homework1

黄舟翔 3220103606

2025-06-26

Q1

a.

运行下列代码后，可以将 Iowa.csv 中的数据提取到 iowa.df 中。

```
iowa.df<-read.csv("data/Iowa.csv", sep = ';', header=T)
```

b.

运行下列代码后，可以查看 iowa.df 的行数和列数，由运行结果可知，iowa.df 有 33 行和 10 列。

```
dimensions <- dim(iowa.df)
cat("b. 数据框有", dimensions[1], " 行和", dimensions[2], " 列\n")
```

```
## b. 数据框有 33 行和 10 列
```

c.

可以通过下列代码获取 iowa.df 的列名。由运行结果可知，列名称为 Year, Rain0, Temp1, Rain1, Temp2, Rain2, Temp3, Rain3, Temp4, Yield

```
column_names <- names(iowa.df)
cat("c. 列名称为:", paste(column_names, collapse = ", "), "\n")
```

```
## c. 列名称为: Year, Rain0, Temp1, Rain1, Temp2, Rain2, Temp3, Rain3, Temp4, Yield
```

d.

用下列代码获取第 5 行第 7 列的值，可得值为 79.7。

```
value_5_7 <- iowa.df[5, 7]
cat("d. 第 5 行第 7 列的值是:", value_5_7, "\n")
```

```
## d. 第5行第7列的值是: 79.7
```

e.

用下列代码获取第 2 行的完整内容。

```
cat("e. 第二行的完整内容:\n")
```

```
## e. 第二行的完整内容:
```

```
print(iowa.df[2, ])
```

```
##   Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield
## 2 1931 14.76  57.5  3.83    75  2.72  77.2   3.3  72.6  32.9
```

Q2

a.

max(vector1)：元素实际是字符类型。R 语言中会按照按字典序比较，即：用 ASCII 值比较："1"(49) < "3"(51) < "5"(53) < "7"(55)

故显示最大值为 "7"

```
vector1 <- c("5", "12", "7", "32")
max(vector1)
```

```
## [1] "7"
```

sort(vector1)：按字符编码升序排列，即：首字符排序：1 < 3 < 5 < 7

结果："12", "32", "5", "7

```r
vector1 <- c("5", "12", "7", "32")
sort(vector1)
```

```
## [1] "12" "32" "5"  "7"
```

sum(vector1)：出现错误，因为字符向量不能直接数学运算。需先转换为数值型：sum(as.numeric(vector1))才能得到 56。

b.

1. 会产生错误。vector2 创建时混合类型触发强制转换，vector2 实际值为 c(“5”, “7”, “12”)。故加法操作”7” + “12” 无意义

2. 数据框不要求各元素类型统一，故保留原始类型：

z1: 字符型 (“5”)

z2: 数值型 (7)

z3: 数值型 (12)

dataframe3[1,2] 和 dataframe3[1,3] 提取数值元素

有效运算：7 + 12 = 19

```r
dataframe3 <- data.frame(z1="5", z2=7, z3=12)
dataframe3[1,2] + dataframe3[1,3]
```

```
## [1] 19
```

3. list4[[2]] → 42 (数值)

list4[[4]] → 126 (数值)

相加结果：168

```r
list4 <- list(z1="6", z2=42, z3="49", z4=126)
list4[[2]] + list4[[4]]
```

```
## [1] 168
```

4. list4[2] → 单元素列表

list4[4] → 单元素列表

列表加法非法

Q3

a.

用下列代码可以实现。

```
seq1 <- seq(from = 1, to = 10000, by = 372)
seq2 <- seq(from = 1, to = 10000, length.out = 50)
```

b.

times = n：

将整个向量重复 n 次

模式：(元素 1, 元素 2, …) → 整体重复

each = n：

将每个元素连续重复 n 次

模式：元素 1×n, 元素 2×n, …

```
rep(1:3, times = 3)
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

```
rep(1:3, each = 3)
```

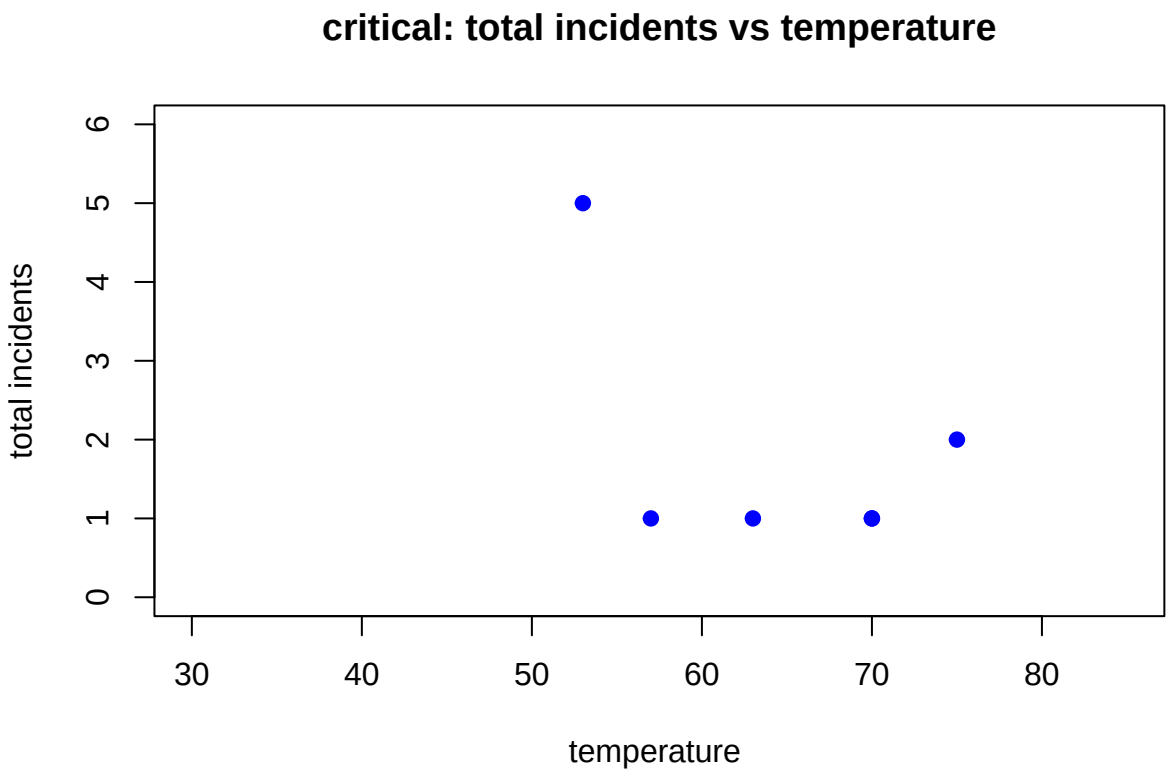
```
## [1] 1 1 1 2 2 2 3 3 3
```

MB.CH1.2

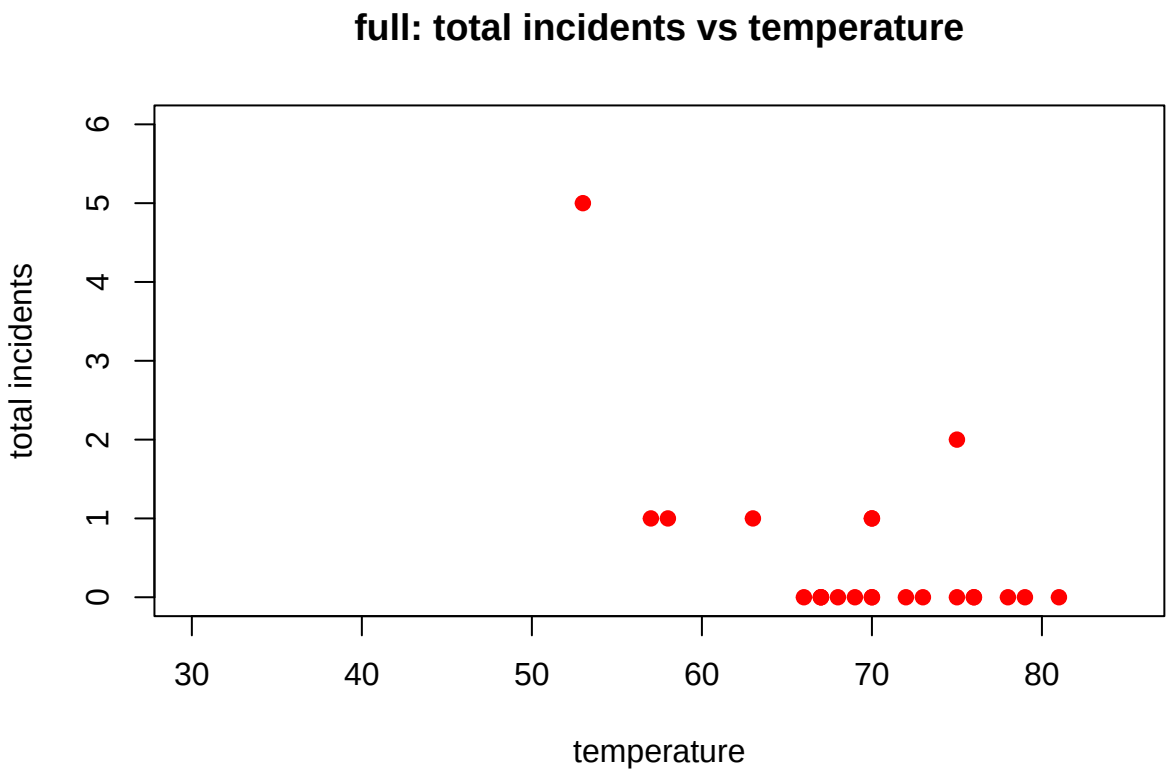
用以下代码解决问题。

```
data(orings)
# 创建子集数据框
selected_rows <- c(1, 2, 4, 11, 13, 18)
orings_subset <- orings[selected_rows, ]

data(orings)
```

```r
# 安装并加载 DAAG 包（如果未安装）
if (!require("DAAG")) install.packages("DAAG")
library(DAAG)

# 加载 orings 数据集
data(orings)

# 提取指定行创建新数据框
selected_rows <- c(1, 2, 4, 11, 13, 18)
orings_subset <- orings[selected_rows, ]

# 添加总事故次数列（Erosion + Blowby）
orings_subset$Total_incidents <- orings_subset$Erosion + orings_subset$Blowby
orings$Total_incidents <- orings$Erosion + orings$Blowby

# 绘制子集数据的图形
plot(Total_incidents ~ Temperature,
     data = orings_subset,
     main = "critical: total incidents vs temperature",
     xlab = "temperature ",
     ylab = "total incidents",
     pch = 19, col = "blue",
     xlim = c(30, 85), ylim = c(0, 6))
```




```r
# 绘制完整数据的图形
plot(Total_incidents ~ Temperature,
     data = orings,
     main = "full: total incidents vs temperature",
     xlab = "temperature",
     ylab = "total incidents",
     pch = 19, col = "red",
     xlim = c(30, 85), ylim = c(0, 6))
```



MB.CH1.4

a.

用以下代码解决问题。不存在缺失的列。

```
library(DAAG)

str(ais)
```

```
## 'data.frame':    202 obs. of  13 variables:
##  $ rcc   : num  3.96 4.41 4.14 4.11 4.45 4.1 4.31 4.42 4.3 4.51 ...
##  $ wcc   : num  7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...
##  $ hc    : num  37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...
##  $ hg    : num  12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...
##  $ ferr  : num  60 68 21 69 29 42 73 44 41 44 ...
##  $ bmi   : num  20.6 20.7 21.9 21.9 19 ...
##  $ ssf   : num  109.1 102.8 104.6 126.4 80.3 ...
##  $ pcBfat: num  19.8 21.3 19.9 23.7 17.6 ...
```

```
##  $ lbm    : num  63.3 58.5 55.4 57.2 53.2 ...
##  $ ht     : num  196 190 178 185 185 ...
##  $ wt     : num  78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...
##  $ sex    : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sport  : Factor w/ 10 levels "B_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
colSums(is.na(ais))
```

```
##    rcc    wcc     hc     hg   ferr    bmi    ssf pcBfat    lbm     ht     wt
##      0      0      0      0      0      0      0      0      0      0      0
##    sex  sport
##      0      0
```

b.

用一下代码解决问题。

```
# 创建性别-运动项目交叉表
gender_sport_table <- table(ais$sport, ais$sex)

# 计算男女比例（男性比例）
male_ratio <- gender_sport_table[, "m"] / rowSums(gender_sport_table)

# 识别失衡项目（男性比例 > 2/3 或 < 1/3）
imbalanced_sports <- names(which(male_ratio > 2/3 | male_ratio < 1/3))

# 打印结果
print(" 性别-运动项目分布表:")
```

```
## [1] "性别-运动项目分布表:"
```

```
print(gender_sport_table)
```

```
##
##            f  m
##   B_Ball  13 12
##   Field    7 12
##   Gym      4  0
##   Netball 23  0
##   Row     22 15
```

```
##   Swim     9 13
##   T_400m  11 18
##   T_Sprnt  4 11
##   Tennis   7  4
##   W_Polo   0 17
```

```
cat("\n性别比例失衡的运动项目（比例 > 2:1）:")
```

```
##
## 性别比例失衡的运动项目（比例 > 2:1）:
```

```
print(imbalanced_sports)
```

```
## [1] "Gym"     "Netball" "T_Sprnt" "W_Polo"
```

MB.CH1.6

a.

Y 轴刻度：表示 $\log_2$(面积)

Y 轴增加 1 单位 → 面积翻倍 ($2^1 = 2$ 倍)

Y 轴增加 2 单位 → 面积变为 4 倍 ($2^2 = 4$ 倍)

Y 轴增加 3 单位 → 面积变为 8 倍 ($2^3 = 8$ 倍)

点标签：

左侧数字：实际面积（km^2）

右侧文字：湖泊名称

关键观察：

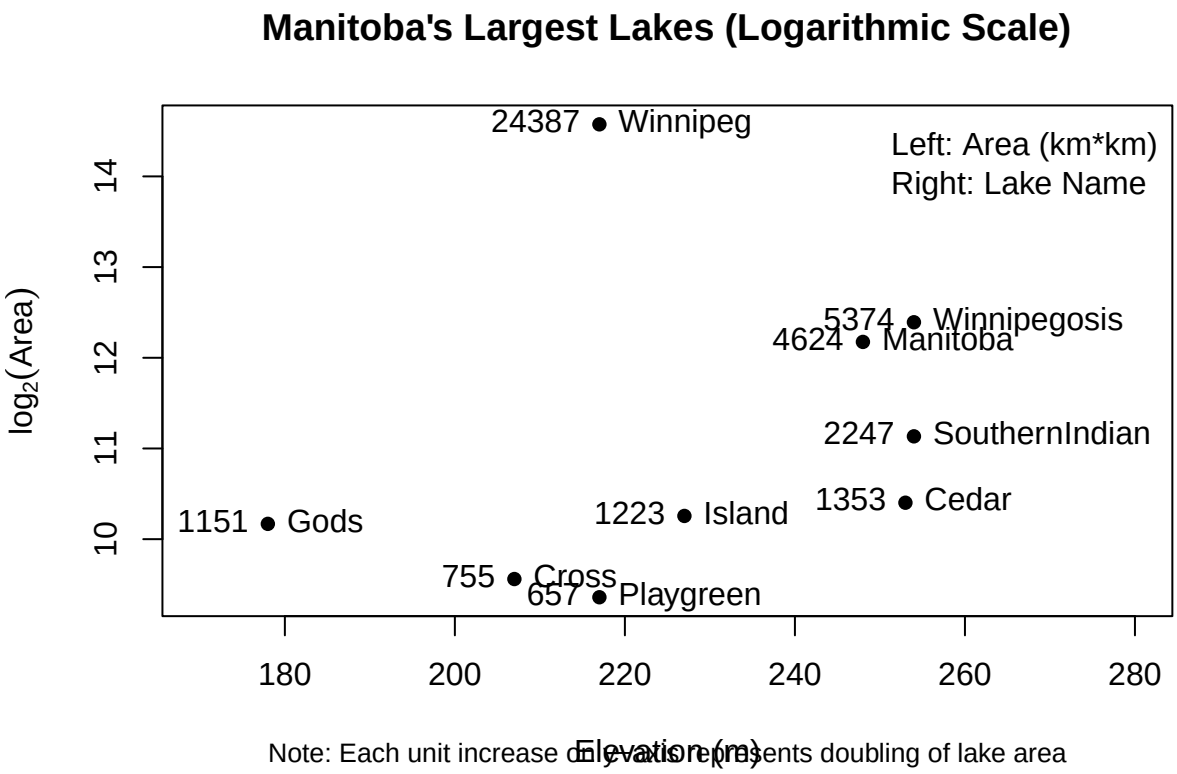
Winnipeg 湖面积最大（$\log_2(24387) \approx 14.6$），远大于其他湖泊

Gods 湖海拔最低但面积中等（178m, 1151km^2）

海拔与面积无明显相关性

```
Manitoba.lakes <- data.frame(
  elevation = c(217, 254, 248, 254, 253, 227, 178, 207, 217),
  area = c(24387, 5374, 4624, 2247, 1353, 1223, 1151, 755, 657)
)
```

```r
row.names(Manitoba.lakes) <- c("Winnipeg", "Winnipegosis", "Manitoba",
                               "SouthernIndian", "Cedar", "Island",
                               "Gods", "Cross", "Playgreen")
attach(Manitoba.lakes)

plot(elevation, log2(area), pch = 16, xlim = c(170, 280),
     xlab = "Elevation (m)", ylab = expression(log[2](Area)),
     main = "Manitoba's Largest Lakes (Logarithmic Scale)")

text(elevation, log2(area), labels = row.names(Manitoba.lakes), pos = 4)


text(elevation, log2(area), labels = area, pos = 2)

legend("topright",
       legend = c("Left: Area (km*km)", "Right: Lake Name"),
       text.col = c("black", "black"), bty = "n")

mtext("Note: Each unit increase on y-axis represents doubling of lake area",
      side = 1, line = 3, cex = 0.8)
```



b.

Y 轴刻度：对数变换后等距

从 500 到 1000 的距离 = 从 1000 到 2000 的距离（均表示面积翻倍）

大湖（Winnipeg）与小湖（Cross）差异更明显

优势：

直接显示实际面积值

保持面积比例关系直观

避免 Winnipeg 湖压缩其他数据点

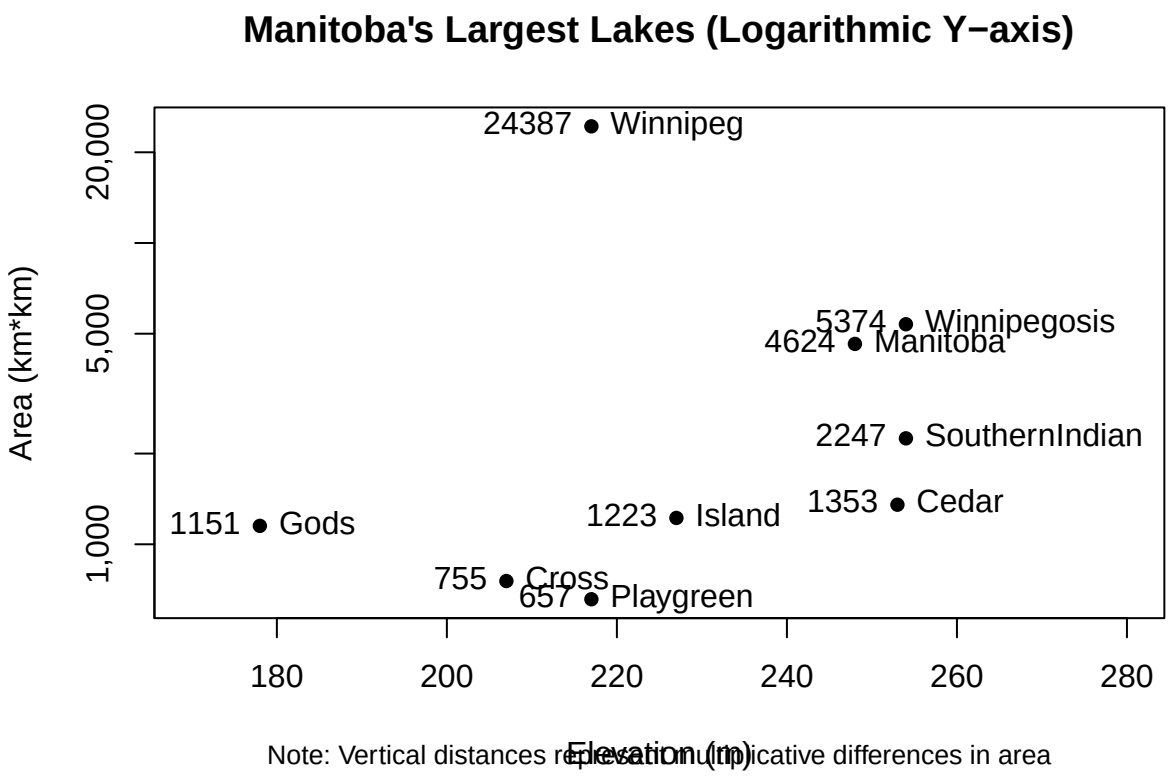
与 (a) 对比：

相同的数据关系

不同的视觉表示

对数 Y 轴图更易解释面积差异

```r
# 创建对数 Y 轴图形
plot(area ~ elevation, data = Manitoba.lakes, pch = 16,
     xlim = c(170, 280), log = "y",
     xlab = "Elevation (m)", ylab = "Area (km*km)",
     main = "Manitoba's Largest Lakes (Logarithmic Y-axis)",
     yaxt = "n")  # 禁用默认 Y 轴

# 添加自定义对数刻度
axis(2, at = c(500, 1000, 2000, 5000, 10000, 20000),
     labels = c("500", "1,000", "2,000", "5,000", "10,000", "20,000"))

# 添加湖泊名称标签 (右侧)
text(Manitoba.lakes$elevation, Manitoba.lakes$area,
     labels = row.names(Manitoba.lakes), pos = 4)

# 添加面积数值标签 (左侧)
text(Manitoba.lakes$elevation, Manitoba.lakes$area,
     labels = Manitoba.lakes$area, pos = 2)

# 添加比例解释
mtext("Note: Vertical distances represent multiplicative differences in area",
      side = 1, line = 3, cex = 0.8)
```



关键结论：两种对数表示都揭示了 Winnipeg 湖的绝对主导地位（占所有湖泊总面积的约 70%），但未显示海拔与面积的显著相关性。对数变换对于可视化跨度大的面积数据至关重要。