homework1

3170102187　　孙晨旭

1 Iowa dataset

```
iowa.df<-read.csv("~/github/Rcourse_1/data/Iowa.csv",sep = ';')
dim(iowa.df)  #rows and columns
```

```
## [1] 33 10
```

```
colnames(iowa.df)
```

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

```
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.

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

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

```
sort(vector1)
```

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

```
#sum(vector1) error
```

摘录自 Help 中的 R:Relational Operators:" Comparison of strings in character vectors is lexicographic within the strings using the collating sequence of the locale in use" 说明字符串依照字典序进行比较，因此排序和取最大值是合理的，而 sum 函数不允许字符串输入因此报错。

b.

```
vector2 <- c("5",7,12)
class(vector2)
```

```
## [1] "character"
```

```
#vector2[2] + vector2[3] error
```

向量结构中元素拥有相同数据类型，后续输入的 7 和 12 被转化为字符串类型，因此不可做加法运算。

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

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

dataframe 结构中元素类型任意，因此输入的数字可以做加法。

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

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

```
#list4[2]+list4[4] error
class(list4[2])
```

```
## [1] "list"
```

list 中的元素由两个方括号提取，使用单个方括号则返回一个单元素的 list。

3 Working with functions and operators

a.

```
seq(1,10000,372) # in increments of 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) #exactly 50 numbers
```

```
##  [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
```

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

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

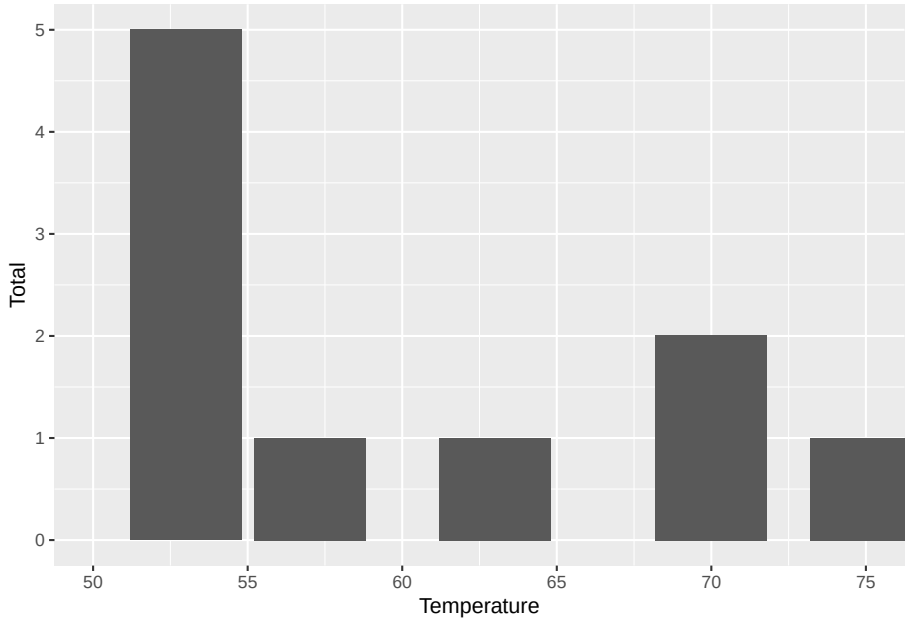
```
rep(1:3,each = 3)
```

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

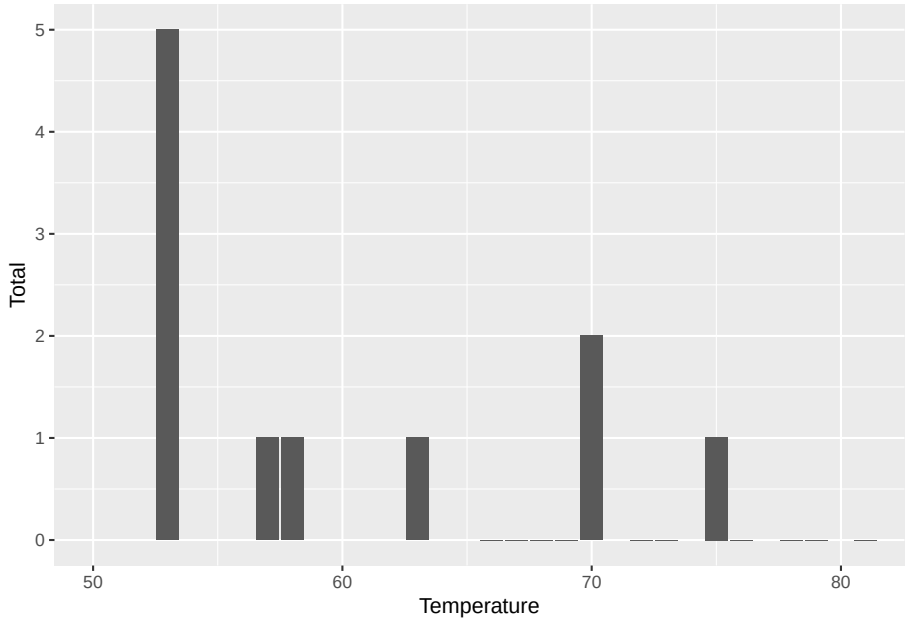
times 参数控制一个向量整体重复循环，each 参数控制向量中每个元素依次重复循环。

4 MB.Ch 1.2.

```
orings_d <- orings[c(1,2,4,11,13,18),]
ggplot(data = orings_d,aes(x = Temperature,y = Total)) +
  geom_col() +
  coord_cartesian(xlim = c(50,75))
```



```
ggplot(data = orings,aes(x = Temperature,y = Total)) +
  geom_col()+
  coord_cartesian(xlim = c(50,81))
```



5 MB.CH 1.4. ais data frame

```
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 ...
```

```
which(is.na(ais)) #no missing values
```

```
## integer(0)
```

```
ais.t <- table(ais$sex,ais$sport)
which(ais.t[1,]/ais.t[2,]>2 |ais.t[1,]/ais.t[2,]<0.5)
```

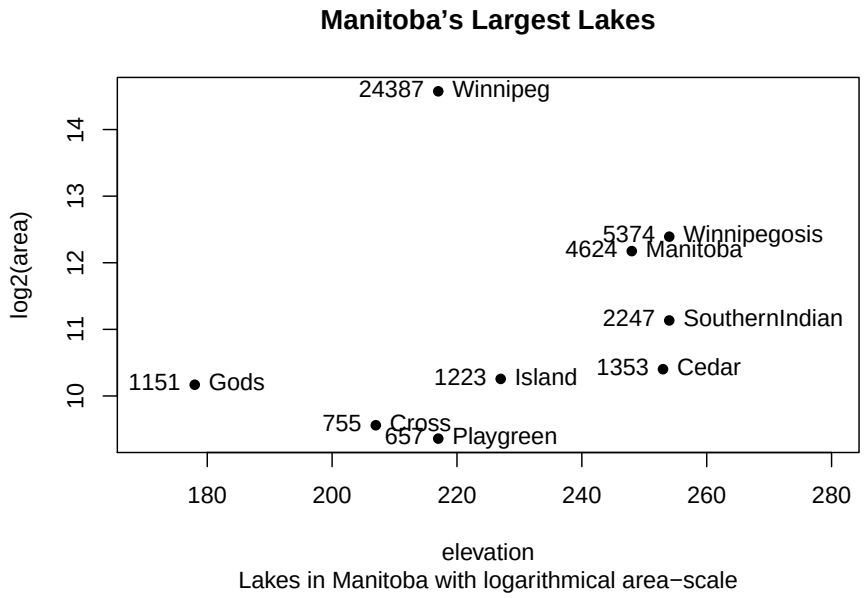
```
##     Gym Netball T_Sprnt  W_Polo
##       3       4       8      10
```

不平衡的项目即为上述四个（包含仅有一种性别的项目）。

6 MB.Ch 1.6. Manitoba.lakes data frame

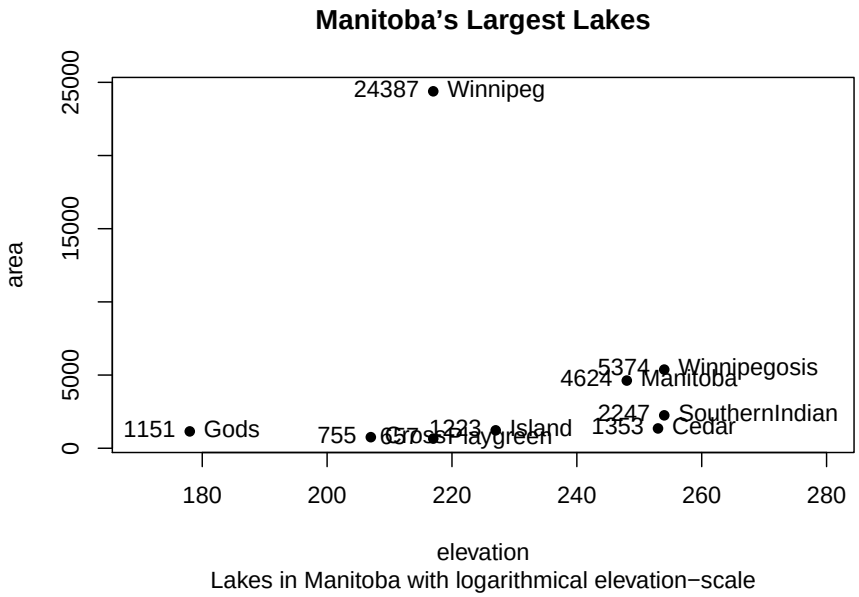
(a)

```
attach(Manitoba.lakes)
plot(log2(area) ~ elevation, pch=16, xlim=c(170,280))
text(log2(area) ~ elevation, labels=row.names(Manitoba.lakes), pos=4)
text(log2(area) ~ elevation, labels=area, pos=2)
title(main = "Manitoba' s Largest Lakes",sub = "Lakes in Manitoba with logarithmical ar
```



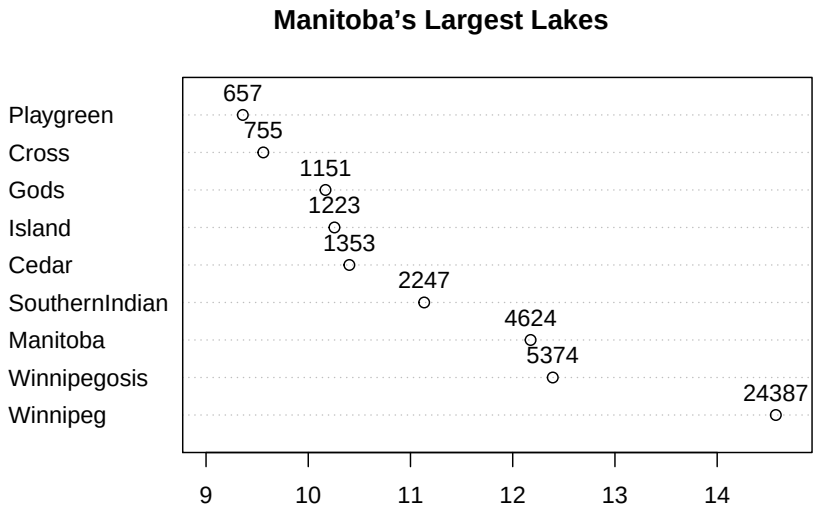
(b)

```
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(main = "Manitoba' s Largest Lakes",sub = "Lakes in Manitoba with logarithmical el
```



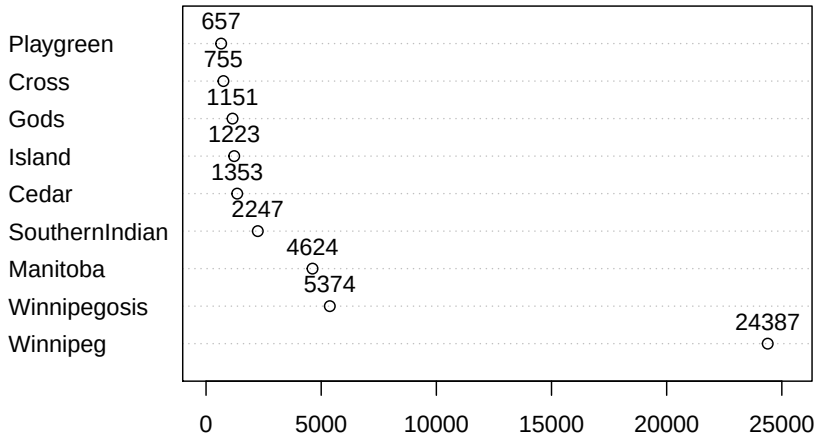
7 MB.Ch 1.7. Manitoba.lakes data frame

```
dotchart(log2(area),labels =  row.names(Manitoba.lakes),xlim = c(9,14.7))
text(1:9~log2(area), labels=area, pos=3)
title(main = "Manitoba' s Largest Lakes",sub = "Area of Each Lake in Manitoba with Loga
```



Area of Each Lake in Manitoba with Logarithmic scale

```
dotchart(area,labels =  row.names(Manitoba.lakes),xlim = c(0,25300))
text(1:9~area, labels=area, pos=3)
title(main = "Manitoba' s Largest Lakes",sub = "Area of Each Lake in Manitoba with line
```



Area of Each Lake in Manitoba with linear scale

MB.Ch 1.8. Manitoba.lakes data frame

```
sum(area)
```

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