Data Wrangling in R

STA 360: Assignment 1, Fall 2020

Due Friday August 21, 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?

I used the read.table function to load the data set into R.

```
rain.df = read.table("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.)

`rain.df` has 5070 rows and 27 columns. I found out by using the nrow() and ncol() functions as shown below.

```
### Find number of rows in rain.df
nrow(rain.df)
```

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

```
### Find number of columns in rain.df
ncol(rain.df)
```

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

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

I would use the names() command. The names are "V1" through "V27".

```
names(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?

To get the value at row 2, column 4, I would put the indices 2 and 4 in brackets as shown below to get the value, which is 0.

```
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?

I would display the whole second row using indices in brackets as shown below. The content of the row is as shown below.

```
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))
```

The command replaces the names of the columns of `rain.df` with "year", "month", "day", and the numbers from 0 to 23, in that order. This is demonstrated with the code below:

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

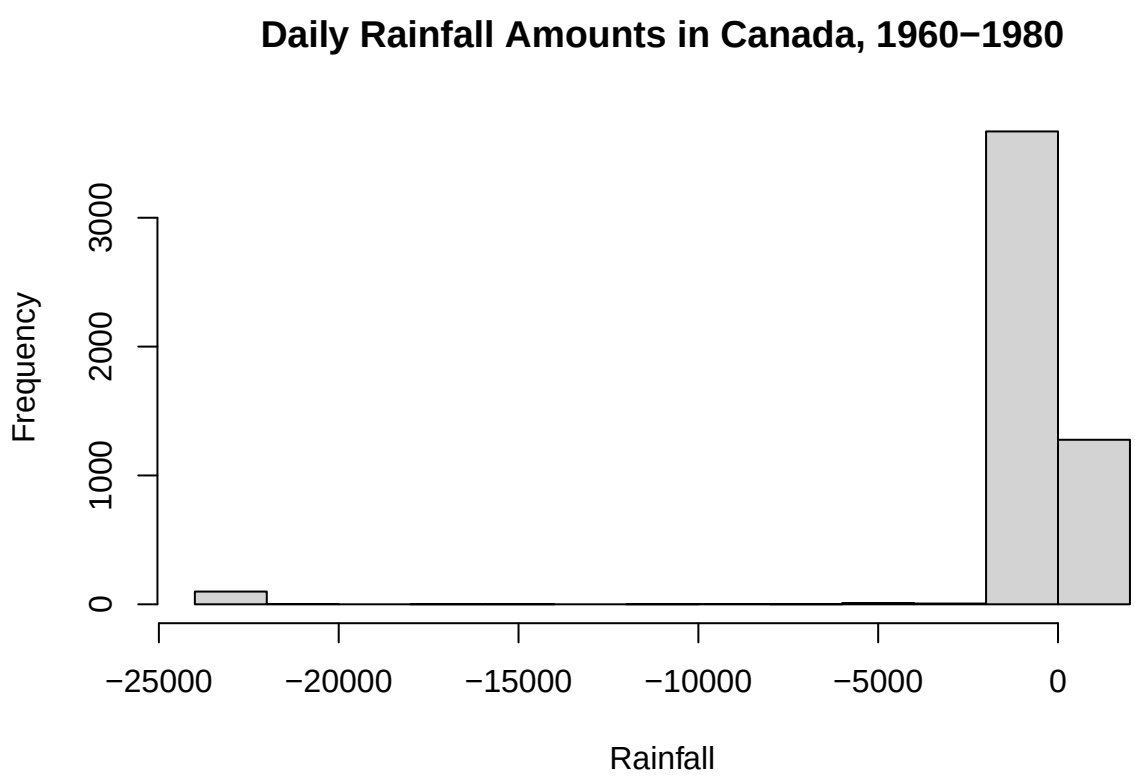
```
##  [1] "year"  "month" "day"   "0"     "1"     "2"     "3"     "4"     "5"
## [10] "6"     "7"     "8"     "9"     "10"    "11"    "12"    "13"    "14"
## [19] "15"    "16"    "17"    "18"    "19"    "20"    "21"    "22"    "23"
```

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

```
### Create daily column using columns 4 to 27, the hourly columns
rain.df$daily <- 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. I used the hist() function to plot the daily rainfall amounts.

```
### Plot daily rainfall amounts in a histogram, adding appropriate labels
hist(rain.df$daily, main = "Daily Rainfall Amounts in Canada, 1960-1980", xlab = "Rainfall")
```



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

The histogram above cannot possibly be right because negative values are present. It is impossible to observe negative rainfall amounts.

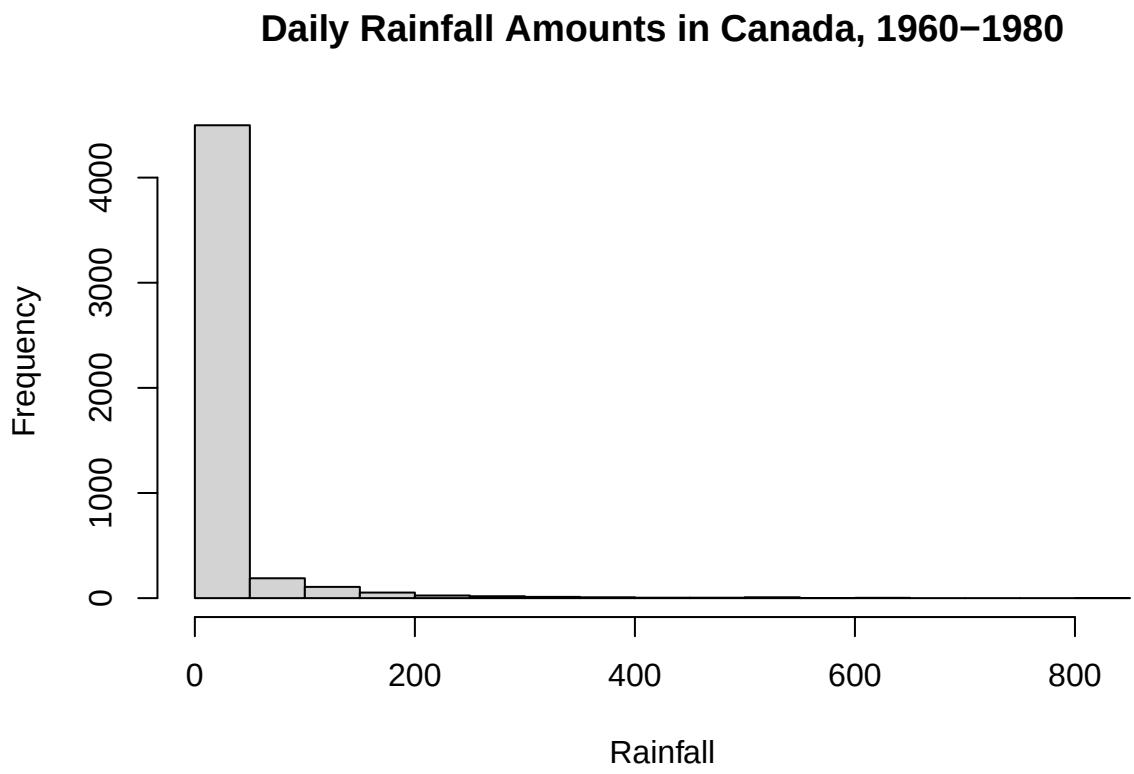
j. Give the command you would use to fix the data frame.

I used an ifelse() command to set negative values to "NA" to indicate that they should not show up in the histogram.

```
### Create a new column, daily_corrected
### Same as daily but negative values are now listed as NA
rain.df$daily_corrected <- ifelse(rain.df$daily >= 0, rain.df$daily, NA)
```

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.

```
### Plot daily rainfall amounts in a histogram without negative values, adding appropriate labels
hist(rain.df$daily_corrected, main = "Daily Rainfall Amounts in Canada, 1960-1980", xlab = "Rainfall")
```



The histogram above is more reasonable because there no more negative rainfall amounts.

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)
```

Because the elements of x are characters, these commands don't give numerically correct answers. The command max(x) returns "7" because "7" is the largest starting character, compared to "5" and the "1" in "12". The command sort(x) returns the list {"12", "5", "7"}, which is what the elements of x are in ascending order. The "1" in "12" is less than "5" and "7", so "12" is listed first. The command sum(x) returns an error because the elements of x are not numeric, so it is not possible to add them together.

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]
```

These commands produce errors because the elements of y are all characters, making it impossible to add them together.

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]
```

These commands produce a result of 19 at the end because z[1,2] references the value 7 and z[1,3] references the value 12. Both z[1,2] and z[1,3] are numeric because z2 and z3 were assigned numeric values in the data frame, so it possible to add them together.

3. (3 pts, equally weighted).

a.) What is the point of reproducible code?

Reproducible code is meant to provide a way to replicate the results you observed. If code is reproducible, both you and others will be able to run the same program and get the same output. It provides proof that your process is legitimate, and allows others to see how your results came about.

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

In a group project, if I write code that has elements of randomness, it is very difficult or impossible for group members to reproduce my results without using the same random numbers. This can be annoying if other group members have work that depend on what my results are. If my code is reproducible, other group members will have fixed results to work from, rather than variable ones.

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.

This assignment was a 4.