Homework1 Report

王梓

2024-07-01

problem 1

a.

```
help("read.csv")
```

```
## 打开httpd帮助服务器… 好了
```

```
iowa.df<-read.csv("data/iowa.csv",header=T)
```

b.

```
nrow(iowa.df)
```

```
## [1] 33
```

```
ncol(iowa.df)
```

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

c.

```
names(iowa.df)
```

```
## [1] "Year.Rain0.Temp1.Rain1.Temp2.Rain2.Temp3.Rain3.Temp4.Yield"
```

d.

```
iowa.df[5, 7]
```

```
## NULL
```

e.

```
iowa.df[2,]
```

```
## [1] "1931;14.76;57.5;3.83;75;2.72;77.2;3.3;72.6;32.9"
```

problem2

a.

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

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

```
sort(vector1)
```

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

```
# sum(vector1)
```

vector1中的数据是 `character` 类型而不是数字类型，因此排序和取最大值时会按照字典序进行排序，sum操作也无法进行。

b.

```
vector2 <- c("5",7,12)
# vector2[2] + vector2[3]
```

`c()` 返回的是 `vector` ，而 `vector` 中的所有数据都是同一类型的。 `c("5",7,12)` 中存在一个 `character` 类型的 `"5"` ，因此 vector2 中所有数据的类型均为 `character` ，无法进行 `+` 操作。

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

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

`dataframe` 中可以包含不同类型的数据， `dataframe3[1,2]` 和 `dataframe3[1,3]` 均为 `double` 类型，因此可以进行 `+` 运算。

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

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

```
# list4[2]+list4[4]
```

`list4[[2]]` 返回的是 `list4` 中第二个元素的数值，而 `list4[2]` 返回的是 `list4` 中第二个元素的引用，因此无法进行 `+` 运算。

problem3

a.

```
seq(1, 10000, 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
```

```
seq(1, 10000, 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

```
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
```

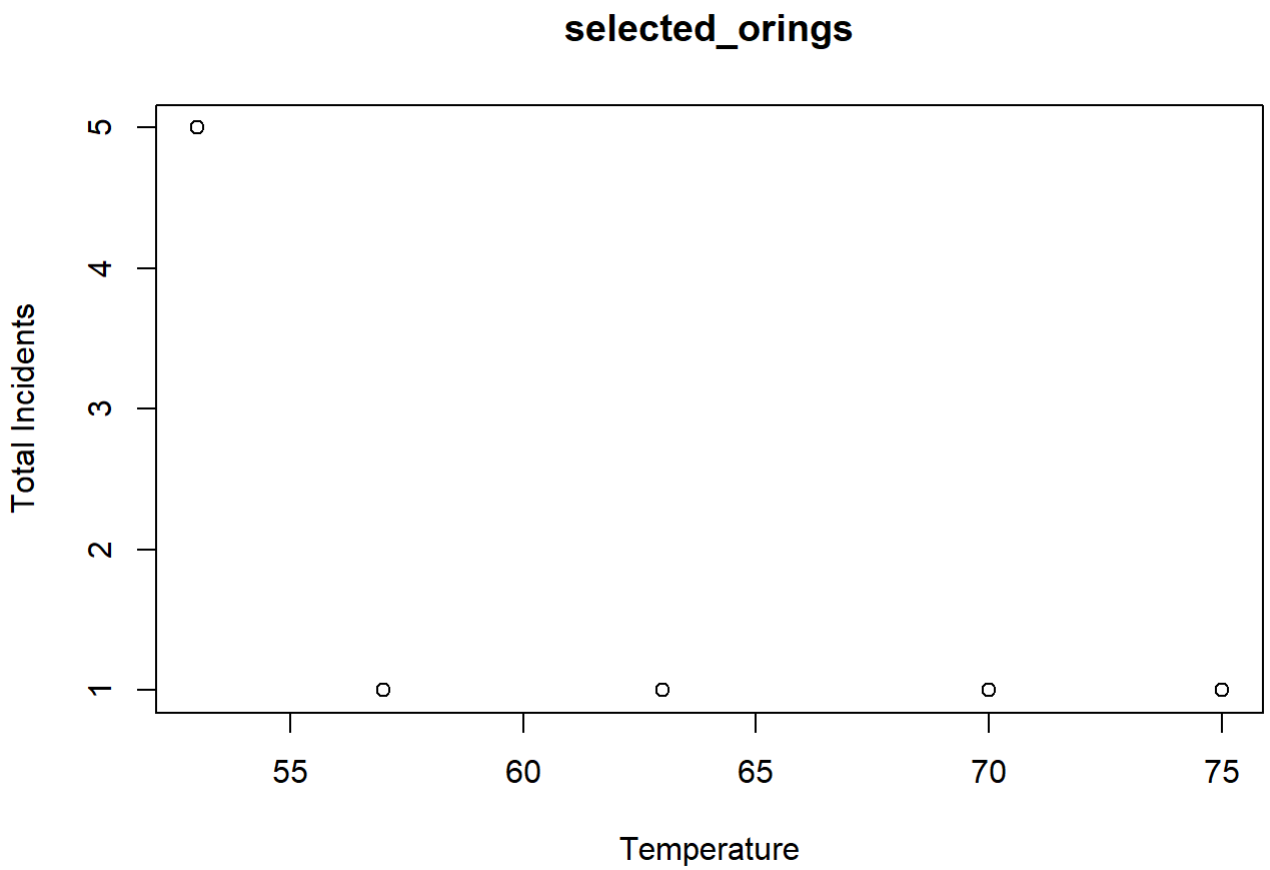
`rep(1:3, times=3)` 是把 `1:3` 这个序列完整地重复3次，而 `rep(1:3, each=3)` 是把 `1:3` 这个序列中的每个元素依次重复3次。

MB.Ch1.2

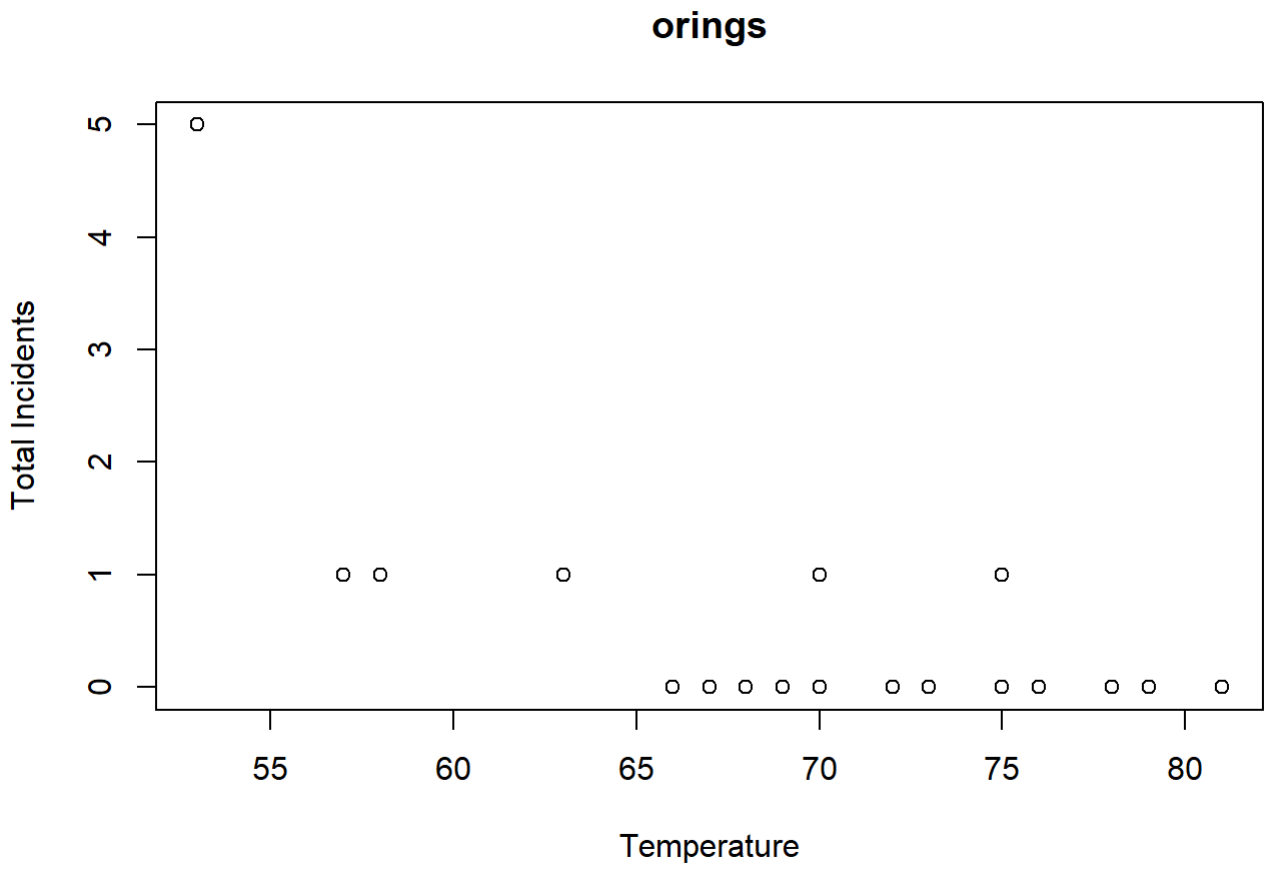
```
if (!require(DAAG)) install.packages("DAAG")
```

```
## 载入需要的程序包：DAAG
```

```
library(DAAG)
data(orings)
selected_orings <- orings[c(1, 2, 4, 11, 13, 18), ]
plot(selected_orings$Temperature, selected_orings$Total, xlab = "Temperature", ylab = "Total Incidents", main = "selected_orings")
```




```
plot(orings$Temperature, orings$Total, xlab = "Temperature", ylab = "Total Incidents", main = "orings")
```



MB.Ch1.4

(a)

```
data(ais)
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 ...
```

```
sum(is.na(ais))
```

```
## [1] 0
```

数据中不存在缺失值

(b)

```
(sex_sport_table <- table(ais$sex, ais$sport))
```

```
##
##     B_Ball Field Gym Netball Row Swim T_400m T_Sprnt Tennis W_Polo
##   f     13     7   4      23  22    9     11       4      7      0
##   m     12    12   0       0  15   13     18      11      4     17
```

```
ratio <- sex_sport_table["f", ] / sex_sport_table["m", ]
imbalanced_sports <- names(which(ratio > 2 | ratio < 0.5))
print(imbalanced_sports)
```

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

MB.Ch.1.6

```
elevation <- c(217, 254, 248, 254, 523, 227, 178, 207, 217)
area <- c(24387, 5374, 4624, 2247, 1353, 1223, 1151, 755, 657)
lake_names <- c("Winnipeg", "Winnipegosis", "Manitoba", "SouthernIndian", "Cedar", "Island", "Gods", "Cross", "Playgreen")

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

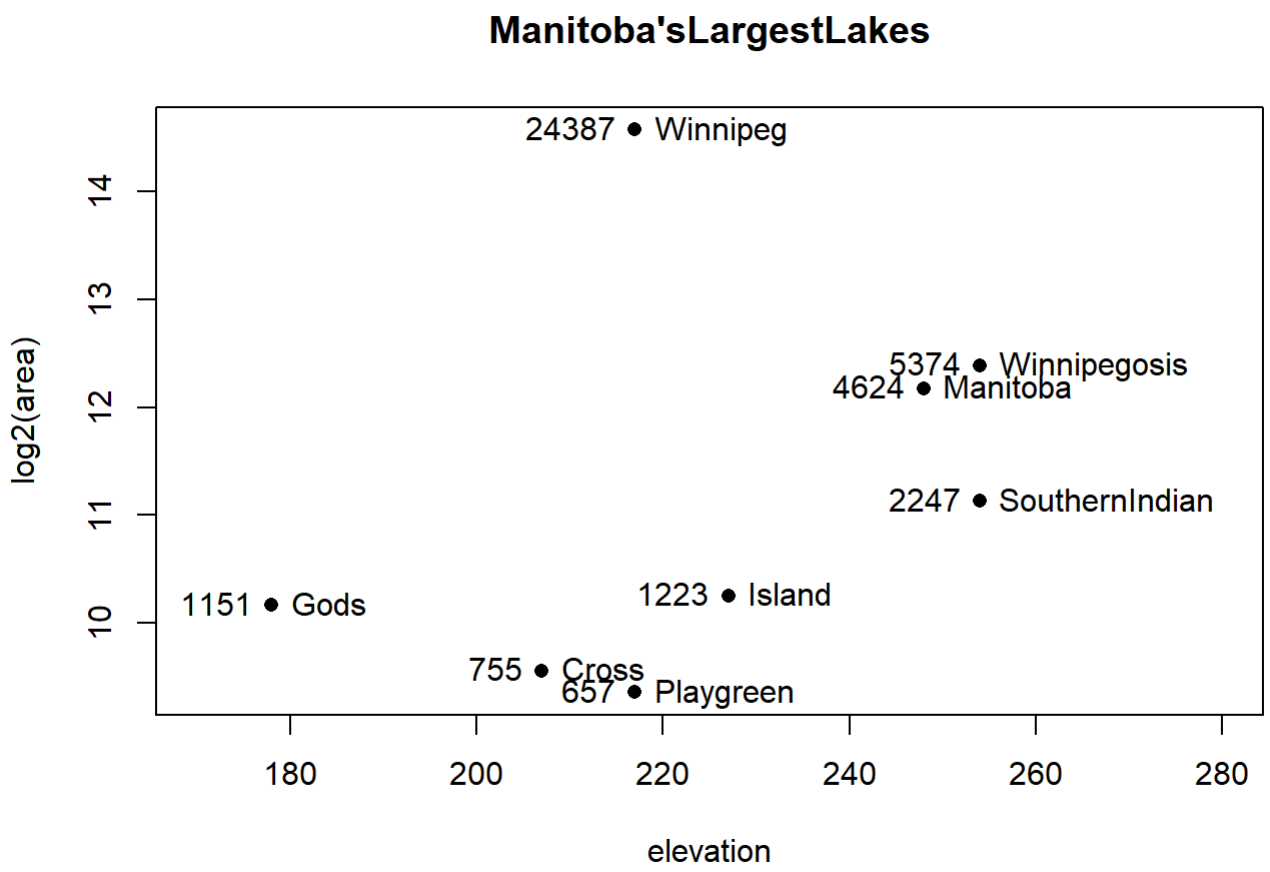
```
##                elevation  area
## Winnipeg             217 24387
## Winnipegosis         254  5374
## Manitoba             248  4624
## SouthernIndian       254  2247
## Cedar                523  1353
## Island               227  1223
## Gods                 178  1151
## Cross                207   755
## Playgreen            217   657
```

(a)

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

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

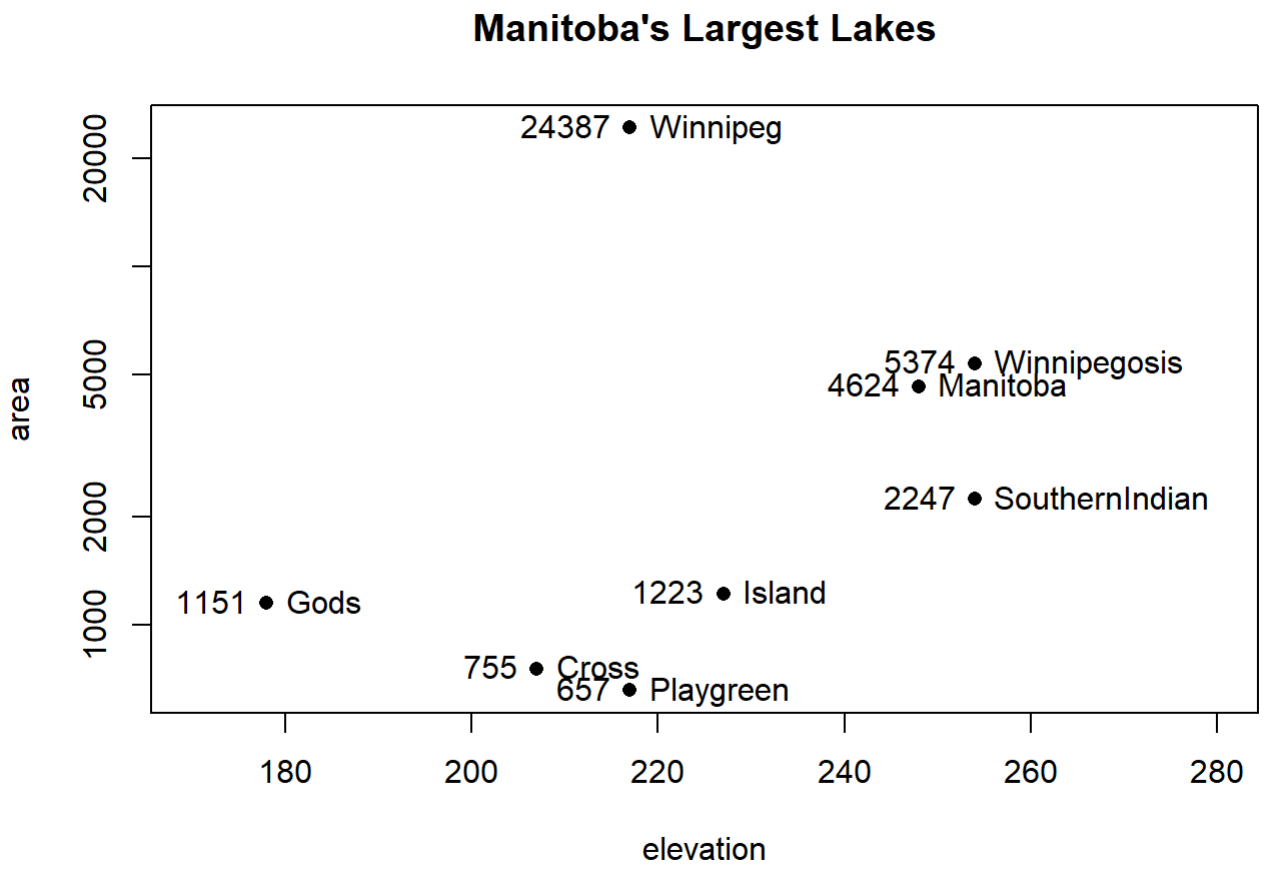
```
plot(log2(area) ~ elevation,pch=16,xlim=c(170,280))
#NB:Doublingtheareaincreaseslog2(area)by1.0
text(log2(area) ~ elevation,labels=row.names(Manitoba.lakes),pos=4)
text(log2(area) ~ elevation,labels=area,pos=2)
title("Manitoba'sLargestLakes")
```



图中的点左侧的label表示的是湖的面积，右侧的label表示的是湖的名称。y轴表示的是湖的面积取以2为底数的对数后的结果，该指标每增长一个单位表示湖的面积增加一倍。

(b)

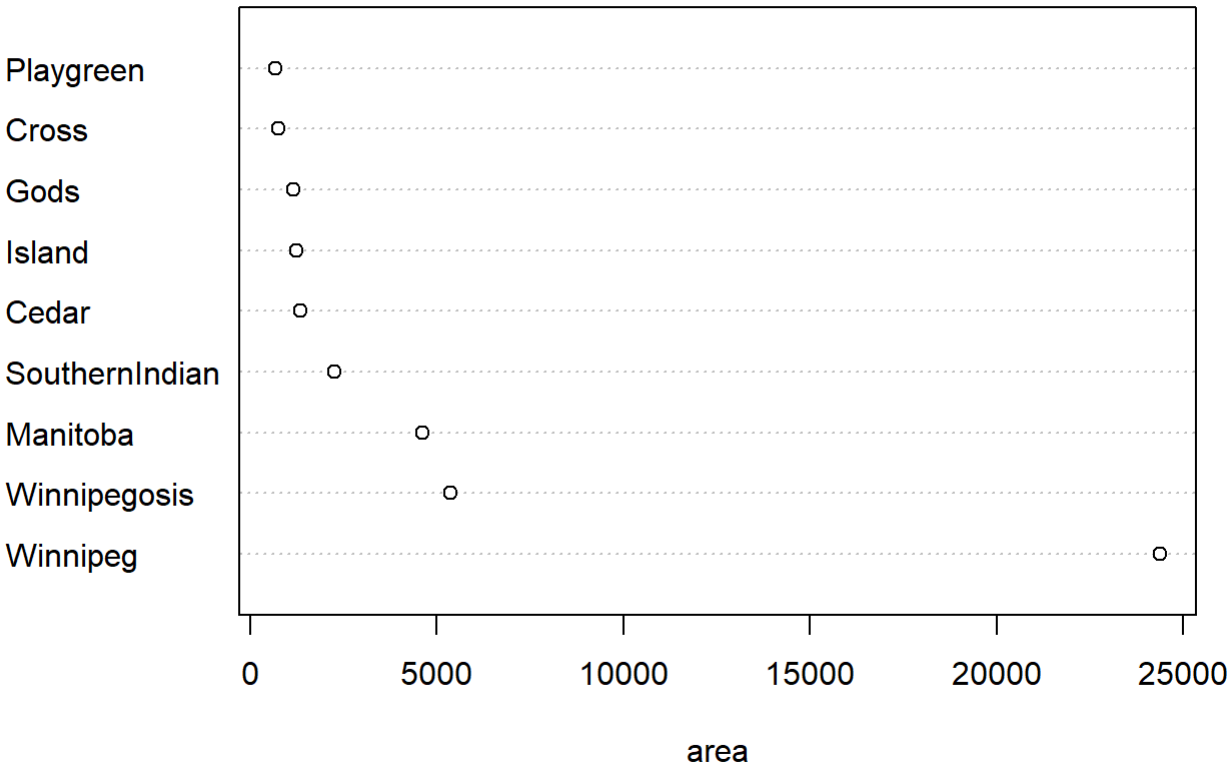
```
plot(area ~ elevation, pch=16, xlim=c(170,280), log="y")
text(area ~ elevation, labels=row.names(Manitoba.lakes), pos=4, ylog=T)
text(area ~ elevation, labels=area, pos=2, ylog=T)
title("Manitoba's Largest Lakes")
```



MB.Ch1.7

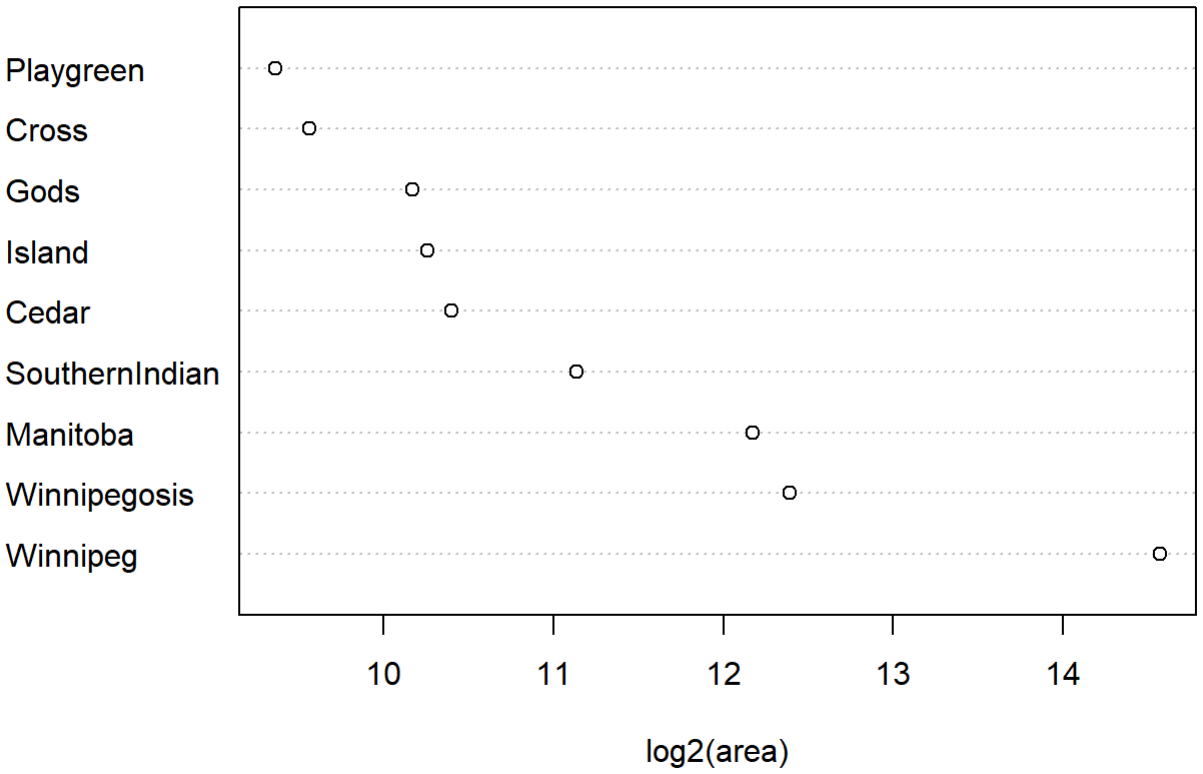
(a)

```
dotchart(area, xlab = "area", labels = row.names(Manitoba.lakes))
```



(b)

```
dotchart(log2(area), xlab = "log2(area)", labels = row.names(Manitoba.lakes))
```



MB.Ch1.8

```
lower_bound_of_area_covered_by_water <- sum(area)
print(lower_bound_of_area_covered_by_water)
```

```
## [1] 41771
```