Homework-1

向晏平

1. The Iowa data set iowa.csv is a toy example that summarises the yield of wheat (bushels per acre) for the state of Iowa between 1930-1962. In addition to yield, year, rainfall and temperature were recorded as the main predictors of yield.

    a. First, we need to load the data set into R using the command `read.csv()`. Use the help function to learn what arguments this function takes. Once you have the necessary input, load the data set into R and make it a data frame called `iowa.df`.
    b. How many rows and columns does `iowa.df` have?
    c. What are the names of the columns of `iowa.df`?
    d. What is the value of row 5, column 7 of `iowa.df`?
    e. Display the second row of `iowa.df` in its entirety.

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

```
## [1] 10
```

```
nr <- nrow(iowa.df);nr
```

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

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

```
## [1] 79.7
```

```
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
```

2. Syntax and class-typing.

    a. For each of the following commands, either explain why they should be errors, or explain the non-erroneous result.

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

vector1 的元素是字符 (character),max() 返回字符串中第一个字符对应顺序最大的字符，sort() 也是按这个规则进行从小到大的排列，而 sum() 函数对象是数字不是字符。b. For the next series of commands, either explain their results, or why they should produce errors.

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

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

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

一个向量中只能有一种数据类型，vector2 以第一个元素为其数据类型，就是字符 (character). 字符不能做加法运算。数据框的数据种类可以是多样的，可以作加法运算。list 的一层引用返回的仍是一个 list，二层引用才是元素。

3. Working with functions and operators.

a. The colon operator will create a sequence of integers in order. It is a special case of the function `seq()` which you saw earlier in this assignment. Using the help command `?seq` to learn about the function, design an expression that will give you the sequence of numbers from 1 to 10000 in increments of 372. Design another that will give you a sequence between 1 and 10000 that is exactly 50 numbers in length.
b. The function `rep()` repeats a vector some number of times. Explain the difference between 'rep(1:3, times=3) and rep(1:3, each=3).

```
seq(1,1000, by=372)
```

```
## [1]   1 373 745
```

```
seq(1,1000, length.out = 50)
```

```
##  [1]    1.00000   21.38776   41.77551   62.16327   82.55102  102.93878
##  [7]  123.32653  143.71429  164.10204  184.48980  204.87755  225.26531
## [13]  245.65306  266.04082  286.42857  306.81633  327.20408  347.59184
## [19]  367.97959  388.36735  408.75510  429.14286  449.53061  469.91837
## [25]  490.30612  510.69388  531.08163  551.46939  571.85714  592.24490
## [31]  612.63265  633.02041  653.40816  673.79592  694.18367  714.57143
## [37]  734.95918  755.34694  775.73469  796.12245  816.51020  836.89796
## [43]  857.28571  877.67347  898.06122  918.44898  938.83673  959.22449
## [49]  979.61224 1000.00000
```

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

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

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

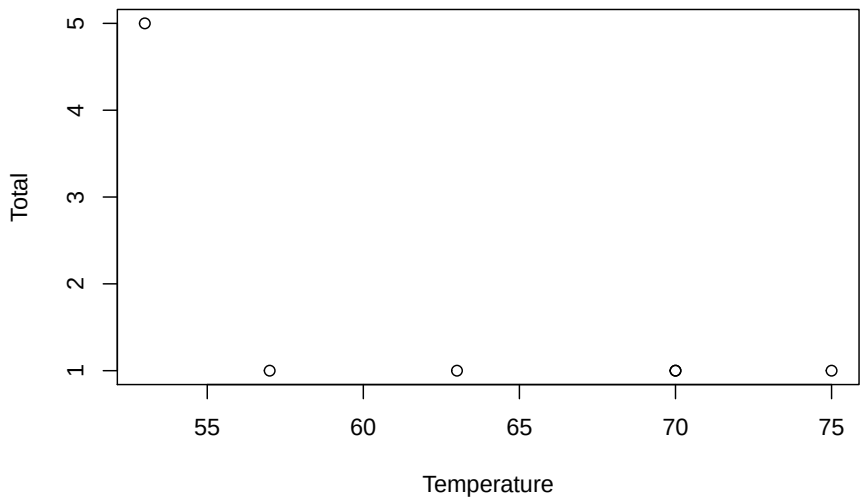
```
## [1] 1 1 1 2 2 2 3 3 3
```

each 参数是逐个输出元素，times 是整个输出数组。

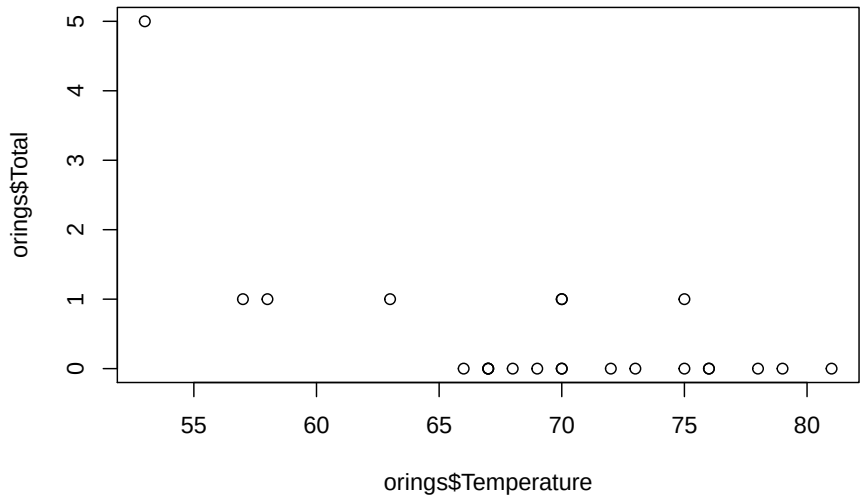
MB.Ch1.2. The orings data frame gives data on the damage that had occurred in US space shuttle launches prior to the disastrous Challenger launch of 28 January 1986. The observations in rows 1, 2, 4, 11, 13, and 18 were included in the pre-launch charts used in deciding whether to proceed with the launch, while remaining rows were omitted.

Create a new data frame by extracting these rows from orings, and plot total incidents against temperature for this new data frame. Obtain a similar plot for the full data set.

```
data(orings)
ri <- c(1,2,4,11,13,18)
plot(orings[ri,"Temperature"],orings[ri,"Total"],xlab = "Temperature",ylab="Total")
```




```
plot(orings$Temperature,orings$Total)
```



MB.Ch1.4. For the data frame ais (DAAG package)

(a) Use the function str() to get information on each of the columns. Determine whether any of the columns hold missing values.

```
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 ...
##  $ sport : Factor w/ 10 levels "B_Ball","Field",..: 1 1 1 1 1 1 1 1 1 ...
```

```
sum(is.na(ais)) #大于0则存在缺失值
```

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

(b) Make a table that shows the numbers of males and females for each different sport. In which sports is there a large imbalance (e.g., by a factor of more than 2:1) in the numbers of the two sexes?

```
attach(ais)
data_ais <- data.frame(ais[,c(12,13)])
result_ais <- aggregate(rep(1,nrow(data_ais)), data_ais, sum)
tb <- table(sport,sex)
which(tb[,2]/tb[,1]>2|tb[,1]/tb[,2]>2)
```

```
##     Gym Netball T_Sprnt  W_Polo
##       3       4       8      10
```

MB.Ch1.6.Create a data frame called Manitoba.lakes that contains the lake's elevation (in meters above sea level) and area (in square kilometers) as listed below. Assign the names of the lakes using the row.names() function. 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

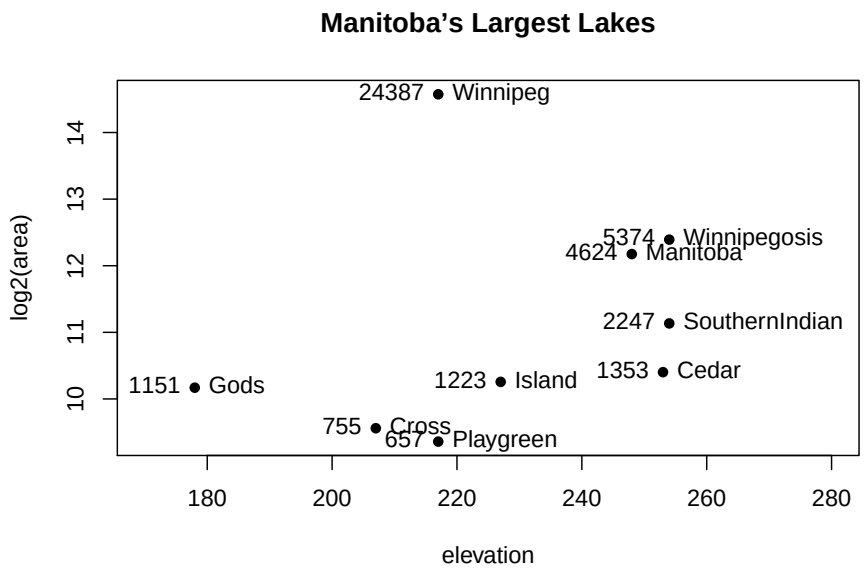
```
Manitoba.lakes <- data.frame(elevation = c(217,254,248,254,253,227,178,207,217),
                             area = c(24387,5374,4624,2247,1353,1223,1151,755,657))
row.names(Manitoba.lakes) <- c("Winnipeg","Winnipegosis","Manitoba","SouthernIndian","C
                               "Island","Gods","Cross","Playgreen")
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) Use the following code to plot log2(area) versus elevation, adding labeling infor- mation (there is an extreme value of area that makes a logarithmic scale pretty much essential):

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

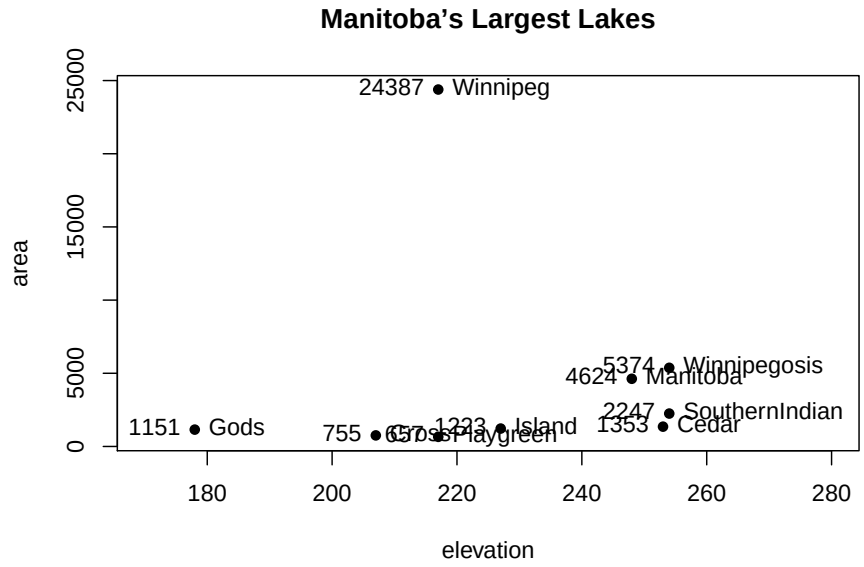


Devise captions that explain the labeling on the points and on the y-axis.

It will be necessary to explain how distances on the scale relate to changes in area.

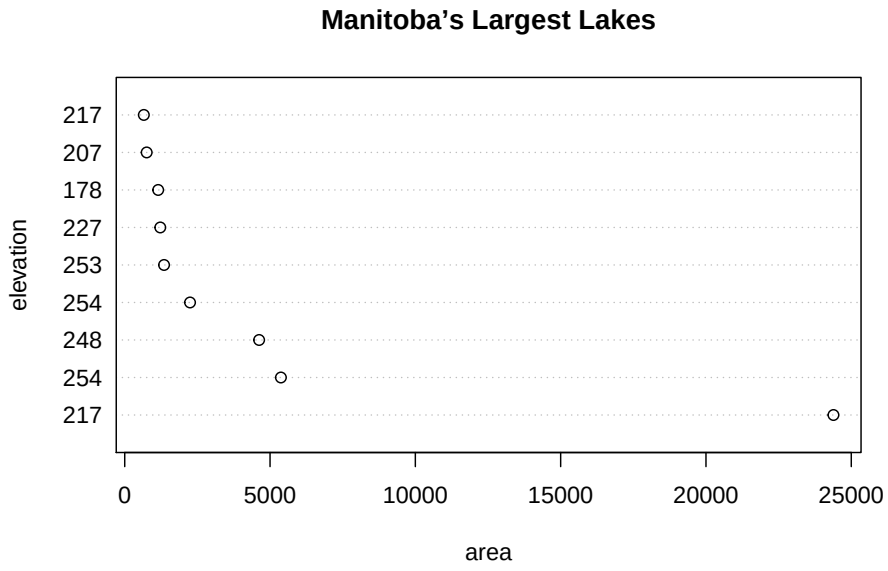
(b) Repeat the plot and associated labeling, now plotting area versus elevation, but specifying log="y" in order to obtain a logarithmic y-scale.

```
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")
```



MB.Ch1.7. Look up the help page for the R function dotchart(). Use this function to display the areas of the Manitoba lakes (a) on a linear scale, and (b) on a logarithmic scale. Add, in each case, suitable labeling information.

```
dotchart(area, elevation,main = "Manitoba's Largest Lakes",  xlab = "area",ylab = "ele
```



MB.Ch1.8. Using the sum() function, obtain a lower bound for the area of Manitoba covered by water.

```
sum(Manitoba.lakes[,"area"])
```

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