STA 325: Homework 1

Connie Xu

2024-08-31

```
# install.packages("ggplot2") FIX
library(ggplot2)
```

Problem 1

a.

```
# read dataset into rain_df variable
rain_df <- read.table("data/rnf6080.dat")
```

The command I used was **read.table**.

b.

```
# get number of rows and columns in rain_df
num_rows <- nrow(rain_df)
num_cols <- ncol(rain_df)

print(num_rows)
```

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

```
print(num_cols)
```

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

Using the **nrow** and **ncol** commands, I found that rain_df has **5070 rows** and **27 columns**.

c.

```
# get column names
col_names <- colnames(rain_df)

print(col_names)
```

```
##  [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"
```

The **colnames** command can be used to get the column names. The names are **"V1", "V2", "V3", "V4", "V5", "V6", "V7", "V8", "V9", V10", "V11", "V12", "V13", "V14", "V15", "V16", "V17", "V18", "V19", "V20", "V21", "V22", "V23", "V24", "V25", "V26", and "V27"**.

d.

```
# get the value at row 2, column 4
value <- rain_df[2, 4]

print(value)
```

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

I can get the value at row 2, column 4 by indexing rain_df by [row, column]. The value is **0**.

e.

```r
# display the whole second row
print(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
```

I can display the entire second row by indexing rain_df with row 2 and printing the result. The content is: V1 = 60, V2 = 4, V3 = 2, V4-V27 = 0.

f.

```r
# renames the column names in ascending order from V1 to V27
names(rain_df) <- c("year","month","day",seq(0,23))
```

This command renames the existing column names in rain_df to the ones specified, in the order that they were specified. The 1st column is renamed to "year", The 2nd column is renamed to "month", and the 3rd column is renamed to "day". seq(0, 23) generates a sequence of numbers from 0 to 23, and the remaining columns (4th to 27th) are renamed to those numbers in order. Column 4 is renamed to 0, column 5 is renamed to 1, . . . , column 27 is renamed to 23.

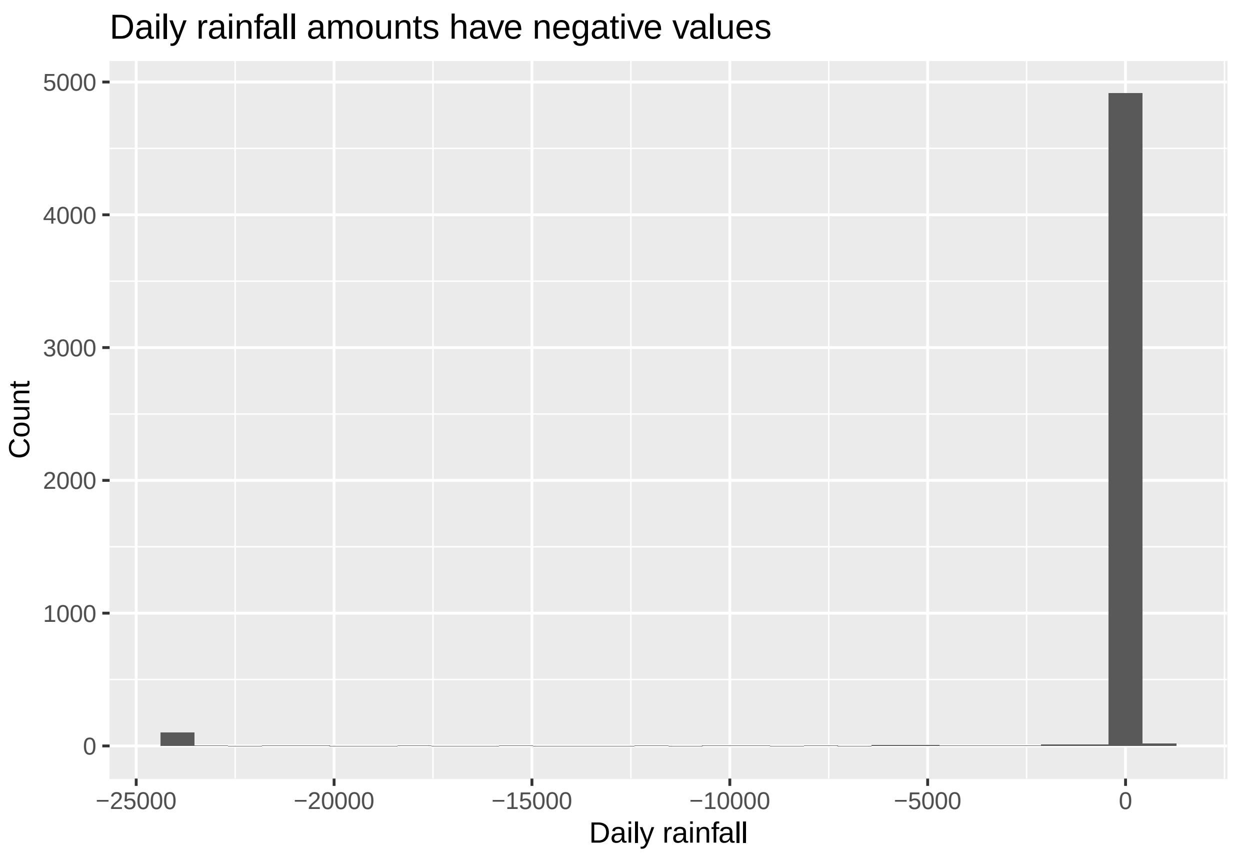
g.

```r
# create daily_rain_fall column, which sums up the 24 hourly columns
# R is 1-indexed
rain_df$daily_rain_fall <- rowSums(rain_df[4:27])
```

h.

```r
histogram <- rain_df |> ggplot(aes(x=daily_rain_fall)) +
  geom_histogram() +
  labs(title="Daily rainfall amounts have negative values",
       x="Daily rainfall",
       y="Count")

print(histogram)
```



i.

```r
unique_values <- unique(rain_df[,4])

# Print the unique values
print(unique_values[unique_values < 0])
```

```
## [1] -999
```

The histogram cannot possibly be right because the daily rainfall amount cannot be negative. One potential explanation is that missing hourly rainfall entries may have been recorded as negative values (e.g. -999).

j.

```r
# change negative values to zero
rain_df$daily_rain_fall <- rowSums(replace(rain_df[4:27],
                                           rain_df[4:27] < 0,
                                           0))
```

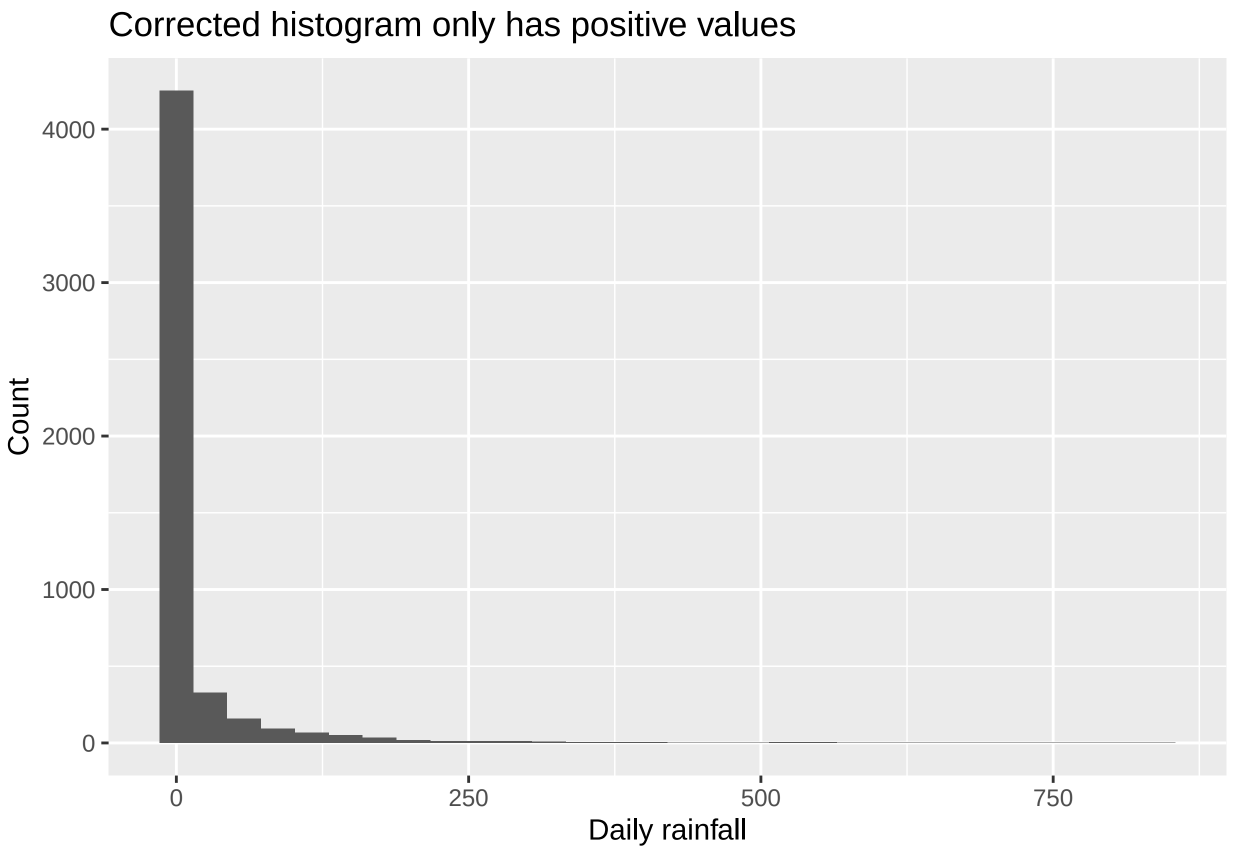
This command changes any negative values in the 4th to 27th columns to zero.

k.

```r
# Create the corrected the histogram

correct_histogram <- rain_df |> ggplot(aes(x=daily_rain_fall)) +
  geom_histogram() +
  labs(title="Corrected histogram only has positive values",
       x="Daily rainfall",
       y="Count")
```

```r
print(correct_histogram)
```



The new histogram is more reasonable because it only includes values that are zero or higher, which reflects realistic rainfall measurements. Most of the values are close to zero, which is reasonable for rainfall data since many days might have little to no rainfall.

Problem 2

a.

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

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

This command assigns the character vector containing "5", "12", and "7" to a variable x.

```r
max(x)
```

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

R treats "5", "12", and "7" as strings and finds the biggest one lexicographically. When comparing the first character in each string, "7" is greater than "5" and the "1" in "12", so the result is is "7".

```r
sort(x)
```

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

R sorts them in ascending order lexicographically. "1" < "5" < "7", so the result is "12", "5", and "7".

```
sum(x)
```

R doesn't know how to sum up strings because it doesn't have a way of knowing what numerical value each string has, so this command returns an error.

b.

```
y <- c("5",7,12)
y
```

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

This command assigns variable y to a character vector containing "5", "7", and "12". R implicitly coerces the integers to be of type character.

`y[2] + y[3]`

R doesn't know how to sum up strings because it doesn't have a way of knowing what numerical value each string has, so this command returns an error.

c.

```
z <- data.frame(z1="5",z2=7,z3=12)
z
```

```
##   z1 z2 z3
## 1  5  7 12
```

This command assigns a variable z to a data frame with one row and 3 columns (z1, z2, and z3). In the row, z1 has a string value of "5", z2 has a numeric value of 7, and z3 has a numeric value of 12. Columns in a data frame can have different data types.

```
z[1,2] + z[1,3]
```

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

This command returns 19 because it adds together the value in the 1st row, 2nd column (which is 7) and the value in the 1st row, 3rd column (which is 12). Both values are numeric, so the summation doesn't return an error.

Problem 3

a. Reproducible code ensures that my results can be replicated and verified by others. This also allows others to collaborate with you and build upon your work.
b. For example, the teaching assistants in this class need to be able to run my code on their own machine to make sure it works/produces the expected output when grading an assignment. If my code isn't reproducible, the TAs may encounter errors and cannot verify that I did the assignment correctly. Reproducibility ensures that the code runs smoothly on their systems, not just my own.
c. I rate this assignment a 4.