DSC630 - Week 1

Chris Goodwin

June 6, 2020

Introduction

For this R refresher assignment, I will be using a dataset of COVID related cases and deaths for each county in the US. This is a csv file, so we will use read.table()

```
file_path <- "C:/Users/goodw/Downloads/us-counties.csv"
data <- read.csv(file_path)
head(data)
```

```
##         date    county      state  fips cases deaths
## 1 2020-01-21 Snohomish Washington 53061     1      0
## 2 2020-01-22 Snohomish Washington 53061     1      0
## 3 2020-01-23 Snohomish Washington 53061     1      0
## 4 2020-01-24      Cook   Illinois 17031     1      0
## 5 2020-01-24 Snohomish Washington 53061     1      0
## 6 2020-01-25    Orange California  6059     1      0
```

Summary statistics

Right off the bat we can see that two columns will be of importance to us - cases and deaths. We will look at some simple statistics about these two columns using the describe() function from the Hmisc library.

```
library("Hmisc")
```

```
## Warning: package 'Hmisc' was built under R version 3.6.2
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.6.2
```

```
## Loading required package: Formula
```

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

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
describe(data$cases)
```

```
## data$cases
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##   209599        0     6286    0.998      360    659.4        1        1
##      .25      .50      .75      .90      .95
```

```
##        4       19       90      394     1001
##
## lowest :      0      1      2      3      4, highest: 208550 209195 209688 210227 210728
```

```
describe(data$deaths)
```

```
## data$deaths
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##   209599        0     1380     0.84    20.23    38.52        0        0
##      .25      .50      .75      .90      .95
##        0        0        3       17       41
##
## lowest :     0     1     2     3     4, highest: 21090 21132 21170 21234 21262
```

We can see right off the bat that there are a wide range of values for these columns. There is at least one county that had over 210,000 cases reported at a given time. And we can also see that there was a county with over 21,000 deaths reported.

To start off, I want to just look at the cases in my home county (Erie, New York). I will create a subset where county = "Erie" and state = "New York".

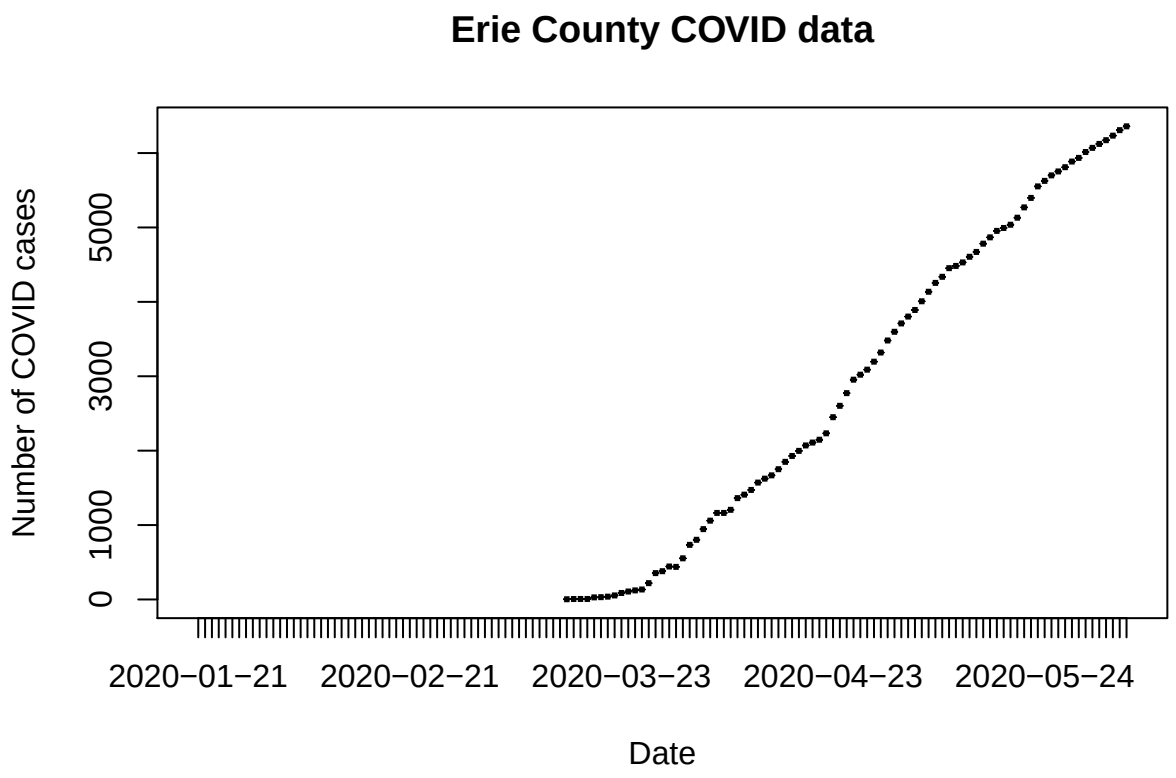
```
erie <- data[ which(data$county == "Erie" & data$state == "New York"), ]
head(erie)
```

```
##            date county    state  fips cases deaths
## 2569 2020-03-15   Erie New York 36029     3      0
## 3028 2020-03-16   Erie New York 36029     6      0
## 3544 2020-03-17   Erie New York 36029     7      0
## 4141 2020-03-18   Erie New York 36029     7      0
## 4870 2020-03-19   Erie New York 36029    28      0
## 5717 2020-03-20   Erie New York 36029    31      0
```

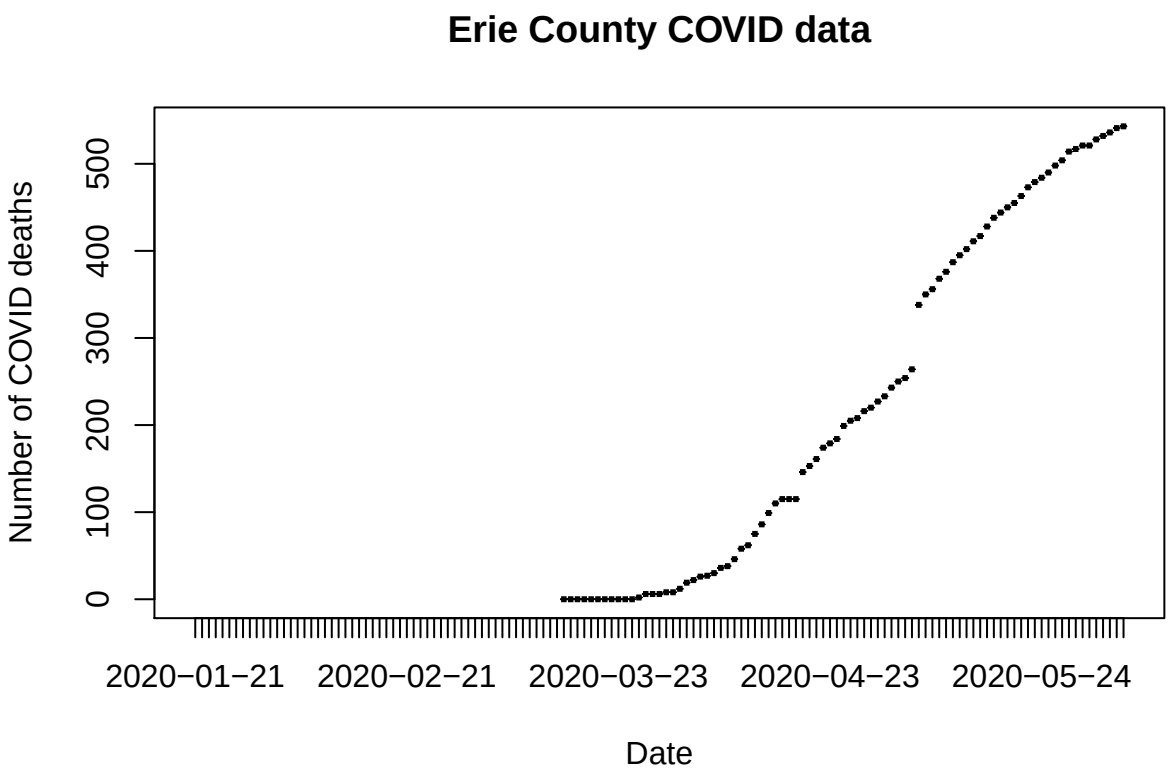
Plots

Now that we have our subset, we can just do some simple scatter plots. I will plot cases over time, and deaths over time.

```
plot(erie$date, erie$cases, main = 'Erie County COVID data', xlab = 'Date', ylab = 'Number of COVID cas
```




```
plot(erie$date, erie$deaths, main = 'Erie County COVID data', xlab = 'Date', ylab = 'Number of COVID dea
```



Now while this is helpful, what would probably be a better scatter plot is to look at new cases and new deaths. I created a for look which does some subtraction to determine the number of new cases and deaths every day.

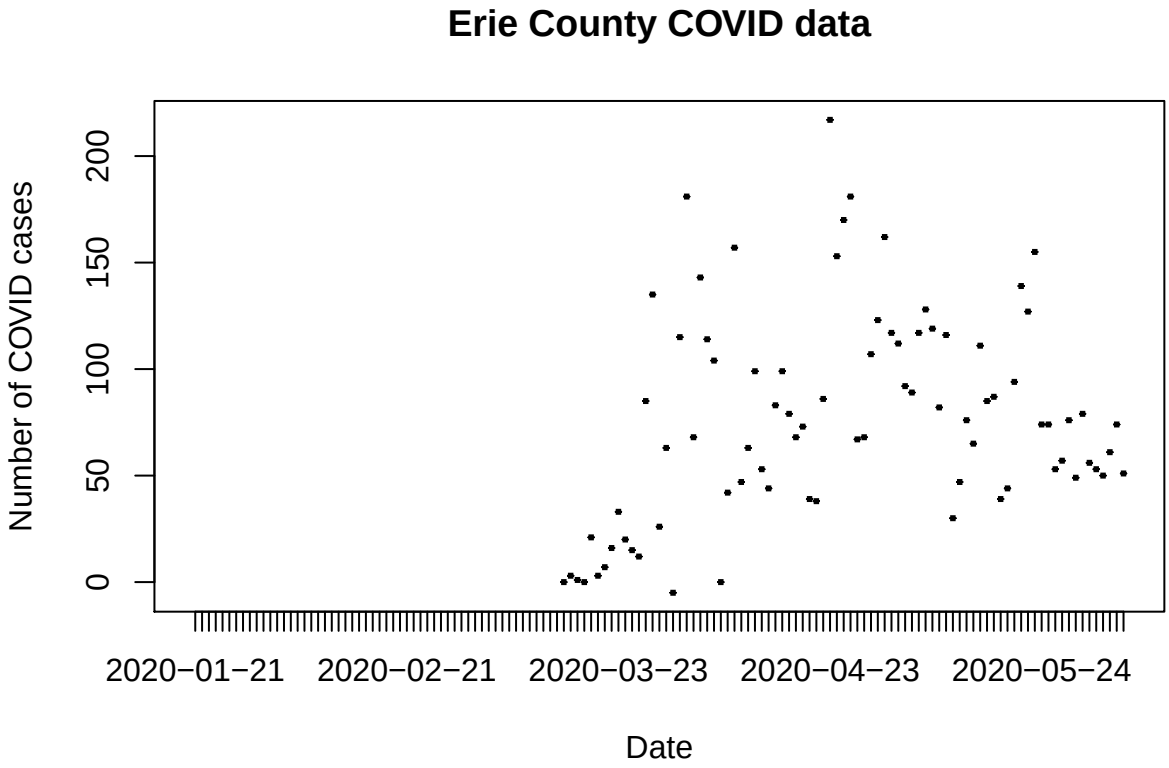
```
erie$new_cases <- 0
erie$new_deaths <- 0

for (i in 1:nrow(erie)) {
  erie[i+1,7] <- erie[i+1,5] - erie[i,5]
  erie[i+1,8] <- erie[i+1,6] - erie[i,6]
}
erie <- na.omit(erie)
tail(erie,10)
```

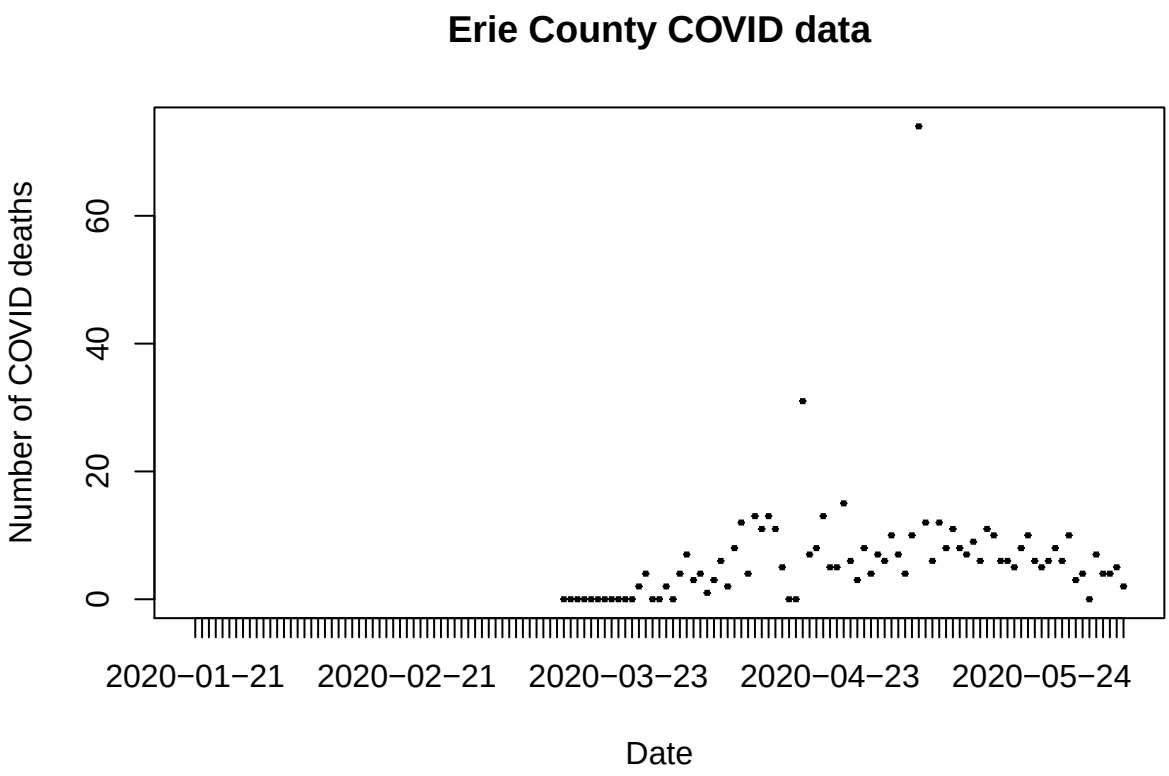
```
##              date county    state  fips cases deaths new_cases new_deaths
## 181433 2020-05-27   Erie New York 36029  5810    504        57          6
## 184411 2020-05-28   Erie New York 36029  5886    514        76         10
## 187391 2020-05-29   Erie New York 36029  5935    517        49          3
## 190374 2020-05-30   Erie New York 36029  6014    521        79          4
## 193362 2020-05-31   Erie New York 36029  6070    521        56          0
## 196354 2020-06-01   Erie New York 36029  6123    528        53          7
## 199347 2020-06-02   Erie New York 36029  6173    532        50          4
## 202344 2020-06-03   Erie New York 36029  6234    536        61          4
## 205346 2020-06-04   Erie New York 36029  6308    541        74          5
## 208348 2020-06-05   Erie New York 36029  6359    543        51          2
```

Now that we have this data, I will produce scatter plots of new cases and new deaths.

```
plot(erie$date, erie$new_cases, main = 'Erie County COVID data', xlab = 'Date', ylab = 'Number of COVID
```



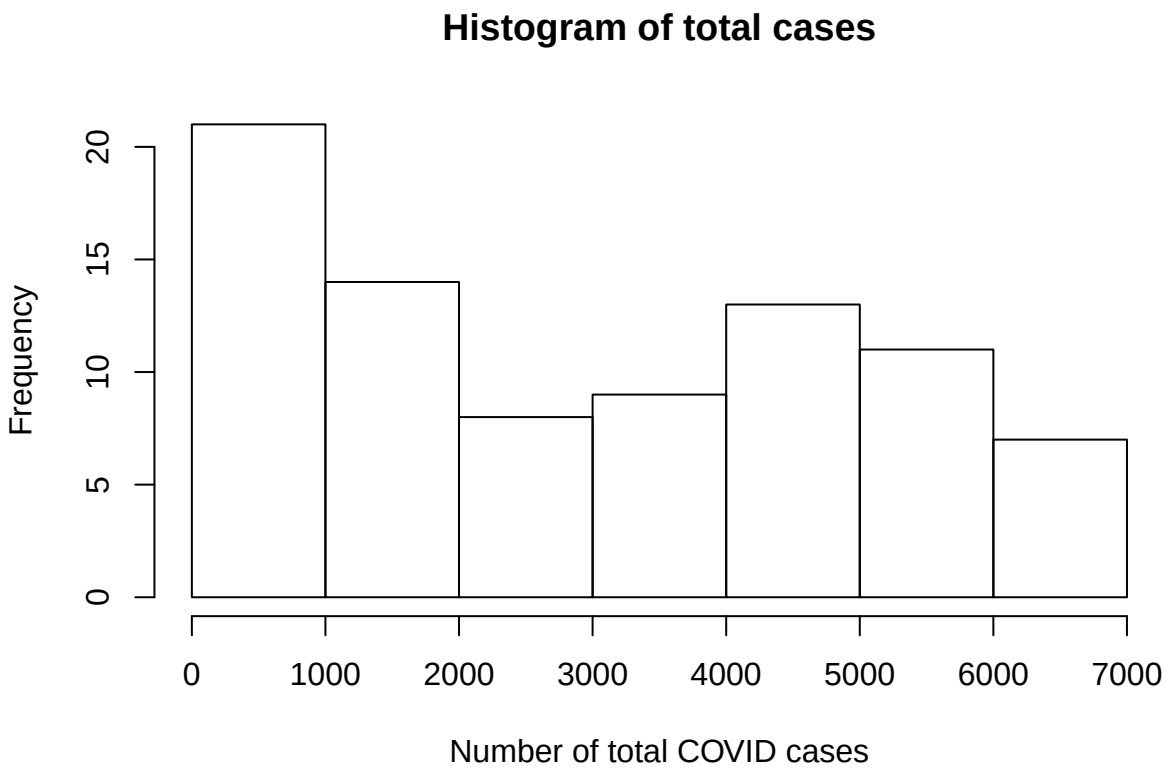

```
plot(erie$date, erie$new_deaths, main = 'Erie County COVID data', xlab = 'Date', ylab = 'Number of COVI
```



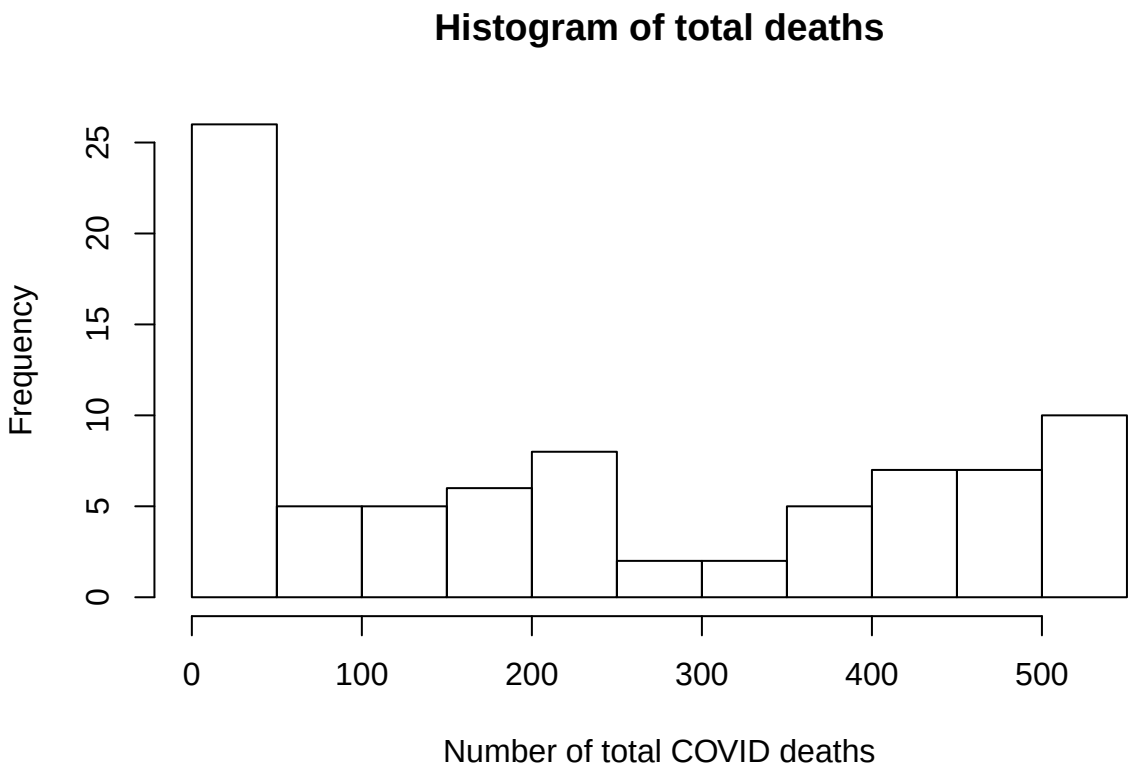
Now this shows that there was a rise and then fall in both, but it is certainly not the clearest way to look at this data.

Let's take a look at some histograms of the cases and deaths.

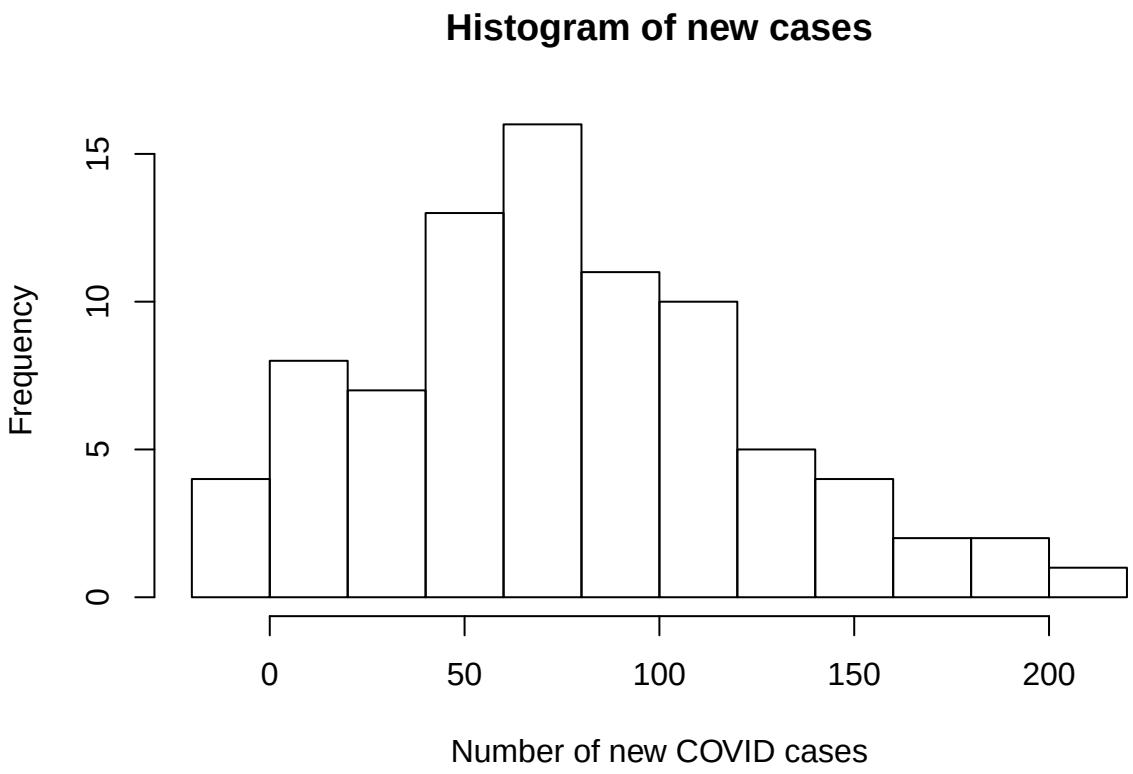
```
hist(erie$cases, xlab = "Number of total COVID cases", main = "Histogram of total cases")
```



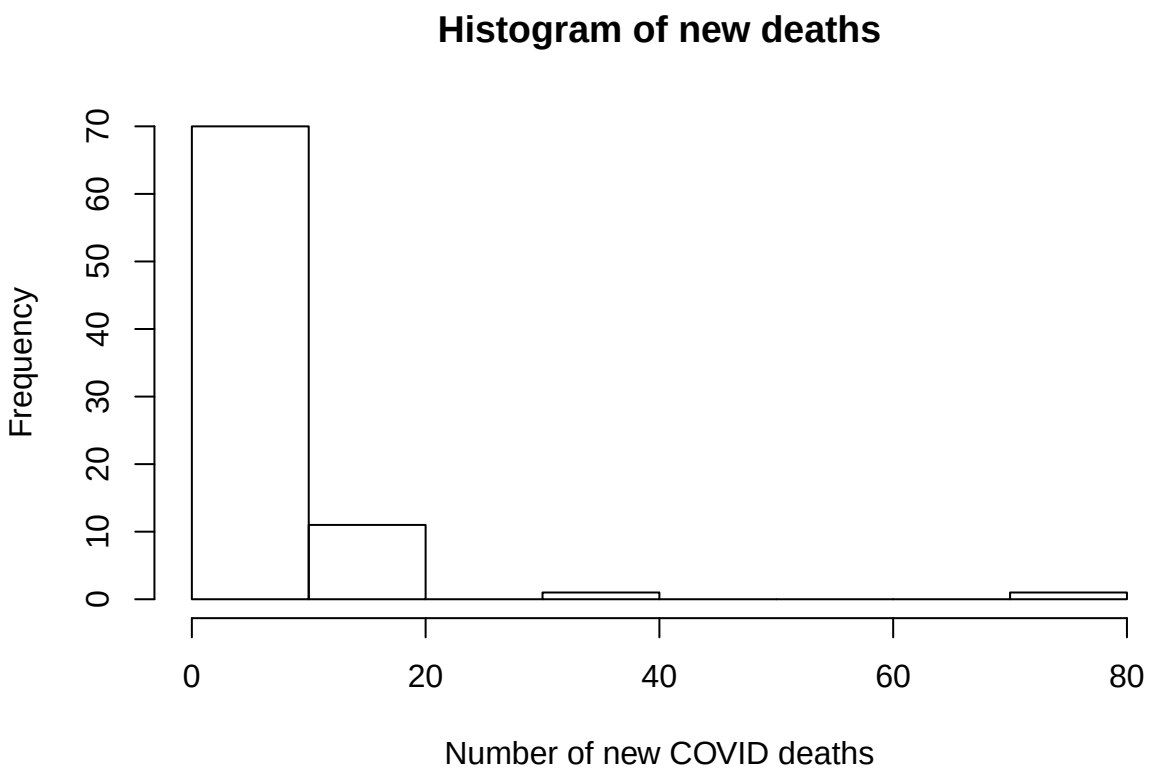

```
hist(erie$deaths, xlab = "Number of total COVID deaths", main = "Histogram of total deaths")
```




```r
hist(erie$new_cases, xlab = "Number of new COVID cases", main = "Histogram of new cases")
```




```r
hist(erie$new_deaths, xlab = "Number of new COVID deaths", main = "Histogram of new deaths")
```

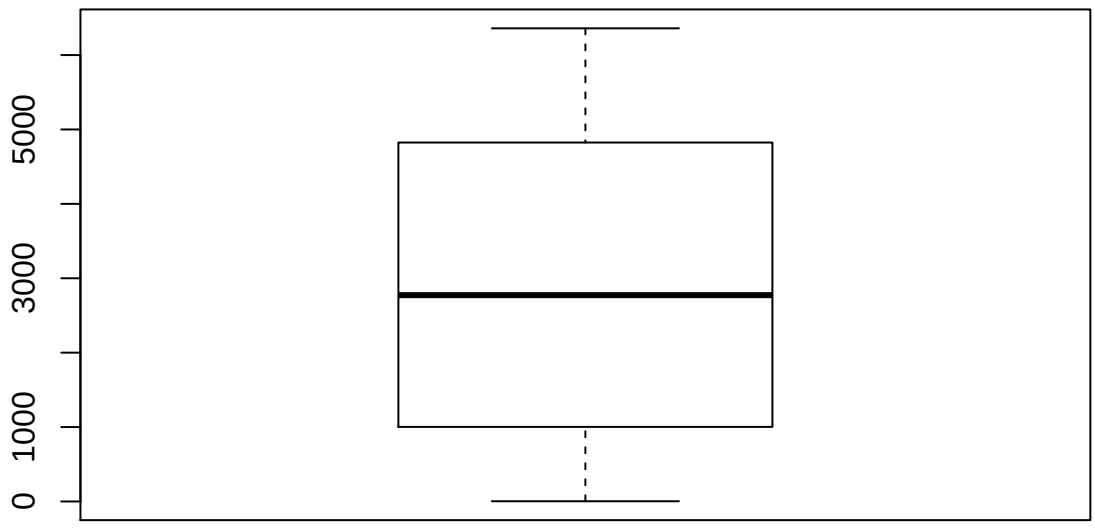


The histograms of total cases and total deaths give us some insight into how long it took to get each 1000 cases. The frequency indicates the number of days. So for total cases, there were 20+ days between the first case and the 1000th case. Then there were only ~15 days between the 1000th case and the 2000th case. And then the smallest interval was that it took less than 10 days to get from 2000 to 3000 cases.

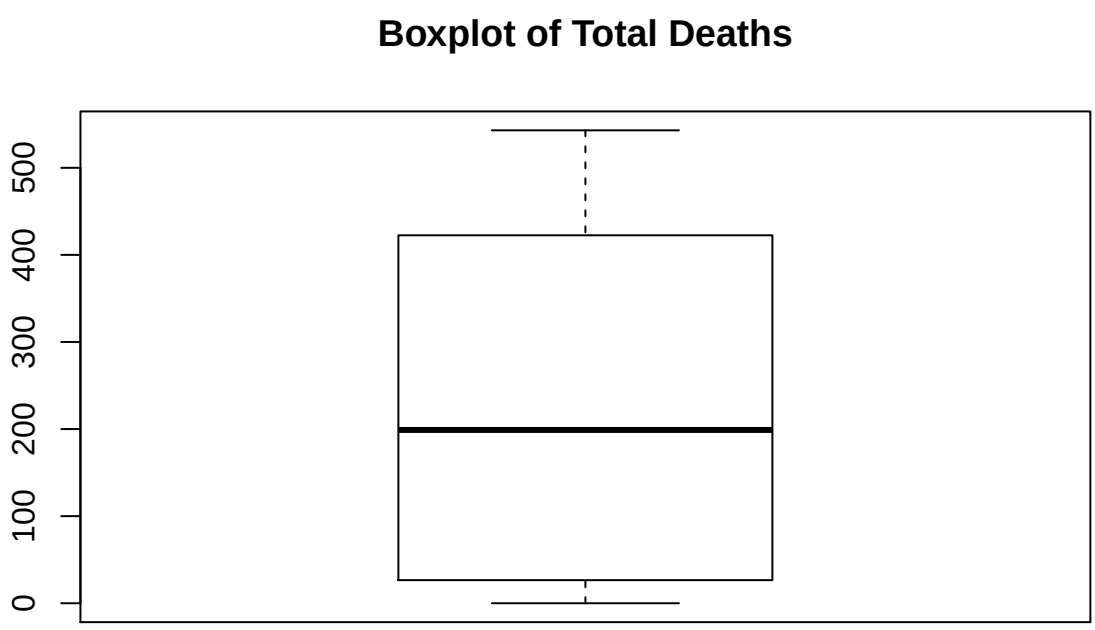
The histograms of new cases and new deaths give us an interesting breakdown of how each individual day was during the course of the pandemic. The vast majority of days had between 50 and 100 new cases, with less than 10 new deaths.

Next we can look at some boxplots of the same data.

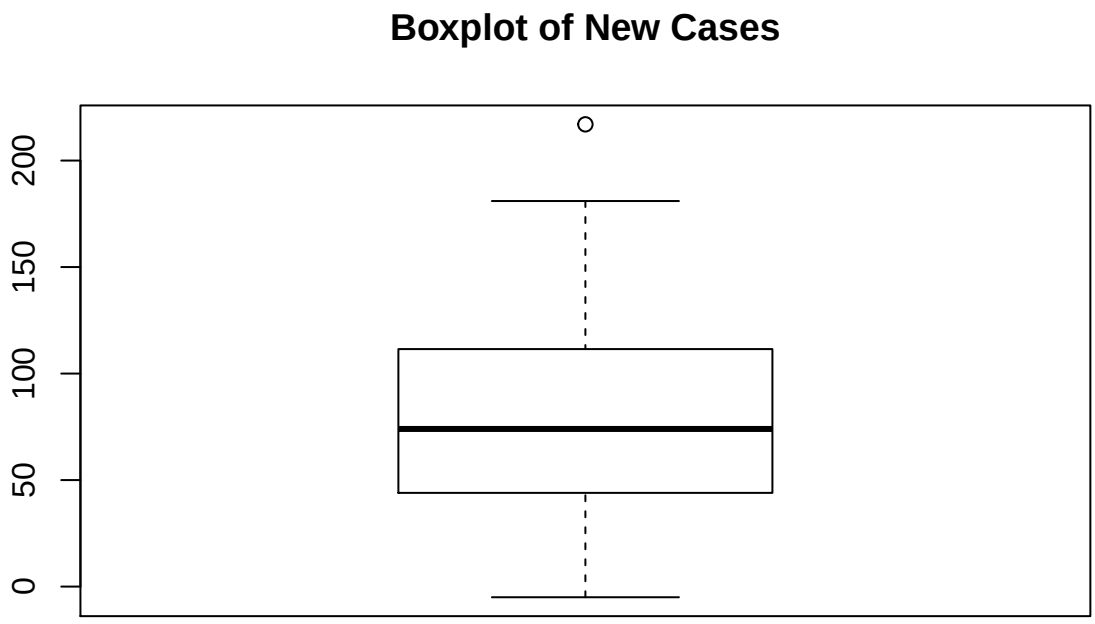
```
boxplot(erie$cases, main = "Boxplot of Total Cases")
```



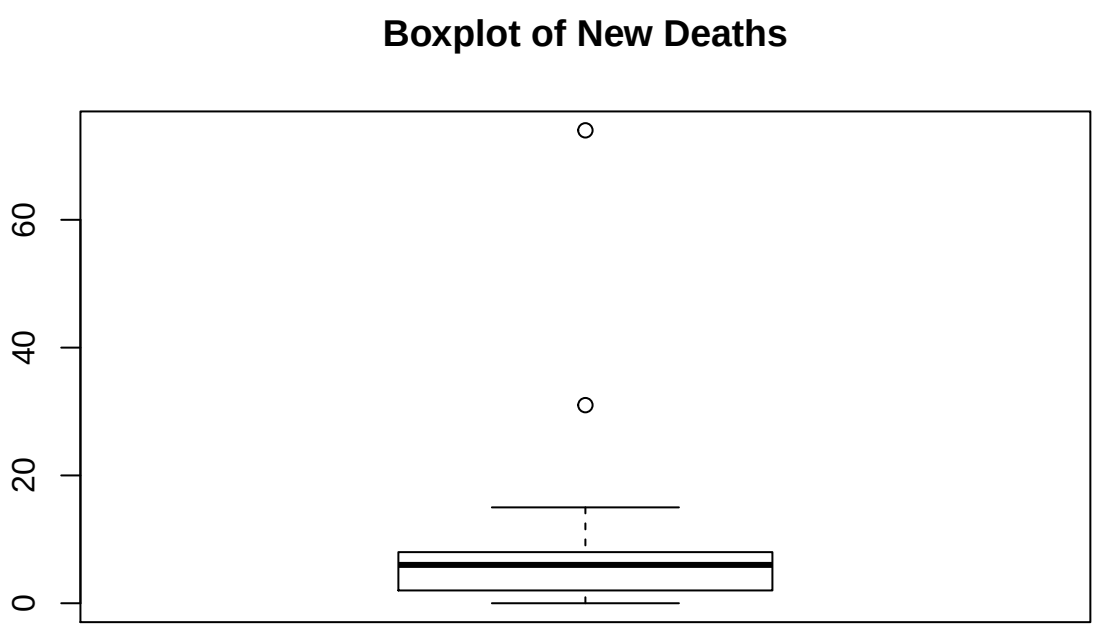

```r
boxplot(erie$deaths, main = "Boxplot of Total Deaths")
```




```r
boxplot(erie$new_cases, main = "Boxplot of New Cases")
```

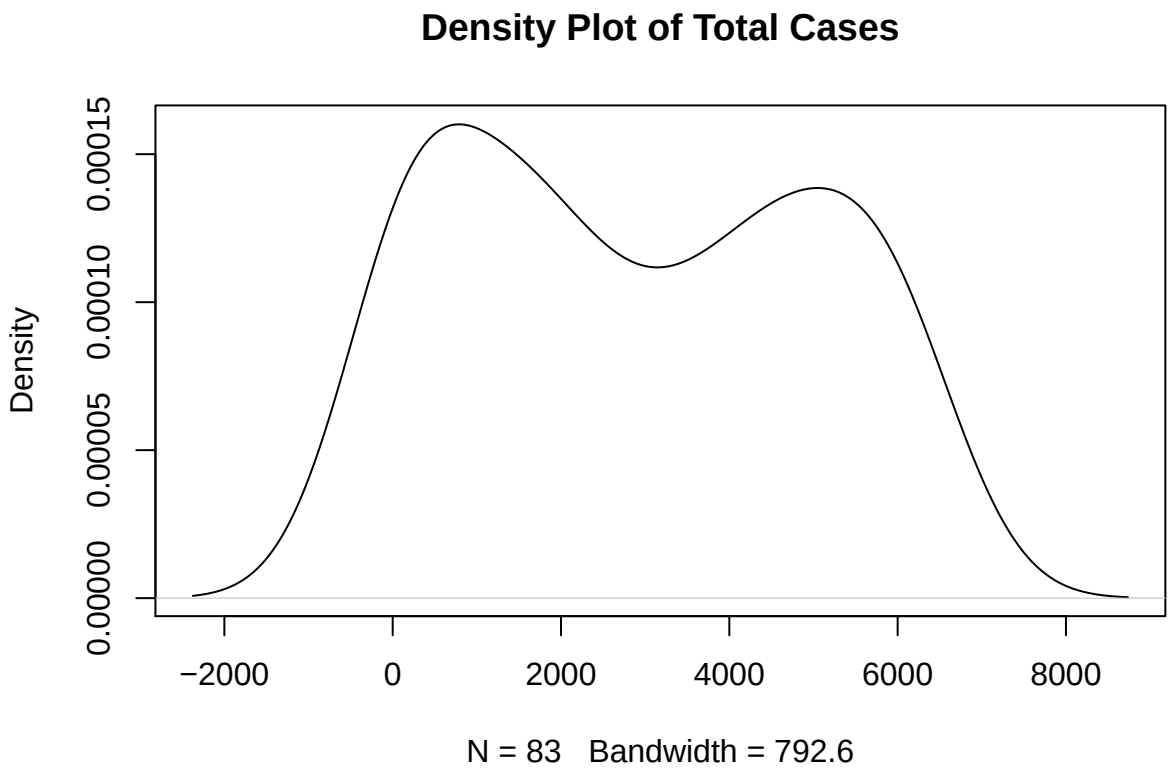



```
boxplot(erie$new_deaths, main = "Boxplot of New Deaths")
```

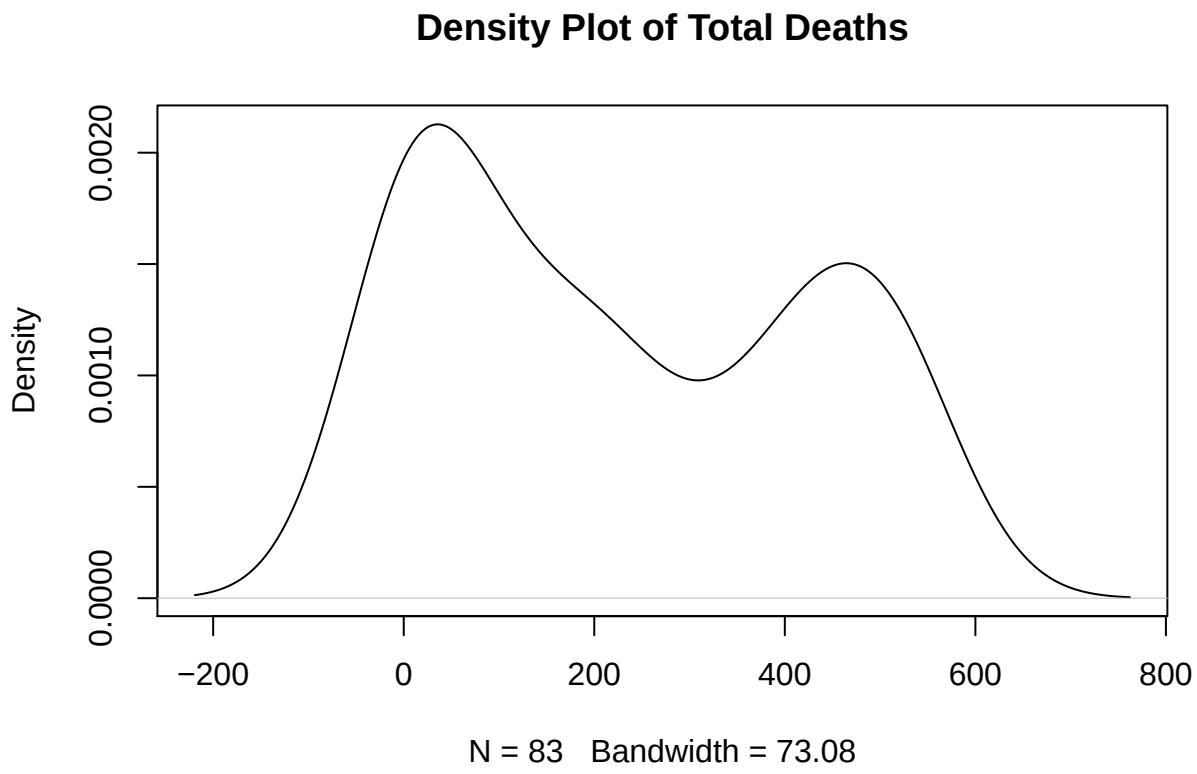


These plots give us a good idea of how the data is distributed. Finally, we will look at some density plots:

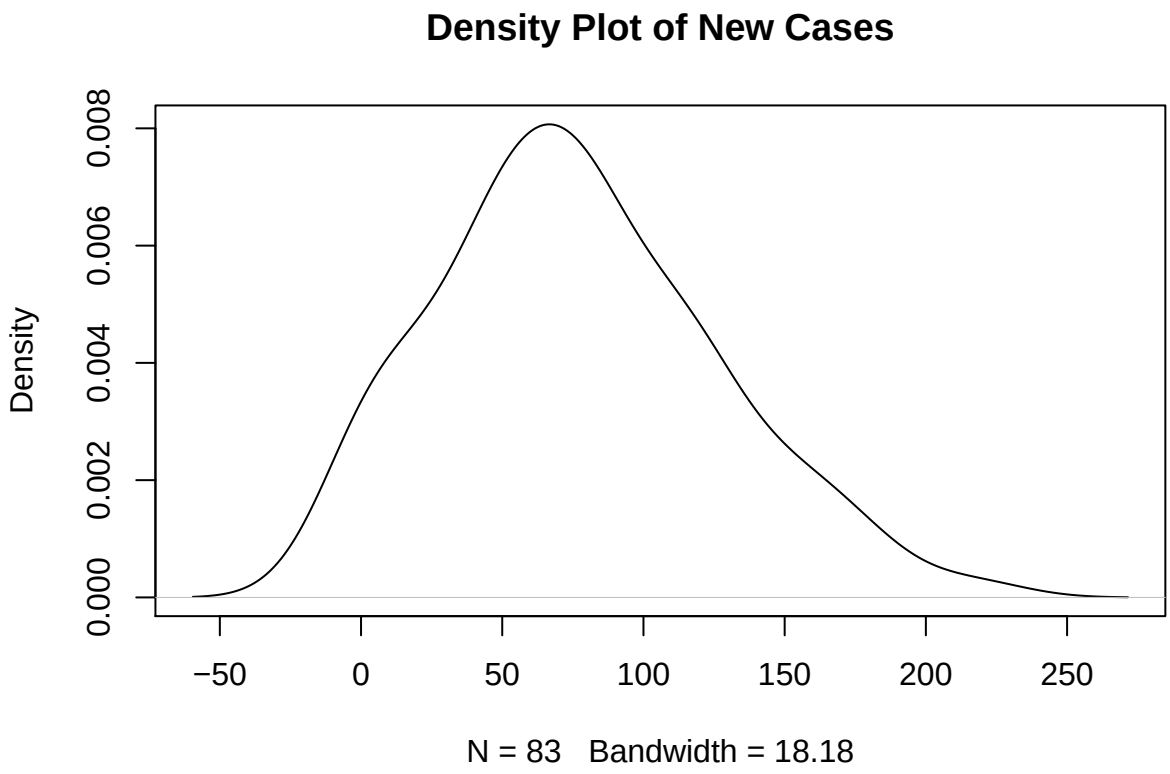
```
plot(density(erie$cases), main = 'Density Plot of Total Cases')
```



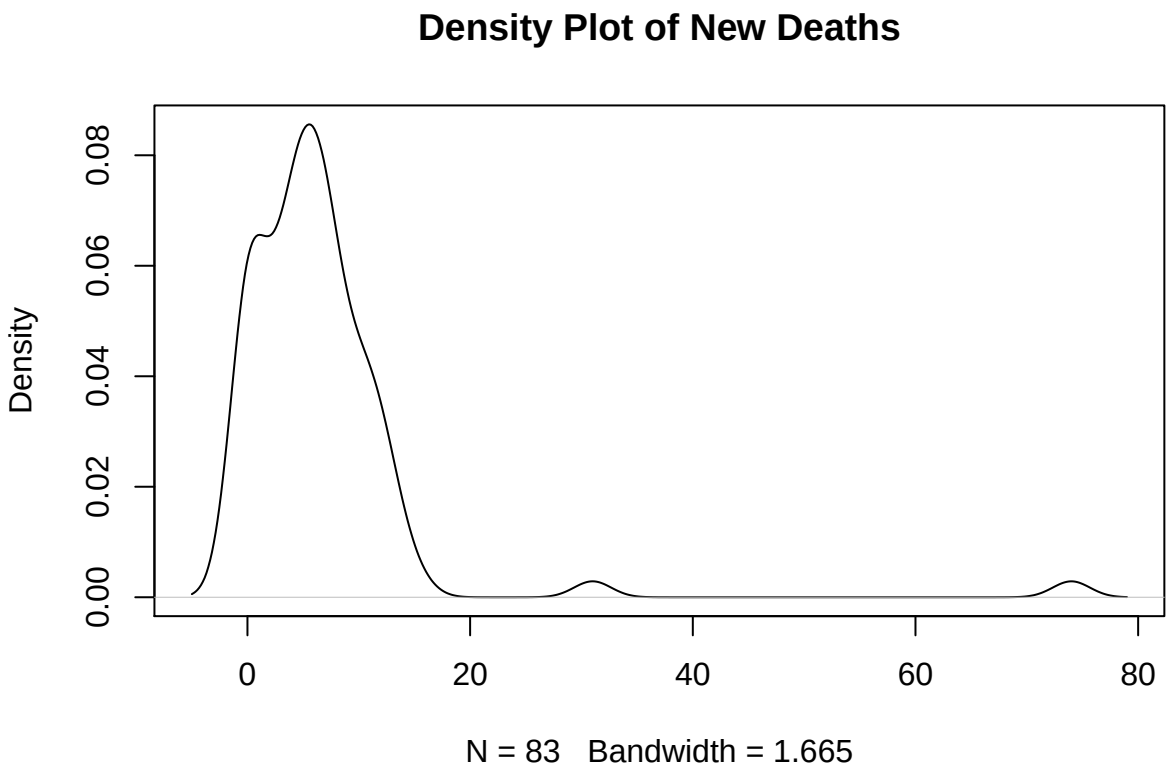

```r
plot(density(erie$deaths), main = 'Density Plot of Total Deaths')
```




```
plot(density(erie$new_cases), main = 'Density Plot of New Cases')
```



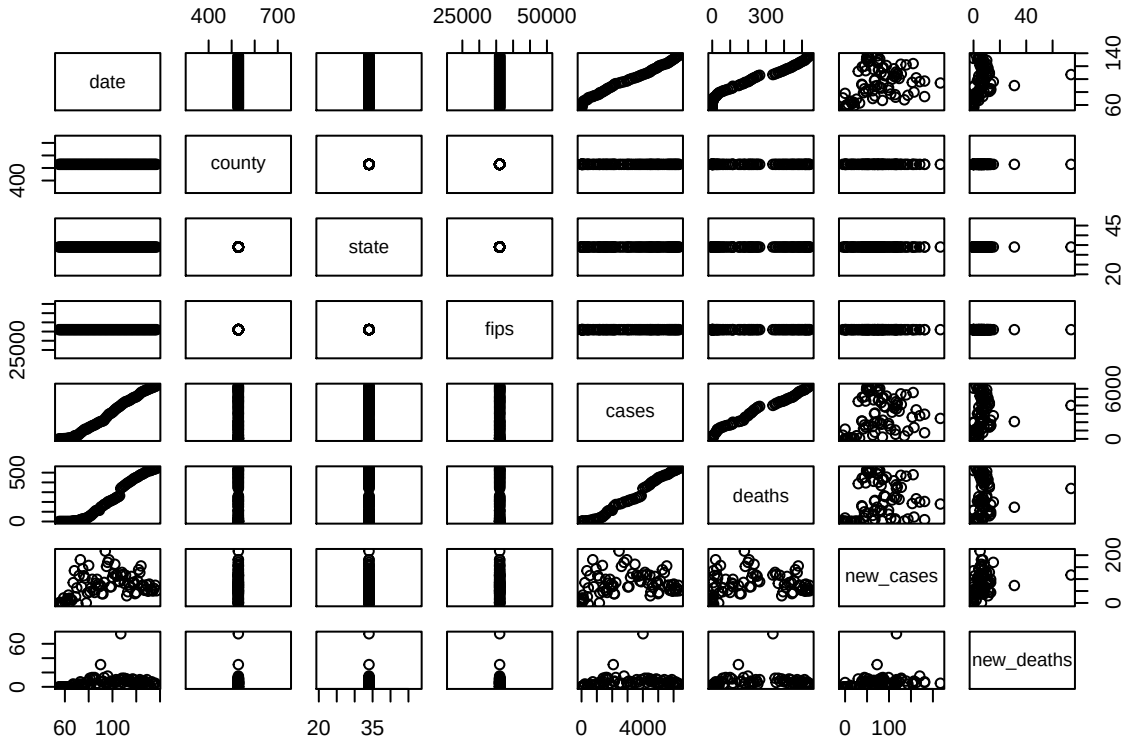

```r
plot(density(erie$new_deaths), main = 'Density Plot of New Deaths')
```



Bivariate relationships

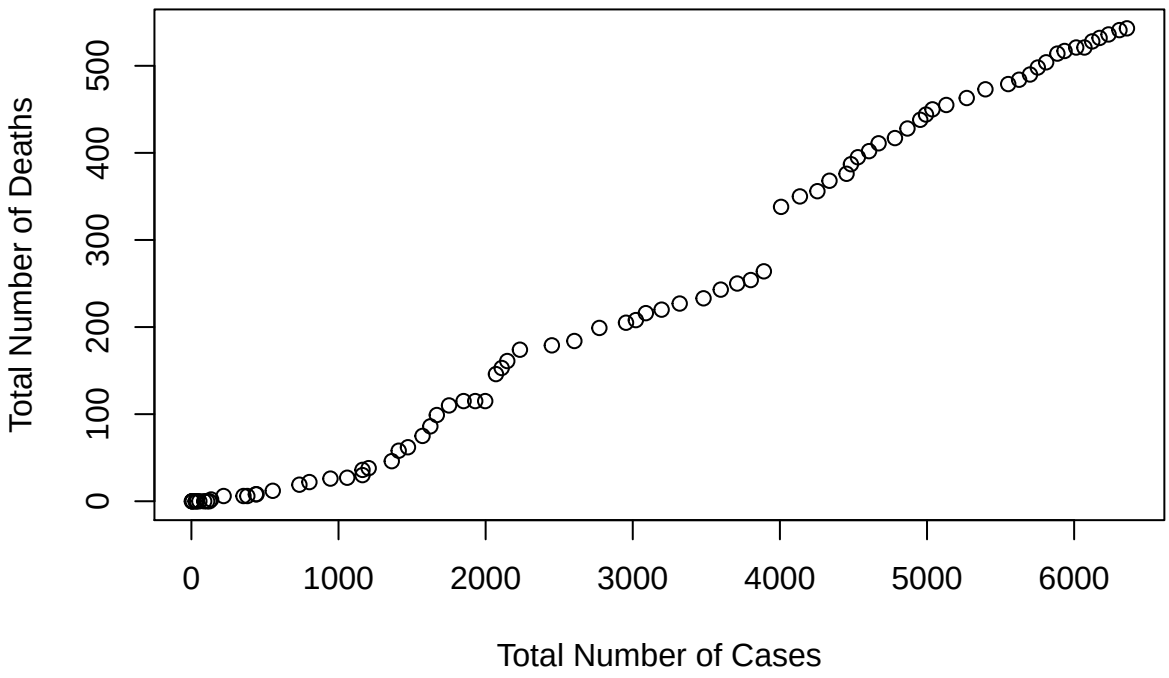
To evaluate some bivariate relationships, we can run the pairs() function, which will produce scatter plots for every pair of variables in our dataset.

```
pairs(erie)
```



We already plotted date vs deaths and cases in previous scatter plots. From this output, we can see there appears to be a strong relationship between total cases and total deaths. While this is not surpising at all, I think we should take a closer look at this scatter plot.

```
plot(erie$cases, erie$deaths, xlab = 'Total Number of Cases', ylab = 'Total Number of Deaths')
```



As I stated above, the graph should not surprise us. As the number of cases of the disease increase, so too do the number of deaths. Another way to review these relationships would be to look at correlation values.

```
cor(erie$cases, erie$deaths)
```

```
## [1] 0.991596
```

A perfect correlation is 1, so a correlation of .99 is incredibly high. These two values are extremely correlated.

If we want to evaluate correlations between all the numeric columns, we can create a new subset of data that only contains the numeric columns

```
numeric <- erie[,5:8]
head(numeric)
```

```
##      cases deaths new_cases new_deaths
## 2569     3      0         0          0
## 3028     6      0         3          0
## 3544     7      0         1          0
## 4141     7      0         0          0
## 4870    28      0        21          0
## 5717    31      0         3          0
```

We can then run cor again on the entire new dataset, which will produce correlation values between all pairs of numeric values.

```
cor(numeric)
```

```
##                cases    deaths new_cases new_deaths
## cases      1.0000000 0.9915960 0.2497462  0.2181193
```

```
## deaths     0.9915960 1.0000000 0.1885446  0.2044791
## new_cases  0.2497462 0.1885446 1.0000000  0.2506017
## new_deaths 0.2181193 0.2044791 0.2506017  1.0000000
```

Not surprisingly, there is not much relationship between any of the variables. The only strong relationship is between total cases and total deaths.

Summary

We now to to summarize our data. This can be accomplished using the aptly named summary() function. First, we will look at a summary of the Erie county data

```
summary(erie)
```

```
##          date          county          state         fips           cases
##  2020-03-15: 1   Erie     :83   New York  :83   Min.   :36029   Min.   :   3
##  2020-03-16: 1   Abbeville: 0   Alabama   : 0   1st Qu.:36029   1st Qu.:1002
##  2020-03-17: 1   Acadia   : 0   Alaska    : 0   Median :36029   Median :2773
##  2020-03-18: 1   Accomack : 0   Arizona   : 0   Mean   :36029   Mean   :2899
##  2020-03-19: 1   Ada      : 0   Arkansas  : 0   3rd Qu.:36029   3rd Qu.:4824
##  2020-03-20: 1   Adair    : 0   California: 0   Max.   :36029   Max.   :6359
##  (Other)   :77   (Other)  : 0   (Other)   : 0
##      deaths        new_cases        new_deaths
##  Min.   :  0.0   Min.   : -5.00   Min.   : 0.000
##  1st Qu.: 26.5   1st Qu.: 44.00   1st Qu.: 2.000
##  Median :199.0   Median : 74.00   Median : 6.000
##  Mean   :226.5   Mean   : 76.58   Mean   : 6.542
##  3rd Qu.:422.5   3rd Qu.:111.50   3rd Qu.: 8.000
##  Max.   :543.0   Max.   :217.00   Max.   :74.000
##
```

Next, we will look at the summary of the overall COVID data.

```
summary(data)
```

```
##          date               county                  state
##  2020-06-05:  3006   Washington:  2302   Texas         : 14714
##  2020-06-04:  3002   Unknown   :  2071   Georgia       : 11769
##  2020-06-03:  2999   Jefferson :  1909   Virginia      :  9130
##  2020-06-02:  2996   Franklin  :  1782   Kentucky      :  7757
##  2020-06-01:  2992   Jackson   :  1618   North Carolina:  7108
##  2020-05-31:  2989   Lincoln   :  1595   Missouri      :  6977
##  (Other)   :191615   (Other)   :198322   (Other)       :152144
##       fips           cases            deaths
##  Min.   : 1001   Min.   :     0   Min.   :    0.00
##  1st Qu.:18077   1st Qu.:     4   1st Qu.:    0.00
##  Median :29053   Median :    19   Median :    0.00
##  Mean   :30062   Mean   :   360   Mean   :   20.23
##  3rd Qu.:45039   3rd Qu.:    90   3rd Qu.:    3.00
##  Max.   :56045   Max.   :210728   Max.   :21262.00
##  NA's   :2246
```

Another thing we will want to look at is the structure of our data, using the str() function. Once again we will first look at the Erie county data.

```
str(erie)
```

```
## 'data.frame':    83 obs. of  8 variables:
```

```
##  $ date      : Factor w/ 137 levels "2020-01-21","2020-01-22",..: 55 56 57 58 59 60 61 62 63 64 ...
##  $ county    : Factor w/ 1772 levels "Abbeville","Acadia",..: 529 529 529 529 529 529 529 529 529 529
##  $ state     : Factor w/ 55 levels "Alabama","Alaska",..: 34 34 34 34 34 34 34 34 34 34 ...
##  $ fips      : int  36029 36029 36029 36029 36029 36029 36029 36029 36029 36029 ...
##  $ cases     : int  3 6 7 7 28 31 38 54 87 107 ...
##  $ deaths    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ new_cases : num  0 3 1 0 21 3 7 16 33 20 ...
##  $ new_deaths: num  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int 84
##   ..- attr(*, "names")= chr "84"
```

And once again we will also look at the overall data as well:

```
str(data)
```

```
## 'data.frame':    209599 obs. of  6 variables:
##  $ date  : Factor w/ 137 levels "2020-01-21","2020-01-22",..: 1 2 3 4 4 5 5 5 6 6 ...
##  $ county: Factor w/ 1772 levels "Abbeville","Acadia",..: 1467 1467 1467 379 1467 1177 379 1467 978 9
##  $ state : Factor w/ 55 levels "Alabama","Alaska",..: 52 52 52 15 52 5 15 52 3 5 ...
##  $ fips  : int  53061 53061 53061 17031 53061 6059 17031 53061 4013 6037 ...
##  $ cases : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ deaths: int  0 0 0 0 0 0 0 0 0 0 ...
```

A few observations:

1) It is interesting to me that the state variable has 55 levels, seeing as how there are only 50 states. This makes me want to take a look at what the extra levels are:

```
levels(data$state)
```

```
##  [1] "Alabama"                   "Alaska"
##  [3] "Arizona"                   "Arkansas"
##  [5] "California"                "Colorado"
##  [7] "Connecticut"               "Delaware"
##  [9] "District of Columbia"      "Florida"
## [11] "Georgia"                   "Guam"
## [13] "Hawaii"                    "Idaho"
## [15] "Illinois"                  "Indiana"
## [17] "Iowa"                      "Kansas"
## [19] "Kentucky"                  "Louisiana"
## [21] "Maine"                     "Maryland"
## [23] "Massachusetts"             "Michigan"
## [25] "Minnesota"                 "Mississippi"
## [27] "Missouri"                  "Montana"
## [29] "Nebraska"                  "Nevada"
## [31] "New Hampshire"             "New Jersey"
## [33] "New Mexico"                "New York"
## [35] "North Carolina"            "North Dakota"
## [37] "Northern Mariana Islands" "Ohio"
## [39] "Oklahoma"                  "Oregon"
## [41] "Pennsylvania"              "Puerto Rico"
## [43] "Rhode Island"              "South Carolina"
## [45] "South Dakota"              "Tennessee"
## [47] "Texas"                     "Utah"
## [49] "Vermont"                   "Virgin Islands"
## [51] "Virginia"                  "Washington"
## [53] "West Virginia"             "Wisconsin"
```

```
## [55] "Wyoming"
```

Ahh, this data also includes US territories Guam, Northern Mariana Islands, Puerto Rico, and the Virgin Islands. We then also have the District of Columbia included.

2) An observation that I found very interesting had less to do with the data itself but rather the structure. When we created a subset of our dataframe (erie), it maintained all of the levels for each factor. Despite only including one county and state, you can see from the str() output that every level was carried over. That will be interesting to note moving forward with this course.

3) The median and mean of new cases are interesting to me as well. Over the course of roughly 3 months, Erie county has averaged almost 77 new cases every day.

4) It is also interesting to me to look at the str() output for the date field. The number for each date represents the number of times this date appears in the table. For some reason, these values have been increasing over the last week. Is this just because more counties are now reporting this information? Do more counties have cases every day? Would definitely be something to investigate further.

Write to csv

As a final step, I will write my DataFrames to a csv file so that someone can execute these steps themselves.

```
write.csv(erie,"C:\\Users\\goodw\\Desktop\\erie.csv")
write.csv(data,"C:\\Users\\goodw\\Desktop\\covid.csv")
```