CSI 2300: Intro to Data Science

In-Class Exercise 03: Introduction to R and RStudio

The data for today's exercises are the Colorado Covid-19 data used in the lecture.

1. Download the data, and then load it into R. To verify that this has been accomplished, show the column names of the data frame, using the `colnames()` command.

```
df <- read.csv("dat/CDPHE_COVID19_Wastewater_Dashboard_Data.csv")
colnames(df)
# [1] "Date"                              "Utility"
# [3] "SARS_CoV_2_copies_L"               "Number_of_New_COVID19_Cases_by_"
# [5] "ObjectId"
```

2. Let's do some light data wrangling of this dataset. First, remove the redundant last column, and overwrite the dataset name with this new dataset containing 4 instead of 5 columns. Show the first 6 rows of the updated dataset to demonstrate.

```
df <- df[,-5]
head(df)
#         Date                      Utility SARS_CoV_2_copies_L
# 1 08/15/2020 Metro Wastewater RWHTF - PRC                  NA
# 2 08/11/2020                   Broomfield                  NA
# 3 08/15/2020                   Northglenn                  NA
# 4 08/11/2020     CO Springs - JD Phillips                  NA
# 5 08/11/2020       CO Springs - Las Vegas                  NA
# 6 08/15/2020                       Pueblo                  NA
#   Number_of_New_COVID19_Cases_by_
# 1                              36
# 2                               0
# 3                               0
# 4                               6
# 5                              22
# 6                               5
```

3. The names of the last two columns could be better. Replace the existing names with the names `sars_rna_copies` and `new_covid_cases`.

```
colnames(df)[c(1,4)]
# [1] "Date"                              "Number_of_New_COVID19_Cases_by_"
colnames(df)[1:4]
# [1] "Date"                              "Utility"
# [3] "SARS_CoV_2_copies_L"               "Number_of_New_COVID19_Cases_by_"
colnames(df)[4]
# [1] "Number_of_New_COVID19_Cases_by_"
#colnames(df)[1,4] #commas separate dimensions of our data objects
```

```
length(colnames(df)) #lenght tells us size of vectors
# [1] 4
#dim(colnames(df)) #useless, dim only works for data frames
dim(df)
# [1] 3498    4

colnames(df)[3] <- "sars_rna_copies"
colnames(df)[4] <- "new_covid_cases"

#alternatives ways to do the same thing:
colnames(df)[3:4] <- c("sars_rna_copies","new_covid_cases")
```

4. How many missing values are there in the `sars_rna_copies` variable? What proportion of the dataset is this?

```
summary(df) #tells us we have 2,647 missing values for sars rna
#      Date             Utility          sars_rna_copies  new_covid_cases
#  Length:3498        Length:3498        Min.   :     0   Min.   :  0.00
#  Class :character   Class :character   1st Qu.: 16366   1st Qu.:  0.00
#  Mode  :character   Mode  :character   Median : 48592   Median : 12.00
#                                        Mean   : 83639   Mean   : 44.68
#                                        3rd Qu.:122190   3rd Qu.: 43.00
#                                        Max.   :822054   Max.   :913.00
#                                        NA's   :2647
2647/nrow(df)
# [1] 0.7567181
```

5. Another issue with the data is that when the count of new cases of Covid-19 is less than 5, the count of new cases is reported as 0 to maintain patient privacy. Filter the data so that only non-NA `sars_rna_copies` are present AND all `new_covid_cases` are 5 or greater. Show the first few rows of this new data frame to demonstrate that you filtered out the undesirable rows.

```
covid_filter <- df[(is.na(df$sars_rna_copies)==FALSE) & (df$new_covid_cases>=5) , ]
head(df)
#         Date                      Utility sars_rna_copies new_covid_cases
# 1 08/15/2020 Metro Wastewater RWHTF - PRC              NA              36
# 2 08/11/2020                   Broomfield              NA               0
# 3 08/15/2020                   Northglenn              NA               0
# 4 08/11/2020     CO Springs - JD Phillips              NA               6
# 5 08/11/2020       CO Springs - Las Vegas              NA              22
# 6 08/15/2020                       Pueblo              NA               5
```

6. Let's do a simple plot of the `new_covid_cases` versus `sars_rna_copies` using the filtered data. Given that we expect the number of new cases to depend on the RNA copies measured, put `new_covid_cases` on the y-axis and `sars_rna_copies` on the

x-axis. Comment on what you observe in this plot.

7. Add nicer labels to the plot by including the arguments `xlab="X Label"` and `ylab="Y Label"` and `main="Overall Title"` in the plot command. Change the labels to something appropriate for this figure.
8. A lot of the points are squished into the bottom left of the figure. They can be spread apart to see the relationship between the two variables more clearly by applying the log to each of the variables. Replot the figure applying the `log()` function to each variable.
9. Describe what you see in the figure from the prior question.
10. Now, let's go back to the full dataset and examine the new case counts in one county, Boulder county. First, filter the data to obtain just the Boulder utility's observations. Then, sort the `new_covid_cases` from smallest to largest. What do you observe?
11. Now we want to plot the new covid cases for Boulder over time, similar to the website where the data are reported[1]. To do this, we want to plot the date on the x-axis and the number of new cases on the y-axis with the following additional instructions:

- install and load the `lubridate` library
- wrap `covid_boulder$Date` with the `mdy()` command from the `lubridate` library, which can then be used as the variable to plot on the x-axis.
- inside the `plot()` command, add the argument `type="l"`
- add sensible labels to x and y axes

12. The lines in the prior plot should not be criss-crossing over themselves. What is the cause of this problem? See if you can fix it. You may find the `order()` command to be useful.
13. What patterns do you observe in the plot from the prior question?

[1] https://cdphe.maps.arcgis.com/apps/opsdashboard/index.html#/d79cf93c3938470ca4bcc4823328946b