Homework 1 Solutions

3190300985 LUIS LUZERN YUVEN

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(DAAG)
```

Exercise 1

(a),(b)

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

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

`iowa.df` has 33 rows and 10 columns

(c),(d),(e)

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

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

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

```
## [1] 70.1
```

```
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
```

Exercise 2

(a)

`vector1 <- c("5", "12", "7", "32")` does not result in error, since the values in the sequence are all of the same type `max(vector1)` compares the first letter of the strings and returns the one with the largest ASCII value `sort(vector1)` command sorts the vector ascendingly according to the first letter of the strings

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

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

```
sort(vector1)
```

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

`sum(vector1)` results in an error, since there is no addition of strings in R

(b)

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

The first command is erroneous, since values assigned to a vector must be the same type

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

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

The `dataframe` function allows the different types of values in one dataframe. Since `dataframe3[1,2]` and `dataframe3[1,3]` are both numeric, addition is non-erroneous, but if, for example, we change `dataframe3[1,3]` to `dataframe3[1,1]`, then we will get an error, since there is no addition of strings and numerics

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

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

`list` allows values that are not necessarily of the same type. This first addition does not result in an error, since the use of ‘`[[]]` drops names and structures, and only takes the value.

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

This is an error, since `[]` does not drop name and structures

Exercise 3

(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(1,10000,10000/50)
```

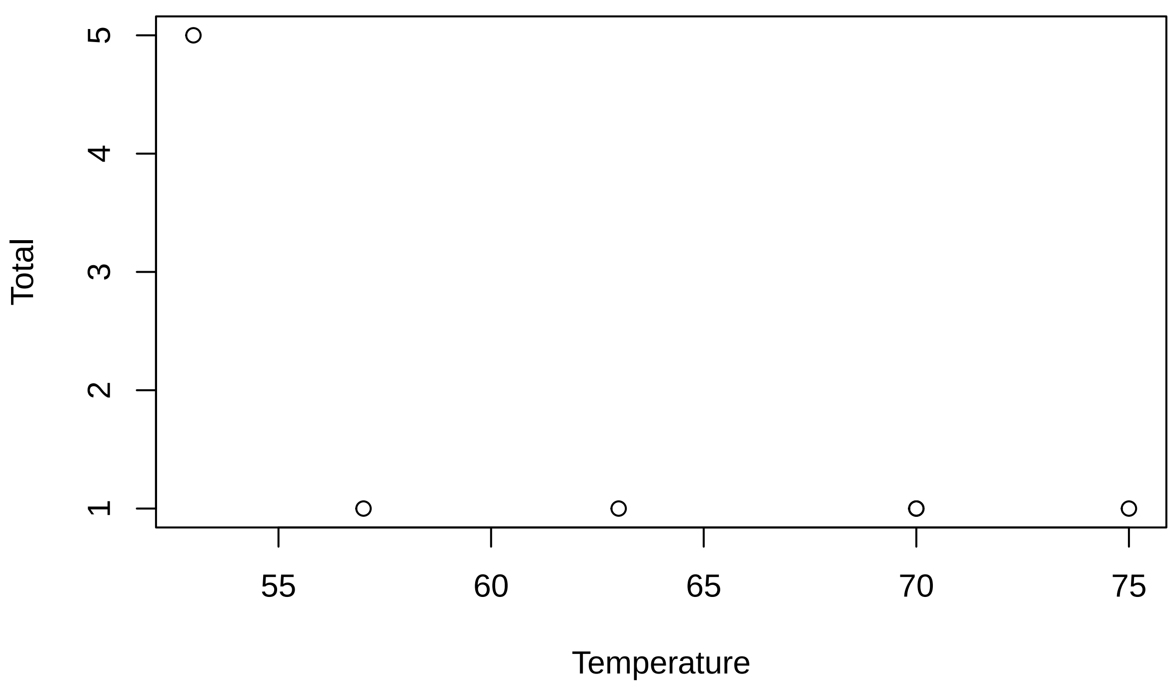
```
##  [1]    1  201  401  601  801 1001 1201 1401 1601 1801 2001 2201 2401 2601 2801
## [16] 3001 3201 3401 3601 3801 4001 4201 4401 4601 4801 5001 5201 5401 5601 5801
## [31] 6001 6201 6401 6601 6801 7001 7201 7401 7601 7801 8001 8201 8401 8601 8801
## [46] 9001 9201 9401 9601 9801
```

(b)

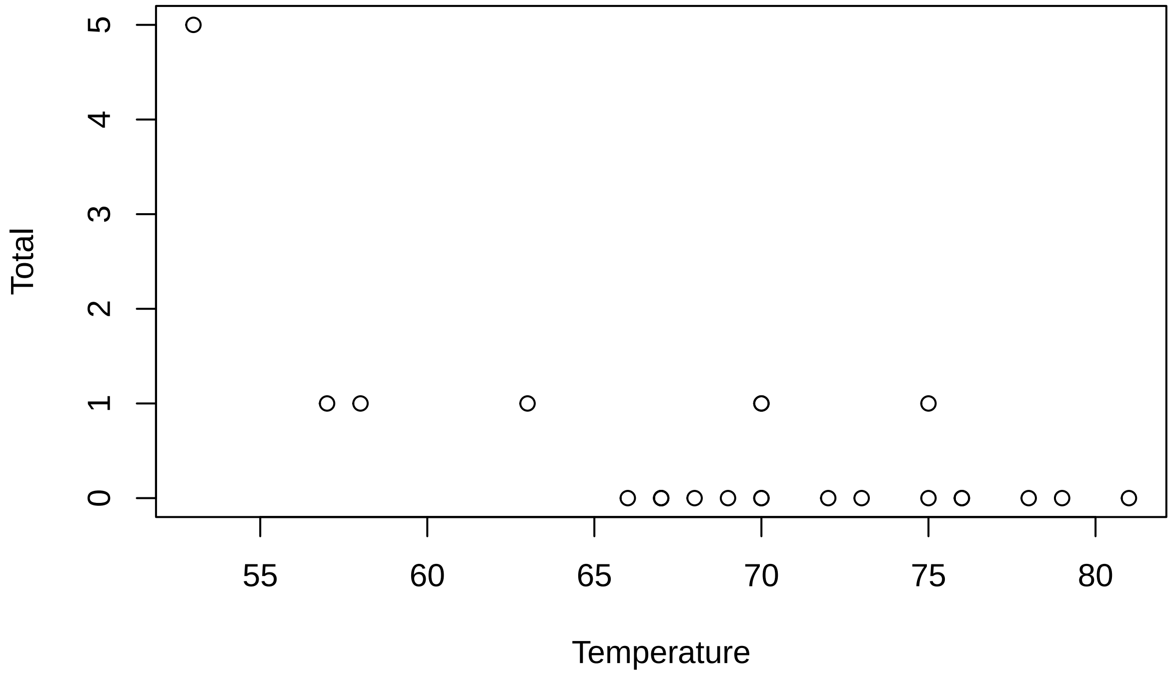
`rep(1:3, times=3)` repeats the sequence `1 2 3` three times, while `rep(1:3, each=3)` repeats each number consecutively for three times.

MB.Ch1.2.

```r
orings.new <- data.frame(orings[c(1:2,4,11,13,18),])
plot(Total ~ Temperature, data=orings.new)
```



```r
plot(Total ~ Temperature, data=orings)
```



MB.Ch1.4.

(a)

```
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 ...
```

None of the columns hold missing values

(b)

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

```
##    
##     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
```

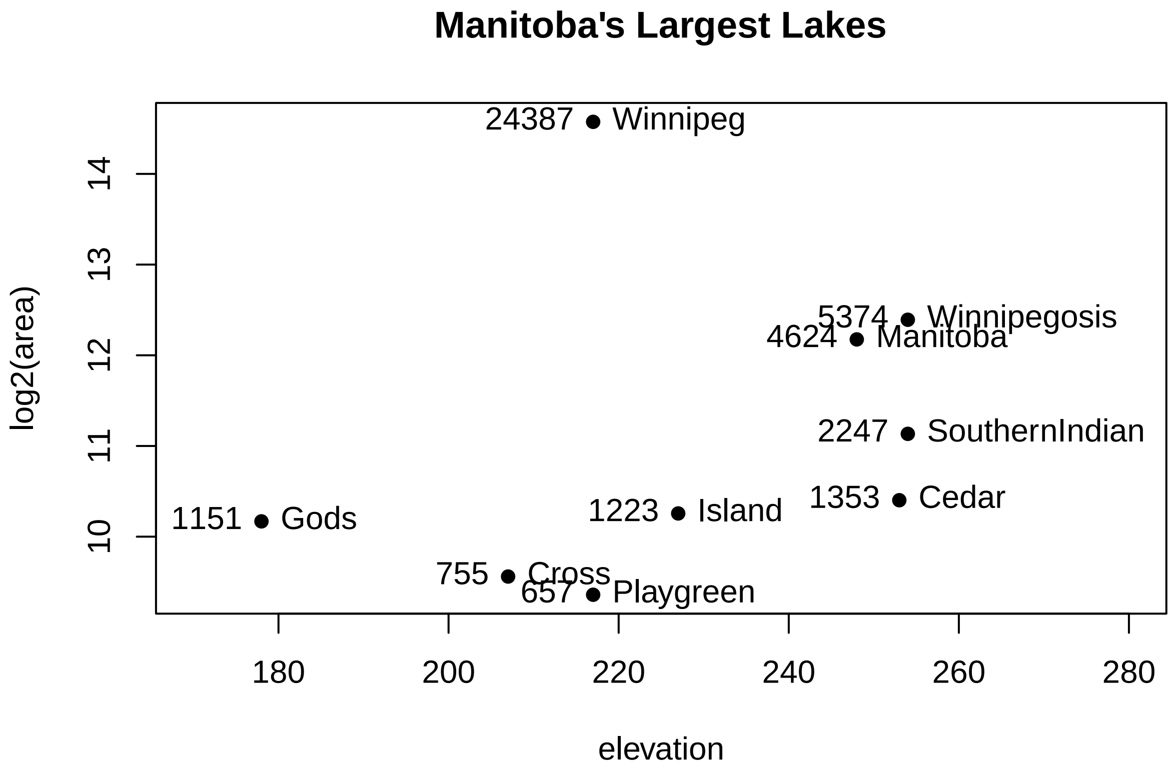
Large imbalance can be seen in `Gym`,`Netball`,`T_Sprnt`, and `W_Polo`

MB.Ch1.6.

```
lakes.matrix <- matrix(c(217,254,248,254,253,227,178,207,217,24387,5374,4624,2247,1353,
                         1223,1151,755,657), nrow=9)
Manitoba.lakes <- data.frame(lakes.matrix)
row.names(Manitoba.lakes) <- c("Winnipeg", "Winnipegosis", "Manitoba", "SouthernIndian",
                               "Cedar", "Island", "Gods", "Cross", "Playgreen")
colnames(Manitoba.lakes) <- c("elevation", "area")
```

(a)

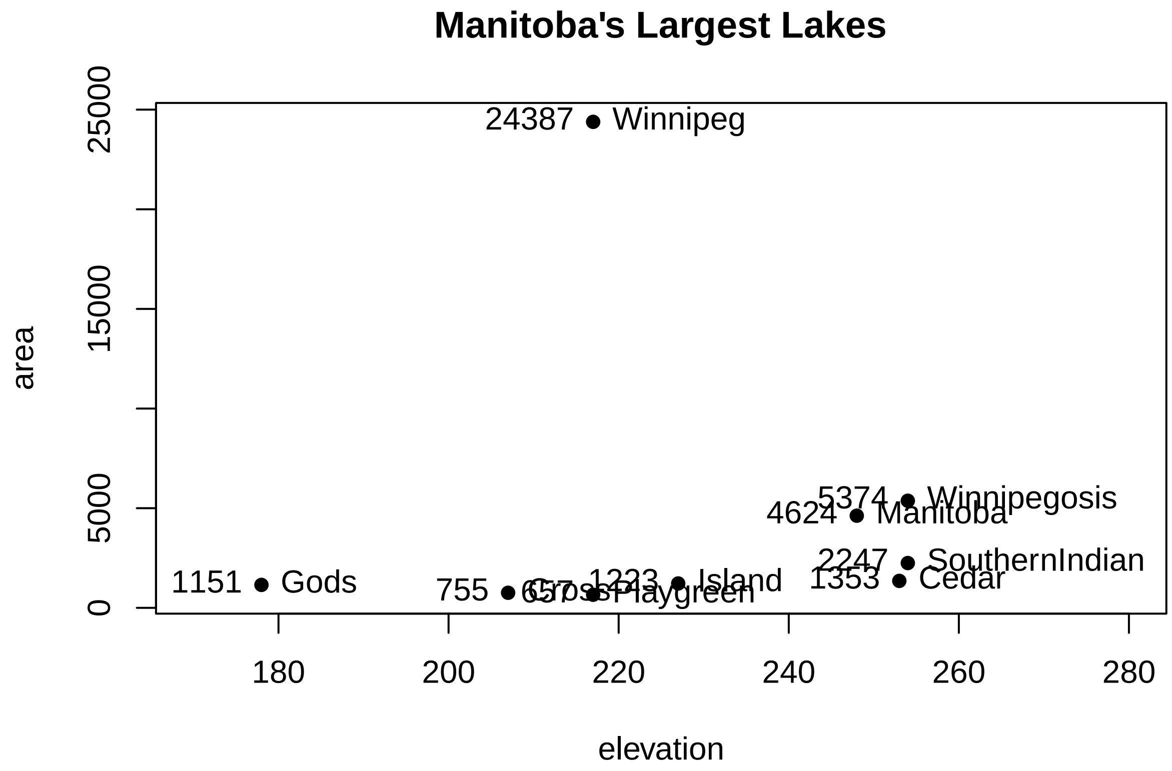
```
attach(Manitoba.lakes)
plot(log2(area) ~ elevation, pch=16, xlim=c(170,280))
# NB: Doubling the area increases log2(area) by 1.0
text(log2(area) ~ elevation, labels=row.names(Manitoba.lakes), pos=4)
text(log2(area) ~ elevation, labels=area, pos=2)
title("Manitoba's Largest Lakes")
```



The y-axis is the log-transformed area of lakes, and every label on the points indicate the name of lakes. The log-transformation makes the plot easier to observe as it is less-skewed.

(b)

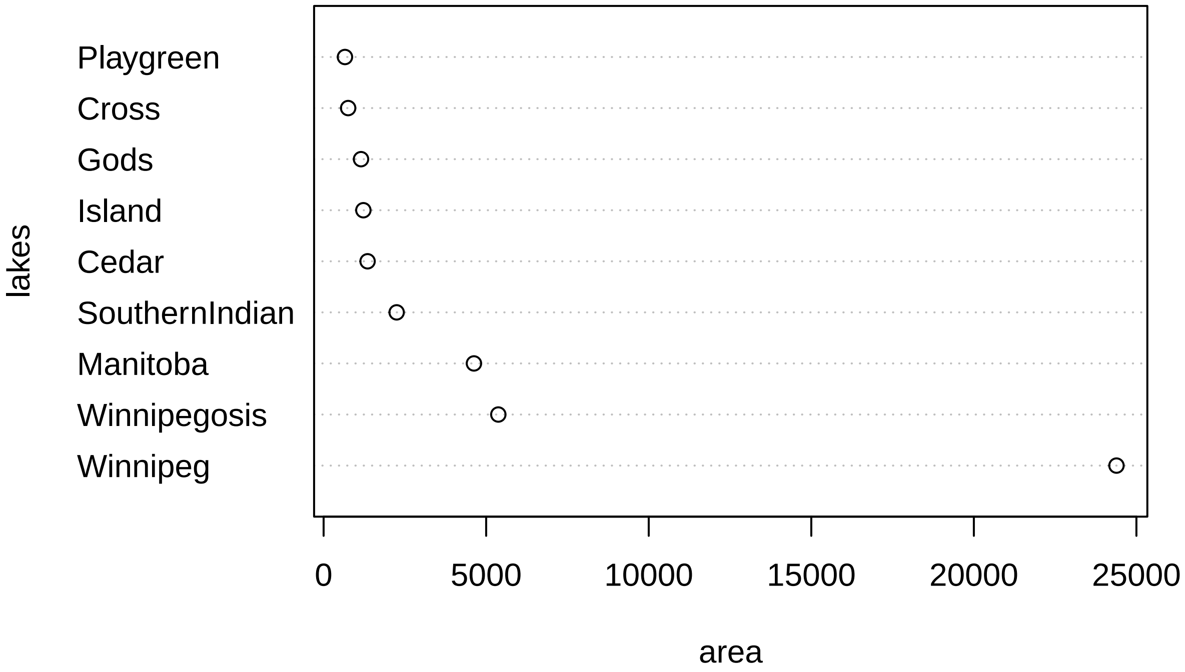
```
plot(area ~ elevation, pch=16, xlim=c(170,280), ylog=T)
text(area ~ elevation, labels=row.names(Manitoba.lakes), pos=4, ylog=T)
text(area ~ elevation, labels=area, pos=2, ylog=T)
title("Manitoba's Largest Lakes")
```



MB.Ch1.7.

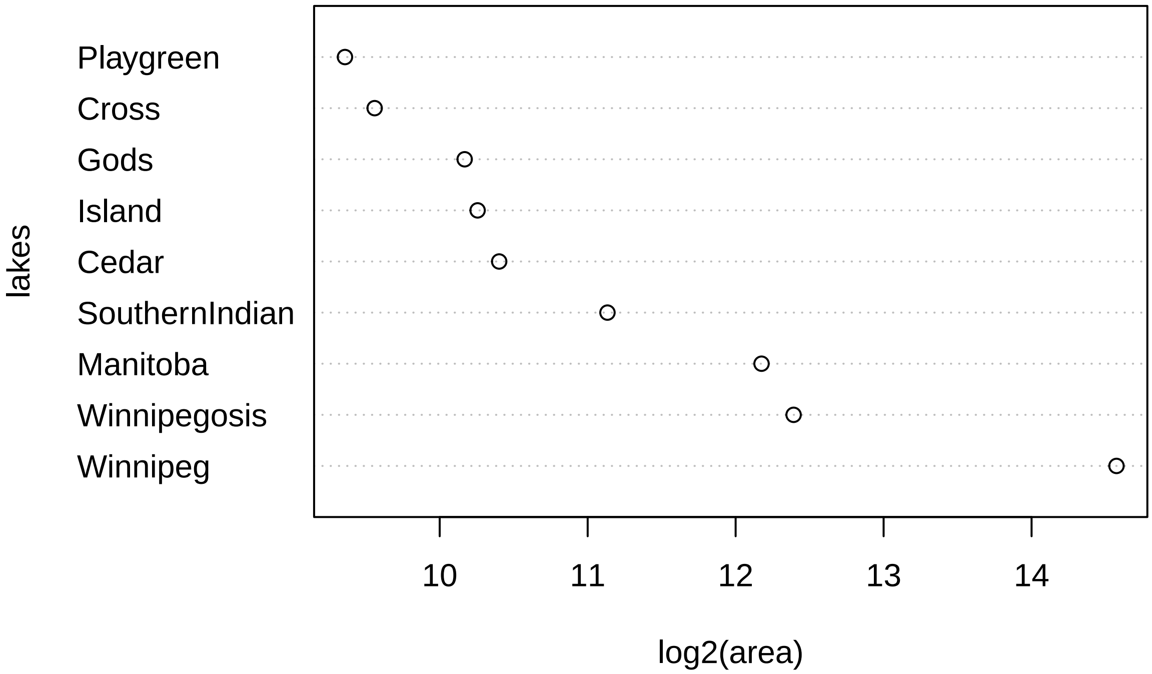
(a)

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



(b)

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



MB.Ch1.8.

```
sum(Manitoba.lakes$area)
```

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

At least 41,771 square kilometres of Manitoba are covered by water