Homework1

Hong Xu

August 27, 2015

1. **Read the rain dataset**

a. read the data and make it a dataframe called rain.df

```
rain.df <- read.table("rnf6080.dat")
```

b. how many rows and columns?

```
print(nrow(rain.df))
```

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

```
print(ncol(rain.df))
```

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

c. what are the names of the columns?

```
print(colnames(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"
```

d. what is the value of row 5, column 7?

```
print(rain.df[5, 7])
```

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

e. Display the second row of rain.df in its entirety.

```
print(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
```

f.

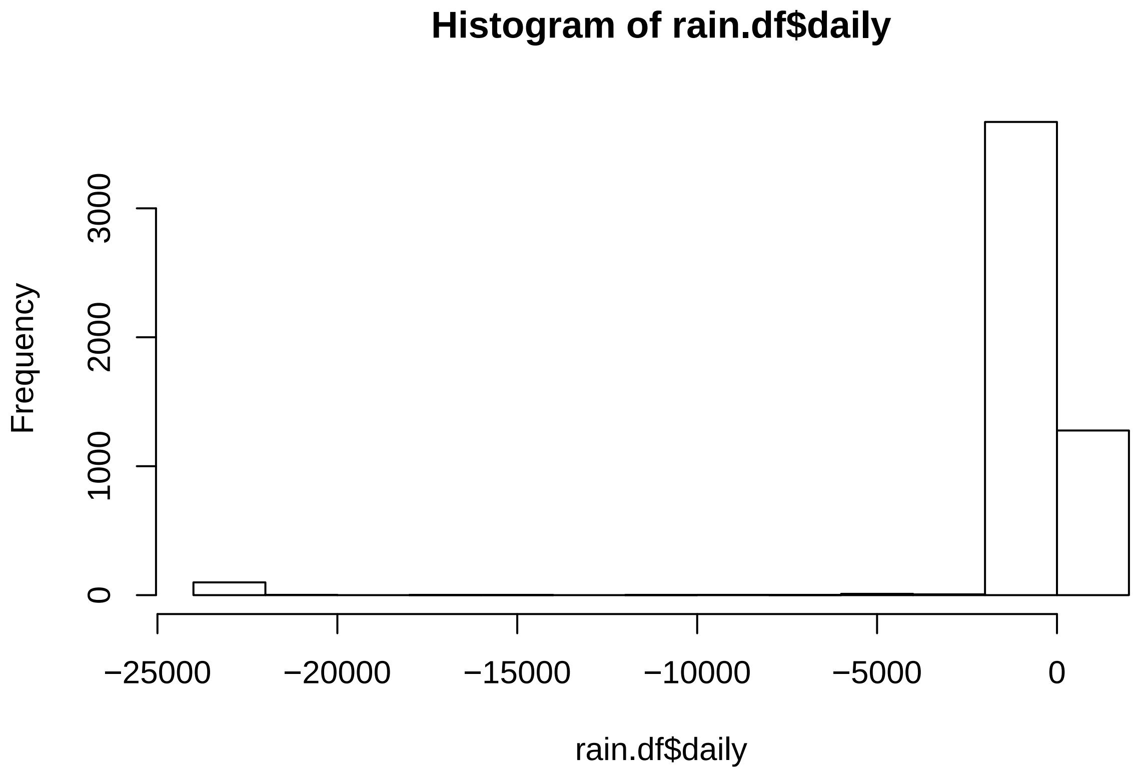
```r
names(rain.df) <- c("year","month","day",seq(0,23))
head(rain.df)
```

```
##   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
## 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
## 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
## 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
## 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
## 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
## 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
##   22 23
## 1  0  0
## 2  0  0
## 3  0  0
## 4  0  0
## 5  0  0
## 6  0  0
```

```r
# what the command does is to assign new column names so that they make more sense.
# now we can see clearly that the first three columns represent the year, month, and day
# and the rest columns are the rainfall level at the specific hour (0-23) of that day
```

g.Create a new column in the data frame called daily, which is the sum of the rightmost 24 columns. With this column, create a histogram of the values in this column, which are supposed to be daily rainfall values. What is wrong with this picture?

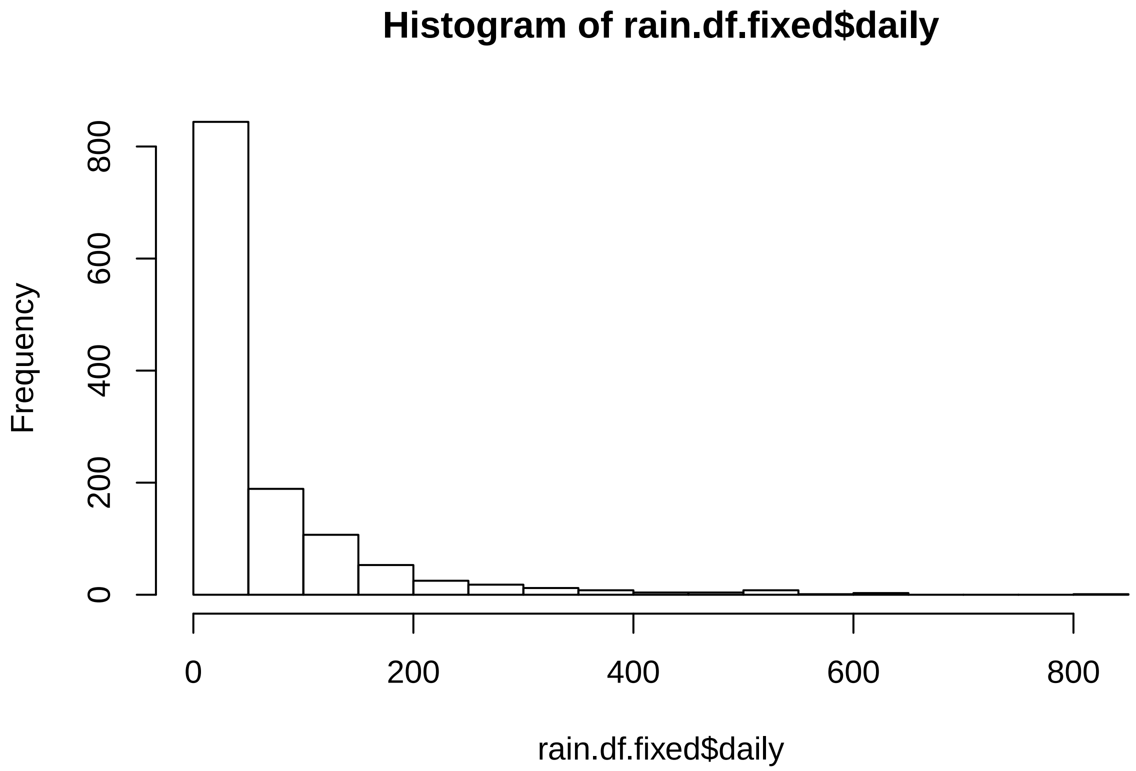
```r
rain.df$daily <- apply(rain.df[, seq(4, 27)], 1, sum)
hist(rain.df$daily)
```



```
# problem: there are negative values in the data! - which is impossible for rainfall levels
```

h. Create a new data frame rain.df.fixed that takes the original and fixes it for the apparent flaw you have discovered. Having done this, produce a new histogram with the corrected data and explain why this is more reasonable.

```
rain.df.fixed <- rain.df[rain.df$daily>0, ]
hist(rain.df.fixed$daily)
```



```
# this new plot is more reasonable since it reflects what you expect to see for
# rainfall levels - majority of them have lower value and as the value increases,
# the fewer occurances.
```

2. **Syntax and class typing**

a.

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

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

```
# correct code, erroneous result: the data type of vector is character,
# so with max you don't get the expected result

sort(vector1)
```

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

```
# correct code, erroneous result: same reason as above

sum(vector1)
```

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

```
# erroneous code: character type does not support add operation
```

b.

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

```
## Error in vector2[2] + vector2[3]: non-numeric argument to binary operator
```

```
# error: c("5",7,12) will coerce 7, 12 to the type of "5",
# which is character, which cannot be added

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

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

```
# correct results: dataframe support heterogeneous data type

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

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

```
# correct results: list support heterogeneous data type
# and using [[ will pull out the actual component of the list

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

```
## Error in list4[2] + list4[4]: non-numeric argument to binary operator
```

```
# error: using [ will return a list thus cannot be added
```

3. **Working with functions and operators**

a.

```
seq(1, 10000, by=372)
```

```
##  [1]    1  373  745 1117 1489 1861 2233 2605 2977 3349 3721 4093 4465 4837
## [15] 5209 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.

```r
rep(1:3, times=3)
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

```r
rep(1:3, each=3)
```

```
## [1] 1 1 1 2 2 2 3 3 3
```

```r
# rep will repeat the whole vector (the first argument x) the "times" times
# while for "each", rep will repeat each element of x that many times, in order
```