

Tidyverse - aula 1 - Introdução

Wilson Souza

2021-03-17

Contents

1	Introdução	5
1.1	Importando os dados	7
2	<i>tibble</i>	9
2.1	as_tibble() - converte um objeto do tipo matriz ou data frame para tbl_df	9
2.2	add_column() - adiciona uma coluna nova a planilha	10
2.3	add_row() - adiciona uma nova linha a planilha	11
2.4	remove_rownames() e column_to_rownames() - Remove os nomes das colunas e adiciona um nome as colunas	12
2.5	rownames_to_column() - adiciona o nome das linhas como uma coluna	13
2.6	subsetting - forma de escrita que permite cortar/selecionar a planilha em função das linhas e colunas	14
3	<i>dplyr</i>	17
3.1	count() e add_count() - conta o número de linhas em função da variável especificada; adiciona uma nova coluna com a contagem do número de linhas em função da variável especificada	17
3.2	arrange() - rearranja as linhas em função da variável especificada	18
3.3	bind_cols() - une planilhas em colunas	19
3.4	bind_rows() - une planilhas em linhas	20
3.5	distinct() - remove as linhas que são exatamente iguais	20
3.6	filter() - filtra as linhas que satisfaçam alguma condição	21
3.7	group_by() - agrupa as linhas em função dos valores de alguma variável	24

3.8	<code>mutate()</code> e <code>transmute()</code> - adiciona novas variáveis e preserva as existentes; adiciona novas variáveis e não preserva as existentes .	25
3.9	<code>na_if()</code> - Substitui o valor especificado por NA	27
3.10	<code>recode()</code> e <code>recode_factor()</code> - substitui um determinado valor por outro, se variável for numérica usar <code>recode()</code> , se for fator usar <code>recode_factor()</code>	28
3.11	<code>relocate()</code> - altera a ordem das variáveis	29
3.12	<code>rename()</code> - altera o nome das variáveis	29
3.13	<code>select()</code> - Seleciona variáveis	30
3.14	<code>slice()</code> - Seleciona linhas	32
3.15	<code>summarise()</code> - sumariza os dados	33
4	<i>tidyr</i>	35
4.1	<code>drop_na()</code> - remove as linhas com NA	35
4.2	<code>replace_na()</code> - Substitui os valores de NA por outro valor	36
4.3	<code>pivot_longer()</code> e <code>pivot_wider()</code> - Aumenta o número de linhas e diminui o número de colunas; aumenta o número de colunas e diminui o número de linhas	36
4.4	<code>separate()</code> e <code>unite()</code> - Separa uma coluna em múltiplas colunas; Une múltiplas colunas	39
4.5	<code>fill()</code> - Preenche as células com NA com o valor posterior ou anterior da mesma coluna	40
5	Integrando os pacotes <i>tibble</i>, <i>dplyr</i>, <i>tidyr</i> e <i>magrittr</i>	41
6	Exercício	45
7	Resposta exercício	49

Chapter 1

Introdução

Para começar precisamos instalar os seguintes pacotes:

```
install.packages("tibble")  
install.packages("dplyr")  
install.packages("tidyr")  
install.packages("magrittr")
```

Ou então, um único pacote **tidyverse**. Que engloba todos os pacotes acima.

```
install.packages("tidyverse")
```

Após instalados precisamos carregá-los. Podemos fazer isso chamando cada um dos pacotes separadamente.

```
library(tibble)  
library(dplyr)  
library(tidyr)  
library(magrittr)
```

OBS: O pacote *magrittr* não faz parte do pacote tidyverse, mas é carregado por ele, dada a importância do operador pipe.

Ou então podemos chamar apenas o pacote *tidyverse*.

```
library(tidyverse)
```

Vamos trabalhar **algumas** funções destes pacotes.

1.0.1 Do pacote *tibble* iremos trabalhar as seguintes funções:

- `add_column()`
- `add_row()`
- `as_tibble()`
- `column_to_rownames()`
- `remove_rownames()`
- `rownames_to_column()`
- `subsetting`

1.0.2 Do pacote *dplyr* iremos trabalhar as seguintes funções:

- `add_count()`
- `arrange()`
- `bind_cols()`
- `bind_rows()`
- `count()`
- `desc()`
- `distinct()`
- `filter()`
- `group_by()`
- `mutate()`
- `na_if()`
- `recode_factor()`
- `relocate()`
- `rename()`
- `select()`
- `slice()`
- `summarise()`
- `transmute()`

1.0.3 Do pacote *tidyr* iremos trabalhar as seguintes funções:

- `drop_na()`
- `fill()`
- `pivot_longer()`
- `pivot_wider()`
- `replace_na()`
- `separate()`
- `unite()`

1.0.4 Do pacote *magrittr* iremos trabalhar com o operador chamado pipe:

- `%>%`¹

1.1 Importando os dados

Caso não souberem importar os dados de um arquivo *.csv ou *.excel presente no seu PC ao fim da aula eu explico. Os exemplos de como as funções trabalham seguem do mesmo jeito.

Vamos trabalhar com conjuntos de dados presentes no R.

```
data(mtcars)
carros <- as_tibble(mtcars, rownames = NA)

data(iris)
flores <- as_tibble(iris)

data(starwars)
starwars <- as_tibble(starwars)
```

¹Vamos utilizar este operador em todos os exemplos

Chapter 2

tibble

2.1 `as_tibble()` - converte um objeto do tipo matriz ou data frame para `tbl_df`

```
obj1 <- carros
obj2 <- carros %>%
  data.frame()

# ["linhas", "colunas"]
obj1[, 1]
```

```
## # A tibble: 32 x 1
##       mpg
##   <dbl>
## 1    21
## 2    21
## 3   22.8
## 4   21.4
## 5   18.7
## 6   18.1
## 7   14.3
## 8   24.4
## 9   22.8
## 10  19.2
## # ... with 22 more rows
```

```
obj2[, 1]
```

```
## [1] 21.0 21.0 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
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

2.2 add_column() - adiciona uma coluna nova a planilha

```
carros %>%
  add_column(teste = 0)
```

```
## # A tibble: 32 x 12
##   mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb  teste
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21     6  160   110  3.9   2.62  16.5    0    1    4     4     0
## 2  21     6  160   110  3.9   2.88  17.0    0    1    4     4     0
## 3  22.8   4  108    93  3.85   2.32  18.6    1    1    4     1     0
## 4  21.4   6  258   110  3.08   3.22  19.4    1    0    3     1     0
## 5  18.7   8  360   175  3.15   3.44  17.0    0    0    3     2     0
## 6  18.1   6  225   105  2.76   3.46  20.2    1    0    3     1     0
## 7  14.3   8  360   245  3.21   3.57  15.8    0    0    3     4     0
## 8  24.4   4  147.    62  3.69   3.19   20     1    0    4     2     0
## 9  22.8   4  141.    95  3.92   3.15  22.9    1    0    4     2     0
## 10 19.2   6  168.   123  3.92   3.44  18.3    1    0    4     4     0
## # ... with 22 more rows
```

```
carros
```

```
## # A tibble: 32 x 11
##   mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
## * <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21     6  160   110  3.9   2.62  16.5    0    1    4     4
## 2  21     6  160   110  3.9   2.88  17.0    0    1    4     4
## 3  22.8   4  108    93  3.85   2.32  18.6    1    1    4     1
## 4  21.4   6  258   110  3.08   3.22  19.4    1    0    3     1
## 5  18.7   8  360   175  3.15   3.44  17.0    0    0    3     2
## 6  18.1   6  225   105  2.76   3.46  20.2    1    0    3     1
## 7  14.3   8  360   245  3.21   3.57  15.8    0    0    3     4
## 8  24.4   4  147.    62  3.69   3.19   20     1    0    4     2
## 9  22.8   4  141.    95  3.92   3.15  22.9    1    0    4     2
```

2.3. ADD_ROW() - ADICIONA UMA NOVA LINHA A PLANILHA 11

```
## 10 19.2      6 168.  123 3.92 3.44 18.3      1      0      4      4
## # ... with 22 more rows
```

```
carros %>%
  add_column(teste = 0, .before = "mpg")
```

```
## # A tibble: 32 x 12
##   teste  mpg  cyl  disp  hp  drat  wt  qsec  vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     0  21      6  160  110  3.9  2.62 16.5    0     1     4     4
## 2     0  21      6  160  110  3.9  2.88 17.0    0     1     4     4
## 3     0 22.8     4  108   93  3.85  2.32 18.6    1     1     4     1
## 4     0 21.4     6  258  110  3.08  3.22 19.4    1     0     3     1
## 5     0 18.7     8  360  175  3.15  3.44 17.0    0     0     3     2
## 6     0 18.1     6  225  105  2.76  3.46 20.2    1     0     3     1
## 7     0 14.3     8  360  245  3.21  3.57 15.8    0     0     3     4
## 8     0 24.4     4  147.   62  3.69  3.19 20      1     0     4     2
## 9     0 22.8     4  141.   95  3.92  3.15 22.9    1     0     4     2
## 10    0 19.2     6  168.  123  3.92  3.44 18.3    1     0     4     4
## # ... with 22 more rows
```

2.3 add_row() - adiciona uma nova linha a planilha

```
carros %>%
  add_row()
```

```
## # A tibble: 33 x 11
##   mpg  cyl  disp  hp  drat  wt  qsec  vs  am  gear  carb
## * <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21      6  160  110  3.9  2.62 16.5    0     1     4     4
## 2  21      6  160  110  3.9  2.88 17.0    0     1     4     4
## 3 22.8     4  108   93  3.85  2.32 18.6    1     1     4     1
## 4 21.4     6  258  110  3.08  3.22 19.4    1     0     3     1
## 5 18.7     8  360  175  3.15  3.44 17.0    0     0     3     2
## 6 18.1     6  225  105  2.76  3.46 20.2    1     0     3     1
## 7 14.3     8  360  245  3.21  3.57 15.8    0     0     3     4
## 8 24.4     4  147.   62  3.69  3.19 20      1     0     4     2
## 9 22.8     4  141.   95  3.92  3.15 22.9    1     0     4     2
## 10 19.2     6  168.  123  3.92  3.44 18.3    1     0     4     4
## # ... with 23 more rows
```

```
carros
```

```
## # A tibble: 32 x 11
##   mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
## * <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21     6  160   110  3.9   2.62  16.5    0   1    4     4
## 2  21     6  160   110  3.9   2.88  17.0    0   1    4     4
## 3  22.8   4  108    93  3.85  2.32  18.6    1   1    4     1
## 4  21.4   6  258   110  3.08  3.22  19.4    1   0    3     1
## 5  18.7   8  360   175  3.15  3.44  17.0    0   0    3     2
## 6  18.1   6  225   105  2.76  3.46  20.2    1   0    3     1
## 7  14.3   8  360   245  3.21  3.57  15.8    0   0    3     4
## 8  24.4   4  147.    62  3.69  3.19   20     1   0    4     2
## 9  22.8   4  141.    95  3.92  3.15  22.9    1   0    4     2
## 10 19.2   6  168.   123  3.92  3.44  18.3    1   0    4     4
## # ... with 22 more rows
```

```
carros %>%
  add_row(.before = 1)
```

```
## # A tibble: 33 x 11
##   mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
## * <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  NA     NA   NA     NA  NA     NA     NA     NA   NA   NA     NA
## 2  21     6  160   110  3.9   2.62  16.5    0   1    4     4
## 3  21     6  160   110  3.9   2.88  17.0    0   1    4     4
## 4  22.8   4  108    93  3.85  2.32  18.6    1   1    4     1
## 5  21.4   6  258   110  3.08  3.22  19.4    1   0    3     1
## 6  18.7   8  360   175  3.15  3.44  17.0    0   0    3     2
## 7  18.1   6  225   105  2.76  3.46  20.2    1   0    3     1
## 8  14.3   8  360   245  3.21  3.57  15.8    0   0    3     4
## 9  24.4   4  147.    62  3.69  3.19   20     1   0    4     2
## 10 22.8   4  141.    95  3.92  3.15  22.9    1   0    4     2
## # ... with 23 more rows
```

2.4 remove_rownames() e column_to_rownames()

- Remove os nomes das colunas e adiciona um nome as colunas

```
carros %>%
  add_column(teste = c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p"))
```

2.5. ROWNAMES_TO_COLUMN() - ADICIONA O NOME DAS LINHAS COMO UMA COLUNA13

```
remove_rownames() %>%  
column_to_rownames("teste")
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## a	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
## b	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
## c	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## d	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## e	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
## f	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
## g	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
## h	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## i	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## j	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
## k	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
## l	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
## m	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
## n	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
## o	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
## p	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
## q	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
## r	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
## s	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
## t	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
## u	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
## v	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
## w	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
## x	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
## y	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
## z	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
## aa	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
## bb	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
## cc	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
## dd	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
## ee	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
## ff	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

2.5 rownames_to_column() - adiciona o nome das linhas como uma coluna

```
carros %>%  
rownames_to_column(var = "teste")
```

```
## # A tibble: 32 x 12
##   teste      mpg   cyl  disp    hp  drat    wt  qsec    vs  am  gear  carb
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Mazda RX4      21     6  160    110  3.9   2.62  16.5     0     1     4     4
## 2 Mazda RX4 ~    21     6  160    110  3.9   2.88  17.0     0     1     4     4
## 3 Datsun 710    22.8     4  108     93  3.85   2.32  18.6     1     1     4     1
## 4 Hornet 4 D~   21.4     6  258    110  3.08   3.22  19.4     1     0     3     1
## 5 Hornet Spo~  18.7     8  360    175  3.15   3.44  17.0     0     0     3     2
## 6 Valiant      18.1     6  225    105  2.76   3.46  20.2     1     0     3     1
## 7 Duster 360    14.3     8  360    245  3.21   3.57  15.8     0     0     3     4
## 8 Merc 240D     24.4     4  147.     62  3.69   3.19   20      1     0     4     2
## 9 Merc 230      22.8     4  141.     95  3.92   3.15  22.9     1     0     4     2
## 10 Merc 280     19.2     6  168.    123  3.92   3.44  18.3     1     0     4     4
## # ... with 22 more rows
```

2.6 subsetting - forma de escrita que permite cortar/selecionar a planilha em função das linhas e colunas

```
# ["linhas", "colunas"]
```

```
carros[1, ]
```

```
## # A tibble: 1 x 11
##   mpg   cyl  disp    hp  drat    wt  qsec    vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    21     6  160    110  3.9   2.62  16.5     0     1     4     4
```

```
carros[2, ]
```

```
## # A tibble: 1 x 11
##   mpg   cyl  disp    hp  drat    wt  qsec    vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    21     6  160    110  3.9   2.88  17.0     0     1     4     4
```

```
carros[1:2, ]
```

```
## # A tibble: 2 x 11
##   mpg   cyl  disp    hp  drat    wt  qsec    vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    21     6  160    110  3.9   2.62  16.5     0     1     4     4
## 2    21     6  160    110  3.9   2.88  17.0     0     1     4     4
```

2.6. SUBSETTING - FORMA DE ESCRITA QUE PERMITE CORTAR/SELECIONAR A PLANILHA EM FUNÇÃO

```
carros[, 1]
```

```
## # A tibble: 32 x 1
##       mpg
##   <dbl>
## 1    21
## 2    21
## 3   22.8
## 4   21.4
## 5   18.7
## 6   18.1
## 7   14.3
## 8   24.4
## 9   22.8
## 10  19.2
## # ... with 22 more rows
```

```
carros[, "mpg"]
```

```
## # A tibble: 32 x 1
##       mpg
##   <dbl>
## 1    21
## 2    21
## 3   22.8
## 4   21.4
## 5   18.7
## 6   18.1
## 7   14.3
## 8   24.4
## 9   22.8
## 10  19.2
## # ... with 22 more rows
```

```
carros[[1]]
```

```
## [1] 21.0 21.0 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
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
carros[["mpg"]]
```

```
## [1] 21.0 21.0 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
```

```
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
carros$mpg
```

```
## [1] 21.0 21.0 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
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
carros[, c("mpg", "cyl")]
```

```
## # A tibble: 32 x 2
##   mpg    cyl
##   <dbl> <dbl>
## 1  21      6
## 2  21      6
## 3 22.8     4
## 4 21.4     6
## 5 18.7     8
## 6 18.1     6
## 7 14.3     8
## 8 24.4     4
## 9 22.8     4
## 10 19.2     6
## # ... with 22 more rows
```


Chapter 3

dplyr

3.1 count() e add_count() - conta o número de linhas em função da variável especificada; adiciona uma nova coluna com a contagem do número de linhas em função da variável especificada

```
flores %>%  
  count(Species)
```

```
## # A tibble: 3 x 2  
##   Species      n  
##   <fct>    <int>  
## 1 setosa      50  
## 2 versicolor  50  
## 3 virginica   50
```

```
flores %>%  
  add_count(Species)
```

```
## # A tibble: 150 x 6  
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species      n  
##           <dbl>      <dbl>      <dbl>      <dbl> <fct>    <int>  
## 1         5.1        3.5        1.4        0.2 setosa     50  
## 2         4.9         3         1.4        0.2 setosa     50
```

```
## 3      4.7      3.2      1.3      0.2 setosa  50
## 4      4.6      3.1      1.5      0.2 setosa  50
## 5      5      3.6      1.4      0.2 setosa  50
## 6      5.4      3.9      1.7      0.4 setosa  50
## 7      4.6      3.4      1.4      0.3 setosa  50
## 8      5      3.4      1.5      0.2 setosa  50
## 9      4.4      2.9      1.4      0.2 setosa  50
## 10     4.9      3.1      1.5      0.1 setosa  50
## # ... with 140 more rows
```

3.2 arrange() - rearranja as linhas em função da variável especificada

```
flores %>%
  arrange(Species)
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1      5.1      3.5      1.4      0.2 setosa
## 2      4.9      3      1.4      0.2 setosa
## 3      4.7      3.2      1.3      0.2 setosa
## 4      4.6      3.1      1.5      0.2 setosa
## 5      5      3.6      1.4      0.2 setosa
## 6      5.4      3.9      1.7      0.4 setosa
## 7      4.6      3.4      1.4      0.3 setosa
## 8      5      3.4      1.5      0.2 setosa
## 9      4.4      2.9      1.4      0.2 setosa
## 10     4.9      3.1      1.5      0.1 setosa
## # ... with 140 more rows
```

```
flores %>%
  arrange(desc(Species))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1      6.3      3.3      6      2.5 virginica
## 2      5.8      2.7      5.1      1.9 virginica
## 3      7.1      3      5.9      2.1 virginica
## 4      6.3      2.9      5.6      1.8 virginica
```

```
## 5      6.5      3      5.8      2.2 virginica
## 6      7.6      3      6.6      2.1 virginica
## 7      4.9      2.5     4.5      1.7 virginica
## 8      7.3      2.9     6.3      1.8 virginica
## 9      6.7      2.5     5.8      1.8 virginica
## 10     7.2      3.6     6.1      2.5 virginica
## # ... with 140 more rows
```

```
flores %>%
  arrange(factor(Species, levels = c("versicolor", "setosa", "virginica")))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1           7         3.2         4.7         1.4 versicolor
## 2          6.4         3.2         4.5         1.5 versicolor
## 3          6.9         3.1         4.9         1.5 versicolor
## 4          5.5         2.3          4         1.3 versicolor
## 5          6.5         2.8         4.6         1.5 versicolor
## 6          5.7         2.8         4.5         1.3 versicolor
## 7          6.3         3.3         4.7         1.6 versicolor
## 8          4.9         2.4         3.3          1 versicolor
## 9          6.6         2.9         4.6         1.3 versicolor
## 10         5.2         2.7         3.9         1.4 versicolor
## # ... with 140 more rows
```

3.3 `bind_cols()` - une planilhas em colunas

```
parte1 <- flores[,1:2]
parte2 <- flores[,3:5]

bind_cols(parte1, parte2)
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1          5.1         3.5         1.4         0.2 setosa
## 2          4.9          3         1.4         0.2 setosa
## 3          4.7         3.2         1.3         0.2 setosa
## 4          4.6         3.1         1.5         0.2 setosa
## 5           5         3.6         1.4         0.2 setosa
## 6          5.4         3.9         1.7         0.4 setosa
```

```
## 7      4.6      3.4      1.4      0.3 setosa
## 8      5      3.4      1.5      0.2 setosa
## 9      4.4      2.9      1.4      0.2 setosa
## 10     4.9      3.1      1.5      0.1 setosa
## # ... with 140 more rows
```

3.4 bind_rows() - une planilhas em linhas

```
parte1 <- flores[1:10,]
parte2 <- flores[11:150,]

bind_rows(parte1, parte2)
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1      5.1      3.5      1.4      0.2 setosa
## 2      4.9      3      1.4      0.2 setosa
## 3      4.7      3.2      1.3      0.2 setosa
## 4      4.6      3.1      1.5      0.2 setosa
## 5      5      3.6      1.4      0.2 setosa
## 6      5.4      3.9      1.7      0.4 setosa
## 7      4.6      3.4      1.4      0.3 setosa
## 8      5      3.4      1.5      0.2 setosa
## 9      4.4      2.9      1.4      0.2 setosa
## 10     4.9      3.1      1.5      0.1 setosa
## # ... with 140 more rows
```

3.5 distinct() - remove as linhas que são exatamente iguais

```
flores
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1      5.1      3.5      1.4      0.2 setosa
## 2      4.9      3      1.4      0.2 setosa
## 3      4.7      3.2      1.3      0.2 setosa
```

3.6. `FILTER()` - FILTRA AS LINHAS QUE SATISFAÇÃO ALGUMA CONDIÇÃO

```
## 4      4.6      3.1      1.5      0.2 setosa
## 5      5       3.6      1.4      0.2 setosa
## 6      5.4      3.9      1.7      0.4 setosa
## 7      4.6      3.4      1.4      0.3 setosa
## 8      5       3.4      1.5      0.2 setosa
## 9      4.4      2.9      1.4      0.2 setosa
## 10     4.9      3.1      1.5      0.1 setosa
## # ... with 140 more rows
```

```
flores %>%
  distinct()
```

```
## # A tibble: 149 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1      5.1      3.5          1.4          0.2 setosa
## 2      4.9      3           1.4          0.2 setosa
## 3      4.7      3.2          1.3          0.2 setosa
## 4      4.6      3.1          1.5          0.2 setosa
## 5      5       3.6          1.4          0.2 setosa
## 6      5.4      3.9          1.7          0.4 setosa
## 7      4.6      3.4          1.4          0.3 setosa
## 8      5       3.4          1.5          0.2 setosa
## 9      4.4      2.9          1.4          0.2 setosa
## 10     4.9      3.1          1.5          0.1 setosa
## # ... with 139 more rows
```

```
flores %>%
  distinct() %>%
  count(Species)
```

```
## # A tibble: 3 x 2
##   Species      n
##   <fct>    <int>
## 1 setosa    50
## 2 versicolor 50
## 3 virginica  49
```

3.6 `filter()` - filtra as linhas que satisfação alguma condição

```
flores %>%
  filter(Species == "setosa")
```

```
## # A tibble: 50 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         5.1         3.5         1.4         0.2 setosa
## 2         4.9         3         1.4         0.2 setosa
## 3         4.7         3.2         1.3         0.2 setosa
## 4         4.6         3.1         1.5         0.2 setosa
## 5         5         3.6         1.4         0.2 setosa
## 6         5.4         3.9         1.7         0.4 setosa
## 7         4.6         3.4         1.4         0.3 setosa
## 8         5         3.4         1.5         0.2 setosa
## 9         4.4         2.9         1.4         0.2 setosa
## 10        4.9         3.1         1.5         0.1 setosa
## # ... with 40 more rows
```

```
flores %>%
  filter(Species %in% c("setosa", "virginica"))
```

```
## # A tibble: 100 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         5.1         3.5         1.4         0.2 setosa
## 2         4.9         3         1.4         0.2 setosa
## 3         4.7         3.2         1.3         0.2 setosa
## 4         4.6         3.1         1.5         0.2 setosa
## 5         5         3.6         1.4         0.2 setosa
## 6         5.4         3.9         1.7         0.4 setosa
## 7         4.6         3.4         1.4         0.3 setosa
## 8         5         3.4         1.5         0.2 setosa
## 9         4.4         2.9         1.4         0.2 setosa
## 10        4.9         3.1         1.5         0.1 setosa
## # ... with 90 more rows
```

```
flores %>%
  filter(Sepal.Length >= mean(Sepal.Length))
```

```
## # A tibble: 70 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         7         3.2         4.7         1.4 versicolor
```

3.6. FILTER() - FILTRA AS LINHAS QUE SATISFAÇÃO ALGUMA CONDIÇÃO

```
## 2      6.4      3.2      4.5      1.5 versicolor
## 3      6.9      3.1      4.9      1.5 versicolor
## 4      6.5      2.8      4.6      1.5 versicolor
## 5      6.3      3.3      4.7      1.6 versicolor
## 6      6.6      2.9      4.6      1.3 versicolor
## 7      5.9      3      4.2      1.5 versicolor
## 8      6      2.2      4      1 versicolor
## 9      6.1      2.9      4.7      1.4 versicolor
## 10     6.7      3.1      4.4      1.4 versicolor
## # ... with 60 more rows
```

```
flores %>%
  filter(Species == "setosa") %>%
  filter(Sepal.Length >= mean(Sepal.Length))
```

```
## # A tibble: 22 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1      5.1      3.5      1.4      0.2 setosa
## 2      5.4      3.9      1.7      0.4 setosa
## 3      5.4      3.7      1.5      0.2 setosa
## 4      5.8      4      1.2      0.2 setosa
## 5      5.7      4.4      1.5      0.4 setosa
## 6      5.4      3.9      1.3      0.4 setosa
## 7      5.1      3.5      1.4      0.3 setosa
## 8      5.7      3.8      1.7      0.3 setosa
## 9      5.1      3.8      1.5      0.3 setosa
## 10     5.4      3.4      1.7      0.2 setosa
## # ... with 12 more rows
```

```
flores %>%
  filter(Petal.Length >= mean(Petal.Length) & Petal.Width >= mean(Petal.Width))
```

```
## # A tibble: 89 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1      7      3.2      4.7      1.4 versicolor
## 2      6.4      3.2      4.5      1.5 versicolor
## 3      6.9      3.1      4.9      1.5 versicolor
## 4      5.5      2.3      4      1.3 versicolor
## 5      6.5      2.8      4.6      1.5 versicolor
## 6      5.7      2.8      4.5      1.3 versicolor
## 7      6.3      3.3      4.7      1.6 versicolor
## 8      6.6      2.9      4.6      1.3 versicolor
```

```
## 9          5.2          2.7          3.9          1.4 versicolor
## 10         5.9          3          4.2          1.5 versicolor
## # ... with 79 more rows
```

```
flores %>%
  filter(Petal.Length >= mean(Petal.Length) & Petal.Width >= mean(Petal.Width)) %>%
  count(Species)
```

```
## # A tibble: 2 x 2
##   Species      n
##   <fct>    <int>
## 1 versicolor    39
## 2 virginica     50
```

3.7 group_by() - agrupa as linhas em função dos valores de alguma variável

#Ver esta função juntamente com mutate(), transmute() e summarise()

```
flores %>%
  group_by(Species)
```

```
## # A tibble: 150 x 5
## # Groups:   Species [3]
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1          5.1          3.5          1.4          0.2 setosa
## 2          4.9          3          1.4          0.2 setosa
## 3          4.7          3.2          1.3          0.2 setosa
## 4          4.6          3.1          1.5          0.2 setosa
## 5          5          3.6          1.4          0.2 setosa
## 6          5.4          3.9          1.7          0.4 setosa
## 7          4.6          3.4          1.4          0.3 setosa
## 8          5          3.4          1.5          0.2 setosa
## 9          4.4          2.9          1.4          0.2 setosa
## 10         4.9          3.1          1.5          0.1 setosa
## # ... with 140 more rows
```


3.8. `MUTATE()` E `TRANSMUTE()` - ADICIONA NOVAS VARIÁVEIS E PRESERVA AS EXISTENTES; ADICIONA NOVAS VARIÁVEIS E NÃO PRESERVA AS EXISTENTES

3.8 `mutate()` e `transmute()` - adiciona novas variáveis e preserva as existentes; adiciona novas variáveis e não preserva as existentes

```
flores %>%  
  mutate(teste = Petal.Length + Sepal.Length)
```

```
## # A tibble: 150 x 6  
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species teste  
##           <dbl>      <dbl>      <dbl>      <dbl> <fct>   <dbl>  
## 1         5.1         3.5         1.4         0.2 setosa    6.5  
## 2         4.9         3         1.4         0.2 setosa    6.3  
## 3         4.7         3.2         1.3         0.2 setosa    6  
## 4         4.6         3.1         1.5         0.2 setosa    6.1  
## 5         5         3.6         1.4         0.2 setosa    6.4  
## 6         5.4         3.9         1.7         0.4 setosa    7.1  
## 7         4.6         3.4         1.4         0.3 setosa    6  
## 8         5         3.4         1.5         0.2 setosa    6.5  
## 9         4.4         2.9         1.4         0.2 setosa    5.8  
## 10        4.9         3.1         1.5         0.1 setosa    6.4  
## # ... with 140 more rows
```

```
flores %>%  
  group_by(Species) %>%  
  mutate(PL_media = mean(Petal.Length))
```

```
## # A tibble: 150 x 6  
## # Groups:   Species [3]  
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species PL_media  
##           <dbl>      <dbl>      <dbl>      <dbl> <fct>   <dbl>  
## 1         5.1         3.5         1.4         0.2 setosa    1.46  
## 2         4.9         3         1.4         0.2 setosa    1.46  
## 3         4.7         3.2         1.3         0.2 setosa    1.46  
## 4         4.6         3.1         1.5         0.2 setosa    1.46  
## 5         5         3.6         1.4         0.2 setosa    1.46  
## 6         5.4         3.9         1.7         0.4 setosa    1.46  
## 7         4.6         3.4         1.4         0.3 setosa    1.46  
## 8         5         3.4         1.5         0.2 setosa    1.46  
## 9         4.4         2.9         1.4         0.2 setosa    1.46  
## 10        4.9         3.1         1.5         0.1 setosa    1.46  
## # ... with 140 more rows
```

```
flores %>%
  group_by(Species) %>%
  transmute(PL.média = mean(Petal.Length))
```

```
## # A tibble: 150 x 2
## # Groups:   Species [3]
##   Species PL.média
##   <fct>     <dbl>
## 1 setosa     1.46
## 2 setosa     1.46
## 3 setosa     1.46
## 4 setosa     1.46
## 5 setosa     1.46
## 6 setosa     1.46
## 7 setosa     1.46
## 8 setosa     1.46
## 9 setosa     1.46
## 10 setosa    1.46
## # ... with 140 more rows
```

```
flores %>%
  group_by(Species) %>%
  transmute(PL.média = mean(Petal.Length),
            SL.média = mean(Sepal.Length))
```

```
## # A tibble: 150 x 3
## # Groups:   Species [3]
##   Species PL.média SL.média
##   <fct>     <dbl>     <dbl>
## 1 setosa     1.46       5.01
## 2 setosa     1.46       5.01
## 3 setosa     1.46       5.01
## 4 setosa     1.46       5.01
## 5 setosa     1.46       5.01
## 6 setosa     1.46       5.01
## 7 setosa     1.46       5.01
## 8 setosa     1.46       5.01
## 9 setosa     1.46       5.01
## 10 setosa    1.46       5.01
## # ... with 140 more rows
```

3.9 na_if() - Substitui o valor especificado por NA

```
flores %>%
  mutate(Species = na_if(Species, "setosa"))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         5.1         3.5         1.4         0.2 <NA>
## 2         4.9         3          1.4         0.2 <NA>
## 3         4.7         3.2         1.3         0.2 <NA>
## 4         4.6         3.1         1.5         0.2 <NA>
## 5         5          3.6         1.4         0.2 <NA>
## 6         5.4         3.9         1.7         0.4 <NA>
## 7         4.6         3.4         1.4         0.3 <NA>
## 8         5          3.4         1.5         0.2 <NA>
## 9         4.4         2.9         1.4         0.2 <NA>
## 10        4.9         3.1         1.5         0.1 <NA>
## # ... with 140 more rows
```

```
flores %>%
  mutate(Petal.Length = na_if(Petal.Length, 1.4))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         5.1         3.5         NA          0.2 setosa
## 2         4.9         3          NA          0.2 setosa
## 3         4.7         3.2         1.3         0.2 setosa
## 4         4.6         3.1         1.5         0.2 setosa
## 5         5          3.6         NA          0.2 setosa
## 6         5.4         3.9         1.7         0.4 setosa
## 7         4.6         3.4         NA          0.3 setosa
## 8         5          3.4         1.5         0.2 setosa
## 9         4.4         2.9         NA          0.2 setosa
## 10        4.9         3.1         1.5         0.1 setosa
## # ... with 140 more rows
```

```
flores %>%
  mutate(Sepal.Length = na_if(Sepal.Length, 5.1))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1           NA         3.5           1.4           0.2 setosa
## 2           4.9         3             1.4           0.2 setosa
## 3           4.7         3.2           1.3           0.2 setosa
## 4           4.6         3.1           1.5           0.2 setosa
## 5           5           3.6           1.4           0.2 setosa
## 6           5.4         3.9           1.7           0.4 setosa
## 7           4.6         3.4           1.4           0.3 setosa
## 8           5           3.4           1.5           0.2 setosa
## 9           4.4         2.9           1.4           0.2 setosa
## 10          4.9         3.1           1.5           0.1 setosa
## # ... with 140 more rows
```

3.10 recode() e recode_factor() - substitui um determinado valor por outro, se variável for numérica usar recode(), se for fator usar recode_factor()

```
flores %>%
  mutate(Sepal.Length = recode(Sepal.Length, `5.1` = 0))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1           0         3.5           1.4           0.2 setosa
## 2           4.9         3             1.4           0.2 setosa
## 3           4.7         3.2           1.3           0.2 setosa
## 4           4.6         3.1           1.5           0.2 setosa
## 5           5           3.6           1.4           0.2 setosa
## 6           5.4         3.9           1.7           0.4 setosa
## 7           4.6         3.4           1.4           0.3 setosa
## 8           5           3.4           1.5           0.2 setosa
## 9           4.4         2.9           1.4           0.2 setosa
## 10          4.9         3.1           1.5           0.1 setosa
## # ... with 140 more rows
```

```
flores %>%
  mutate(Species = recode_factor(Species, setosa = "sts"))
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1         5.1         3.5         1.4         0.2 sts
## 2         4.9         3         1.4         0.2 sts
## 3         4.7         3.2         1.3         0.2 sts
## 4         4.6         3.1         1.5         0.2 sts
## 5         5         3.6         1.4         0.2 sts
## 6         5.4         3.9         1.7         0.4 sts
## 7         4.6         3.4         1.4         0.3 sts
## 8         5         3.4         1.5         0.2 sts
## 9         4.4         2.9         1.4         0.2 sts
## 10        4.9         3.1         1.5         0.1 sts
## # ... with 140 more rows
```

3.11 relocate() - altera a ordem das variáveis

```
flores %>%
  relocate(Species, .before = Sepal.Length)
```

```
## # A tibble: 150 x 5
##   Species Sepal.Length Sepal.Width Petal.Length Petal.Width
##   <fct>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 setosa         5.1         3.5         1.4         0.2
## 2 setosa         4.9         3         1.4         0.2
## 3 setosa         4.7         3.2         1.3         0.2
## 4 setosa         4.6         3.1         1.5         0.2
## 5 setosa         5         3.6         1.4         0.2
## 6 setosa         5.4         3.9         1.7         0.4
## 7 setosa         4.6         3.4         1.4         0.3
## 8 setosa         5         3.4         1.5         0.2
## 9 setosa         4.4         2.9         1.4         0.2
## 10 setosa        4.9         3.1         1.5         0.1
## # ... with 140 more rows
```

3.12 rename() - altera o nome das variáveis

```
flores %>%
  rename(sp = Species)
```

```
## # A tibble: 150 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width sp
##         <dbl>         <dbl>         <dbl>         <dbl> <fct>
## 1         5.1           3.5           1.4           0.2 setosa
## 2         4.9           3             1.4           0.2 setosa
## 3         4.7           3.2           1.3           0.2 setosa
## 4         4.6           3.1           1.5           0.2 setosa
## 5         5             3.6           1.4           0.2 setosa
## 6         5.4           3.9           1.7           0.4 setosa
## 7         4.6           3.4           1.4           0.3 setosa
## 8         5             3.4           1.5           0.2 setosa
## 9         4.4           2.9           1.4           0.2 setosa
## 10        4.9           3.1           1.5           0.1 setosa
## # ... with 140 more rows
```

3.13 select() - Seleciona variáveis

```
flores %>%
  select(Species)
```

```
## # A tibble: 150 x 1
##   Species
##   <fct>
## 1 setosa
## 2 setosa
## 3 setosa
## 4 setosa
## 5 setosa
## 6 setosa
## 7 setosa
## 8 setosa
## 9 setosa
## 10 setosa
## # ... with 140 more rows
```

```
flores %>%
  select(starts_with("Sepal"))
```

```
## # A tibble: 150 x 2
##   Sepal.Length Sepal.Width
##         <dbl>         <dbl>
## 1         5.1           3.5
```

```
## 2      4.9      3
## 3      4.7      3.2
## 4      4.6      3.1
## 5      5      3.6
## 6      5.4      3.9
## 7      4.6      3.4
## 8      5      3.4
## 9      4.4      2.9
## 10     4.9      3.1
## # ... with 140 more rows
```

```
flores %>%
  select(ends_with("Length"))
```

```
## # A tibble: 150 x 2
##   Sepal.Length Petal.Length
##   <dbl>         <dbl>
## 1      5.1      1.4
## 2      4.9      1.4
## 3      4.7      1.3
## 4      4.6      1.5
## 5      5      1.4
## 6      5.4      1.7
## 7      4.6      1.4
## 8      5      1.5
## 9      4.4      1.4
## 10     4.9      1.5
## # ... with 140 more rows
```

```
flores %>%
  select(Species, Sepal.Length)
```

```
## # A tibble: 150 x 2
##   Species Sepal.Length
##   <fct>         <dbl>
## 1 setosa      5.1
## 2 setosa      4.9
## 3 setosa      4.7
## 4 setosa      4.6
## 5 setosa      5
## 6 setosa      5.4
## 7 setosa      4.6
## 8 setosa      5
## 9 setosa      4.4
```

```
## 10 setosa          4.9
## # ... with 140 more rows
```

3.14 slice() - Selecciona linhas

```
flores %>%
  slice(3:15)
```

```
## # A tibble: 13 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         4.7         3.2         1.3         0.2 setosa
## 2         4.6         3.1         1.5         0.2 setosa
## 3         5         3.6         1.4         0.2 setosa
## 4         5.4         3.9         1.7         0.4 setosa
## 5         4.6         3.4         1.4         0.3 setosa
## 6         5         3.4         1.5         0.2 setosa
## 7         4.4         2.9         1.4         0.2 setosa
## 8         4.9         3.1         1.5         0.1 setosa
## 9         5.4         3.7         1.5         0.2 setosa
## 10        4.8         3.4         1.6         0.2 setosa
## 11        4.8         3         1.4         0.1 setosa
## 12        4.3         3         1.1         0.1 setosa
## 13        5.8         4         1.2         0.2 setosa
```

```
flores %>%
  slice_sample(n = 10)
```

```
## # A tibble: 10 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         6.3         2.5         5         1.9 virginica
## 2         7         3.2         4.7         1.4 versicolor
## 3         6.3         2.3         4.4         1.3 versicolor
## 4         5.4         3.9         1.7         0.4 setosa
## 5         5         3.2         1.2         0.2 setosa
## 6         4.8         3.1         1.6         0.2 setosa
## 7         5.6         3         4.5         1.5 versicolor
## 8         4.9         3         1.4         0.2 setosa
## 9         6         2.2         4         1 versicolor
## 10        6.4         3.1         5.5         1.8 virginica
```



```
flores %>%
  slice_min(Petal.Length, n = 10)
```

```
## # A tibble: 11 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         4.6         3.6          1          0.2 setosa
## 2         4.3         3           1.1        0.1 setosa
## 3         5.8         4           1.2        0.2 setosa
## 4         5         3.2          1.2        0.2 setosa
## 5         4.7         3.2          1.3        0.2 setosa
## 6         5.4         3.9          1.3        0.4 setosa
## 7         5.5         3.5          1.3        0.2 setosa
## 8         4.4         3           1.3        0.2 setosa
## 9         5         3.5          1.3        0.3 setosa
## 10        4.5         2.3          1.3        0.3 setosa
## 11        4.4         3.2          1.3        0.2 setosa
```

```
flores %>%
  slice_max(Petal.Length, n = 10)
```

```
## # A tibble: 11 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <fct>
## 1         7.7         2.6         6.9         2.3 virginica
## 2         7.7         3.8         6.7         2.2 virginica
## 3         7.7         2.8         6.7         2   virginica
## 4         7.6         3         6.6         2.1 virginica
## 5         7.9         3.8         6.4         2   virginica
## 6         7.3         2.9         6.3         1.8 virginica
## 7         7.2         3.6         6.1         2.5 virginica
## 8         7.4         2.8         6.1         1.9 virginica
## 9         7.7         3         6.1         2.3 virginica
## 10        6.3         3.3         6         2.5 virginica
## 11        7.2         3.2         6         1.8 virginica
```

3.15 summarise() - sumariza os dados

```
flores %>%
  group_by(Species) %>%
  summarise(N = n(),
```

```
PL.média = mean(Petal.Length),  
SL.média = mean(Sepal.Length),  
PW.média = mean(Petal.Width),  
SW.média = mean(Sepal.Width))
```

```
## # A tibble: 3 x 6  
##   Species      N PL.média SL.média PW.média SW.média  
##   <fct>    <int>   <dbl>   <dbl>   <dbl>   <dbl>  
## 1 setosa     50    1.46    5.01    0.246    3.43  
## 2 versicolor 50    4.26    5.94    1.33     2.77  
## 3 virginica  50    5.55    6.59    2.03     2.97
```

Chapter 4

tidyr

4.1 `drop_na()` - remove as linhas com NA

```
starwars %>%  
  select(hair_color) %>%  
  drop_na()
```

```
## # A tibble: 82 x 1  
##   hair_color  
##   <chr>  
## 1 blond  
## 2 none  
## 3 brown  
## 4 brown, grey  
## 5 brown  
## 6 black  
## 7 auburn, white  
## 8 blond  
## 9 auburn, grey  
## 10 brown  
## # ... with 72 more rows
```

```
starwars %>%  
  drop_na()
```

```
## # A tibble: 6 x 14  
##   name      height  mass hair_color skin_color eye_color birth_year sex  gender
```

```
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>      <dbl> <chr> <chr>
## 1 Luke Sk~   172    77 blond      fair      blue      19   male masculi~
## 2 Obi-Wan~   182    77 auburn, wh~ fair      blue-gray  57   male masculi~
## 3 Anakin ~   188    84 blond      fair      blue      41.9 male masculi~
## 4 Chewbac~  228   112 brown      unknown   blue      200  male masculi~
## 5 Wedge A~   170    77 brown      fair      hazel     21   male masculi~
## 6 Darth M~   175    80 none       red       yellow    54   male masculi~
## # ... with 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

4.2 replace_na() - Substitui os valores de NA por outro valor

```
starwars %>%
  replace_na(list(hair_color = "orange"))
```

```
## # A tibble: 87 x 14
##   name      height mass hair_color skin_color eye_color birth_year sex  gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>      <dbl> <chr> <chr>
## 1 Luke S~   172    77 blond      fair      blue      19   male masculi~
## 2 C-3P0     167    75 orange     gold      yellow    112  none masculi~
## 3 R2-D2      96    32 orange     white, bl~ red      33   none masculi~
## 4 Darth ~   202   136 none       white     yellow    41.9 male masculi~
## 5 Leia O~   150    49 brown      light     brown     19   fema~ femin~
## 6 Owen L~   178   120 brown, grey light     blue      52   male masculi~
## 7 Beru W~   165    75 brown      light     blue      47   fema~ femin~
## 8 R5-D4      97    32 orange     white, red red      NA   none masculi~
## 9 Biggs ~   183    84 black      light     brown     24   male masculi~
## 10 Obi-Wa~  182    77 auburn, wh~ fair      blue-gray  57   male masculi~
## # ... with 77 more rows, and 5 more variables: homeworld <chr>, species <chr>,
## #   films <list>, vehicles <list>, starships <list>
```

4.3 pivot_longer() e pivot_wider() - Aumenta o número de linhas e diminui o número de colunas; aumenta o número de colunas e diminui o número de linhas

4.3. PIVOT_LONGER() E PIVOT_WIDER() - AUMENTA O NÚMERO DE LINHAS E DIMINUI O NÚMERO DE COLUNAS

```
starwars[1:10,] %>%  
  select(homeworld, skin_color, mass) %>%  
  pivot_wider(names_from = homeworld, values_from = mass, values_fn = list)
```

```
## # A tibble: 6 x 5  
##   skin_color Tatooine Naboo Alderaan Stewjon  
##   <chr>      <list>  <list>  <list>  <list>  
## 1 fair      <dbl [1]> <NULL>  <NULL>  <dbl [1]>  
## 2 gold      <dbl [1]> <NULL>  <NULL>  <NULL>  
## 3 white, blue <NULL>   <dbl [1]> <NULL>  <NULL>  
## 4 white     <dbl [1]> <NULL>  <NULL>  <NULL>  
## 5 light     <dbl [3]> <NULL>  <dbl [1]> <NULL>  
## 6 white, red <dbl [1]> <NULL>  <NULL>  <NULL>
```

```
starwars[1:10,] %>%  
  select(homeworld, skin_color, mass) %>%  
  pivot_wider(names_from = homeworld, values_from = mass, values_fn = list)
```

```
## # A tibble: 6 x 5  
##   skin_color Tatooine Naboo Alderaan Stewjon  
##   <chr>      <list>  <list>  <list>  <list>  
## 1 fair      <dbl [1]> <NULL>  <NULL>  <dbl [1]>  
## 2 gold      <dbl [1]> <NULL>  <NULL>  <NULL>  
## 3 white, blue <NULL>   <dbl [1]> <NULL>  <NULL>  
## 4 white     <dbl [1]> <NULL>  <NULL>  <NULL>  
## 5 light     <dbl [3]> <NULL>  <dbl [1]> <NULL>  
## 6 white, red <dbl [1]> <NULL>  <NULL>  <NULL>
```

```
starwars[1:10,] %>%  
  select(homeworld, skin_color, mass) %>%  
  pivot_wider(names_from = homeworld, values_from = mass) %>%  
  unchop(c(2:5))
```

```
## # A tibble: 8 x 5  
##   skin_color Tatooine Naboo Alderaan Stewjon  
##   <chr>      <dbl> <dbl>  <dbl>  <dbl>  
## 1 fair      77    NA    NA     77  
## 2 gold      75    NA    NA     NA  
## 3 white, blue NA    32    NA     NA  
## 4 white     136   NA    NA     NA  
## 5 light     120   NA    49     NA  
## 6 light      75    NA    49     NA  
## 7 light      84    NA    49     NA  
## 8 white, red  32    NA    NA     NA
```

```
starwars[1:10,] %>%
  select(skin_color, homeworld, mass) %>%
  pivot_wider(names_from = homeworld, values_from = mass) %>%
  pivot_longer(cols = 2:5, names_to = "homeworld", values_to = "mass")
```

```
## # A tibble: 24 x 3
##   skin_color homeworld mass
##   <chr>      <chr>    <list>
## 1 fair      Tatooine  <dbl [1]>
## 2 fair      Naboo    <NULL>
## 3 fair      Alderaan <NULL>
## 4 fair      Stewjon  <dbl [1]>
## 5 gold      Tatooine  <dbl [1]>
## 6 gold      Naboo    <NULL>
## 7 gold      Alderaan <NULL>
## 8 gold      Stewjon  <NULL>
## 9 white, blue Tatooine  <NULL>
## 10 white, blue Naboo    <dbl [1]>
## # ... with 14 more rows
```

```
starwars[1:10,] %>%
  select(homeworld, skin_color, mass) %>%
  pivot_wider(names_from = homeworld, values_from = mass) %>%
  pivot_longer(cols = 2:5, names_to = "homeworld", values_to = "mass") %>%
  unchop(everything()) %>%
  drop_na()
```

```
## # A tibble: 10 x 3
##   skin_color homeworld mass
##   <chr>      <chr>    <dbl>
## 1 fair      Tatooine    77
## 2 fair      Stewjon    77
## 3 gold      Tatooine    75
## 4 white, blue Naboo     32
## 5 white      Tatooine   136
## 6 light      Tatooine   120
## 7 light      Tatooine    75
## 8 light      Tatooine    84
## 9 light      Alderaan    49
## 10 white, red Tatooine    32
```

4.4. SEPARATE() E UNITE() - SEPARA UMA COLUNA EM MÚLTIPLAS COLUNAS; UNE MÚLTIPLAS COLUNAS

4.4 separate() e unite() - Separa uma coluna em múltiplas colunas; Une múltiplas colunas

```
starwars[1:10,] %>%  
  select(sex, gender, homeworld) %>%  
  unite("sexgender", sex:gender, sep = "-")
```

```
## # A tibble: 10 x 2  
##   sexgender      homeworld  
##   <chr>         <chr>  
## 1 male-masculine Tatooine  
## 2 none-masculine Tatooine  
## 3 none-masculine Naboo  
## 4 male-masculine Tatooine  
## 5 female-feminine Alderaan  
## 6 male-masculine Tatooine  
## 7 female-feminine Tatooine  
## 8 none-masculine Tatooine  
## 9 male-masculine Tatooine  
## 10 male-masculine Stewjon
```

```
starwars[1:10,] %>%  
  select(sex, gender, homeworld) %>%  
  unite("sexgender", sex:gender, sep = "-") %>%  
  separate(sexgender, c("sex", "gender"), sep = "-")
```

```
## # A tibble: 10 x 3  
##   sex  gender homeworld  
##   <chr> <chr>    <chr>  
## 1 male  masculine Tatooine  
## 2 none  masculine Tatooine  
## 3 none  masculine Naboo  
## 4 male  masculine Tatooine  
## 5 female feminine Alderaan  
## 6 male  masculine Tatooine  
## 7 female feminine Tatooine  
## 8 none  masculine Tatooine  
## 9 male  masculine Tatooine  
## 10 male  masculine Stewjon
```

4.5 fill() - Preenche as células com NA com o valor posterior ou anterior da mesma coluna

```
starwars %>%  
  select(hair_color) %>%  
  fill(hair_color)
```

```
## # A tibble: 87 x 1  
##   hair_color  
##   <chr>  
## 1 blond  
## 2 blond  
## 3 blond  
## 4 none  
## 5 brown  
## 6 brown, grey  
## 7 brown  
## 8 brown  
## 9 black  
## 10 auburn, white  
## # ... with 77 more rows
```


Chapter 5

Integrando os pacotes *tibble*, *dplyr*, *tidyr* e *magrittr*

Importar o arquivo excel de nome “dados”: (<https://github.com/wilson0106/tidyverse/blob/master/dados.xlsx>)

```
dados %>%  
  select(1:4) %>%  
  distinct() %>%  
  filter(ano == "2020") %>%  
  pivot_wider(id_cols = c(ano, mes),  
              names_from = especie,  
              values_from = abundancia)
```

```
## # A tibble: 12 x 5  
##      ano mes      sp1    sp2    sp3  
##    <dbl> <chr>    <dbl> <dbl> <dbl>  
##  1 2020 janeiro    132   105   80  
##  2 2020 fevereiro    56     3    2  
##  3 2020 março       41     4    8  
##  4 2020 abril        5    85  166  
##  5 2020 maio        72   152   35  
##  6 2020 junho       15    38  148  
##  7 2020 julho       57   184  141  
##  8 2020 agosto       74    NA   55  
##  9 2020 setembro     20    55  131  
## 10 2020 outubro    184    66   49
```

```
## 11 2020 novembro 145 135 72
## 12 2020 dezembro NA 151 177
```

```
dados %>%
  select(mes, temperatura) %>%
  group_by(mes) %>%
  summarise(temp_media = mean(temperatura, na.rm = TRUE)) %>%
  arrange(factor(mes, levels = c("janeiro", "fevereiro", "março", "abril", "maio", "junho", "julho", "agosto", "setembro", "outubro", "novembro", "dezembro")))
```

```
## # A tibble: 12 x 2
##   mes      temp_media
##   <chr>      <dbl>
## 1 janeiro      19.5
## 2 fevereiro    24.1
## 3 março        21.9
## 4 abril        19.2
## 5 maio         22.1
## 6 junho        22.4
## 7 julho        25.0
## 8 agosto       20.1
## 9 setembro    25.0
## 10 outubro    21.2
## 11 novembro    23.2
## 12 dezembro    22.8
```

```
dados %>%
  select(mes, temperatura) %>%
  group_by(mes) %>%
  summarise(temp_media = mean(temperatura, na.rm = TRUE),
            temp_sd = sd(temperatura, na.rm = TRUE)) %>%
  arrange(factor(mes, levels = c("janeiro", "fevereiro", "março", "abril", "maio", "junho", "julho", "agosto", "setembro", "outubro", "novembro", "dezembro")))
```

```
## # A tibble: 12 x 3
##   mes      temp_media temp_sd
##   <chr>      <dbl>   <dbl>
## 1 janeiro      19.5    0.249
## 2 fevereiro    24.1    1.11
## 3 março        21.9    3.87
## 4 abril        19.2    0
## 5 maio         22.1    4.16
## 6 junho        22.4    4.19
## 7 julho        25.0    1.16
## 8 agosto       20.1    2.01
## 9 setembro    25.0    1.30
```

```
## 10 outubro      21.2  0.954
## 11 novembro     23.2  0.738
## 12 dezembro     22.8  4.00
```

```
dados %>%
  select(mes, temperatura, salinidade) %>%
  group_by(mes) %>%
  summarise(temp_media = mean(temperatura, na.rm = TRUE),
            temp_sd = sd(temperatura, na.rm = TRUE),
            sal_media = mean(salinidade, na.rm = TRUE),
            sal_sd = sd(salinidade, na.rm = TRUE)) %>%
  arrange(factor(mes, levels = c("janeiro", "fevereiro", "março", "abril", "maio", "junho", "julho", "agosto", "setembro", "outubro", "novembro", "dezembro")))
  mutate(temp_sal = temp_media/sal_media)
```

```
## # A tibble: 12 x 6
##   mes      temp_media temp_sd sal_media  sal_sd temp_sal
##   <chr>      <dbl>   <dbl>    <dbl>   <dbl>   <dbl>
## 1 janeiro      19.5  0.249     24.5  5.30     0.797
## 2 fevereiro    24.1  1.11     26.6  0.473    0.908
## 3 março        21.9  3.87     29.9  2.47     0.732
## 4 abril        19.2  0        17.3  0.00594   1.11
## 5 maio         22.1  4.16     25.7  7.95     0.861
## 6 junho        22.4  4.19     20.8  3.91     1.08
## 7 julho        25.0  1.16     28.8  2.19     0.867
## 8 agosto       20.1  2.01     24.0  1.82     0.840
## 9 setembro     25.0  1.30     30.8  0.611    0.810
## 10 outubro     21.2  0.954     22.2  6.51     0.956
## 11 novembro     23.2  0.738     26.1  9.69     0.890
## 12 dezembro     22.8  4.00     18.1  0.00526   1.26
```


Chapter 6

Exercício

Importar o arquivo excel de nome “tarefa” (<https://github.com/wilson0106/tidyverse/blob/master/tarefa.xlsx>).

1 - A partir da seguinte planilha:

especie	experimento	replica	mortalidade	avaliador
sp1	baixa	1	20	alguém
sp1	baixa	2	15	alguém
sp1	baixa	3	17	alguém
sp1	baixa	3	17	alguém
sp1	media	1	55	alguém
sp1	media	2	NA	ninguém
sp1	media	3	80	ninguém
sp1	media	3	80	ninguém
sp1	alta	1	78	alguém
sp1	alta	2	78	alguém
sp1	alta	3	71	alguém
sp2	baixa	1	34	alguém
sp2	baixa	2	30	ninguém
sp2	baixa	2	30	ninguém
sp2	baixa	3	31	ninguém
sp2	media	1	NA	ninguém
sp2	media	2	79	ninguém
sp2	media	3	68	ninguém
sp2	alta	1	100	ninguém
sp2	alta	2	100	alguém
sp2	alta	3	95	alguém
sp3	baixa	1	5	alguém
sp3	baixa	2	9	alguém
sp3	baixa	3	4	alguém
sp3	media	1	32	alguém
sp3	media	2	28	alguém
sp3	media	3	24	alguém
sp3	alta	1	45	alguém
sp3	alta	2	50	alguém
sp3	alta	3	52	alguém
Pre-teste	baixa	1	10	alguém
Pre-teste	baixa	2	15	alguém
Pre-teste	baixa	3	13	alguém
Pre-teste	media	2	56	alguém
Pre-teste	media	3	51	alguém
Pre-teste	alta	1	84	ninguém
Pre-teste	alta	2	84	ninguém

Produza as tabelas a seguir. Tente executar com apenas um comando e o mínimo de funções possíveis. **Utilize o operador pipe: %>%**. Se houver observações iguais remova-as

experimento	replica	sp1	sp2	sp3
baixa	1	20	34	5
baixa	2	15	30	9
baixa	3	17	31	4
media	1	55	NA	32
media	3	80	68	24
media	2	NA	79	28
alta	1	78	100	45
alta	2	78	100	50
alta	3	71	95	52

especie	experimento	media	desvio	n
sp1	baixa	17.33333	2.516611	3
sp2	baixa	31.66667	2.081666	3
sp3	baixa	6.00000	2.645751	3
sp1	media	67.50000	17.677670	2
sp2	media	73.50000	7.778175	2
sp3	media	28.00000	4.000000	3
sp1	alta	75.66667	4.041452	3
sp2	alta	98.33333	2.886751	3
sp3	alta	49.00000	3.605551	3

2 - A partir das seguintes tabelas:

Salinidade

replica	1h	2h	3h	4h	5h
Rep_1	28	26	32	26	29
Rep_2	25	27	32	26	25
Rep_3	29	25	31	25	NA
Rep_4	33	27	28	31	28
Rep_5	26	29	29	33	30

Temperatura

replica	1h	2h	3h	4h	5h
Rep_1	19	23	21	22	19
Rep_2	22	22	17	26	23
Rep_3	17	NA	22	26	26
Rep_4	23	24	20	24	20
Rep_5	20	17	19	23	19

pH

replica	1h	2h	3h	4h	5h
Rep_1	7.63	6.51	7.52	6.21	6.90
Rep_2	6.36	7.42	7.73	7.27	6.55
Rep_3	6.02	6.56	7.35	6.71	7.62
Rep_4	7.30	7.57	NA	6.79	7.66
Rep_5	6.38	6.24	7.84	7.87	7.21

Mortalidade

replica	1h	2h	3h	4h	5h
Rep_1	27	51	67	59	16
Rep_2	40	52	30	NA	43
Rep_3	27	77	35	37	42
Rep_4	30	84	30	34	45
Rep_5	NA	73	69	17	26

Produza a seguinte planilha. Tente executar com mínimo de funções possíveis.

Dica: converta cada tabela em uma planilha e as una como colunas. “Utilize o operador pipe: %>%”. Se houver observações iguais remova-as

replica	hora	salinidade	temperatura	pH	mortalidade
Rep_1	1h	28	19	7.63	27
Rep_1	2h	26	23	6.51	51
Rep_1	3h	32	21	7.52	67
Rep_1	4h	26	22	6.21	59
Rep_1	5h	29	19	6.90	16
Rep_2	1h	25	22	6.36	40
Rep_2	2h	27	22	7.42	52
Rep_2	3h	32	17	7.73	30
Rep_2	5h	25	23	6.55	43
Rep_3	1h	29	17	6.02	27
Rep_3	3h	31	22	7.35	35
Rep_3	4h	25	26	6.71	37
Rep_4	1h	33	23	7.30	30
Rep_4	2h	27	24	7.57	84
Rep_4	4h	31	24	6.79	34
Rep_4	5h	28	20	7.66	45
Rep_5	2h	29	17	6.24	73
Rep_5	3h	29	19	7.84	69
Rep_5	4h	33	23	7.87	17
Rep_5	5h	30	19	7.21	26

Chapter 7

Resposta exercício

Será postada assim que todos que participaram da aula enviarem as suas respostas. ;)