Biến đổi dữ liệu

với dplyr & tidyr

Cheat Sheet

RStudio

Tidy Data (Dữ liệu sạch) – Nền tảng của việc biến đổi dữ liệu

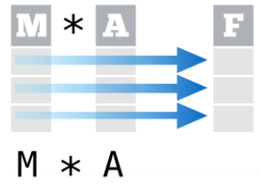
Dữ liệu sạch

Mỗi **cột** chứa một **biến**

&

Mỗi **hàng** chứa một **quan sát** (observation)

Trong dữ liệu sạch, R sẽ giữ nguyên định dạng các biến theo dạng vector



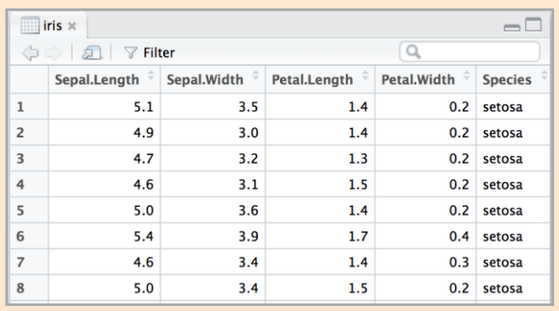
Câu lệnh

dplyr::**tbl_df(iris)**
Biến đổi dữ liệu sang dạng **tbl**, dạng dữ liệu này đơn giản hơn khi kiểm tra dữ liệu. R chỉ hiển thị dữ liệu vừa với màn hình hiển thị

```
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)**
Tóm tắt thông tin của định dạng dữ liệu tbl
utils::**View(iris)**
Hiển thị dữ liệu dạng bảng (tương tự như excel)



	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

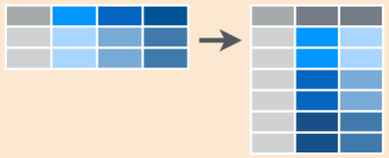
dplyr::**%>%**
Chuyển đối tượng (object) ở phía bên trái dấu %>% thành biến (argument) của hàm bên phải dấu %>%

x %>% f(y) *cho kết quả tương tự như* **f(x, y)**
y %>% f(x, ., z) *cho kết quả tương tư như* **f(x, y, z)**

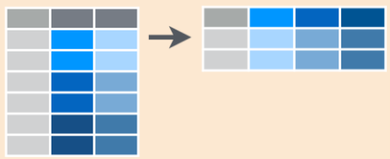
Sử dụng "piping" **%>%** làm cho việc theo dõi code đơn giản hơn rất nhiều, ví dụ:

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

Reshaping Data - Thay đổi cấu trúc dữ liệu

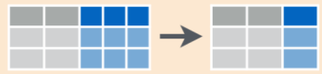


tidyr::**gather(cases, "year", "n", 2:4)**
Biến đổi dữ liệu dạng cột thành hàng



tidyr::**spread(pollution, size, amount)**
Biến đổi dữ liệu dạng hàng thành cột

tidyr::**separate(storms, date, c("y", "m", "d"))**
Tách 1 cột thành nhiều cột



tidyr::**unite(data, col, ..., sep)**
Ghép nhiều cột thành một.

dplyr::**data_frame(a = 1:3, b = 4:6)**
Nhóm các véc-tơ thành data frame
dplyr::**arrange(mtcars, mpg)**
Sắp xếp các dòng theo một biến từ nhỏ đến lớn
dplyr::**arrange(mtcars, desc(mpg))**
Sắp xếp các dòng theo một biến từ lớn đến nhỏ
dplyr::**rename(tb, y = year)**
Thay đổi tên của một biến

Lấy dữ liệu theo dòng

dplyr::**filter(iris, Sepal.Length > 7)**
Lọc các dòng với điều kiện xác định
dplyr::**distinct(iris)**
Xóa các dòng có giá trị trùng nhau
dplyr::**sample_frac(iris, 0.5, replace = TRUE)**
Lấy ngẫu nhiên các quan sát theo tỉ lệ cho trước
dplyr::**sample_n(iris, 10, replace = TRUE)**
Lấy ngẫu nhiêu n dòng
dplyr::**slice(iris, 10:15)**
Lấy các dòng theo vị trí
dplyr::**top_n(storms, 2, date)**
Lấy và sắp xếp thứ tự n dòng đầu tiên

Các phép toán logic trong R

<	Nhỏ hơn	!=	Không bằng
>	Lớn hơn	%in%	Chứa trong nhóm
==	Bằng	is.na	Kiểm tra dữ liệu thiếu (NA)
<=	Nhỏ hơn hoặc bằng	!is.na	Kiểm tra dữ liệu không thiếu
>=	Lớn hơn hoặc bằng	&,|,!,xor,any,all	Toán tử kết hợp logic đơn giản trong R

Lấy dữ liệu: **devtools::install_github("rstudio/EDAWR")**

Lấy dữ liệu theo cột

dplyr::**select(iris, Sepal.Width, Petal.Length, Species)**
Chọn các biến (cột) theo tên

Các hàm với "select"- ?select

select(iris, contains("."))
Chọn các biến mà tên biến đó có chứa ký tự
select(iris, ends_with("Length"))
Chọn các biến mà tên biến đó kết thúc với một chuỗi ký tự
select(iris, everything())
Chọn tất cả các biến
select(iris, matches(".t."))
Chọn biến mà tên biến đó khớp với điền kiện cho trước
select(iris, num_range("x", 1:5))
Chọn các biên tên x1, x2, x3, x4, x5.
select(iris, one_of(c("Species", "Genus")))
Chọn các biến mà tên biến đó nằm trong 1 nhóm các tên biến
select(iris, starts_with("Sepal"))
Chọn các biến có tên bắt đầu với một chuỗi ký tự
select(iris, Sepal.Length:Petal.Width)
Chọn tất cả các biến từ Sepal.Length đến Petal.Width
select(iris, -Species)
Chọn tất cả các biến ngoại trừ Species

Translator: ranalytics.vn

Tóm tắt dữ liệu

dplyr::**summarise(iris, avg = mean(Sepal.Length))**
Tóm tắt dữ liệu thành các dòng
dplyr::**summarise_each(iris, funs(mean))**
Áp dụng hàm với tất cả các cột trong dữ liệu
dplyr::**count(iris, Species, wt = Sepal.Length)**
Đếm số lượng các dòng khi giá trị của biến là duy nhất (có hoặc không có trọng số)

Summarise sử dụng các hàm tính toán một véc-tơ các giá trị và trả về giá trị, ví dụ:

dplyr::**first**
Giá trị đầu tiên của véc-tơ
dplyr::**last**
Giá trị cuối cùng của véc-tơ.
dplyr::**nth**
Giá trị thứ n của véc-tơ.
dplyr::**n**
Số lượng các quan sát trong véc-tơ
dplyr::**n_distinct**
Số lượng các giá trị khác nhau trong véc-tơ
IQR
IQR của một véc-tơ.

min
Giá trị nhỏ nhất của véc-tơ
max
Giá trị lớn nhất của véc-tơ
mean
Giá trị trung bình của véc-tơ
median
Trung vị của véc-tơ
var
Phương sai của véc-tơ
sd
Độ lệch tiêu chuẩn của véc-tơ

Nhóm dữ liệu

dplyr::**group_by(iris, Species)**
Nhóm các biến thành các hàng có cùng giá trị của Species
dplyr::**ungroup(iris)**
Loại bỏ nhóm
iris %>% group_by(Species) %>% summarise(…)
Tính toán và trả các giá trị với mỗi nhóm

Tạo các biến mới

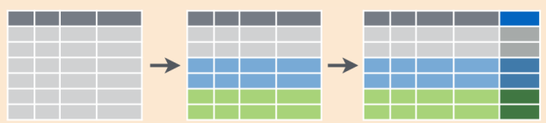
dplyr::**mutate(iris, sepal = Sepal.Length + Sepal. Width)**
Tính toán và tạo một hoặc nhiều biến mới
dplyr::**mutate_each(iris, funs(min_rank))**
Áp dụng hàm cho mỗi biến
dplyr::**transmute(iris, sepal = Sepal.Length + Sepal. Width)**
Tính toán và tạo biến mới, loại bỏ biến cũ.

Sử dụng "window function", các hàm này tính toán 1 véc-tơ & trả ra giá trị là 1 véc-tơ, ví dụ:

dplyr::**lead**
Copy & tạo 1 véc-tơ có giá trị của biến nhanh hơn 1 đơn vị
dplyr::**lag**
Copy & tạo 1 véc-tơ có giá trị của biến chậm hơn 1 đơn vị
dplyr::**dense_rank**
Thứ tự giá trị của biến (không tính trùng giá trị)
dplyr::**min_rank**
Thứ tự giá trị của biến (tính trùng giá trị)
dplyr::**percent_rank**
Thứ tự giá trị trong khoảng [0, 1].
dplyr::**row_number**
Thứ tự giá trị của biến (tính trùng giá trị - các giá trị trùng nhau được xếp thứ tự khác nhau)
dplyr::**ntile(x,n)**
Chia véc-tơ thành n nhóm
dplyr::**between(x,a,b)**
Kiểm tra các giá trị của x trong khoảng [a,b]

dplyr::**cume_dist**
Véc-tơ giá trị hàm phân phối với từng giá trị của biến
dplyr::**cumall**
Véc-tơ giá trị logic tích lũy theo điều kiện cho trước (tất cả các giá trị phải thỏa mãn)
dplyr::**cumany**
Véc-tơ giá trị logic theo điều kiện cho trước (có ít nhất 1 giá trị thỏa mãn)
dplyr::**cummean**
Véc-tơ giá trị trung bình tích lũy của x
Cumsum
Véc-tơ giá trị tổng tích lũy của x
Cummax
Véc-tơ chứa các giá trị lớn nhất của x theo thứ tự
cummin
Véc-tơ chứa các giá trị nhỏ nhất của x theo thứ tự
cumprod
Véc-tơ giá trị của tích
pmax
Véc-tơ giá trị lớn nhất
pmin
Véc-tơ giá trị nhỏ nhất

iris %>% group_by(Species) %>% mutate(…)
Tạo các biến mới với mỗi nhóm



Lấy dữ liệu: devtools::**install_github("rstudio/EDAWR")**

Nối dữ liệu (Combine Data Sets)

a

x1	x2
A	1
B	2
C	3

+

b

x1	x3
A	T
B	F
D	T

=

Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::**left_join(a, b, by = "x1")**
Ghép các dòng có cùng giá trị từ b đến a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::**right_join(a, b, by = "x1")**
Ghép các dòng có cùng giá trị từ a đến b

x1	x2	x3
A	1	T
B	2	F

dplyr::**inner_join(a, b, by = "x1")**
Giữ các giá trị có cả ở a & b

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

dplyr::**full_join(a, b, by = "x1")**
Giữ tất cả các giá trị ở a & b

Filtering Joins

x1	x2
A	1
B	2

dplyr::**semi_join(a, b, by = "x1")**
Giữ lại các dòng xuất hiện trong b

x1	x2
C	3

dplyr::**anti_join(a, b, by = "x1")**
Giu lại các dòng không có trong b

y

x1	x2
A	1
B	2
C	3

+

z

x1	x2
B	2
C	3
D	4

=

Set Operations

x1	x2
B	2
C	3

dplyr::**intersect(y, z)**
Các dòng có ở cả y & z

x1	x2
A	1
B	2
C	3
D	4

dplyr::**union(y, z)**
Dòng có trong y hoặc z.

x1	x2
A	1

dplyr::**setdiff(y, z)**
Dòng có trong y nhưng không có trong z

Binding

x1	x2
A	1
B	2
C	3
B	2
C	3
D	4

dplyr::**bind_rows(y, z)**
Nối dòng dữ liệu từ z sang y,

x1	x2	x1	x2
A	1	B	2
B	2	C	3
C	3	D	4

dplyr::**bind_cols(y, z)**
Nối cột dữ liệu từ z sang y
Lưu ý: trùng giá trị của biến.