

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)[3:4] <- c("sars_rna_copies", "new_covid_cases")

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  |

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

```
sum(is.na(df$sars_rna_copies))
# [1] 2647

nrow(df)
# [1] 3498

sum(is.na(df$sars_rna_copies))/nrow(df)*100
# [1] 75.67181
```

*The percentage of missing values is 75.672%*

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.

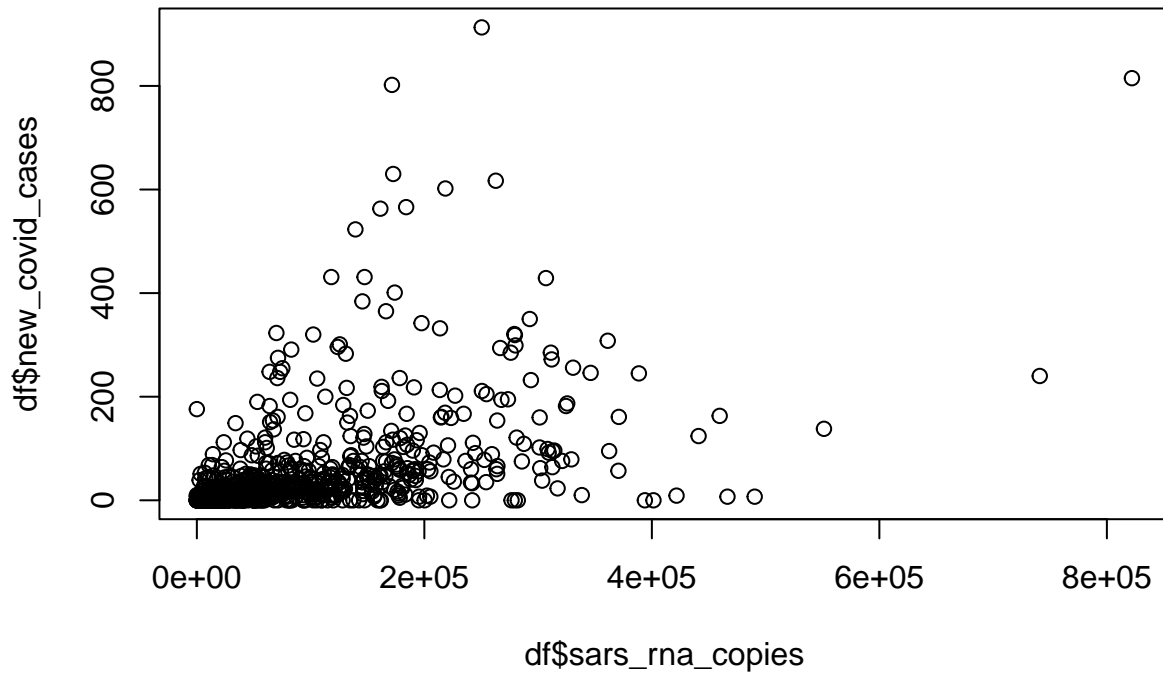
```
df_filtered <- df[!is.na(df$sars_rna_copies) & df$new_covid_cases >= 5, ]

head(df_filtered)
```

| #    | Date       | Utility                      | sars_rna_copies | new_covid_cases |
|------|------------|------------------------------|-----------------|-----------------|
| # 16 | 08/06/2020 | Metro Wastewater RWHTF - CC  | 17308.9         | 40              |
| # 20 | 08/06/2020 | Metro Wastewater RWHTF - PRC | 7078.3          | 53              |
| # 23 | 08/02/2020 | CO Springs - JD Phillips     | 24177.4         | 5               |
| # 29 | 08/02/2020 | CO Springs - Las Vegas       | 50393.5         | 15              |
| # 31 | 08/06/2020 | Pueblo                       | 17677.7         | 6               |
| # 41 | 08/06/2020 | South Adams County           | 0.0             | 7               |

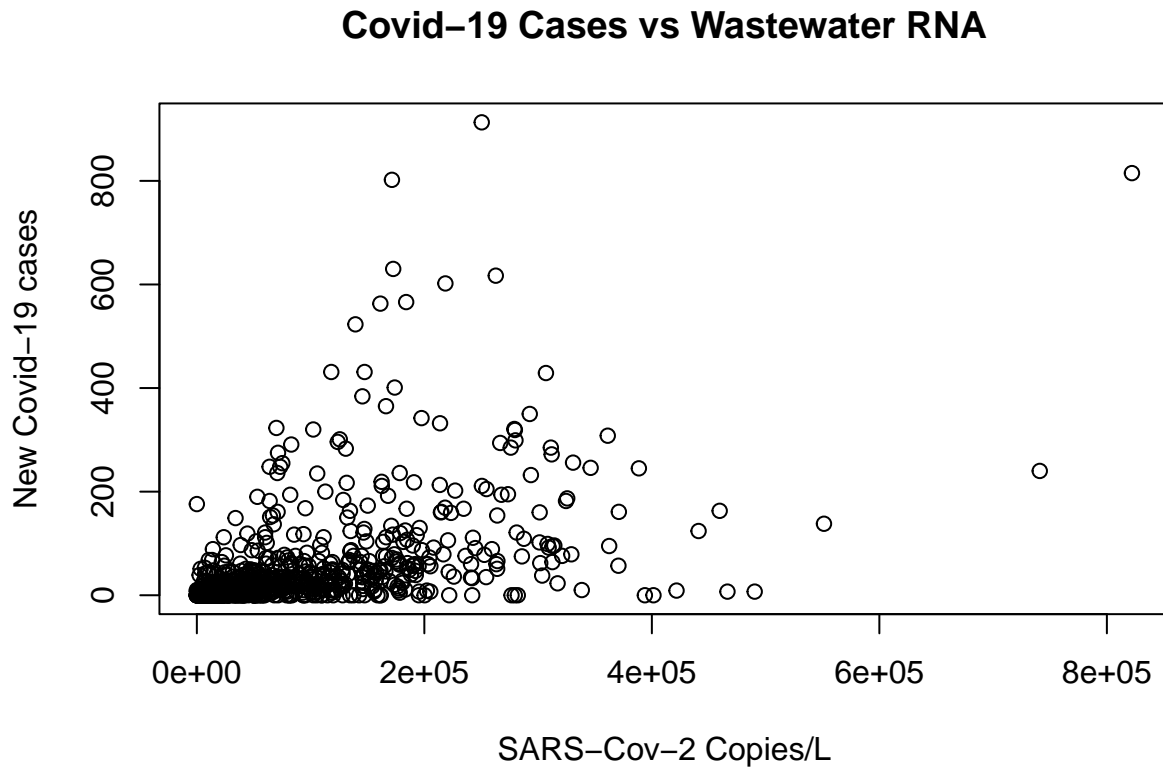
6. Let's do a simple plot of the `new_covid_cases` versus `sars_rna_copies` using the filtered data. Given that we expect the number of new cases to depend on the RNA copies measured, put `new_covid_cases` on the y-axis and `sars_rna_copies` on the x-axis. Comment on what you observe in this plot.

```
plot(df$sars_rna_copies, df$new_covid_cases)
```



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.

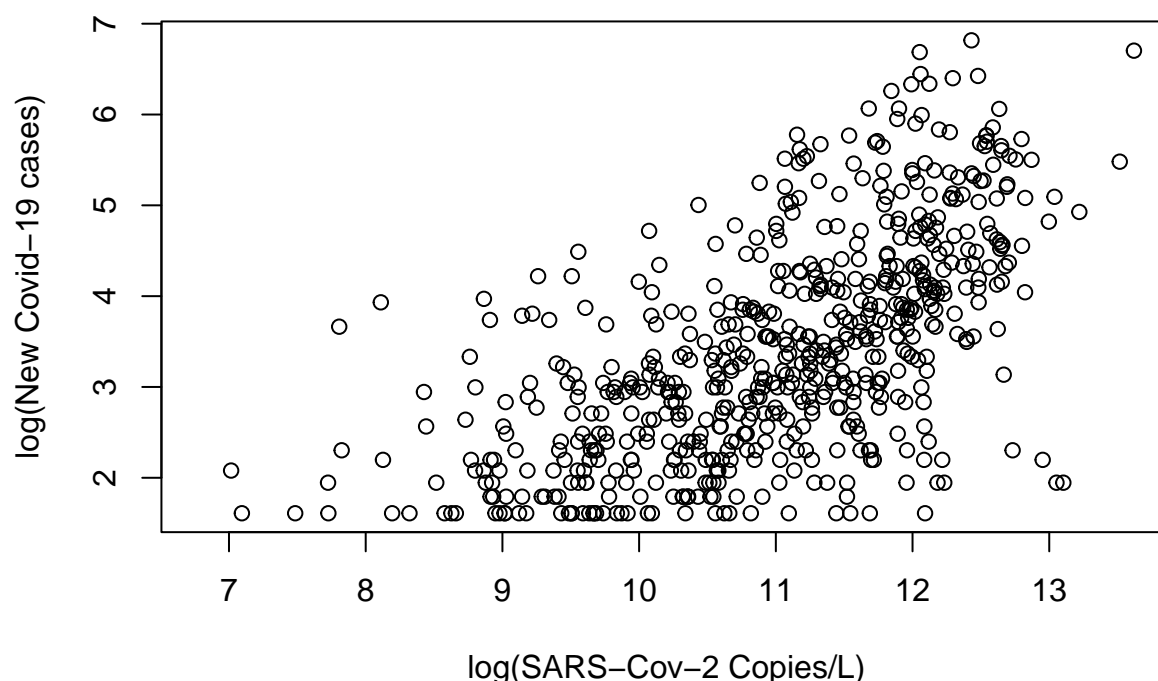
```
plot(df$sars_rna_copies, df$new_covid_cases,  
     main = "Covid-19 Cases vs Wastewater RNA",  
     xlab = "SARS-Cov-2 Copies/L",  
     ylab = "New Covid-19 cases")
```



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.

```
plot(log(df$sars_rna_copies), log(df$new_covid_cases),  
     main = "Covid-19 Cases vs Wastewater RNA",  
     xlab = "log(SARS-Cov-2 Copies/L)",  
     ylab = "log(New Covid-19 cases)")
```

## Covid-19 Cases vs Wastewater RNA



9. Describe what you see in the figure from the prior question.

*The linear-ish plotting showing a a positive correlation between wastewater SARS-CoV-2 RNA levels and new Covid-19 cases. As the RNA concentration increases, the number of new Covid-19 cases also tends to increase.*

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?

```
covid_boulder <- df[df$Utility == "Boulder",]
sort(covid_boulder$new_covid_cases)
```

|   |       |    |    |    |    |    |    |    |    |    |    |    |     |     |     |     |    |    |
|---|-------|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|----|----|
| # | [1]   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0  | 0  |
| # | [19]  | 0  | 5  | 5  | 5  | 5  | 5  | 5  | 6  | 6  | 6  | 6  | 7   | 7   | 7   | 7   | 8  | 8  |
| # | [37]  | 8  | 8  | 9  | 9  | 9  | 9  | 10 | 10 | 10 | 10 | 10 | 10  | 11  | 12  | 12  | 12 | 13 |
| # | [55]  | 13 | 13 | 14 | 14 | 14 | 15 | 15 | 15 | 15 | 16 | 16 | 16  | 16  | 17  | 17  | 17 | 18 |
| # | [73]  | 19 | 19 | 19 | 20 | 20 | 20 | 20 | 20 | 21 | 21 | 21 | 21  | 21  | 22  | 22  | 22 | 24 |
| # | [91]  | 24 | 24 | 25 | 26 | 27 | 27 | 27 | 29 | 30 | 30 | 30 | 30  | 31  | 31  | 31  | 31 | 33 |
| # | [109] | 34 | 34 | 35 | 35 | 35 | 35 | 36 | 37 | 38 | 40 | 41 | 41  | 43  | 44  | 46  | 46 | 47 |
| # | [127] | 47 | 48 | 52 | 53 | 53 | 55 | 55 | 55 | 56 | 57 | 58 | 58  | 58  | 58  | 58  | 60 | 61 |
| # | [145] | 61 | 62 | 64 | 64 | 66 | 69 | 76 | 78 | 79 | 82 | 99 | 101 | 121 | 127 | 148 |    |    |

11. Now we want to plot the new covid cases for Boulder over time, similar to the website where the data are reported<sup>1</sup>. 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:

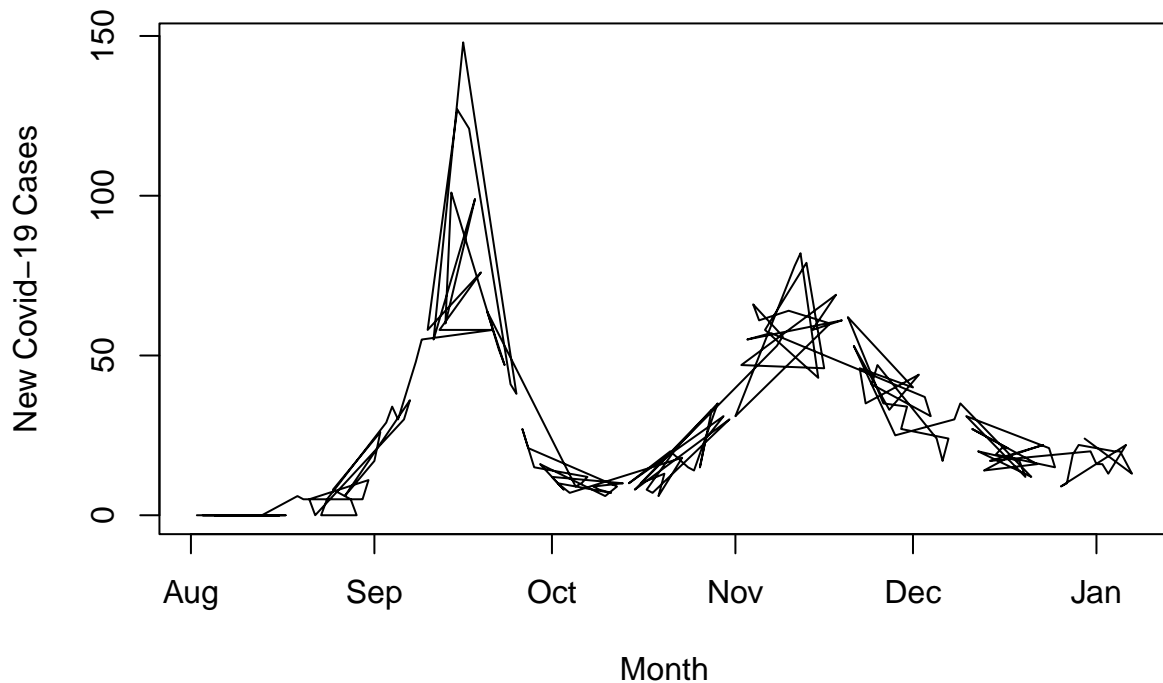
- install and load the `lubridate` library
- wrap `covid_boulder$Date` with the `mdy()` command from the `lubridate` library, which can then be used as the variable to plot on the x-axis.
- inside the `plot()` command, add the argument `type="l"`
- add sensible labels to x and y axes

```
library(lubridate)
#
# Attaching package: 'lubridate'
# The following objects are masked from 'package:base':
#
#   date, intersect, setdiff, union
covid_boulder$Date <- mdy(covid_boulder$Date)
plot(covid_boulder$Date, covid_boulder$new_covid_cases,
     main = "Covid-19 Cases Over Time in Boulder County",
     xlab = "Month",
     ylab = "New Covid-19 Cases",
     type = "l")
```

---

<sup>1</sup><https://cdphe.maps.arcgis.com/apps/opsdashboard/index.html#/d79cf93c3938470ca4bcc4823328946b>

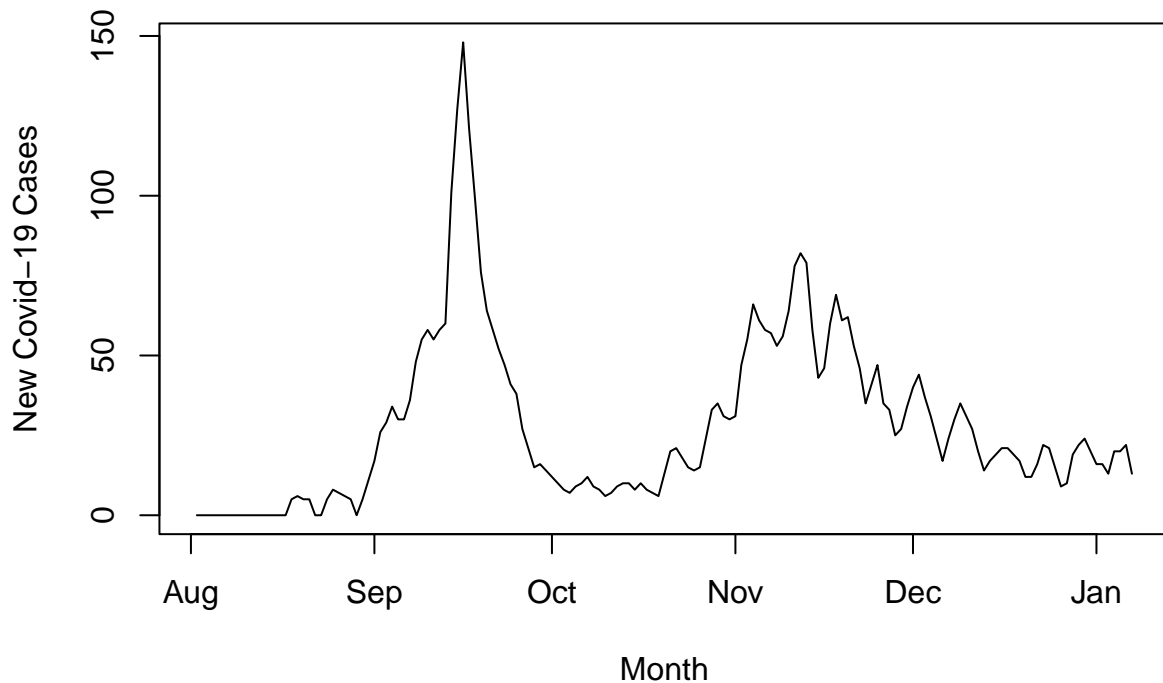
## Covid-19 Cases Over Time in Boulder County



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.

```
new_index <- order(covid_boulder$Date)
plot(x = covid_boulder$Date[new_index],
     y = covid_boulder$new_covid_cases[new_index],
     type = "l",
     main = "Covid-19 Cases Over Time in Boulder County",
     xlab = "Month",
     ylab = "New Covid-19 Cases")
```

### Covid-19 Cases Over Time in Boulder County



13. What patterns do you observe in the plot from the prior question?

*The number of new COVID-19 cases starts off relatively low, so it started with minimal spread. There is a significant spike in cases, peaking in early October. After the peak, the number of cases drops, possibly due to public health interventions. After the drop there is another more gradual rise in November with causing another peak, then the case numbers steadily decline after December and appear to stabilize at a lower level.*