STA 360/602: Assignment 1, Spring 2019

Kuei-Yueh (Clint) Ko

Due Tuesday, 15 January 2019, 10 AM, Sakai

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.

Syllabus: (https://github.com/resteorts/modern-bayes/blob/master/syllabus/syllabus-sta602-spring19.pdf)

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.

```
library(tidyverse)
```

Working with data

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?

```
dat_dir = "./data"
rain.df = read_table(file.path(dat_dir, "rnf6080.dat"), col_names = FALSE)
head(rain.df)
```

```
## # A tibble: 6 x 27
##      X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12
##   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1    60     4     1     0     0     0     0     0     0     0     0     0
## 2    60     4     2     0     0     0     0     0     0     0     0     0
## 3    60     4     3     0     0     0     0     0     0     0     0     0
## 4    60     4     4     0     0     0     0     0     0     0     0     0
## 5    60     4     5     0     0     0     0     0     0     0     0     0
## 6    60     4     6     0     0     0     0     0     0     0     0     0
## # ... with 15 more variables: X13 <int>, X14 <int>, X15 <int>, X16 <int>,
## #   X17 <int>, X18 <int>, X19 <int>, X20 <int>, X21 <int>, X22 <int>,
## #   X23 <int>, X24 <int>, X25 <int>, X26 <int>, X27 <int>
```

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.)

There are 5070 rows and 27 columns.

```
dim(rain.df)
```

```
## [1] 5070   27
```

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

```
colnames(rain.df)
```

```
##  [1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10" "X11"
## [12] "X12" "X13" "X14" "X15" "X16" "X17" "X18" "X19" "X20" "X21" "X22"
## [23] "X23" "X24" "X25" "X26" "X27"
```

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

```
rain.df[2, 4]
```

```
## # A tibble: 1 x 1
##      X4
##   <int>
## 1     0
```

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

```
rain.df[2, ]
```

```
## # A tibble: 1 x 27
##      X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12
##   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1    60     4     2     0     0     0     0     0     0     0     0     0
## # ... with 15 more variables: X13 <int>, X14 <int>, X15 <int>, X16 <int>,
## #   X17 <int>, X18 <int>, X19 <int>, X20 <int>, X21 <int>, X22 <int>,
## #   X23 <int>, X24 <int>, X25 <int>, X26 <int>, X27 <int>
```

f. What does the following command do?

This command assigned the column names to the dataframe.

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

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

```
##  [1] "year"  "month" "day"   "0"     "1"     "2"     "3"     "4"
##  [9] "5"     "6"     "7"     "8"     "9"     "10"    "11"    "12"
## [17] "13"    "14"    "15"    "16"    "17"    "18"    "19"    "20"
## [25] "21"    "22"    "23"
```

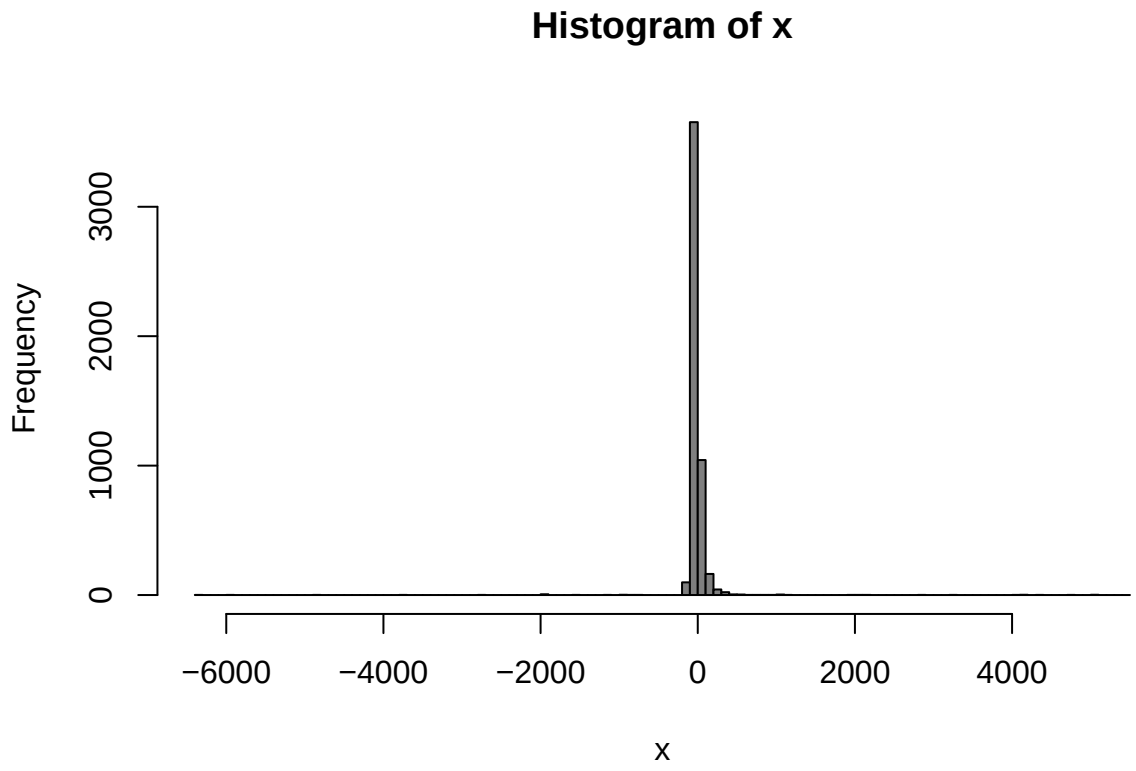
g. Create a new column called `daily`, which is the sum of the 24 hourly columns.

```
rain.df2 = rain.df
rain.df2["dialy"] = apply(rain.df[, -c(1:3)], 1, sum)
colnames(rain.df2)
```

```
##  [1] "year"  "month" "day"   "0"     "1"     "2"     "3"     "4"
##  [9] "5"     "6"     "7"     "8"     "9"     "10"    "11"    "12"
## [17] "13"    "14"    "15"    "16"    "17"    "18"    "19"    "20"
## [25] "21"    "22"    "23"    "dialy"
```

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.

```
x = rain.df2$dialy
hist(x,
     col = "grey50",
     breaks = seq(from = -6400, to = 5500, by = 100))
```



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

There should not be negative value in the data of hourly rainfall.

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

First observe what are the values of negative number

```
tmp = do.call(c, rain.df2)
idx = which(tmp < 0)
unique(tmp[idx])
```

```
##  [1]  -999 -5986 -1944 -1929  -751 -3795  -108 -6399 -2786  -801  -932
## [12] -1104 -1914 -4896 -1520 -1998
```

assign zero to all negative values

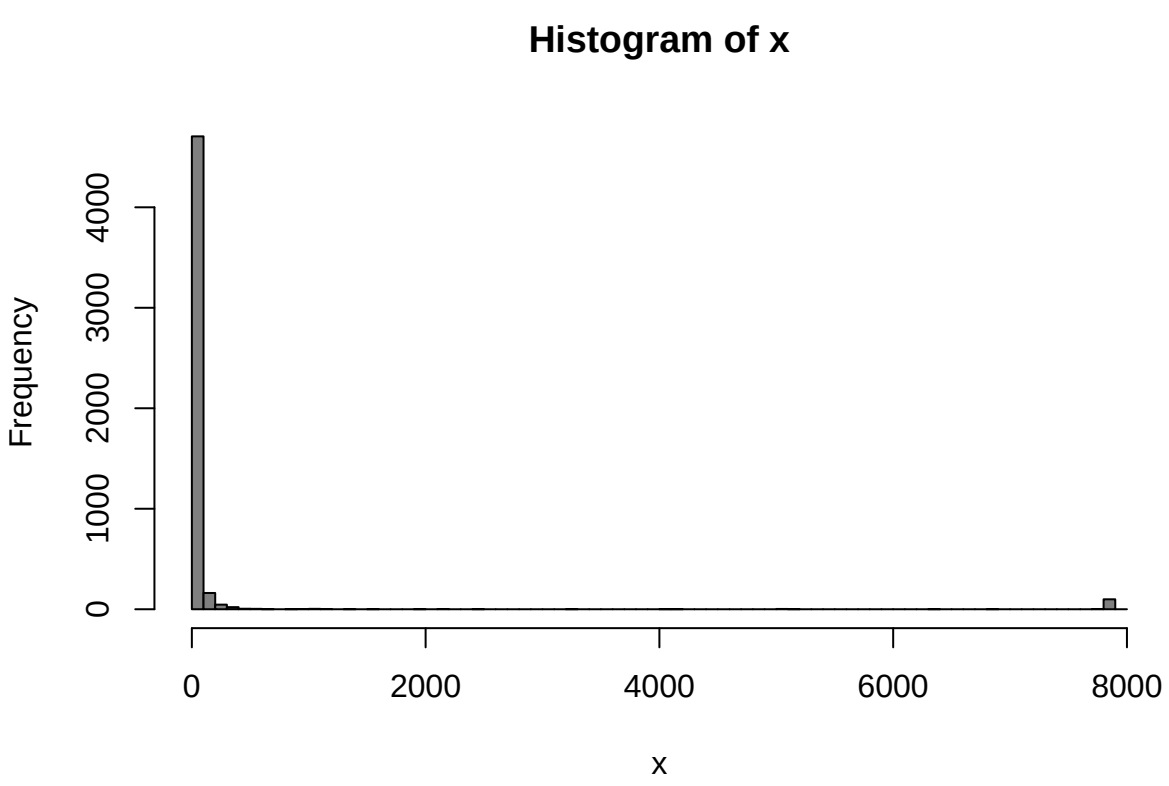
```
### assign negative values into zero
rain.df2 = rain.df
idx = rain.df < 0
rain.df2[idx] = 0

### calculate daily rainfall
rain.df2["daily"] = apply(rain.df2[, -c(1:3)], 1, sum)
```

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.

The figure below is more reasonable because now all the amounts of daily rainfall are positive.

```
### replot the histogram of daily rainfall
x = rain.df2$daily
hist(x,
     col = "grey50",
     breaks = seq(from = 0, to = 8000, by = 100))
```



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)
```

Since the values in vector x are characters, the values are compared alphabetically, i.e. "2" > "11" > "100". Therefore, the output of max(x) is "7" and sort(x) is c("12", "5", "7"). However, it is impossible to sum character values (unless you sum the ascii code of the characters). As a result, sum(x) would return error.

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]
```

It produces error because R do not have the method to sum character values. As the vector c("5",7,12) is created, 7 and 12 is coerced into "7" and "12" because a simple vector in R does not allow more than one data type element.

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 output value would be 7 + 12 = 19. In R, a list or dataframe allows multiple elements with different data structure. Therefore, unlike the previous problem where type conversion occur, 7 and 12 remain as numeric.

3. (3 pts, equally weighted).

a.) What is the point of reproducible code?

Writing reproducible code is important so that any researcher (including the author) could reproduce the results of data analysis. If a data analysis process is not reproducible, other may question whether the research findings are true or not.

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 the code I wrote could not be executed by others, other researchers would not be able to recreate my results and check if my report is correct

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.

I think the this assignment is easy (1) since I already had the background in R programming. However, it is always great to review the important concept of the language.