Regression Example: Transformations and COVID-19

Extra Topic

Spring 2020

Be advised

- This example contains COVID-19 data, if you prefer not to think about that, we will have somewhat similar examples on a worksheet posted on Moodle.
- The conditions of the model we will make are almost all violated. I want this to be an example of "bad modeling".
  - But the analysis and code are still completely correct.
- The interpretations are a little complex, and take some brain power to think about.

I think this is a really interesting topic and the ability to model what's happening in our world right now is very important. But the models that are useful are very complex and beyond the scope of Stat 212. That doesn't mean that we can't do some simple modeling (albeit, eventually incorrect modeling as mentioned above).

> I also think it is really important for YOU to be able to read the news about COVID-19 and question models that you see that try to predict what is and will happen moving forward. There are TONS of models out there, some good, and some not so good. I'd like you to have the ability to look at these models with a critical eye and make a decision yourself on how they meet conditions, or if the predictions are taking into account enough factors.

It's a very uncertain time and we all want answers and ways to inform our decisions. Data is an excellent way to help us do this, but it's important to remember a very famous (among statisticians at least) and appropriate quote, "**ALL models are WRONG... but SOME models are USEFUL.**" - George Box

An Example in R

modeling COVID-19 Cases

As I'm sure many of you are aware, there is constant pressure for us to track and model the rate of COVID-19 cases across the world. There are many issues with the data that we have (not enough tests, unreliable results, etc), but let's use some of the data from the US. Below is the total number of confirmed cases in the US from Jan. 22 through Mar. 31 (70 days).

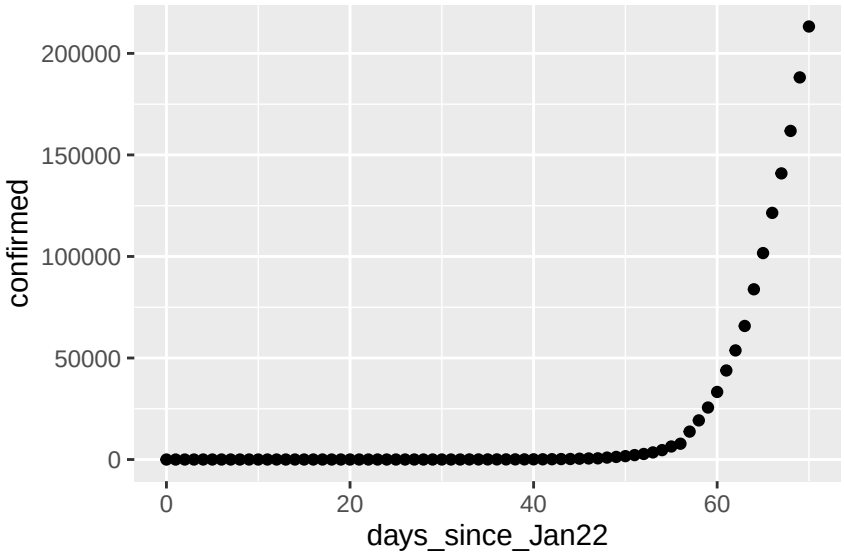
```
## # A tibble: 71 x 3
##    date                confirmed days_since_Jan22
##    <dttm>                  <dbl>            <dbl>
##  1 2020-01-22 00:00:00         1                0
##  2 2020-01-23 00:00:00         1                1
##  3 2020-01-24 00:00:00         2                2
##  4 2020-01-25 00:00:00         2                3
##  5 2020-01-26 00:00:00         5                4
##  6 2020-01-27 00:00:00         5                5
##  7 2020-01-28 00:00:00         5                6
##  8 2020-01-29 00:00:00         5                7
##  9 2020-01-30 00:00:00         5                8
```

```
## 10 2020-01-31 00:00:00         7                9
## # ... with 61 more rows
```

Let's consider whether we can use the date to predict the number of confirmed cases in the country.

- **Response:** Confirmed Cases
- **Explanatory:** Number of days since January 22

Take a look at our scatterplot



Uhhhh, that doesn't look good! It's not linear at all! So... what, are we done? Do we quit?

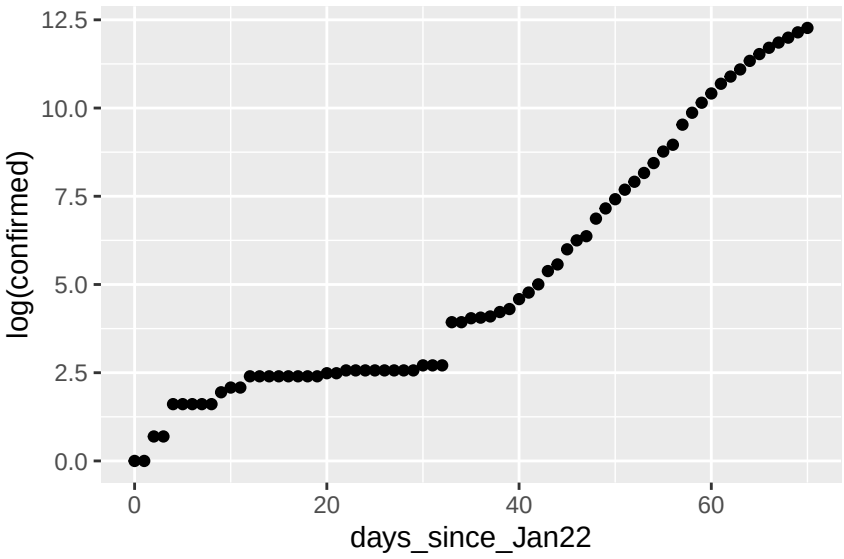
NEVER!

The spread of infectious diseases like COVID-19 have what's called an exponential growth curve. In fact, a lot of other really common numeric variables have exponential relationships, like income and housing prices. It happens when there's a lot of really small values and only a few really large ones. Luckily, we can fix this by **transforming** our response variable.

Transformations

Variable transformations are a common way for us to try and "force" a linear relationship. And the best way to transform exponential data is to use a **log transformation** (for those of you who are math majors, this transformation should make a lot of sense).

Let's try it out by using `log(confirmed)` instead of the original numbers.




```
## [1] 0.956451
```

It looks much better, and the correlation is really improved. But now if we move forward with this, **we NEED to remember that the response variable is transformed**.

Give it a shot

Can you:

- Find the fitted model for `log(confirmed)` and `days_since_Jan22`, write out the equation, and make a new scatterplot with the regression line on it?
- Test for the significance of the relationship (or you can think about it as testing `days_since_Jan22` as a significant predictor of `log(confirmed)`)?
- Predict the `log(confirmed)` cases for day 71 (April 1st) and day 90 (April 21)?
- Try to come up with a meaningful interpretation of the slope and the predictions you made?
  - This is not an easy task with the transformation, but give it a shot and I'll have my interpretations later.
- Check the conditions for the model using the `plot()` function?

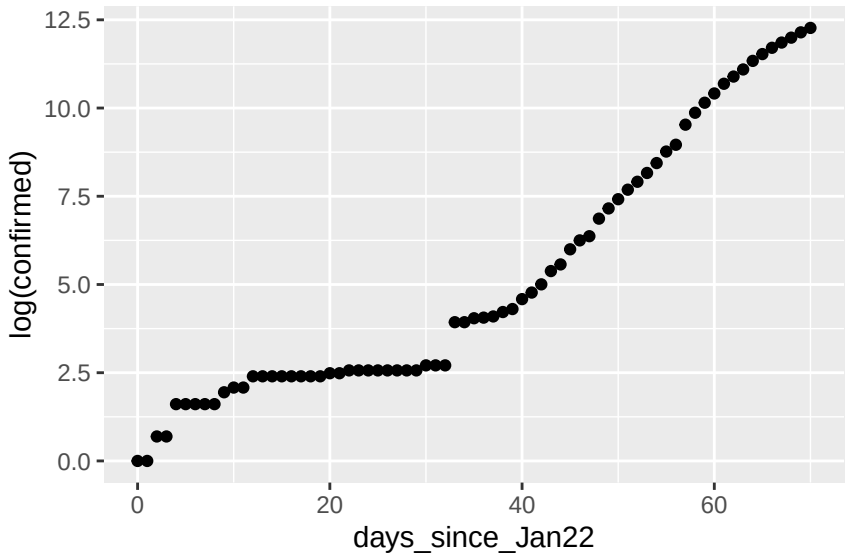
My Results

How did it go? The code for fitting the model and making the plots should have all been the same as the code used in the previous tutorials. The hard part is trying to explain what was going on. Hopefully my code and interpretations will make sense, we will have an example session this week on Wednesday and Friday. I will go over examples of regular linear regression as well as another example of transforming data for linear regression.

EDA

I'll start again with the scatterplot and correlation of the transformed data.

```
gf_point(log(confirmed) ~ days_since_Jan22, data = confirmed_march)
```




```
cor(log(confirmed_march$confirmed), confirmed_march$days_since_Jan22)
```

```
## [1] 0.956451
```

Fit the Model

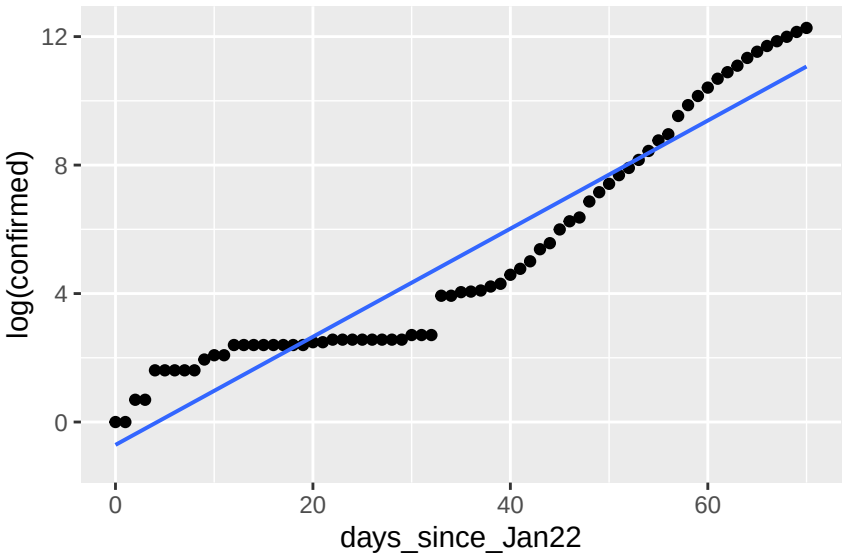
Next, I will fit the linear regression model, and write out the equation.

```
## Save the model using a unique name
covid_model <- lm(log(confirmed) ~ days_since_Jan22, data = confirmed_march)

## get the model coefficients
summary(covid_model)
```

```
##
## Call:
## lm(formula = log(confirmed) ~ days_since_Jan22, data = confirmed_march)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.96568 -1.00466  0.06503  1.04741  1.64733
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.710980   0.250745  -2.835    0.006 **
## days_since_Jan22  0.168272   0.006182  27.218   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.068 on 69 degrees of freedom
## Multiple R-squared:  0.9148, Adjusted R-squared:  0.9136
## F-statistic: 740.8 on 1 and 69 DF,  p-value: < 2.2e-16
```

```
## new scatterplot with the regression line
gf_point(log(confirmed) ~ days_since_Jan22, data = confirmed_march) %>%
  gf_lm()
```



$$\widehat{\log(\text{confirmed})} = -0.71 + 0.168 \cdot \text{days since Jan 22}$$

Test the slope

Using the summary from the model above, I can test for the significance of the slope of the model:

$H_0 : \beta_1 = 0$; The number of days since January 22 is not a significant predictor of the log(confirmed) cases in the US.

$H_a : \beta_1 \neq 0$; The number of days since January 22 is a good predictor of the log(confirmed) cases in the US.

We still need to talk in terms of the transformed response variable.

Test statistic	p-value
27.218	≈ 0

We have really strong evidence that the number of days since January 22 is a significant predictor of the log(confirmed) COVID-19 cases. We should have expected this, cases will increase as the days go by. I think the real potential for this model is whether it can make predictions about how many cases we should expect in the coming days.

Use the model for predictions

Predict the log(confirmed) cases for 71 days and 90 days since January 22.

```
## I can predict two new observations at the same time by using c()
new_days <- data.frame(days_since_Jan22 = c(71,90))

## You could plug in 71 and 90 to the regression equation to get exact values
## I am going to make a prediction interval for new observations
predict(covid_model, new_days, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 11.23635  9.04615 13.42654
## 2 14.43352 12.18408 16.68295
```

So I would predict with 95% confidence that on April 1st there would be between 9 and 13.5 log(confirmed) cases of COVID-19. And on April 21st, I'm 95% confident that there will be between 12.2 and 16.7 log(confirmed) cases of COVID-19.

> It may be hard for you to think about these numbers as meaningful. You're not alone, as humans, we don't think in terms of logarithms or exponentials well. It's hard for me to know if 14 log(confirmed) cases is good or bad. **We need to transform our data BACK in order for it to make sense!!**

Luckily, we have a way of doing this. Remember that in order to get rid of the exponential relationship we took the log of the response variable. Now, to transform back, we can take the exponential again and get values we can *actually* interpret. We are taking advantage of the mathematical property of:

$$e^{log(x)} = x$$

Interpret Transformed Data

For ALL of my interpretable values, I can take the exponential (`exp()`) to get the response variable back into just confirmed cases.

```
## My predictions - this step is only necessary if we've transformed the data
exp(predict(covid_model, new_days, interval = "prediction"))
```

```
##          fit        lwr      upr
## 1   75837.31   8485.807   677755
## 2 1855226.55 195650.063 17591947
```

Now, this makes more sense and our model is potentially more useful. We can predict that on April 1st, we are 95% confident the number of actual confirmed cases will be between 8,485 and 677,755. And on April 21st, we are 95% confident the number of actual confirmed cases would be between 195,650 and 17,591,947.

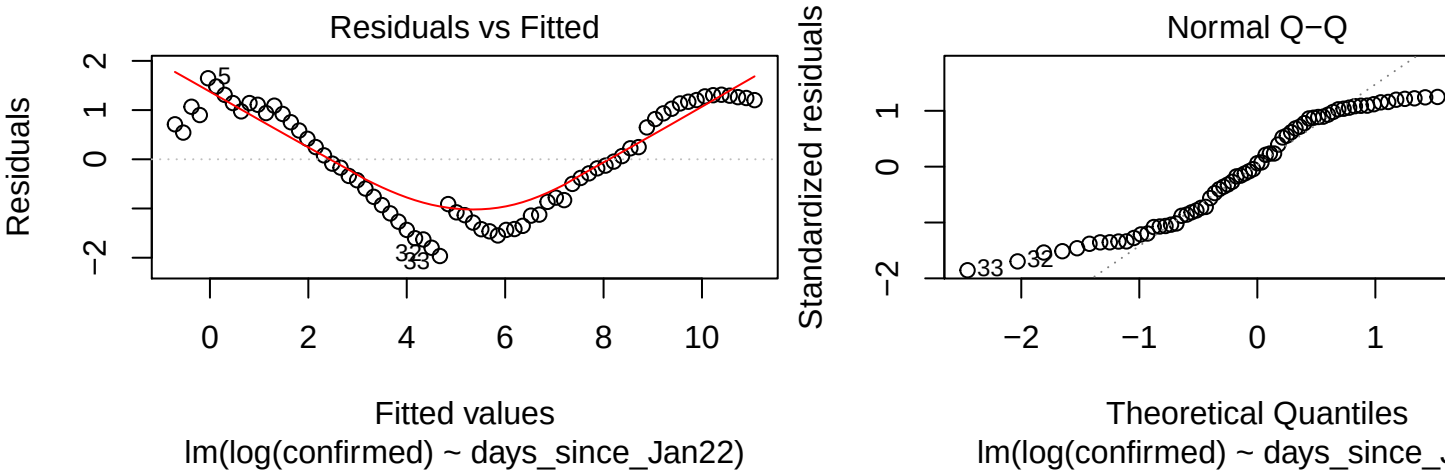
These are the predictions our model suggests, but there seems to be a some issues:

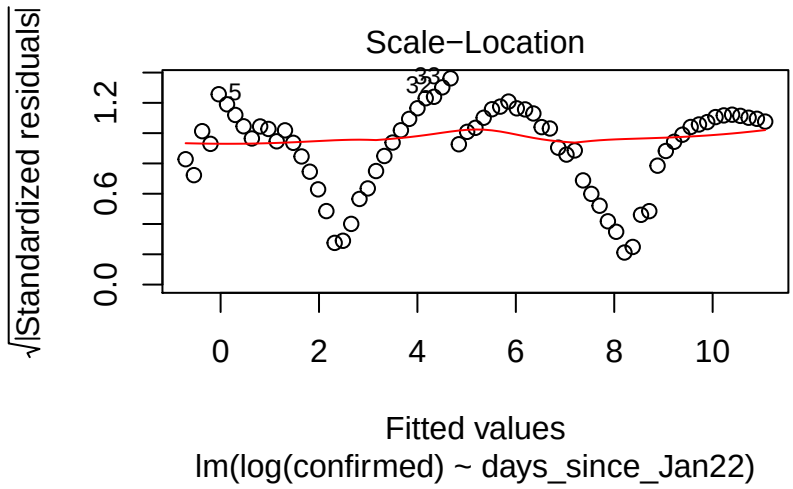
1. The range of possible confirmed cases is HUGE - This often the case when we transform data, there's a lot of uncertainty.
2. We are making predictions well outside of our observed data - Can we trust the predictions since we are **extrapolating** our model?

Check Conditions

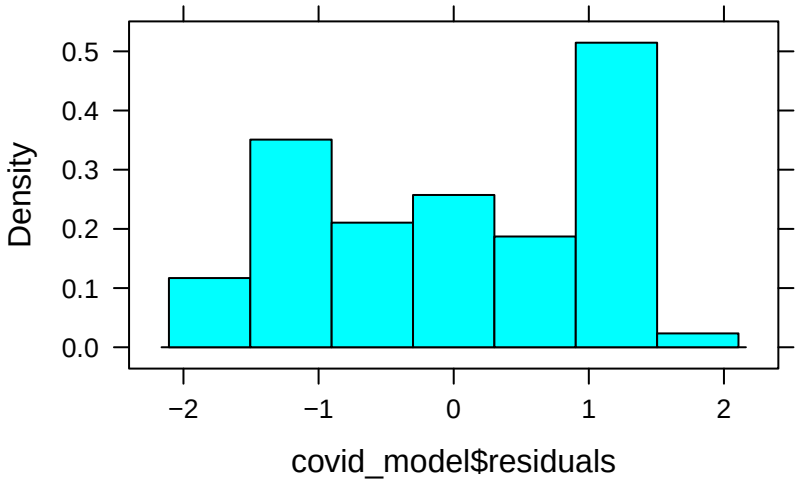
Check the conditions for using the model. Really we should have done this before making predictions, but I wanted to let you get some practice with reading the model summaries and making predictions.

```
## This gives the residual plot, Normal Q-Q plot, and Scale-Location plot
plot(covid_model, which = 1:3)
```






```
## This is the histogram of the residuals
histogram(covid_model$residuals)
```



Yikes! This does not look good. From our first plot (and maybe you noticed it in the scatterplot using log(confirmed)) we can see that the data are STILL not linear. Even with our log transformation, it wasn't enough to make the data linear. Sure, it made the data look *better*, but it didn't completely fix the problem.

Linearity condition is VIOLATED.

Looking at the Normal Q-Q plot and the histogram, we can also see that the points on the Q-Q plot don't follow the diagonal line very well, there is a distinct "S" pattern to the dots. And the histogram isn't really "Normal looking", I don't know how I'd describe it. . . just not really Normal. **Normality condition is VIOLATED**.

For constant variance, I don't think we necessarily have an issue, there don't seem to be areas on the scatterplot, or residual plot, or Scale-Location plot where the range from top to bottom is drastically larger/smaller on one side or the other. **Constant Variance is OK**.

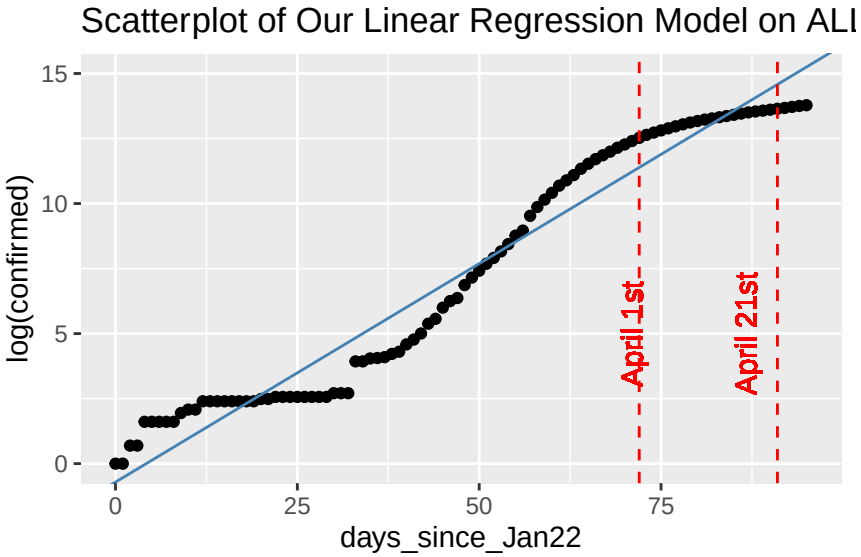
Finally, maybe the biggest issue we have here is the independence of observations. If each observation is another day since January 22, then hopefully it is clear that the cases are not independent. If we had 5 cases yesterday, I know that I'll have at least 5 cases today. **Independent observations condition is VIOLATED**.

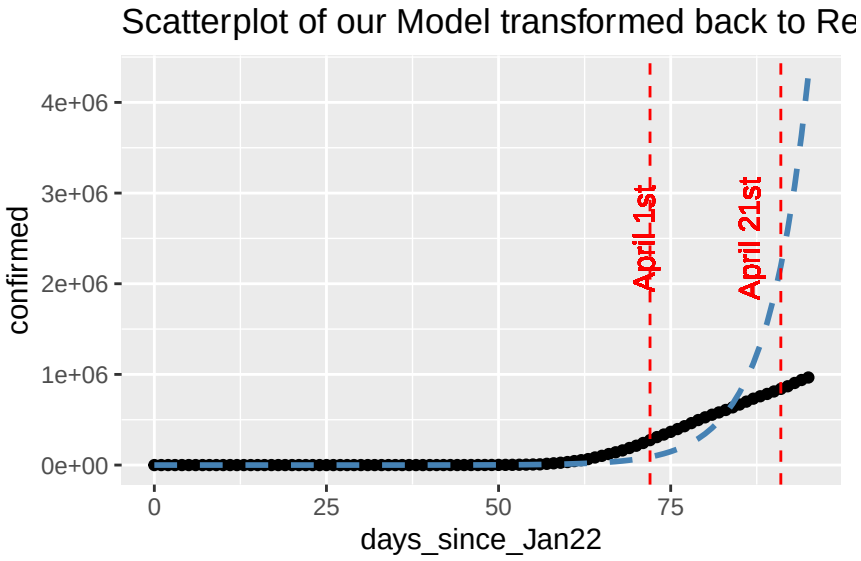
Linear regression, even with a transformation of data is NOT appropriate in this case. We have better ways of trying to model this type of complex data, many of which you can learn about in classes like Stat 272 - Statistical modeling.

What does violating the conditions lead to?

Let's take a look at our linear regression and prediction again with the bigger picture. We actually do know how many confirmed cases there were on April 1st and April 21st.

Date	Predicted log(confirmed) range	Actual log(confirmed) cases	Predicted cases range	Actual confirmed cases
4/1	[9 - 13.5]	12.4	[8,485 - 677,755]	243,622
4/21	[12.2 - 16.7]	13.6	[195,650 - 17,591,947]	811,865





What you can see in both the plots for the predicted log(confirmed) and actual confirmed cases is that our model **underestimated** the number of cases on April 1st, but totally **overestimated** the number of cases on April 21st.

Why would it do this?

One main reason is that this isn't a true exponential curve. Human intervention and public health policies have helped slow the spread of the disease. What you see is what we could expect to happen if we did nothing (our model) and what we actually observed is the results of "flattening the curve" through social distancing practices. Which is good, because our exponential model would not be a pretty picture.

The Takeaway

So why did I show you this example?

I think it is a good way to show you a common approach we use in statistics, log transformations. And beyond that, as I mentioned above, I wanted to expose you to a simplified version of many of the models you can find right now with a quick web search. And this method of looking at the data isn't actually that far off from what is really being done to analyze COVID-19 data. You have the basics down just with the information in Ch. 5.

I hope that you were able to follow this example without too much difficulty, and I hope that it was at least somewhat interesting to you.