talk06 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-`talk06` 作业`.pdf`，并提交到老师指定的平台/钉群。

0.2 Talk06 及 talk06-practices 内容回顾

1. tidyr

2. 3 个生信任务的 R 解决方案
3. forcats

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"
```

0.4 练习与作业 1：tidyr

0.4.1 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. 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(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(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
```

```
## # A tibble: 9 x 3
##   name        course         grade
##   <chr>       <chr>          <dbl>
## 1 Zhi Liu     Microbiology     100
## 2 Zhi Liu     English           50
## 3 Zhi Liu     Chinese           69
## 4 Weihua Chen Microbiology      89
## 5 Weihua Chen English           99
## 6 Weihua Chen Bioinformatics    99
## 7 Kang Ning   Bioinformatics   100
## 8 Kang Ning   Chinese           20
## 9 Kang Ning   Chemistry         76
```

```
grades_wide<-grades%>%
pivot_wider(names_from="course",values_from="grade");
```

```
grades_wide;
```

```
## # A tibble: 3 x 6
##   name        Microbiology English Chinese Bioinformatics Chemistry
##   <chr>              <dbl>   <dbl>   <dbl>          <dbl>     <dbl>
## 1 Zhi Liu              100      50      69             NA        NA
## 2 Weihua Chen           89      99      NA             99        NA
## 3 Kang Ning             NA      NA      20            100        76
```

```
grades_long<-grades_wide%>%
  pivot_longer(-name,names_to="course",values_to="grade");
grades_long
```

```
## # A tibble: 15 x 3
##    name        course         grade
##    <chr>       <chr>          <dbl>
##  1 Zhi Liu     Microbiology     100
##  2 Zhi Liu     English           50
##  3 Zhi Liu     Chinese           69
##  4 Zhi Liu     Bioinformatics    NA
##  5 Zhi Liu     Chemistry         NA
##  6 Weihua Chen Microbiology      89
##  7 Weihua Chen English           99
##  8 Weihua Chen Chinese           NA
##  9 Weihua Chen Bioinformatics    99
## 10 Weihua Chen Chemistry         NA
## 11 Kang Ning   Microbiology      NA
## 12 Kang Ning   English           NA
## 13 Kang Ning   Chinese           20
## 14 Kang Ning   Bioinformatics   100
## 15 Kang Ning   Chemistry         76
```

3. 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_long1<-grades_long[!is.na(grades_long$grade),];
grades_long1
```

```
## # A tibble: 9 x 3
##   name        course         grade
##   <chr>       <chr>          <dbl>
## 1 Zhi Liu     Microbiology     100
## 2 Zhi Liu     English           50
## 3 Zhi Liu     Chinese           69
## 4 Weihua Chen Microbiology      89
## 5 Weihua Chen English           99
## 6 Weihua Chen Bioinformatics    99
## 7 Kang Ning   Chinese           20
## 8 Kang Ning   Bioinformatics   100
## 9 Kang Ning   Chemistry         76
```

4. 以下代码有什么作用？

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

答：直接输出 grades 会去掉 NA 的行，这个代码可以显示完整版的 grades, 含有带 NA 的行。

0.5 练习与作业 2：作图

0.5.1 用下面的数据作图

1. 利用下面代码读取一个样本的宏基因组相对丰度数据

```
abu <-
  read_delim(
```

```
  file = "../data/talk06/relative_abundance_for_RUN_ERR1072629_taxonlevel_species.txt
  delim = "\t", quote = "", comment = "#");
```

2. 取前 5 个丰度最高的菌，将其它的相对丰度相加并归为一类 Qita;

3. 用得到的数据画如下的空心 pie chart:

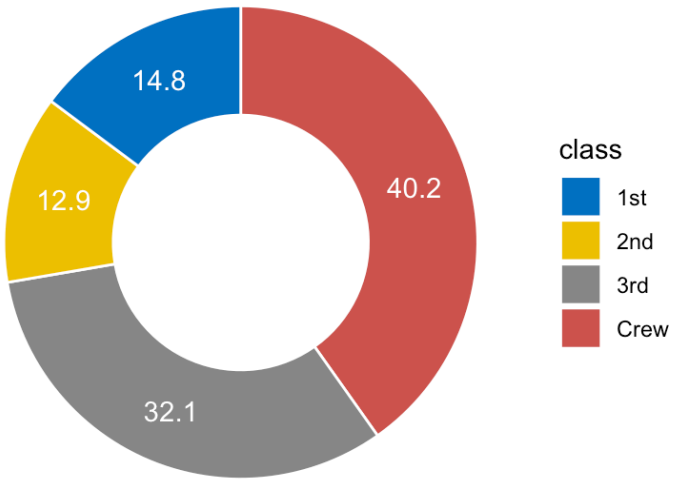


图 1: make a pie chart like this using the meteagenomics data

```
## 代码写这里，并运行;
abu <-
      read_delim(
        file = "../data/talk06/relative_abundance_for_RUN_ERR1072629_taxonlevel_species
        delim = "\t", quote = "", comment = "#");
```

```
## Rows: 122 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
```

```
## chr (1): scientific_name
## dbl (2): ncbi_taxon_id, relative_abundance
##
## 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.
```

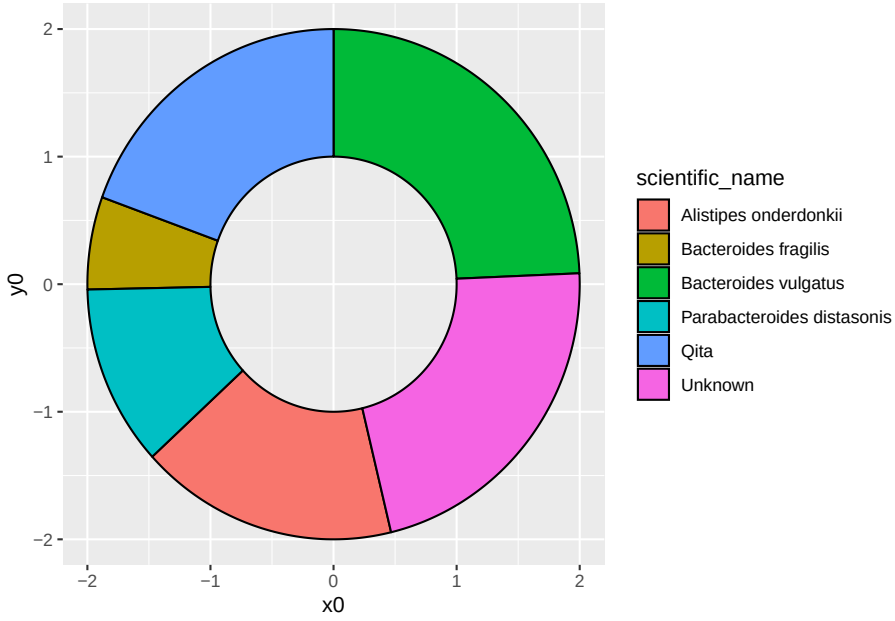
```
library(tidytidbits);
abu.dat<-
abu%>%arrange(desc(relative_abundance))%>%
lump_rows(scientific_name,relative_abundance,n=5,other_level="Qita");
head(abu.dat,n=6);
```

```
## # A tibble: 6 x 3
##   ncbi_taxon_id relative_abundance scientific_name
##           <dbl>              <dbl> <chr>
## 1           821              24.3  Bacteroides vulgatus
## 2            -1              21.9  Unknown
## 3        328813              16.9  Alistipes onderdonkii
## 4           823              11.5  Parabacteroides distasonis
## 5           817               5.87 Bacteroides fragilis
## 6      31848070              19.5  Qita
```

```
library(ggforce)
library(ggplot2)
ggplot()+

geom_arc_bar(data=abu.dat,
stat = "pie",
aes(x0=0,y0=0,r0=1,r=2,

amount=relative_abundance,fill=scientific_name))
```

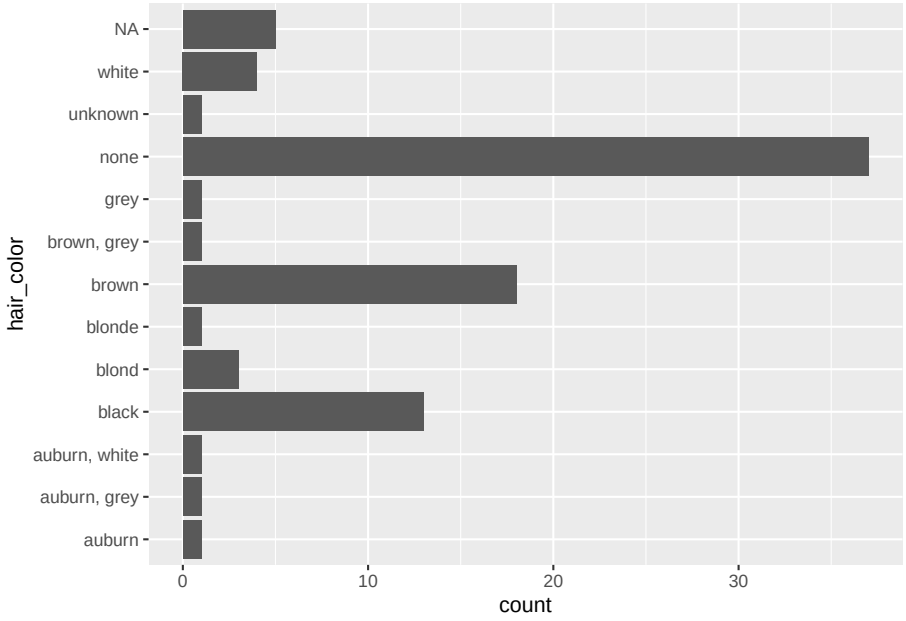


0.5.2 使用 starwars 变量做图

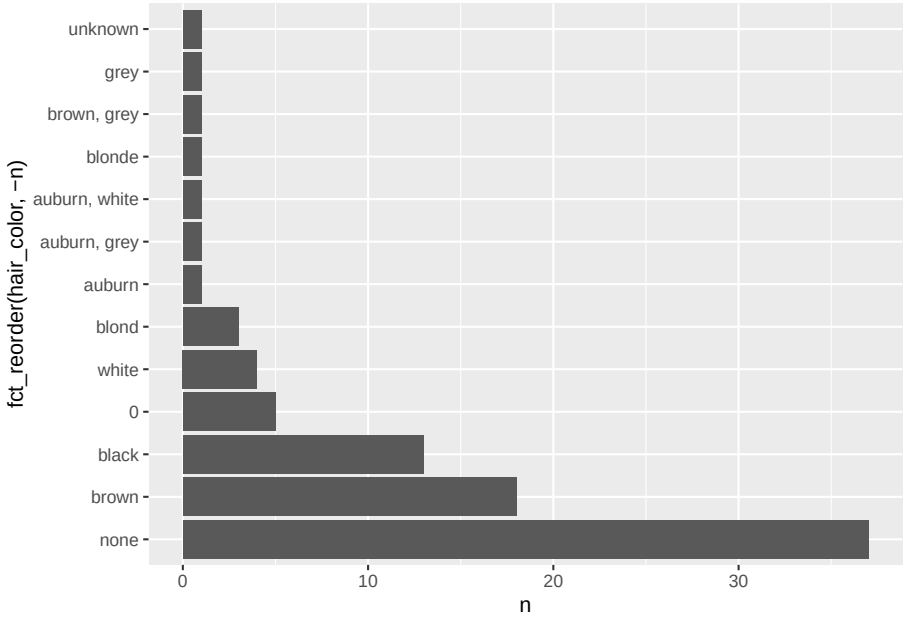
1. 统计 starwars 中 hair_color 的种类与人数时，可用下面的代码：

但是，怎么做到**按数量从小到大**排序？

```
library(dplyr)
library(ggplot2)
library(forcats)
ggplot(starwars, aes(x = hair_color)) +
  geom_bar() +
  coord_flip()
```




```
## 代码写这里，并运行；
starwars_part <- starwars %>% count(hair_color) %>% arrange(n)
starwars_part<- mutate_all(starwars_part, ~replace(., is.na(.), 0))
 ggplot(starwars_part, aes(x = fct_reorder(hair_color,-n), y = n)) +
   geom_bar( stat =  "identity") +
  coord_flip()
```



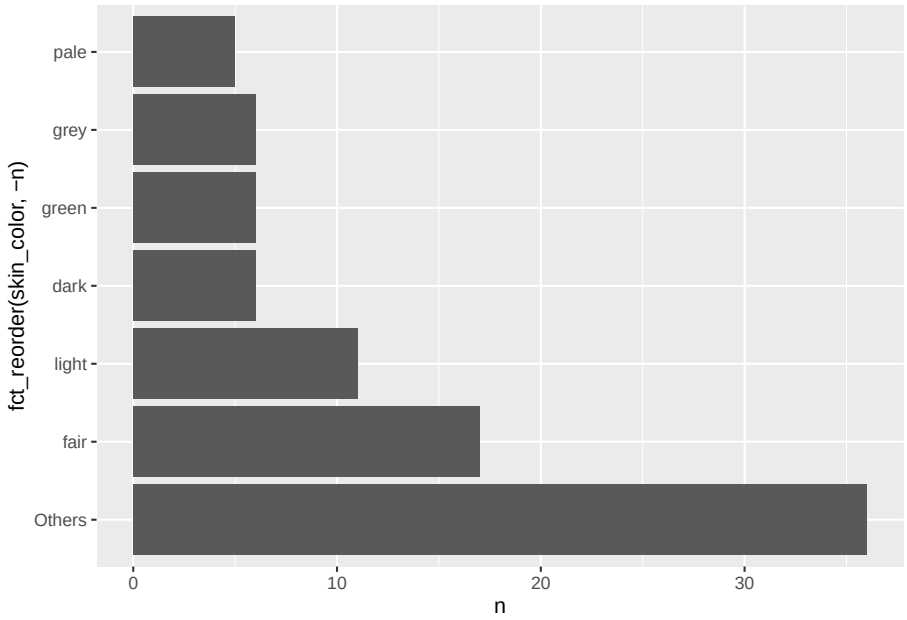
2. 统计 skin_color 时，将出现频率小于 0.05（即 5%）的颜色归为一类 Others，按出现次数排序后，做与上面类似的 barplot；

```
## 代码写这里，并运行；
starwars_part1<-
starwars%>%count(skin_color) %>% arrange(n)%>%
lump_rows(skin_color,n);
a<-sum(starwars_part1$n)
starwars_part2<-starwars_part1%>%filter(n/a<=0.05);
starwars_part3<-starwars_part1%>%filter( n >= a*0.05)
starwars_part4<-starwars_part3%>%add_row(skin_color = "Others", n = sum(starwars_part2$
starwars_part4
```

```
## # A tibble: 7 x 2
##   skin_color     n
##   <chr>      <int>
## 1 pale           5
## 2 dark           6
## 3 green          6
```

```
## 4 grey            6
## 5 light          11
## 6 fair           17
## 7 Others         36
```

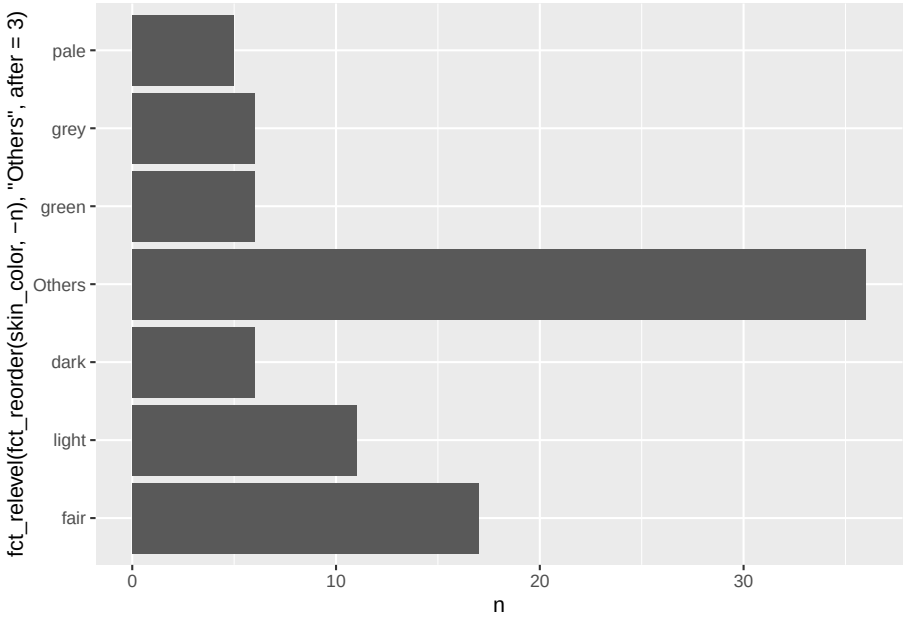
```
ggplot(starwars_part4, aes(x = fct_reorder(skin_color,-n), y = n)) +
   geom_bar( stat =  "identity") +
  coord_flip()
```



3. 使用 2 的统计结果，但画图时，调整 bar 的顺序，使得 `Others` 处于第 4 的位置上。提示，可使用 `fct_relevel` 函数；

```
## 代码写这里，并运行；

ggplot(starwars_part4,aes(x = fct_relevel(fct_reorder(skin_color,-n),"Others", after =
   geom_bar( stat =  "identity" ) +
  coord_flip()
```



0.6 练习与作业 3：数据分析

0.6.1 使用 STRING PPI 数据分析并作图

1. 使用以下代码，装入 PPI 数据；

```
ppi <- read_delim( file = "../data/talk06/ppi900.txt.gz", col_names = T,
                   delim =  "\t", quote = "" );
```

2. **随机挑选**一个基因，得到类似于本章第一部分的互作网络图；

```
## 代码写这里，并运行；
```

0.6.2 对宏基因组相对丰度数据进行分析

1.data/talk06 目录下有 6 个文本文件，每个包含了一个宏基因组样本的分析结果：

```
relative_abundance_for_curated_sample_PRJEB6070-DE-073_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-074_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-075_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-076_at_taxonlevel_species.txt
relative_abundance_for_curated_sample_PRJEB6070-DE-077_at_taxonlevel_species.txt
```

2. 分别读取以上文件，提取 scientific_name 和 relative_abundance 两列；
3. 添加一列为样本名，比如 PRJEB6070-DE-073, PRJEB6070-DE-074 … ；
4. 以 scientific_name 为 key，将其内容合并为一个 data.frame 或 tibble，其中每行为一个样本，每列为样本的物种相对丰度。注意：用 join 或者 spread 都可以，只要能解决问题。
5. 将 NA 值改为 0。

```
## 代码写这里，并运行；
```