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] "sicheng.wu"
```

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

```
## [1] "/home/vkorpela"
```

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;
x <- 114; y <- "KRKS"; z <- 5.14;
table1 <- read.table("./data/Table1.txt", header = TRUE)

save.image(file = "./talk04-homework-saved.RData")

ls()
```

```
## [1] "encoding"  "inputFile" "pSubTitle" "table1"    "x"         "y"
## [7] "z"
```

```
rm(list = ls())

load(file = "./talk04-homework-saved.RData")
ls()
```

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

```
rm(x); rm(z);
```

```
## Warning in rm(x): object 'x' not found

## Warning in rm(z): object 'z' not found
```

```
ls()
```

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

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

x <- as.factor(x)
levels(x)
```

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

```
# 以下操作不能成功
x[3] <- "widowed"
```

```
## Warning in `[<-.factor`(`*tmp*`, 3, value = "widowed"): invalid factor level, NA
## generated
```

```
x
```

```
## [1] single  married <NA>    single
## Levels: married single
```

```
# 解决方案
levels(x) <- c(levels(x), "widowed")
x[3] <- "widowed"
x
```

```
## [1] single  married widowed single
## 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;
v <- c("a", "b", "a", "c", "b")
(v <- as.factor(v))
```

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

```
levels(v)[1] <- "k"
v
```

```
## [1] k b k c b
## Levels: k 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：改变 levels 的操作使得 v 的内容与 levels 同步发生了变化，但内容对应关系保持不变。

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

```
## 代码写这里，并运行;
(fact1 <- as.factor(sample(LETTERS, 10, replace = TRUE)))
```

```
##  [1] I A J D E P M Y O N
## Levels: A D E I J M N O P Y
```

```
(fact2 <- as.factor(sample(LETTERS, 12, replace = TRUE)))
```

```
##  [1] O K B O G T C Z Y D G P
## Levels: B C D G K O P T Y Z
```

```
(fact3 <- c(fact1, fact2))
```

```
##  [1] I A J D E P M Y O N O K B O G T C Z Y D G P
## Levels: A D E I J M N O P Y B C G K T 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；
mon <- c("Mar", "Nov", "Mar", "Aug", "Sep", "Jun", "Nov", "Nov", "Oct", "Jun", "May", "

# 包含的月份数
length(unique(mon))
```

```
## [1] 9
```

```
# 按月份排序
mon_levels <- c(
  "Jan", "Feb", "Mar", "Apr", "May", "Jun",
  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"
  )
mon_fact <- factor(mon, levels = mon_levels)
sort(mon_fact)
```

```
##  [1] Mar Mar May Jun Jun Jul Aug Sep Sep Oct Nov Nov Nov Nov Dec
## 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

```
## 代码写这里，并运行;
library(forcats)

# fct_inorder 根据首次出现的顺序重新排列 levels
fct1 <- fct_inorder(gss_cat[["marital"]])
levels(fct1)
```

```
## [1] "Never married" "Divorced"      "Widowed"       "Married"
## [5] "Separated"     "No answer"
```

```
# fct_infreq 根据出现频次从高到低重新排列 levels
fct2 <- fct_infreq(gss_cat[["marital"]])
levels(fct2)
```

```
## [1] "Married"       "Never married" "Divorced"      "Widowed"
## [5] "Separated"     "No answer"
```

```
# fct_inseq 根据值从小到大重新排列 levels
# gss_cat 中没有包含可以强制转换为数值的 factor，故使用 years 列
fct3 <- fct_inseq(as.factor(gss_cat[["year"]]))
levels(fct3)
```

```
## [1] "2000" "2002" "2004" "2006" "2008" "2010" "2012" "2014"
```

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 …

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

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1
```

```
## -- 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   1.2.0      v stringr 1.4.1
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
gene <- read_tsv("../data/talk04/mouse_genes_biomart_sep2018.txt")
```

```
## Rows: 138532 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
```

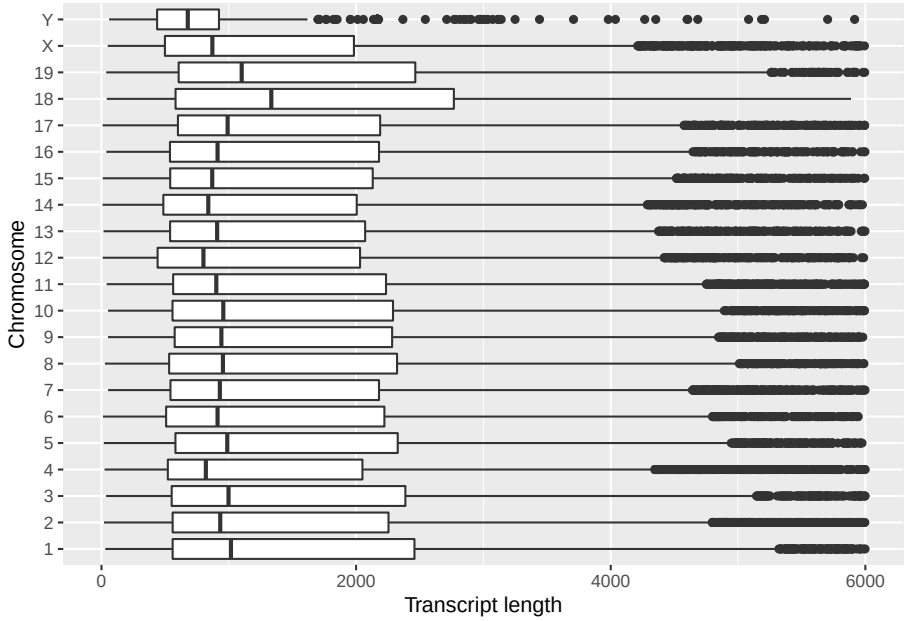
```
## chr (5): Gene stable ID, Transcript stable ID, Protein stable ID, Transcript...
## dbl (1): Transcript length (including UTRs and CDS)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
chromo <- c(1:19, "X", "Y")

gene.filtered <- gene %>% filter(`Chromosome/scaffold name` %in% chromo)
gene.filtered$`Chromosome/scaffold name` <-
  factor(gene.filtered$`Chromosome/scaffold name`,
         levels = chromo)

plot1 <-
  ggplot(
    data = gene.filtered,
    aes(
      x = `Chromosome/scaffold name`,
      y = `Transcript length (including UTRs and CDS)`
    )
  ) +
  geom_boxplot() +
  coord_flip() +
  xlab("Chromosome") +
  ylab("Transcript length") +
  ylim(0, 6000)
plot1
```

```
## Warning: Removed 3926 rows containing non-finite values (stat_boxplot).
```




```
# ylim 会删掉超范围的数据，可能会导致排序的结果和图上画出来的中值不一致
# 所以为了数据符合直观，只有设置成这样了 :(
plot2 <-
  ggplot(
    data = gene.filtered,
    aes(
      x = fct_reorder(`Chromosome/scaffold name`,
                      `Transcript length (including UTRs and CDS)`,
                      median),
      y = `Transcript length (including UTRs and CDS)`
    )
  ) +
  geom_boxplot() +
  coord_flip() +
  xlab("Chromosome") +
  ylab("Transcript length") +
  ylim(0, 6000)
plot2
```

```
## Warning: Removed 3926 rows containing non-finite values (stat_boxplot).
```