

利用dplyr和tidyr进行数据再加工

速查表



由 supstat ANALYTICS 翻译

语法 - 有用的数据再加工规则

dplyr::tbl_df(iris)

将数据转化为tbl类。相比数据框，tbl更易于查看。R只会显示适合屏幕大小的数据：

```
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)

tbl数据的信息密集概括。

utils::View(iris)

在电子表格样式的显示中查看数据集。（标为大写V）。

dplyr::%>%

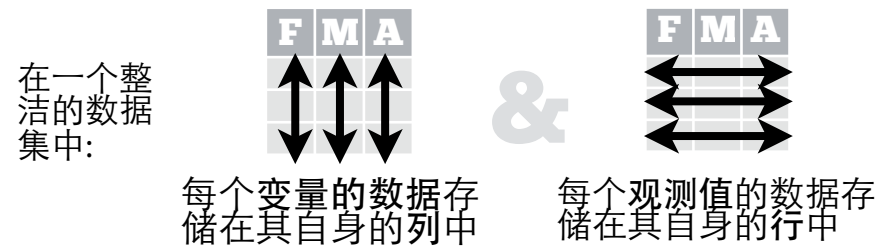
将左边的对象作为第一个参数（或参数.）传递到右边的函数中。

x %>% f(y) 相当于 f(x, y)
y %>% f(x, ., z) 相当于 f(x, y, z)

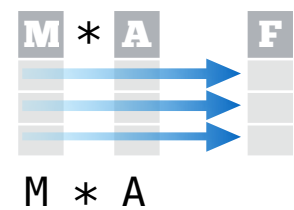
利用%>%进行“Piping”管道操作增强了代码的可读性，例如：

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

整洁数据 - 在R中进行数据再加工的基础



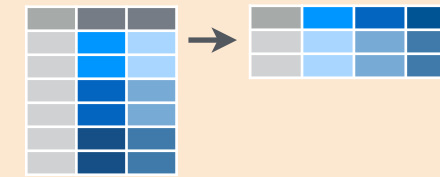
整洁数据与R语言的向量化操作相辅相成。当你在使用变量进行操作时，R会自动保存你的数据记录。任何其他语言都不能像R一样灵活运作。



重组数据 - 改变数据集的布局



tidyr::gather(cases, "year", "n", 2:4)
将列聚集成行。



tidyr::spread(pollution, size, amount)
将行展开为列。



tidyr::separate(storms, date, c("y", "m", "d"))
将单列分离成多列。



tidyr::unite(data, col, ..., sep)
将多列统一为单列。

dplyr::data_frame(a = 1:3, b = 4:6)
将向量合并成数据框（已优化）

dplyr::arrange(mtcars, mpg)
对单列中的行数据值进行排序（从低到高）。

dplyr::arrange(mtcars, desc(mpg))
对单列中的行数据值进行排序（从高到低）。

dplyr::rename(tb, y = year)
重命名数据框中的列变量。

子集观测值 (行)



dplyr::filter(iris, Sepal.Length > 7)
抽取符合逻辑条件的数据记录。

dplyr::distinct(iris)
删除重复记录。

dplyr::sample_frac(iris, 0.5, replace = TRUE)
随机选取部分数据记录。

dplyr::sample_n(iris, 10, replace = TRUE)
随机选取n条数据记录。

dplyr::slice(iris, 10:15)
通过位置选取数据记录。

dplyr::top_n(storms, 2, date)
选取并排列前n条数据记录（若为分组数据则按组排序）

子集变量 (列)



dplyr::select(iris, Sepal.Width, Petal.Length, Species)
通过列名或帮助函数选取列变量。

选取操作的帮助函数 - ?select

- select(iris, contains("."))
选取名称中含有字符的列。
- select(iris, ends_with("Length"))
选取名称以指定字符串结尾的列。
- select(iris, everything())
选取每一列。
- select(iris, matches(".t."))
选取名称符合指定表达式规则的列。
- select(iris, num_range("x", 1:5))
选取名为x1、x2、x3、x4、x5的列。
- select(iris, one_of(c("Species", "Genus")))
选取名称在指定名字组内的列。
- select(iris, starts_with("Sepal"))
选取名称以指定字符串为首的列。
- select(iris, Sepal.Length:Petal.Width)
选取在Sepal.Length和Petal.Width之间的所有列（包含Sepal.Length和Petal.Width）。
- select(iris, -Species)
选取除Species以外的所有列。

R中的逻辑运算 - ?Comparison, ?base::Logic

<	小于	!=	不等于
>	大于	%in%	组成员
==	等于	is.na	为缺失值
<=	小于或等于	!is.na	不为缺失值
>=	大于或等于	&, , !, xor, any, all	Boolean运算符

通过 devtools::install_github("rstudio/EDAWR") 获取数据集

通过命令 browseVignettes(package = c("dplyr", "tidyr")) 知道更多 • dplyr 0.4.0 • tidyr 0.2.0 • Updated:

概述数据



`dplyr::summarise(iris, avg = mean(Sepal.Length))`
将数据概括为单行数值。

`dplyr::summarise_each(iris, funs(mean))`
对每一列运行概述函数。

`dplyr::count(iris, Species, wt = Sepal.Length)`
计算各变量中每一个特定值的行数（带权重或不带权重）。



利用概述函数概括数据信息，输入数值向量而返回单一数值，如：

`dplyr::first`
向量的第一个值。

`dplyr::last`
向量的最后一个值。

`dplyr::nth`
向量的第n个值。

`dplyr::n`
向量中元素的个数。

`dplyr::n_distinct`
向量中的不同元素的个数。

`IQR`
向量的IQR（四分位距）。

`min`
向量中的最小值。

`max`
向量中的最大值。

`mean`
向量中的均值。

`median`
向量中的中位数。

`var`
向量中的方差。

`sd`
向量中的标准差。

分组数据

`dplyr::group_by(iris, Species)`
把在Species中的值相同的数据组合成行。

`dplyr::ungroup(iris)`
从数据框中移除组合信息。

`iris %>% group_by(Species) %>% summarise(...)`
为每一个分组分别计算行概述。



创建新变量



`dplyr::mutate(iris, sepal = Sepal.Length + Sepal.Width)`
计算并添加一个或多个新列。

`dplyr::mutate_each(iris, funs(min_rank))`
对每一列运行窗体函数。

`dplyr::transmute(iris, sepal = Sepal.Length + Sepal.Width)`
计算一个或多个新列并删除原列。



利用窗体函数变换数据，输入数值向量而返回另外的数值向量，如：

`dplyr::lead`
把除第一个值以外的所有元素提前，最后一个元素为NA。

`dplyr::lag`
把除最后一个值以外的元素延后，第一个元素为NA。

`dplyr::dense_rank`
无缝排序。

`dplyr::min_rank`
排序。并列时，其他序号顺延。

`dplyr::percent_rank`
把数据在[0, 1]中重组并排列。

`dplyr::row_number`
排序。并列时，位置在前的并列数据序号在前。

`dplyr::ntile`
把向量分为n份。

`dplyr::between`
数值是否在a和b之间？

`dplyr::cume_dist`
累积分布。

`dplyr::cumall`
累积all函数

`dplyr::cumany`
累积any函数

`dplyr::cummean`
累积mean函数

`cumsum`
累积sum函数

`cummax`
累积max函数

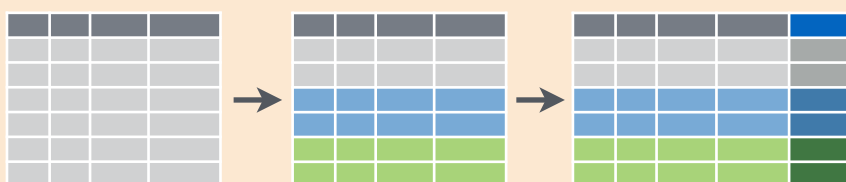
`cummin`
累积min函数

`cumprod`
累积prod函数

`pmax`
针对元素的max函数

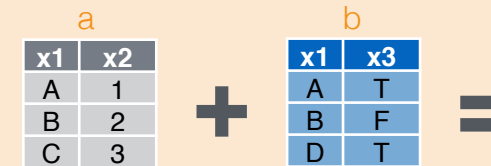
`pmin`
针对元素的min函数

`iris %>% group_by(Species) %>% mutate(...)`
按组计算新变量。



通过 `devtools::install_github("rstudio/EDAWR")` 获取数据集

合并数据集



转换与合并

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

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

x1	x2	x3
A	1	T
B	2	F

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

`dplyr::left_join(a, b, by = "x1")`
向数据集a中加入匹配的数据集b记录。

`dplyr::right_join(a, b, by = "x1")`
向数据集b中加入匹配的数据集a记录。

`dplyr::inner_join(a, b, by = "x1")`
合并数据。仅保留匹配的记录。

`dplyr::outer_join(a, b, by = "x1")`
合并数据。保留所有记录，所有行。

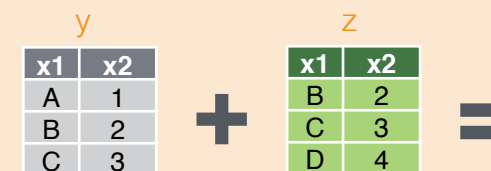
筛选与合并

x1	x2
A	1
B	2

x1	x2
C	3

`dplyr::semi_join(a, b, by = "x1")`
数据集a中能和数据集b匹配的记录。

`dplyr::anti_join(a, b, by = "x1")`
数据集a中与数据集b不匹配的记录。



集处理

x1	x2
B	2
C	3

x1	x2
A	1
B	2
C	3
D	4

x1	x2
A	1

`dplyr::intersect(y, z)`
均出现在数据集y和z中的记录。

`dplyr::union(y, z)`
出现在数据集y或z中的记录。

`dplyr::setdiff(y, z)`
仅出现在数据集y而不在z中的记录。

捆绑

x1	x2
A	1
B	2
C	3
D	4

`dplyr::bind_rows(y, z)`
把数据集z作为新的行添加到y中。

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

`dplyr::bind_cols(y, z)`
把数据集z作为新的列添加到y中。
注意：数据按所在位置匹配。