Manipulando Dados
com dplyr e tidyr
Folha de Referência

RStudio

Dados Organizados - O fundamento para a manipulação em R

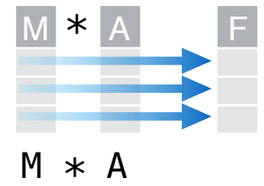
Em uma base de dados organizada:

Cada **variável** tem sua própria coluna

&

Cada **observação** tem sua própria **linha**

Dados arrumados complementam as **operações vetorizadas** de R. Ele automaticamente preservará as observações enquanto você manipula as variáveis. Nenhum outro formato é tão intuitivo quanto esse com R.



Sintaxe - Convenções úteis

dplyr::**tbl_df(iris)**

Converte os dados para a classe tbl. tbl's são mais fáceis de examinar do que data frames. R mostra apenas os dados que cabem na tela:

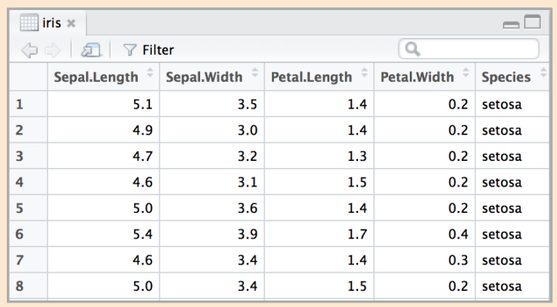
```
Source: local data frame [150 x 5]

   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::**glimpse(iris)**

Sumário denso dos dados em tbl.

utils::**View(iris)**

Visualiza os dados em um visor no formato de planilha (note o V maiúsculo).



iris

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	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	5.0	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	5.0	3.4	1.5	0.2	setosa

dplyr::**%>%**

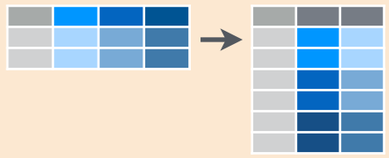
Passa o objeto do lado esquerdo como o primeiro argumento (ou o argumento **.**) da função do lado direito.

`x %>% f(y)` *é o mesmo que* **`f(x, y)`**
`y %>% f(x, ., z)` *é o mesmo que* **`f(x, y, z)`**

"Entubando" (ou *"Piping"*) com %>% deixa o código mais legível, e.g.

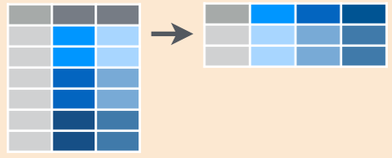
```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Remodelando Dados - Mude o formato dos dados



tidyr::**gather(cases, "year", "n", 2:4)**

Reúne colunas em linhas.



tidyr::**spread(pollution, size, amount)**

Espalha linhas em colunas.

tidyr::**separate(storms, date, c("y", "m", "d"))**

Separa uma coluna em várias.

tidyr::**unite(data, col, ..., sep)**

Une várias colunas em uma.

dplyr::**data_frame(a = 1:3, b = 4:6)**

Combina vetores em um data frame (otimizado).

dplyr::**arrange(mtcars, mpg)**

Ordena linhas pelos valores de uma coluna (menor para o maior).

dplyr::arrange(mtcars, **desc(mpg)**)

Ordena linhas pelos valores de uma coluna (maior para menor).

dplyr::**rename(tb, y = year)**

Renomeia colunas de um data frame.

Extração de Observações (Linhas)

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

Extrai as linhas que satisfazem o critério lógico.

dplyr::**distinct(iris)**

Remove linhas duplicadas.

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

Seleciona frações de linhas aleatoriamente.

dplyr::**sample_n(iris, 10, replace = TRUE)**

Seleciona n linhas aleatoriamente.

dplyr::**slice(iris, 10:15)**

Seleciona linhas pela posição.

dplyr::**top_n(storms, 2, date)**

Seleciona e ordena as top n entradas (por grupo se os dados estiverem agrupados).

Lógica em R - ?Comparison, ?base::Logic

<	Menor que	!=	Diferente de
>	Maior que	%in%	Pertence a
==	Igual a	is.na	É NA
<=	Menor que ou igual a	!is.na	Não é NA
>=	Maior que ou igual a	&,|,!,xor,any,al	Operadores

devtools::install_github("rstudio/EDAWR") para bases de dados

Extração de Variáveis (Colunas)

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

Seleciona colunas por nome ou funções auxiliares.

Funções auxiliares para a seleção - ?select

select(iris, contains("."))
Seleciona colunas cujo nome contém caracteres string.

select(iris, ends_with("Length"))
Seleciona colunas cujo nomes terminam com caracteres string.

select(iris, everything())
Seleciona todas as colunas.

select(iris, matches(".t."))
Seleciona colunas cujos nomes se adequam a uma expressão regular.

select(iris, num_range("x", 1:5))
Seleciona colunas nomeadas x1, x2, x3, x4, x5.

select(iris, one_of(c("Species", "Genus")))
Seleciona colunas cujos nomes estão em um grupo de nomes.

select(iris, starts_with("Sepal"))
Seleciona colunas cujos nomes começam com caracteres string.

select(iris, Sepal.Length:Petal.Width)
Seleciona todas as colunas entre Sepal.Length e Petal.Width (inclusive).

select(iris, -Species)
Seleciona todas as colunas exceto Species.

Traduzido por Augusto Queiroz de Macedo • br.linkedin.com/in/augusto-queiroz-de-macedo-552b9822

Resumir Dados

dplyr::**summarise(iris, avg = mean(Sepal.Length))**
Resume os dados em uma única linha de valores.

dplyr::**summarise_each(iris, funs(mean))**
Aplica uma função de resumo em cada coluna.

dplyr::**count(iris, Species, wt = Sepal.Length)**
Conta o número de linhas com cada valor único da variável Species (com ou sem o peso wt).

Summarise usa **funções de resumo**, as quais recebem um vetor de valores e retornam um único valor, como:

dplyr::**first**
Primeiro valor de um vetor.

dplyr::**last**
Último valor de um vetor.

dplyr::**nth**
N-ésimo valor de um vetor.

dplyr::**n**
de valores de um vetor.

dplyr::**n_distinct**
de valores distintos de um vetor.

IQR
IQR de um vetor.

min
Mínimo de um vetor.

max
Máximo de um vetor.

mean
Média de um vetor.

median
Mediana de um vetor.

var
Variância de um vetor.

sd
Desvio padrão de um vetor.

Agrupar Dados

dplyr::**group_by(iris, Species)**
Agrupa dados em linhas com iguais valores de Species.

dplyr::**ungroup(iris)**
Remove a informação do grupo do data frame.

iris %>% group_by(Species) %>% summarise(…)
Calcula resumos separados para cada grupo.

Criar Novas Variáveis

dplyr::**mutate(iris, sepal = Sepal.Length + Sepal. Width)**
Calcula e acrescenta uma ou mais novas colunas.

dplyr::**mutate_each(iris, funs(min_rank))**
Aplica uma função de janelamento para cada coluna.

dplyr::**transmute(iris, sepal = Sepal.Length + Sepal. Width)**
Calcula um ou mais novas colunas. Remove as originais.

Mutate usa **funções de janelamento**, as quais recebem um vetor de valores e retornam outro vetor de valores, como:

dplyr::**lead**
Copia com valores adiantados por 1.

dplyr::**lag**
Copia com valores atrasados por 1.

dplyr::**dense_rank**
Ranking sem brechas.

dplyr::**min_rank**
Ranking. Empates recebem o rank mínimo.

dplyr::**percent_rank**
Ranking redimensionado para [0, 1].

dplyr::**row_number**
Ranking. Empates recebem o primeiro valor.

dplyr::**ntile**
Separa vetor em n partes.

dplyr::**between**
Os valores estão entre a e b?

dplyr::**cume_dist**
Distribuição cumulativa.

dplyr::**cumall**
`all` cumulativo

dplyr::**cumany**
`any` cumulativo

dplyr::**cummean**
`mean` cumulativo

cumsum
`sum` cumulativo

cummax
`max` cumulativo

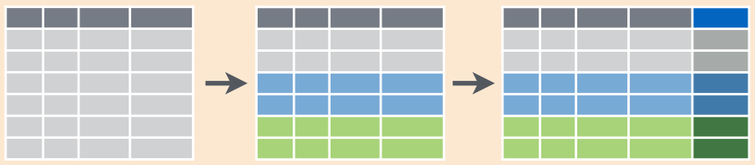
cummin
`min` cumulativo

cumprod
`prod` cumulativo

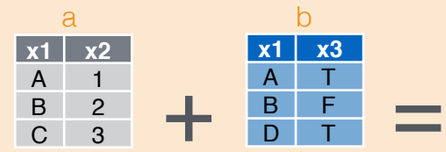
pmax
max por elementos

pmin
`min` por elementos

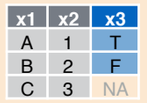
iris %>% group_by(Species) %>% mutate(…)
Calcula novas variáveis por grupo.



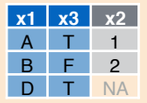
Combinar Conjuntos de Dados



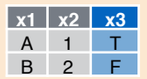
Uniões Mutantes



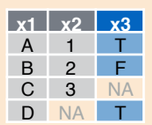
dplyr::**left_join(a, b, by = "x1")**
Junta linhas coincidentes de *b* para *a*.



dplyr::**right_join(a, b, by = "x1")**
Junta linhas coincidentes de *a* para *b*.

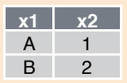


dplyr::**inner_join(a, b, by = "x1")**
Junção de dados. Mantém apenas as linhas em ambos os conjuntos.

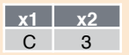


dplyr::**full_join(a, b, by = "x1")**
Junção de dados. Mantém todos os valores, todas as linhas.

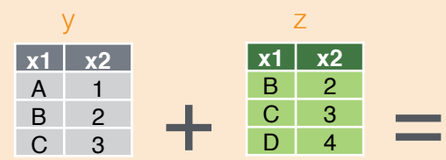
Uniões como Filtros



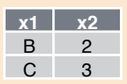
dplyr::**semi_join(a, b, by = "x1")**
Todas as linhas em *a* presentes em *b*.



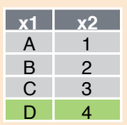
dplyr::**anti_join(a, b, by = "x1")**
Todas as linhas em *a ausentes* em *b*.



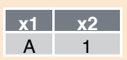
Operações em Conjuntos



dplyr::**intersect(y, z)**
Linhas que aparecem em ambos *y* e *z*.

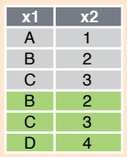


dplyr::**union(y, z)**
Linhas que aparecem em um ou em ambos *y* e *z*.

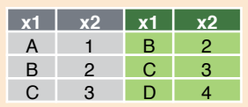


dplyr::**setdiff(y, z)**
Linhas que aparecem em *y* mas não em *z*.

Juntar



dplyr::**bind_rows(y, z)**
Junta *z* em *y* como novas linhas.



dplyr::**bind_cols(y, z)**
Junta *z* em *y* como novas colunas.
Cuidado: coincide linhas pela posição.