myHomework1

chenpeng

2024-07-01

目录

Homework 1

T1

a.

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

b.

```
cat("iowa.df has",nrow(iowa.df),"rows and",ncol(iowa.df),"columns\n")
```

```
## iowa.df has 33 rows and 10 columns
```

c.

```r
cat("The names of iowa.df:\n")
```

```
## The names of iowa.df:
```

```r
cat(names(iowa.df),sep = ", ")
```

```
## Year, Rain0, Temp1, Rain1, Temp2, Rain2, Temp3, Rain3, Temp4, Yield
```

d.

```r
cat("The value of row 5, column 7 of iowa.df:",iowa.df[5, 7])
```

```
## The value of row 5, column 7 of iowa.df: 79.7
```

e.

```r
cat("The second row of iowa.df:\n")
```

```
## The second row of iowa.df:
```

```r
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
```

T2

a.

```r
# 正确，向量赋值
vector1 <- c("5", "12", "7", "32")
# 正确，取最大值
max(vector1)
```

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

```r
# 正确，字符串排序
sort(vector1)
```

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

```r
# 错误，字符串不能求和
sum(vector1)
```

```
## Error in sum(vector1): 'type'(character)参数无效
```

b.

```r
vector2 <- c("5",7,12)
# 错误，由于此时 vector2 为字符型向量
vector2[2] + vector2[3]
```

```
## Error in vector2[2] + vector2[3]: 二进列运算符中有非数值参数
```

```r
# 正确，由于 data.frame 类型可以存储多种类型的数据
dataframe3 <- data.frame(z1="5",z2=7,z3=12)
dataframe3[1,2] + dataframe3[1,3]
```

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

```r
list4 <- list(z1="6", z2=42, z3="49", z4=126)
# 正确，引用的是数值
list4[[2]]+list4[[4]]
```

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

```r
# 错误，引用的是列表
list4[2]+list4[4]
```

```
## Error in list4[2] + list4[4]: 二进列运算符中有非数值参数
```

T3

a.

```r
seq(1, 1e4, by = 372)
```

```
##  [1]    1  373  745 1117 1489 1861 2233 2605 2977 3349 3721 4093 4465 4837 5209
## [16] 5581 5953 6325 6697 7069 7441 7813 8185 8557 8929 9301 9673
```

```r
seq(1, 1e4, length.out = 50)
```

```
##  [1]     1.0000   205.0612   409.1224   613.1837   817.2449  1021.3061
##  [7]  1225.3673  1429.4286  1633.4898  1837.5510  2041.6122  2245.6735
## [13]  2449.7347  2653.7959  2857.8571  3061.9184  3265.9796  3470.0408
## [19]  3674.1020  3878.1633  4082.2245  4286.2857  4490.3469  4694.4082
## [25]  4898.4694  5102.5306  5306.5918  5510.6531  5714.7143  5918.7755
## [31]  6122.8367  6326.8980  6530.9592  6735.0204  6939.0816  7143.1429
## [37]  7347.2041  7551.2653  7755.3265  7959.3878  8163.4490  8367.5102
## [43]  8571.5714  8775.6327  8979.6939  9183.7551  9387.8163  9591.8776
## [49]  9795.9388 10000.0000
```

b.

```r
# 1:3 序列，重复输出整个序列三次
rep(1:3, times=3)
```

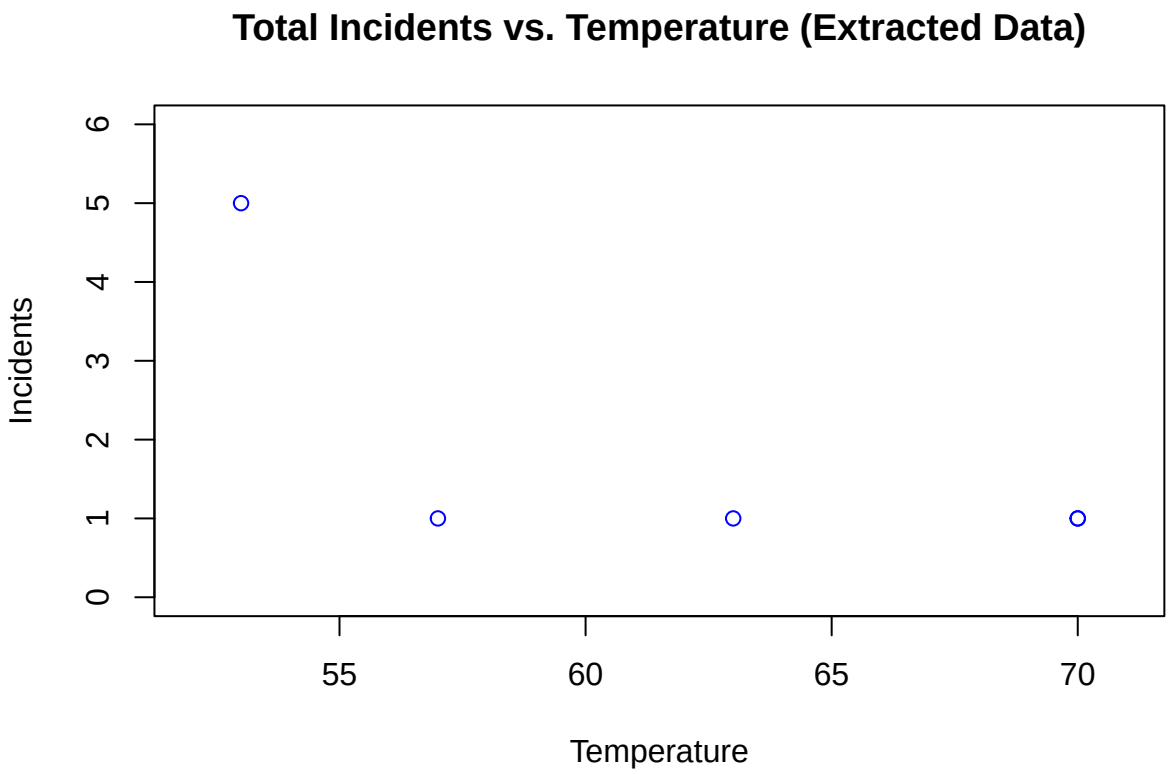
```
## [1] 1 2 3 1 2 3 1 2 3
```

```r
# 1:3 序列，序列中每个数重复输出三次
rep(1:3, each=3)
```

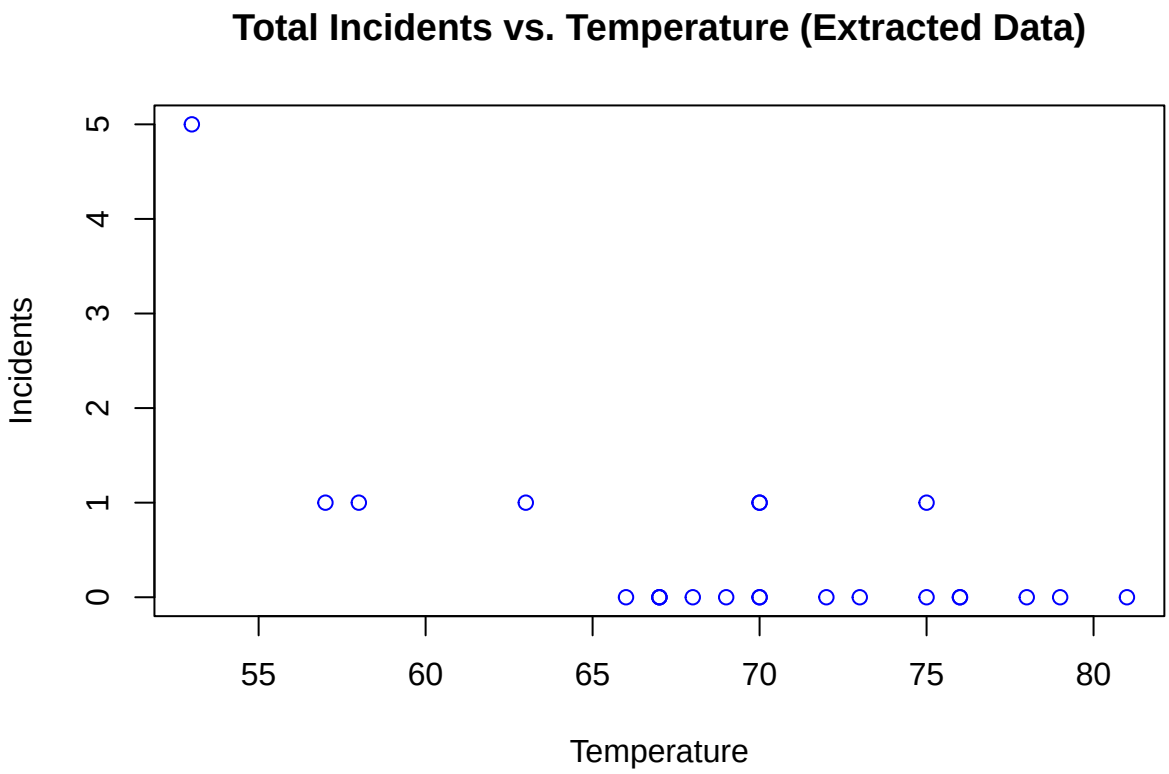
```
## [1] 1 1 1 2 2 2 3 3 3
```

MB.Ch1.2

```r
library(DAAG)
obv <- DAAG::orings
obv_ch <- obv[c(1, 2, 4, 11, 13),]
plot(obv_ch$Temperature, obv_ch$Total,
     col = "blue",
     main = "Total Incidents vs. Temperature (Extracted Data)",
     xlab = "Temperature", ylab = "Incidents",
     xlim = c(min(obv_ch$Temperature)-1,max(obv_ch$Temperature)+1), ylim = c(min(obv_ch$Total)-1,m
```




```
plot(obv$Temperature, obv$Total,
     col = "blue",
     main = "Total Incidents vs. Temperature (Extracted Data)",
     xlab = "Temperature", ylab = "Incidents")
```



MB.Ch1.4

(a)

```
# Load the dataset 'ais'
data(ais)

# Use str() to get information about the dataset
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 ...
```

```r
# Check for missing values in each column
any_na <- apply(is.na(ais), 2, any)

# Print the result
cat("\n judge whether the column holds NA:\n")
```

```
##
##  judge whether the column holds NA:
```

```r
print(any_na)
```

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

(b)

```r
gender_sport_table <- table(ais$sport, ais$sex)

gender_sport_df <- as.data.frame.matrix(gender_sport_table)

# Print the table
print(gender_sport_df)
```

```
##           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
```

MB.Ch1.6

```
lake_names <- c("Winnipeg", "Winnipegosis", "Manitoba", "SouthernIndian",
                "Cedar", "Island", "Gods", "Cross", "Playgreen")
elevation <- c(217, 254, 248, 254, 253, 227, 178, 207, 217)
area <- c(24387, 5374, 4624, 2247, 1353, 1223, 1151, 755, 657)

Manitoba.lakes <- data.frame(elevation = elevation, area = area)
row.names(Manitoba.lakes) <- lake_names

print(Manitoba.lakes)
```

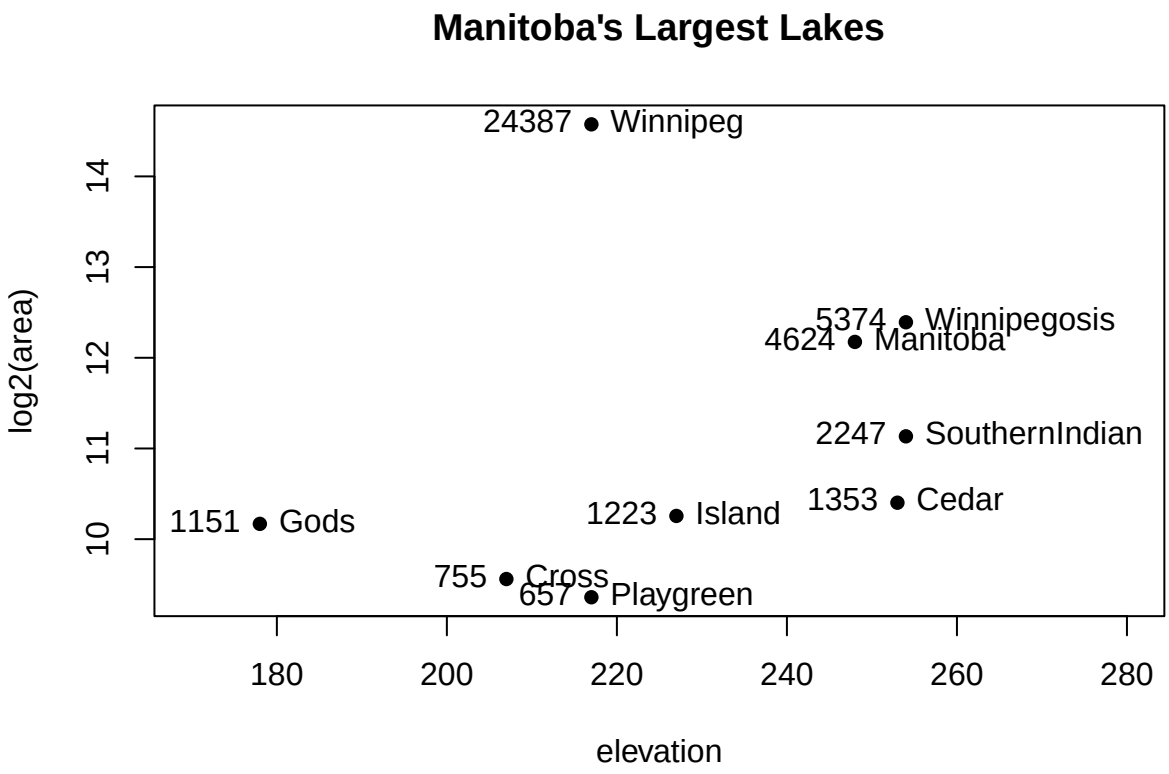
```
##                elevation  area
## Winnipeg             217 24387
## Winnipegosis         254  5374
## Manitoba             248  4624
## SouthernIndian       254  2247
## Cedar                253  1353
## Island               227  1223
## Gods                 178  1151
## Cross                207   755
## Playgreen            217   657
```

(a)

```
attach(Manitoba.lakes)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     area, elevation
```

```
plot(log2(area) ~ elevation, pch=16, xlim=c(170,280))
# NB: Doubling the area increases log2(area) by 1.0
text(log2(area) ~ elevation, labels=row.names(Manitoba.lakes), pos=4)
text(log2(area) ~ elevation, labels=area, pos=2)
title("Manitoba's Largest Lakes")
```



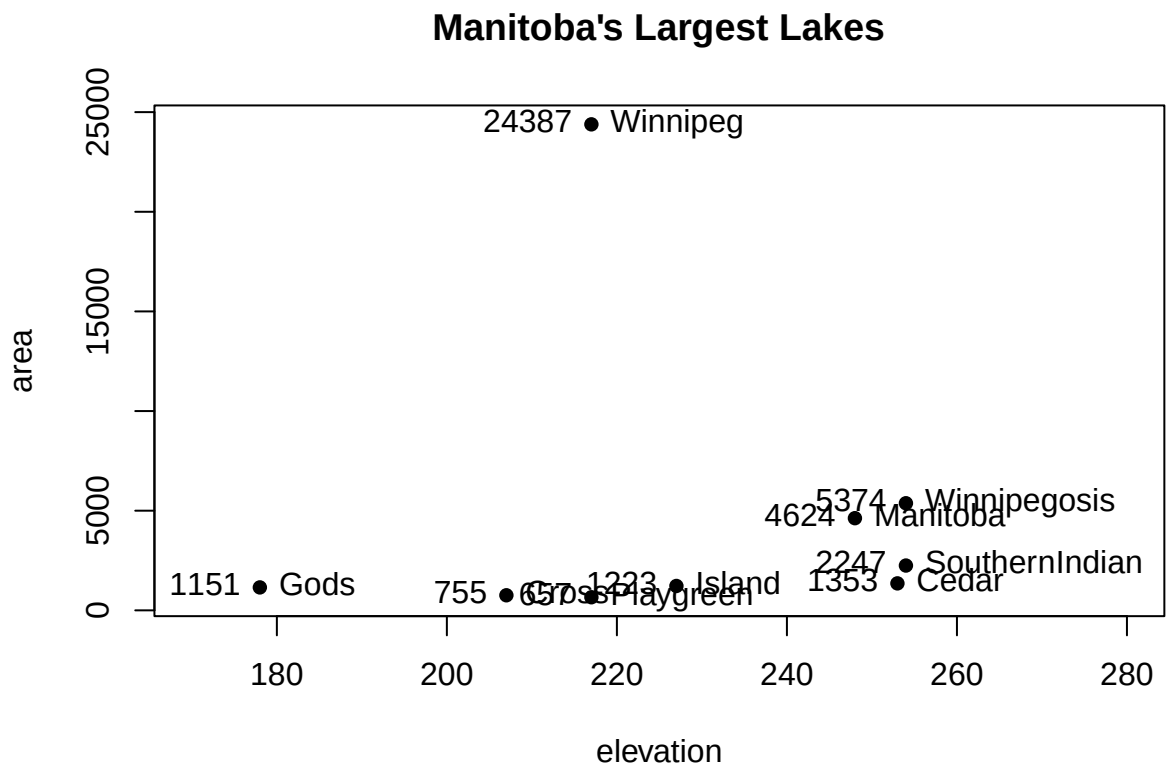
The label on the left of the point is its area. The label on the right of the point is its name. The y-axis is labeled as $\log_2(area)$.

This transformation is applied because the area values of the lakes vary widely, with some being much larger than others. Using $\log_2(area)$ allows for better visualization by compressing the scale while maintaining the proportional differences between the lakes' areas. Each tick on the y-axis represents a doubling (or halving) of the area due to the logarithmic nature of the scale:

- A move upward by 1 unit on the y-axis corresponds to doubling the area of the lake.
- Conversely, a move downward by 1 unit on the y-axis corresponds to halving the area of the lake.

(b)

```
plot(area ~ elevation, pch=16, xlim=c(170,280), ylog=T)
text(area ~ elevation, labels=row.names(Manitoba.lakes), pos=4, ylog=T)
text(area ~ elevation, labels=area, pos=2, ylog=T)
title("Manitoba's Largest Lakes")
```


Manitoba's Largest Lakes
area
elevation
0
5000
15000
25000
180
200
220
240
260
280
24387 Winnipeg
5374 Winnipegosis
4624 Manitoba
2247 SouthernIndian
1353 Cedar
1223 Island
1151 Gods
755
657
Playgreen