

Introdução à Análise de Dados com Linguagem R

Aula 4

1. Data Frames

dataframe indiscutivelmente é a estrutura de dados mais importante em R, é nesta estrutura que a maioria dos seus dados será armazenada para análise. Combina a estrutura de uma matriz com a flexibilidade de ter diferentes tipos de dados em cada coluna. Pense em cada coluna como um vetor armazenando um tipo de dado específico. Para criar um *dataframe* utilizamos a função `dataframe()`.

Criar um data frame

```
In [19]: df <- data.frame(  
  # Coluna id  
  id = c(1, 2, 3, 4, 5),  
  # Coluna nome  
  nome = c('Mezilaurus itauba', 'Apuleia leiocarpa', 'Cedrela odorata',  
           'Amburana acreana', 'Hymenolobium excelsum'),  
  # Coluna volume  
  volume = c(3.25, 6.51, 7.45, 8.81, 4.35)  
)
```

```
In [20]: df
```

id	nome	volume
1	Mezilaurus itauba	3.25
2	Apuleia leiocarpa	6.51
3	Cedrela odorata	7.45
4	Amburana acreana	8.81
5	Hymenolobium excelsum	4.35

Carregar um data frame a partir de um arquivo

Em R é possível a leitura de vários formatos de arquivos utilizados para armazenamento de dados, a exemplo de:

- **.csv**
- `.dbf`

- `.dta` (Stata)
- `.fst`
- `.h5`
- `.mtp` (Minitab)
- `.parquet`
- `.rda`
- `.rds`
- `.RData`
- `.spss` (SPSS)
- `.txt`
- `.xls` e `.xlsx`
- `.xml`
- `.xport` (SAS)

Abordaremos neste módulo apenas os formatos mais frequentemente utilizados, a saber:

`.csv`, `.txt`, `.rds` e `.xls`

Para leitura de arquivos nos formatos `.csv` e `.txt`, podemos utilizar a função `read.csv()` ou `read.csv2()`, a primeira por padrão lê dados de planilhas onde o separador decimal é o `.` e o separador de colunas é a `,`; ao passo que a segunda função é utilizada para leitura de planilhas onde o separador decimal é a `,` e o separador de colunas é o `;`

Leitura de Arquivos `.csv` e `.txt`

O trecho de código a seguir faz a leitura de uma planilha em formato `.csv`, a qual é atribuída ao objeto R denominado `inventario`, o qual após leitura dos dados passa a ser um dataframe, pois as colunas da planilha são importadas como vetores. Assim, podemos definir dataframe como um conjunto de vetores, mas pode também armazenar listas.

```
In [2]: # Lê uma planilha csv com dados de inventário florestal
inventario <- read.csv2('./data/UMF_4_UPA_4F_SINAFLO_03.csv',
                        encoding = 'latin1')
```

A seguir usamos a função `head()` para mostrar as linhas iniciais do dataframe `inventario`. Por padrão esta função mostra apenas as seis primeiras linhas do dataframe, caso queiramos ler as dez primeiras linhas, devemos especificar este número após o nome do dataframe: `head(inventario, 10)`

```
In [14]: head(inventario)
```

A data.frame: 6 × 13

	N_arv	UPA	UT	Nome_Cientifico	Nome_Popular	DAP_cm	Alt	Categoria
	<int>	<int>	<int>	<chr>	<chr>	<dbl>	<int>	<chr>
1	10001	6	1	Parkia gigantocarpa	Fava-atanã	114.59	19	Remanescente
2	10002	6	1	Bagassa guianensis	Tatajuba	67.48	17	Remanescente
3	10003	6	1	Castilla ulei	Caucho	52.52	11	Explorar
4	10004	6	1	Castilla ulei	Caucho	40.74	13	Remanescente
5	10005	6	1	Vochysia maxima	Quaruba	47.75	15	Remanescente
6	10006	6	1	Copaifera duckei	Copaíba	43.93	18	Remanescente

A seguir usamos a função `tail()` para mostrar as últimas linhas do dataframe `inventario`. Por padrão esta função mostra apenas as seis últimas linhas do dataframe, caso queiramos ler as dez últimas linhas, devemos especificar este número após o nome do dataframe: `tail(inventario, 10)`

```
In [15]: tail(inventario)
```

A data.frame: 6 × 13

	N_arv	UPA	UT	Nome_Cientifico	Nome_Popular	DAP_cm	Alt	Cate
	<int>	<int>	<int>	<chr>	<chr>	<dbl>	<int>	<
18583	290695	6	29	Parkia multijuga	Fava-benguê	52.52	15	Remanes
18584	290696	6	29	Couratari guianensis	Tauari	92.95	24	Exp
18585	290697	6	29	Vochysia maxima	Quaruba	112.68	20	Remanes
18586	290698	6	29	Hymenaea parvifolia	Jutaí-mirim	66.53	18	Remanes
18587	290699	6	29	Vochysia maxima	Quaruba	127.32	19	Exp
18588	290700	6	29	Vochysia maxima	Quaruba	54.75	17	Remanes

Obter os nomes das colunas de um Dataframe

Alguns Dataframes contém um número muito grande de colunas e precisamos atribuir os mesmos nomes a outros dataframes que formos criando, para evitar ter que digitar estes nomes outras vezes, podemos usar as função `names()` e `attributes()`.

A primeira retornará um saída vetorial ao passo que a segunda uma lista com três elementos, o primeiro é `names`: nomes das colunas; `class`: mostra a estrutura de dados,

no caso data.frame; `row.names`: referente ao nomes dos rótulos das linhas, se houver. Para acessarmos somente os nomes das colunas usando a função `attributes()` devemos chamá-la da seguinte forma: `attributes(dataframe)[1]`

```
In [16]: names(inventario)
```

```
'N_arv' · 'UPA' · 'UT' · 'Nome_Cientifico' · 'Nome_Popular' · 'DAP_cm' · 'Alt' · 'Categoria' · 'QF' ·  
'Vol' · 'g' · 'lat' · 'lon'
```

```
In [17]: attributes(inventario)
```

\$names	'N_arv' · 'UPA' · 'UT' · 'Nome_Cientifico' · 'Nome_Popular' · 'DAP_cm' · 'Alt' · 'Categoria' · 'QF' · 'Vol' · 'g' · 'lat' · 'lon'
\$class	'data.frame'
\$row.names	1 · 2 · 3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 · 21 · 22 · 23 · 24 · 25 · 26 · 27 · 28 · 29 · 30 · 31 · 32 · 33 · 34 · 35 · 36 · 37 · 38 · 39 · 40 · 41 · 42 · 43 · 44 · 45 · 46 · 47 · 48 · 49 · 50 · 51 · 52 · 53 · 54 · 55 · 56 · 57 · 58 · 59 · 60 · 61 · 62 · 63 · 64 · 65 · 66 · 67 · 68 · 69 · 70 · 71 · 72 · 73 · 74 · 75 · 76 · 77 · 78 · 79 · 80 · 81 · 82 · 83 · 84 · 85 · 86 · 87 · 88 · 89 · 90 · 91 · 92 · 93 · 94 · 95 · 96 · 97 · 98 · 99 · 100 · 101 · 102 · 103 · 104 · 105 · 106 · 107 · 108 · 109 · 110 · 111 · 112 · 113 · 114 · 115 · 116 · 117 · 118 · 119 · 120 · 121 · 122 · 123 · 124 · 125 · 126 · 127 · 128 · 129 · 130 · 131 · 132 · 133 · 134 · 135 · 136 · 137 · 138 · 139 · 140 · 141 · 142 · 143 · 144 · 145 · 146 · 147 · 148 · 149 · 150 · 151 · 152 · 153 · 154 · 155 · 156 · 157 · 158 · 159 · 160 · 161 · 162 · 163 · 164 · 165 · 166 · 167 · 168 · 169 · 170 · 171 · 172 · 173 · 174 · 175 · 176 · 177 · 178 · 179 · 180 · 181 · 182 · 183 · 184 · 185 · 186 · 187 · 188 · 189 · 190 · 191 · 192 · 193 · 194 · 195 · 196 · 197 · 198 · 199 · 200 · ... · 18389 · 18390 · 18391 · 18392 · 18393 · 18394 · 18395 · 18396 · 18397 · 18398 · 18399 · 18400 · 18401 · 18402 · 18403 · 18404 · 18405 · 18406 · 18407 · 18408 · 18409 · 18410 · 18411 · 18412 · 18413 · 18414 · 18415 · 18416 · 18417 · 18418 · 18419 · 18420 · 18421 · 18422 · 18423 · 18424 · 18425 · 18426 · 18427 · 18428 · 18429 · 18430 · 18431 · 18432 · 18433 · 18434 · 18435 · 18436 · 18437 · 18438 · 18439 · 18440 · 18441 · 18442 · 18443 · 18444 · 18445 · 18446 · 18447 · 18448 · 18449 · 18450 · 18451 · 18452 · 18453 · 18454 · 18455 · 18456 · 18457 · 18458 · 18459 · 18460 · 18461 · 18462 · 18463 · 18464 · 18465 · 18466 · 18467 · 18468 · 18469 · 18470 · 18471 · 18472 · 18473 · 18474 · 18475 · 18476 · 18477 · 18478 · 18479 · 18480 · 18481 · 18482 · 18483 · 18484 · 18485 · 18486 · 18487 · 18488 · 18489 · 18490 · 18491 · 18492 · 18493 · 18494 · 18495 · 18496 · 18497 · 18498 · 18499 · 18500 · 18501 · 18502 · 18503 · 18504 · 18505 · 18506 · 18507 · 18508 · 18509 · 18510 · 18511 · 18512 · 18513 · 18514 · 18515 · 18516 · 18517 · 18518 · 18519 · 18520 · 18521 · 18522 · 18523 · 18524 · 18525 · 18526 · 18527 · 18528 · 18529 · 18530 · 18531 · 18532 · 18533 · 18534 · 18535 · 18536 · 18537 · 18538 · 18539 · 18540 · 18541 · 18542 · 18543 · 18544 · 18545 · 18546 · 18547 · 18548 · 18549 · 18550 · 18551 · 18552 · 18553 · 18554 · 18555 · 18556 · 18557 · 18558 · 18559 · 18560 · 18561 · 18562 · 18563 · 18564 · 18565 · 18566 · 18567 · 18568 · 18569 · 18570 · 18571 · 18572 · 18573 · 18574 · 18575 · 18576 · 18577 · 18578 · 18579 · 18580 · 18581 · 18582 · 18583 · 18584 · 18585 · 18586 · 18587 · 18588

In [18]: `attributes(inventario)[1]`

\$names =

'N_arv' · 'UPA' · 'UT' · 'Nome_Cientifico' · 'Nome_Popular' · 'DAP_cm' · 'Alt' · 'Categoria' · 'QF' ·
'Vol' · 'g' · 'lat' · 'lon'

Obter número de linhas e colunas de uma Dataframe

A função `dim()` recebe como parâmetro um dataframe e retorna o número de linhas e colunas.

```
In [19]: dim(inventario)
```

18588 · 13

- Podemos acessar apenas o número de linhas ou de colunas: `dim(inventario)[1]` e `dim(inventario)[2]`, respetivamente.

```
In [21]: # Mostrar apenas o número de linhas do dataframe  
dim(inventario)[1]
```

18588

```
In [22]: # Mostrar apenas o número de colunas do dataframe  
dim(inventario)[2]
```

13

Os comandos acima podem ser simplificados com o uso apenas das funções `nrow()` e `ncol()`. veja os exemplos a seguir:

```
In [24]: # Obter o número de linhas de um dataframe  
nrow(inventario)
```

18588

```
In [23]: # Obter o número de colunas de um dataframe  
ncol(inventario)
```

13

Dos exemplos acima podemos observar que foram inventariadas 18.588 árvores (número de linhas) e que o dataframe contém 13 variáveis (colunas)

Obter a Estrutura dos Dados de um DataFrame

```
In [26]: # Verificar a estrutura dos dados  
str(inventario)
```

```
'data.frame': 18588 obs. of 13 variables:
 $ N_arv      : int  10001 10002 10003 10004 10005 10006 10007 10008 10009 10
010 ...
 $ UPA        : int   6 6 6 6 6 6 6 6 6 6 ...
 $ UT         : int   1 1 1 1 1 1 1 1 1 1 ...
 $ Nome_Cientifico: chr  "Parkia gigantocarpa" "Bagassa guianensis" "Castilla ulei"
"Castilla ulei" ...
 $ Nome_Popular  : chr  "Fava-atanã" "Tatajuba" "Caucho" "Caucho" ...
 $ DAP_cm       : num  114.6 67.5 52.5 40.7 47.8 ...
 $ Alt         : int   19 17 11 13 15 18 16 20 20 15 ...
 $ Categoria    : chr  "Remanescente" "Remanescente" "Explorar" "Remanescente"
...
 $ QF          : int   1 1 1 1 2 1 2 1 1 1 ...
 $ Vol         : num   12.49 4.39 1.83 1.11 1.83 ...
 $ g           : num   1.031 0.358 0.217 0.13 0.179 ...
 $ lat         : num  -5.91 -5.91 -5.91 -5.91 -5.91 ...
 $ lon         : num  -55 -55 -55 -55 -55 ...
```

Resumo dos Dados de um Dataframe

```
In [79]: # Mostrar o resumo dos dados
summary(inventario)
```

N_arv	UPA	UT	Nome_Cientifico
Min. : 10001	Min. : 6	Min. : 1.00	Length:18588
1st Qu.: 90010	1st Qu.: 6	1st Qu.: 9.00	Class :character
Median :160623	Median : 6	Median :16.00	Mode :character
Mean :161674	Mean : 6	Mean :16.13	
3rd Qu.:240164	3rd Qu.: 6	3rd Qu.:24.00	
Max. :290700	Max. : 6	Max. :29.00	
Nome_Popular	DAP_cm	Alt	Categoria
Length:18588	Min. : 39.79	Min. : 7.00	Length:18588
Class :character	1st Qu.: 54.11	1st Qu.:16.00	Class :character
Mode :character	Median : 66.85	Median :18.00	Mode :character
	Mean : 71.95	Mean :18.12	
	3rd Qu.: 84.99	3rd Qu.:20.00	
	Max. :312.26	Max. :40.00	
QF	Vol	g	lat
Min. :1.000	Min. : 0.7213	Min. :0.1243	Min. : -5.918
1st Qu.:1.000	1st Qu.: 2.7023	1st Qu.:0.2300	1st Qu.: -5.897
Median :1.000	Median : 4.4863	Median :0.3509	Median : -5.881
Mean :1.282	Mean : 5.5925	Mean :0.4523	Mean : -5.883
3rd Qu.:2.000	3rd Qu.: 7.3581	3rd Qu.:0.5673	3rd Qu.: -5.871
Max. :3.000	Max. :39.0858	Max. :7.6582	Max. : -5.848
lon			
Min. : -55.00			
1st Qu.: -54.97			
Median : -54.96			
Mean : -54.96			
3rd Qu.: -54.95			
Max. : -54.94			

Acessar colunas do data frame

Para acessar uma coluna específica de um data frame usamos o sinal `$`, se queremos acessar somente a coluna referente a variável volume, denominada em nossa data frame de `Vol`: `inventario$Vol`. Isso nos permite aplicar uma função apenas a esta coluna:

```
In [3]: # Selecionar a coluna "Vol" e Mostrar o resumo dos dados da coluna "Vol"
summary(inventario$Vol)
```

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.7213  2.7023  4.4863  5.5925  7.3581 39.0858
```

Filtrar Linhas e/ou Colunas de uma Data Frame

Em função do data frame possuir duas dimensões, é possível localizar uma linha e/ou uma coluna através das coordenadas de suas dimensões. Vamos considerar como exemplo os dados referente a lista de espécies da fauna brasileira ameaçadas de extinção.

```
In [30]: fauna <- read.csv2('./data/DF_port_MMA_300-2022_fauna.csv')
head(fauna)
```

n	port443	classe	ordem	familia	especie_subespecie	categoria
1	*	Aves	Accipitriformes	Accipitridae	Amadonastur lacernulatus	VU
2	*	Aves	Accipitriformes	Accipitridae	Circus cinereus	VU
3	*	Aves	Accipitriformes	Accipitridae	Harpia harpyja	VU
4	*	Aves	Accipitriformes	Accipitridae	Leptodon forbesi	EN
5	*	Aves	Accipitriformes	Accipitridae	Morphnus guianensis	VU
6	*	Aves	Accipitriformes	Accipitridae	Urubitinga coronata	EN

Para filtrar a segunda linha passamos o endereço desta linha no data frame

```
In [24]: fauna[2, ]
```

	n	port443	classe	ordem	familia	especie_subespecie	categoria
2	2	*	Aves	Accipitriformes	Accipitridae	Circus cinereus	VU

Para filtrar apenas a terceira coluna

```
In [ ]: print(fauna[, 3])
```

Também podemos filtrar um data frame com base em uma coluna

```
In [27]: fauna[fauna$especie_subespecie == 'Eurola ziczac', ]
```


	n	port443	classe	ordem	familia	especie_subespecie	categoria
672	679	*	Bivalvia	Ostreoida	Pectinidae	Euvola ziczac	EN

Filtrar dados com a função `subset`

```
In [3]: # Gerar um subconjunto das árvores selecionadas para corte
especies_explorar <- subset(inventario, Categoria == 'Explorar')

# Mostrar o subconjunto
head(especies_explorar)
```

	N_arv	UPA	UT	Nome_Cientifico	Nome_Popular	DAP_cm	Alt	Categoria	QF	\
3	10003	6	1	Castilla ulei	Caucho	52.52	11	Explorar	1	1.82
7	10007	6	1	Pseudopiptadenia psilostachya	Timborana	59.52	16	Explorar	2	3.20
9	10009	6	1	Hymenolobium petraeum	Angelim-pedra	72.57	20	Explorar	1	5.73
13	10013	6	1	Goupia glabra	Cupiúba	74.80	16	Explorar	2	5.17
23	10023	6	1	Manilkara elata	Maçaranduba	74.17	19	Explorar	1	5.77
24	10024	6	1	Goupia glabra	Cupiúba	63.66	14	Explorar	1	3.36

```
In [4]: # Número de Árvores a Explorar
nrow(especies_explorar)
```

8221

Criar uma nova coluna

Iremos criar uma coluna em função de um filtro baseado em dados de outro data frame. Vamos filtrar os dados onde as espécies do `inventario` são espécies ameaçadas de extinção, as espécies que forem ameaçadas receberão a denominação conforme sua categoria definida na Portaria MMA 300/2022, para isso teremos que utilizar carregar um *data frame* com os dados da portaria e criar uma coluna no `inventario` denominada `status_ecologico`, a qual receberá a categoria de ameaça.

```
In [27]: # Ler os dados da portaria MMA 300/2022
port_300_flora <- read.csv2('./data/DF_port_MMA_300-2022_flora.csv')

# Mostrar na tela apenas 6 linhas do data frame
head(port_300_flora)
```

n	port443	familia	especie_subespecie_var	categoria
1	*	Acanthaceae	Aphelandra espirito-santensis	EN
2	*	Acanthaceae	Aphelandra margaritae	VU
3	*	Acanthaceae	Aphelandra maximiliana	EN
4		Acanthaceae	Aphelandra rigida	EN
5		Acanthaceae	Aphelandra stephanophysa	VU
6	*	Acanthaceae	Dyschoriste lavandulacea	EN

```
In [28]: # Filtar as espécies do inventario que coincidem as espécies da Portaria
# MMA 300/2022 e atribuir a categoria de ameaça a uma nova coluna em inventario,
# a ser denominada status_ecologico.
inventario_ameacadas <- inventario[inventario$Nome_Cientifico %in% port_300_flor
```

```
In [29]: unique(inventario_ameacadas$Nome_Cientifico)
```

'Bertholletia excelsa' · 'Apuleia leiocarpa' · 'Cedrela odorata' · 'Hymenaea parvifolia' ·
'Virola surinamensis' · 'Mezilaurus itauba'

Excluir Linhas e Colunas de um Data Frame

```
In [2]: # Excluir a linha 2
fauna <- fauna[-2, ]

# Conferir a remoção da linha 2
head(fauna)
```

A data.frame: 6 × 7

	n	port443	classe	ordem	familia	especie_subespecie	categoria
	<int>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	1	*	Aves	Accipitriformes	Accipitridae	Amadonastur lacernulatus	VU
3	3	*	Aves	Accipitriformes	Accipitridae	Harpia harpyja	VU
4	4	*	Aves	Accipitriformes	Accipitridae	Leptodon forbesi	EN
5	5	*	Aves	Accipitriformes	Accipitridae	Morphnus guianensis	VU
6	6	*	Aves	Accipitriformes	Accipitridae	Urubitinga coronata	EN
7	7	*	Anthozoa	Actiniaria	Actiniidae	Condylactis gigantea	EN

```
In [3]: # Excluir a coluna n
fauna$n <- NULL

# Conferir a remoção da coluna 1
head(fauna)
```

A data.frame: 6 × 6

	port443	classe	ordem	familia	especie_subespecie	categoria
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	*	Aves	Accipitriformes	Accipitridae	Amadonastur lacernulatus	VU
3	*	Aves	Accipitriformes	Accipitridae	Harpia harpyja	VU
4	*	Aves	Accipitriformes	Accipitridae	Leptodon forbesi	EN
5	*	Aves	Accipitriformes	Accipitridae	Morphnus guianensis	VU
6	*	Aves	Accipitriformes	Accipitridae	Urubitinga coronata	EN
7	*	Anthozoa	Actiniaria	Actiniidae	Condylactis gigantea	EN

Ler Dados de Arquivo Demilitado por Tabulação

```
In [27]: dados <- read.delim2('./data/Bivalvia_tab.txt')
head(dados)
```

	n	port443	classe	ordem	familia	especie_subespecie	categoria
679		*	Bivalvia	Ostreoida	Pectinidae	Euvola ziczac	EN
1275		*	Bivalvia	Unionoida	Hyriidae	Diplodon koseritzi	EN
1276		*	Bivalvia	Unionoida	Mycetopodidae	Mycetopoda legumen	EN

Ler Dados de Arquivo Delimitado por Espaço

```
In [13]: dados <- read.delim2('./data/Separado_por_Espaco.txt', sep = ' ')
head(dados)
```

id	nome	volume
1	Mezilaurus itauba	3.25
2	Apuleia leiocarpa	6.51
3	Cedrela odorata	7.45
4	Amburana acreana	8.81
5	Hymenolobium excelsum	4.35

Ler Dados de Arquivo Hospedado na Internet

A seguir é mostrado como ler dados diretamente da internet, como exemplo iremos ler um conjunto de dados referente

```
In [22]: link <- 'http://www.ibama.gov.br/phocadownload/sinaflor/2022/2022-07-22_Lista_es
```

```
sp_sistaxon <- read.csv(link)
```

```
In [31]: head(sp_sistaxon)
```

Código.da.espécie	Nome.cientifico	Código.Nome.Popular	Nome.popular
1980924	Abarema adenophora	88915	Olho-de-peixe
1980927	Abarema cochliacarpus	88658	Babatenon
1049335	Abarema filamentosa	88649	Olho-de-pomba
35843	Abarema jupunba	45309	Angelim-falso
35843	Abarema jupunba	47519	Contas-de-nossa-senhora
35843	Abarema jupunba	43009	Fava-amargosa

Leitura de Dados em Formatos de Arquivos de Softwares Proprietários

Arquivo	Pacote	Função
.dbf	foreign	read.dbf()
.dta	foreign	read.tda()
.fst	fst	read_fst()
.mtp	foreign	read.mtp()
.spss	foreign	read.spss()
.xls	readxl	read_xls()
.xls	readxl	read_xlsx()
.xport	foreign	read.xport()

Ler Dados de Arquivos .xls e .xlsx

```
In [2]: # Carregar o pacote readxl
library(readxl)

# Listar as planilhas presentes no arquivo .xls
xls <- excel_sheets('./data/aves_reptilia.xls')
xls
```

'aves' · 'reptilia'

Por padrão a função `read_xls()` lê a primeira planilha do arquivo.

```
In [ ]: # Ler dados da primeira planilha "aves" e atribuir a um dataframe
dados_fauna <- read_xls('./data/aves_reptilia.xls')
```

```
# Mostrar as primeiras seis linhas do dataframe  
head(dados_fauna)
```

Para ler uma outra planilha devemos passar essa informação para o parâmetro `sheets` .
Se quisermos ler a planilha `reptilia` o código seria:

```
read_xls('./data/aves_reptilia.xls', sheet = 'reptilia') ou  
read_xls('./data/aves_reptilia.xls', sheet = 2)
```

```
In [ ]: # Ler dados da primeira planilha "reptilia" e atribuir a um dataframe  
dados_fauna <- read_xls('./data/aves_reptilia.xls', sheet = 'reptilia')  
  
# Mostrar as primeiras seis linhas do dataframe  
head(dados_fauna)
```

```
In [ ]:
```