Data Wrangling in `R`

Garrett Allen

Due Friday August 27, 5 PM EDT

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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?

```
rain.df <- read.delim("data/rnf6080.dat",sep = "",header = FALSE)
```

I used the read.delim command to read in the .dat file, with sep = "", since this was the format of the .dat file. I additionally had to specify there was a header row with the labels for the columns.

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
rowcol = c(nrow(rain.df),ncol(rain.df))
rowcol
```

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

I used the commands nrow and ncol, which return the number of rows of the input dataframe and number of columns, respectively.

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

```r
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"
```

The name of the ith column is V_i, where i ranges from 1 to 27. To do this, I used the function colnames(x), which returns a list of the names of the columns in the dataframe x.

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

```r
val2 = rain.df[2,4]
val2
```

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

You would use the command df[i,j] to retrieve the value in the ith row and jth column. df[2,4] = 0.

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

```r
row2 = rain.df[2,]
row2
```

```
##   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
```

You would use the command rain.df[i,] to get the contents of the ith row. THe content of rain.df[2,] is printed above.

f. What does the following command do?

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

It renames the first three columns to "year", "month", and then "day", and then the next 24 columns are renamed the numbers from 0 to 23.

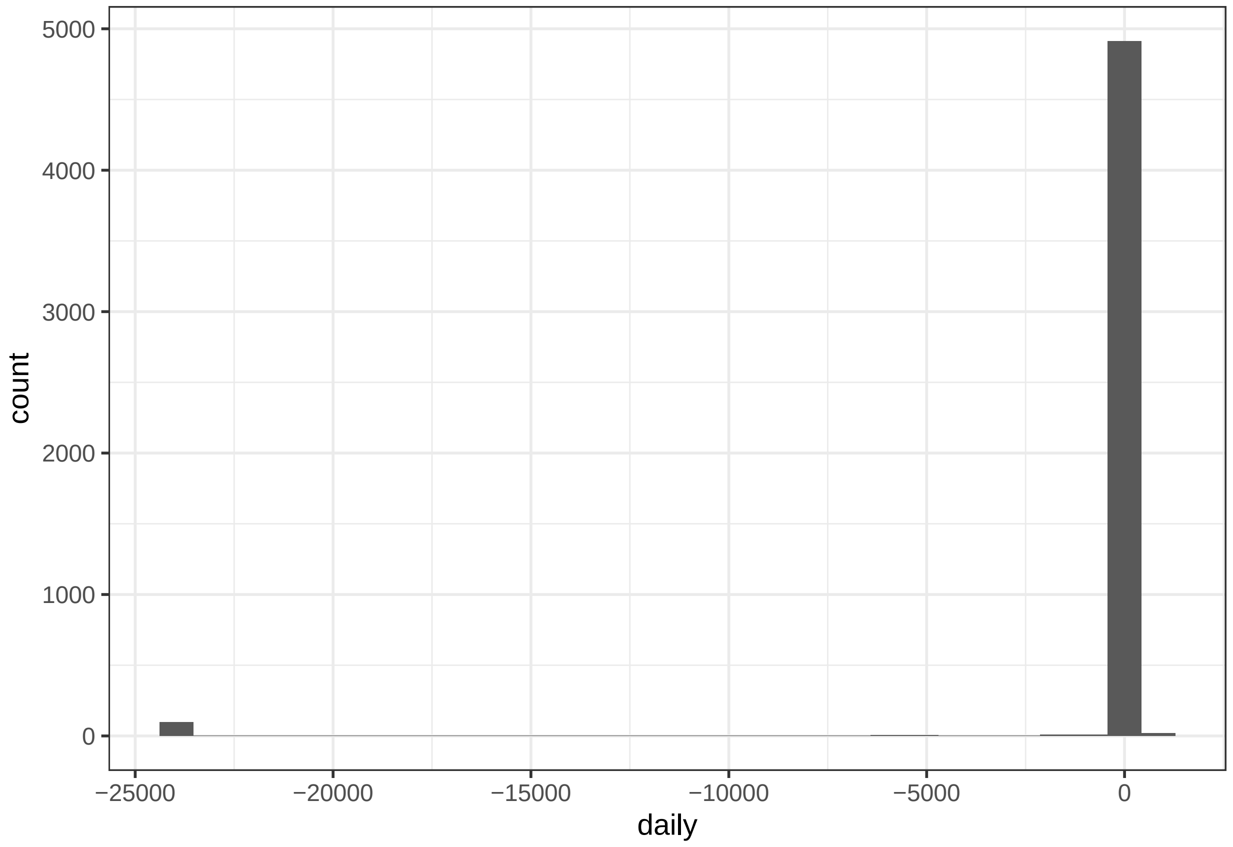
g. Create a new column called `daily`, which is the sum of the 24 hourly columns.

```r
subset = rain.df[,4:27]
rain.df$daily = rowSums(subset)
```

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
rain.df %>%
  ggplot(aes(x = daily)) +
  geom_histogram() +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



i. Explain why that histogram above cannot possibly be right. It cannot possibly be right because there are values in the dataset that have a negative value for total rainfall. This obviously does not make sense, as rainfall for a given day should be greater than or equal to zero.

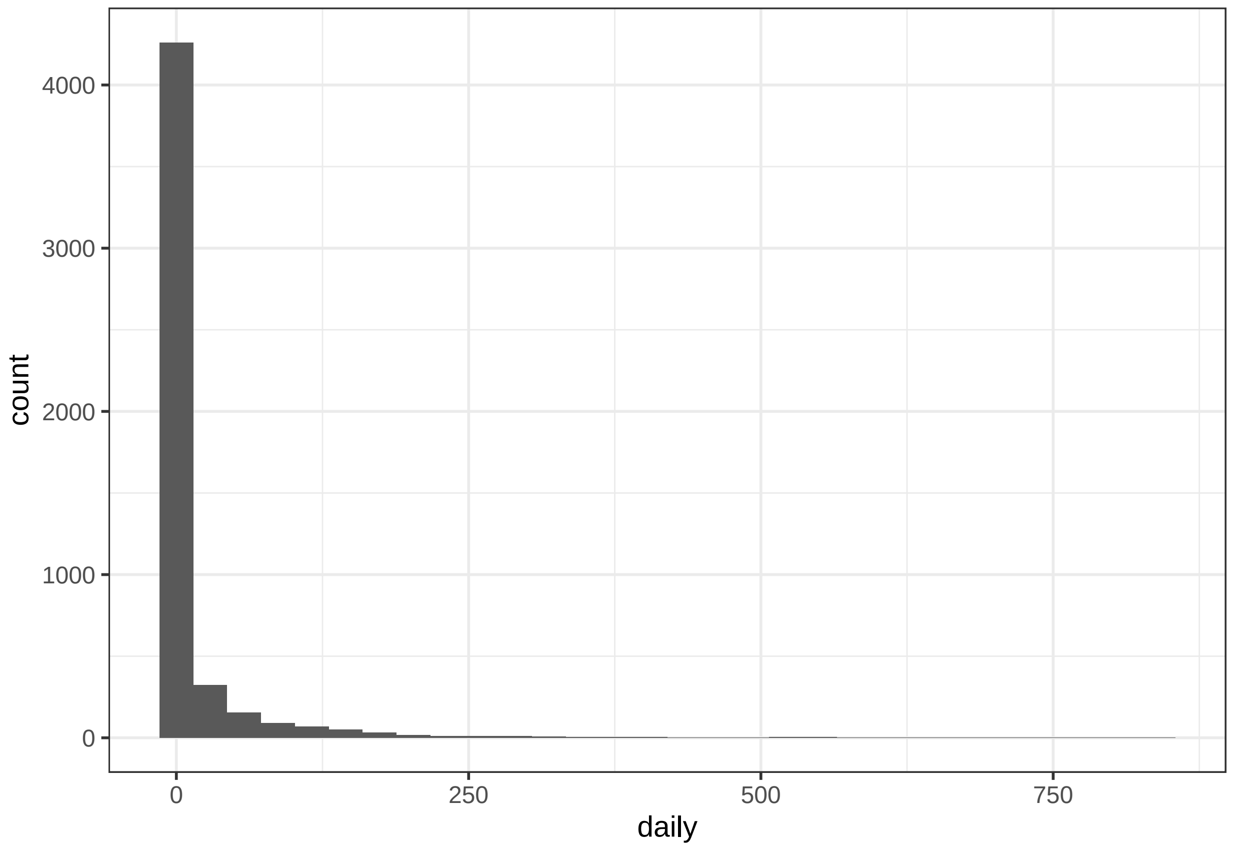
j. Give the command you would use to fix the data frame.

```
rain.df[rain.df < 0] <- 0
```

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.

```
rain.df %>%
  ggplot(aes(x = daily)) +
  geom_histogram() +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



It is a much more reasonable histogram because it includes no negative values of rainfall, and because it has a distribution we would expect; specifically, right-skewed. This is because most days we would expect no rain, some days would have a little bit of rain, and a few days would have an incredible amount of rain over this period. We could have instead replaced daily values with the mean value of the prior histogram (without the negatives) but I decided to put zero instead given the large skew, as I wanted to ensure the shape of the distribution remained roughly similar instead of overemphasizing the mean.

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)
```

For the 1st command (x <- c("5","12","7"))) there are no errors and it creates a vector of characters. None of the values here are numeric, so certain numeric commands won't work as expected.

For the 2nd command (max(x)) it returns "7", which is returning the max in lexographic order since the entries of the vector are characters. That is, since "7" comes last alphabetically (after 1 and 5), that is the maximum.

For the 3rd command, it again sorts the characters in lexographic order. Specifically, in the order ("12", "5", "7") since "1" < "5" < "7" in lexographic order.

For the 4th command, it does error; this is because it does not make sense to sum character values, only numeric values. Therefore it returns a type 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]
```

On the first command, there are no errors. When R encounters a vector initialized with several different types of values (numeric, character, etc.) it coerces all of them to characters. So even though 7 and 12 are inputted as numeric values, they are characters because "5" is.

This leads to an error on the second command (y[2] + y[3]) because you cannot sum characters with a binary operator like addition. Thus there is an error, as you try to add non-numeric datatypes.

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]
```

There are no errors on either command here. In the first command, z is initialized as a dataframe where column z1 is a character vector with value "5" in the first entry, z2 is a numeric vector with 7 in the first entry, and z3 is a numeric vector with 12 in the first entry.

The second command does not error either as a result; since z[1,2] + z[1,3] is adding two numeric values, it returns 19, which is 7 + 12 as expected.

3. (3 pts, equally weighted).

a.) What is the point of reproducible code?

Reproducible code is firstly important for transparency; if another person comes along and wants to see why your analysis makes sense, they should also be able to run your code and verify. This leads to the second reason reproducible code is important; if it is not reproducible, then future people trying to modify or work with your code will be unable to, even if your results are correct. Future data scientists should be able to use and analyze your code hassle free, both to verify and expand on your work, and without reproducible code, neither of these goals can be met.

b.) Given an example of why making your code reproducible is important for you to know in this class and moving forward. It is important to know in this class so that I can easily collaborate with classmates, TAS, or Professor Steorts if needed. It would be difficult to work on the homework or understand the material with others if my code only worked on my machine via hard coding, rather than being easily transferable via Github. Additionally, since I plan to continue into grad school, I know I will need to produce reproducible code so that other academics can verify my results when I conduct research. It will also be important when I collaborate in grad school; other people cannot build on my work or work with me if my code is not reproducible.

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. 3