Homework1

JinDian

2024-09-14

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 = ";")
iowa.df
```

```
##    Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield
## 1  1930 17.75  60.2  5.83  69.0  1.49  77.9  2.42  74.4  34.0
## 2  1931 14.76  57.5  3.83  75.0  2.72  77.2  3.30  72.6  32.9
## 3  1932 27.99  62.3  5.17  72.0  3.12  75.8  7.10  72.2  43.0
## 4  1933 16.76  60.5  1.64  77.8  3.45  76.4  3.01  70.5  40.0
## 5  1934 11.36  69.5  3.49  77.2  3.85  79.7  2.84  73.4  23.0
## 6  1935 22.71  55.0  7.00  65.9  3.35  79.4  2.42  73.6  38.4
## 7  1936 17.91  66.2  2.85  70.1  0.51  83.4  3.48  79.2  20.0
## 8  1937 23.31  61.8  3.80  69.0  2.63  75.9  3.99  77.8  44.6
## 9  1938 18.53  59.5  4.67  69.2  4.24  76.5  3.82  75.7  46.3
## 10 1939 18.56  66.4  5.32  71.4  3.15  76.2  4.72  70.7  52.2
## 11 1940 12.45  58.4  3.56  71.3  4.57  76.7  6.44  70.7  52.3
## 12 1941 16.05  66.0  6.20  70.0  2.24  75.1  1.94  75.1  51.0
## 13 1942 27.10  59.3  5.93  69.7  4.89  74.3  3.17  72.2  59.9
## 14 1943 19.05  57.5  6.16  71.6  4.56  75.4  5.07  74.0  54.7
## 15 1944 20.79  64.6  5.88  71.7  3.73  72.6  5.88  71.8  52.0
## 16 1945 21.88  55.1  4.70  64.1  2.96  72.1  3.43  72.5  43.5
## 17 1946 20.02  56.5  6.41  69.8  2.45  73.8  3.56  68.9  56.7
## 18 1947 23.17  55.6 10.39  66.3  1.72  72.8  1.49  80.6  30.5
## 19 1948 19.15  59.2  3.42  68.6  4.14  75.0  2.54  73.9  60.5
## 20 1949 18.28  63.5  5.51  72.4  3.47  76.2  2.34  73.0  46.1
## 21 1950 18.45  59.8  5.70  68.4  4.65  69.7  2.39  67.7  48.2
## 22 1951 22.00  62.2  6.11  65.2  4.45  72.1  6.21  70.5  43.1
## 23 1952 19.05  59.6  5.40  74.2  3.84  74.7  4.78  70.0  62.2
## 24 1953 15.67  60.0  5.31  73.2  3.28  74.6  2.33  73.2  52.9
## 25 1954 15.92  55.6  6.36  72.9  1.79  77.4  7.10  72.1  53.9
## 26 1955 16.75  63.6  3.07  67.2  3.29  79.8  1.79  77.2  48.4
## 27 1956 12.34  62.4  2.56  74.7  4.51  72.7  4.42  73.0  52.8
## 28 1957 15.82  59.0  4.84  68.9  3.54  77.9  3.76  72.9  62.1
## 29 1958 15.24  62.5  3.80  66.4  7.55  70.5  2.55  73.0  66.0
## 30 1959 21.72  62.8  4.11  71.5  2.29  72.3  4.92  76.3  64.2
## 31 1960 25.08  59.7  4.43  67.4  2.76  72.6  5.36  73.2  63.2
## 32 1961 17.79  57.4  3.36  69.4  5.51  72.6  3.04  72.4  75.4
## 33 1962 26.61  66.6  3.12  69.1  6.27  71.6  4.31  72.5  76.0
```

b.33 rows,10 columns.

c.Year,Rain0,Temp1,Rain1,Temp2,Rain2,Temp3,Rain3,Temp4,Yield.

d.the value of row 5,column 7 of iowa.df is 79.7.

e.

```
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

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)

1.vector1是一个由字符”5” “12” “7” “32”组成的向量

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

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

2.max()函数输出最大值，由于vector1向量中都是字符，字符从左开始比较字符大小，分别比较5,1,7,3.显然最大的是”7”

```
max(vector1)
```

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

3.sort()函数默认对vector1向量进行递增排序，根据字符大小比较的规则，从小到大依次是”12” “32” “5” “7” 。

```
sort(vector1)
```

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

4.出现error,因为vector1向量全是字符类型，无法进行sum().

b. For the next series of commands, either explain their results, or why they should produce errors.

vector2 <- c(“5”,7,12) vector22 + 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]] list42+list4[4]

1.向量必须包含同一种类型的元素，这里由于同时包含字符和数字类型数据，所以会把数字类型元素变成字符类型元素

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

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

会出现error:二进列运算符中有非数值参数.原因是在给vector2赋值时，其第二个和第三个元素都变成了字符类型数据，所以不能进行加运算

2.

```
dataframe3 <- data.frame(z1="5",z2=7,z3=12)
dataframe3
```

```
##   z1 z2 z3
## 1  5  7 12
```

```
dataframe3[1,2] + dataframe3[1,3]
```

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

dataframe3是dataframe类型的数据，它的nrow=1,dataframe3[1,2] =7,dataframe3[1,3]=12，两个数加起来是19.

3.list4是lists类型数据，list4[2]和list4[[4]]分别表示list4的第2个和第4个元素，二者都是数字类型，可以进行相加计算.list42和list4[4]分别表示只包含第2个和第4个元素的新list，两个list不能直接进行相加运算。

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.

```
seq1 <- seq(1,10000,by = 372)
seq1
```

```
##  [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
```

```
seq2 <- seq(1,10000,length.out = 50)
seq2
```

```
##  [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. 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).

```
rep1 <- rep(1:3, times=3)
rep1
```

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

```
rep2 <- rep(1:3, each=3)
rep2
```

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

rep(1:3, times=3)表示把1:3一起重复3次，rep(1:3, each=3)表示把1:3分别重复3次.

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.

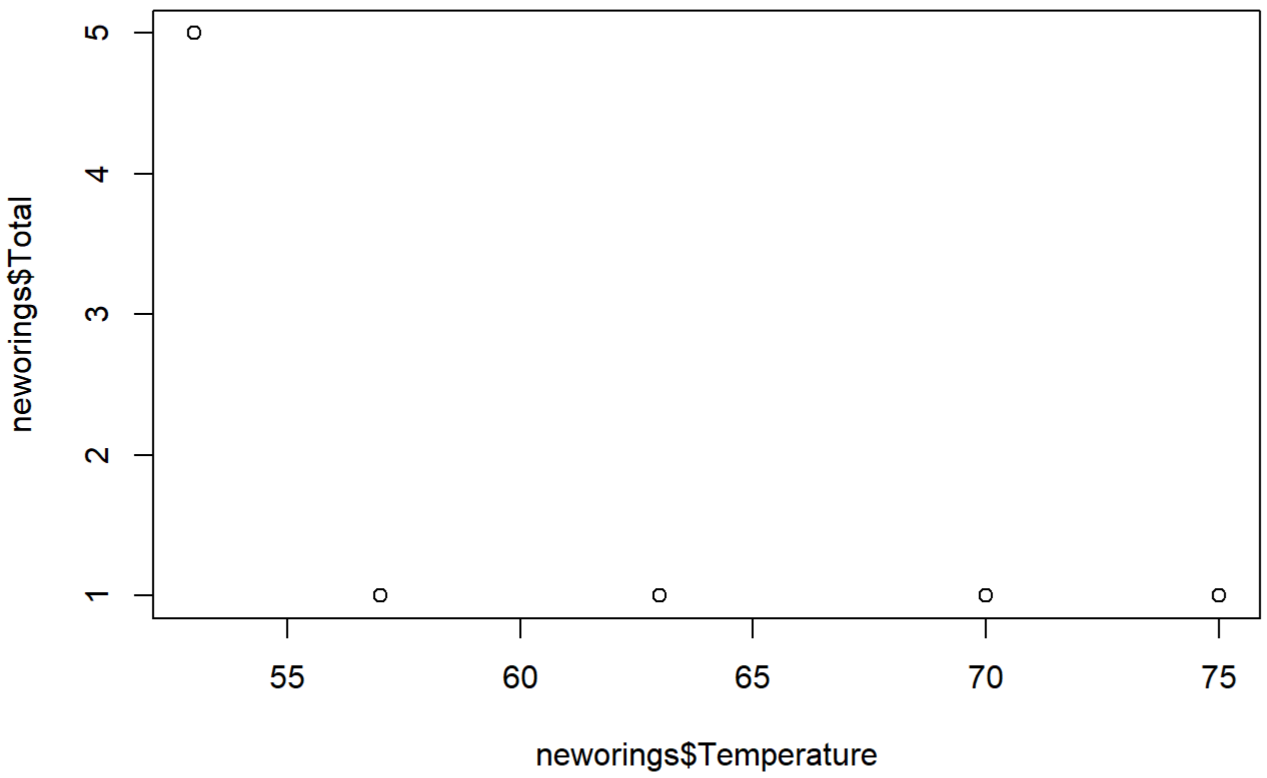
```
library(DAAG)
orings
```

```
##    Temperature Erosion Blowby Total
## 1           53       3      2     5
## 2           57       1      0     1
## 3           58       1      0     1
## 4           63       1      0     1
## 5           66       0      0     0
## 6           67       0      0     0
## 7           67       0      0     0
## 8           67       0      0     0
## 9           68       0      0     0
## 10          69       0      0     0
## 11          70       1      0     1
## 12          70       0      0     0
## 13          70       1      0     1
## 14          70       0      0     0
## 15          72       0      0     0
## 16          73       0      0     0
## 17          75       0      0     0
## 18          75       0      2     1
## 19          76       0      0     0
## 20          76       0      0     0
## 21          78       0      0     0
## 22          79       0      0     0
## 23          81       0      0     0
```

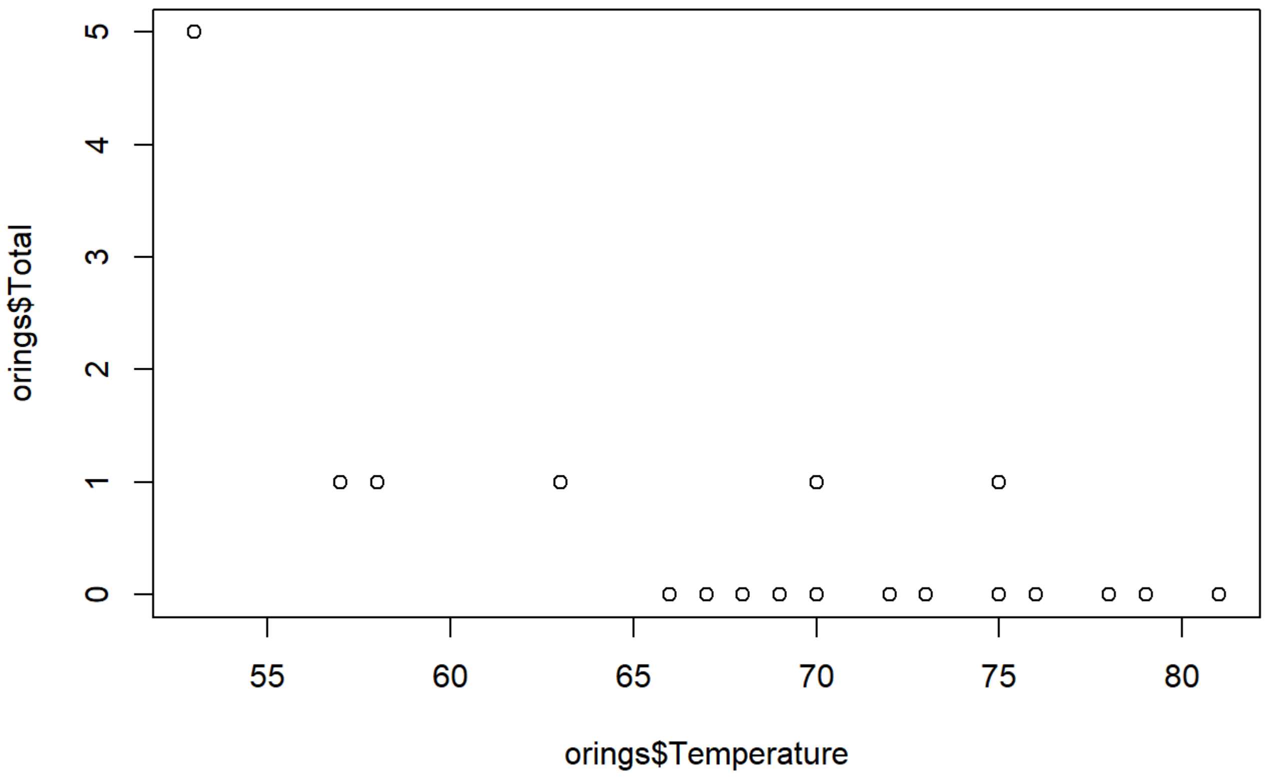
```
neworings <- orings[c(1,2,4,11,13,18),]
neworings
```

```
##    Temperature Erosion Blowby Total
## 1           53       3      2     5
## 2           57       1      0     1
## 4           63       1      0     1
## 11          70       1      0     1
## 13          70       1      0     1
## 18          75       0      2     1
```

```
plot(neworings$Total~neworings$Temperature)
```



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



##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. ###(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?

(a).没有缺失值

```
library(DAAG)

ais
```

```
##     rcc  wcc   hc   hg ferr   bmi   ssf pcBfat   lbm    ht   wt sex   sport
## 1  3.96  7.5 37.5 12.3   60 20.56 109.1  19.75 63.32 195.9 78.9   f  B_Ball
## 2  4.41  8.3 38.2 12.7   68 20.67 102.8  21.30 58.55 189.7 74.4   f  B_Ball
## 3  4.14  5.0 36.4 11.6   21 21.86 104.6  19.88 55.36 177.8 69.1   f  B_Ball
## 4  4.11  5.3 37.3 12.6   69 21.88 126.4  23.66 57.18 185.0 74.9   f  B_Ball
## 5  4.45  6.8 41.5 14.0   29 18.96  80.3  17.64 53.20 184.6 64.6   f  B_Ball
## 6  4.10  4.4 37.4 12.5   42 21.04  75.2  15.58 53.77 174.0 63.7   f  B_Ball
## 7  4.31  5.3 39.6 12.8   73 21.69  87.2  19.99 60.17 186.2 75.2   f  B_Ball
## 8  4.42  5.7 39.9 13.2   44 20.62  97.9  22.43 48.33 173.8 62.3   f  B_Ball
## 9  4.30  8.9 41.1 13.5   41 22.64  75.1  17.95 54.57 171.4 66.5   f  B_Ball
## 10 4.51  4.4 41.6 12.7   44 19.44  65.1  15.07 53.42 179.9 62.9   f  B_Ball
## 11 4.71  5.3 41.4 14.0   38 25.75 171.1  28.83 68.53 193.4 96.3   f  B_Ball
## 12 4.62  7.3 43.8 14.7   26 21.20  76.8  18.08 61.85 188.7 75.5   f  B_Ball
## 13 4.35  7.8 41.4 14.1   30 22.03 117.8  23.30 48.32 169.1 63.0   f  B_Ball
## 14 4.26  6.2 41.0 13.9   48 25.44  90.2  17.71 66.24 177.9 80.5   f     Row
## 15 4.63  6.0 43.7 14.7   30 22.63  97.2  18.77 57.92 177.5 71.3   f     Row
## 16 4.36  5.8 40.3 13.3   29 21.86  99.9  19.83 56.52 179.6 70.5   f     Row
## 17 3.91  7.3 37.6 12.9   43 22.27 125.9  25.16 54.78 181.3 73.2   f     Row
## 18 4.51  8.3 43.7 14.7   34 21.27  69.9  18.04 56.31 179.7 68.7   f     Row
## 19 4.37  8.1 41.8 14.3   53 23.47  98.0  21.79 62.96 185.2 80.5   f     Row
## 20 4.90  6.9 44.0 14.5   59 23.19  96.8  22.25 56.68 177.3 72.9   f     Row
## 21 4.46  5.7 39.2 13.0   43 23.17  80.3  16.25 62.39 179.3 74.5   f     Row
## 22 3.95  3.3 36.9 12.5   40 24.54  74.9  16.38 63.05 175.3 75.4   f     Row
## 23 4.46  9.5 41.5 14.5   92 22.96  83.0  19.35 56.05 174.0 69.5   f     Row
## 24 5.02  6.4 44.8 15.2   48 19.76  91.0  19.20 53.65 183.3 66.4   f     Row
## 25 4.26  5.8 41.2 14.1   77 23.36  76.2  17.89 65.45 184.7 79.7   f     Row
## 26 4.46  5.6 41.1 14.3   71 22.67  52.6  12.20 64.62 180.2 73.6   f     Row
## 27 4.16  5.8 39.8 13.3   37 24.24 111.1  23.70 60.05 180.2 78.7   f     Row
## 28 4.49  7.6 41.8 14.4   71 24.21 110.7  24.69 56.48 176.0 75.0   f     Row
## 29 4.21  7.5 38.4 13.2   73 20.46  74.7  16.58 41.54 156.0 49.8   f     Row
## 30 4.57  6.6 42.8 14.5   85 20.81 113.5  21.47 52.78 179.7 67.2   f     Row
## 31 4.87  6.4 44.8 15.0   64 20.17  99.8  20.12 52.72 180.9 66.0   f     Row
## 32 4.44 10.1 42.7 14.0   19 23.06  80.3  17.51 61.29 179.5 74.3   f     Row
## 33 4.45  6.6 42.6 14.1   39 24.40 109.5  23.70 59.59 178.9 78.1   f     Row
## 34 4.41  5.9 41.1 13.5   41 23.97 123.6  22.39 61.70 182.1 79.5   f     Row
## 35 4.87  7.3 44.1 14.8   13 22.62  91.2  20.43 62.46 186.3 78.5   f     Row
## 36 4.56 13.3 42.2 13.6   20 19.16  49.0  11.29 53.14 176.8 59.9   f Netball
## 37 4.15  6.0 38.0 12.7   59 21.15 110.2  25.26 47.09 172.6 63.0   f Netball
## 38 4.16  7.6 37.5 12.3   22 21.40  89.0  19.39 53.44 176.0 66.3   f Netball
## 39 4.32  6.4 37.7 12.3   30 21.03  98.3  19.63 48.78 169.9 60.7   f Netball
## 40 4.06  5.8 38.7 12.8   78 21.77 122.1  23.11 56.05 183.0 72.9   f Netball
## 41 4.12  6.1 36.6 11.8   21 21.38  90.4  16.86 56.45 178.2 67.9   f Netball
## 42 4.17  5.0 37.4 12.7  109 21.47 106.9  21.32 53.11 177.3 67.5   f Netball
## 43 3.80  6.6 36.5 12.4  102 24.45 156.6  26.57 54.41 174.1 74.1   f Netball
## 44 3.96  5.5 36.3 12.4   71 22.63 101.1  17.93 55.97 173.6 68.2   f Netball
## 45 4.44  9.7 41.4 14.1   64 22.80 126.4  24.97 51.62 173.7 68.8   f Netball
## 46 4.27 10.6 37.7 12.5   68 23.58 114.0  22.62 58.27 178.7 75.3   f Netball
## 47 3.90  6.3 35.9 12.1   78 20.06  70.0  15.01 57.28 183.3 67.4   f Netball
## 48 4.02  9.1 37.7 12.7  107 23.01  77.0  18.14 57.30 174.4 70.0   f Netball
## 49 4.39  9.6 38.3 12.5   39 24.64 148.9  26.78 54.18 173.3 74.0   f Netball
## 50 4.52  5.1 38.8 13.1   58 18.26  80.1  17.22 42.96 168.6 51.9   f Netball
## 51 4.25 10.7 39.5 13.2  127 24.47 156.6  26.50 54.46 174.0 74.1   f Netball
## 52 4.46 10.9 39.7 13.7  102 23.99 115.9  23.01 57.20 176.0 74.3   f Netball
## 53 4.40  9.3 40.4 13.6   86 26.24 181.7  30.10 54.38 172.2 77.8   f Netball
## 54 4.83  8.4 41.8 13.4   40 20.04  71.6  13.93 57.58 182.7 66.9   f Netball
## 55 4.23  6.9 38.3 12.6   50 25.72 143.5  26.65 61.46 180.5 83.8   f Netball
## 56 4.24  8.4 37.6 12.5   58 25.64 200.8  35.52 53.46 179.8 82.9   f Netball
## 57 3.95  6.6 38.4 12.8   33 19.87  68.9  15.59 54.11 179.6 64.1   f Netball
## 58 4.03  8.5 37.7 13.0   51 23.35 103.6  19.61 55.35 171.7 68.8   f Netball
## 59  4.36  5.5 41.4 13.8   82 22.42  71.3  14.52  55.39 170.0  64.8   f    Swim
## 60  4.07  5.9 39.5 13.3   25 20.42  54.6  11.47  52.23 170.0  59.0   f    Swim
## 61  4.17  4.9 38.9 12.9   86 22.13  88.2  17.71  59.33 180.5  72.1   f    Swim
## 62  4.23  8.1 38.2 12.7   22 25.17  95.4  18.48  61.63 173.3  75.6   f    Swim
## 63  4.46  8.3 42.2 14.4   30 23.72  47.5  11.22  63.39 173.5  71.4   f    Swim
## 64  4.38  5.8 42.0 14.0   27 21.28  55.6  13.61  60.22 181.0  69.7   f    Swim
## 65  4.31  5.3 41.1 13.9   60 20.87  62.9  12.78  55.73 175.0  63.9   f    Swim
## 66  4.51  5.1 40.9 14.0  115 19.00  52.5  11.85  48.57 170.3  55.1   f    Swim
## 67  4.13  7.0 39.7 13.1  124 22.04  62.6  13.35  51.99 165.0  60.0   f    Swim
## 68  4.48  9.5 36.5 13.3   54 20.12  49.9  11.77  51.17 169.8  58.0   f   Field
## 69  5.31  9.5 47.1 15.9   29 21.35  57.9  11.07  57.54 174.1  64.7   f  T_400m
## 70  4.58  5.8 42.1 14.7  164 28.57 109.6  21.30  68.86 175.0  87.5   f   Field
## 71  4.81  6.8 42.7 15.3   50 26.95  98.5  20.10  63.04 171.1  78.9   f   Field
## 72  4.51  9.0 39.7 14.3   36 28.13 136.3  24.88  63.03 172.7  83.9   f   Field
## 73  4.77  7.1 40.6 14.6   40 26.85 103.6  19.26  66.85 175.6  82.8   f   Field
## 74  5.33  9.3 47.0 15.0   62 25.27 102.8  19.51  59.89 171.6  74.4   f   Field
## 75  4.75  7.5 43.8 15.2   90 31.93 131.9  23.01  72.98 172.3  94.8   f   Field
## 76  4.11  7.3 38.7 12.4   12 16.75  33.8   8.07  45.23 171.4  49.2   f  T_400m
## 77  4.76  7.6 42.9 13.4   36 19.54  43.5  11.05  55.06 178.0  61.9   f T_Sprnt
## 78  4.27  6.9 44.1 14.7   45 20.42  46.2  12.39  46.96 162.0  53.6   f  T_400m
## 79  4.44  6.1 42.6 13.9   43 22.76  73.9  15.95  53.54 167.3  63.7   f  T_400m
## 80  4.20  6.5 39.1 13.0   51 20.12  36.8   9.91  47.57 162.0  52.8   f  T_400m
## 81  4.71  6.9 43.5 13.8   22 22.35  67.0  16.20  54.63 170.8  65.2   f  T_400m
## 82  4.09  6.4 40.1 13.2   44 19.16  41.1   9.02  46.31 163.0  50.9   f  T_400m
## 83  4.24  6.6 38.2 12.6   26 20.77  59.4  14.26  49.13 166.1  57.3   f  T_400m
## 84  3.90  6.0 38.9 13.5   16 19.37  48.4  10.48  53.71 176.0  60.0   f  T_400m
## 85  4.82  7.6 43.2 14.4   58 22.37  50.0  11.64  53.11 163.9  60.1   f T_Sprnt
## 86  4.32  6.8 40.6 13.7   46 17.54  54.6  12.16  46.12 173.0  52.5   f  T_400m
## 87  4.77  7.2 43.3 14.8   43 19.06  42.3  10.53  53.41 177.0  59.7   f  T_400m
## 88  5.16  8.2 45.3 14.7   34 20.30  46.1  10.15  51.48 168.0  57.3   f T_Sprnt
## 89  4.97  7.8 44.7 14.2   41 20.15  46.3  10.74  53.20 172.0  59.6   f T_Sprnt
## 90  4.00  4.2 36.6 12.0   57 25.36 109.0  20.86  56.58 167.9  71.5   f  Tennis
## 91  4.40  4.0 40.8 13.9   73 22.12  98.1  19.64  56.01 177.5  69.7   f  Tennis
## 92  4.38  7.9 39.8 13.5   88 21.25  80.6  17.07  46.52 162.5  56.1   f  Tennis
## 93  4.08  6.6 37.8 12.1  182 20.53  68.3  15.31  51.75 172.5  61.1   f  Tennis
## 94  4.98  6.4 44.8 14.8   80 17.06  47.6  11.07  42.15 166.7  47.4   f  Tennis
## 95  5.16  7.2 44.3 14.5   88 18.29  61.9  12.92  48.76 175.0  56.0   f  Tennis
## 96  4.66  6.4 40.9 13.9  109 18.37  38.2   8.45  41.93 157.9  45.8   f  Tennis
## 97  4.19  9.0 39.0 13.4   69 18.93  43.5  10.16  42.95 158.9  47.8   f     Gym
## 98  4.53  5.0 40.7 14.0   41 17.79  56.8  12.55  38.30 156.9  43.8   f     Gym
## 99  4.09  4.9 36.0 12.5   66 17.05  41.6   9.10  34.36 148.9  37.8   f     Gym
## 100 4.42  6.4 42.8 14.5   63 20.31  58.9  13.46  39.03 149.0  45.1   f     Gym
## 101 5.13  7.1 46.8 15.9   34 22.46  44.5   8.47  61.00 172.7  67.0   m    Swim
## 102 4.83  7.6 45.2 15.2   97 23.88  41.8   7.68  69.00 176.5  74.4   m    Swim
## 103 5.09  4.7 46.6 15.9   55 23.68  33.7   6.16  74.00 183.0  79.3   m    Swim
## 104 5.17  4.1 44.9 15.0   76 23.15  50.9   8.56  80.00 194.4  87.5   m    Swim
## 105 5.11  6.7 46.1 15.6   93 22.32  40.5   6.86  78.00 193.4  83.5   m    Swim
## 106 5.03  7.1 45.1 15.2   46 24.02  51.2   9.40  71.00 180.2  78.0   m    Swim
## 107 5.32  6.0 47.5 16.3  155 23.29  54.4   9.17  71.00 183.0  78.0   m    Swim
## 108 4.75  8.6 45.5 15.2   99 25.11  52.3   8.54  78.00 184.0  85.0   m    Swim
## 109 5.34  6.6 48.6 16.5   35 22.81  57.0   9.20  77.00 192.7  84.7   m    Swim
## 110 4.87  4.8 44.9 15.4  124 26.25  65.3  11.72  81.00 187.2  92.0   m    Swim
## 111 5.33  5.2 47.8 16.1  176 21.38  52.0   8.44  66.00 183.9  72.3   m    Swim
## 112 4.81  6.2 45.2 15.3  107 22.52  42.7   7.19  77.00 192.0  83.0   m    Swim
## 113 4.32  4.3 41.6 14.0  177 26.73  35.2   6.46  91.00 190.4  96.9   m    Swim
## 114 4.87  8.2 43.8 15.0  130 23.57  49.2   9.00  78.00 190.7  85.7   m     Row
## 115 5.04  7.1 44.0 14.8   64 25.84  61.8  12.61  75.00 181.8  85.4   m     Row
## 116 4.40  5.3 42.5 14.5  109 24.06  46.5   9.03  78.00 188.3  85.3   m     Row
## 117 4.95  5.9 45.4 15.5  125 23.85  34.8   6.96  87.00 198.0  93.5   m     Row
## 118 4.78  9.3 43.0 14.7  150 25.09  60.2  10.05  78.00 186.0  86.8   m     Row
## 119 5.21  6.8 44.5 15.4 115 23.84  48.1  9.56  79.00 192.0  87.9   m     Row
## 120 5.22  8.4 47.5 16.2  89 25.31  44.5  9.36  79.00 185.6  87.2   m     Row
## 121 5.18  6.5 45.4 14.9  93 19.69  54.0 10.81  48.00 165.3  53.8   m     Row
## 122 5.40  6.8 49.5 17.3 183 26.07  44.7  8.61  82.00 185.6  89.8   m     Row
## 123 4.92  5.4 46.2 15.8  84 25.50  64.9  9.53  82.00 189.0  91.1   m     Row
## 124 5.24  7.5 46.5 15.5  70 23.69  43.8  7.42  82.00 193.4  88.6   m     Row
## 125 5.09 10.1 44.9 14.8 118 26.79  58.3  9.79  83.00 185.6  92.3   m     Row
## 126 4.83  5.0 43.8 15.1  61 25.61  52.8  8.97  88.00 194.6  97.0   m     Row
## 127 5.22  6.0 46.6 15.7  72 25.06  43.1  7.49  83.00 189.0  89.5   m     Row
## 128 4.71  8.0 45.5 15.6  91 24.93  78.0 11.95  78.00 188.1  88.2   m     Row
## 129 5.24  7.2 46.6 15.9  58 22.96  40.8  7.35  85.00 200.4  92.2   m  B_Ball
## 130 4.54  5.9 44.4 15.6  97 20.69  41.5  7.16  73.00 195.3  78.9   m  B_Ball
## 131 5.13  5.8 46.1 15.9 110 23.97  50.9  8.77  82.00 194.1  90.3   m  B_Ball
## 132 5.00  6.7 45.3 15.7  72 24.64  49.6  9.56  79.00 187.9  87.0   m  B_Ball
## 133 5.17  8.0 47.9 16.4  36 25.93  88.9 14.53  97.00 209.4 113.7   m  B_Ball
## 134 4.89  7.5 41.6 14.4  53 23.69  48.3  8.51  90.00 203.4  98.0   m  B_Ball
## 135 4.50  9.2 40.7 13.7  72 25.38  61.8 10.64  90.00 198.7 100.2   m  B_Ball
## 136 4.84  8.3 46.3 15.9  39 22.68  43.0  7.06  74.00 187.1  79.4   m  B_Ball
## 137 4.13  8.9 40.3 13.5  61 23.36  61.1  8.87  82.00 196.6  90.3   m  B_Ball
## 138 4.87  7.4 43.5 15.0  49 22.44  43.8  7.88  72.00 186.1  77.7   m  B_Ball
## 139 4.82  6.4 44.3 14.8  35 22.57  54.2  9.20  76.00 192.8  83.9   m  B_Ball
## 140 4.73  6.7 42.8 14.9   8 19.81  41.8  7.19  70.00 195.2  75.5   m  B_Ball
## 141 4.55  5.6 42.6 14.4 106 21.19  34.1  6.06  57.00 169.1  60.6   m  T_400m
## 142 4.71  7.2 43.6 14.0  32 20.39  30.5  5.63  67.00 186.6  71.0   m  T_400m
## 143 4.93  7.3 46.2 15.1  41 21.12  34.0  6.59  67.00 184.4  71.8   m  T_400m
## 144 5.21  7.5 47.5 16.5  20 21.89  46.7  9.50  70.00 187.3  76.8   m  T_400m
## 145 5.09  8.9 46.3 15.4  44 29.97  71.1 13.97  88.00 185.1 102.7   m   Field
## 146 5.11  9.6 48.2 16.7 103 27.39  65.9 11.66  83.00 185.5  94.2   m   Field
## 147 4.94  6.3 45.7 15.5  50 23.11  34.3  6.43  74.00 184.9  79.0   m   Field
## 148 4.87  6.3 45.8 16.1  41 21.75  34.6  6.99  62.00 175.0  66.6   m  T_400m
## 149 4.41  4.5 44.2 15.0 101 20.89  31.8  6.00  67.00 185.4  71.8   m  T_400m
## 150 4.86  3.9 44.9 15.4  73 22.83  34.5  6.56  70.00 181.0  74.8   m  T_400m
## 151 4.91  9.0 46.3 15.4  56 22.02  31.0  6.03  64.00 176.0  68.2   m  T_400m
## 152 4.93  7.3 45.2 15.8  74 20.07  32.6  6.33  58.00 176.2  62.3   m  T_400m
## 153 4.20  4.5 41.2 14.3  58 20.15  31.5  6.82  57.00 174.0  61.0   m  T_400m
## 154 5.10  6.1 45.3 14.9  87 21.24  32.6  6.20  73.00 191.0  77.5   m  T_400m
## 155 4.50  6.1 42.2 14.7 139 19.63  31.0  5.93  54.00 171.0  57.4   m  T_400m
## 156 4.89  5.8 45.5 15.6  82 23.58  28.0  5.80  67.00 174.0  71.4   m T_Sprnt
## 157 5.13  4.0 44.1 15.2  87 21.65  33.7  6.56  66.00 180.2  70.3   m T_Sprnt
## 158 4.88  4.3 45.6 15.5  80 25.17  30.3  6.76  75.00 178.5  80.2   m T_Sprnt
## 159 5.00  8.2 46.8 14.7  67 23.25  38.0  7.22  78.00 190.3  84.2   m   Field
## 160 5.48  4.6 49.4 18.0 132 32.52  55.7  8.51 102.00 185.0 111.3   m   Field
## 161 5.93  6.4 49.1 16.1  43 22.59  37.5  7.72  74.00 189.0  80.7   m   Field
## 162 5.01  8.9 46.0 15.9 212 30.18 112.5 19.94  78.00 180.1  97.9   m   Field
## 163 5.48  6.2 48.2 16.3  94 34.42  82.7 13.91 106.00 189.2 123.2   m   Field
## 164 5.16  8.4 44.4 15.5 213 21.86  29.7  6.10  68.00 182.6  72.9   m T_Sprnt
## 165 4.64  9.0 42.9 14.9 122 23.99  38.9  7.52  77.00 186.0  83.0   m T_Sprnt
## 166 6.72  7.1 59.7 19.2  76 24.81  44.8  9.56  69.00 174.9  75.9   m T_Sprnt
## 167 4.83  6.6 43.8 14.3  53 21.68  30.9  6.06  66.00 180.6  70.7   m  T_400m
## 168 5.34  7.6 48.3 16.2  91 21.04  44.0  7.35  62.00 178.6  67.1   m  T_400m
## 169 5.13  4.6 45.3 16.8  36 23.12  37.5  6.00  65.00 173.0  69.2   m  T_400m
## 170 4.68  4.8 43.0 14.8 101 20.76  37.6  6.92  62.00 179.7  67.1   m  T_400m
## 171 5.00  5.2 45.1 15.1 184 23.13  31.7  6.33  66.00 174.6  70.5   m T_Sprnt
## 172 4.99  7.2 41.4 14.9  44 22.35  36.6  5.90  67.00 178.0  70.8   m T_Sprnt
## 173 5.49  5.9 47.7 15.9  66 22.28  48.0  8.84  65.00 178.5  71.0   m  T_400m
## 174 5.59  7.9 49.7 17.2 220 23.55  41.9  8.94  63.00 171.3  69.1   m T_Sprnt
## 175 5.03  6.6 44.7 15.9 191 19.85  30.9  6.53  59.00 178.0  62.9   m  T_400m
## 176 5.50  6.4 48.1 16.5  40 26.51  52.8  9.40  86.00 189.1  94.8   m T_Sprnt
## 177 5.11  9.3 45.4 15.8 189 24.78  43.2  8.18  87.00 195.4  94.6   m   Field
## 178 4.96  8.3 45.3 15.7 141 33.73 113.5 17.41  89.00 179.1 108.2   m   Field
## 179 5.01  8.9 46.0 15.9  212 30.18  96.9  18.08  80.00 180.1  97.9   m   Field
## 180 5.11  8.7 46.5 16.3   97 23.31  49.3   9.86  68.00 179.6  75.2   m   Field
## 181 5.69 10.8 50.5 18.5   53 24.51  42.3   7.29  69.00 174.7  74.8   m T_Sprnt
## 182 4.63  9.1 42.1 14.4  126 25.37  96.3  18.72  77.00 192.7  94.2   m  W_Polo
## 183 4.91 10.2 45.0 15.2  234 23.67  56.5  10.12  68.00 179.3  76.1   m  W_Polo
## 184 4.95  7.5 44.5 15.0   50 24.28 105.7  19.17  77.00 197.5  94.7   m  W_Polo
## 185 5.34 10.0 46.8 16.2   94 25.82 100.7  17.24  71.00 182.7  86.2   m  W_Polo
## 186 5.16 12.9 47.6 15.6  156 21.93  56.8   9.89  72.00 190.5  79.6   m  W_Polo
## 187 5.29 12.7 48.0 16.2  124 23.38  75.9  13.06  74.00 191.0  85.3   m  W_Polo
## 188 5.02  6.1 43.6 14.8   87 23.07  52.8   8.84  68.00 179.6  74.4   m  W_Polo
## 189 5.01  9.8 46.5 15.8   97 25.21  47.8   8.87  85.00 192.6  93.5   m  W_Polo
## 190 5.03  7.5 43.6 14.4  102 23.25  76.0  14.69  75.00 194.1  87.6   m  W_Polo
## 191 5.25  7.4 47.3 15.8   55 22.93  61.2   8.64  78.00 193.0  85.4   m  W_Polo
## 192 5.08  8.5 46.3 15.6  117 26.86  75.6  14.98  86.00 193.9 101.0   m  W_Polo
## 193 5.04  6.0 45.9 15.0   52 21.26  43.3   7.82  69.00 187.7  74.9   m  W_Polo
## 194 4.63 14.3 44.8 15.0  133 25.43  49.5   8.97  79.00 185.3  87.3   m  W_Polo
## 195 5.11  7.0 47.7 15.8  214 24.54  70.0  11.63  80.00 191.5  90.0   m  W_Polo
## 196 5.34  6.2 49.8 17.2  143 27.79  75.7  13.49  82.00 184.6  94.7   m  W_Polo
## 197 4.86  8.9 46.9 15.8   65 23.58  57.7  10.25  68.00 179.9  76.3   m  W_Polo
## 198 4.90  7.6 45.6 16.0   90 27.56  67.2  11.79  82.00 183.9  93.2   m  W_Polo
## 199 5.66  8.3 50.2 17.7   38 23.76  56.5  10.05  72.00 183.5  80.0   m  Tennis
## 200 5.03  6.4 42.7 14.3  122 22.01  47.6   8.51  68.00 183.1  73.8   m  Tennis
## 201 4.97  8.8 43.0 14.9  233 22.34  60.4  11.50  63.00 178.4  71.1   m  Tennis
## 202 5.38  6.3 46.0 15.7   32 21.07  34.9   6.26  72.00 190.8  76.7   m  Tennis
```

```
for(i in 1:13){
  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 ...
##  num [1:202] 7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...
##  num [1:202] 37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...
##  num [1:202] 12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...
##  num [1:202] 60 68 21 69 29 42 73 44 41 44 ...
##  num [1:202] 20.6 20.7 21.9 21.9 19 ...
##  num [1:202] 109.1 102.8 104.6 126.4 80.3 ...
##  num [1:202] 19.8 21.3 19.9 23.7 17.6 ...
##  num [1:202] 63.3 58.5 55.4 57.2 53.2 ...
##  num [1:202] 196 190 178 185 185 ...
##  num [1:202] 78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...
##  Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
##  Factor w/ 10 levels "B_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
missing_values <- colSums(is.na(ais))
missing_values
```

```
##    rcc    wcc     hc     hg   ferr    bmi    ssf pcBfat    lbm     ht     wt 
##      0      0      0      0      0      0      0      0      0      0      0 
##    sex  sport 
##      0      0
```

(b).根据计算，Gym、Netball、T_Sprnt、W_Polo这四项运动男女数量差距比较不平衡.

```
library(dplyr)
```

```
##
## 载入程序包：'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
  library(tidyr)
  library(tibble)
  library(DAAG)
  result <- ais |>  group_by(sex) |> count(sport,name = "Count") |>  pivot_wider(names_from = sport, value
s_from = Count, values_fill = 0) |> column_to_rownames(var = "sex")

  print(result)
```

```
##   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
```

```
  tresult <- (t(result))
  tresult <- as.data.frame(tresult)
  tresult
```

```
##          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
```

```
  imbalance <- filter(tresult,(f / m >= 2)|(f / m <=0.5))
  imbalance
```

```
##          f  m
## Gym      4  0
## Netball 23  0
## T_Sprnt  4 11
## W_Polo   0 17
```

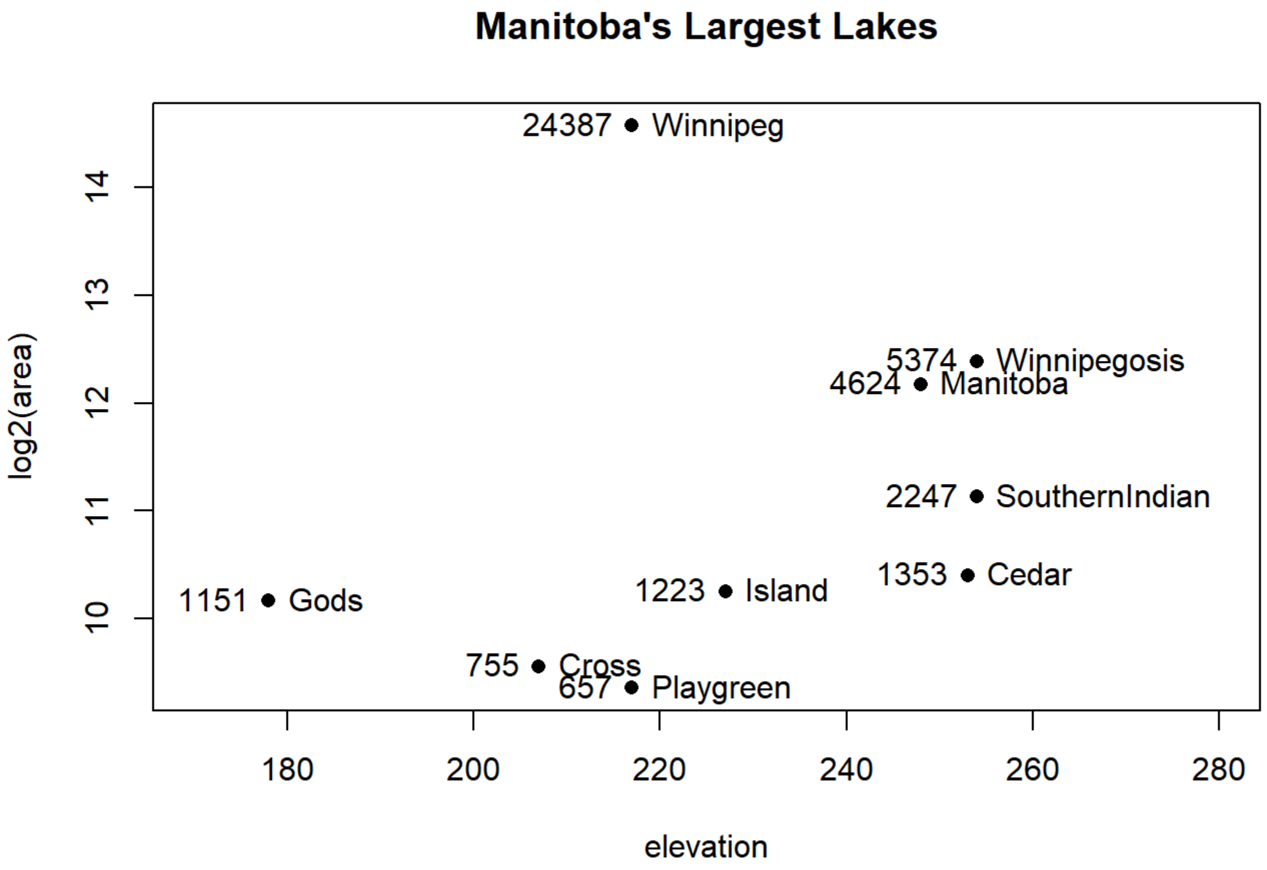
##MB.Ch1.6. ###Create a data frame called Manitoba.lakes that contains the lake's elevation (in meters above sea level) and area (in square kilometers) as listed below. Assign the names of the lakes using the row.names() function.

```
  Manitoba.lakes <- data.frame(elevation = c(217,254,248,254,253,227,178,207,217),area = c(24387,5374,462
4,2247,1353,1223,1151,755,657) )
  row.names(Manitoba.lakes) <-c("Winnipeg","Winnipegosis","Manitoba","SouthernIndian","Cedar","Island","Go
ds","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
```

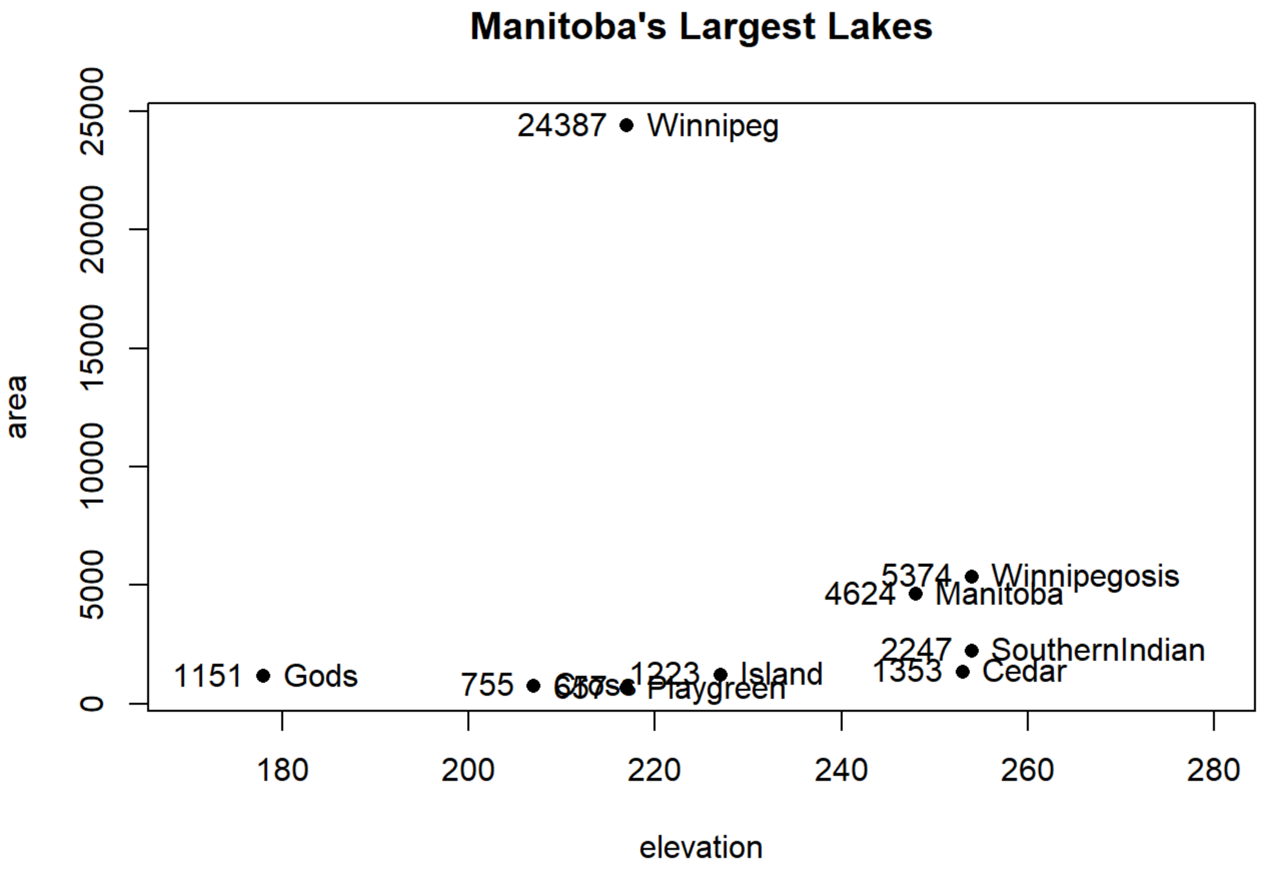
a. Use the following code to plot log2(area) versus elevation, adding labeling information (there is an extreme value of area that makes a logarithmic scale pretty much essential):

```
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")
```



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

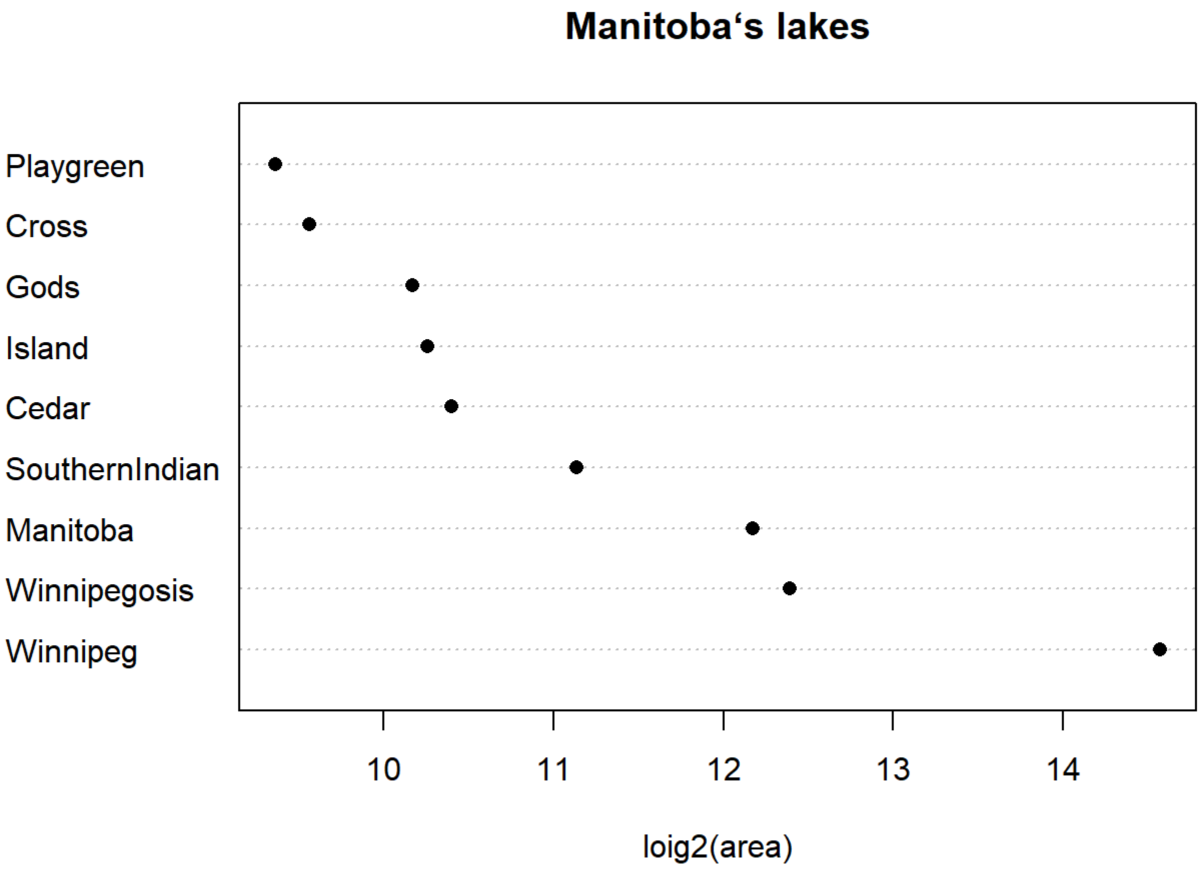
```
plot(area ~ elevation, pch=16, xlim=c(170,280), ylog=T)
text(area ~ elevation, labels=row.names(Manitoba.lakes), pos=4, ylog=T)
text(area ~ elevation, labels=area, pos=2, ylog=T)
title("Manitoba's Largest Lakes")
```



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

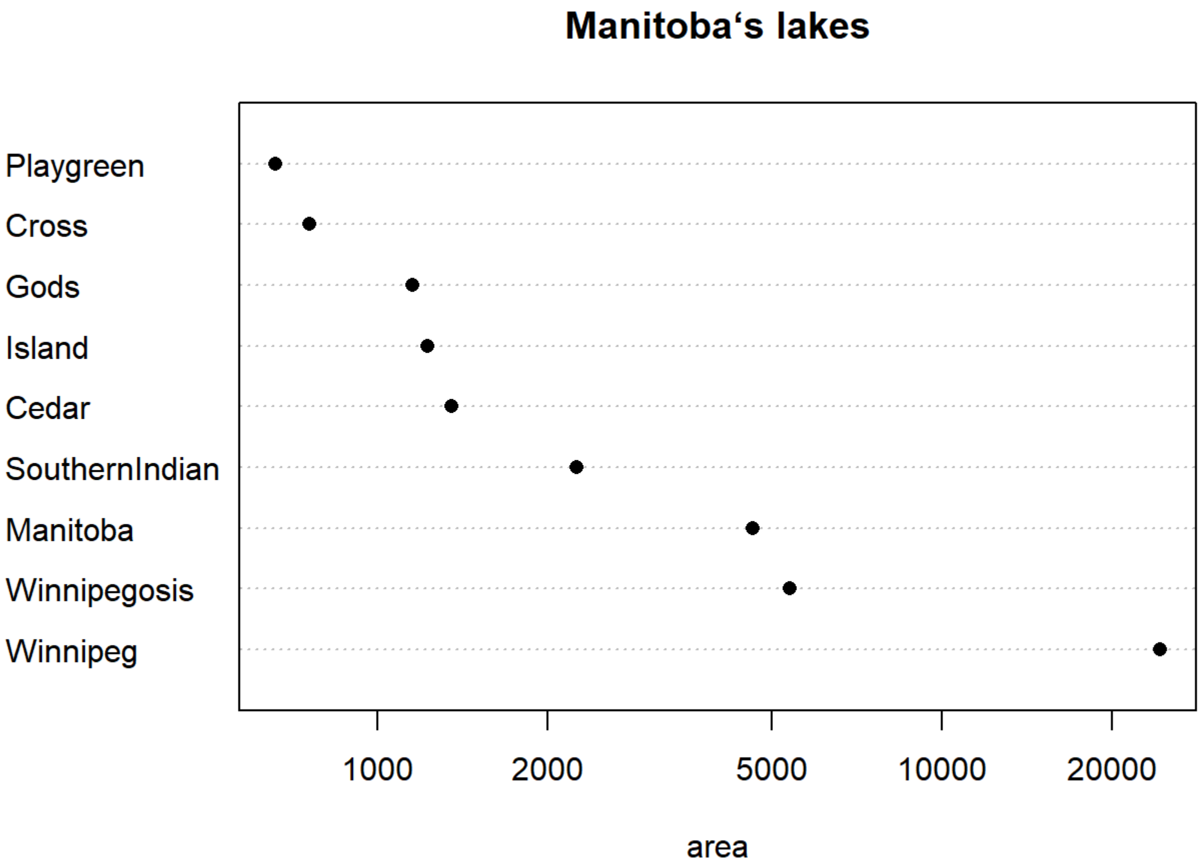
(a).

```
  dotchart(log2(area) ,labels = row.names(Manitoba.lakes),pch = 16,main = "Manitoba ‘s lakes",xlab ="loig2
(area)")
```



(b).

```
 dotchart(area ,labels = row.names(Manitoba.lakes),log = ″x″,pch = 16,main = ″Manitoba ‘s lakes″,xlab =″a
rea″)
```



##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
```