Introdução ao Tratamento e Análise de Dados em R

Aula 4 - Limpando e organizando seus dados

Sérgio Rivero

PPGE-UFPA

17 de maio de 2019





Sumário

Objetivos da Aula

- O que é uma boa base de dados e que tipos de bases existem?
- 3 Exercícios





Objetivos da Aula

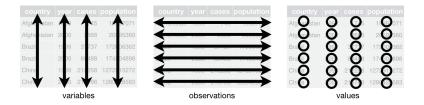
Discutir algumas funções de correção e limpeza de bases de dados em R.





O que é uma boa base de dados e que tipos de bases existem?

Boas bases de dados para trabalhar, sobretudo com tidy, são bases que possuem variáveis nas colunas e observações nas linhas. Podemos identificar tal estrutura na figura 1 abaixo:



Organização ideal para base de dados





Tipos de Bases de Dados

Há três tipos bases de dados principais que comportam a estrutura de variáveis nas colunas e observações nas linhas que são:

- Séries Temporais
- Cortes Transversais
- Dados em Painel





Série Temporal

Séries temporais(ou longitudinais) são identificadas como observações ordenadas ao longo do "tempo". Exemplos:

- PIB distribuído anualmente;
- Arrecadação mensal de um ou mais Estados;
- Taxa de câmbio diária.





A base do BCB

Um bom exemplo de fonte de bases de dados de series temporais é o SGS - Banco Central. Segue o link abaixo:

	Parâmetros Informados
Séries selecionadas	
20783 - Spread médio das operaç	ões de crédito - Total
Período	Função
01/03/2018 a 27/03/2019	Linear
Registros encontrados por série:	12
	Lista de valores (Formato numérico: Europeu - 123.456.789,00)
Data	
mês/AAAA	
mar/2018	19,93
abr/2018	19,56
mai/2018	18,46
jun/2018	17,68
jul/2018	17,69
ago/2018	17,58
set/2018	17,26
out/2018	17,88
nov/2018	18,08
dez/2018	16,94
jan/2019	18,58
fev/2019	19,04
Fonte	BCB-DSTAT

Exemplo de série temporal

https://www3.bcb.gov.br/sgspub/localizarseries/
localizarSeries.do?method=prepararTelaLocalizarSeries
FACECON

Corte Transversal

Dados transversais(ou cross-section) são identificados quando não há importância de ordenação das observações, além de representar dados em um único ponto do "tempo". Exemplos:

- Arrecadação de ICSM para todos Estados do Brasil para o ano de 2018;
- Dados demográficos para todos os municípios do Pará para o ano de 2010.

Um bom exemplo de fonte de bases de dados de cortes transversais é o Sidra - IBGE. Segue o link abaixo:

https://sidra.ibge.gov.br/home





A Série

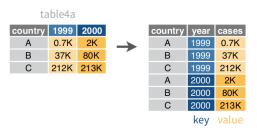


Exemplo de corte transversal





gather



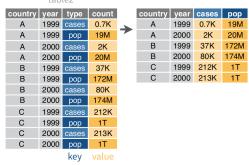
Exemplo da função gather: gather(table4a, '1999', '2000',key = "year", value = "cases")





spread





Exemplo da função spread: spread(table2, type, count)





unite

table5

country	century	year		country	year
Afghan	19	99		Afghan	1999
Afghan	20	0	_	Afghan	2000
Brazil	19	99		Brazil	1999
Brazil	20	0		Brazil	2000
China	19	99		China	1999
China	20	0		China	2000

Exemplo da função unite: unite(table5, century, year, col = "year", sep = "")





separate

table

country	year	rate		country	year	cases	рор
Α	1999	0.7K / 19M		Α	1999	0.7K	19M
Α	2000	2K/20M	-	Α	2000	2K	20M
В	1999	37K / 172M		В	1999	37K	172
В	2000	80K / 174M		В	2000	80K	174
С	1999	212K/1T		С	1999	212K	1T
С	2000	213K / 1T		С	2000	213K	1T

Exemplo da função separate: separate(table3, rate, into = c("cases", "pop"))





separate_rows



Exemplo da função separate_rows: separate_rows(table3, rate)

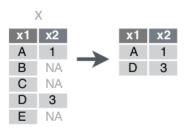
С

2000





Valores Omissos (NA)



$$drop_na(x, x2)$$

Exemplo da função drop_na: drop_na(x, x2)





Selecionando observações com dplyr

Subset Observations (Rows)



dplyr::filter(iris, Sepal.Length > 7)

Extract rows that meet logical criteria.

dplyr::distinct(iris)

Remove duplicate rows.

dplyr::sample_frac(iris, 0.5, replace = TRUE)

Randomly select fraction of rows.

dplyr::sample n(iris, 10, replace = TRUE)

Randomly select n rows.

dplyr::slice(iris, 10:15)

Select rows by position.

dplyr::top_n(storms, 2, date)

Select and order top n entries (by group if grouped data).

	Logic in R - ?(omparison, ?base	::Logic
<	Less than	!=	Not equal to
>	Greater than	%in%	Group membership
==	Equal to	is.na	Is NA
<=	Less than or equal to	!is.na	Is not NA
>=	Greater than or equal to	&, ,!,xor,any,all	Boolean operators

Subset Variables (Columns)



dplyr::select(iris, Sepal, Width, Petal, Length, Species)

Select columns by name or helper function.

Helper functions for select - ?select

select(iris, contains("."))

Select columns whose name contains a character string.

select(iris, ends_with("Length"))

Select columns whose name ends with a character string.

select(iris, everything())
Select every column.

select(iris, matches(".t."))

Select columns whose name matches a regular expression.

select(iris, num_range("x", 1:5))

Select columns named x1, x2, x3, x4, x5.

select(iris, one_of(c("Species", "Genus")))

Select columns whose names are in a group of names.

select(iris, starts_with("Sepal"))

Select columns whose name starts with a character string.

select(iris, Sepal.Length:Petal.Width)

Select all columns between Sepal.Length and Petal.Width (inclusive).

select(iris, -Species)

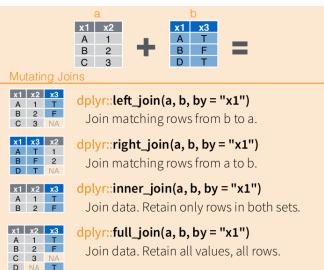
Select all columns except Species.



Algumas funçoes para alteração de dataframes



Combinando Datasets







Pipes

Além de todas essas funções apresentadas anteriormente, o dplyr permite concatenar funções por meio de um mecanismo chamado **pipe**. Para utilizar o pipe é preciso colocar tais símbolos:

• %>%

Abaixo temos um exemplo da utilização do pipe. O exemplo consiste em calcular média e desvio padrão a partir de um agrupamento por espécie da base de dados starwars.

```
starwars81 <- starwars %>%
group_by(species) %>%
summarise(desvpad = sd(birth_year, na.rm = TRUE),
avg = mean(birth_year, na.rm = TRUE)
)
```





Exercícios

Exercícios



