Homework 1

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? 31 行 1 列
    c. What are the names of the columns of `iowa.df`? colname = ["Year" "Rain0" "Temp1" "Rain1" "Temp2" "Rain2" "Temp3" "Rain3" "Temp4" "Yield"]
    d. What is the value of row 5, column 7 of `iowa.df`? 值是 79.7
    e. Display the second row of `iowa.df` in its entirety. row2 = [1931 14.76 57.5 3.83 75 2.72 77.2 3.3 72.6 32.9]

```
iowa.df<-read.csv("data/iowa.csv",sep = ";",header=T)
a <- dim(iowa.df)
colname <- colnames(iowa.df)
value <- iowa.df[5,7]
row2 <- iowa.df[2,]
```

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") #对的, vector1 = ["5"  "12" "7"  "32"]
max(vector1) #对的,max(vector1) = "7"
sort(vector1) #对的,sort(vector1) = ["12" "32" "5"  "7"]
sum(vector1) #错的, 字符不能求和
```

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] #错了, 二进列运算符中有非数值参数

dataframe3 <- data.frame(z1="5",z2=7,z3=12)
dataframe3[1,2] + dataframe3[1,3] #结果是19

list4 <- list(z1="6", z2=42, z3="49", z4=126)
list4[[2]]+list4[[4]] #对的, 结果是168
list4[2]+list4[4] #错的, 不是数值不能运算
```

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.

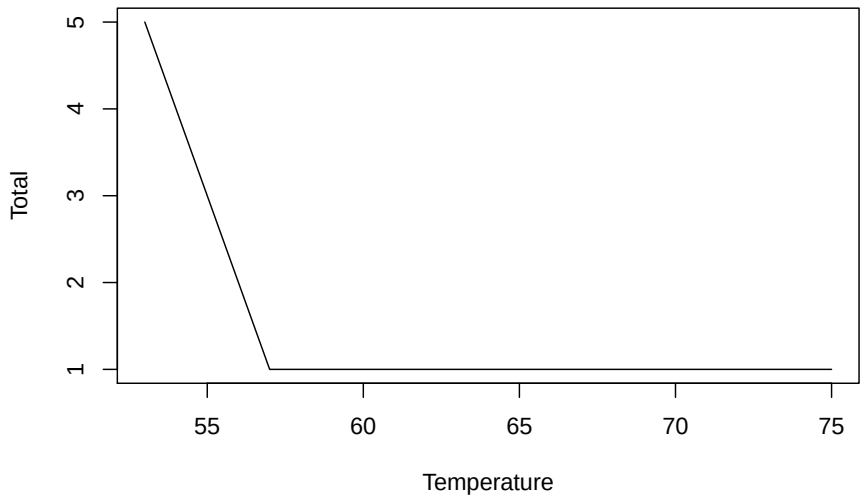
   b. The function `rep()` repeats a vector some number of times. Explain the difference between 'rep(1:3, times=3) and rep(1:3, each=3). rep(1:3, times=3) 的结果是 [1 2 3 1 2 3 1 2 3], rep(1:3,each = 3) 的结果是 [1 1 1 2 2 2 3 3 3]。一个重复三次, 一个是一次重复三次

```
a1 <- seq(1,10000,372)
a2 <- seq(1,10000,length.out = 50)
```

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.

```
a <- orings[1,]
for( i in 1:4) {
a[2,i] = orings[2,i]
a[3,i] = orings[4,i]
a[4,i] = orings[11,i]
a[5,i] = orings[13,i]
a[6,i] = orings[18,i]}
plot(a[,1],a[,4],type = 'l',xlab = "Temperature",ylab = "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.

```
for(i in 1:dim(ais)[2]){
    print(str(ais[,i]))
}
```

```
##  num [1:202] 3.96 4.41 4.14 4.11 4.45 4.1 4.31 4.42 4.3 4.51 ...
## NULL
##  num [1:202] 7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...
## NULL
##  num [1:202] 37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...
## NULL
##  num [1:202] 12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...
## NULL
##  num [1:202] 60 68 21 69 29 42 73 44 41 44 ...
## NULL
##  num [1:202] 20.6 20.7 21.9 21.9 19 ...
## NULL
##  num [1:202] 109.1 102.8 104.6 126.4 80.3 ...
## NULL
##  num [1:202] 19.8 21.3 19.9 23.7 17.6 ...
## NULL
##  num [1:202] 63.3 58.5 55.4 57.2 53.2 ...
## NULL
##  num [1:202] 196 190 178 185 185 ...
## NULL
##  num [1:202] 78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...
## NULL
##  Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
## NULL
##  Factor w/ 10 levels "B_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...
## NULL
```

(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?

```
aissport <- unique(as.character(ais$sport))
b <- list()
for (i in 1:length(aissport)){
  b <- c(b,list(c(0,0)))
}
names(b) <- aissport
for (i in 1:dim(ais)[1]){
  if (ais[i,12] == "f"){
    b[[ais[i,13]]][1] <- b[[ais[i,13]]][1] + 1
  }
  if (ais[i,12] == "m"){
    b[[ais[i,13]]][2] <- b[[ais[i,13]]][2] + 1
  }
}
d <- c()
for(i in 1:length(aissport)){
  if((b[[i]][2]/ b[[i]][1] > 2)){
    d <- c(d , aissport[i])
  }
}
print("男女比例不平衡的运动: ",)
```

```
## [1] "男女比例不平衡的运动: "
```

```
print(d)
```

```
## [1] "Tennis" "W_Polo"
```

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

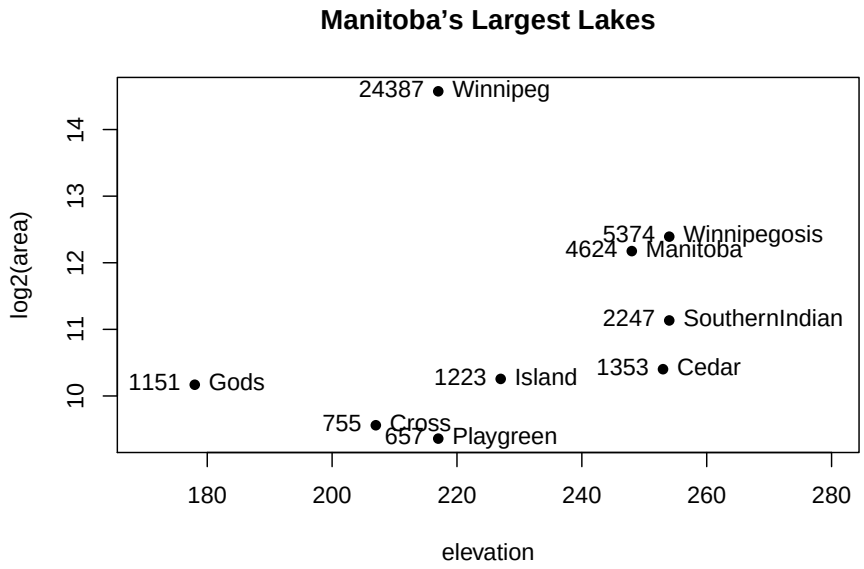
4624 SouthernIndian 254 2247 Cedar 253 1353 Island 227 1223 Gods 178 1151 Cross 207 755 Playgreen 217 657

```
a <- data.frame(elevation = c(217,254,248,254,253,227,178,207, 217),area = c(24387,5374
print(a)
```

```
##                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):

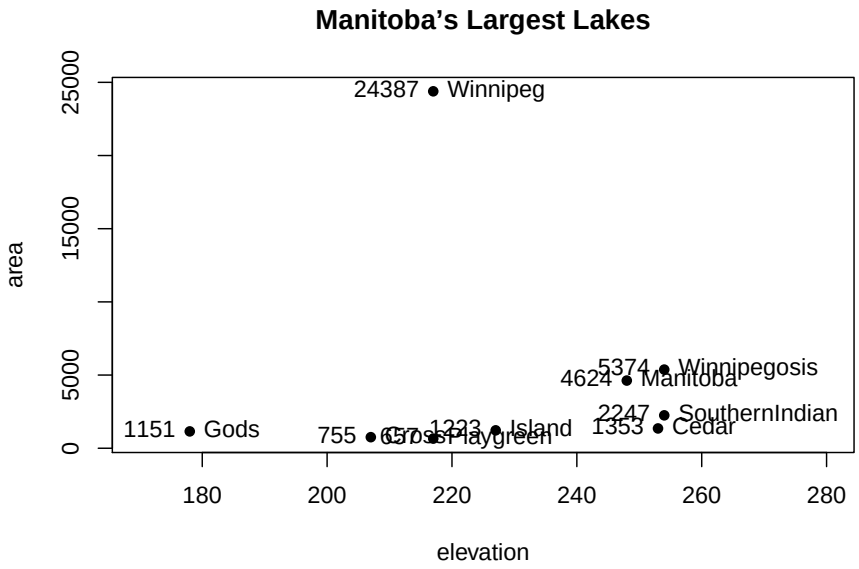
```
attach(Manitoba.lakes)
plot(log2(area) ~ elevation, pch=16, xlim=c(170,280))
# NB: Doubling the area increases log2(area) by 1.0
text(log2(area) ~ elevation, labels=row.names(Manitoba.lakes), pos=4)
text(log2(area) ~ elevation, labels=area, pos=2)
title("Manitoba' s Largest Lakes")
```



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.

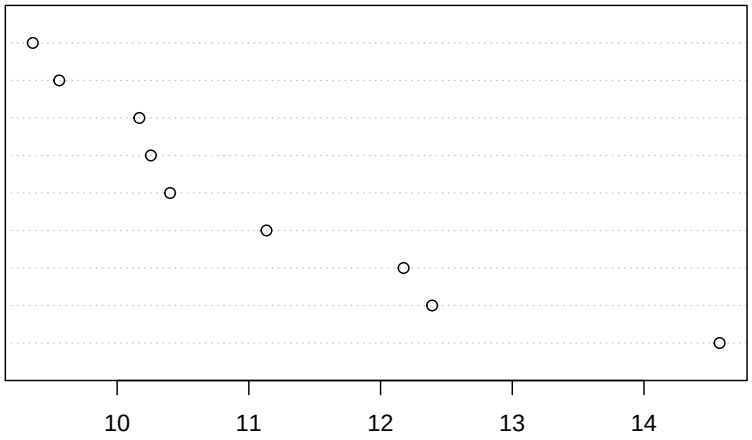
(b) Repeat the plot and associated labeling, now plotting area versus elevation, but specifying log="y" in order to obtain a logarithmic y-scale.

```
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(log2(area))
```



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

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

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