Homework 1

Nathan Yang

8-30-2024

Exercise 1

a. Load the Data using the read.table() function

```
rain_df <- read.table(file = "data/rnf6080.dat")
```

b. I can use `dim(rain_df)` to get the number of rows and columns

```
dim(rain_df)
```

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

From this, I know that `rain_df` has 5070 rows and 27 columns

c. I can use `colnames(rain_df)` to get all the column names

```
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. I can index the dataframe to get the value in the 2nd row 4th column

```
rain_df[2, 4]
```

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

e. I can display the whole second row by indexing it with the column value removed

```
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. This command renames the column names of rain_df to be “year”, “month”, “day”, and the numbers 0 through 23

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

g. Adding a column that aggregates daily rainfall

```
if (!require("dplyr")) {
  install.packages("dplyr")
}
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(dplyr)

rain_df <- rain_df |>
  mutate(
    daily_rain_fall = rowSums(select(rain_df, one_of(as.character(0:23))))
  )
```

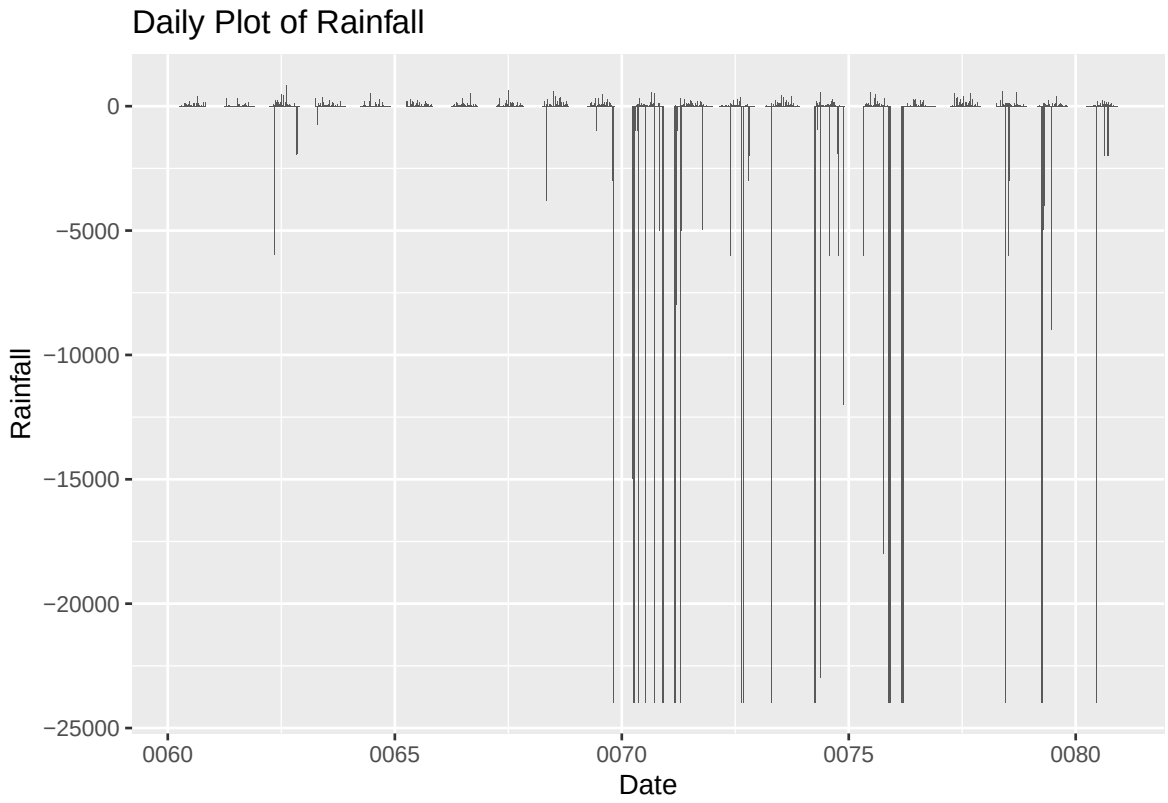
h. Creating a histogram

```r
if (!require("ggplot2")) {
  install.packages("ggplot2")
  library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```r
library(ggplot2)

ggplot(rain_df) +
  geom_col(
    aes(
      x = as.Date(paste(year, month, day, sep = "-")),
      y = daily_rain_fall
      )
    ) +
  labs(x = "Date", y = "Rainfall", title = "Daily Plot of Rainfall")
```



i. This histogram can't be right because there can't exist days with negative rainfall

j. Here is how I would "fix" the dataframe.
First, I needed to look at the rows that had negative values for `daily_rain_fall`

```
weird_rows <- rain_df |>
  filter(daily_rain_fall < 0)
tibble(weird_rows)
```

```
## # A tibble: 139 x 28
##     year month   day   `0`   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`
##    <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
##  1    62     5     7     0     0     0     0     0     0     0     0     0     0
##  2    62    11     3     0     0     0     0     0     0     0     0     0     0
##  3    62    11     6     0     0     0     3    10    15    23     8    10     0
##  4    63     4    16     0     0     5     0    18     0    13     3    13    18
##  5    68     5     8     3     0     3    13     3     5    20    20    38    43
##  6    69     6    12     0     0     0     0     0     0     0  -999     5     0
##  7    69    10    15  -999     0     0     0     0     0     0     0     0     0
##  8    69    10    30  -999  -999  -999  -999  -999  -999  -999  -999  -999  -999
##  9    70     4     1     0     0     0     0     0     0     0     0     0  -999
## 10    70     4     2  -999  -999  -999  -999  -999  -999  -999  -999  -999  -999
## # i 129 more rows
## # i 15 more variables: `10` <int>, `11` <int>, `12` <int>, `13` <int>,
## #   `14` <int>, `15` <int>, `16` <int>, `17` <int>, `18` <int>, `19` <int>,
## #   `20` <int>, `21` <int>, `22` <int>, `23` <int>, daily_rain_fall <dbl>
```

Next, I need to replace all the all the "-999" values since they are obviously incorrect. Here I chose to replace them all with 0s

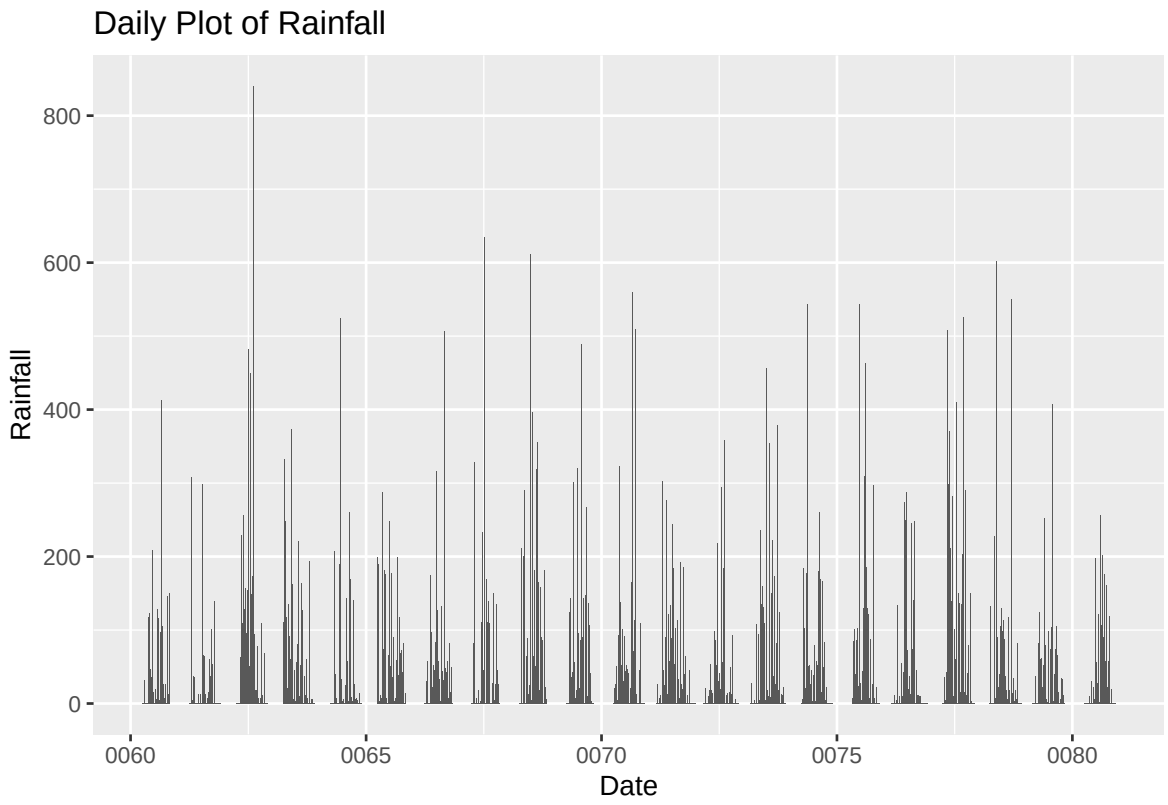
```
imputed_df <- rain_df |>
  mutate_all(~ replace(., . == -999, 0))
```

Next I would need to recalculate the daily rainfall sums

```
corrected_df <- imputed_df |>
  mutate(
    daily_rain_fall = rowSums(select(imputed_df, one_of(as.character(0:23))))
  )
```

k. Here is the regenerated histogram

```
ggplot(corrected_df) +
  geom_col(
    aes(
      x = as.Date(paste(year, month, day, sep = "-")),
      y = daily_rain_fall
      )
    ) +
  labs(x = "Date", y = "Rainfall", title = "Daily Plot of Rainfall")
```



This is more reasonable than the previous histogram because there are no more negative values and the range of values is much more in line with expected rainfall.

Exercise 2

a. The result looks strange because the numbers are treated as strings here.
`max(x)` returns "7" due to it being the latest alphanumerically
`sort(x)` returns `["12", "5", "7"]` as this is the alphanumeric order

`sum(x)` is an error since strings cannot be added together by `sum()`

b. The `+` operation produces an error because one variable being a string determined `y` to be a vector of strings

c. The `+` operation works correctly because the 2 referenced values were numbers. Only the first value was determined to be a string while the second and third values were added.

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

```
## [1] 19
```

Exercise 3

a. The point of reproducible code is to ensure other people are able to confirm my work/findings as well as demonstrate a complete workflow.

b. An example of why reproducible code is so important is to verify my methods were correct, consistent, and statistically acceptable.

c. I rate this assignment in terms of difficulty a 6/10 since I struggled with correcting and aggregating the daily rainfall