Homework 1

Sara Shao

7/25/2021

1a.

```
data <- read.table("homeworks/homework-1/data/rnf6080.dat")
rain.df <- data.frame(data)
```

To load the data into R, I used read.table. To make the data into a data frame, I used data.frame.

b. There are 5070 rows and 27 columns, which I know because in the environment tab the description of rain.df says "5070 obs. of 27 variables." Observations translates to rows and variables translates to columns.

c. colnames gives the column names, which are outputted below:

```
colnames(rain.df)
```

```
##  [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"
```

d.

```
rain.df[2,4]
```

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

e.

```
rain.df[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
```

f.

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

```
##   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
```

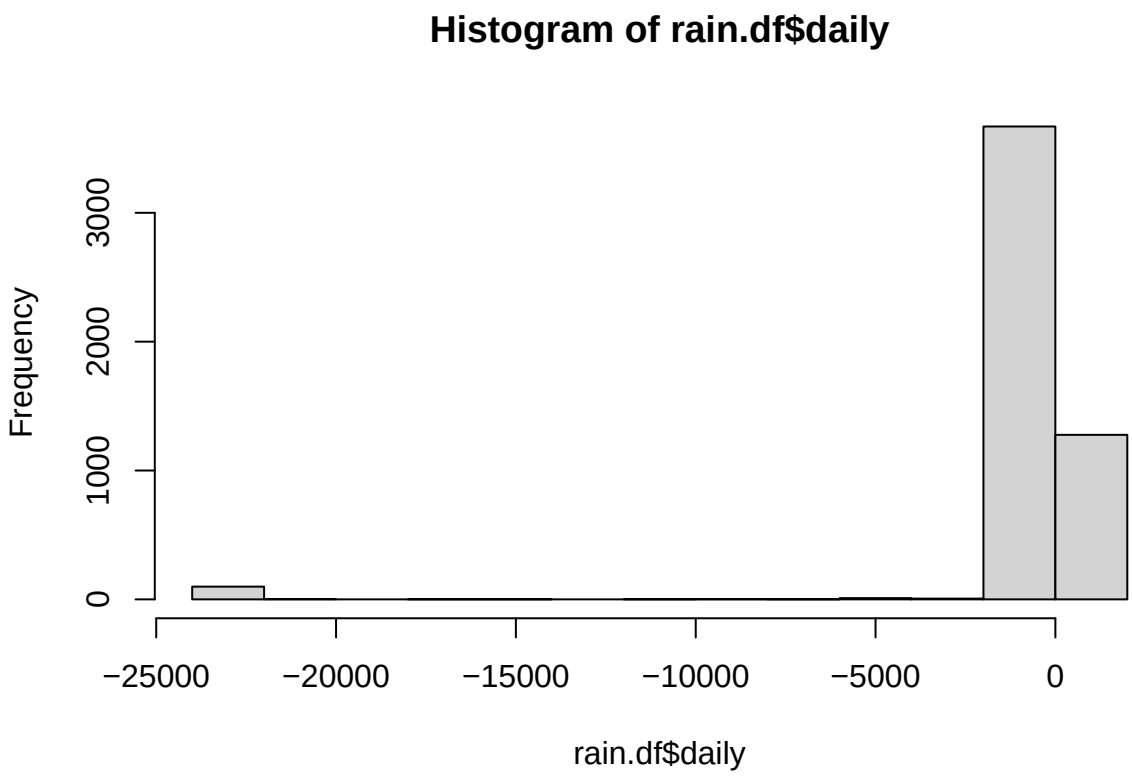
The above command changes the names of the first three columns to the year, day, and month of the observations, and the next 24 columns represent the hour of the day that the observation was taken.

g.

```
rain.df$daily <- rowSums(rain.df[, c(4:27)])
```

h.

```
hist(rain.df$daily)
```



i. The above histogram cannot be right because it shows that some of the daily rainfall amounts were negative, which is not possible.

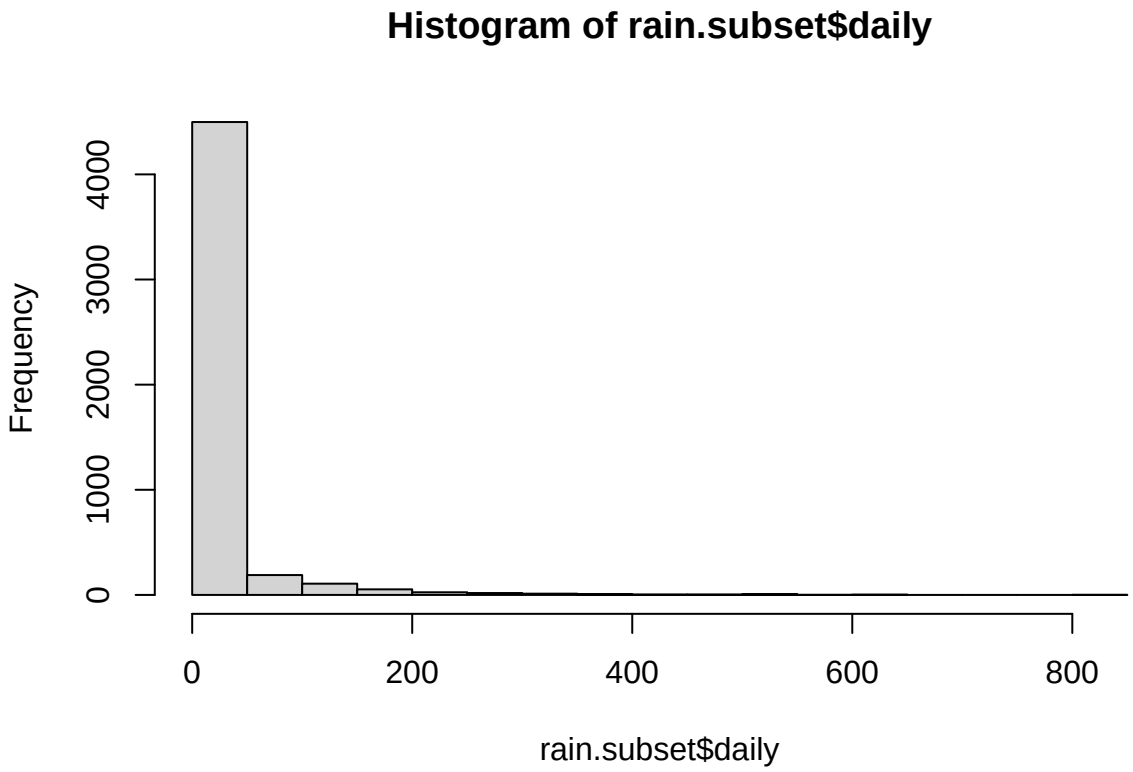
j.

```
rain.subset <- subset(rain.df, daily >= 0)
```

The above command remove rows that have negative values.The reason the dataset had negative values was because the dataset represented N/A values as -999.

k.

```
hist(rain.subset$daily)
```



This histogram is more reasonable because it no longer includes negative values for daily amounts of rainfall.

2a.

max(x) would return "7" in this case because when making string comparisons, one character is compared at a time, so in comparing the first character of each string: "5" to "1" to "7", "7" is the highest based on its ASCII value.

sort(x) returns "12" "5" "7" by the same logic of comparing one character at a time based on their ASCII values.

sum(x) returns an error because the sum() function only works on numeric values.

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

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

```
sort(x)
```

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

b. y <- c("5",7,12) y[2] + y[3]

These two commands would produce an error. This is because arrays can only store one type of value, and since one value is a string, the other values are automatically converted into strings as well. Therefore, when the second command tries to add the second and third values in the array, it results in an error since only numeric values can be added.

c.

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

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

These two commands result in the value 19 because unlike an array, a data frame can store values of different types in different columns, so there is no type conversion on the assigned values. The double values in columns 2 and 3 of row 1 are 7 and 12 respectively, which when added together results in 19.

3a. The point of reproducible code is to make it easier to judge the validity and reliability of the results.

b. If the code you write for a homework assignment or research project produces different results than a peer's, and you've set the seeds to be the same, you will know it's because of differences in the functionality of the code. Otherwise, it could also be due to differences in random sampling, but it can become difficult to tell whether it's one or the other.

c. 3