用dplyr函数实现数据变换：：速查表

dplyr 函数可应用于“pipes”结构，使用前请保证数据是清洁的，清洁的数据应具有以下特征：

每个**变量**占一**列（column）**

&

每个**案例**，或者说每组**观测值**占一**行（row）**

x %>% f(y)
也可以写作：
f(x, y)

对案例 (cases) 进行描述性总结

这些对数据进行描述性总结的函数作用于列，能够生成对数据进行描述的表格。这些函数输入的是向量 (vector)，返回的是一个值（见第二页）。

描述总结性函数

summarise(.data, …)
生成对数据进行描述性总结的表格。
summarise(mtcars, avg = mean(mpg))

count(x, ..., wt = NULL, sort = FALSE)
根据每个变量的值，统计每组的行数。也可写作 **tally**().
count(iris, Species)

其他形式

summarise_all() - 函数将作用于每一列。
summarise_at() - 函数将作用于特定列。
summarise_if() - 函数将作用于满足条件的所有列。

对案例进行分组

group_by() 函数能够创建一份“被分组”的表格。
dplyr 函数能对每组数据进行分别处理，然后整合结果。

```
mtcars %>%
group_by(cyl) %>%
summarise(avg = mean(mpg))
```

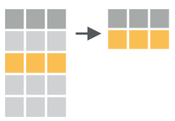
group_by(.data, ..., add = FALSE**)**
这一函数能够返回一份**被分组**的表格 。
g_iris <- group_by(iris, Species)

ungroup(x, …**)**
这一函数能够返回一份**被取消分组**的表格。
ungroup(g_iris)

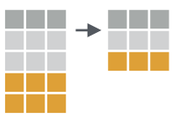
对案例的操作

提取部分案例

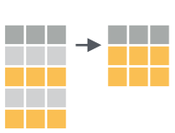
这些函数作用于行，能够返回子集，形成新表格。



filter(.data, …**)** 提取满足逻辑性标准的行。*filter(iris, Sepal.Length > 7)*

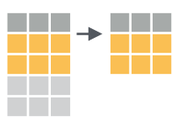


distinct(.data, ..., .keep_all = FALSE**)** 移除重复的行。
distinct(iris, Species)



sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame()**)** 任意选取几行（输入小数确定选取案例的百分比）。
sample_frac(*iris, 0.5, replace = TRUE*)

sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame()**)** 任意选取几行（输入整数确定选取几行案例）。
sample_n(iris, 10, replace = TRUE)



slice(.data, …**)** 任意选取几行（输入行数范围进行选取）。
slice(iris, 10:15)

top_n(x, n, wt**)** 选取前n行，并排序（如果是分组的数据则分组选取排序）。
top_n(iris, 5, Sepal.Width)

配合函数 filter() 使用的逻辑/布尔数学体系的运算符

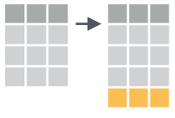
<	<=	is.na()	%in%	|	xor()
>	>=	!is.na()	!	&	

输入 **?base::logic** 和 **?Comparison** 获取帮助。

为案例排序

arrange(.data, …**)** 按照一列或多列数据的值对案例进行排序（由低到高）。如要由高到低排序，使用**desc()**。
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

增加案例



add_row(.data, ..., .before = NULL, .after = NULL**)**
在表格内增添一个或多个案例。
add_row(faithful, eruptions = 1, waiting = 1)

对变量的操作

提取部分变量

这些函数作用于列，能够返回一组列，形成新的向量或表格。

pull(.data, var = -1**)** 提取单列数据形成向量（输入列名或下标索引）。
pull(iris, Sepal.Length)

select(.data, …**)**
提取列形成表格，也写作**select_if()** *select(iris, Sepal.Length, Species)*

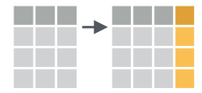
select ()函数的详细用法如下：
e.g. select(iris, starts_with("Sepal"))

contains(match)	num_range(prefix, range)	:, e.g. mpg:cyl
ends_with(match)	one_of(…)	-, e.g, -Species
matches(match)	starts_with(match)	

创建新变量

这些是作用于列的向量化函数。这些函数输入的是向量，返回的是等长的向量（见第二页）。

向量化函数

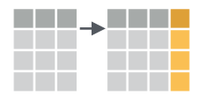


mutate(.data, …**)**
创建新列。
mutate(mtcars, gpm = 1/mpg)

transmute(.data, …**)**
创建新列，删除其他列。
transmute(mtcars, gpm = 1/mpg)

mutate_all(.tbl, .funs, …**)** 将函数用于所有列，和 **funs()** 函数一起使用，也写作 **mutate_if()**
mutate_all(faithful, funs(log(.), log2(.)))
mutate_if(iris, is.numeric, funs(log(.)))

mutate_at(.tbl, .cols, .funs, …**)** 将函数用于特定列，和 **funs(), vars()** 函数及select()函数的辅助函数(helper functions)一起使用。
mutate_at(iris, vars(-Species), funs(log(.)))



add_column(.data, ..., .before = NULL, .after = NULL**)**增加新列，另有 **add_count()**和 **add_tally()**.
add_column(mtcars, new = 1:32)

rename(.data, …**)** 重命名列。
rename(iris, Length = Sepal.Length)

翻译：于艾岑 Aicen Yu / aerzen@outlook.com

向量函数

与MUTATE () 一起使用

mutate() 和 **transmute()** 将向量函数作用于列，以创造新的列。向量函数输入的是向量，返回的是与输出等长的向量。

向量函数

数据“偏移”

dplyr::**lag()** - 将数据向后推一格
dplyr::**lead()** - 将数据向前推一格

累计计算

dplyr::**cumall()** - 累计判断逻辑向量是否**为**“真”
dplyr::**cumany()** - 累计判断逻辑向量是否**有**“真”
cummax() - 累计最大值
dplyr::**cummean()** - 累计平均数
cummin() - 累计最小值
cumprod() - 累计乘积()
cumsum() - 累计和()

排名

dplyr::**cume_dist()** - <=当前排名之前所有值的比例
dplyr::**min_rank()** - 同rank(), 默认ties = “min”
dplyr::**dense_rank()** - 同上，排名间无间隔
dplyr::**ntile()** - 分成n份
dplyr::**percent_rank()** - 同min_rank，用0～1体现百分比
dplyr::**row_number()** - 同rank(), 默认ties = “first”

数学运算

+, - , *, /, ^, %/%, %% - 运算符
log(), log2(), log10() - 对数
<, <=, >, >=, !=, == - 逻辑比较
dplyr::**between()** - 判断x中的值是否在范围内（通过left, right 设定）
dplyr::**near()** - safe == 用于浮点数

其他

dplyr::**case_when()** - 同 if_else()，但能处理多个案例
dplyr::**coalesce()** - 寻找第一个非NA的值
dplyr::**if_else()** - 基于元素的 if() + else()
dplyr::**na_if()** - 用 NA 取代特定值
pmax() - 基于元素的 max()
pmin() - 基于元素的 min()
dplyr::**recode()** - 向量版本的 switch()
dplyr::**recode_factor()** - 向量版本的 switch() ， 用于因子

生成描述性数据的函数

与SUMMARISE ()一起使用

summarise() 对每列数据进行描述性总结，形成一张新表格。总结性函数输入向量，返回单个数值。

描述总结性函数

个数

dplyr::**n()** - 值/行的个数
dplyr::**n_distinct()** - # 不重复的值的个数
sum(!is.na()) - # 非NA数值的个数

位置

mean() - 平均数，或：**mean(!is.na())**
median() - 中位数

逻辑

mean() - 为“真”（TRUE）的比例
sum() - # 为“真”（TRUE）的描述性数据

顺序

dplyr::**first()** - 第一个值
dplyr::**last()** - 最后一个值
dplyr::**nth()** - 返回向量中在特定位置的值

排序

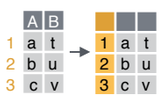
quantile() - 分位数
min() - 最小值
max() - 最大值

散布

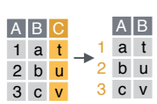
IQR() - 四分位距
mad() - 绝对中位差
sd() - 标准差
var() - 方差

行名

清洁的数据不使用行名，因其将变量置于列外。
如果要编辑行名，需先将其移到一列内。



rownames_to_column()
将行名移到一列内。
a <- rownames_to_column(iris, var = "C")

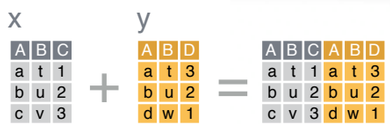


column_to_rownames()
将一列移到行名处。
column_to_rownames(a, var = "C")

此外还有: **has_rownames()**, **remove_rownames()**

合并表格

基于变量的合并



bind_cols() 将两个表格肩并肩放置。

bind_cols(…) 将两个表格肩并肩放置，形成一个表格。
※ 注意行数的一致性。

“变形合并” 基于变量将两个表格合并，并基于值匹配案例。以下是合并两个表格的不同方式。

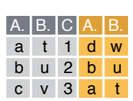
left_join(x, y, by = NULL, copy=FALSE，suffix=c(“.x”,“.y”),…**)**
将y并入x

right_join(x, y, by = NULL, copy = FALSE，suffix=c(“.x”,“.y”),…**)**
将x并入y

inner_join(x, y, by = NULL, copy = FALSE，suffix=c(“.x”,“.y”),…**)**
合并数据，仅保留匹配的行。

full_join(x, y, by = NULL, copy=FALSE，suffix=c(“.x”,“.y”),…**)**
合并数据，保留所有行和所有列。

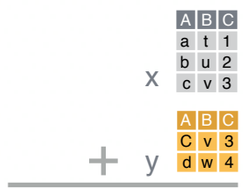
用 by = c("col1", "col2", …) 来规定匹配所用的共同列。
left_join(x, y, by = "A")



用 **by = c("col1" = "col2")**, 来指明共同列在不同表格中的不同名称。
left_join(x, y, by = c("C" = "D"))

用suffix = c("1", “2")为在不同表格中名称相同却不匹配的列设定后缀。
left_join(x, y, by = c("C" = "D"), suffix = c("1", "2"))

基于案例的合并



bind_rows() 将两个表格上下放置。

bind_rows(…, .id = NULL**)**
将两个表格上下放置，形成一个表格。可以用.id= 设置一个指明其原表格名称的列（如图中的DF列）。

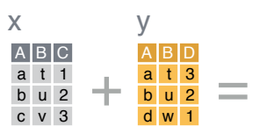
intersect(x, y, …)
在x和y中都出现的行。

setdiff(x, y, …)
在x中出现，在y中未出现的行。

union(x, y, …)
在x或y中出现的行（重复的行删去）。
如要保留重复行，使用union_all()。

setequal() 函数能够检查两组数据是否拥有完全一致的行（顺序不限）。

提取行



“筛选合并" 能用一组数据筛选另一组数据。

semi_join(x, y, by = NULL, …**)**
返回 x 的行中与 y 中的行相同的部分。※ 可用来查看哪些**能**被合并。

anti_join(x, y, by = NULL, …**)**
返回y中没有的、 x中的行。
※ 可用来查看哪些**不能**被合并。

翻译：于艾岑 Aicen Yu / aerzen@outlook.com