Data Wrangling in `R`

STA 360/602: Homework 1

Due Friday January 7 at 5 PM EDT

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
# rain.df <- read.table("data/rnf6080.dat", header=F)
rain.df <- read.table("https://raw.githubusercontent.com/resteorts/modern-bayes/master/homeworks/homewo

# look at the first few rows of rain.df
head(rain.df)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21
## 1 60  4  1  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
## 3 60  4  3  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
## 5 60  4  5  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
##   V22 V23 V24 V25 V26 V27
## 1   0   0   0   0   0   0
## 2   0   0   0   0   0   0
## 3   0   0   0   0   0   0
## 4   0   0   0   0   0   0
## 5   0   0   0   0   0   0
## 6   0   0   0   0   0   0
```

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.)

```
cat('number of rows =', nrow(rain.df), '\n')
```

```
## number of rows = 5070
```

```
cat('number of columns =', ncol(rain.df), '\n')
```

```
## number of columns = 27
```

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

```
# use the function colnames() to get the names of the columns, which are as printed 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. What command would you use to get the value at row 2, column 4? What is the value?

```
# using square brackets to index the dataframe to get the value at row 2 and column 4
rain.df[2,4]
```

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

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

```
# like before, but leave the column index blank to access values from the whole row
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. What does the following command do?

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

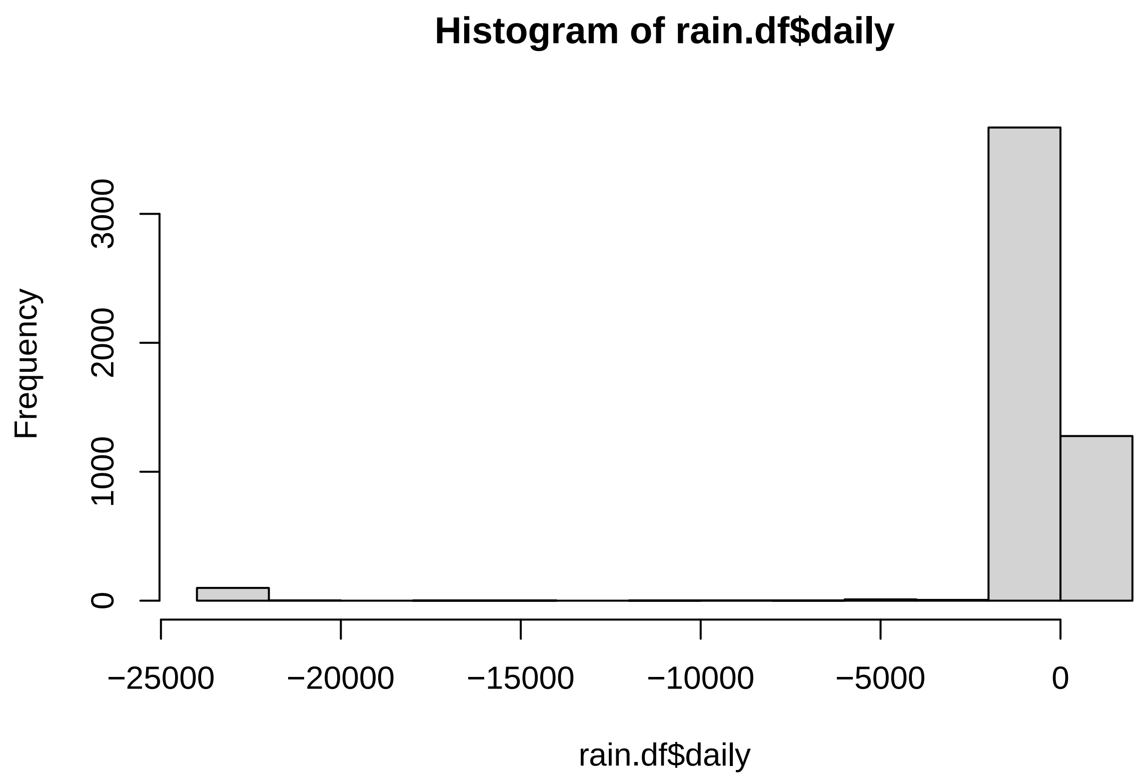
- This command will change the names of the variables (columns) in the dataframe; the first three columns will be 'year', 'month', and 'day', and the following 24 columns will be named as integers 0-23.

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

```
# missing data are coded as -999
# change them to NA so that they will not affect the mean or sum
rain.df$daily <- rowSums(rain.df[, as.character(c(0:23))],
                         na.rm=T)
```

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.

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



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

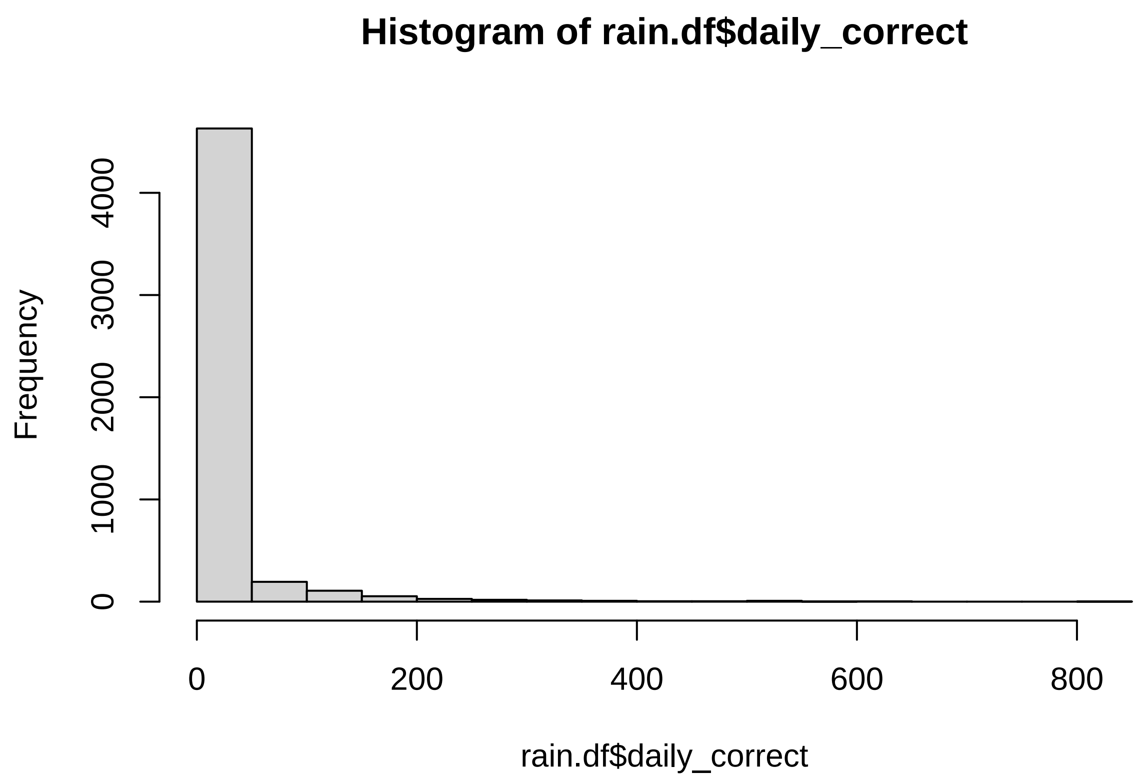
- Rainfall amount cannot possibly be negative.

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

```r
# missing data are coded as -999
# change them to NA so that they will not affect the mean or sum
rain.df[rain.df==-999] <- NA
# ignore NA entries in the dataframe with na.rm=T
rain.df$daily_correct <- rowSums(rain.df[, as.character(c(0:23))],
                         na.rm=T)
```

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
hist(rain.df$daily_correct)
```



- This is more reasonable as 1) it does not contain negative values and 2) it has a long right tail showing that most daily rainfall amounts are close to zero.

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.

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

- The first `c()` command combines the three variable of character type into one vector `"5" "12" "7"` and assigns it to variable `x`.
- The second `max()` command finds the maximum of the three elements, which would be `"7"` because the comparison is done on the first character of each element in the vector.
- The third `sort()` command sorts the elements in the vector in ascending order of the first character of each element and returns `"12" "5"  "7"`
- The `sum(x)` command will throw an error as characters cannot be added together.It seems that variable `x` should have been a numeric array `c(5,12,7)` for the intended operations.

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 command using the `c()` function combines arguments to form a vector, while coercing them all the be characters. Hence the result is equivalent to that of `y <- c("5", "7", "12")`.
- The second command will throw an error because characters cannot be added.

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]
```

- The first command will create a data frame with 1 row and 3 columns (named z1, z2, and z3), and the entries are specified and their data types preserved.
- The second command will add the entries in row 1 column 2 and row 1 column 3, which is `7+12=19`.

3. (3 pts, equally weighted).

a.) What is the point of reproducible code?

- Reproducible code is easier for others (and my future self) to understand, trace back any potential source of errors, and reuse for similar practices in the future.

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

- In this class, reproducibility is part of the homework grade.

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.

- 2