Data Wrangling

STA 325: Homework 1

Today's agenda: Manipulating data objects; using the built-in functions, doing numerical calculations, and basic plots; reinforcing core probabilistic ideas.

General instructions for homeworks: Please follow the uploading file instructions according to the syllabus. You will give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used. Your code must be completely reproducible and must compile.

Advice: Start early on the homeworks and it is advised that you not wait until the day of. While the professor and the TA's check emails, they will be answered in the order they are received and last minute help will not be given unless we happen to be free.

Commenting code Code should be commented. See the Google style guide for questions regarding commenting or how to write code https://google.github.io/styleguide/Rguide.xml. No late homework's will be accepted.

R Markdown Test

0. Open a new R Markdown file; set the output to HTML mode and "Knit". This should produce a web page with the knitting procedure executing your code blocks. You can edit this new file to produce your homework submission.

Working with data

Total points on assignment: 10 (reproducibility) + 22 (Q1) + 9 (Q2) + 3 (Q3) = 44 points

Reproducibility component: 10 points.

1. (22 points total, equally weighted) The data set **rnf6080.dat** records hourly rainfall at a certain location in Canada, every day from 1960 to 1980.

a. Load the data set into R and make it a data frame called rain_df. What command did you use?

```r
#read rnf6080 into rain_df data frame using read.table command
rain_df <- read.table("~/data-clean/homeworks/homework-1/data/rnf6080.dat")
```

b. How many rows and columns does rain_df have? How do you know? (If there are not 5070 rows and 27 columns, you did something wrong in the first part of the problem.)

```r
#store number of rows in to a variable named rows_num using nrow command
rows_num <- nrow(rain_df)
#store number of columns in to a variable named cols_num using ncol command
cols_num <- ncol(rain_df)

#print number of rows and columns
#There are 5070 rows
print(rows_num)
```

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

```r
#There are 27 columns
print(cols_num)
```

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

c. What command would you use to get the names of the columns of rain_df? What are those names?

```r
# Get the column names of the data frame using names command
col_names <- names(rain_df)

# Print column names
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 names are V1 to V27.

d. What command would you use to get the value at row 2, column 4? What is the value?

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

# Print the value
#The value is 0
print(value)
```

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

e. What command would you use to display the whole second row? What is the content of that row?

```r
# Store the second row into a varible named second_row
second_row <- rain_df[2, ]

# Print the second row
print(second_row)
```

```
##   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
```

Second row V1 is 60, second row V2 is 4, second row V3 is 2 while the rest of the row are 0.

f. What does the following command do?

```r
names(rain_df) <- c("year","month","day",seq(0,23))
```

The first three columns are renamed with "year", "month", and "day". The remaining columns, from the fourth to the twenty-seventh, are renamed with the numbers from 0 to 23.

g. Create a new column called daily_rain_fall, which is the sum of the 24 hourly columns.

```r
# Sum columns 4 through 27 (which are hour 0 to hour 23)
#and assign it to a new column called 'daily_rain_fall'
rain_df$daily_rain_fall <- rowSums(rain_df[, 4:27])
```

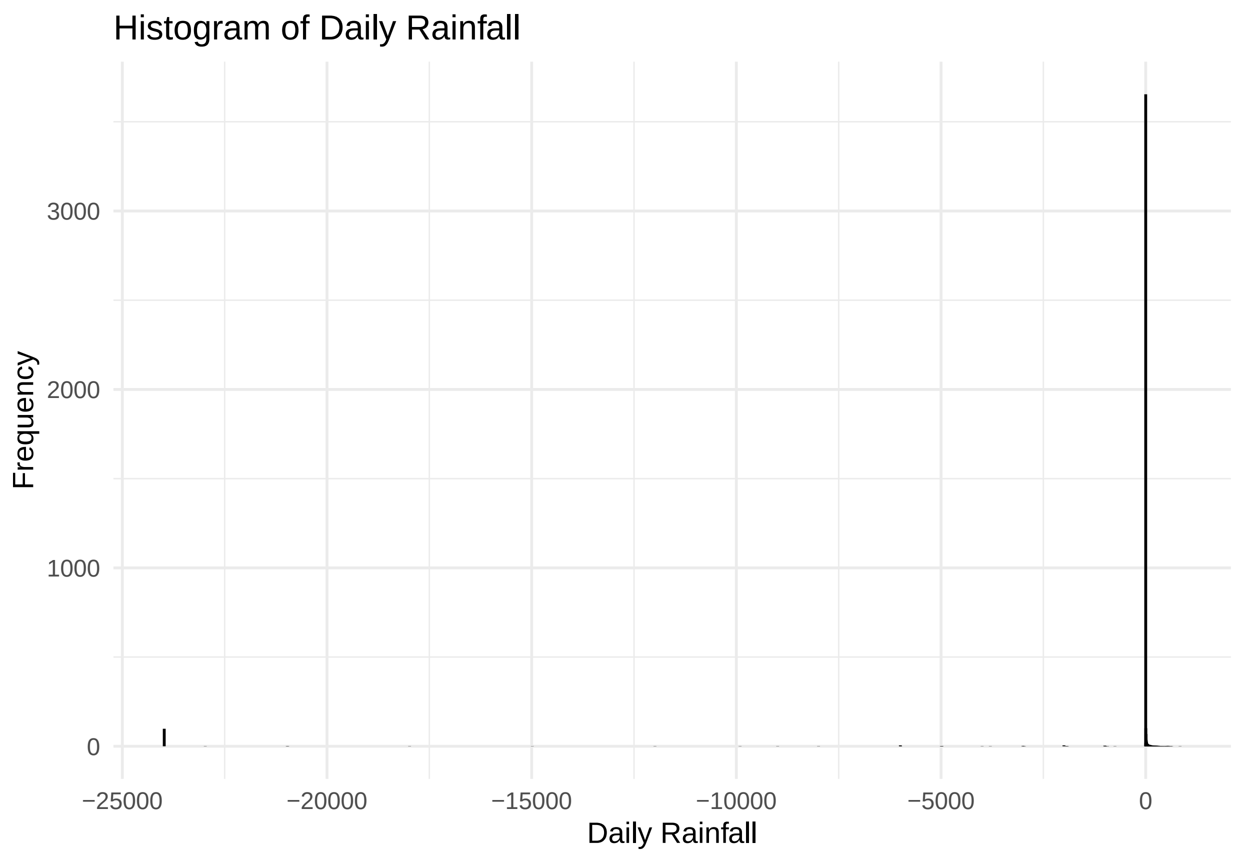
h. Give the command you would use to create a histogram of the daily rainfall amounts. Please make sure to attach your figures in your .pdf report.

```r
# Install ggplot2
install.packages("ggplot2")
```

```
## Installing package into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

```r
# Load the ggplot2 library
library(ggplot2)
```

```r
# Create a histogram of the 'daily_rain_fall' column
ggplot(rain_df, aes(x = daily_rain_fall)) +
  geom_histogram(binwidth = 0.1, fill = "blue", color = "black") +
  labs(title = "Histogram of Daily Rainfall",
       x = "Daily Rainfall",
       y = "Frequency") +
  theme_minimal()
```



i. Explain why that histogram above cannot possibly be right.

Currently, there are negative values in the daily_rain_fall column, which is unexpected because daily rainfall cannot be negative. These negative values likely indicate data recording errors, possibly due to incorrect entries for dates when no data was measured.

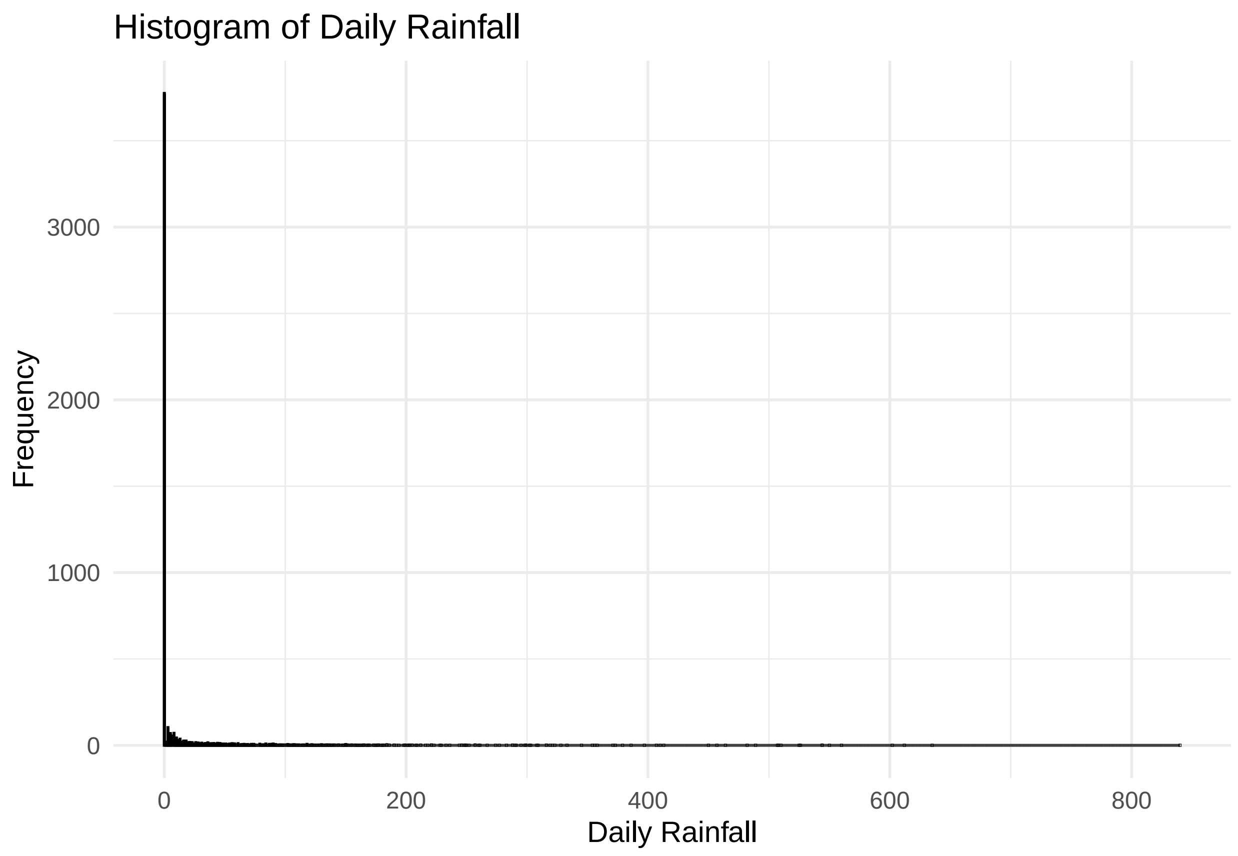
j. Give the command you would use to fix the data frame.

```r
#replace negative hourly rainfall with 0
#(We will not use NA because NA values can't be added)
rain_df[, 4:27][rain_df[, 4:27] < 0] <- 0
#update daily rainfall with correct hourly rainfall
```

```r
rain_df$daily_rain_fall <- rowSums(rain_df[, 4:27])
```

k. Create a corrected histogram and again include it as part of your submitted report. Explain why it is more reasonable than the previous histogram.

```r
# Update the histogram of the 'daily_rain_fall' column
ggplot(rain_df, aes(x = daily_rain_fall)) +
  geom_histogram(binwidth = 0.1, fill = "blue", color = "black") +
  labs(title = "Histogram of Daily Rainfall",
       x = "Daily Rainfall",
       y = "Frequency") +
  theme_minimal()
```



The histogram now shows a range from 0 to 800 for daily rainfall, with no negative values, which makes more sense and aligns with the expected data distribution.

Data types

2. (9 points, equally weighted) Make sure your answers to different parts of this problem are compatible with each other.

a. For each of the following commands, either explain why they should be errors, or explain the non-erroneous result.

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

The first line creates a vector named x with character values of “5”,“12”,and “7”. The second line identifies the largest value in the vector. We would expect the following outcomes for numerical data: the maximum value being 12, the sorted output being 5, 7, 12. However, since 5, 12, and 7 are treated as character data types, the results differ. In lexicographical order, the maximum value is 7, as it’s the highest when compared as strings. Sorting the values lexicographically yields 12, 5, and 7. The third row intends to sum all the values in the vector, but calculating the sum is not feasible with character data types.

b. For the next two commands, either explain their results, or why they should produce errors.

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

The first row created a vector named y contained mixed data types, and in R, when a vector holds different types of data, it is coerced to the most flexible type that can accommodate all values. In this case, that type is character.

As a result, the second value in vector y y[2] and the third value in vector y y[3] are treated as characters, which leads to an error when attempting to perform arithmetic operations on them in the second line.

c. For the next two commands, either explain their results, or why they should produce errors.

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

We created a data frame named z with one row and three columns with name z1, z2, and z3. The value in the first row, first column is the character “5”, while the values in the first row, second column, and first row, third column are the numbers 7 and 12, respectively. When the second line of code adds the values from the first row, second column (7) and the first row, third column (12), both are numeric and can be added together, resulting in 19.

3. (3 pts, equally weighted).

a.) What is the point of reproducible code? Reproducible code allows anyone, including original author, to produce the same code, regardless of environment and time. It allows validation and collaboration , ensuring that the code can trusted and built upon on.

b.) Given an example of why making your code reproducible is important for you to know in this class and moving forward.

For example, if I perform an analysis in class and obtain unexpected results, the TAs can only assist me effectively if they can reproduce the same error using my code. Reproducible code allows them to understand the issue clearly and provide targeted feedback to help me resolve it.

c.) On a scale of 1 (easy) – 10 (hard), how hard was this assignment. If this assignment was hard (> 5), please state in one sentence what you struggled with. On a scale of 1 – 10, the assignment is 4.