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] "ZiHaoFang"
```

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

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

```
getwd(); ## 显示当前工作目录
```

```
## [1] "C:/Users/ZiHaoFang/Documents/GitHub/R-for-bioinformatics/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     1.3.0.9000
## 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
```

```
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))
```

```
knitr::kable(grades2)
```

Name	Occupation	English	ComputerScience	Biology	Bioinformatics
Weihua Chen	Teacher	65	81	84	59
Mm Hu	Student	97	90	57	45
John Doe	Teacher	74	80	55	88
Jane Doe	Student	94	85	73	50
Warren Buffet	Entrepreneur	77	82	58	80
Elon Musk	Entrepreneur	86	87	68	82
Jack Ma	Entrepreneur	82	86	87	42

```
grades2.melted <- grades2 %>%
  gather( course, grade, -Name, -Occupation, na.rm = T )
grades2_M<-grades2.melted %>%
  arrange( Name , -grade )
grades2_M %>%
  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     Biology                 68
## 2 Jack Ma       Bioinformatics          42
## 3 Jane Doe      Bioinformatics          50
## 4 John Doe      Biology                 55
## 5 Mm Hu         Bioinformatics          45
## 6 Warren Buffet Biology                 58
## 7 Weihua Chen   Bioinformatics          59
```

```
grades2_M2<-grades2.melted %>%
  group_by( Occupation ) %>%
  summarise( Avg_Grade = mean(grade) ) %>%
  arrange( -Avg_Grade )
knitr::kable(grades2_M2)
```

Occupation	Avg_Grade
Entrepreneur	76.41667
Student	73.87500
Teacher	73.25000

```
(grades_Occupation<-grades2_M2 %>%
  summarise(Best_Avg_Occuption=first(Occupation)))
```

```
## # A tibble: 1 x 1
##   Best_Avg_Occuption
##   <chr>
## 1 Entrepreneur
```

```
(grades2_M3<-grades2.melted %>%
  group_by(course) %>%
  summarise(avggrade=mean(grade)) %>%
  arrange(-avggrade))
```

```
## # A tibble: 4 x 2
##   course          avggrade
##   <chr>              <dbl>
## 1 ComputerScience     84.4
## 2 English             82.1
## 3 Biology             68.9
## 4 Bioinformatics      63.7
```

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；
```

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；
```

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：

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；
```

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；
```

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；
```

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；
```

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'