1. Generates regular sequences from 1 to 10000 at regular intervals of 372

```
> seq(from=1,to=10000,by=372)
 [1]    1  373  745 1117 1489 1861 2233 2605 2977 3349
[11] 3721 4093 4465 4837 5209 5581 5953 6325 6697 7069
[21] 7441 7813 8185 8557 8929 9301 9673
```

Generates regular sequences from 1 to 10000 with total length of the sequnce as 50

```
> seq(from=1,to=10000,length.out=50)
 [1]     1.0000   205.0612   409.1224   613.1837
 [5]   817.2449  1021.3061  1225.3673  1429.4286
 [9]  1633.4898  1837.5510  2041.6122  2245.6735
[13]  2449.7347  2653.7959  2857.8571  3061.9184
[17]  3265.9796  3470.0408  3674.1020  3878.1633
[21]  4082.2245  4286.2857  4490.3469  4694.4082
[25]  4898.4694  5102.5306  5306.5918  5510.6531
[29]  5714.7143  5918.7755  6122.8367  6326.8980
[33]  6530.9592  6735.0204  6939.0816  7143.1429
[37]  7347.2041  7551.2653  7755.3265  7959.3878
[41]  8163.4490  8367.5102  8571.5714  8775.6327
[45]  8979.6939  9183.7551  9387.8163  9591.8776
[49]  9795.9388 10000.0000
```

2.**rep-Replicate elements of vectors and lists**

```
> rep(1:3, times=3)
[1] 1 2 3 1 2 3 1 2 3
```

Repeats the sequnce 1 to 3 a total of three times

```
> rep(1:3, each=3)
[1] 1 1 1 2 2 2 3 3 3
```

Repeat the sequence such that each number of the sequnce is repeatde 3 times one after the other times

A integer vector giving the (non-negative) number of times to repeat each element if of length of length(x)

each non-negative integer. Each element of x is repeated each times.

3.

```
> vector1 <- c("5", "12", "7", "32")
> max(vector1)
[1] "7"
```

```
> sort(vector1)
[1] "12" "32" "5"  "7"
```

```
> sum(vector1)
Error in sum(vector1) : invalid 'type' (character) of argument
```

Arguments for sum() should be numeric or complex or logical vectors.

4.

```
> vector2 <- c("5",7,12)
> vector2[2] + vector2[3]
Error in vector2[2] + vector2[3] :
  non-numeric argument to binary operator quotation marks reserved for cha
racter variables. Remove “” in vector to get 19
```

```
> dataframe3 <- data.frame(z1="5",z2=7,z3=12)
> dataframe3[1,2] + dataframe3[1,3]
[1] 19
```

A data frame, a matrix-like structure whose columns may be of differing types (numeric, logical, character etc). Here we are taking and adding the inputs by [row,column] from our dataframe

```
> list4 <- list(z1="6", z2=42, z3="49", z4=126)
> list4[[2]]+list4[[4]]
[1] 168*
```

Returning the values from a list based on the index position within the list

```
> list4[2]+list4[4]
A list is a generic vector containing other objects
Error in list4[2] + list4[4] : non-numeric argument to binary operator
#This command makes little sense. You are calling and adding the actual va
riable name z1 z4, single bracket, without calling the vales within those
index positions 42 and 126, double brackets.
```

5.

a.

```
> rainfalldata<-read.table("http://www.stats.uwo.ca/faculty/braun/data/rnf
6080.dat")
> rainfall_dataframe<-data.frame(rainfalldata) #creating data frame
```

b.

```
> dim(rainfall_dataframe)
[1] 5070   27
```

Dim() displays the no of rows and columns in a dataframe.

c.

```
> colnames(rainfall_dataframe)
[1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11""V12"
[13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22""V23""V24"
[25] "V25" "V26" "V27
```

Colnames() will display column names of all the columns of the rainfall_dataframe

d.

```
> rainfall_dataframe[5,c('V7')]
[1] 0
```

Outputting the value corresponding to the 5th row and 7th column.

e.

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

Outputting all the values corresponding to the 2nd row of the rainfall_dataframe

f.

```
> names(rainfall_dataframe) <- c("year","month","day",seq(0,23))
> head(rainfall_dataframe)
  year month day 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
1   60     4   1 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2   60     4   2 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
3   60     4   3 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
4   60     4   4 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5   60     4   5 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6   60     4   6 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
> tail(rainfall_dataframe)
     year month day 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
5065   80    11  25 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5066   80    11  26 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5067   80    11  27 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5068   80    11  28 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5069   80    11  29 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
5070   80    11  30 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

We have assigned names to the first 3 columns of the rainfall dataframe and sequenced numbers from 0-23 for the remaining 24 columns. It is clear that these are hourly rainfall values in a day.

g.

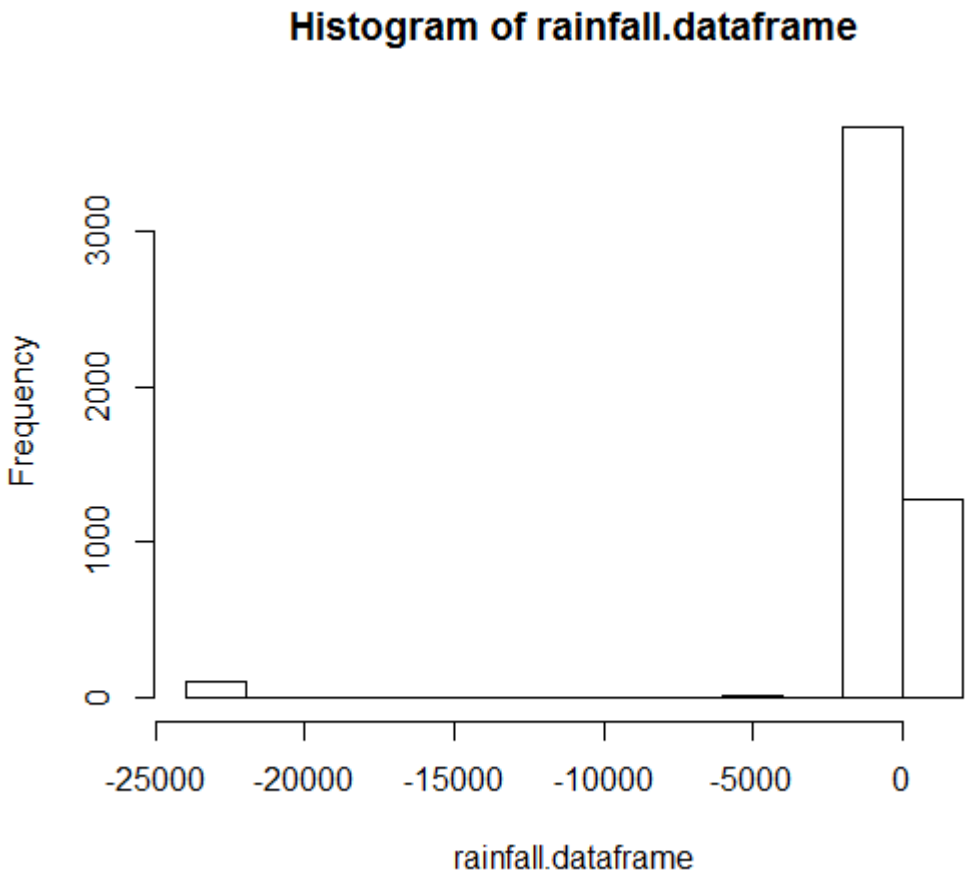
```
> rainfall_dataframe$daily <-rainfall_dataframe[4]+rainfall_dataframe[5] +
rainfall_dataframe[6]+rainfall_dataframe[7]+rainfall_dataframe[8]+rainfall
_dataframe[9]+rainfall_dataframe[10]+rainfall_dataframe[11]+rainfall_dataf
rame[12]+rainfall_dataframe[13]+rainfall_dataframe[14]+rainfall_dataframe[
15]+rainfall_dataframe[16]+rainfall_dataframe[17]+rainfall_dataframe[18]+r
ainfall_dataframe[19]+rainfall_dataframe[20]+rainfall_dataframe[21]+rainfa
ll_dataframe[22]+rainfall_dataframe[23]+rainfall_dataframe[24]+rainfall_da
taframe[25]+rainfall_dataframe[26]+rainfall_dataframe[27]
#is there an easier way to do this?
```

```
> head(rainfall_dataframe)
  year month day 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 daily
1   60     4   1 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0
2   60     4   2 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0
3   60     4   3 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0
4   60     4   4 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0
5   60     4   5 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0
6   60     4   6 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 0
```

```
> rainfall_dataframe$daily <-as.numeric(unlist(lapply(rainfall_dataframe$d
aily,as.numeric)))
```

Converting daily column in rainfall_dataframe into numeric type.

```
> hist(rainfall_dataframe$daily)
```



As seen from the histogram, rainfall shows negative values as well with all these values being -999. This is an error value since rainfall cannot be negative. Probably, missing or NA values where converted to -999 because of which the histogram shows an incorrect distribution. Assuming these records have 0 rainfall, we need to replace all values of -999 with 0, and then plot the histogram.

h. **As explained in g, replacing the missing values with 0. Naming the new dataframe as rainfall_dataframe.fixed**

Replacing the negative values (values<0) with 0:

```
> rainfall_dataframe.fixed <- rainfall_dataframe
```

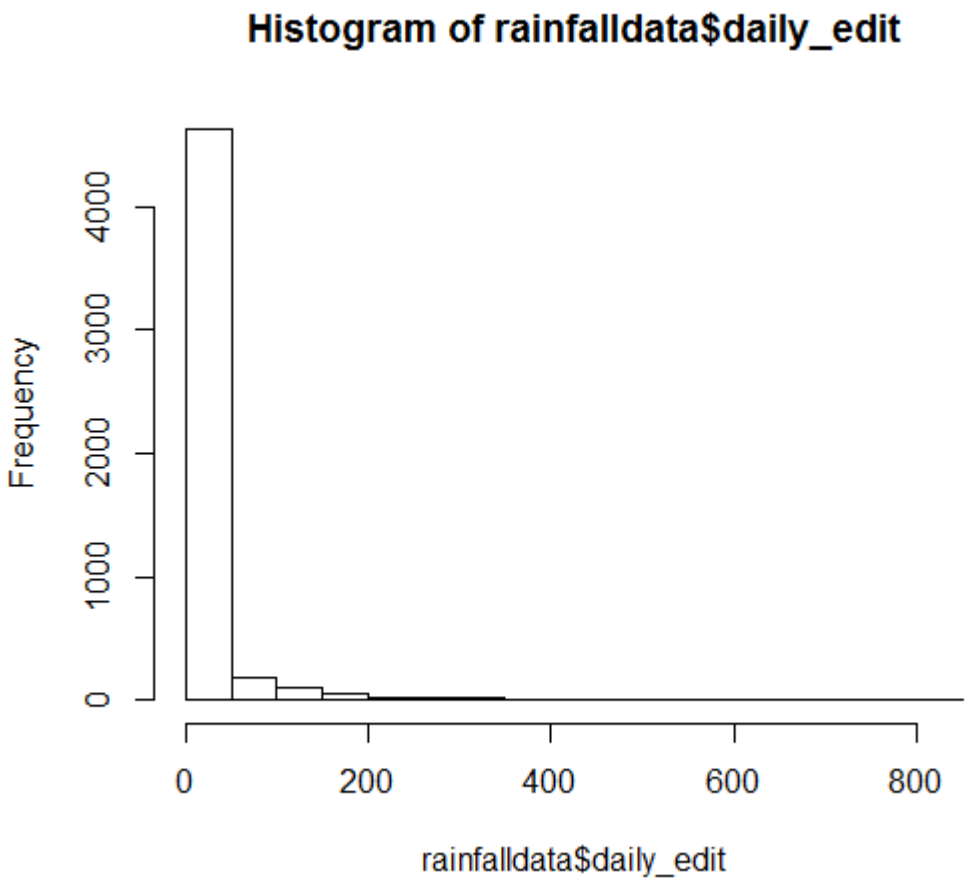
```
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V4 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V5 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V6 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V7 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V8 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V9 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V10 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V11 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V12 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V13 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V14 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V15 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V16 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V17 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V18 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V19 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V20 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V21 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V22 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V23 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V24 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V25 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V26 < 0, ] <- 0
> rainfall_dataframe.fixed[rainfall_dataframe.fixed$V27 < 0, ] <- 0
```

```
Creating the column daily
```

```
> rainfall_dataframe.fixed$daily <-(rainfall_dataframe.fixed[4]+rainfall_d
ataframe.fixed[5] + rainfall_dataframe.fixed[6]+rainfall_dataframe.fixed[7
]+rainfall_dataframe.fixed[8]+rainfall_dataframe.fixed[9]+rainfall_datafra
me.fixed[10]+rainfall_dataframe.fixed[11]+rainfall_dataframe.fixed[12]+rai
nfall_dataframe.fixed[13]+rainfall_dataframe.fixed[14]+rainfall_dataframe.
fixed[15]+rainfall_dataframe.fixed[16]+rainfall_dataframe.fixed[17]+rainfa
ll_dataframe.fixed[18]+rainfall_dataframe.fixed[19]+rainfall_dataframe.fix
ed[20]+rainfall_dataframe.fixed[21]+rainfall_dataframe.fixed[22]+rainfall_
dataframe.fixed[23]+rainfall_dataframe.fixed[24]+rainfall_dataframe.fixed[
25]+rainfall_dataframe.fixed[26]+rainfall_dataframe.fixed[27])
```

```
> rainfall_dataframe.fixed$daily <- as.numeric(unlist(lapply(rainfall_data
frame.fixed$daily,as.numeric)))
```

```
> hist(rainfall_dataframe.fixed$daily)
```


Histogram of rainfalldata$daily_edit
Frequency
0
1000
2000
3000
4000
0
200
400
600
800
rainfalldata$daily_edit