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 内容回顾

1. 3 个生信任务的 R 解决方案
2. factors 的更多应用 (forcats)
3. pipe

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] "lucas"
```

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

```
## [1] "/Users/lucas"
```

0.4 练习与作业 1：作图

0.4.1 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:

```
## 代码写这里，并运行；
# Load the library
library(readr)
```

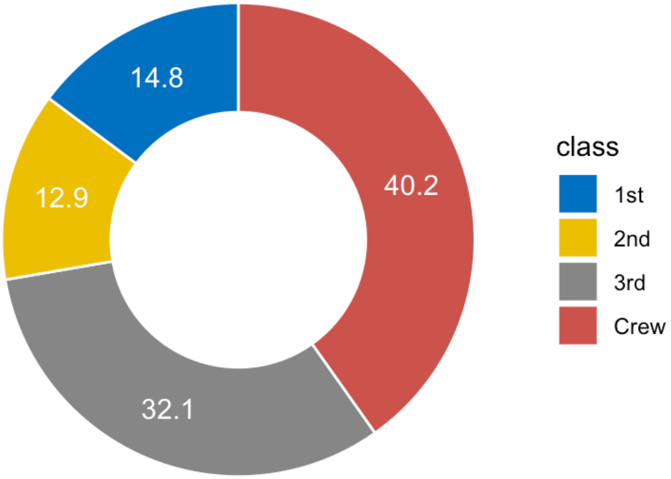


图 1: make a pie chart like this using the meteagenomics data

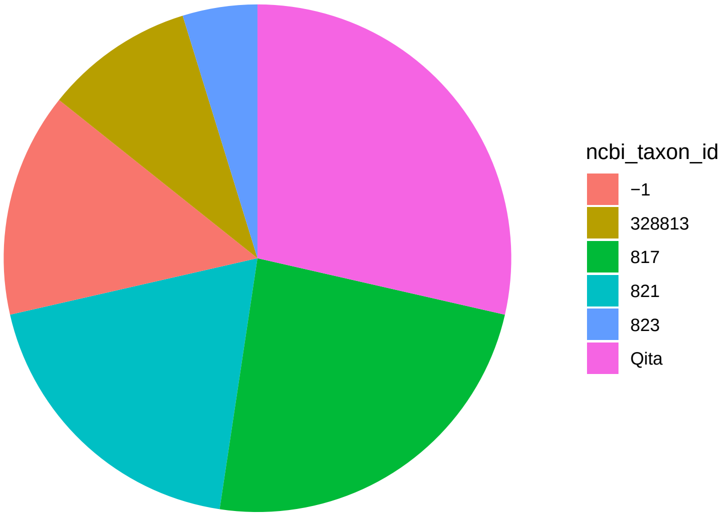
```r
library(ggplot2)
library(gridExtra)

# Read the data
abu =
  read_delim(
    file =
      "../data/talk06/relative_abundance_for_RUN_ERR1072629_taxonlevel_species.txt",
    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.
```

```r
# Re-order the data
abu_reorder =
  abu[order(-abu$relative_abundance), ]

# Find out top 5
abu_top5 =
  abu_reorder[1:5, ]

# Add the rest of the data
abu_qita =
  sum(abu[6:nrow(abu_reorder), ]$relative_abundance)

# Combine them
abu_full =
  rbind(abu_top5,c("Qita", abu_qita))
```

```
## Warning in rbind(deparse.level, ...): number of columns of result, 3, is not a
## multiple of vector length 2 of arg 2
```

```
# Draw the plot
abu_pie_chart =
  ggplot(data = abu_full,
         aes(x = "",
             y = relative_abundance,
             fill = ncbi_taxon_id)) +
  geom_bar(stat = "Identity", width = 1) +
  coord_polar(theta = "y") +
  theme_void() +
  theme(legend.position = "right")

# Print the pie chart
print(abu_pie_chart)
```



0.4.2 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：

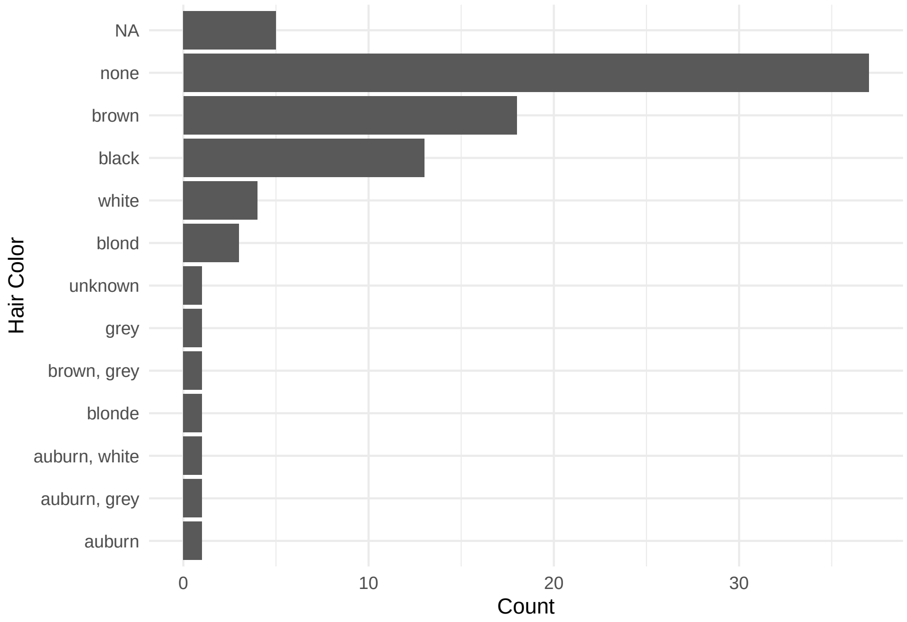
但是，怎么做到**按数量从小到大**排序?

```
# Load the library
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

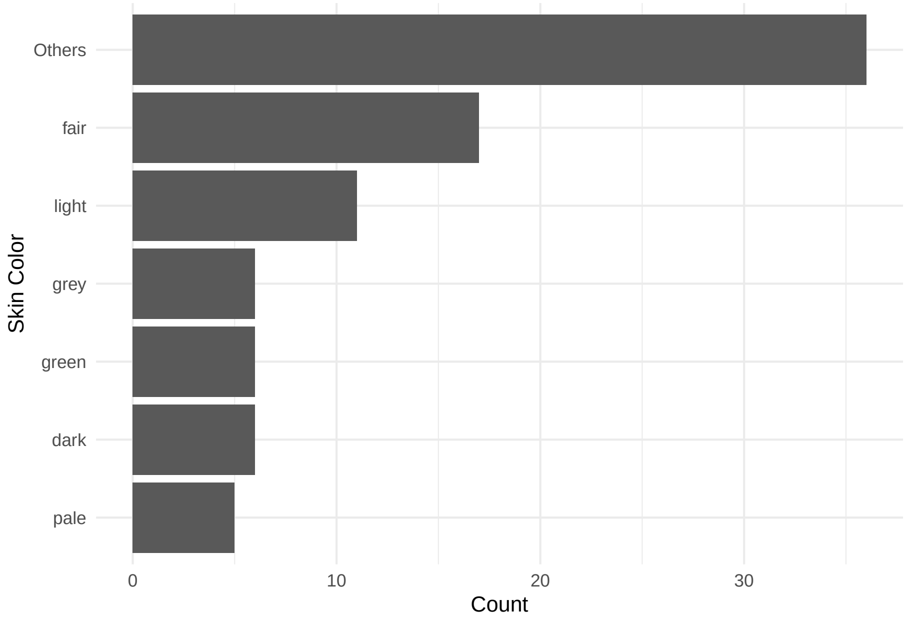
```
# Count the number o
# and sort them in descending order.
starwars_counts =
  starwars %>%
  group_by(hair_color) %>%
  summarise(count = n()) %>%
  arrange(count)
```

```
# Draw the plot
starwars_plot01=
  ggplot(
    starwars_counts,
    aes(
      x = reorder(
        hair_color,
        count),
      y = count)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(
    x = "Hair Color",
    y = "Count") +
  theme_minimal()

# Print the plot
print(starwars_plot01)
```



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；

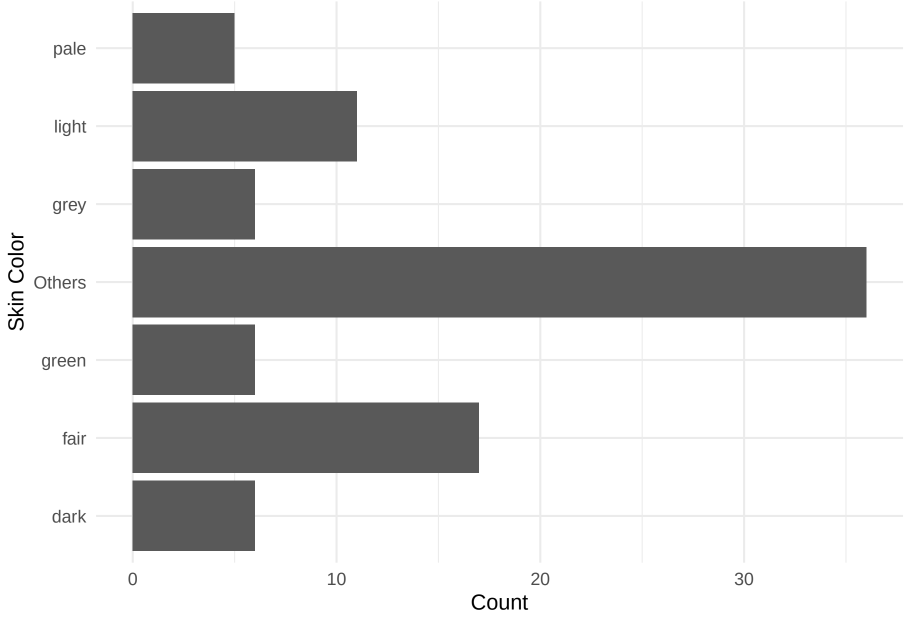
```
## 代码写这里，并运行；

# Count the number of
# skin_color types and people
starwars_counts_skin =
  starwars %>%
  group_by(skin_color) %>%
  summarise(count = n())

# Group the category 'Others'
total_count_group =
  sum(
    starwars_counts_skin$count)
threshold =
  0.05 * total_count_group

starwars_counts_skin_group =
  starwars_counts_skin %>%
  mutate(
    skin_color =
      ifelse(
        count < threshold,
        "Others",
        as.character(skin_color)))

# Recount the type and number.
starwars_counts_skin_group2 =
  starwars_counts_skin_group %>%
  group_by(skin_color) %>%
  summarise(count = sum(count)) %>%
  arrange(count)

# Draw the plot
starwars_plot02 =
  ggplot(
    starwars_counts_skin_group2,
    aes(
      x = reorder(
        skin_color,
        count),
      y = count)) +
  geom_bar(
    stat = "identity") +
  coord_flip() +
  labs(
    x = "Skin Color",
    y = "Count") +
  theme_minimal()

# Print the plot
print(starwars_plot02)
```



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；
# Load the library
library(forcats)

# Count the number
starwars_counts_skin2 =
  starwars %>%
  group_by(skin_color) %>%
  summarise(count = n())

# Group the category 'Others'
total_count_group2 =
  sum(starwars_counts_skin2$count)
threshold =
  0.05 * total_count_group2
```

```r
starwars_counts_skin2_group =
  starwars_counts_skin2 %>%
  mutate(
    skin_color =
      ifelse(
        count < threshold,
        "Others",
        as.character(skin_color)))

# Recount the type and number.
starwars_counts_skin2_group2 =
  starwars_counts_skin2_group %>%
  group_by(skin_color) %>%
  summarise(count = sum(count))

# Adjusting the position of 'Others'
starwars_counts_skin2_group2$skin_color =
  fct_relevel(
  starwars_counts_skin2_group2$skin_color,
  "Others",
  after = 3  # to 4th
)

# Draw the plot
starwars_plot03 =
  ggplot(
    starwars_counts_skin2_group2,
    aes(
      x = skin_color,
      y = count)) +
  geom_bar(
    stat = "identity") +
  labs(
    x = "Skin Color",
    y = "Count") +
  coord_flip() +
  theme_minimal()

# Print the plot
print(starwars_plot03)
```



0.5 练习与作业 2：数据分析

0.5.1 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;
```

0.5.2 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;
```