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
  - 长宽数据转换
  - 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] "s56hh"
```

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

```
## [1] "C:/Users/s56hh/Documents"
```

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

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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.
```

```
dat1 <- mouse.tibble %>%
       select( CHR = `Chromosome/scaffold name`, TYPE = `Transcript type`,
                              GENE_ID = `Gene stable ID`,
                              GENE_LEN = `Transcript length (including UTRs and CDS)` )
       arrange( CHR , -GENE_LEN ) %>%
       group_by( CHR, TYPE ) %>%
       summarise( count = n_distinct( GENE_ID ), mean_len = mean( GENE_LEN ),max_len=ma
```

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

```
dat1
```

```
## # A tibble: 919 x 8
## # Groups:   CHR [117]
##    CHR   TYPE                    count mean_len max_len min_len max_GENE min_G~1
##    <chr> <chr>                   <int>    <dbl>   <dbl>   <dbl> <chr>    <chr>
##  1 1     antisense                 224   1236.     7928      78 ENSMUSG~ ENSMUS~
##  2 1     lincRNA                   347   1207.     9720     154 ENSMUSG~ ENSMUS~
##  3 1     miRNA                     128     98.0     442      53 ENSMUSG~ ENSMUS~
##  4 1     misc_RNA                   30    239.      353      85 ENSMUSG~ ENSMUS~
```

```
##  5 1     non_stop_decay              2   1526      1800    1252 ENSMUSG~ ENSMUS~
##  6 1     nonsense_mediated_decay   314   1844.    10770     284 ENSMUSG~ ENSMUS~
##  7 1     polymorphic_pseudogene      2   1122      1194    1050 ENSMUSG~ ENSMUS~
##  8 1     processed_pseudogene      627    728.     4530      30 ENSMUSG~ ENSMUS~
##  9 1     processed_transcript      462    951.     7640      65 ENSMUSG~ ENSMUS~
## 10 1     protein_coding           1200   2700.    40378      75 ENSMUSG~ ENSMUS~
## # ... with 909 more rows, and abbreviated variable name 1: min_GENE
```

```
dat2 <- mouse.tibble %>%
       select( CHR = `Chromosome/scaffold name`, TYPE = `Transcript type`,
                              GENE_ID = `Gene stable ID`,
                              GENE_LEN = `Transcript length (including UTRs and CDS)` )
       group_by( CHR ) %>%
       summarise( count = n_distinct( GENE_ID ), mean_len = mean( GENE_LEN ),max_len=ma
       filter(count>500) %>%
       arrange( CHR,-count )
dat2
```

```
## # A tibble: 21 x 7
##    CHR   count mean_len max_len min_len max_GENE           min_GENE
##    <chr> <int>    <dbl>   <dbl>   <dbl> <chr>              <chr>
##  1 1      3443    1875.   40378      30 ENSMUSG00000097109 ENSMUSG00000026014
##  2 10     2630    1723.  123179      51 ENSMUSG00000097353 ENSMUSG00000111262
##  3 11     3011    1694.   24175      41 ENSMUSG00000097062 ENSMUSG00000020745
##  4 12     2481    1570.   21704      10 ENSMUSG00000097698 ENSMUSG00000021057
##  5 13     2505    1644.   52401      16 ENSMUSG00000097582 ENSMUSG00000113027
##  6 14     2532    1564.   93147       9 ENSMUSG00000097358 ENSMUSG00000002326
##  7 15     1851    1679.   19823      10 ENSMUSG00000097234 ENSMUSG00000115301
##  8 16     1594    1657.   26901      38 ENSMUSG00000097551 ENSMUSG00000039200
##  9 17     1823    1697.   15768       9 ENSMUSG00000097676 ENSMUSG00000036594
## 10 18      904    2030.   13391      42 ENSMUSG00000097738 ENSMUSG00000040957
## # ... with 11 more rows
```

0.4.2 使用 grades 变量做练习

1. 装入 grades 变量；

```
library(dplyr); grades <- read_tsv( file = "data/talk05/grades.txt"
);
```

2. 尝试使用 spread 和 gather 函数将其变宽后再变长；

```
## 代码写这里，并运行；
library(dplyr);
grades <- read_tsv( file = "data/talk05/grades.txt" );
```

```
## Rows: 9 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): name, course
## dbl (1): grade
##
## 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.
```

```
library(tidyr);
grades_spread<-grades %>% spread( course, grade )
knitr::kable( grades_spread );
```

name	Bioinformatics	Chemistry	Chinese	English	Microbiology
Kang Ning	100	76	20	NA	NA
Weihua Chen	99	NA	NA	99	89
Zhi Liu	NA	NA	69	50	100

```
grades_gather<-grades_spread %>% gather( course, grade,-name )
knitr::kable( grades_gather );
```

name	course	grade
Kang Ning	Bioinformatics	100
Weihua Chen	Bioinformatics	99
Zhi Liu	Bioinformatics	NA
Kang Ning	Chemistry	76
Weihua Chen	Chemistry	NA
Zhi Liu	Chemistry	NA
Kang Ning	Chinese	20
Weihua Chen	Chinese	NA
Zhi Liu	Chinese	69
Kang Ning	English	NA
Weihua Chen	English	99
Zhi Liu	English	50
Kang Ning	Microbiology	NA
Weihua Chen	Microbiology	89
Zhi Liu	Microbiology	100

3. 研究并使用 `tidyr` 包里的 `pivot_longer` 和 `pivot_wider` 函数对 `grades` 变量进行宽长转换；

```
## 代码写这里，并运行；
library(tidyr);
library(dplyr);
grades <- read_tsv( file = "data/talk05/grades.txt" );
```

```
## Rows: 9 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): name, course
```

```
## dbl (1): grade
##
## 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.
```

```
grades_wider = pivot_wider(grades, names_from = 'course', values_from = 'grade')
knitr::kable(grades_wider)
```

name	Microbiology	English	Chinese	Bioinformatics	Chemistry
Zhi Liu	100	50	69	NA	NA
Weihua Chen	89	99	NA	99	NA
Kang Ning	NA	NA	20	100	76

```
grades_longer = pivot_longer(grades_wider,2:6, names_to = 'course', values_to = 'grade'
knitr::kable(grades_longer)
```

name	course	grade
Zhi Liu	Microbiology	100
Zhi Liu	English	50
Zhi Liu	Chinese	69
Zhi Liu	Bioinformatics	NA
Zhi Liu	Chemistry	NA
Weihua Chen	Microbiology	89
Weihua Chen	English	99
Weihua Chen	Chinese	NA
Weihua Chen	Bioinformatics	99
Weihua Chen	Chemistry	NA
Kang Ning	Microbiology	NA
Kang Ning	English	NA
Kang Ning	Chinese	20

name	course	grade
Kang Ning	Bioinformatics	100
Kang Ning	Chemistry	76

4. 使用 pivot_longer 时，有时会产生 na 值，如何使用此函数的参数去除带 na 的行？

```
## 代码写这里，并运行；
grades_longer1 = pivot_longer(grades_wider,2:6, names_to = 'course', values_to = 'grade
knitr::kable(grades_longer1)
```

name	course	grade
Zhi Liu	Microbiology	100
Zhi Liu	English	50
Zhi Liu	Chinese	69
Weihua Chen	Microbiology	89
Weihua Chen	English	99
Weihua Chen	Bioinformatics	99
Kang Ning	Chinese	20
Kang Ning	Bioinformatics	100
Kang Ning	Chemistry	76

5. 以下代码有什么作用？

```
grades %>% complete( name, course )
```

答：显示完整的含 NA 的表

0.4.3 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)
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_gather<-grades2 %>% gather( course, grade,-Name,-Occupation )
grades2_aaaa <- grades2_gather %>% arrange( Name, -grade );
grades2_aaaa %>%
group_by(Name) %>%
summarise( worst_course = last( course ),
worst_grade = last( grade ));
```

```
## # A tibble: 7 x 3
##   Name          worst_course   worst_grade
```

```
##   <chr>         <chr>                <int>
## 1 Elon Musk     Bioinformatics          44
## 2 Jack Ma       Bioinformatics          79
## 3 Jane Doe      Biology                 68
## 4 John Doe      Bioinformatics          76
## 5 Mm Hu         Bioinformatics          66
## 6 Warren Buffet Bioinformatics          58
## 7 Weihua Chen   Bioinformatics          51
```

```
grades2_aaaa %>%
group_by(Occupation) %>%
summarise( avg_grades = mean( grade ) ) %>%
arrange( -avg_grades );
```

```
## # A tibble: 3 x 2
##   Occupation   avg_grades
##   <chr>             <dbl>
## 1 Teacher            77.6
## 2 Student            77.2
## 3 Entrepreneur       76.6
```

```
grades2_aaaa %>%
group_by(Occupation) %>%
summarise( best_course = first( course ),
  avg_grades = mean( grade ) ) %>%
arrange( -avg_grades );
```

```
## # A tibble: 3 x 3
##   Occupation   best_course     avg_grades
##   <chr>        <chr>                <dbl>
## 1 Teacher      English               77.6
## 2 Student      ComputerScience       77.2
## 3 Entrepreneur English               76.6
```

0.4.4 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；
starwars %>% group_by(name) %>%
  summarise(BMI=mass/(height*height)*10000)
```

```
## # A tibble: 87 x 2
##    name                  BMI
##    <chr>               <dbl>
##  1 Ackbar               25.6
##  2 Adi Gallia           14.8
##  3 Anakin Skywalker     23.8
##  4 Arvel Crynyd         NA
##  5 Ayla Secura          17.4
##  6 Bail Prestor Organa  NA
##  7 Barriss Offee        18.1
##  8 BB8                  NA
##  9 Ben Quadinaros       24.5
## 10 Beru Whitesun lars   27.5
## # ... with 77 more rows
```

```
starwars %>% group_by(name) %>%
  summarise(BMI=mass/(height*height)*10000,
            name=name,
            sex=sex,
            homeworld=homeworld)%>%
  filter( BMI >= 30 ) %>%
  summarise(name=name,
            sex=sex,
            homeworld=homeworld)
```

```
## # A tibble: 12 x 3
##    name                  sex           homeworld
##    <chr>                 <chr>         <chr>
##  1 Bossk                 male          Trandosha
##  2 Darth Vader           male          Tatooine
##  3 Dud Bolt              male          Vulpter
##  4 Grievous              male          Kalee
##  5 IG-88                 none          <NA>
##  6 Jabba Desilijic Tiure hermaphroditic Nal Hutta
##  7 Jek Tono Porkins      male          Bestine IV
##  8 Owen Lars             male          Tatooine
##  9 R2-D2                 none          Naboo
## 10 R5-D4                 none          Tatooine
## 11 Sebulba               male          Malastare
## 12 Yoda                  male          <NA>
```

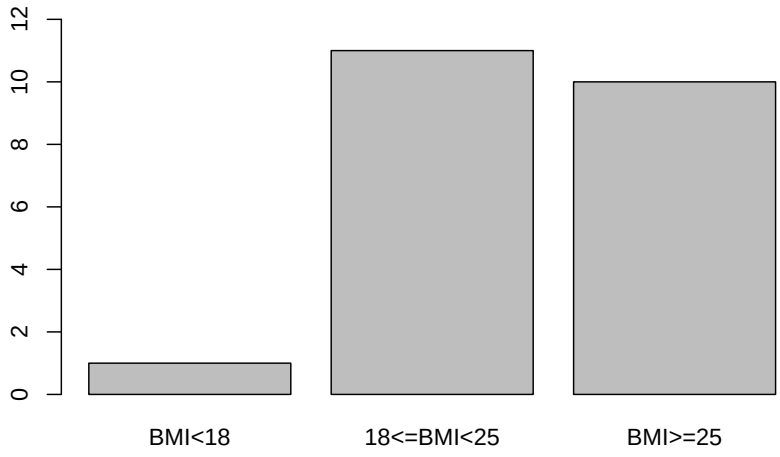
3. 挑选出所有人类；
4. 按 BMI 将他们分为三组，<18, 18~25, >25，统计每组的人数，并用 barplot 进行展示；注意：展示时三组的按 BMI 从小到大排序；
5. 改变排序方式，按每组人数从小到大排序；

```
## 代码写这里，并运行；
(Human<-starwars %>% filter(species == "Human"))
```

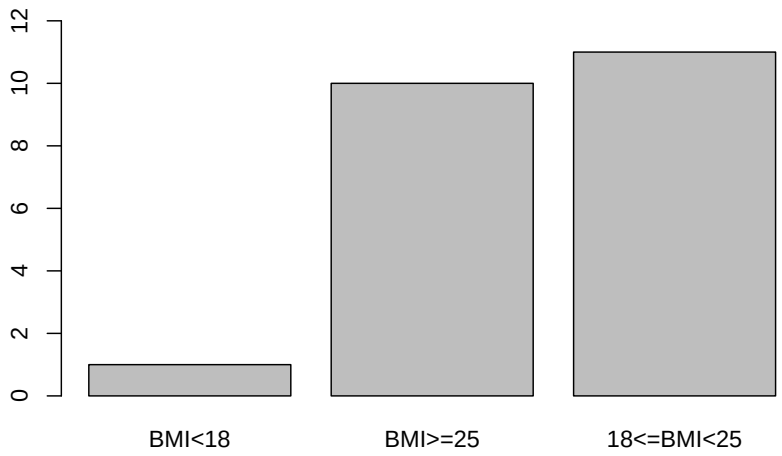
```
## # A tibble: 35 x 14
##    name        height  mass hair_~1 skin_~2 eye_c~3 birth~4 sex   gender homew~5
##    <chr>        <int> <dbl> <chr>   <chr>   <chr>     <dbl> <chr> <chr>  <chr>
##  1 Luke Skywa~    172    77 blond   fair    blue       19   male  mascu~ Tatooi~
##  2 Darth Vader    202   136 none    white   yellow     41.9 male  mascu~ Tatooi~
##  3 Leia Organa    150    49 brown   light   brown      19   fema~ femin~ Aldera~
##  4 Owen Lars      178   120 brown,~ light   blue       52   male  mascu~ Tatooi~
##  5 Beru White~    165    75 brown   light   blue       47   fema~ femin~ Tatooi~
##  6 Biggs Dark~    183    84 black   light   brown      24   male  mascu~ Tatooi~
##  7 Obi-Wan Ke~    182    77 auburn~ fair    blue-g~    57   male  mascu~ Stewjon
```

```
##  8 Anakin Sky~    188    84 blond   fair    blue       41.9 male  mascu~ Tatooi~
##  9 Wilhuff Ta~    180    NA auburn~ fair    blue       64   male  mascu~ Eriadu
## 10 Han Solo       180    80 brown   fair    brown      29   male  mascu~ Corell~
## # ... with 25 more rows, 4 more variables: species <chr>, films <list>,
## #   vehicles <list>, starships <list>, and abbreviated variable names
## #   1: hair_color, 2: skin_color, 3: eye_color, 4: birth_year, 5: homeworld
```

```
H1<-Human %>% group_by(name) %>%
  summarise(BMI=mass/(height*height)*10000)%>%filter(BMI<18)
H2<-Human %>% group_by(name) %>%
  summarise(BMI=mass/(height*height)*10000)%>%filter(BMI>=18,BMI<25)
H3<-Human %>% group_by(name) %>%
  summarise(BMI=mass/(height*height)*10000)%>%filter(BMI>=25)
barplot(height = c(nrow(H1), nrow(H2), nrow(H3)),
        names.arg = c('BMI<18', '18<=BMI<25', 'BMI>=25'),
        ylim = c(0,12)
       )
```



```
barplot(height = c(nrow(H1), nrow(H3), nrow(H2)),
        names.arg = c('BMI<18', 'BMI>=25', '18<=BMI<25'),
        ylim = c(0,12)
       )
```



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：都是 <chr[X]> 这样的形式。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；
starwars1<-add_column(starwars,film_num=NA)
starwars1
```

```
## # A tibble: 87 x 15
##    name        height  mass hair_~1 skin_~2 eye_c~3 birth~4 sex   gender homew~5
```

```
##    <chr>        <int> <dbl> <chr>   <chr>   <chr>     <dbl> <chr> <chr>  <chr>
##  1 Luke Skywa~   172    77 blond   fair    blue       19   male  mascu~ Tatooi~
##  2 C-3PO         167    75 <NA>    gold    yellow    112   none  mascu~ Tatooi~
##  3 R2-D2          96    32 <NA>    white,~ red        33   none  mascu~ Naboo
##  4 Darth Vader   202   136 none    white   yellow     41.9 male  mascu~ Tatooi~
##  5 Leia Organa   150    49 brown   light   brown      19   fema~ femin~ Aldera~
##  6 Owen Lars     178   120 brown,~ light   blue       52   male  mascu~ Tatooi~
##  7 Beru White~   165    75 brown   light   blue       47   fema~ femin~ Tatooi~
##  8 R5-D4          97    32 <NA>    white,~ red        NA   none  mascu~ Tatooi~
##  9 Biggs Dark~   183    84 black   light   brown      24   male  mascu~ Tatooi~
## 10 Obi-Wan Ke~   182    77 auburn~ fair    blue-g~    57   male  mascu~ Stewjon
## # ... with 77 more rows, 5 more variables: species <chr>, films <list>,
## #   vehicles <list>, starships <list>, film_num <lgl>, and abbreviated variable
## #   names 1: hair_color, 2: skin_color, 3: eye_color, 4: birth_year,
## #   5: homeworld
```

0.4.5 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；
The1<-select(Theoph,Subject:Dose)
head(The1)
```

```
##   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
```

```
length(The1)
```

```
## [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；
The2<-Theoph %>% filter(Dose>5,Time>mean(Time))
head(The2)
```

```
##   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；
The3<-Theoph %>% mutate(trend=Time-mean(Time))
head(The3)
```

```
##   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'