talk05 练习与作业

目录

0.1 练习和作业说明

将相关代码填写入以 "'{r} "' 标志的代码框中，运行并看到正确的结果；

完成后，用工具栏里的"Knit" 按键生成 PDF 文档；

将 PDF 文档改为：`姓名-学号-talk05 作业.pdf`，并提交到老师指定的平台/钉群。

0.2 Talk05 内容回顾

- dplyr 、tidyr (超级强大的数据处理) part 1
  - pipe
  - dplyr 几个重要函数

0.3 练习与作业：用户验证

请运行以下命令，验证你的用户名。

如你当前用户名不能体现你的真实姓名，请改为拼音后再运行本作业！

```
Sys.info()[["user"]]
```

```
## [1] "Zhu Fangannan"
```

```
Sys.getenv("HOME")
```

```
## [1] "C:/Users/Zhu Fangannan/Documents"
```

```
getwd(); ## 显示当前工作目录
```

```
## [1] "D:/R-for-data-science/Exercises and homework"
```

0.4 练习与作业 1：dplyr 练习

0.4.1 使用 mouse.tibble 变量做统计

- 每个染色体（或 scaffold）上每种基因类型的数量、平均长度、最大和最小长度，挑出最长和最短的基因
- 去掉含有 500 以下基因的染色体（或 scaffold），按染色体（或 scaffold）、数量高 -> 低进行排序

挑战题（可选做）：

实现上述目标（即：去掉少于 500 基因的染色体、排序、并统计）时不使用中间变量；

```
## 代码写这里，并运行；
```

0.4.2 使用 grades2 变量做练习

首先，用下面命令生成 grades2 变量：

```
grades2 <- 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)
                  );
```

然后统计：1. 每个人最差的学科和成绩分别是什么？2. 哪个职业的平均成绩最好？3. 每个职业的最佳学科分别是什么（按平均分排序)???

```
## 代码写这里，并运行；
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
```

```
library(dplyr)
grades2 <- 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)
                  );
grades2
```

```
## # A tibble: 7 x 6
##   Name          Occupation   English ComputerScience Biology Bioinformatics
##   <chr>         <chr>          <int>           <int>   <int>          <int>
## 1 Weihua Chen   Teacher           80              80      85             52
## 2 Mm Hu         Student           87              81      65             49
## 3 John Doe      Teacher           69              87      81             79
## 4 Jane Doe      Student           62              86      83             53
## 5 Warren Buffet Entrepreneur      68              82      58             58
## 6 Elon Musk     Entrepreneur      65              90      94             45
## 7 Jack Ma       Entrepreneur      96              85      66             82
```

```
grades.melted<-grades2 %>%
gather(course,grade,-Name,-Occupation,na.rm=T);

grades.melted2<-
  grades.melted %>%
arrange(Name,-grade);

grades.melted2 %>%
group_by(Name) %>%
summarise(worst_course=last(course),
worst_grade=last(grade)) %>%
  arrange(-worst_grade);
```

```
## # A tibble: 7 x 3
##   Name          worst_course   worst_grade
##   <chr>         <chr>                <int>
## 1 John Doe      English                 69
## 2 Jack Ma       Biology                 66
## 3 Warren Buffet Bioinformatics          58
## 4 Jane Doe      Bioinformatics          53
## 5 Weihua Chen   Bioinformatics          52
## 6 Mm Hu         Bioinformatics          49
## 7 Elon Musk     Bioinformatics          45
```

```
grades.melted3<-grades2 %>%
gather(course,grade,-Name,-Occupation,na.rm=T);
grades.melted3%>%
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.
```

```
## # A tibble: 7 x 4
## # Groups:   Name [7]
##   Name          Occupation   avg_grades courses_count
##   <chr>         <chr>             <dbl>         <int>
## 1 Jack Ma       Entrepreneur       82.2             4
## 2 John Doe      Teacher            79               4
## 3 Weihua Chen   Teacher            74.2             4
## 4 Elon Musk     Entrepreneur       73.5             4
## 5 Jane Doe      Student            71               4
## 6 Mm Hu         Student            70.5             4
## 7 Warren Buffet Entrepreneur       66.5             4
```

```
  grades.melted4<-grades.melted3 %>%
gather(courses_count,avg_grades,-Name,-Occupation,na.rm=T);
grades.melted4%>%
group_by(Occupation)%>%
summarise(avg=mean(avg_grades))%>%
arrange(-avg);
```

```
## Warning: There were 3 warnings in `summarise()`.
## The first warning was:
```

```
## i In argument: `avg = mean(avg_grades)`.
## i In group 1: `Occupation = "Entrepreneur"`.
## Caused by warning in `mean.default()`:
## ! 参数不是数值也不是逻辑值：回覆NA
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
## # A tibble: 3 x 2
##   Occupation     avg
##   <chr>        <dbl>
## 1 Entrepreneur    NA
## 2 Student         NA
## 3 Teacher         NA
```

0.4.3 使用 starwars 变量做计算

1. 计算每个人的 BMI；
2. 挑选出肥胖（BMI >= 30）的人类，并且只显示其 name，sex 和 homeworld；

```
## 代码写这里，并运行；
head(starwars);
```

```
## # A tibble: 6 x 14
##   name      height  mass hair_color skin_color eye_color birth_year sex   gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
## 1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
## 2 C-3PO        167    75 <NA>       gold       yellow         112   none  mascu~
## 3 R2-D2         96    32 <NA>       white, bl~ red             33   none  mascu~
## 4 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
## 5 Leia Org~    150    49 brown      light      brown           19   fema~ femin~
## 6 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
## # i 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
stats<-
starwars%>%
select(name,height,mass,gender,homeworld,species)%>%
mutate(bmi=mass/((height/100)*(height/100)));
head(stats);
```

```
## # A tibble: 6 x 7
##   name           height  mass gender    homeworld species   bmi
##   <chr>           <int> <dbl> <chr>     <chr>     <chr>   <dbl>
## 1 Luke Skywalker    172    77 masculine Tatooine  Human    26.0
## 2 C-3PO             167    75 masculine Tatooine  Droid    26.9
## 3 R2-D2              96    32 masculine Naboo     Droid    34.7
## 4 Darth Vader       202   136 masculine Tatooine  Human    33.3
## 5 Leia Organa       150    49 feminine  Alderaan  Human    21.8
## 6 Owen Lars         178   120 masculine Tatooine  Human    37.9
```

```
stats2<-stats%>%select(name,gender,homeworld,bmi,species)%>%
filter(bmi>=30&species=="Human");
head(stats2%>%select(-bmi,-species));
```

```
## # A tibble: 3 x 3
##   name            gender    homeworld
##   <chr>           <chr>     <chr>
## 1 Darth Vader     masculine Tatooine
## 2 Owen Lars       masculine Tatooine
## 3 Jek Tono Porkins masculine Bestine IV
```

3. 挑选出所有人类；
4. 按 BMI 将他们分为三组，<18, 18~25, >25，统计每组的人数，并用 barplot 进行展示；注意：展示时三组的按 BMI 从小到大排序；
5. 改变排序方式，按每组人数从小到大排序；

```
## 代码写这里，并运行；
head(starwars);
```

```
## # A tibble: 6 x 14
##   name      height  mass hair_color skin_color eye_color birth_year sex   gender
##   <chr>      <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
## 1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
## 2 C-3PO        167    75 <NA>       gold       yellow         112   none  mascu~
## 3 R2-D2         96    32 <NA>       white, bl~ red             33   none  mascu~
## 4 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
## 5 Leia Org~    150    49 brown      light      brown           19   fema~ femin~
## 6 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
## # i 5 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

```
stats0<-
starwars%>%
select(name,height,mass,gender,homeworld,species)%>%
mutate(bmi=mass/((height/100)*(height/100)));
stats3<-stats0%>%select(name,gender,homeworld,bmi,species)%>%
filter(species=="Human");
head(stats3);
```

```
## # A tibble: 6 x 5
##   name               gender    homeworld   bmi species
##   <chr>              <chr>     <chr>     <dbl> <chr>
## 1 Luke Skywalker     masculine Tatooine   26.0 Human
## 2 Darth Vader        masculine Tatooine   33.3 Human
## 3 Leia Organa        feminine  Alderaan   21.8 Human
## 4 Owen Lars          masculine Tatooine   37.9 Human
## 5 Beru Whitesun lars feminine  Tatooine   27.5 Human
## 6 Biggs Darklighter  masculine Tatooine   25.1 Human
```

6. 查看 starwars 的 films 列，它有什么特点？data.frame 可以实现类似的功能吗？

答：适合屏幕且显示列的类型。不可以。

7. 为 starwars 增加一列，用于统计每个角色在多少部电影中出现。

```
## 代码写这里，并运行；
starwars%>%
mutate(count=lengths(films));
```

```
## # A tibble: 87 x 15
##    name     height  mass hair_color skin_color eye_color birth_year sex   gender
##    <chr>     <int> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
##  1 Luke Sk~    172    77 blond      fair       blue            19   male  mascu~
##  2 C-3PO       167    75 <NA>       gold       yellow         112   none  mascu~
##  3 R2-D2        96    32 <NA>       white, bl~ red             33   none  mascu~
##  4 Darth V~    202   136 none       white      yellow          41.9 male  mascu~
##  5 Leia Or~    150    49 brown      light      brown           19   fema~ femin~
##  6 Owen La~    178   120 brown, gr~ light      blue            52   male  mascu~
##  7 Beru Wh~    165    75 brown      light      blue            47   fema~ femin~
##  8 R5-D4        97    32 <NA>       white, red red             NA   none  mascu~
##  9 Biggs D~    183    84 black      light      brown           24   male  mascu~
## 10 Obi-Wan~    182    77 auburn, w~ fair       blue-gray       57   male  mascu~
## # i 77 more rows
## # i 6 more variables: homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>, count <int>
```

0.4.4 使用 Theoph 变量做练习

注：以下练习请只显示结果的前6行；

1. 选取从 Subject 到 Dose 的列；总共有几列？

```
## 代码写这里，并运行；
head(Theoph);
```

```
##   Subject   Wt Dose Time  conc
## 1       1 79.6 4.02 0.00  0.74
## 2       1 79.6 4.02 0.25  2.84
## 3       1 79.6 4.02 0.57  6.57
## 4        1 79.6 4.02 1.12 10.50
## 5        1 79.6 4.02 2.02  9.66
## 6        1 79.6 4.02 3.82  8.58
```

```
the<-
Theoph%>%
select(Subject:Dose)
head(the);
```

```
##   Subject   Wt Dose
## 1        1 79.6 4.02
## 2        1 79.6 4.02
## 3        1 79.6 4.02
## 4        1 79.6 4.02
## 5        1 79.6 4.02
## 6        1 79.6 4.02
```

```
ncol(the)
```

```
## [1] 3
```

2. 用 filter 选取 Dose 大于 5，且 Time 高于 Time 列平均值的行；

```
## 代码写这里，并运行；
 average <- mean(Theoph$Time)
the1<-Theoph%>%filter(Dose>5&Time>average);
head(the1);
```

```
##   Subject   Wt Dose  Time conc
## 1        5 54.6 5.86  7.02 7.09
## 2        5 54.6 5.86  9.10 5.90
## 3        5 54.6 5.86 12.00 4.37
## 4        5 54.6 5.86 24.35 1.57
## 5       10 58.2 5.50  7.08 8.02
## 6       10 58.2 5.50  9.38 7.14
```

3. 用 mutate 函数产生新列 trend，其值为 Time 与 Time 列平均值的差；

注意：请去除可能产生的 na 值；

```
## 代码写这里，并运行；
average1 <- mean(Theoph$Time)
the2<-
Theoph%>%
mutate(trend=Time-average1);
head(the2,);
```

```
##   Subject   Wt Dose Time  conc     trend
## 1       1 79.6 4.02 0.00  0.74 -5.894621
## 2       1 79.6 4.02 0.25  2.84 -5.644621
## 3       1 79.6 4.02 0.57  6.57 -5.324621
## 4       1 79.6 4.02 1.12 10.50 -4.774621
## 5       1 79.6 4.02 2.02  9.66 -3.874621
## 6       1 79.6 4.02 3.82  8.58 -2.074621
```

4. 用 mutate 函数产生新列 weight_cat ，其值根据 Wt 的取值范围而不同：

- 如果 Wt > 76.2，为 'Super-middleweight'，否则
- 如果 Wt > 72.57，为 'Middleweight'，否则
- 如果 Wt > 66.68，为 'Light-middleweight'
- 其它值，为 'Welterweight'

```
## 代码写这里，并运行；
the3<-
Theoph%>%
mutate(weight_cat =ifelse (Wt> 76.2,"Super-middleweight",ifelse(Wt> 72.57,"Middleweight
head(the3);
```

```
##   Subject   Wt Dose Time  conc         weight_cat
## 1       1 79.6 4.02 0.00  0.74 Super-middleweight
## 2       1 79.6 4.02 0.25  2.84 Super-middleweight
## 3       1 79.6 4.02 0.57  6.57 Super-middleweight
## 4       1 79.6 4.02 1.12 10.50 Super-middleweight
## 5        1 79.6 4.02 2.02  9.66 Super-middleweight
## 6        1 79.6 4.02 3.82  8.58 Super-middleweight
```