Sars vs. COVID-19

Melodie Irvin, mki77

Virology has always excited me, and with the coronavirus looming around everywhere we go, I found it compelling to compare the current data regarding the deaths and confirmed cases of COVID-19 with it's sister virus, SARS-CoV from 2003. Considering they are of the same genus and species of virus, I was expecting to find similar results regarding the number of deaths and rate of infection. As you will see, that is not at all the case. The Sars dataset was gathered from Kaggle while the COVID dataset was taken from ourworldindata.org. Both were combined in Excel prior to importing. The COVID dataset is up-to-date as of March 13, 2020.

```r
options(repos = "https://cran.rstudio.com")
library(dplyr)
library(formatR)
library(ggplot2)
library(readxl)
COVID_and_SARS <- read_excel("C:/Users/Melodie/Desktop/COVID_and_SARS.xlsx")
COVID_and_SARS <- COVID_and_SARS %>% slice(1:70)
COVID <- COVID_and_SARS %>% select(Date.Rank, Sum.of.COVID.Confirmed,
    Sum.of.COVID.Deaths)
SARS <- COVID_and_SARS %>% select(by = -c(Sum.of.COVID.Confirmed,
    Sum.of.COVID.Deaths)) %>% select(Day = Date.Rank, Sum.of.Sars.Deaths,
    Sum.of.Sars.Confirmed)
```

```r
library(tidyverse)
FULL <- full_join(COVID, SARS, by = c(Date.Rank = "Day"))
FULL_2 <- FULL %>% mutate(Week.Number = case_when(between(Date.Rank,
    1, 7) ~ "Week One", between(Date.Rank, 8, 14) ~ "Week Two",
    between(Date.Rank, 15, 21) ~ "Week Three", between(Date.Rank,
        22, 28) ~ "Week Four", between(Date.Rank, 29, 35) ~ "Week Five",
    between(Date.Rank, 36, 42) ~ "Week Six", between(Date.Rank,
        43, 49) ~ "Week Seven", between(Date.Rank, 50, 56) ~
        "Week Eight", between(Date.Rank, 57, 63) ~ "Week Nine",
    between(Date.Rank, 64, 70) ~ "Week Ten", between(Date.Rank,
        71, 77) ~ "Week Eleven", between(Date.Rank, 78, 84) ~
        "Week Twelve", between(Date.Rank, 85, 91) ~ "Week Thirteen",
    between(Date.Rank, 92, 96) ~ "Week Fourteen")) %>% select(Date.Rank,
    Sum.of.COVID.Confirmed, Sum.of.Sars.Confirmed, Sum.of.COVID.Deaths,
    Sum.of.Sars.Deaths, Week.Number)
FULL_New <- FULL_2 %>% arrange(Date.Rank) %>% mutate(COVID.New.Cases = Sum.of.COVID.Confirmed -
    lag(Sum.of.COVID.Confirmed)) %>% mutate(COVID.New.Deaths = Sum.of.COVID.Deaths -
    lag(Sum.of.COVID.Deaths)) %>% mutate(Sars.New.Cases = Sum.of.Sars.Confirmed -
    lag(Sum.of.Sars.Confirmed)) %>% mutate(Sars.New.Deaths = Sum.of.Sars.Deaths -
    lag(Sum.of.Sars.Deaths)) %>% select(Date.Rank, Sum.of.COVID.Confirmed,
    Sum.of.Sars.Confirmed, Sum.of.COVID.Deaths, Sum.of.Sars.Deaths,
    COVID.New.Cases, Sars.New.Cases, COVID.New.Deaths, Sars.New.Deaths,
    Week.Number)
```

Given that this data is tidy, the pivot_longer and pivot_wider fuctions will be used in another section. The object of using those functions, however, is to organize data so that each observation has it's own row and each variable has it's own column. I chose to do a full_join of the data because I imported a single dataset and wanted to ensure I kept all components, therefore, no results were lost. A few additional columns were made for gathering summary statistics as well as to create a categorical variable.

```
FULL_New %>% filter(between(Date.Rank, 25, 50))
```

```
## # A tibble: 26 x 10
##    Date.Rank Sum.of.COVID.Co~ Sum.of.Sars.Con~ Sum.of.COVID.De~ Sum.of.Sars.Dea~
##        <dbl>            <dbl>            <dbl>            <dbl>            <dbl>
##  1        25            49053             3233             1383              144
##  2        26            50580             3298             1526              154
##  3        27            51857             3357             1669              159
##  4        28            71429             3448             1775              165
##  5        29            73332             3595             1873              182
##  6        30            75204             4090             2009              182
##  7        31            75748             4180             2129              217
##  8        32            76769             4561             2247              229
##  9        33            77794             4713             2359              251
## 10        34            78811             4921             2463              263
## # ... with 16 more rows, and 5 more variables: COVID.New.Cases <dbl>,
## #   Sars.New.Cases <dbl>, COVID.New.Deaths <dbl>, Sars.New.Deaths <dbl>,
## #   Week.Number <chr>
```

```
FULL_New %>% select(Sum.of.COVID.Confirmed, Sum.of.Sars.Confirmed,
    everything()) %>% arrange(desc(Sum.of.COVID.Confirmed))
```

```
## # A tibble: 70 x 10
##    Sum.of.COVID.Co~ Sum.of.Sars.Con~ Date.Rank Sum.of.COVID.De~ Sum.of.Sars.Dea~
##               <dbl>            <dbl>     <dbl>            <dbl>            <dbl>
##  1           132758             7624        53             4955              610
##  2           125260             7569        52             4613              597
##  3           118319             7467        51             4292              586
##  4           113702             7445        50             4012              572
##  5           109577             7404        49             3809              551
##  6           105592             7280        48             3584              525
##  7           101927             7178        47             3486              513
##  8            98192             7074        46             3380              505
##  9            95324             6923        45             3280              494
## 10            93090             6800        44             3198              477
## # ... with 60 more rows, and 5 more variables: COVID.New.Cases <dbl>,
## #   Sars.New.Cases <dbl>, COVID.New.Deaths <dbl>, Sars.New.Deaths <dbl>,
## #   Week.Number <chr>
```

```
FULL_New %>% slice(1:53) %>% group_by(Week.Number) %>% summarize(mean_COVNew = mean(COVID.New.Cases,
    na.rm = T), sd_COVNew = sd(COVID.New.Cases, na.rm = T))
```

```
## # A tibble: 8 x 3
##   Week.Number mean_COVNew sd_COVNew
##   <chr>             <dbl>     <dbl>
## 1 Week Eight        5795.     1672.
## 2 Week Five         1129.      563.
## 3 Week Four         4411.     6698.
## 4 Week One           419.      286.
## 5 Week Seven        2947       847.
## 6 Week Six          1374.      402.
## 7 Week Three        3309       423.
## 8 Week Two          2085.      484.
```

```
FULL_New %>% slice(1:53) %>% summarize(median(Sum.of.COVID.Confirmed))
```

```
## # A tibble: 1 x 1
##   `median(Sum.of.COVID.Confirmed)`
##                               <dbl>
## 1                             51857
```

```
FULL_New %>% summarize(median(Sum.of.Sars.Confirmed))
```

```
## # A tibble: 1 x 1
##   `median(Sum.of.Sars.Confirmed)`
##                              <dbl>
## 1                            5212.
```

```
FULL_New %>% slice(1:53) %>% summarize(first(Sum.of.COVID.Deaths),
    last(Sum.of.COVID.Deaths), n_Week = n_distinct(Week.Number))
```

```
## # A tibble: 1 x 3
##   `first(Sum.of.COVID.Deaths)` `last(Sum.of.COVID.Deaths)` n_Week
##                          <dbl>                       <dbl>  <int>
## 1                            6                        4955      8
```

```
FULL_New %>% summarize(first(Sum.of.Sars.Deaths), last(Sum.of.Sars.Deaths),
    n_Week = n_distinct(Week.Number))
```

```
## # A tibble: 1 x 3
##   `first(Sum.of.Sars.Deaths)` `last(Sum.of.Sars.Deaths)` n_Week
##                         <dbl>                      <dbl>  <int>
## 1                           4                        774     10
```

```
FULL_New %>% group_by(Week.Number) %>% summarize(mean_SARNew = mean(Sars.New.Cases,
    na.rm = T), sd_SARNew = sd(Sars.New.Cases, na.rm = T))
```

```
## # A tibble: 10 x 3
##    Week.Number mean_SARNew sd_SARNew
##    <chr>             <dbl>     <dbl>
##  1 Week Eight        52.9      26.5
##  2 Week Five        237.      146.
##  3 Week Four         95.9      53.7
##  4 Week Nine         36.7      17.4
##  5 Week One          59.2      30.6
##  6 Week Seven       126.       19.9
##  7 Week Six         202        65.2
##  8 Week Ten           9.57      6.53
##  9 Week Three       129.      135.
## 10 Week Two         193.      288.
```

```
FULL_New %>% na.omit %>% summarize(cor(Sars.New.Cases, COVID.New.Cases))
```

```
## # A tibble: 1 x 1
##   `cor(Sars.New.Cases, COVID.New.Cases)`
##                                    <dbl>
## 1                                 -0.124
```

```
FULL_New %>% slice(1:53) %>% summarize(mean_CNC = mean(COVID.New.Cases,
    na.rm = T), sd(COVID.New.Cases, na.rm = T))
```

```
## # A tibble: 1 x 2
##   mean_CNC `sd(COVID.New.Cases, na.rm = T)`
##      <dbl>                            <dbl>
## 1    2548.                            2841.
```

```
FULL_New %>% na.omit %>% summarize(var(COVID.New.Cases, COVID.New.Deaths))
```

```
## # A tibble: 1 x 1
##   `var(COVID.New.Cases, COVID.New.Deaths)`
##                                      <dbl>
## 1                                   81616.
```

Given the amount of numeric varibles in the dataset, I only performed summary statistics on a few using the core dplyr functions. In order to find the difference in statistics on a set day, I filtered by date rank and found that the number of confirmed COVID cases on day 25 was more than 1500 percent of Sars cases on the same day in 2003. In addition, I selected the sums of COVID deaths and Sars deaths and arranged it in descending order of COVID deaths. Considering the COVID dataset was up-to-date as of March 13 (Day 53), I wanted to compare the values of both viruses. As of this time, the COVID death rate is around 3.73%, while the Sars virus from onset to conclusion resulted in around a 9.56% death rate. In other words, Sars was not as successful at spreading to as many hosts as COVID, but the people that contracted Sars had a higher chance of mortality than ones that have contracted COVID. The group_by function allowed me to group the categories based on week number and further summarize to find the mean and standard deviation of the variable regarding new cases of COVID per week. As shown, the highest average new cases for COVID occurred in the most recent week (week 8), while the highest for Sars came in week four. Considering some of the variable observations were accumulations, I took the median number for both regarding the sum of confirmed cases. COVID's median was 51,857 in 53 observations, while Sars median was 5,211 in 70 observations. This can be used as an indicator of the quantity of cases at half the time; days were used as the factor in this case. First and last sums of deaths were found to be four on day one and 774 on the last day for Sars virus. A correlation was done between new cases of Sars and new cases of COVID and was found to have a slighty negative correlation.

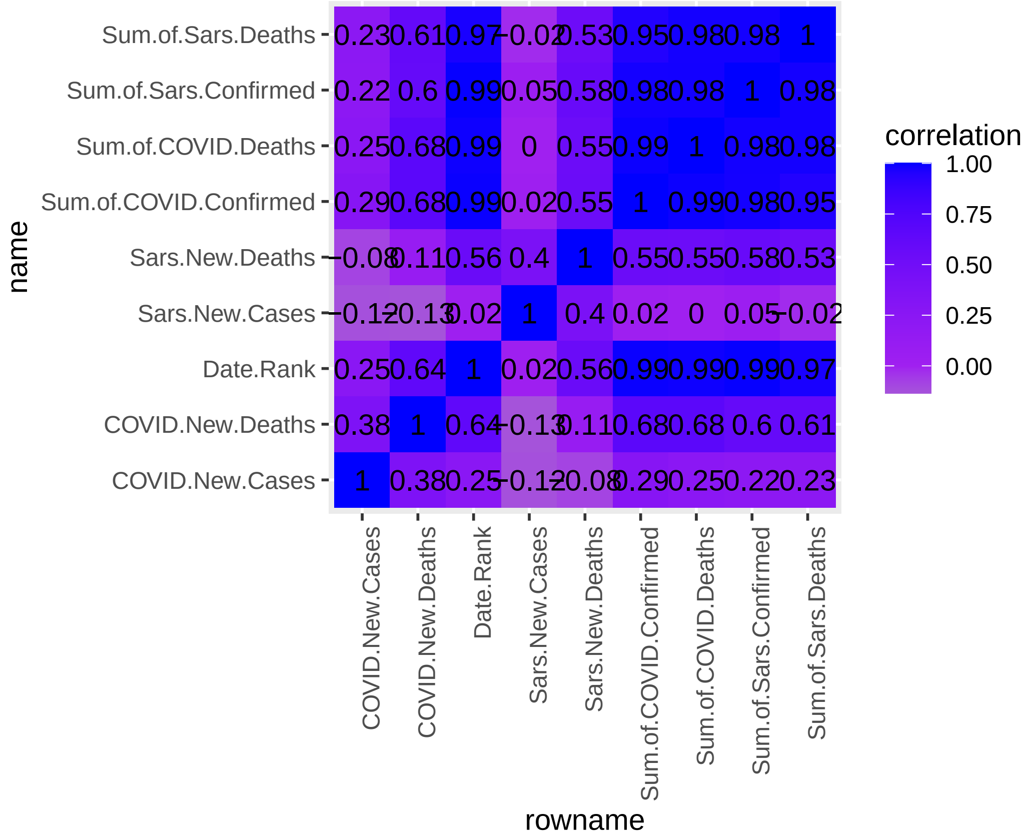
```
library(tidyverse)
library(kableExtra)
COOR <- FULL_New %>% na.omit %>% select_if(is.numeric)
kable(COOR) %>% kable_styling(fixed_thead = T)
```

Date.Rank	Sum.of.COVID.Confirmed	Sum.of.Sars.Confirmed	Sum.of.COVID.Deaths	Sum.of.Sars.Deaths	COVI
2	314	231	6	4	
3	581	278	17	9	
4	846	317	25	10	
5	1320	362	41	10	
6	2014	403	56	11	
7	2798	522	80	17	
8	4593	554	106	17	
9	6065	1392	132	49	
10	7818	1478	170	53	
11	9826	1555	213	53	
12	11953	1619	259	54	
13	14557	1690	305	58	
14	17391	1871	362	62	
15	20630	2289	426	78	
16	24544	2338	492	79	
17	28276	2419	565	84	
18	31481	2479	638	89	
19	34886	2660	724	98	
20	37558	2725	813	103	
21	40554	2777	910	106	
22	43103	2843	1018	111	
23	45171	2952	1115	116	
24	46997	3022	1369	119	
25	49053	3233	1383	144	
26	50580	3298	1526	154	
27	51857	3357	1669	159	
28	71429	3448	1775	165	
29	73332	3595	1873	182	
30	75204	4090	2009	182	
31	75748	4180	2129	217	
32	76769	4561	2247	229	
33	77794	4713	2359	251	
34	78811	4921	2463	263	
35	79331	5105	2618	274	
36	80239	5318	2700	293	
37	81109	5504	2762	321	
38	82294	5688	2804	353	
39	83652	5870	2858	372	
40	85403	6211	2924	391	
41	87137	6349	2977	417	
42	88948	6519	3043	435	
43	90869	6671	3112	460	
44	93090	6800	3198	477	
45	95324	6923	3280	494	
46	98192	7074	3380	505	
47	101927	7178	3486	513	
48	105592	7280	3584	525	
49	109577	7404	3809	551	
50	113702	7445	4012	572	
51	118319	7467	4292	586	
52	125260	7569	4613	597	
53	132758	7624	4955	610	

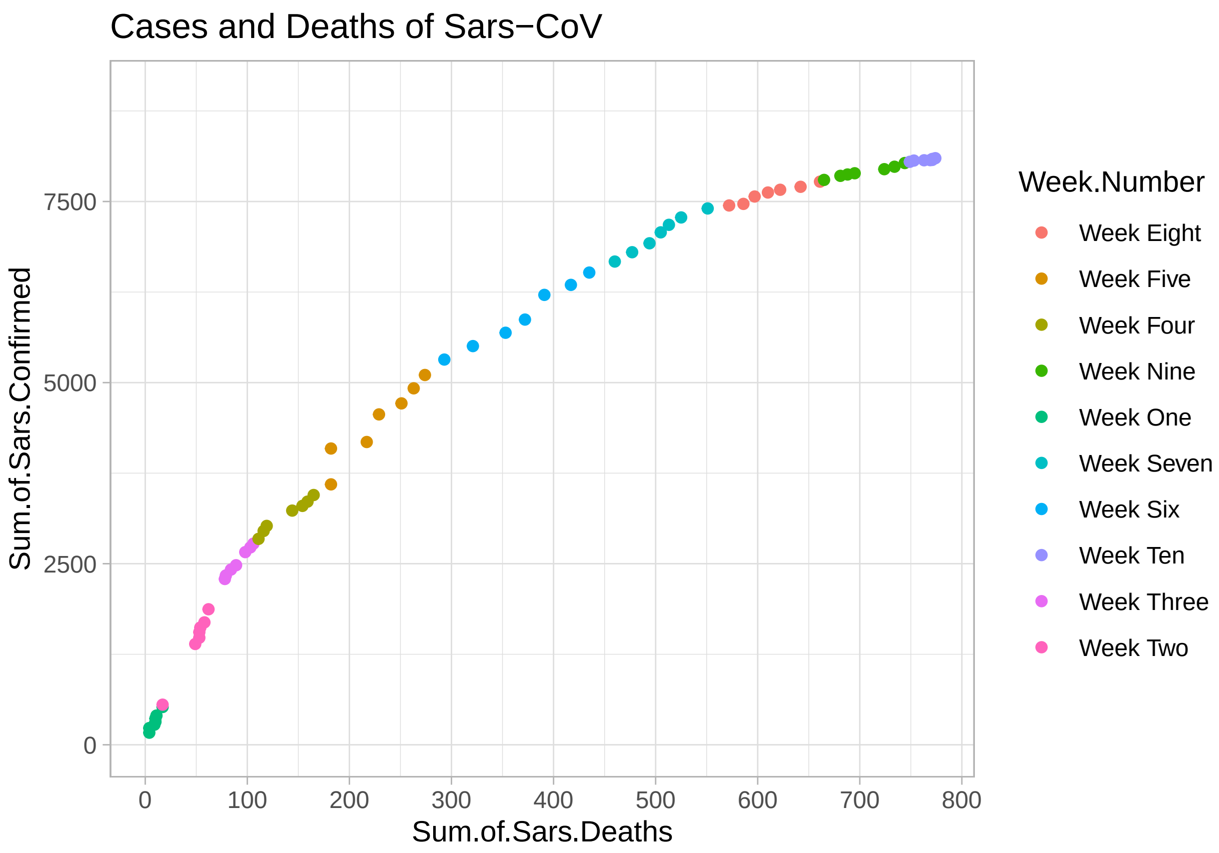
```r
Tidycor <- cor(COOR) %>% as.data.frame %>% rownames_to_column %>%
    pivot_longer(-1, names_to = "name", values_to = "correlation")
head(Tidycor)
```

```
## # A tibble: 6 x 3
##   rowname   name                    correlation
##   <chr>     <chr>                         <dbl>
## 1 Date.Rank Date.Rank                     1
## 2 Date.Rank Sum.of.COVID.Confirmed        0.991
## 3 Date.Rank Sum.of.Sars.Confirmed         0.995
## 4 Date.Rank Sum.of.COVID.Deaths           0.986
## 5 Date.Rank Sum.of.Sars.Deaths            0.968
## 6 Date.Rank COVID.New.Cases               0.254
```

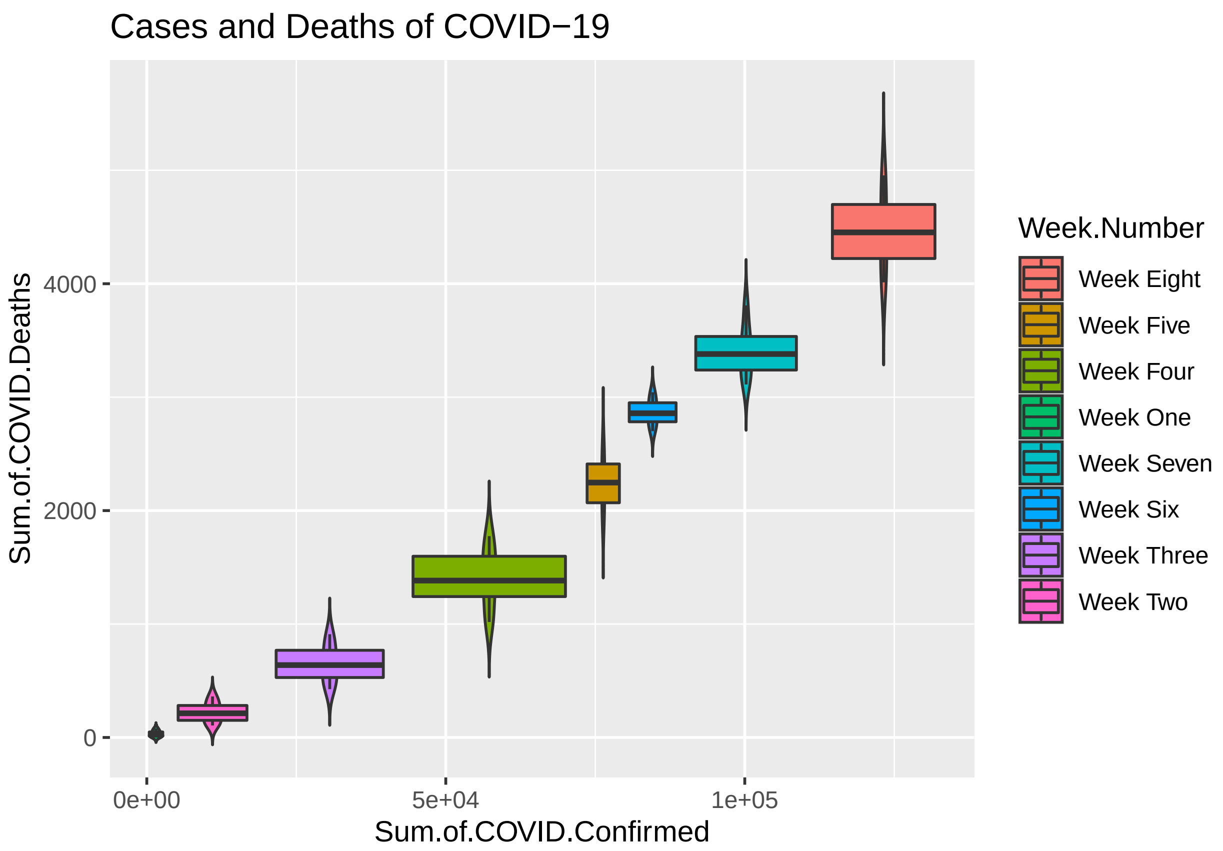
```r
Tidycor %>% ggplot(aes(rowname, name, fill = correlation)) +
    geom_tile() + scale_fill_gradient2(low = "green", mid = "purple",
    high = "blue") + geom_text(aes(label = round(correlation,
    2)), color = "black", size = 4) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1)) + coord_fixed()
```



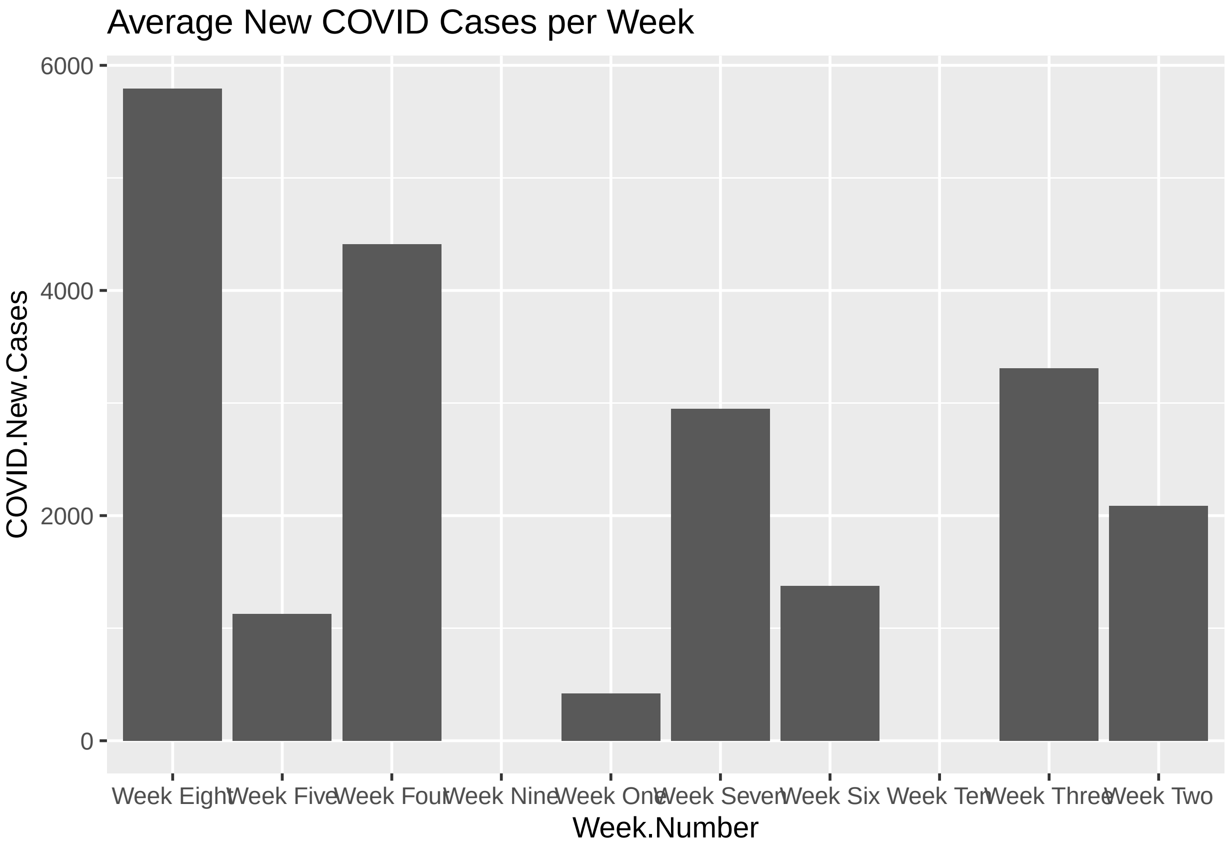
```r
ggplot(FULL_New, aes(Sum.of.Sars.Deaths, Sum.of.Sars.Confirmed,
    color = Week.Number)) + geom_point() + theme_light() + scale_x_continuous(breaks = seq(0,
    900, by = 100)) + scale_y_continuous(lim = c(0, 9000)) +
    ggtitle("Cases and Deaths of Sars-CoV")
```



```r
ggplot(FULL_New, aes(x = Sum.of.COVID.Confirmed, y = Sum.of.COVID.Deaths,
    fill = Week.Number)) + geom_violin(trim = F) + geom_boxplot(width = 0.1) +
    ggtitle("Cases and Deaths of COVID-19")
```



```
ggplot(FULL_New, aes(x = Week.Number)) + geom_bar(aes(y = COVID.New.Cases),
    stat = "summary", fun.y = "mean") + ggtitle("Average New COVID Cases per Week")
```



A coorelation heatmap calculates an integer to represent the relationship between variables. A number of "1" indicates sameness between the two variables and any coorelation close to that is considered to have a strong relationship. Some examples of this are seen between the sum of COVID confirmed cases and the date. However, some variables, such as the sum of COVID deaths and Sars new cases had a correlation of "0", meaning there is no relationship between the two. There were about equal distributions of strongly correlated variables as well as variables that had weak correlations. Only a few would be considered "moderate", as is seen between the sum of COVID confirmed and COVID new cases.

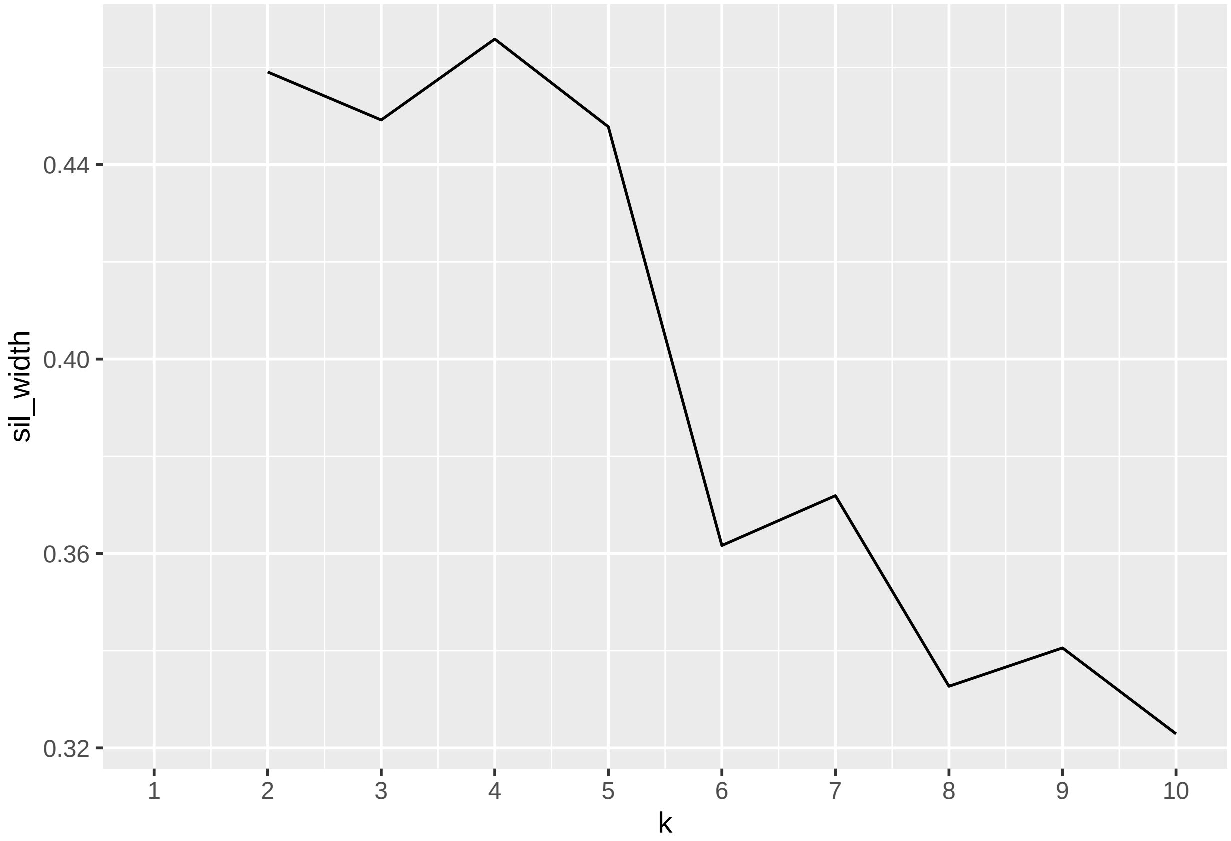
The cases and deaths of Sars virus were plotted with the color of dot indicating the week in which the deaths and cases occurred. The steepest slope which indicated the highest death rate per sum of cases happened around the onset of the virus and slowed toward its ending. There is a positive relationship between the two variables indicated by the positive slope of the line. As the sum of Sars deaths increased, the sum of confirmed cases increased.

Box and whisker plots show summary statistics between variables. The box itself indicates the IQR, representing the length between the lower and upper quartile, the middle line indicates the median value, and the whiskers show minimums and maximums. This third plot illustrated the sum of COVID confirmed cases and sum of COVID deaths. As suspected, there was a positive relationship between the two with a steady increase. As can be seen by the spacing in the third plot, the highest increase in deaths occurred between week seven and eight.

The final plot used stat=summary to find the mean of new COVID cases by week number. Week one had the lowest average new cases, with week five coming in at second lowest. The highest average new case count came from week eight, with close to 6,000 new cases.

```
library(cluster)
Clustered <- FULL_New %>% dplyr::select(-Week.Number)
```

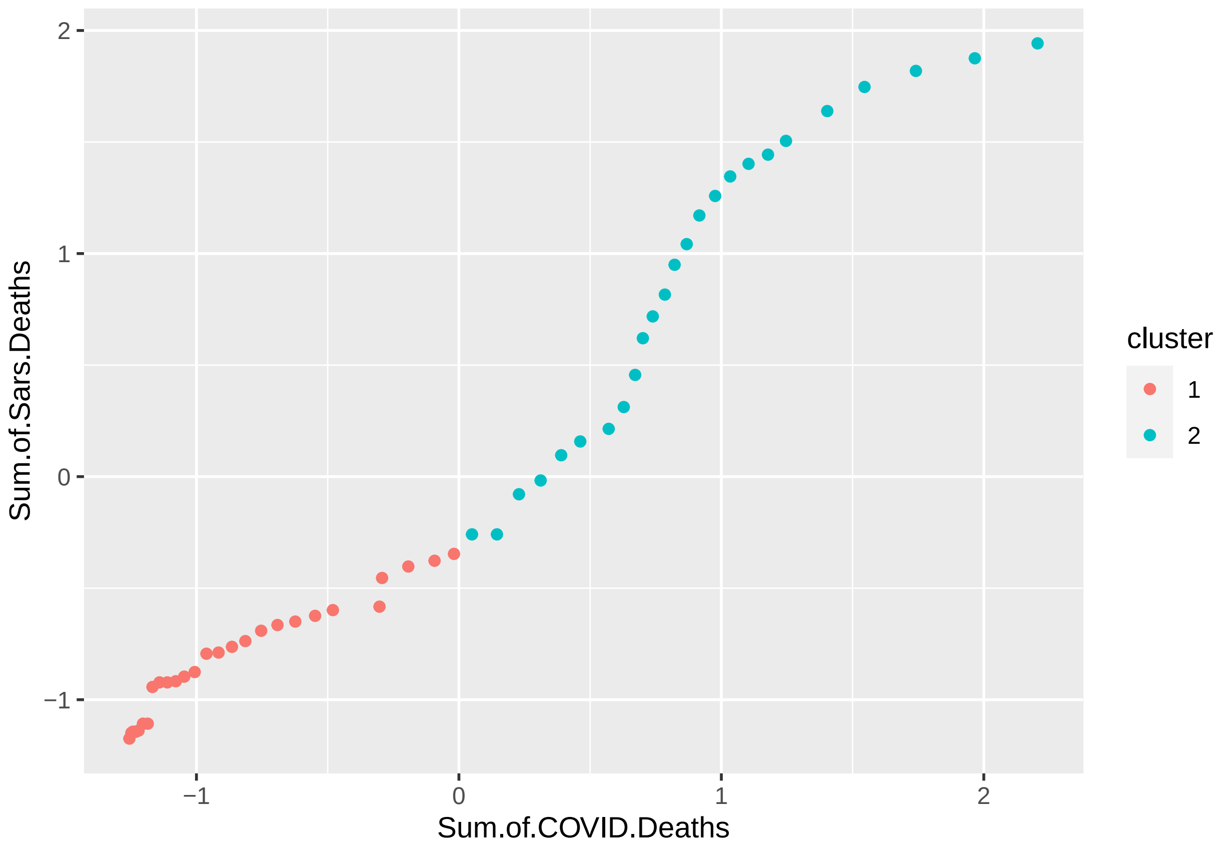
```r
Clustered1 <- Clustered %>% na.omit %>% scale %>% as.data.frame()
sil_width <- vector()
for (i in 2:10) {
    kms <- kmeans(Clustered1, centers = i)
    sil <- silhouette(kms$cluster, dist(Clustered1))
    sil_width[i] <- mean(sil[, 3])
}
ggplot() + geom_line(aes(x = 1:10, y = sil_width)) + scale_x_continuous(name = "k",
    breaks = 1:10)
```



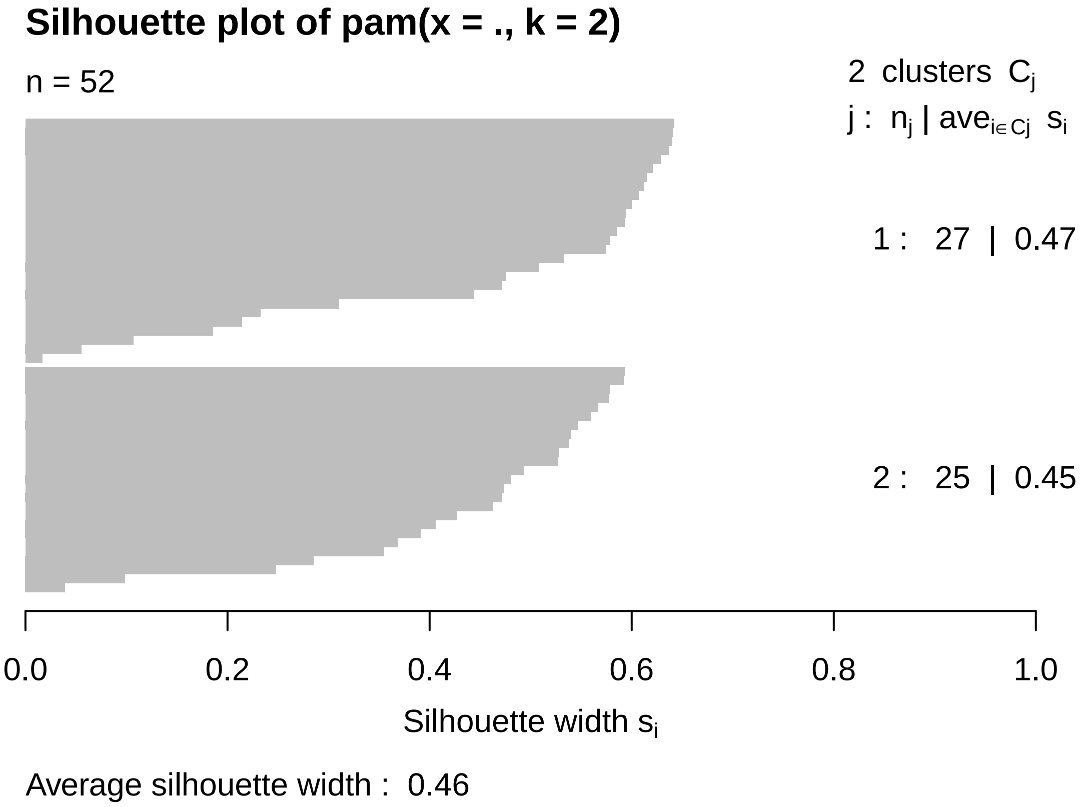
```r
pam1 <- Clustered1 %>% pam(k = 2)
final <- Clustered1 %>% mutate(cluster = as.factor(pam1$clustering))
confmat <- final %>% count(cluster) %>% arrange(desc(n)) %>%
    pivot_wider(names_from = "cluster", values_from = "n", values_fill = list(n = 0))
confmat
```

```
## # A tibble: 1 x 2
##     `1`   `2`
##   <int> <int>
## 1    27    25
```

```r
ggplot(final, aes(x = Sum.of.COVID.Deaths, y = Sum.of.Sars.Deaths,
    color = cluster)) + geom_point()
```



```
plot(pam1, which = 2)
```



According to the ggplot, the highest average silhouette width came from five clusters (.463), but the second highest was very close and suggested two (.459). Given it is better to choose fewer clusters to achieve parsimony, I decided to have two. The quantity of clusters recommended was based on all numeric variables in the dataset, but only the sums of COVID deaths and Sars deaths were visualized. As seen on the ggplot, there were no real "clusters" or clear separation, but more evenly spaced out points throughout the graph. Given the silhouette width used to derive the clusters was not very high, the cluster solution achieved did not elicit much information.