DATA 605 - Discussion #12

Zach Alexander

4/15/2020

One-Factor Linear Regression Analysis of COVID-19 Data by U.S. County

---

For the discussion questions over the next two weeks, I thought I'd build a regression model with county-level COVID-19 data and U.S. Census estimates. Because confirmed case counts are highly dependent on access to testing, access to hospitals (most confirmed case counts come directly from hospital administrators), and other factors, this model will likely reflect rough estimates of predicted counts, but I thought it was timely and useful to practice with this type of information.

For this first week, I'll build a simple one-factor regression model using population estimates and the number of confirmed COVID-19 cases by U.S. County as of April 14th, 2020 (yesterday).

Step 1: Downloading the Data

I decided to utilize the New York Times dataset that was made open-source at the end of March. For more information about the methodology of these confirmed case counts, you can find a detailed explanation here.

Additionally, I'll be using U.S. Census data estimates as factors for my regression model. This week, I decided to utilize population estimates as my factor for my regression. I found a large dataset on the Census website with population estimates as recent as 2019.

I saved both of these files to my github account and read them into R:

```
county_covid <- read.csv('https://raw.githubusercontent.com/zachalexander/data605_cuny/master/Homework1

county_population <- read.csv('https://raw.githubusercontent.com/zachalexander/data605_cuny/master/Home
```

Step 2: Tidying the Data

With the data successfully loaded into R, I then had to do a bit of tidying to ensure that my county-level data was accurate and reflected the most up-to-date COVID-19 confirmed case counts.

First, I decided to tidy my COVID-19 data. Since the confirmed counts are cumulative and broken out by each day since late January, I needed to filter the dataset to only include the confirmed counts for April 14th, 2020.

```
covid_sum <- county_covid %>%
  group_by(fips, county, state) %>%
  filter(date == '2020-04-14')
```

Next, I noticed that the New York Times groups the confirmed counts for New York City into one row, instead of breaking it into the five counties that comprise of "New York City" (Queens, Kings, New York, Bronx and Richmond). Therefore, I decided to use the FIPS code associated with New York County (36061) as my identifier when I eventually merge the population data into the confirmed case data. I then adjusted the population estimate to reflect the population of all five counties instead of just New York County (seen later).

```
covid_sum <- within(covid_sum, {
    f <- county == 'New York City'
    fips[f] <- '36061'
})
```

For the merge, I also thought it would be helpful to make the county names consistent across both datasets.

```
covid_sum$county <- paste(covid_sum$county, 'County')
```

With the COVID-19 data file ready for the merge, I then turned my attention to my population data file. In order to use county FIPS codes as my identifier in both datasets, I had to generate the FIPS codes in the population file.

```
county_population <- county_population %>%
  select(STATE, COUNTY, STNAME, CTYNAME, POPESTIMATE2019)

county_population$fips <- NA
for (i in 1:length(county_population$STATE)) {
  if(county_population$COUNTY[i] < 10) {
    county_population$fips[i] <- paste0(county_population$STATE[i], '00', county_population$COUNTY[i])
  }
  if(county_population$COUNTY[i] < 100 & county_population$COUNTY[i] >= 10) {
    county_population$fips[i] <- paste0(county_population$STATE[i], '0', county_population$COUNTY[i])
  }
  if(county_population$COUNTY[i] >=100) {
    county_population$fips[i] <- paste0(county_population$STATE[i], county_population$COUNTY[i])
  }
}

county_population <- county_population %>%
  select(fips, CTYNAME, STNAME, POPESTIMATE2019)
names(county_population) <- c('fips', 'county', 'state', 'pop_estimate')
```

I also noticed that this file had overall state population estimates, so before merging, I made sure to filter these out.

```
county_population <- county_population %>%
  filter(grepl("County",county))
```

With both data files ready to go, I then merged the population estimates data into the COVID-19 data file and created one final dataframe. I also updated the population count for New York City to ensure that it didn't just account for the population in New York County, but also the four other counties in the metropolitan area.

```r
fnl <- merge(covid_sum, county_population, by='fips')

fnl <- fnl %>%
  select(fips, county.x, state.x, cases, pop_estimate)

names(fnl) <- c('fips', 'county', 'state', 'case_count', 'pop_estimate')

fnl$pop_estimate[fnl$county == 'New York City County'] <- 8398748
fnl$county[fnl$county == 'New York City County'] <- 'New York City'

fnl <- fnl %>%
  arrange(desc(case_count))
```

Here is a look at my final dataframe, ready to start my regression analysis:

```r
kable(head(fnl, n=15L)) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

fips	county	state	case_count	pop_estimate
36061	New York City	New York	110465	8398748
36059	Nassau County	New York	25250	1356924
36103	Suffolk County	New York	22462	1476601
36119	Westchester County	New York	20191	967506
17031	Cook County	Illinois	16323	5150233
26163	Wayne County	Michigan	12209	1749343
34003	Bergen County	New Jersey	10426	932202
6037	Los Angeles County	California	10047	10039107
36087	Rockland County	New York	8335	325789
34017	Hudson County	New Jersey	8242	672391
34013	Essex County	New Jersey	8212	798975
12086	Miami-Dade County	Florida	7711	2716940
34039	Union County	New Jersey	7265	556341
42101	Philadelphia County	Pennsylvania	7130	1584064
34031	Passaic County	New Jersey	6438	501826

Initial Analysis

First, it's always good to take an initial look at the dataset. We can identify the dimensions:

```r
dim(fnl)
```

```
## [1] 2563    5
```

Our dataframe consists of 2563 counties, which is about 85% of all counties in the United States (3,009 total counties in the US).

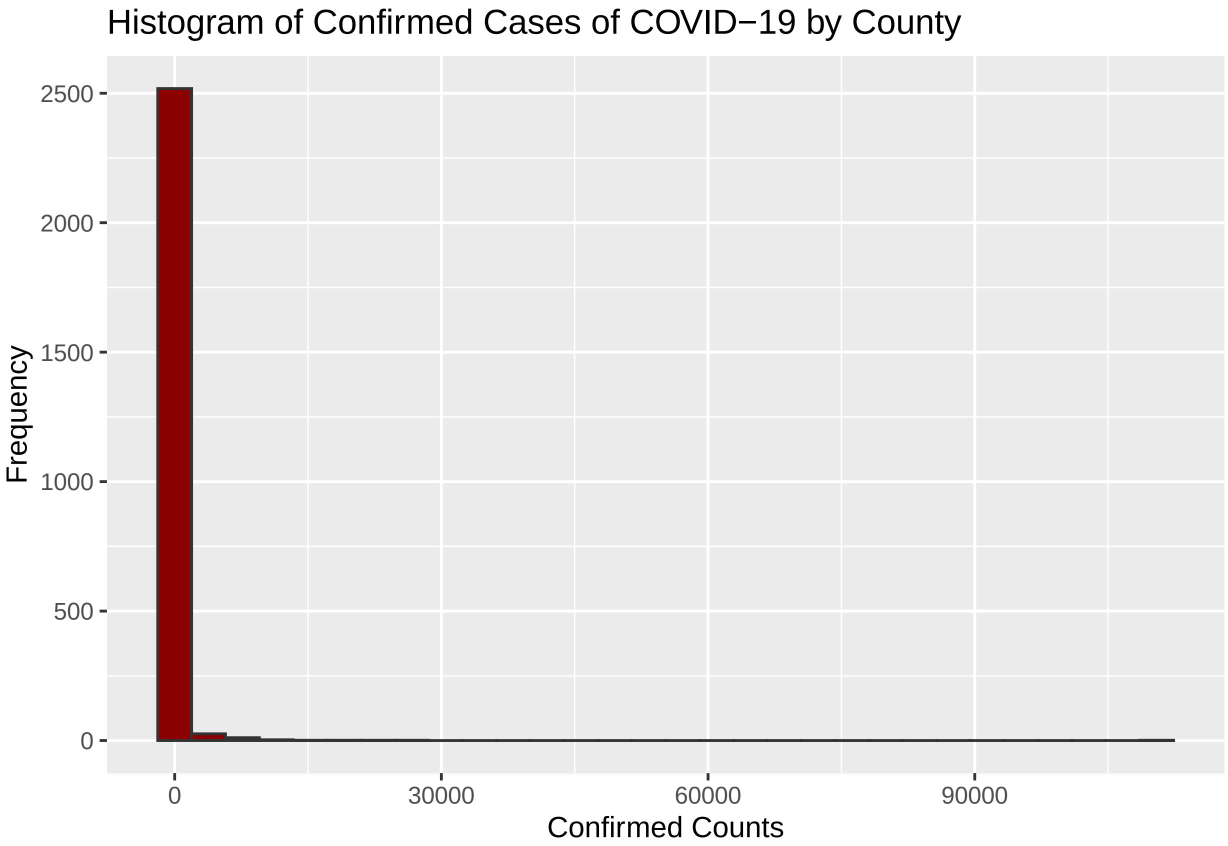
Next, we can take some summary statistics of the number of confirmed cases for COVID-19 by county:

```
summary(fnl$case_count)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      1.0      4.0     12.0    223.7     47.0 110465.0
```

We can see that, on average, there are about 224 confirmed cases by county, but the mean appears to be skewed since the median value is only 12 confirmed cases and 75% of counties have confirmed case counts by that fall below 50. As we've been hearing on the news, hotspots seem to develop, which is supported by these summary statistics. Additionally, New York City has been hosting the brunt of these confirmed cases, shown in the maximum above (110,465 confirmed cases as of April 14th, 2020). Therefore, the spread of confirmed case counts is quite large.

This can also be seen when we plot a histogram of confirmed case counts by county:



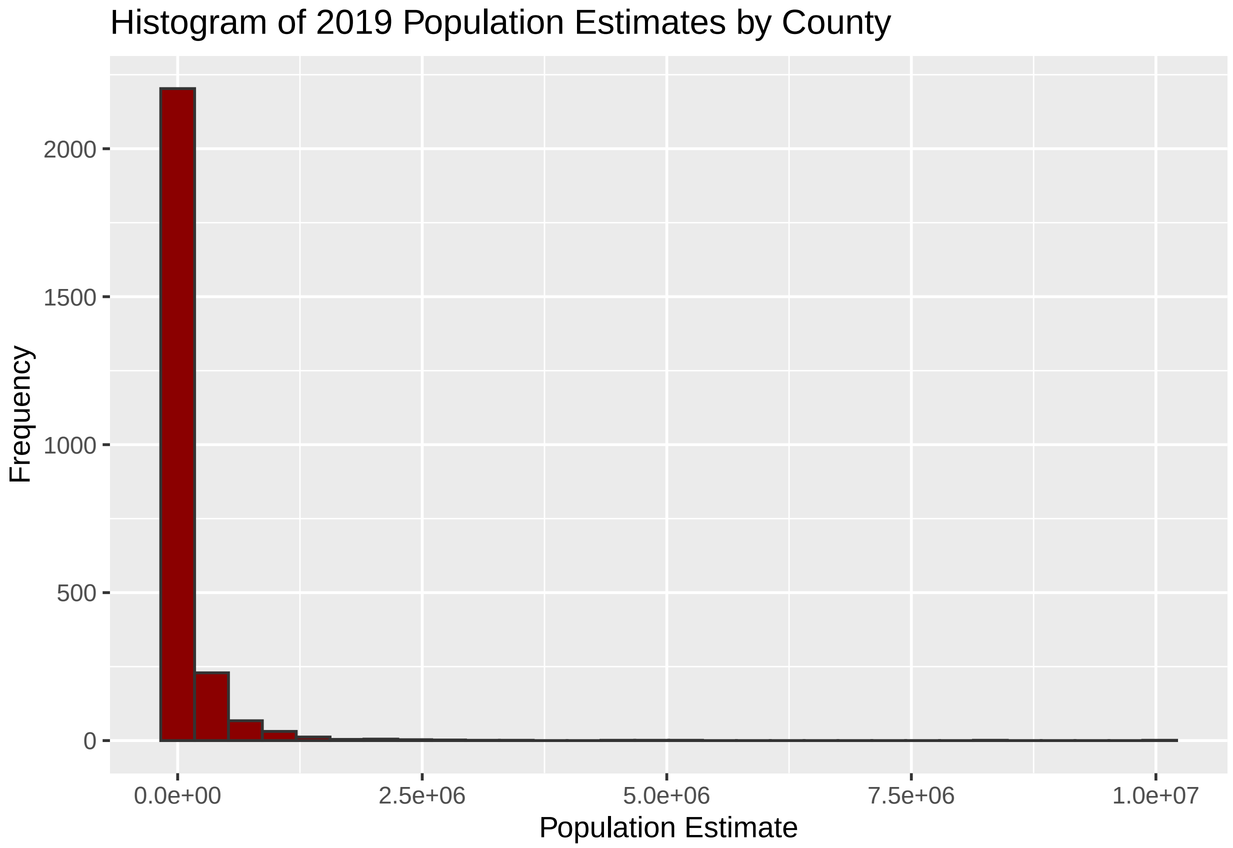
We'll do the same for the population estimates by county. Here's the summary:

```
summary(fnl$pop_estimate)
```

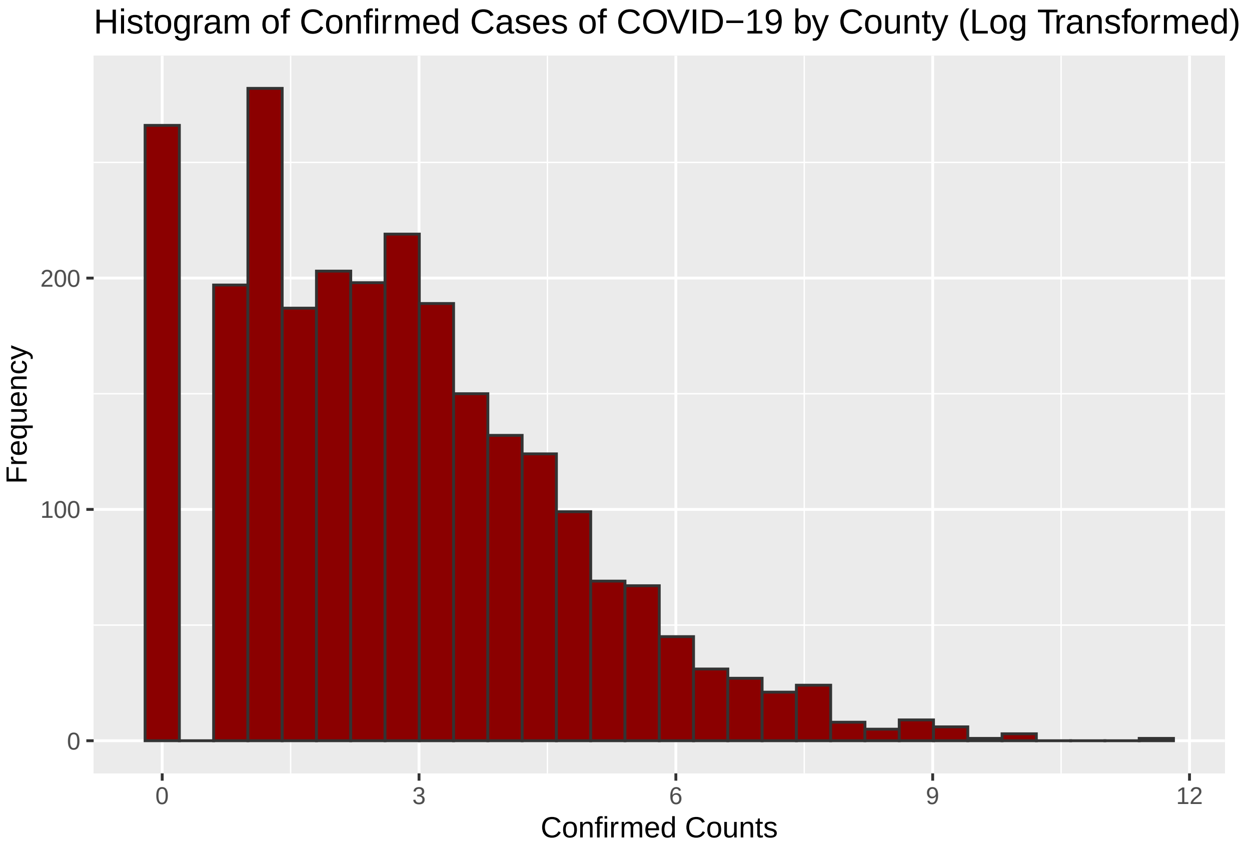
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      750    15727    33562   123134    85216 10039107
```

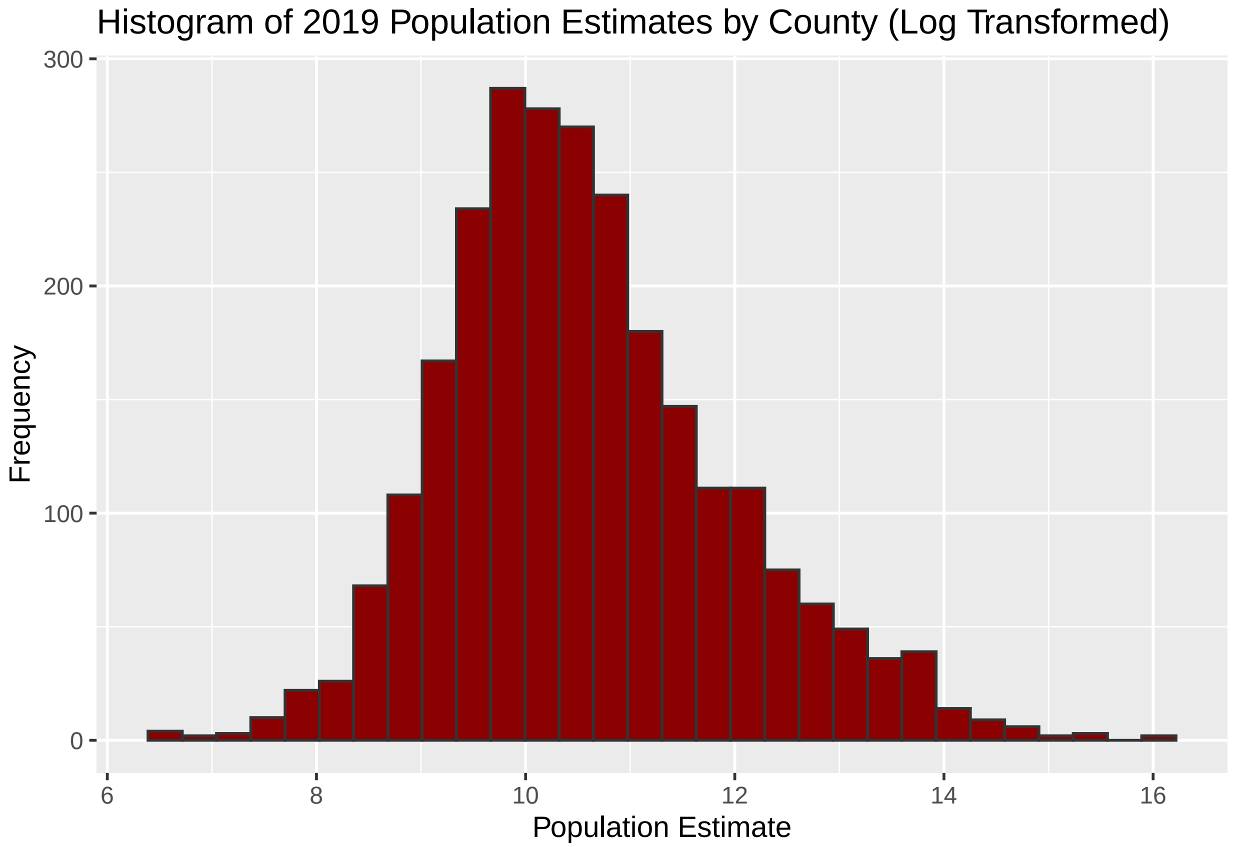
For the population estimates, on average, there are about 123,000 people in each county, but the mean appears to be skewed since the median value is only at roughly 33,500 people in each county, and 75% of counties have population estimates below 85,000 people. As expected, many people cluster in cities, but a majority of counties tend to be more rural and suburban, with fewer people.

This can also be seen when we plot a histogram of population estimates by county:



With this in mind, we may run into some issues building a reliable linear regression model if both our predictor and response variable(s) are heavily right skewed. I decided to attempt a log transformation of the population estimates and confirmed case counts to see if this will help my regression diagnostics later.


Histogram of Confirmed Cases of COVID−19 by County (Log Transformed)
Frequency
0
100
200
0
3
6
9
12
Confirmed Counts



As we can see, after conducting log transformations on both variables, we do start to see more normal distributions. We'll use these factors as we start to build our one-factor linear regression.

Building the One-Factor Linear Regression

With our factors ready to go, we can create a linear regression model.

```
covid_lm <- lm(fnl$log_pop ~ fnl$log_cases)
covid_lm
```

```
##
## Call:
## lm(formula = fnl$log_pop ~ fnl$log_cases)
##
## Coefficients:
##   (Intercept)  fnl$log_cases
##        8.9620         0.5914
```

Above, we can see the intercept and slope of our linear regression (8.962 and 0.5914 respectively). To get a more detailed outlook of the performance of our model, we can use the `summary()` function in R:

```r
summary(covid_lm)
```

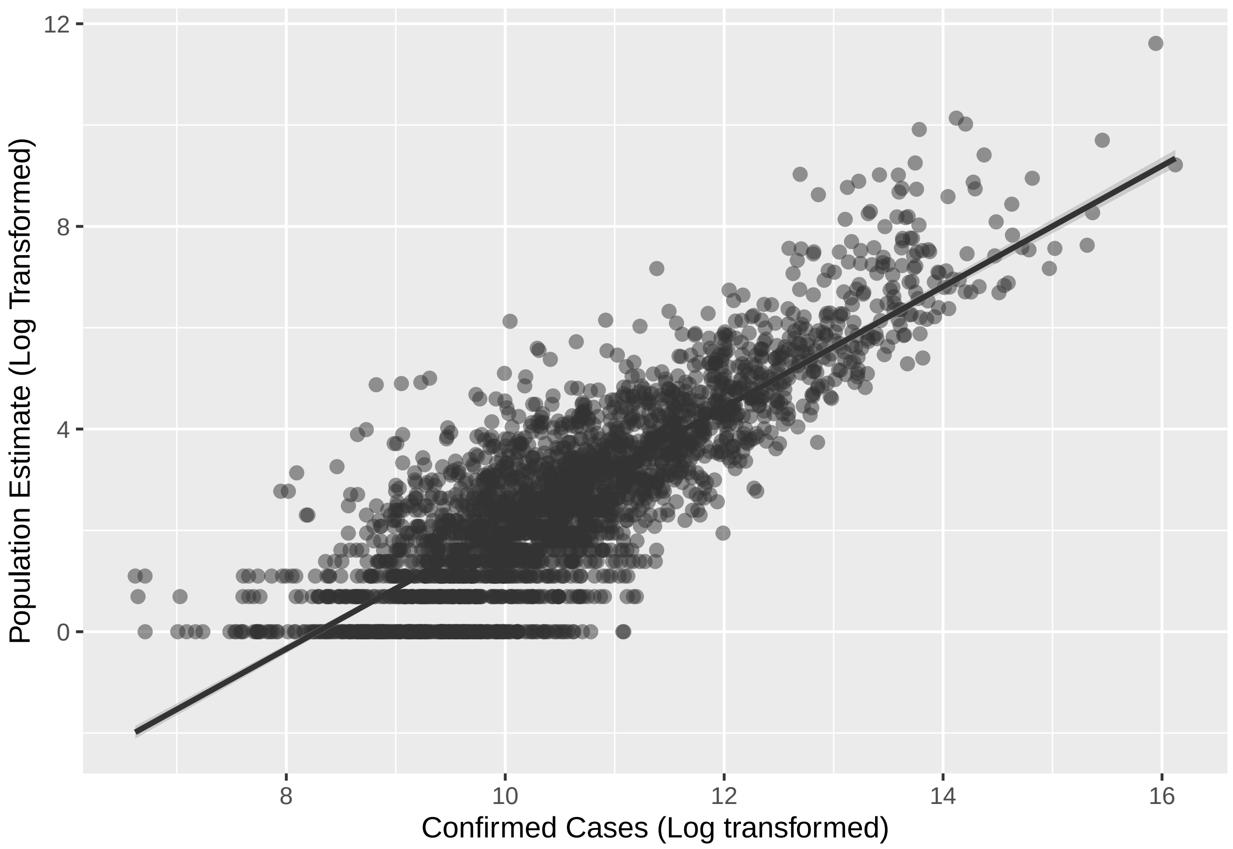
```
##
## Call:
## lm(formula = fnl$log_pop ~ fnl$log_cases)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02386 -0.43448  0.04335  0.48671  2.12144
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.96200    0.02526  354.77   <2e-16 ***
## fnl$log_cases  0.59142    0.00756   78.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7342 on 2561 degrees of freedom
## Multiple R-squared:  0.705,  Adjusted R-squared:  0.7049
## F-statistic:  6119 on 1 and 2561 DF,  p-value: < 2.2e-16
```

Evaluating the Model and Residual Analysis

After running the summary above, we can see a few things:

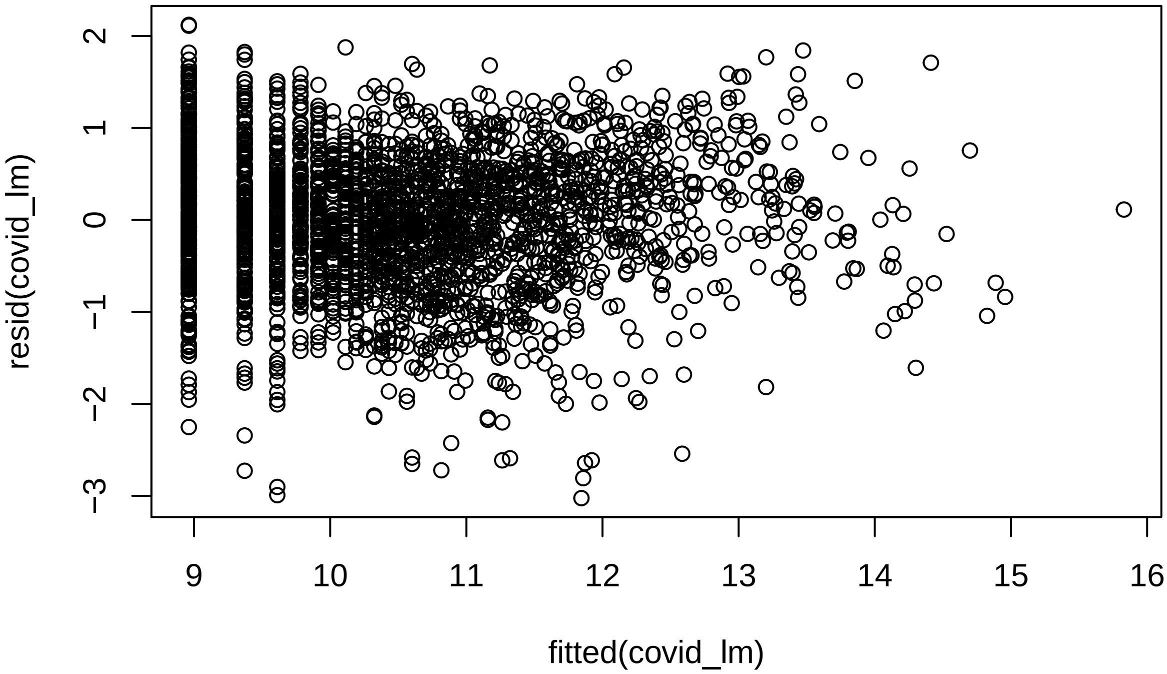
- The median residual value is around zero, which is a good sign.
- Additionally, the minimum and maximum values of the residuals are roughly around the same magnitude (3 and 2 respectively)
- The standard error appears to be at least five times smaller than the corresponding coefficient (0.5914 / 0.00756 = 78.2275)
- Our Multiple R-squared value is 0.705, which indicates that population size by county accounts for about 70.5% of the variability in the number of confirmed COVID-19 cases by county. This is a pretty good start for our regression model, especially with one factor.

Here's a plot of our model, with the line shown in black:



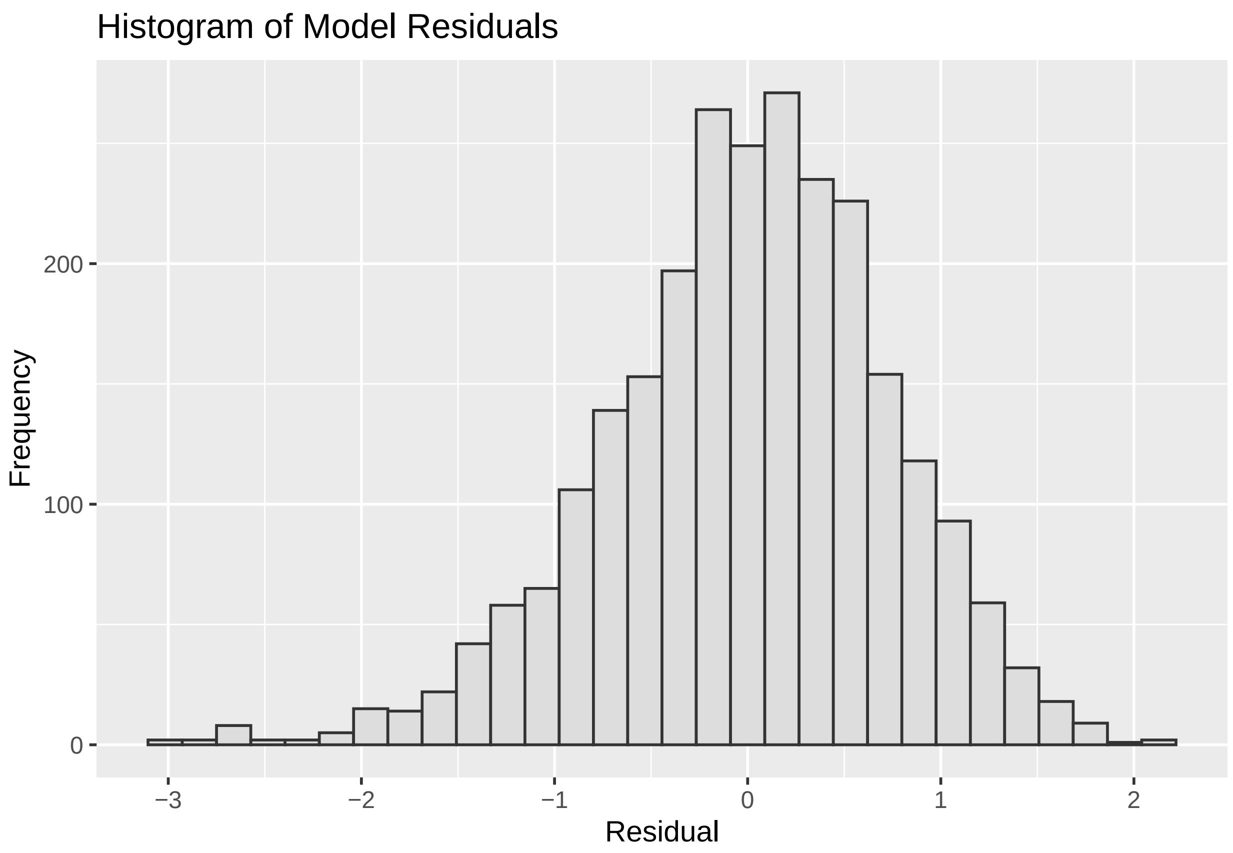
From above, we can see a pretty well-fit line, and when plotting residuals (below), we can confirm that most seem to be uniformly scattered above and below zero:

```
plot(fitted(covid_lm),resid(covid_lm))
```

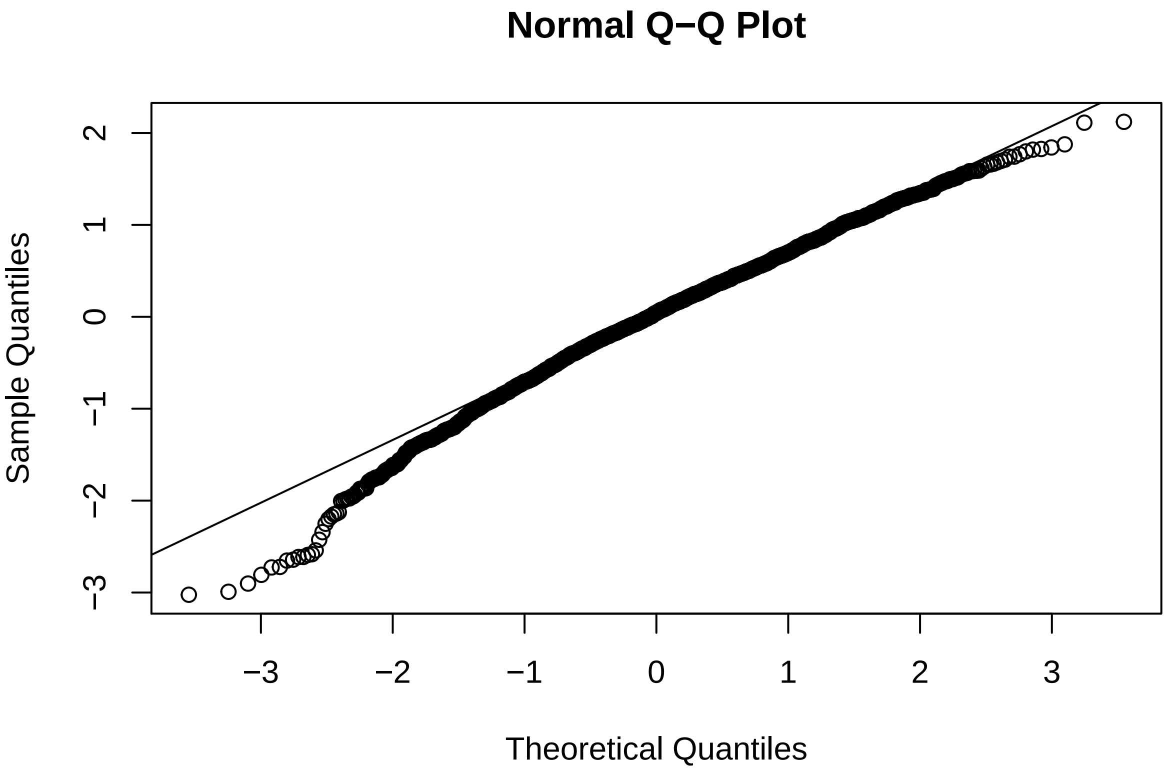


This can also be visualized in a basic histogram, or a quantile-versus-quantile (Q-Q) plot (see below):

```
p<-ggplot(fnl, aes(x=resid(covid_lm))) +
  geom_histogram(color="#333333", fill="#dddddd", bins = 30)
p<- p + ggtitle("Histogram of Model Residuals") +
  xlab("Residual") + ylab("Frequency")
p
```



```
qqnorm(resid(covid_lm))
qqline(resid(covid_lm))
```



Was the Linear Model Appropriate?

From this residual analysis, we can conclude that using population estimates to predict the number of confirmed COVID-19 cases is a pretty good start, and works as a decent predictor without adding other factors. However, it'll be interesting when I expand this work next week and add more factors to my regression model. I think at this point, without additional factors, we can say that this linear model is appropriate – however, next week I may have a different conclusion!