R for bioinformatics, data wrangler, part 1

HUST Bioinformatics course series

Wei-Hua Chen (CC BY-NC 4.0)

23 September, 2024

section 1: TOC

前情提要

1. IO, project management, working environment management
2. factors: R 中最重要的概念之一

- factors 基本概念
- factors 操作
- factors 在做图中的使用
- ggplot2 和 pipe %>% 初步

今次提要

- pipe
- dplyr 、tidyr (超级强大的数据处理) part 1

section 2: pipe

什么是 pipe ？

- pipe 就是 %>%
- it comes from the magrittr package by **Stefan Milton Bache**
- **实质是中间值的传递**

示例:

比较传统代码（如下）**和** pipe 代码（下页）:

```
## 传统代码:
a <- subset( swiss, Fertility > 20 );
cor.test(a$Fertility, a$Education);
```

```
##
##  Pearson's product-moment correlation
##
## data:  a$Fertility and a$Education
## t = -5.9536, df = 45, p-value = 3.659e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7987075 -0.4653206
## sample estimates:
##       cor
## -0.6637889
```

pipe 版本

```
## -- 新代码 ...
library(tidyverse); ## 装入包
library(magrittr);
swiss %>%
  subset(., Fertility > 20) %$%
  cor.test( Education , Fertility );
```

```
##
##  Pearson's product-moment correlation
##
## data:  Education and Fertility
## t = -5.9536, df = 45, p-value = 3.659e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7987075 -0.4653206
## sample estimates:
##       cor
## -0.6637889
```

是否所有函数都支持 pipe ？

是的。

通常需要用 . 指代传递来的数据，并以参数的形式赋予下游函数:

```
swiss %>% do( head(.,  n = 4 ) );
```

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
##              Infant.Mortality
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
```

以上也可写为：

```
swiss %>% head(.,  n = 4 );
```

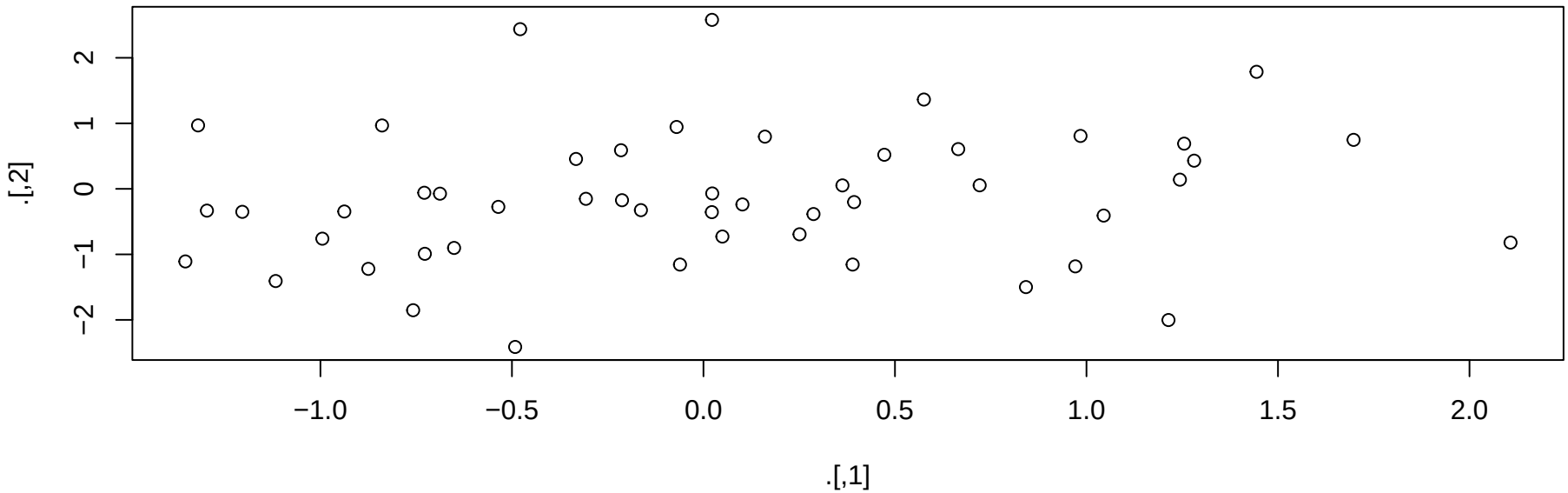
```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0           15        12     9.96
## Delemont          83.1        45.1            6         9    84.84
## Franches-Mnt      92.5        39.7            5         5    93.40
## Moutier           85.8        36.5           12         7    33.77
##              Infant.Mortality
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
```

注：这里的 . 被称为 占位符，用于指代上游传递来的数据。

其它形式的 pipe

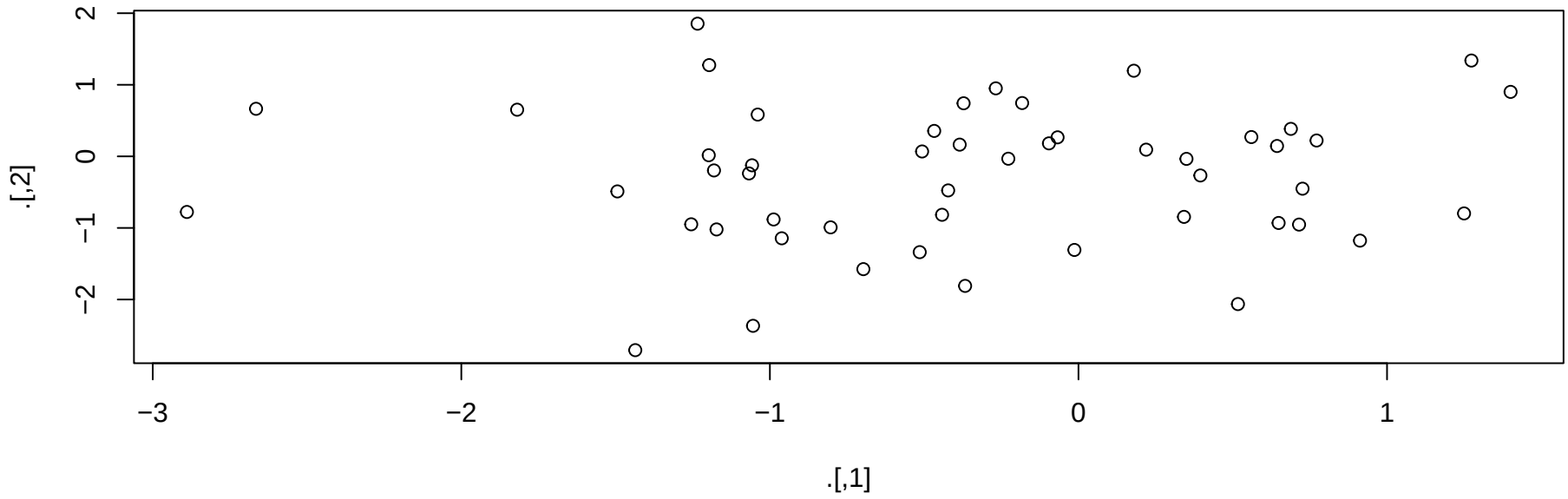
%T>%：返回上游的值（???）

```
## 示例：res1 是空值 ...
res1 <-
  rnorm(100) %>%
    matrix(ncol = 2) %>%
    plot();
```



%T>%: 返回上游值 (left-side values)

```
## 示例: res2 是 matrix() 内容 ...
res2 <-
  rnorm(100) %>%
    matrix(ncol = 2) %T>%
    plot();
```



%T>%: 返回上游值 (left-side values), cont.

```
head(res2);
```

```
##             [,1]        [,2]
## [1,] -0.5144860 -1.33839316
## [2,] -1.0546155 -2.36770028
## [3,] -0.5067454  0.06834602
## [4,] -0.0676967  0.26577224
## [5,] -1.4362156 -2.70771764
## [6,]  0.2193216  0.09351415
```

%$% : attach ???

```
attach( mtcars ); ## note the warning message ...
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

```
cor.test( cyl, mpg ); ## 汽缸数与燃油效率
```

```
##
##  Pearson's product-moment correlation
##
## data:  cyl and mpg
## t = -8.9197, df = 30, p-value = 6.113e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9257694 -0.7163171
## sample estimates:
##       cor
## -0.852162
```

%$% : attach ??? , cont.

```
detach( mtcars );
with( mtcars, cor.test( cyl, mpg ) );
```

```
##
##  Pearson's product-moment correlation
##
## data:  cyl and mpg
## t = -8.9197, df = 30, p-value = 6.113e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9257694 -0.7163171
## sample estimates:
##       cor
## -0.852162
```

%$% : attach ??? , cont.

```
mtcars %$%
  cor.test( cyl, mpg );
```

```
##
##  Pearson's product-moment correlation
##
## data:  cyl and mpg
## t = -8.9197, df = 30, p-value = 6.113e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9257694 -0.7163171
## sample estimates:
##       cor
## -0.852162
```

其它 pipe 及注意事项

```
## 双向 pipe
mtcars %<>% transform(cyl = cyl * 2);
```

注

- pipe 的使用可以使思路更清晰
- 因此，尽量使用 %>% （方向明确），而不使用其它方向不明确的 pipe

R 自带的 pipe

自 R 4.1.0 起，系统自带了 |> ，其用法与 %>% 基本相同，但也有不少区别，详见：

https://www.tidyverse.org/blog/2023/04/base-vs-magrittr-pipe/

本教程重点介绍 %>%。

pipe 的更多用法

之后会详细介绍。

section 3: data wrangler - dplyr

dplyr 提要

- 什么是 dplyr
- dplyr 安装
- mouse gene 分析示例
- 学习成绩分析示例
- 星战人物分析示例

1. 什么是 dplyr

what is dplyr ?

- the next iteration of plyr,
- focusing on only data frames (also tibble),
- **row-based** manipulation,
- dplyr is faster and has a more consistent API.

dplyr 函数概览

dplyr provides a consistent set of verbs that help you **solve the most common data manipulation challenges**:

- select() 选择列，根据列名规则
- filter() 按规则过滤行
- mutate() 增加新列，从其它列计算而得（不改变行数）
- summarise() 将多个值转换为单个值（通过 mean, median, sd 等操作），生成新列（总行数减少，通常与 `group_by` 配合使用）
- arrange() 对行进行排序

2. dplyr 安裝

正常安裝

```
# The easiest way to get dplyr is to install the whole tidyverse:
install.packages("tidyverse")

# Alternatively, install just dplyr:
install.packages("dplyr")
```

Development version

```
# install.packages("devtools")
devtools::install_github("tidyverse/dplyr")
```

Get the cheatsheet at here.

3. mouse gene 分析示例

get the data ready

```
mouse.tibble <- read_delim( file = "data/talk04/mouse_genes_biomart_sep2018.txt",
                            delim = "\t", quote = "" );
```

```
## Rows: 138532 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): Gene stable ID, Transcript stable ID, Protein stable ID, Transcript...
## dbl (1): Transcript length (including UTRs and CDS)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

查看 mouse.tibble 的內容

```
( ttype.stats <- mouse.tibble %>% count( `Transcript type` ) %>% arrange(-n) );
```

```
## # A tibble: 48 x 2
##    `Transcript type`            n
##    <chr>                    <int>
##  1 protein_coding           58384
##  2 retained_intron          21021
##  3 processed_transcript     15572
##  4 processed_pseudogene      9425
##  5 lincRNA                   8557
##  6 nonsense_mediated_decay   6755
##  7 antisense                 4289
##  8 TEC                       3265
##  9 unprocessed_pseudogene    2650
## 10 miRNA                     2265
## # i 38 more rows
```

查看 mouse.tibble 的內容, cont.

```
( chr.stats <- mouse.tibble %>% count( `Chromosome/scaffold name` ) %>% arrange(-n) );
```

```
## # A tibble: 117 x 2
##    `Chromosome/scaffold name`     n
##    <chr>                      <int>
##  1 7                          12344
##  2 2                          10877
##  3 5                           8955
##  4 11                          8673
##  5 1                           8553
##  6 9                           8030
##  7 6                           7845
##  8 4                           7573
##  9 3                           6938
## 10 10                          6568
## # i 107 more rows
```

分析任务

1. 将染色体限制在常染色体和 XY 上（去掉未组装的小片段）；处理行
2. 将基因类型限制在 protein_coding, miRNA 和 lincRNA 这三种；处理行
3. 统计每条染色体上不同类型基因（protein_coding, miRNA, lincRNA）的数量
4. 按染色体（正）、基因数量（倒）进行排序

用 dplyr 实现

```
dat <- mouse.tibble %>%
  ## 1.

  filter( `Chromosome/scaffold name` %in% c( 1:19, "X", "Y" )   ) %>%

  ## 2.
  filter( `Transcript type` %in% c( "protein_coding", "miRNA", "lincRNA" ) ) %>%

  ## change column name ...
  select( CHR = `Chromosome/scaffold name`, TYPE = `Transcript type`,
          GENE_ID = `Gene stable ID`,
          GENE_LEN =  `Transcript length (including UTRs and CDS)`  ) %>%

  ## 3.
  group_by( CHR, TYPE ) %>%
  summarise( count = n_distinct( GENE_ID ), mean_len = mean( GENE_LEN ) ) %>%

  ## 4.
  arrange(  CHR  , desc( count ) );
```

```
## `summarise()` has grouped output by 'CHR'. You can override using the `.groups`
## argument.
```

检查运行结果

CHR	TYPE	count	mean_len
1	protein_coding	1200	2699.59009
1	lincRNA	347	1206.76149
1	miRNA	128	97.97656
10	protein_coding	1020	2408.16454
10	lincRNA	398	1220.35543
10	miRNA	91	89.87912
11	protein_coding	1640	2431.87666
11	lincRNA	189	1134.49174
11	miRNA	137	87.48905
12	protein_coding	644	2523.94822
12	lincRNA	327	1277.14979
12	miRNA	146	86.24658
13	protein_coding	831	2380.41499
13	lincRNA	428	1251.04552
13	miRNA	97	105.52577

dplyr 中其它取行的操作

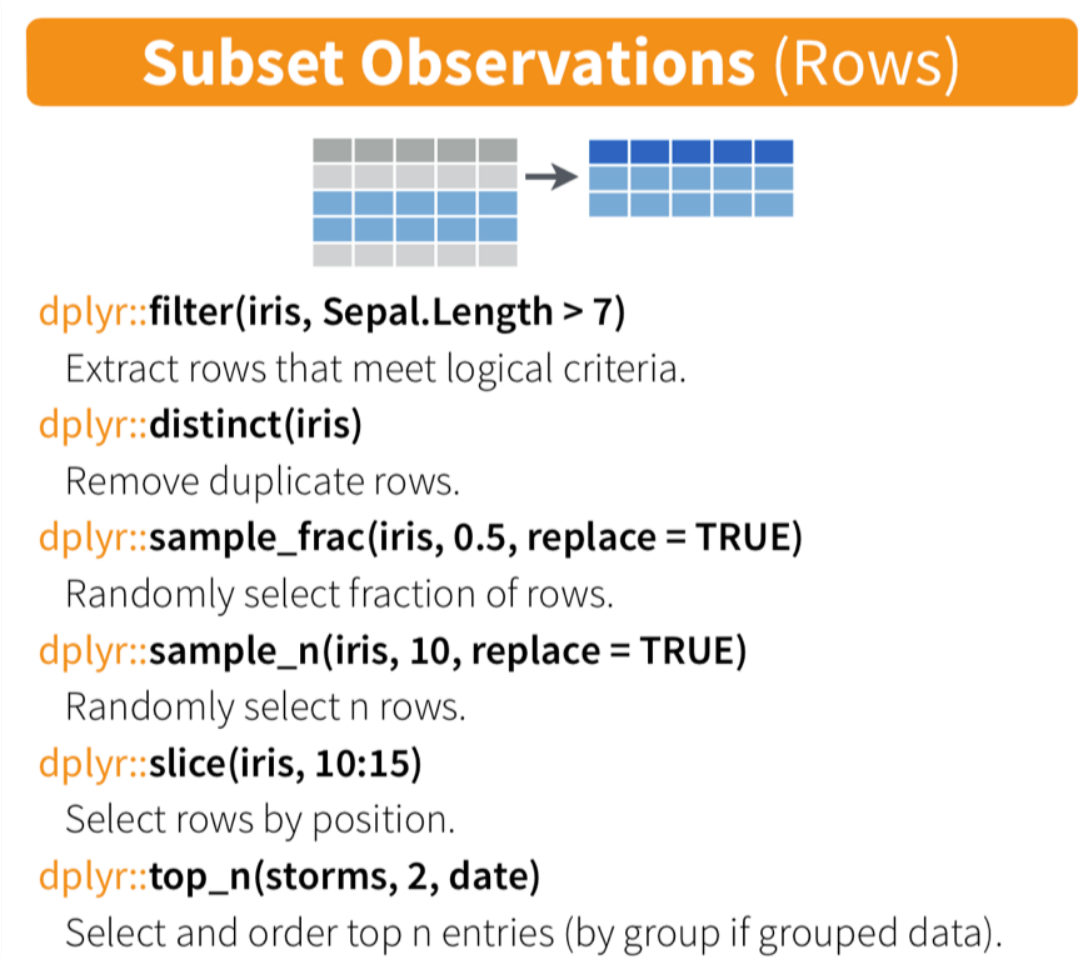


Figure 2: dplyr 与行相关的操作

4. 学习成绩分析示例

先创建一个新 tibble

```
grades <- tibble( "Name" = c("Weihua Chen", "Mm Hu", "John Doe", "Jane Doe",
                             "Warren Buffet", "Elon Musk", "Jack Ma"),
                  "Occupation" = c("Teacher", "Student", "Teacher", "Student",
                                   rep( "Entrepreneur", 3 ) ),
                  "English" = sample( 60:100, 7 ),
                  "ComputerScience" = sample(80:90, 7),
                  "Biology" = sample( 50:100, 7),
                  "Bioinformatics" = sample( 40:90, 7)
                  );
kbl(grades);
```

Name	Occupation	English	ComputerScience	Biology	Bioinformatics
Weihua Chen	Teacher	90	83	68	78
Mm Hu	Student	84	85	51	52
John Doe	Teacher	92	86	99	45
Jane Doe	Student	67	80	66	46
Warren Buffet	Entrepreneur	85	81	61	81
Elon Musk	Entrepreneur	82	82	77	60
Jack Ma	Entrepreneur	69	90	94	49

use gather & dplyr functions

Question: 1. 每个人平均成绩是多少？2. 哪个人的平均成绩最高？

```
grades.melted <- grades %>%
  gather( course, grade, -Name, -Occupation, na.rm = T );

## 检查数据 ...
kbl( head(grades.melted) );
```

Name	Occupation	course	grade
Weihua Chen	Teacher	English	90
Mm Hu	Student	English	84
John Doe	Teacher	English	92
Jane Doe	Student	English	67
Warren Buffet	Entrepreneur	English	85
Elon Musk	Entrepreneur	English	82

注： gather 的用法之后介绍

成绩分析，cont

```
stat3 <-
  grades.melted %>%
  group_by(Name, Occupation) %>%
  summarise( avg_grades = mean( grade ), courses_count = n() ) %>%
  arrange( -avg_grades );
```

```
## `summarise()` has grouped output by 'Name'. You can override using the
## `.groups` argument.
```

```
## 显示最终结果
kbl( head( stat3 ) );
```

Name	Occupation	avg_grades	courses_count
John Doe	Teacher	80.50	4
Weihua Chen	Teacher	79.75	4
Warren Buffet	Entrepreneur	77.00	4
Jack Ma	Entrepreneur	75.50	4
Elon Musk	Entrepreneur	75.25	4
Mm Hu	Student	68.00	4

use gather & dplyr functions

问题：每个人的最强科目是什么 ??

```
## 步骤 1: 排序:
grades.melted2 <-
  grades.melted %>%
  arrange( Name, -grade );

kbl( head(grades.melted2) );
```

Name	Occupation	course	grade
Elon Musk	Entrepreneur	English	82
Elon Musk	Entrepreneur	ComputerScience	82
Elon Musk	Entrepreneur	Biology	77
Elon Musk	Entrepreneur	Bioinformatics	60
Jack Ma	Entrepreneur	Biology	94
Jack Ma	Entrepreneur	ComputerScience	90

最强科目问题，cont.

```
## 步骤 2: 分组、每组取第一个
best_by_course<-
  grades.melted2 %>%
  group_by(Name) %>%
  summarise( best_course = first( course ),
             best_grade = first( grade ),
             avg_grades = mean( grade ) ) %>%
  arrange( -avg_grades );

kbl( best_by_course );
```

Name	best_course	best_grade	avg_grades
John Doe	Biology	99	80.50
Weihua Chen	English	90	79.75
Warren Buffet	English	85	77.00
Jack Ma	Biology	94	75.50
Elon Musk	English	82	75.25
Mm Hu	ComputerScience	85	68.00
Jane Doe	ComputerScience	80	64.75

dplyr::summarise 的其它操作

dplyr::**first**
First value of a vector.

dplyr::**last**
Last value of a vector.

dplyr::**nth**
Nth value of a vector.

dplyr::**n**
# of values in a vector.

dplyr::**n_distinct**
# of distinct values in a vector.

IQR
IQR of a vector.

min
Minimum value in a vector.

max
Maximum value in a vector.

mean
Mean value of a vector.

median
Median value of a vector.

var
Variance of a vector.

sd
Standard deviation of a vector.

Figure 3: dplyr::summarise 可用的操作

练习考察

问题 1：每个人的最差科目是什么 ??

5. 星战人员分析示例：列的使用！

```
kbl(head(starwars));
```

name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gend
Luke Skywalker	172	77	blond	fair	blue	19.0	male	masc
C-3PO	167	75	NA	gold	yellow	112.0	none	masc
R2-D2	96	32	NA	white, blue	red	33.0	none	masc
Darth Vader	202	136	none	white	yellow	41.9	male	masc
Leia Organa	150	49	brown	light	brown	19.0	female	femi
Owen Lars	178	120	brown, grey	light	blue	52.0	male	masc

note 包含 87 行 13 列，星战部分人物的信息，包括身高、体重、肤色等

用 `?starwars` 获取更多帮助

dplyr::mutate - 产生新列，不改变行数

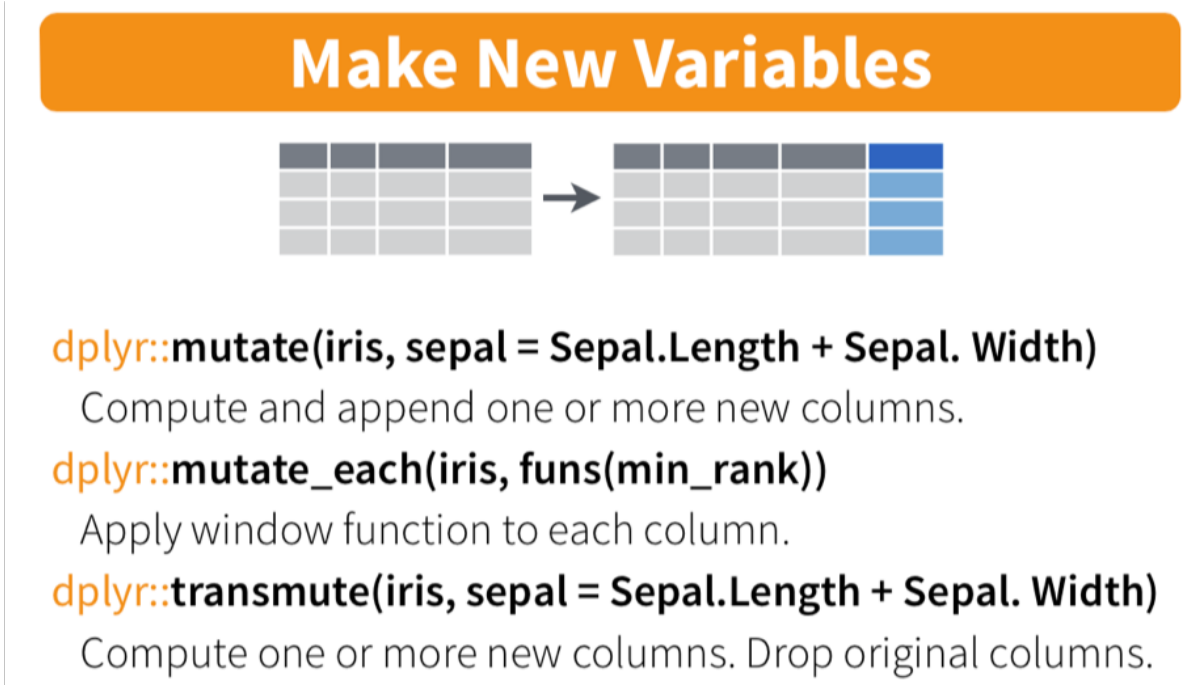


Figure 4: dplyr::mutate

另见下页的例子

dplyr::select - 取列

目标：

- 取出相关列，用于计算人物的 BMI

```
stats <-
  starwars %>%
  select( name, height, mass ) %>%
  mutate( bmi = mass / ( (height / 100 ) ^ 2 ) ) ;

kbl(head(stats));
```

name	height	mass	bmi
Luke Skywalker	172	77	26.02758
C-3PO	167	75	26.89232
R2-D2	96	32	34.72222
Darth Vader	202	136	33.33007
Leia Organa	150	49	21.77778
Owen Lars	178	120	37.87401

dplyr::select - 取列, cont.

由于 name, height 和 mass 正好是相邻列，可以用 name:mass 获取：

```
stats <-
  starwars %>%
  select( name:mass ) %>%
  mutate( bmi = mass / ( (height / 100 ) ^ 2 ) ) ;

kbl(head(stats));
```

name	height	mass	bmi
Luke Skywalker	172	77	26.02758
C-3PO	167	75	26.89232
R2-D2	96	32	34.72222
Darth Vader	202	136	33.33007
Leia Organa	150	49	21.77778
Owen Lars	178	120	37.87401

dplyr::select - 取列, cont.

获取与颜色相关的列: hair_color, skin_color, eye_color

```
stats2 <- starwars %>%
  select( name, ends_with("color") );

kbl(head(stats2));
```

name	hair_color	skin_color	eye_color
Luke Skywalker	blond	fair	blue
C-3PO	NA	gold	yellow
R2-D2	NA	white, blue	red
Darth Vader	none	white	yellow
Leia Organa	brown	light	brown
Owen Lars	brown, grey	light	blue

dplyr::select - 去除列, cont.

请自行检查以下操作的结果

```
kbl(head( starwars %>% select( -hair_color, -eye_color )  ));
```

dplyr::select - 其它操作, cont.

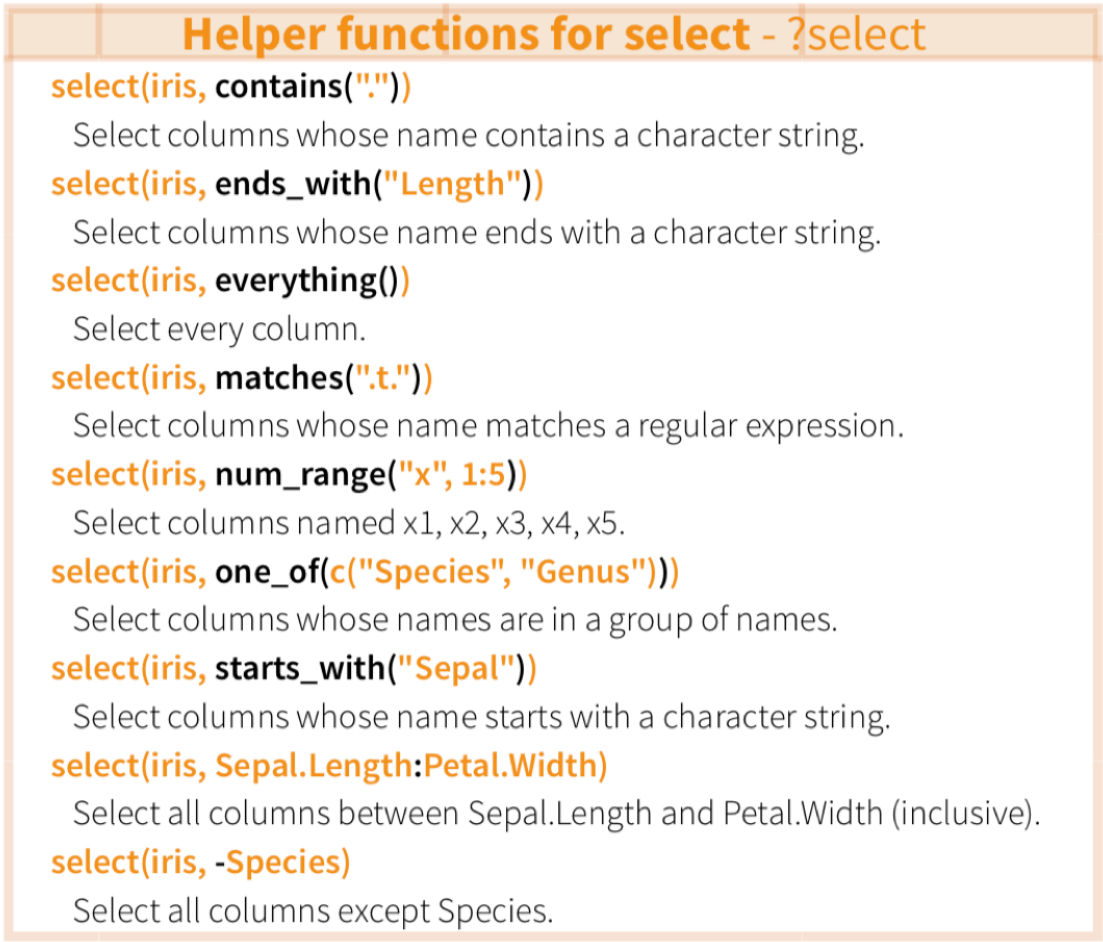


Figure 5: dplyr::select 支持的操作

同时对行列进行操作

任务：从星战中挑选金发碧眼的人物

```
starwars %>% select( name, ends_with("color"), gender, species ) %>%
  filter( hair_color == "blond" & eye_color == "blue" );
```

```
## # A tibble: 3 x 6
##   name             hair_color skin_color eye_color gender    species
##   <chr>            <chr>      <chr>      <chr>     <chr>     <chr>
## 1 Luke Skywalker   blond      fair       blue      masculine Human
## 2 Anakin Skywalker blond      fair       blue      masculine Human
## 3 Finis Valorum    blond      fair       blue      masculine Human
```

练习考察

问题 2：从 `starwars` 中选出 $18 < \text{bmi} < 25$ 的人物，统计他们的 `homeworld` 分布情况

提示：1. 需计算 bmi；2. 用 `count` 函数计算 `homeworld` 的分布

section 4：练习与作业

小结

今次提要

- pipe
- dplyr 、tidyr (超级强大的数据处理) part 1

下次预告

- 长宽数据转换
- dplyr, tidyr 和 forcats 的更多功能与生信操作实例

important

- all codes are available at Github:
  `https://github.com/evolgeniusteam/R-for-bioinformatics`

练习 & 作业

- Exercises and homework 目录下 talk05-homework.Rmd 文件;
- 完成时间：见钉群的要求