# Problem Set1

*Ankita Shankhdhar*

*Due 18th September 2015*

Question 1:

Part a: I loaded the file and set the header to False

```
rain.df=read.table("http://www.stats.uwo.ca/faculty/braun/data/rnf6080.dat",header=FALSE)
```

Part b-e: The dimensions, names, and values of the array are:

```
dim(rain.df)
```

```
## [1] 5070   27
```

```
names(rain.df)
```

```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11"
## [12] "V12" "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22"
## [23] "V23" "V24" "V25" "V26" "V27"
```

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

```
## [1] 0
```

```
rain.df[2,]
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
## 2 60  4  2  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
##   V21 V22 V23 V24 V25 V26 V27
## 2   0   0   0   0   0   0   0
```

Part f:

```
names(rain.df) <- c("year","month","day",seq(0,23))
```

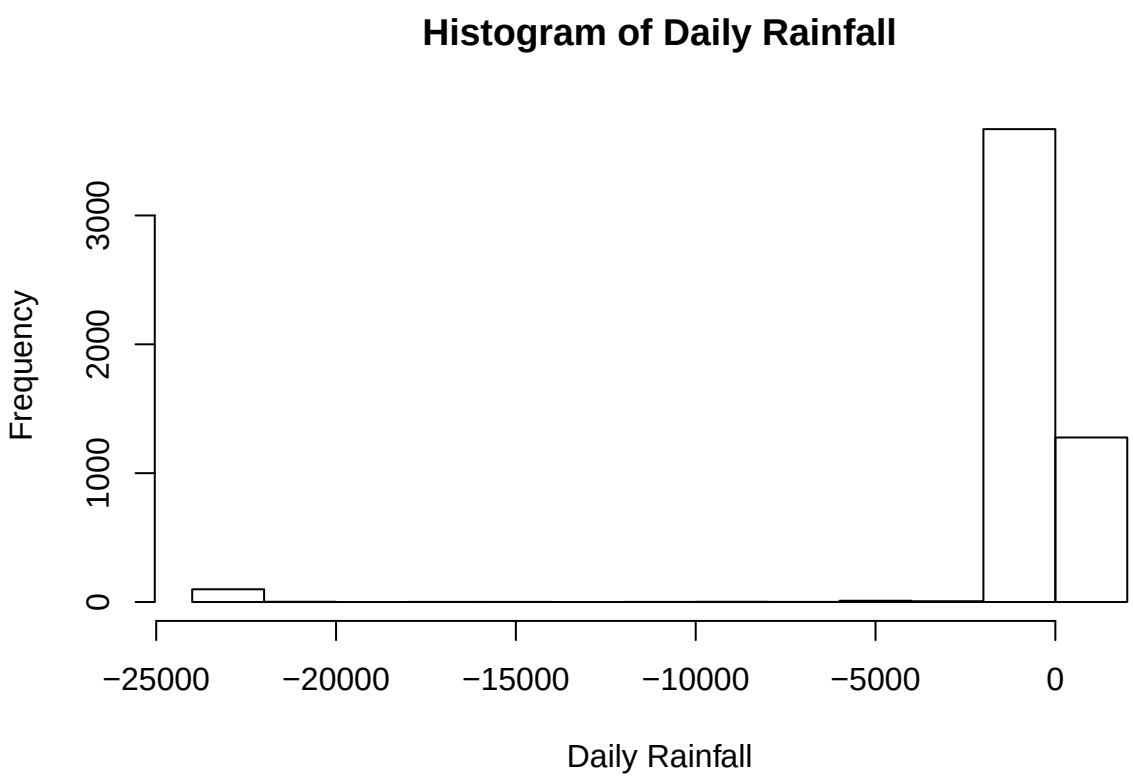
This line allows you to change the names of the first 3 columns to year, month and day respectively. The seq(0,23) tells the renamining columns to be named 0-23. The last 24 columns represent each hour of the day and the input represent how much rainfall happened at that hour.

Part g:

```
rain.df<-cbind(rain.df,c(rowSums(rain.df[,4:27])))
names(rain.df)[28]<-"daily"
```

This here creates a new row called daily which shows the sum of hours of rainfall during a day.

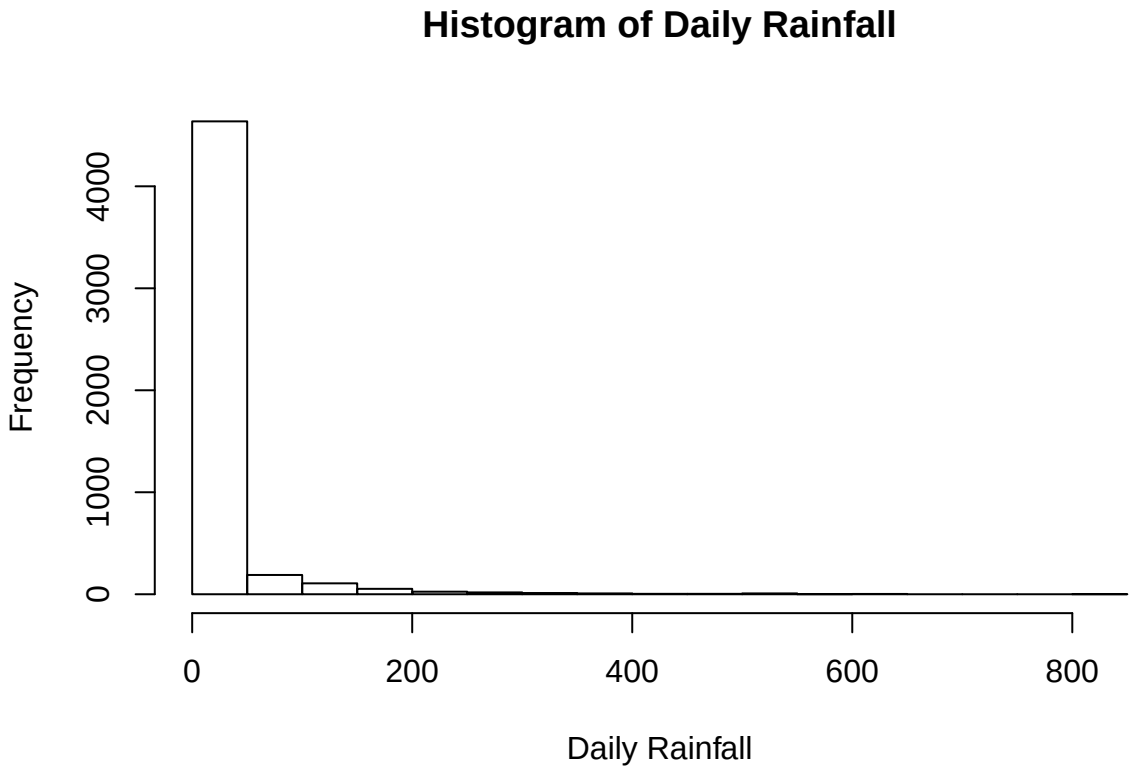
```r
hist(rain.df$daily,xlab="Daily Rainfall",main="Histogram of Daily Rainfall")
```




There are sums that are negative and you can not have negative rainfall in a day.

Part h:

```r
rain.mat <- as.matrix(rain.df)
rain.mat[rain.mat<0] <- 0
rain.df.fixed <- as.data.frame(rain.mat)
hist(rain.df.fixed$daily,xlab="Daily Rainfall",main="Histogram of Daily Rainfall")
```




Here, I first converted the data frame into a matrix and searched for all the values in that matrix that were negative. Those negatively values were then set to 0. After that I converted the matrix back into out main data frame and plotted those values. This makes more sense because you now have positive daily rainfall or no rainfall at all.

Question 2:

Part a:

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

The first line makes a vector of 4 elements that are characters. The maximum function is supposed to return 32 if the elements were numerics. Since they are character, R is lexographically sorting them and finding the maximum in the alphabetical form. The same is being done in sort(vector1). The sum function however only works on numeric values which results in R giving us an error.

Part b:

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

This should create a vector however it would be a vector of only characters. So it will not take 7,12 as numeric; instead it will convert them into characters. Since vector2 will be a vector of characters, we will not be able to do arithmetic operations on it. So the addition would lead us to an error.

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

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

In a data frame we can store various types. So for dataframe3 we can definitely call arimethic operations on numeric values, which is what line 2 is doing.

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

Lists can also store various types however the way you call them can make things a little different. So for the second line drops names and structures but the third lines does not. Instead it acts like a bector and since the first entry of the list is a character, it makes the whole vector into a character vector.

Question 3:

Part a:

```
seqvec1 <- c(seq(from=1,to=1000,by=372))
seqvec2 <- c(seq(from=1,to=1000,length.out=50))
```

The first creates a sequence from 1-100 in increments of 372. The second command creates a sequence from 1-1000 with a total length of 50.

Part b:

```
repvec1 <- c(rep(1:3,times=3))
repvec2 <- c(rep(1:3,each=3))
```

In the two cases above: rep(1:3,times=3) would repeat 1,2,3 3 times whereas rep(1:3,each=3) repeats “1” 3 times, then “2” 3 times and then “3” 3 times.