talk04 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-talk04 作业.pdf`，并提交到老师指定的平台/钉群。

0.2 Talk04 内容回顾

待写 …

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：R session 管理

0.4.1 完成以下操作

- 定义一些变量（比如 x, y , z 并赋值；内容随意）
- 从外部文件装入一些数据（可自行创建一个 4 行 5 列的数据，内容随意）
- 保存 workspace 到.RData
- 列出当前工作空间内的所有变量
- 删除当前工作空间内所有变量
- 从.RData 文件恢复保存的数据
- 再次列出当前工作空间内的所有变量，以确认变量已恢复
- 随机删除两个变量
- 再次列出当前工作空间内的所有变量

```
## 代码写这里，并运行；
# Creating some variables
x = 1
y = 2
table = read.table(file = "data/Table0.txt",
           col.names = c("Name", "Age", "Height", "Weight", "Gender"))
```

```r
# Store these variables
ls()
```

```
## [1] "encoding"  "inputFile" "pSubTitle" "table"     "x"         "y"
```

```r
# Save .RData
save.image(file = ".talk03.RData")

# Remove the variables
rm(list = ls())

# List the variables
ls()
```

```
## character(0)
```

```r
# Restore data from file
load(file = ".talk03.RData")

# List the variables
ls()
```

```
## character(0)
```

```r
# Remove two variables
rm(list = c("x", "y"))
```

```
## Warning in rm(list = c("x", "y")): object 'x' not found

## Warning in rm(list = c("x", "y")): object 'y' not found
```

0.5 练习与作业 2：Factor 基础

0.5.1 factors 增加

- 创建一个变量:

```
x <- c("single", "married", "married", "single");
```

- 为其增加两个 levels，single, married;
- 以下操作能成功吗?

```
x[3] <- "widowed";
```

- 如果不，请提供解决方案;

```
## 代码写这里，并运行;
# Creating the variable
x = c("single", "married", "married", "single")

# Adding two levels
x = as.factor(x)
x_level = levels(x)
str(x)
```

```
##  Factor w/ 2 levels "married","single": 2 1 1 2
```

```
# Print the result
print(x_level)
```

```
## [1] "married" "single"
```

```
# Add "widowed" to the variable
levels(x) = c(levels(x), "widowed")
x[length(x) + 1] = "widowed"
str(x)
```

```
##  Factor w/ 3 levels "married","single",..: 2 1 1 2 3
```

```
# Print the result
print(x)
```

```
## [1] single  married married single  widowed
## Levels: married single widowed
```

0.5.2 factors 改变

- 创建一个变量:

```
v = c("a", "b", "a", "c", "b")
```

- 将其转化为 factor，查看变量内容
- 将其第一个 levels 的值改为任意字符，再次查看变量内容

```
## 代码写这里，并运行;
# Creating the variable
v = c("a", "b", "a", "c", "b")

# Convert the variable to factor
if(!is.factor(v))
  v = as.factor(v)

# Print the result
is.factor(v)
```

```
## [1] TRUE
```

```
print(v)
```

```
## [1] a b a c b
## Levels: a b c
```

```r
# Change the factor
v[1] = "c"

# Print the result
print(v)
```

```
## [1] c b a c b
## Levels: a b c
```

- 比较改变前后的 v 的内容，改变 levels 的操作使 v 发生了什么变化?

答:

When changing the levels of a factor, the categories or labels associated with that factor are essentially being modified. In this example, I modify the lable of the first level from “a” to “c”.

0.5.3 factors 合并

- 创建两个由随机大写字母组成的 factors
- 合并两个变量，使其 factors 得以在合并后保留

```r
## 代码写这里，并运行;
# Set a random seed
set.seed(sample(1:1000, 1))

# Create two random factors
factor1 = factor(sample(LETTERS, 5, replace = TRUE))
factor2 = factor(sample(LETTERS, 5, replace = TRUE))

# Print the two factors
print(factor1)
```

```
## [1] G Q X Y A
## Levels: A G Q X Y
```

```
print(factor2)
```

```
## [1] Q V P Z B
## Levels: B P Q V Z
```

```
# Merge the two factors into a new factor
merged_factor = factor(paste(factor1, factor2, sep = "_"))

# Print the result
print(merged_factor)
```

```
## [1] G_Q Q_V X_P Y_Z A_B
## Levels: A_B G_Q Q_V X_P Y_Z
```

0.5.4 利用 factor 排序

以下变量包含了几个月份，请使用 factor，使其能按月份，而不是英文字符串排序:

```
mon <- c("Mar","Nov","Mar","Aug","Sep","Jun","Nov","Nov","Oct","Jun","May","Sep","Dec",
```

```
## 代码写这里，并运行;
# Create the required factor and a factor containing months, sorted in month order
mon = c("Mar", "Nov", "Mar", "Aug", "Sep", "Jun", "Nov", "Nov", "Oct", "Jun", "May", "S
months = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov",

# Use the factor function to convert mon into a factor and specify the levels parameter
mon_factor = factor(mon, levels = months)
```

```
# Print the result
print(mon_factor)
```

```
##  [1] Mar Nov Mar Aug Sep Jun Nov Nov Oct Jun May Sep Dec Jul Nov
## Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

0.5.5 forcats 的问题

forcats 包中的 fct_inorder, fct_infreq 和 fct_inseq 函数的作用是什么?

请使用 forcats 包中的 gss_cat 数据举例说明

答:

forcats 包是用于操作因子变量的包，它提供了许多有用的函数来处理和修改因子。其中包括 fct_inorder()、fct_infreq() 和 fct_inseq() 函数，分别用于以下用途:

1. fct_inorder(): 此函数用于按照因子的当前顺序对因子水平进行排序。它不会改变因子水平的值，只会按照它们出现的顺序重新排列因子水平。这对于将因子水平按照它们在数据中的出现顺序进行排序非常有用。
2. fct_infreq(): 此函数用于按照因子水平的频率（出现次数）对因子水平进行排序。它会将最常出现的水平排在前面，依次排列其他水平。这对于查看或可视化因子水平的频率分布非常有用。
3. fct_inseq(): 此函数用于按照一组自定义的水平顺序对因子水平进行排序。您可以提供一个水平的向量，指定所需的排序顺序。这对于按照特定顺序对因子水平进行排序非常有用。

下面是使用 forcats 包中的 gss_cat 数据说明如何使用这些函数的例子:

```
## 代码写这里，并运行;
library(forcats)

# Using the gss_cat dataset
data("gss_cat")

# View the first few rows of the gss_cat dataset
head(gss_cat)
```

```
##   year       marital age  race        rincome             partyid
## 1 2000 Never married  26 White  $8000 to 9999       Ind,near rep
## 2 2000      Divorced  48 White  $8000 to 9999 Not str republican
## 3 2000       Widowed  67 White Not applicable        Independent
## 4 2000 Never married  39 White Not applicable       Ind,near rep
## 5 2000      Divorced  25 White Not applicable   Not str democrat
## 6 2000       Married  25 White $20000 - 24999    Strong democrat
##                relig            denom tvhours
## 1         Protestant Southern baptist      12
## 2         Protestant Baptist-dk which      NA
## 3         Protestant  No denomination       2
## 4 Orthodox-christian   Not applicable       4
## 5               None   Not applicable       1
## 6         Protestant Southern baptist      NA
```

```
# Sorting 'relig' variables in the gss_cat dataset with fct_inorder()
gss_cat$relig = fct_inorder(gss_cat$relig)
head(gss_cat$relig)
```

```
## [1] Protestant         Protestant         Protestant         Orthodox-christian
## [5] None               Protestant
## 16 Levels: Protestant Orthodox-christian None Christian Jewish ... Not applicable
```

```r
# Sorting 'relig' variables in the gss_cat dataset using fct_infreq()
gss_cat$relig = fct_infreq(gss_cat$relig)
head(gss_cat$relig)
```

```
## [1] Protestant         Protestant         Protestant         Orthodox-christian
## [5] None               Protestant
## 16 Levels: Protestant Catholic None Christian Jewish Other ... Not applicable
```

```r
sum(is.na(gss_cat$tvhours))
```

```
## [1] 10146
```

```r
# Define custom sort order
custom_order = c("12", "16", "20", "8", "9", "5", "6", "7", "10", "11", "4", "3", "2",
gss_cat$tvhours = as.factor(gss_cat$tvhours)
gss_cat$tvhours = fct_inseq(gss_cat$tvhours, ordered = TRUE)

head(gss_cat$tvhours)
```

```
## [1] 12   <NA> 2    4    1    <NA>
## 24 Levels: 0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < 11 < 12 < 13 < ... < 24
```

0.6 练习与作业 3：用 mouse genes 数据做图

0.6.1 画图

1. 用 readr 包中的函数读取 mouse genes 文件（从本课程的 Github 页面下载 data/talk04/ ）
2. 选取常染色体（1-19）和性染色体（X，Y）的基因
3. 画以下两个基因长度 boxplot：

- 按染色体序号排列，比如 1, 2, 3 …. X, Y
- 按基因长度中值排列，从短 -> 长 …
- 作图结果要求：
  - 要清晰显示 boxplot 的主体；
  - 严格按照中值进行排序；注：‘ylim()‘限制时会去除一些值，造成中值错位。可考虑使用其它函数或调整参数。

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

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

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

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

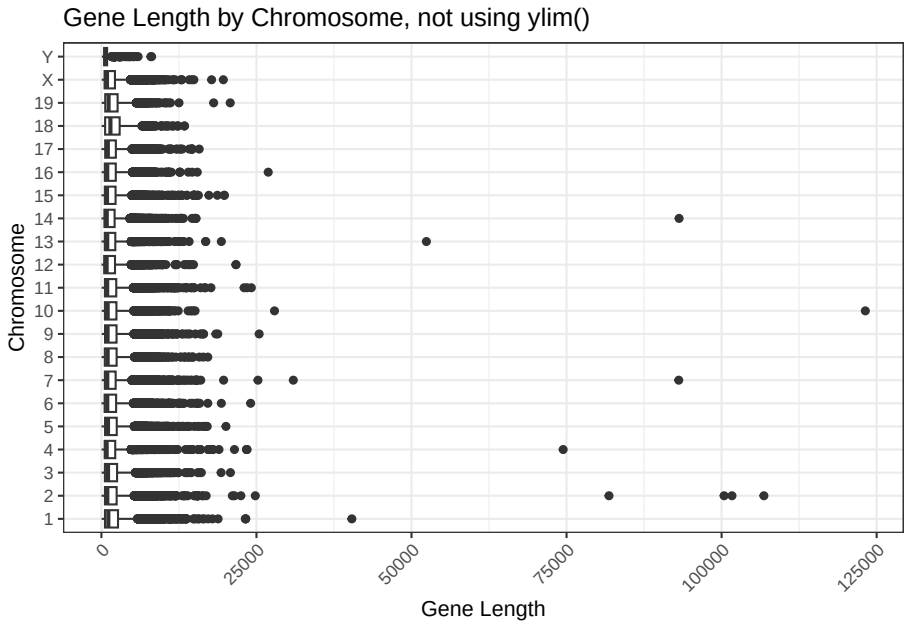
```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

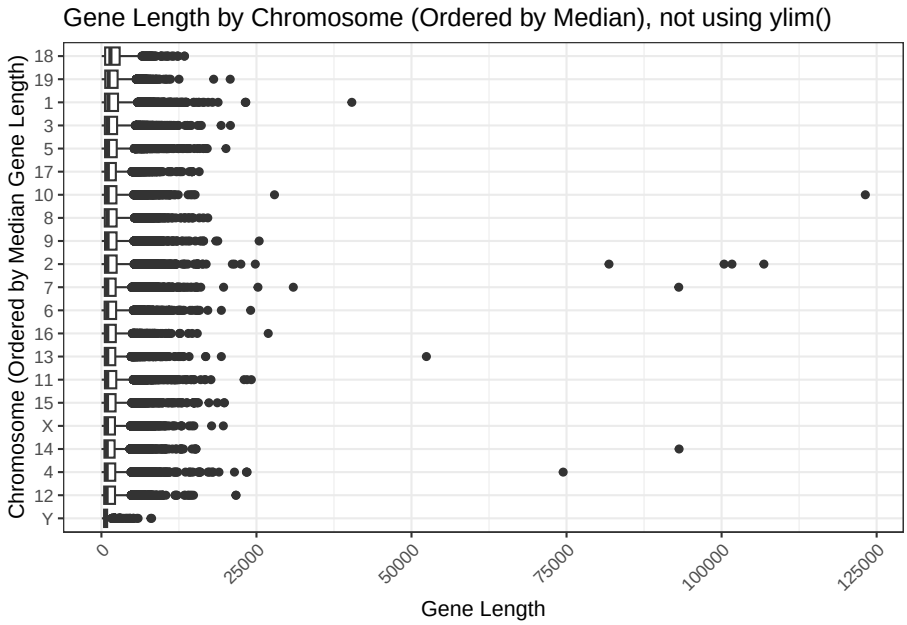
```r
# 1. Read the mouse genes file with functions from the readr package
file_path = "../data/talk04/mouse_genes_biomart_sep2018.txt"
mouse_genes = read_delim(file_path,
                         delim = "\t",
                         show_col_types = FALSE)
mouse_genes = tibble(mouse_genes)


# 2. Selection of genes for autosomes (1-19) and sex chromosomes (X, Y)
selected_chromosomes = c(1:19, "X", "Y")
filtered_genes = mouse_genes %>%
  filter(`Chromosome/scaffold name` %in% selected_chromosomes)

# 3. Draw a boxplot of the lengths of the two genes

# 3.1. Arrangement by chromosome number

filtered_genes$`Chromosome/scaffold name` = factor(filtered_genes$`Chromosome/scaffold

# Not using ylim()
p1 = ggplot(filtered_genes,
            aes(x = factor(`Chromosome/scaffold name`),
                y = `Transcript length (including UTRs and CDS)`)) +
  geom_boxplot() +
  coord_flip() +
  labs(
    x = "Chromosome",
    y = "Gene Length",
    title = "Gene Length by Chromosome, not using ylim()"
  ) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```r
# Using ylim()
p2 = ggplot(filtered_genes,
            aes(x = factor(`Chromosome/scaffold name`),
                y = `Transcript length (including UTRs and CDS)`)) +
  geom_boxplot() +
  ylim(0, 5000) +
  coord_flip() +
  labs(
    x = "Chromosome",
    y = "Gene Length",
    title = "Gene Length by Chromosome, using ylim()"
  ) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# 3.2. Ordered by median gene length
# Not Using ylim()
p3 = ggplot(filtered_genes,
            aes(x = reorder(`Chromosome/scaffold name`, `Transcript length (including U
                y = `Transcript length (including UTRs and CDS)`)) +
  geom_boxplot() +
  coord_flip() +
  labs(
    x = "Chromosome (Ordered by Median Gene Length)",
    y = "Gene Length",
    title = "Gene Length by Chromosome (Ordered by Median), not using ylim()"
  ) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Using ylim()
p4 = ggplot( data = filtered_genes,
             aes( x = reorder( `Chromosome/scaffold name`, `Transcript length (includin
```

```
                    y = `Transcript length (including UTRs and CDS)` ) ) +
  geom_boxplot() +
  coord_flip() +
  ylim(0, 5000) +
  theme_bw() +
  labs(
    x = "Chromosome (Ordered by Median Gene Length)",
    y = "Gene Length",
    title = "Gene Length by Chromosome (Ordered by Median), using ylim()"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


# Drawing two shapes
pic1 = grid.arrange(p1)
```

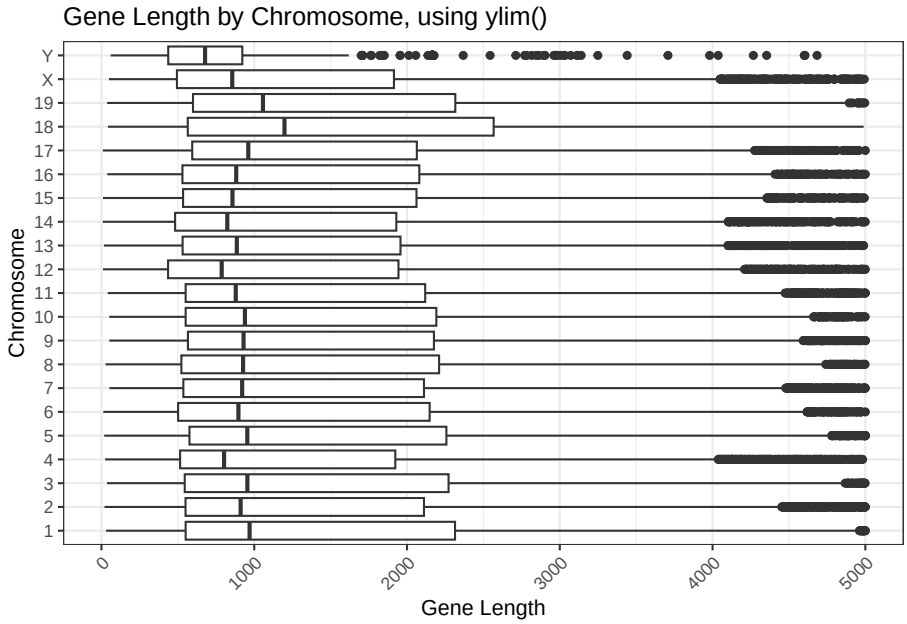


```
pic3 = grid.arrange(p3)
```



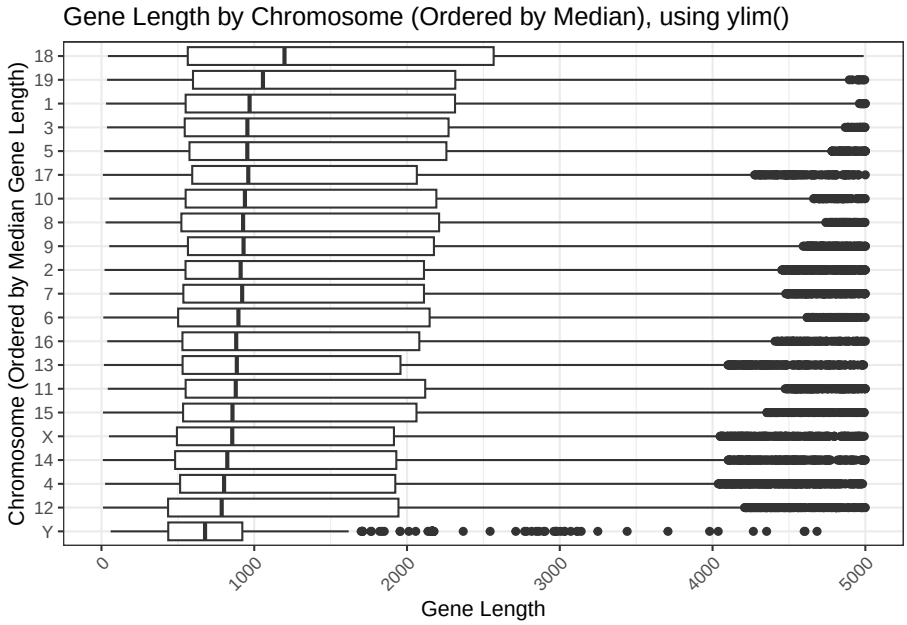

```
pic2 = grid.arrange(p2)
```

```
## Warning: Removed 6639 rows containing non-finite values (`stat_boxplot()`).
```




```
pic4 = grid.arrange(p4)
```

```
## Warning: Removed 6639 rows containing non-finite values (`stat_boxplot()`).
```



All the codes above partly based on ChatGPT 3.5 to fix some bugs.

Refer to the DigitalOcean forum for the usage of some of the functions.