name = c("Ardifo", "Irene", "Kefas", "Sherly", "Bakti"),
                  salary = c(578.0, 722.5, 632.8, 632.8, NA),
                  start_date = as.Date(c("2022-05-21", "2022-07-30", "2022-06-17",
                                         "2022-07-30", "2018-09-03")),
                  dept = c("Actuaries", "Actuaries", "CA", "DE", "Lecturer"), stringsAsFactors =
F)
df2
```

A data.frame: 5 × 5

id	name	salary	start_date	dept
<int>	<chr>	<dbl>	<date>	<chr>
6	Ardifo	578.0	2022-05-21	Actuaries
7	Irene	722.5	2022-07-30	Actuaries
8	Kefas	632.8	2022-06-17	CA
9	Sherly	632.8	2022-07-30	DE
10	Bakti	NA	2018-09-03	Lecturer

```
In [114]: df3 <- rbind(df1,df2)                                     # Combinando os dois data frames
df3
```

A data.frame: 10 × 5

id	name	salary	start_date	dept
<int>	<chr>	<dbl>	<date>	<chr>
1	Julian	623.30	2022-01-01	DS
2	Vanessa	515.20	2022-09-23	DS
3	Jeffry	611.00	2022-11-15	BA
4	Angel	729.00	2022-05-11	DA
5	Nikki	843.25	2022-03-27	DS
6	Ardifo	578.00	2022-05-21	Actuaries
7	Irene	722.50	2022-07-30	Actuaries
8	Kefas	632.80	2022-06-17	CA
9	Sherly	632.80	2022-07-30	DE
10	Bakti	NA	2018-09-03	Lecturer

In [115]: `head(df3)`

print as primeiras seis linhas

A data.frame: 6 × 5

	id	name	salary	start_date	dept
	<int>	<chr>	<dbl>	<date>	<chr>
1	1	Julian	623.30	2022-01-01	DS
2	2	Vanessa	515.20	2022-09-23	DS
3	3	Jeffry	611.00	2022-11-15	BA
4	4	Angel	729.00	2022-05-11	DA
5	5	Nikki	843.25	2022-03-27	DS
6	6	Ardifo	578.00	2022-05-21	Actuaries

```
In [116]: head(df3,4)
```

```
# print as primeiras quatro linhas
```

A data.frame: 4 × 5

	id	name	salary	start_date	dept
	<int>	<chr>	<dbl>	<date>	<chr>
1	1	Julian	623.3	2022-01-01	DS
2	2	Vanessa	515.2	2022-09-23	DS
3	3	Jeffry	611.0	2022-11-15	BA
4	4	Angel	729.0	2022-05-11	DA

```
In [117]: class(df3)                                # o tipo do data frame
```

```
'data.frame'
```

```
In [118]: str(df3) # Estrutura do data frame
```

```
'data.frame':  10 obs. of  5 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10
 $ name    : chr  "Julian" "Vanessa" "Jeffry" "Angel" ...
 $ salary  : num  623 515 611 729 843 ...
 $ start_date: Date, format: "2022-01-01" "2022-09-23" ...
 $ dept    : chr  "DS" "DS" "BA" "DA" ...
```


In [119]: `dim(df3)`

Dimensões do data frame

10 5

Os dataframes geralmente são criados pela leitura em um conjunto de dados usando `read.table()` ou `read.csv()`.

No entanto, os dataframes também podem ser criados explicitamente com a função `dataframe()` ou podem ser obtidos de outros tipos de objetos, como listas.

Extração

R fornece muitas maneiras de subconjunto e extrair elementos de vetores e outros objetos.

O básico é bastante simples, mas não prestar atenção à “personalidade” de cada mecanismo de extração pode causar muita dor de cabeça.

Para começar, a extração é feita com o operador `[]`.

O operador pode pegar vetores de vários tipos.

In [120]: mtcars

A data.frame: 32 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6

Maserati Bora	150 mpg	8 cyl	301.0 disp	135 hp	354 drat	3570 wt	14.60 qsec	0s	1 am	5 gear	8 carb
Volvo 142E	21.4 <dbl>	4 <dbl>	121.0 <dbl>	108 <dbl>	4.11 <dbl>	2780 <dbl>	18.60 <dbl>	1 <dbl>	1 <dbl>	4 <dbl>	2 <dbl>

```
In [121]: mtcars [1,5] # extrai o elemento da 1ª linha e 5ª coluna.
```

3.9

```
In [122]: mtcars [1: 5,] # extrai as primeiras seis linhas de mtcars
```

A data.frame: 5 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

```
In [124]: mtcars[, 1: 2] # extrai as primeiras duas colunas de mtcars
```

A data.frame: 32 × 2

	mpg	cyl
	<dbl>	<dbl>
Mazda RX4	21.0	6
Mazda RX4 Wag	21.0	6
Datsun 710	22.8	4
Hornet 4 Drive	21.4	6
Hornet Sportabout	18.7	8
Valiant	18.1	6
Duster 360	14.3	8
Merc 240D	24.4	4
Merc 230	22.8	4
Merc 280	19.2	6
Merc 280C	17.8	6
Merc 450SE	16.4	8
Merc 450SL	17.3	8
Merc 450SLC	15.2	8
Cadillac Fleetwood	10.4	8
Lincoln Continental	10.4	8
Chrysler Imperial	14.7	8
Fiat 128	32.4	4
Honda Civic	30.4	4
Toyota Corolla	33.9	4
Toyota Corona	21.5	4
Dodge Challenger	15.5	8
AMC Javelin	15.2	8
Camaro Z28	13.3	8
Pontiac Firebird	19.2	8
Fiat X1-9	27.3	4
Porsche 914-2	26.0	4
Lotus Europa	30.4	4
Ford Pantera L	15.8	8
Ferrari Dino	19.7	6

Maserati Bora	15.0	8
	mpg	yl
Volvo 142E	24.4	4
	<dbl>	<dbl>

```
In [125]: mtcars[, 'mpg'] # extrair coluna específica
```

```
21. 21. 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8.  
16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5.  
15.2 13.3 19.2 27.3 26. 30.4 15.8 19.7 15. 21.4
```

In [126]: `mtcars$hp # extract Coluna específica`

```
110 · 110 · 93 · 110 · 175 · 105 · 245 · 62 · 95 · 123 · 123 · 180 ·  
180 · 180 · 205 · 215 · 230 · 66 · 52 · 65 · 97 · 150 · 150 · 245 ·  
175 · 66 · 91 · 113 · 264 · 175 · 335 · 109
```

```
In [127]: mtcars ['Mazda RX4',] # extrair linha específica
```

A data.frame: 1 × 11

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4

```
In [128]: subset(mtcars, select = mpg) # extract / subset Coluna específica
```

A data.frame: 32 × 1

	mpg
	<dbl>
Mazda RX4	21.0
Mazda RX4 Wag	21.0
Datsun 710	22.8
Hornet 4 Drive	21.4
Hornet Sportabout	18.7
Valiant	18.1
Duster 360	14.3
Merc 240D	24.4
Merc 230	22.8
Merc 280	19.2
Merc 280C	17.8
Merc 450SE	16.4
Merc 450SL	17.3
Merc 450SLC	15.2
Cadillac Fleetwood	10.4
Lincoln Continental	10.4
Chrysler Imperial	14.7
Fiat 128	32.4
Honda Civic	30.4
Toyota Corolla	33.9
Toyota Corona	21.5
Dodge Challenger	15.5
AMC Javelin	15.2
Camaro Z28	13.3
Pontiac Firebird	19.2
Fiat X1-9	27.3
Porsche 914-2	26.0
Lotus Europa	30.4
Ford Pantera L	15.8
Ferrari Dino	19.7

Maserati Bora	15.0
	mpg
Volvo 142E	24.4
	<dbt>

```
In [129]: subset(mtcars, select = 1) # extract / subset Coluna específica
```

A data.frame: 32 × 1

	mpg
	<dbl>
Mazda RX4	21.0
Mazda RX4 Wag	21.0
Datsun 710	22.8
Hornet 4 Drive	21.4
Hornet Sportabout	18.7
Valiant	18.1
Duster 360	14.3
Merc 240D	24.4
Merc 230	22.8
Merc 280	19.2
Merc 280C	17.8
Merc 450SE	16.4
Merc 450SL	17.3
Merc 450SLC	15.2
Cadillac Fleetwood	10.4
Lincoln Continental	10.4
Chrysler Imperial	14.7
Fiat 128	32.4
Honda Civic	30.4
Toyota Corolla	33.9
Toyota Corona	21.5
Dodge Challenger	15.5
AMC Javelin	15.2
Camaro Z28	13.3
Pontiac Firebird	19.2
Fiat X1-9	27.3
Porsche 914-2	26.0
Lotus Europa	30.4
Ford Pantera L	15.8
Ferrari Dino	19.7

Maserati Bora	15.0
	mpg
Volvo 142E	24.4
	<dbt>


```
In [130]: subset(mtcars, select = c(1,2,3)) # extrair / subconjunto da primeira e segunda coluna
```

A data.frame: 32 × 3

	mpg	cyl	disp
	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160.0
Mazda RX4 Wag	21.0	6	160.0
Datsun 710	22.8	4	108.0
Hornet 4 Drive	21.4	6	258.0
Hornet Sportabout	18.7	8	360.0
Valiant	18.1	6	225.0
Duster 360	14.3	8	360.0
Merc 240D	24.4	4	146.7
Merc 230	22.8	4	140.8
Merc 280	19.2	6	167.6
Merc 280C	17.8	6	167.6
Merc 450SE	16.4	8	275.8
Merc 450SL	17.3	8	275.8
Merc 450SLC	15.2	8	275.8
Cadillac Fleetwood	10.4	8	472.0
Lincoln Continental	10.4	8	460.0
Chrysler Imperial	14.7	8	440.0
Fiat 128	32.4	4	78.7
Honda Civic	30.4	4	75.7
Toyota Corolla	33.9	4	71.1
Toyota Corona	21.5	4	120.1
Dodge Challenger	15.5	8	318.0
AMC Javelin	15.2	8	304.0
Camaro Z28	13.3	8	350.0
Pontiac Firebird	19.2	8	400.0
Fiat X1-9	27.3	4	79.0
Porsche 914-2	26.0	4	120.3
Lotus Europa	30.4	4	95.1
Ford Pantera L	15.8	8	351.0
Ferrari Dino	19.7	6	145.0

Maserati Bora	15.0 mpg	8yl	301.0 dis
Volvo 142E	24.4 <dbl>	4dbl>	121.0 <dbl>

```
In [131]: subset(mtcars, select = c(1: 5)) # extract / subset Columnas específicas
```

A data.frame: 32 × 5

	mpg	cyl	disp	hp	drat
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160.0	110	3.90
Mazda RX4 Wag	21.0	6	160.0	110	3.90
Datsun 710	22.8	4	108.0	93	3.85
Hornet 4 Drive	21.4	6	258.0	110	3.08
Hornet Sportabout	18.7	8	360.0	175	3.15
Valiant	18.1	6	225.0	105	2.76
Duster 360	14.3	8	360.0	245	3.21
Merc 240D	24.4	4	146.7	62	3.69
Merc 230	22.8	4	140.8	95	3.92
Merc 280	19.2	6	167.6	123	3.92
Merc 280C	17.8	6	167.6	123	3.92
Merc 450SE	16.4	8	275.8	180	3.07
Merc 450SL	17.3	8	275.8	180	3.07
Merc 450SLC	15.2	8	275.8	180	3.07
Cadillac Fleetwood	10.4	8	472.0	205	2.93
Lincoln Continental	10.4	8	460.0	215	3.00
Chrysler Imperial	14.7	8	440.0	230	3.23
Fiat 128	32.4	4	78.7	66	4.08
Honda Civic	30.4	4	75.7	52	4.93
Toyota Corolla	33.9	4	71.1	65	4.22
Toyota Corona	21.5	4	120.1	97	3.70
Dodge Challenger	15.5	8	318.0	150	2.76
AMC Javelin	15.2	8	304.0	150	3.15
Camaro Z28	13.3	8	350.0	245	3.73
Pontiac Firebird	19.2	8	400.0	175	3.08
Fiat X1-9	27.3	4	79.0	66	4.08
Porsche 914-2	26.0	4	120.3	91	4.43
Lotus Europa	30.4	4	95.1	113	3.77
Ford Pantera L	15.8	8	351.0	264	4.22
Ferrari Dino	19.7	6	145.0	175	3.62

Maserati Bora	15.0	8yl	301.0	1835	3.54
	mpg		disp	hp	drat
Volvo 142E	21.4	4dbl>	121.0	108	4.11
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>

```
In [132]: min(mtcars$mpg) # encontre o mínimo de milhas/galão (US)
```

10.4

```
In [134]: max(mtcars$mpg) # encontre o máximo de milhas / galão (EUA)
```

33.9

```
In [135]: mean(mtcars$mpg, na.rm = TRUE) # encontre a média de milhas / galão (EUA)
```

```
20.090625
```

```
In [136]: var(mtcars$mpg, na.rm = TRUE) # encontre a variancia de milhas / galão (EUA)
```

```
36.3241028225806
```



```
In [137]: summary(mtcars$mpg, na.rm = TRUE) # encontre o desvio padrão Milhas / galão (EUA)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
10.40	15.43	19.20	20.09	22.80	33.90

Adicionar variáveis a um `data.frame` pode ser feito atribuindo novos vetores.

O importante do `data.frame` é que ele aceita quase todos os tipos de vetores, ou seja, inteiros, numéricos, lógicos, fatores e caracteres.

```
In [141]: mtcars$newvar1 <- mtcars$mpg - mtcars$qsec
```

In [142]: `mtcars$newvar2 <- mtcars$newvar1 > 0`

```
In [143]: mtcars$newvar3 <- ifelse(mtcars$newvar2, "good", "bad")
```

[illegible]

In [145]:

mtcars

A data.frame: 32 × 15

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	newvar1	newvar2	newvar3	newvar4
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<chr>	<fct>
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	4.54	TRUE	good	level1
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	3.98	TRUE	good	level1
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	4.19	TRUE	good	level1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	1.96	TRUE	good	level1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	1.68	TRUE	good	level1
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	-2.12	FALSE	bad	level1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	-1.54	FALSE	bad	level1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	4.40	TRUE	good	level1
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	-0.10	FALSE	bad	level1
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4	0.90	TRUE	good	level1
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4	-1.10	FALSE	bad	level1
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3	-1.00	FALSE	bad	level1
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3	-0.30	FALSE	bad	level1
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3	-2.80	FALSE	bad	level1
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	-7.58	FALSE	bad	level1
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4	-7.42	FALSE	bad	level1
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4	-2.72	FALSE	bad	level1
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	12.93	TRUE	good	level2
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2	11.88	TRUE	good	level2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1	14.00	TRUE	good	level2
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1	1.49	TRUE	good	level1

Dodge Challenger	mpg	cyl	displacement	hwy	drat	wt	qsec	vs	am	gear	carb	newvar1	newvar2	newvar3	newvar4
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<chr>	<fct>
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2	-2.10	FALSE	bad	level1
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4	-2.11	FALSE	bad	level1
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2	2.15	TRUE	good	level1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1	8.40	TRUE	good	level1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2	9.30	TRUE	good	level1
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2	13.50	TRUE	good	level2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4	1.30	TRUE	good	level1
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6	4.20	TRUE	good	level1
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8	0.40	TRUE	good	level1
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2	2.80	TRUE	good	level1

IMPORTAÇÃO

INTRODUÇÃO

Nesta seção, vamos introduzir os principais pacotes para importar dados para o R. Mostraremos como importar dados de arquivos de texto e de planilhas do excel.

Antes de começarmos, vale a pena tocarmos num ponto importante. As funções de importação do tidyverse carregam os dados em tibbles, que diferem da classe `data.frames` usual em dois pontos importantes:

imprime os dados na tela de maneira muito mais organizada, resumida e legível; e permite a utilização de `list-columns`.

Se você não estiver familiarizado com o conceito de `list-columns`, não se preocupe. Trataremos melhor do assunto no tópico sobre funcionais.

IMPORTANDO ARQUIVOS DE TEXTO

Para importar arquivos de texto para R, como .txt ou .csv, utilizaremos o pacote readr.

Como exemplo, utilizaremos uma base de filmes do IMDB, gravada em diversos formatos. Os arquivos podem ser encontrados neste [link \(https://github.com/curso-r/site-v2/tree/master/content/material/importacao/data\)](https://github.com/curso-r/site-v2/tree/master/content/material/importacao/data).

In [146]: `library(readr)`

```
imdb_csv <- read_csv(file = "https://raw.githubusercontent.com/curso-r/site-v2/master/content/material/importacao/data/imdb.csv")
imdb_txt <- read_delim(file = "https://raw.githubusercontent.com/curso-r/site-v2/master/content/material/importacao/data/imdb.txt", delim = " ")
```

Parsed with column specification:

```
cols(
  titulo = col_character(),
  ano = col_double(),
  diretor = col_character(),
  duracao = col_double(),
  cor = col_character(),
  generos = col_character(),
  pais = col_character(),
  classificacao = col_character(),
  orcamento = col_double(),
  receita = col_double(),
  nota_imdb = col_double(),
  likes_facebook = col_double(),
  ator_1 = col_character(),
  ator_2 = col_character(),
  ator_3 = col_character()
)
```

Parsed with column specification:

```
cols(
  titulo = col_character(),
  ano = col_double(),
  diretor = col_character(),
  duracao = col_double(),
  cor = col_character(),
  generos = col_character(),
  pais = col_character(),
  classificacao = col_character(),
  orcamento = col_double(),
  receita = col_double(),
  nota_imdb = col_double(),
  likes_facebook = col_double(),
  ator_1 = col_character(),
  ator_2 = col_character(),
  ator_3 = col_character()
)
```

```
    documento = col_double(),  
    receita = col_double(),  
    nota_imdb = col_double(),  
    likes_facebook = col_double(),  
    ator_1 = col_character(),  
    ator_2 = col_character(),  
    ator_3 = col_character()  
)  
)
```

```
In [147]: dim(imdb_csv)
```

```
3807 · 15
```

In [148]: imdb_txt

A spec_tbl_df: 3807 × 15

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Avatar	2009	James Cameron	178	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: At World's End	2007	Gore Verbinski	169	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Dark Knight Rises	2012	Christopher Nolan	164	Color	Action Thriller	USA	A partir de 13 anos
John Carter	2012	Andrew Stanton	132	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Spider-Man 3	2007	Sam Raimi	156	Color	Action Adventure Romance	USA	A partir de 13 anos
Tangled	2010	Nathan Greno	100	Color	Adventure Animation Comedy Family Fantasy Musical Romance	USA	Livre
Avengers: Age of Ultron	2015	Joss Whedon	141	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Batman v Superman: Dawn of Justice	2016	Zack Snyder	183	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Superman Returns	2006	Bryan Singer	169	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: Dead Man's Chest	2006	Gore Verbinski	151	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Lone Ranger	2013	Gore Verbinski	150	Color	Action Adventure Western	USA	A partir de 13 anos
Man of Steel	2013	Zack Snyder	143	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
The Chronicles of Narnia: Prince Caspian	2008	Andrew Adamson	150	Color	Action Adventure Family Fantasy	USA	Livre

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Avengers	2012	Joss Whedon	173	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: On Stranger Tides	2011	Rob Marshall	136	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Men in Black 3	2012	Barry Sonnenfeld	106	Color	Action Adventure Comedy Family Fantasy Sci-Fi	USA	A partir de 13 anos
The Amazing Spider-Man	2012	Marc Webb	153	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Robin Hood	2010	Ridley Scott	156	Color	Action Adventure Drama History	USA	A partir de 13 anos
The Hobbit: The Desolation of Smaug	2013	Peter Jackson	186	Color	Adventure Fantasy	USA	A partir de 13 anos
The Golden Compass	2007	Chris Weitz	113	Color	Adventure Family Fantasy	USA	A partir de 13 anos
Titanic	1997	James Cameron	194	Color	Drama Romance	USA	A partir de 13 anos
Captain America: Civil War	2016	Anthony Russo	147	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Battleship	2012	Peter Berg	131	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Jurassic World	2015	Colin Trevorrow	124	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Spider-Man 2	2004	Sam Raimi	135	Color	Action Adventure Fantasy Romance	USA	A partir de 13 anos
Iron Man 3	2013	Shane Black	195	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Alice in Wonderland	2010	Tim Burton	108	Color	Adventure Family Fantasy	USA	Livre
Monsters University	2013	Dan Scanlon	104	Color	Adventure Animation Comedy Family Fantasy	USA	Outros
Transformers: Revenge of the Fallen	2009	Michael Bay	150	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Transformers: Age of Extinction	2014	Michael Bay	165	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
...
The Exploding Girl	2009	Bradley Rust Gray	79	Color	Drama	USA	Outros
The Legend of God's Gun	2007	Mike Bruce	78	Color	Western	USA	A partir de 18 anos
Mutual Appreciation	2005	Andrew Bujalski	109	Black and White	Comedy	USA	A partir de 18 anos
Her Cry: La Llorona Investigation	2013	Damir Catic	89	Color	Horror	USA	Outros
Clerks	1994	Kevin Smith	102	Black and White	Comedy	USA	A partir de 18 anos
Pink Narcissus	1971	James Bidgood	65	Color	Drama Fantasy	USA	Outros
Funny Ha Ha	2002	Andrew Bujalski	85	Color	Comedy Drama	USA	Outros
Sabotage	2014	David Ayer	109	Color	Action Crime Drama Thriller	USA	A partir de 18 anos
Manito	2002	Eric Eason	79	Color	Drama Family	USA	Outros
Slacker	1991	Richard Linklater	100	Black and White	Comedy Drama	USA	A partir de 18 anos
Dutch Kills	2015	Joseph Mazzella	90	Color	Crime Drama Thriller	USA	Outros
Dry Spell	2013	Travis Legge	90	Color	Comedy Romance	USA	Outros
Flywheel	2003	Alex Kendrick	120	Color	Drama	USA	Outros
Exeter	2015	Marcus Nispel	91	Color	Horror Mystery Thriller	USA	A partir de 18 anos
The Ridges	2011	Brandon Landers	143	NA	Drama Horror Thriller	USA	Outros

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Puffy Chair	2005	Jay Duplass	85	Color	Comedy Drama Romance	USA	A partir de 18 anos
Breaking Upwards	2009	Daryl Wein	88	Color	Romance	USA	Outros
All Superheroes Must Die	2011	Jason Trost	78	Color	Sci-Fi Thriller	USA	Outros
Pink Flamingos	1972	John Waters	108	Color	Comedy Crime Horror	USA	A partir de 18 anos
On the Downlow	2004	Tadeo Garcia	84	Color	Drama	USA	Outros
Sanctuary; Quite a Conundrum	2012	Thomas L. Phillips	82	Color	Comedy Horror Thriller	USA	Outros
Bang	1995	Ash Baron-Cohen	98	Color	Crime Drama	USA	Outros
Primer	2004	Shane Carruth	77	Color	Drama Sci-Fi Thriller	USA	A partir de 13 anos
El Mariachi	1992	Robert Rodriguez	81	Color	Action Crime Drama Romance Thriller	USA	A partir de 18 anos
The Mongol King	2005	Anthony Vallone	84	Color	Crime Drama	USA	A partir de 13 anos
Newlyweds	2011	Edward Burns	95	Color	Comedy Drama	USA	Outros
The Following	NA	NA	43	Color	Crime Drama Mystery Thriller	USA	Outros
A Plague So Pleasant	2013	Benjamin Roberds	76	Color	Drama Horror Thriller	USA	Outros
Shanghai Calling	2012	Daniel Hsia	100	Color	Comedy Drama Romance	USA	A partir de 13 anos
My Date with Drew	2004	Jon Gunn	90	Color	Documentary	USA	Livre

Repare que o argumento `file=` representa o caminho até o arquivo. Se o arquivo a ser lido não estiver no diretório de trabalho da sua sessão, você precisa especificar o caminho até o arquivo. O argumento `delim=` indica qual caracter separa cada coluna no arquivo de texto.

Para a maioria das funções `read_`, existe uma respectiva função `write_`.

Essas funções servem para salvar bases em um formato específico de arquivo.

Além do nome do arquivo a ser criado, você também precisa passar o objeto que será gravado. Repare nos exemplos abaixo que você precisa especificar a extensão do arquivo corretamente.

```
In [150]: write_csv(x = mtcars, path = "data\\mtcars.csv")  
          write_delim(x = mtcars, delim = " ", path = "data\\mtcars.txt")
```

Também é possível salvar objetos, como `data.frames` em um tipo especial de arquivos, o `.rds`.

A vantagem dessa extensão é guardar a estrutura dos dados salvos, como a classe das colunas de um `data.frame`.

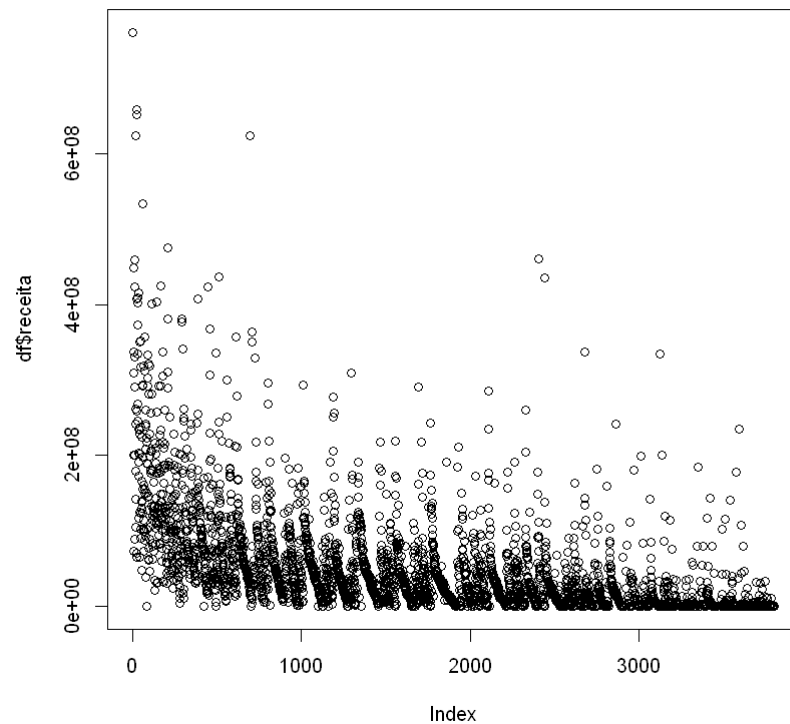
Além disso, é uma boa alternativa para lidar com grandes bancos de dados, já que arquivos `.rds` serão bem mais compactos do que arquivos Excel.

```
In [151]: imdb_rds <- read_rds(path = "imdb.rds")  
write_rds(mtcars, path = "mtcars.rds")
```

IMPORTANDO ARQUIVOS DO EXCEL

O pacote `readxl` contém funções para ler dados de arquivos do Excel, como `.xls` e `xlsx`.

```
In [152]: df <- readxl::read_xls(path = "imdb.xls")  
plot(df$receita)
```



In [153]: `readxl::read_xlsx(path = "imdb.xlsx")`

A tibble: 3807 × 15

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Avatar	2009	James Cameron	178	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: At World's End	2007	Gore Verbinski	169	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Dark Knight Rises	2012	Christopher Nolan	164	Color	Action Thriller	USA	A partir de 13 anos
John Carter	2012	Andrew Stanton	132	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Spider-Man 3	2007	Sam Raimi	156	Color	Action Adventure Romance	USA	A partir de 13 anos
Tangled	2010	Nathan Greno	100	Color	Adventure Animation Comedy Family Fantasy Musical Romance	USA	Livre
Avengers: Age of Ultron	2015	Joss Whedon	141	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Batman v Superman: Dawn of Justice	2016	Zack Snyder	183	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Superman Returns	2006	Bryan Singer	169	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: Dead Man's Chest	2006	Gore Verbinski	151	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Lone Ranger	2013	Gore Verbinski	150	Color	Action Adventure Western	USA	A partir de 13 anos
Man of Steel	2013	Zack Snyder	143	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
The Chronicles of Narnia: Prince Caspian	2008	Andrew Adamson	150	Color	Action Adventure Family Fantasy	USA	Livre

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Avengers	2012	Joss Whedon	173	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: On Stranger Tides	2011	Rob Marshall	136	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Men in Black 3	2012	Barry Sonnenfeld	106	Color	Action Adventure Comedy Family Fantasy Sci-Fi	USA	A partir de 13 anos
The Amazing Spider-Man	2012	Marc Webb	153	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Robin Hood	2010	Ridley Scott	156	Color	Action Adventure Drama History	USA	A partir de 13 anos
The Hobbit: The Desolation of Smaug	2013	Peter Jackson	186	Color	Adventure Fantasy	USA	A partir de 13 anos
The Golden Compass	2007	Chris Weitz	113	Color	Adventure Family Fantasy	USA	A partir de 13 anos
Titanic	1997	James Cameron	194	Color	Drama Romance	USA	A partir de 13 anos
Captain America: Civil War	2016	Anthony Russo	147	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Battleship	2012	Peter Berg	131	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Jurassic World	2015	Colin Trevorrow	124	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Spider-Man 2	2004	Sam Raimi	135	Color	Action Adventure Fantasy Romance	USA	A partir de 13 anos
Iron Man 3	2013	Shane Black	195	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Alice in Wonderland	2010	Tim Burton	108	Color	Adventure Family Fantasy	USA	Livre
Monsters University	2013	Dan Scanlon	104	Color	Adventure Animation Comedy Family Fantasy	USA	Outros
Transformers: Revenge of the Fallen	2009	Michael Bay	150	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Transformers: Age of Extinction	2014	Michael Bay	165	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
...
The Exploding Girl	2009	Bradley Rust Gray	79	Color	Drama	USA	Outros
The Legend of God's Gun	2007	Mike Bruce	78	Color	Western	USA	A partir de 18 anos
Mutual Appreciation	2005	Andrew Bujalski	109	Black and White	Comedy	USA	A partir de 18 anos
Her Cry: La Llorona Investigation	2013	Damir Catic	89	Color	Horror	USA	Outros
Clerks	1994	Kevin Smith	102	Black and White	Comedy	USA	A partir de 18 anos
Pink Narcissus	1971	James Bidgood	65	Color	Drama Fantasy	USA	Outros
Funny Ha Ha	2002	Andrew Bujalski	85	Color	Comedy Drama	USA	Outros
Sabotage	2014	David Ayer	109	Color	Action Crime Drama Thriller	USA	A partir de 18 anos
Manito	2002	Eric Eason	79	Color	Drama Family	USA	Outros
Slacker	1991	Richard Linklater	100	Black and White	Comedy Drama	USA	A partir de 18 anos
Dutch Kills	2015	Joseph Mazzella	90	Color	Crime Drama Thriller	USA	Outros
Dry Spell	2013	Travis Legge	90	Color	Comedy Romance	USA	Outros
Flywheel	2003	Alex Kendrick	120	Color	Drama	USA	Outros
Exeter	2015	Marcus Nispel	91	Color	Horror Mystery Thriller	USA	A partir de 18 anos
The Ridges	2011	Brandon Landers	143	NA	Drama Horror Thriller	USA	Outros

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Puffy Chair	2005	Jay Duplass	85	Color	Comedy Drama Romance	USA	A partir de 18 anos
Breaking Upwards	2009	Daryl Wein	88	Color	Romance	USA	Outros
All Superheroes Must Die	2011	Jason Trost	78	Color	Sci-Fi Thriller	USA	Outros
Pink Flamingos	1972	John Waters	108	Color	Comedy Crime Horror	USA	A partir de 18 anos
On the Downlow	2004	Tadeo Garcia	84	Color	Drama	USA	Outros
Sanctuary; Quite a Conundrum	2012	Thomas L. Phillips	82	Color	Comedy Horror Thriller	USA	Outros
Bang	1995	Ash Baron-Cohen	98	Color	Crime Drama	USA	Outros
Primer	2004	Shane Carruth	77	Color	Drama Sci-Fi Thriller	USA	A partir de 13 anos
El Mariachi	1992	Robert Rodriguez	81	Color	Action Crime Drama Romance Thriller	USA	A partir de 18 anos
The Mongol King	2005	Anthony Vallone	84	Color	Crime Drama	USA	A partir de 13 anos
Newlyweds	2011	Edward Burns	95	Color	Comedy Drama	USA	Outros
The Following	NA	NA	43	Color	Crime Drama Mystery Thriller	USA	Outros
A Plague So Pleasant	2013	Benjamin Roberds	76	Color	Drama Horror Thriller	USA	Outros
Shanghai Calling	2012	Daniel Hsia	100	Color	Comedy Drama Romance	USA	A partir de 13 anos
My Date with Drew	2004	Jon Gunn	90	Color	Documentary	USA	Livre

A função `read_excel()` auto detecta a extensão do arquivo.

In [154]: `readxl::read_excel(path = "imdb.xls")`

A tibble: 3807 × 15

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Avatar	2009	James Cameron	178	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: At World's End	2007	Gore Verbinski	169	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Dark Knight Rises	2012	Christopher Nolan	164	Color	Action Thriller	USA	A partir de 13 anos
John Carter	2012	Andrew Stanton	132	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Spider-Man 3	2007	Sam Raimi	156	Color	Action Adventure Romance	USA	A partir de 13 anos
Tangled	2010	Nathan Greno	100	Color	Adventure Animation Comedy Family Fantasy Musical Romance	USA	Livre
Avengers: Age of Ultron	2015	Joss Whedon	141	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Batman v Superman: Dawn of Justice	2016	Zack Snyder	183	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Superman Returns	2006	Bryan Singer	169	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: Dead Man's Chest	2006	Gore Verbinski	151	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Lone Ranger	2013	Gore Verbinski	150	Color	Action Adventure Western	USA	A partir de 13 anos
Man of Steel	2013	Zack Snyder	143	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
The Chronicles of Narnia: Prince Caspian	2008	Andrew Adamson	150	Color	Action Adventure Family Fantasy	USA	Livre

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Avengers	2012	Joss Whedon	173	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: On Stranger Tides	2011	Rob Marshall	136	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Men in Black 3	2012	Barry Sonnenfeld	106	Color	Action Adventure Comedy Family Fantasy Sci-Fi	USA	A partir de 13 anos
The Amazing Spider-Man	2012	Marc Webb	153	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Robin Hood	2010	Ridley Scott	156	Color	Action Adventure Drama History	USA	A partir de 13 anos
The Hobbit: The Desolation of Smaug	2013	Peter Jackson	186	Color	Adventure Fantasy	USA	A partir de 13 anos
The Golden Compass	2007	Chris Weitz	113	Color	Adventure Family Fantasy	USA	A partir de 13 anos
Titanic	1997	James Cameron	194	Color	Drama Romance	USA	A partir de 13 anos
Captain America: Civil War	2016	Anthony Russo	147	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Battleship	2012	Peter Berg	131	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Jurassic World	2015	Colin Trevorrow	124	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Spider-Man 2	2004	Sam Raimi	135	Color	Action Adventure Fantasy Romance	USA	A partir de 13 anos
Iron Man 3	2013	Shane Black	195	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Alice in Wonderland	2010	Tim Burton	108	Color	Adventure Family Fantasy	USA	Livre
Monsters University	2013	Dan Scanlon	104	Color	Adventure Animation Comedy Family Fantasy	USA	Outros
Transformers: Revenge of the Fallen	2009	Michael Bay	150	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Transformers: Age of Extinction	2014	Michael Bay	165	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
...
The Exploding Girl	2009	Bradley Rust Gray	79	Color	Drama	USA	Outros
The Legend of God's Gun	2007	Mike Bruce	78	Color	Western	USA	A partir de 18 anos
Mutual Appreciation	2005	Andrew Bujalski	109	Black and White	Comedy	USA	A partir de 18 anos
Her Cry: La Llorona Investigation	2013	Damir Catic	89	Color	Horror	USA	Outros
Clerks	1994	Kevin Smith	102	Black and White	Comedy	USA	A partir de 18 anos
Pink Narcissus	1971	James Bidgood	65	Color	Drama Fantasy	USA	Outros
Funny Ha Ha	2002	Andrew Bujalski	85	Color	Comedy Drama	USA	Outros
Sabotage	2014	David Ayer	109	Color	Action Crime Drama Thriller	USA	A partir de 18 anos
Manito	2002	Eric Eason	79	Color	Drama Family	USA	Outros
Slacker	1991	Richard Linklater	100	Black and White	Comedy Drama	USA	A partir de 18 anos
Dutch Kills	2015	Joseph Mazzella	90	Color	Crime Drama Thriller	USA	Outros
Dry Spell	2013	Travis Legge	90	Color	Comedy Romance	USA	Outros
Flywheel	2003	Alex Kendrick	120	Color	Drama	USA	Outros
Exeter	2015	Marcus Nispel	91	Color	Horror Mystery Thriller	USA	A partir de 18 anos
The Ridges	2011	Brandon Landers	143	NA	Drama Horror Thriller	USA	Outros

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Puffy Chair	2005	Jay Duplass	85	Color	Comedy Drama Romance	USA	A partir de 18 anos
Breaking Upwards	2009	Daryl Wein	88	Color	Romance	USA	Outros
All Superheroes Must Die	2011	Jason Trost	78	Color	Sci-Fi Thriller	USA	Outros
Pink Flamingos	1972	John Waters	108	Color	Comedy Crime Horror	USA	A partir de 18 anos
On the Downlow	2004	Tadeo Garcia	84	Color	Drama	USA	Outros
Sanctuary; Quite a Conundrum	2012	Thomas L. Phillips	82	Color	Comedy Horror Thriller	USA	Outros
Bang	1995	Ash Baron-Cohen	98	Color	Crime Drama	USA	Outros
Primer	2004	Shane Carruth	77	Color	Drama Sci-Fi Thriller	USA	A partir de 13 anos
El Mariachi	1992	Robert Rodriguez	81	Color	Action Crime Drama Romance Thriller	USA	A partir de 18 anos
The Mongol King	2005	Anthony Vallone	84	Color	Crime Drama	USA	A partir de 13 anos
Newlyweds	2011	Edward Burns	95	Color	Comedy Drama	USA	Outros
The Following	NA	NA	43	Color	Crime Drama Mystery Thriller	USA	Outros
A Plague So Pleasant	2013	Benjamin Roberds	76	Color	Drama Horror Thriller	USA	Outros
Shanghai Calling	2012	Daniel Hsia	100	Color	Comedy Drama Romance	USA	A partir de 13 anos
My Date with Drew	2004	Jon Gunn	90	Color	Documentary	USA	Livre

In [155]: readxl::read_excel(path = "imdb.xlsx")

A tibble: 3807 × 15

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Avatar	2009	James Cameron	178	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: At World's End	2007	Gore Verbinski	169	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Dark Knight Rises	2012	Christopher Nolan	164	Color	Action Thriller	USA	A partir de 13 anos
John Carter	2012	Andrew Stanton	132	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Spider-Man 3	2007	Sam Raimi	156	Color	Action Adventure Romance	USA	A partir de 13 anos
Tangled	2010	Nathan Greno	100	Color	Adventure Animation Comedy Family Fantasy Musical Romance	USA	Livre
Avengers: Age of Ultron	2015	Joss Whedon	141	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Batman v Superman: Dawn of Justice	2016	Zack Snyder	183	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Superman Returns	2006	Bryan Singer	169	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: Dead Man's Chest	2006	Gore Verbinski	151	Color	Action Adventure Fantasy	USA	A partir de 13 anos
The Lone Ranger	2013	Gore Verbinski	150	Color	Action Adventure Western	USA	A partir de 13 anos
Man of Steel	2013	Zack Snyder	143	Color	Action Adventure Fantasy Sci-Fi	USA	A partir de 13 anos
The Chronicles of Narnia: Prince Caspian	2008	Andrew Adamson	150	Color	Action Adventure Family Fantasy	USA	Livre

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Avengers	2012	Joss Whedon	173	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Pirates of the Caribbean: On Stranger Tides	2011	Rob Marshall	136	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Men in Black 3	2012	Barry Sonnenfeld	106	Color	Action Adventure Comedy Family Fantasy Sci-Fi	USA	A partir de 13 anos
The Amazing Spider-Man	2012	Marc Webb	153	Color	Action Adventure Fantasy	USA	A partir de 13 anos
Robin Hood	2010	Ridley Scott	156	Color	Action Adventure Drama History	USA	A partir de 13 anos
The Hobbit: The Desolation of Smaug	2013	Peter Jackson	186	Color	Adventure Fantasy	USA	A partir de 13 anos
The Golden Compass	2007	Chris Weitz	113	Color	Adventure Family Fantasy	USA	A partir de 13 anos
Titanic	1997	James Cameron	194	Color	Drama Romance	USA	A partir de 13 anos
Captain America: Civil War	2016	Anthony Russo	147	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Battleship	2012	Peter Berg	131	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Jurassic World	2015	Colin Trevorrow	124	Color	Action Adventure Sci-Fi Thriller	USA	A partir de 13 anos
Spider-Man 2	2004	Sam Raimi	135	Color	Action Adventure Fantasy Romance	USA	A partir de 13 anos
Iron Man 3	2013	Shane Black	195	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
Alice in Wonderland	2010	Tim Burton	108	Color	Adventure Family Fantasy	USA	Livre
Monsters University	2013	Dan Scanlon	104	Color	Adventure Animation Comedy Family Fantasy	USA	Outros
Transformers: Revenge of the Fallen	2009	Michael Bay	150	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
Transformers: Age of Extinction	2014	Michael Bay	165	Color	Action Adventure Sci-Fi	USA	A partir de 13 anos
...
The Exploding Girl	2009	Bradley Rust Gray	79	Color	Drama	USA	Outros
The Legend of God's Gun	2007	Mike Bruce	78	Color	Western	USA	A partir de 18 anos
Mutual Appreciation	2005	Andrew Bujalski	109	Black and White	Comedy	USA	A partir de 18 anos
Her Cry: La Llorona Investigation	2013	Damir Catic	89	Color	Horror	USA	Outros
Clerks	1994	Kevin Smith	102	Black and White	Comedy	USA	A partir de 18 anos
Pink Narcissus	1971	James Bidgood	65	Color	Drama Fantasy	USA	Outros
Funny Ha Ha	2002	Andrew Bujalski	85	Color	Comedy Drama	USA	Outros
Sabotage	2014	David Ayer	109	Color	Action Crime Drama Thriller	USA	A partir de 18 anos
Manito	2002	Eric Eason	79	Color	Drama Family	USA	Outros
Slacker	1991	Richard Linklater	100	Black and White	Comedy Drama	USA	A partir de 18 anos
Dutch Kills	2015	Joseph Mazzella	90	Color	Crime Drama Thriller	USA	Outros
Dry Spell	2013	Travis Legge	90	Color	Comedy Romance	USA	Outros
Flywheel	2003	Alex Kendrick	120	Color	Drama	USA	Outros
Exeter	2015	Marcus Nispel	91	Color	Horror Mystery Thriller	USA	A partir de 18 anos
The Ridges	2011	Brandon Landers	143	NA	Drama Horror Thriller	USA	Outros

titulo	ano	diretor	duracao	cor	generos	pais	classificacao
<chr>	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>
The Puffy Chair	2005	Jay Duplass	85	Color	Comedy Drama Romance	USA	A partir de 18 anos
Breaking Upwards	2009	Daryl Wein	88	Color	Romance	USA	Outros
All Superheroes Must Die	2011	Jason Trost	78	Color	Sci-Fi Thriller	USA	Outros
Pink Flamingos	1972	John Waters	108	Color	Comedy Crime Horror	USA	A partir de 18 anos
On the Downlow	2004	Tadeo Garcia	84	Color	Drama	USA	Outros
Sanctuary; Quite a Conundrum	2012	Thomas L. Phillips	82	Color	Comedy Horror Thriller	USA	Outros
Bang	1995	Ash Baron-Cohen	98	Color	Crime Drama	USA	Outros
Primer	2004	Shane Carruth	77	Color	Drama Sci-Fi Thriller	USA	A partir de 13 anos
El Mariachi	1992	Robert Rodriguez	81	Color	Action Crime Drama Romance Thriller	USA	A partir de 18 anos
The Mongol King	2005	Anthony Vallone	84	Color	Crime Drama	USA	A partir de 13 anos
Newlyweds	2011	Edward Burns	95	Color	Comedy Drama	USA	Outros
The Following	NA	NA	43	Color	Crime Drama Mystery Thriller	USA	Outros
A Plague So Pleasant	2013	Benjamin Roberds	76	Color	Drama Horror Thriller	USA	Outros
Shanghai Calling	2012	Daniel Hsia	100	Color	Comedy Drama Romance	USA	A partir de 13 anos
My Date with Drew	2004	Jon Gunn	90	Color	Documentary	USA	Livre