

# Homework 01

Jinhai Chen,3170104884

06/07/2020

## Homework for July 6th

### Solution for Exercise 1

a.

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

b.

```
dim(iowa.df)
```

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

c.

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

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

d.

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

```
## [1] 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
```

### Solution for Exercise 2

a.

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

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

```
sort(vector1)
```

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

```
#sum(vector1)
```

The first computation can lead out to results, while the last two can not. The reason is that vector1 consists of 4 character variable, but the sort and sum can only sort or add up the numbers rather than the characters. The following codes will do.

```
vec1 <-c(5,12,7,32)
max(vec1)
```

```
## [1] 32
```

```
sort(vec1)
```

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

```
sum(vec1)
```

```
## [1] 56
```

b.

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

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

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

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

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

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

The first question is that the type of vector2 is

```
typeof(vector2)
```

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

as a result of which can not be added up. the summation can only work for numbers as follows:

```
vec2 <- c(5,7,12)
vec2[2] + vec2[3]
```

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

Second, function data.frame creates coupled collections of variables like lists or matrices, which in that way makes the summation available. Last, [] and [[]] are used to access elements in a list: [] gain access the name and the value, while [[]] gain access to only the number. So list4[[2]]+list4[[4]] returns a number of 168 while the other returns an error.

### Solution for Exercise 3

a.

```
seq(1,10000,by=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)
```

```
## [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.

```
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
```

The first one repeats the sequence from 1 to 3 for three times while the second one repeats each elements for three times inside the sequence.

## Solution for MB.Ch1.2

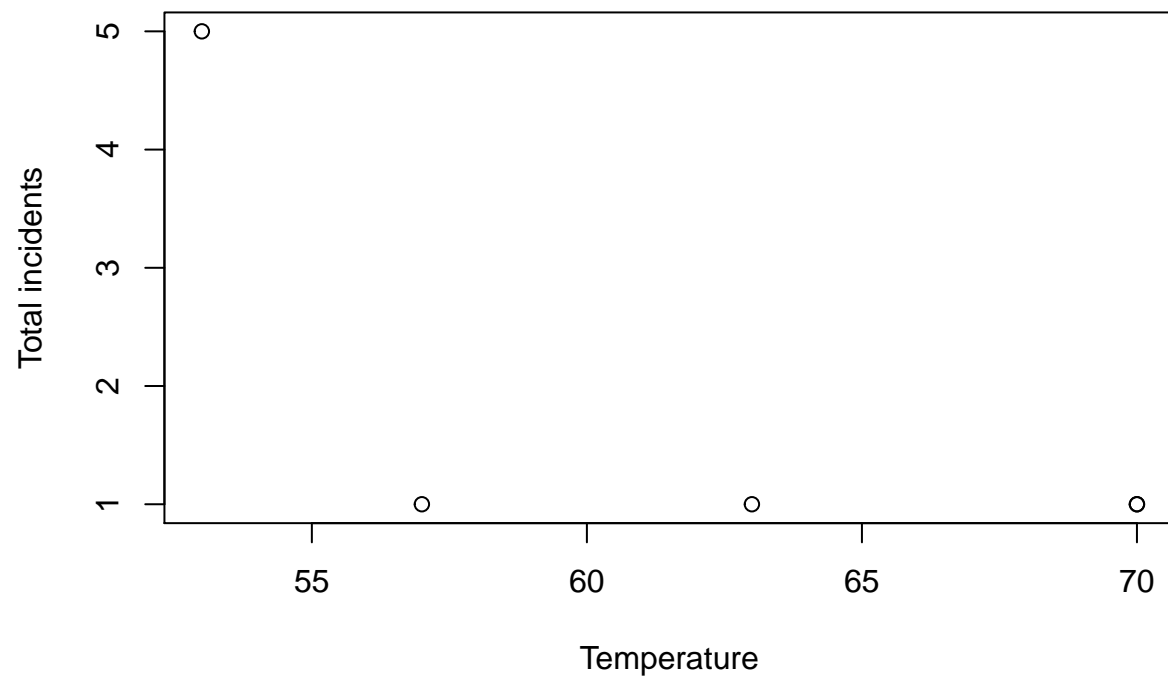
```
library(DAAG)
```

```
## Loading required package: lattice
```

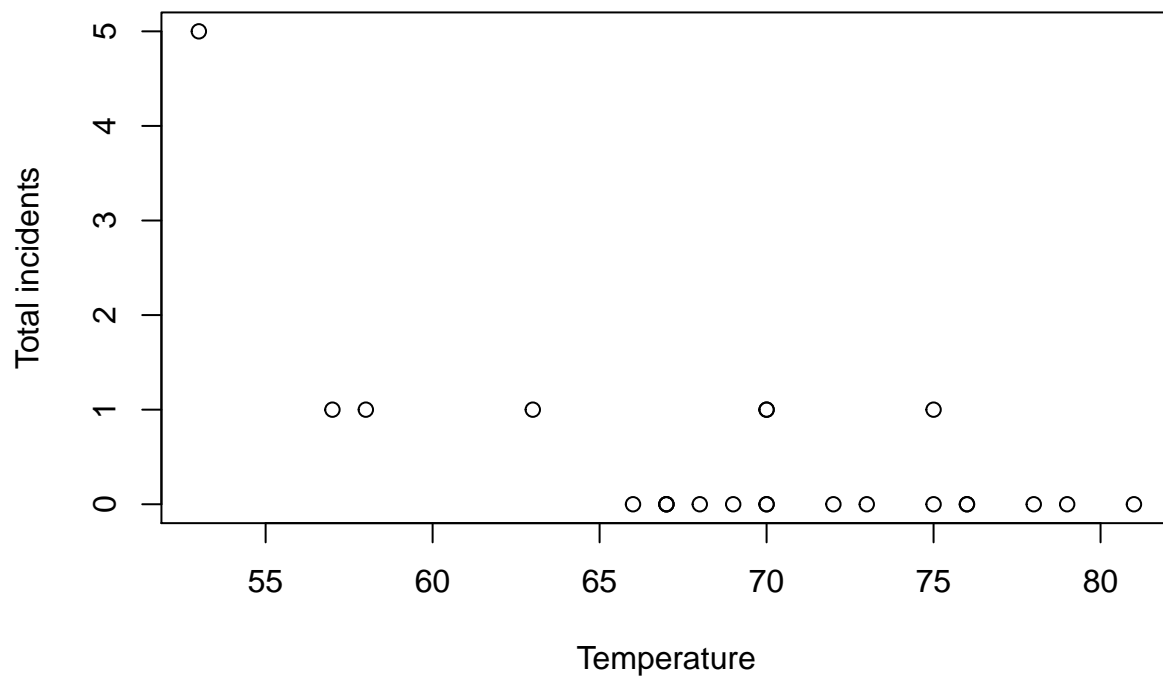
```
orings <- DAAG::orings
```

```
dataframe1 <- orings[c(1,2,4,11,13),]
```

```
plot(dataframe1[,1],dataframe1[,4],xlab = "Temperature",ylab = "Total incidents")
```



```
plot(orings[,1],orings[,4],xlab = "Temperature",ylab = "Total incidents")
```



#### Solution for MB.Ch1.4

a.

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