

Importação de outros formatos retangulares

Benilton Carvalho, Guilherme Ludwig

Revisão

```
publishers
```

```
## # A tibble: 3 x 2
##   publisher yr_founded
##   <chr>      <dbl>
## 1 DC          1934
## 2 Marvel      1939
## 3 Image      1992
```

```
superheroes
```

```
## # A tibble: 7 x 4
##   name      alignment gender publisher
##   <chr>    <chr>    <chr> <chr>
## 1 Magneto  bad      male   Marvel
## 2 Storm    good     female Marvel
## 3 Mystique bad      female Marvel
## 4 Batman   good     male   DC
## 5 Joker    bad      male   DC
## 6 Catwoman bad      female DC
## 7 Hellboy  good     male   Dark Horse Comics
```

Revisão

1. Qual a sintaxe do comando que combinará apenas os registros existentes em ambas as tabelas, `superheroes` e `publishers`? Qual é o resultado esperado?
2. Qual a sintaxe do comando que combinará os registros de `superheroes` e `publishers`, de forma que todos os heróis sejam listados na tabela resultante? Qual é o resultado esperado?
3. Qual a sintaxe do comando que apresentará todos os registros de `superheroes` cuja chave não exista em `publishers`?

Combinar apenas os registros existentes em ambas as tabelas, superheroes e publishers?

```
superheroes %>% inner_join(publishers, by='publisher')
```

```
## # A tibble: 6 x 5
##   name      alignment gender publisher yr_founded
##   <chr>    <chr>    <chr> <chr>      <dbl>
## 1 Magneto  bad      male   Marvel     1939
## 2 Storm    good     female Marvel     1939
## 3 Mystique bad      female Marvel     1939
## 4 Batman   good     male   DC         1934
## 5 Joker    bad      male   DC         1934
## 6 Catwoman bad      female DC         1934
```

Combinar os registros de superheroes e publishers, de forma que todos os heróis sejam listados na tabela resultante?

```
superheroes %>% left_join(publishers, by='publisher')
```

```
## # A tibble: 7 x 5
##   name      alignment gender publisher      yr_founded
##   <chr>    <chr>      <chr> <chr>         <dbl>
## 1 Magneto  bad        male   Marvel        1939
## 2 Storm   good       female Marvel        1939
## 3 Mystique bad        female Marvel        1939
## 4 Batman  good       male   DC            1934
## 5 Joker   bad        male   DC            1934
## 6 Catwoman bad       female DC            1934
## 7 Hellboy good       male   Dark Horse Comics NA
```

Apresentar todos os registros de superheroes cuja chave não exista em publishers?

```
superheroes %>% anti_join(publishers, by='publisher')
```

```
## # A tibble: 1 x 4
##   name      alignment gender publisher
##   <chr>    <chr>      <chr> <chr>
## 1 Hellboy good        male   Dark Horse Comics
```

Pacote haven

- Permite a leitura de diversos tipos utilizados por outros pacotes estatísticos;
- Wrapper do software ReadStat (biblioteca em C, linha de comando);
- Parte do tidyverse;
- Resultados são tibble.



Arquivos SAS

Tipo de Arquivo	Formato	Operação	Comando
.sas7bdat	SAS	Leitura	read_sas()
.sas7bcat	SAS	Leitura	read_sas()
.xpt	SAS	Leitura	read_xpt()
.sas7bdat	SAS	Escrita	write_sas()

Exemplo Escrita/Leitura SAS

```
library(haven)
path = system.file("examples", package="haven")
dir(path)
```

```
## [1] "iris.dta"      "iris.sas7bdat" "iris.sav"
```

```
read_sas(file.path(path, "iris.sas7bdat"))[1:3,]
```

```
## # A tibble: 3 x 5
##   Sepal_Length Sepal_Width Petal_Length Petal_Width Species
##   <dbl>        <dbl>        <dbl>        <dbl> <chr>
## 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
```

```
write_sas(mtcars, "mtcars.sas7bdat")
```

Arquivos Stata

Tipo de Arquivo	Formato	Operação	Comando
.dta	Stata	Leitura	<code>read_dta()</code>
.dta	Stata	Escrita	<code>write_dta()</code>

Exemplo Escrita/Leitura Stata

```
library(haven)
path = system.file("examples", package="haven")
dir(path)
```

```
## [1] "iris.dta"      "iris.sas7bdat" "iris.sav"
```

```
read_dta(file.path(path, "iris.dta"))[1:3,]
```

```
## # A tibble: 3 x 5
##   sepallength sepalwidth petallength petalwidth species
##   <dbl>       <dbl>       <dbl>       <dbl> <chr>
## 1     5.10         3.5         1.40         0.200 setosa
## 2     4.90         3           1.40         0.200 setosa
## 3     4.70         3.20        1.30         0.200 setosa
```

```
write_dta(mtcars, "mtcars.dta")
```

Arquivos SPSS

Tipo de Arquivo	Formato	Operação	Comando
.sav	Stata	Leitura	<code>read_sav()</code>
.sav	Stata	Escrita	<code>write_sav()</code>

Exemplo Escrita/Leitura SPSS

```
library(haven)
path = system.file("examples", package="haven")
dir(path)
```

```
## [1] "iris.dta"      "iris.sas7bdat" "iris.sav"
```

```
read_sav(file.path(path, "iris.sav"))[1:3,]
```

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

```
write_sav(mtcars, "mtcars.sav")
```

Arquivos Excel e Pacote `readxl`

- Em um arquivo Excel:
 - Células que você vê podem não existir;
 - Células que você não vê podem existir;
- Pacote `readxl`:
 - Funciona em qualquer ambiente (Windows, Linux, Mac) sem dependências externas;
 - Permite a leitura de arquivos Excel (`.xls` ou `.xlsx`);
 - Lê apenas células com conteúdo;
 - Células vazias em colunas existentes são preenchidas com NA;
 - Resultados são `tibble`.

readxl busca identificar geometria

	A	B	C	D	E
1					
2					
3		B3	C3	D3	
4		B4	C4	D4	
5		B5	C5	D5	
6		B6	C6	D6	
7					
8					

```
library(readxl)
read_excel(readxl_example("geometry.xlsx"))
```

```
## # A tibble: 3 x 3
##   B3     C3     D3
##   <chr> <chr> <chr>
## 1 B4     C4     D4
## 2 B5     C5     D5
## 3 B6     C6     D6
```

readr pode ler faixas de dados

	A	B	C	D	E
1					
2					
3		B3	C3	D3	
4		B4	C4	D4	
5		B5	C5	D5	
6		B6	C6	D6	
7					
8					

readr e faixas de dados

```
library(readxl)
read_excel(readxl_example("geometry.xlsx"), range = "A2:C4")
```

```
## New names:
## * ` ` -> ...1
## * ` ` -> ...2
## * ` ` -> ...3

## # A tibble: 2 x 3
##   ...1   ...2   ...3
##   <lgl> <chr> <chr>
## 1 NA     B3     C3
## 2 NA     B4     C4
```

Leitura por partes com `readxl`

- `cell_rows()`
 - `cell_rows(1:10)`
- `cell_cols()`
 - `cell_cols(4:8)`
- `anchored()`
 - `anchored("C5", c(3, 4))`
- `cell_limits()`
 - `cell_limits(c(5, 3), c(NA, NA))`

Pacote r i o

- Busca simplificar importação e exportação de arquivos:
 - Importar arquivos: `import()`
 - Exportar para arquivos: `export()`
 - Utiliza extensão do arquivo para identificar padrão;
- Em arquivos sem extensão, o argumento `format=` deve ser utilizado:
 - `import("arquivo_sem_extensao", format='csv')`

Importando Arquivo SAS

```
library(magrittr)
library(tibble)
library(rio)
x1 = import("mtcars.sas7bdat")
x1 %>% as_tibble()
```

```
## # 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
```

Importando Arquivo Stata

```
library(magrittr)
library(tibble)
library(rio)
x2 = import("mtcars.dta")
x2 %>% as_tibble()
```

```
## # 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
```

Importando Arquivo SPSS

```
library(magrittr)
library(tibble)
library(rio)
x3 = import("mtcars.sav")
x3 %>% as_tibble()
```

```
## # 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
```

Exportar para Formato

```
path = 'exAula04'  
dir.create(path)  
export(x1, file=file.path(path, 'x1.sas7bdat'))  
export(x2, file=file.path(path, 'x2.dta'))  
export(x3, file=file.path(path, 'x3.sav'))  
export(list(mtcars1 = mtcars[1:10,],  
           mtcars2 = mtcars[11:20,],  
           mtcars2 = mtcars[21:32,]),  
       file.path(path, "mtcars.xlsx"))
```

Note: zip::zip() is deprecated, please use zip::zipr() instead

```
dir(path)
```

```
## [1] "mtcars.xlsx" "x1.feather" "x1.sas7bdat" "x2.dta" "x3.sav"
```

Conversão de Formatos via rio

Formatos de arquivos reconhecidos pelo pacote `rio` podem ser convertidos para outros formatos utilizando o comando `convert`.

```
list.files(path)
```

```
## [1] "mtcars.xlsx" "x1.feather" "x1.sas7bdat" "x2.dta" "x3.sav"
```

```
f1 = file.path(path, 'x1.sas7bdat')  
f2 = file.path(path, 'x1.feather')  
convert(f1, f2)  
list.files(path)
```

```
## [1] "mtcars.xlsx" "x1.feather" "x1.sas7bdat" "x2.dta" "x3.sav"
```


Formatos Suportados via rio

- Para instalação completa, utilizar:

```
install.packages('rio')  
install_formats()
```

- Dentre os formatos reconhecidos:
 - CSV, TSV, JSON e todos os "retangulares" texto;
 - SAS, SPSS, Stata, Minitab, Systat;
 - Excel;
 - Epiinfo;
 - XBASE;
 - Weka;
 - Matlab;
 - Feather.