COVID19 assignment for STAT 413/613

J. Wall

2020-03-31

We will use this assignment to look at COVID-19 data

# First we will look at data in the US by county

The first portion of this work is thanks to Max Kuhn of RStudio.

theme\_set(theme\_bw())  
  
dat <-  
 read\_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv")  
  
deaths\_by\_state <- dat %>%  
 group\_by(state) %>%  
 summarize(total\_deaths = max(deaths)) %>%  
 arrange(desc(total\_deaths))  
  
deaths\_by\_state

## # A tibble: 55 x 2  
## state total\_deaths  
## <chr> <dbl>  
## 1 New York 914  
## 2 Washington 152  
## 3 Louisiana 86  
## 4 Michigan 83  
## 5 California 44  
## 6 Illinois 44  
## 7 New Jersey 41  
## 8 Georgia 23  
## 9 Connecticut 21  
## 10 Nevada 15  
## # ... with 45 more rows

# total deaths   
sum(deaths\_by\_state$total\_deaths)

## [1] 1618

# Latest data from:  
max(dat$date)

## [1] "2020-03-30"

* **Exercise 1:** (1/2 point) Produce a tibble that has both deaths and total cases per state, arranged by the total number of cases in descending order

## # A tibble: 55 x 3  
## state total\_deaths total\_cases  
## <chr> <dbl> <dbl>  
## 1 New York 914 38087  
## 2 New Jersey 41 3840  
## 3 Illinois 44 3727  
## 4 Michigan 83 3195  
## 5 California 44 2474  
## 6 Washington 152 2332  
## 7 Florida 12 1700  
## 8 Louisiana 86 1480  
## 9 Connecticut 21 1445  
## 10 Massachusetts 10 1141  
## # ... with 45 more rows

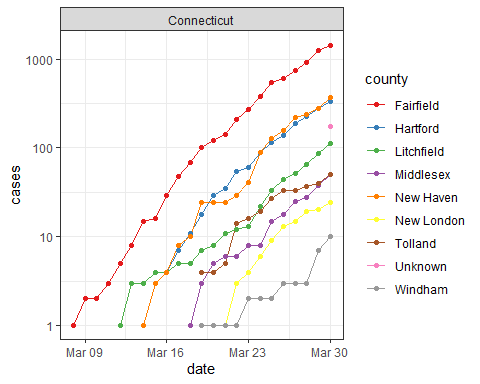
## [1] "total deaths = 1618"

## [1] "total cases = 71206"

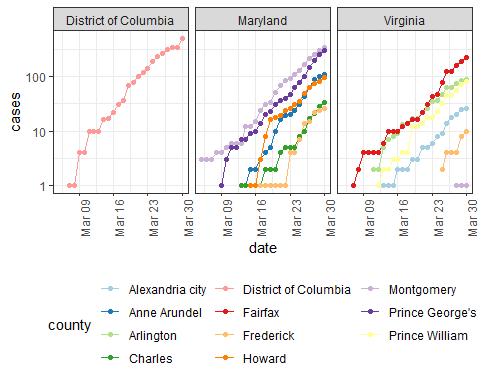
## [1] "latest data from: 2020-03-30"

Now we will choose just a few states to look at

dat\_small <-  
 dat %>%  
 filter(state %in% c("South Carolina", "Connecticut")) %>%  
 mutate(county = factor(county))  
  
dat\_small %>%  
 filter(state == "Connecticut") %>%  
 ggplot(aes(x = date, y = cases, group = county, col = county)) +  
 geom\_line() +  
 geom\_point() +  
 facet\_wrap(~ state) +  
 scale\_y\_log10() +  
 scale\_color\_brewer(palette = "Set1")



* **Exercise2:** (1 point) Do the analysis like above for the DMV area (DC, MD and VA) for both deaths and total cases. Use <http://www.theusgenweb.org/dcgenweb/geography/counties.shtml> for the counties in the DMV area. Be sure you get them all (hint: look at spelling in the dataset versus spelling you are entering)

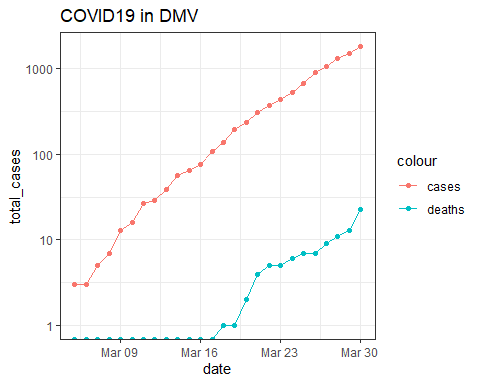
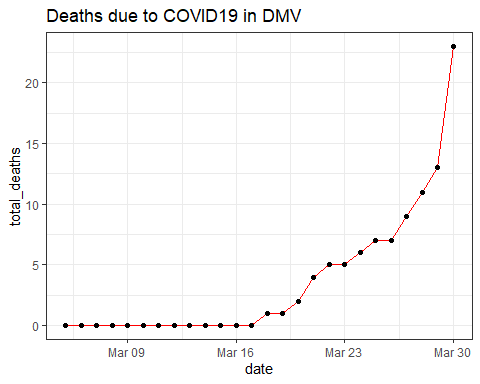
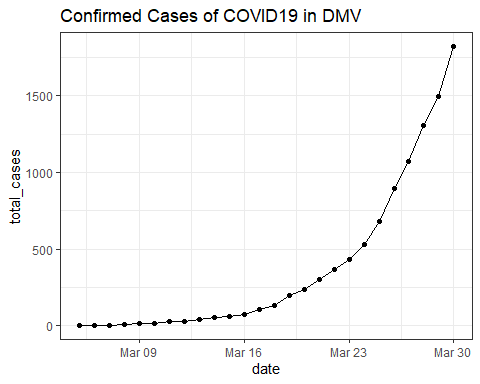


* **Exercise 3:** (1 point) Graph the total number of cases by date in the DMV area and the total deaths by date.

## [1] "total deaths = 23"

## [1] "total cases = 1822"

## [1] "latest data from: 2020-03-30"



# Get and clean worldwide Data

* **Exercise 4:** (1/2 point) Get data from (<https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/archived_data/archived_time_series/time_series_19-covid-Confirmed_archived_0325.csv>). After you have read in the dataset, create a new variable called Country/State that combines the country with the province/state and tidy the dataset.

## # A tibble: 6 x 7  
## `Country/State` `Province/State` `Country/Region` Lat Long Date   
## <chr> <chr> <chr> <dbl> <dbl> <date>   
## 1 Thailand <NA> Thailand 15 101 2020-01-22  
## 2 Thailand <NA> Thailand 15 101 2020-01-23  
## 3 Thailand <NA> Thailand 15 101 2020-01-24  
## 4 Thailand <NA> Thailand 15 101 2020-01-25  
## 5 Thailand <NA> Thailand 15 101 2020-01-26  
## 6 Thailand <NA> Thailand 15 101 2020-01-27  
## # ... with 1 more variable: Confirmed\_Cases <dbl>

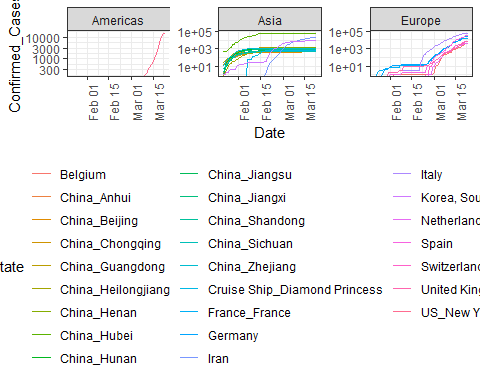
* **Exercise 5:** (1/2 point) Add a new variable called continent to the dataset. Be sure there are no NA’s for continent.

## # A tibble: 6 x 8  
## `Country/State` `Province/State` `Country/Region` Lat Long Date   
## <chr> <chr> <chr> <dbl> <dbl> <date>   
## 1 Thailand <NA> Thailand 15 101 2020-01-22  
## 2 Thailand <NA> Thailand 15 101 2020-01-23  
## 3 Thailand <NA> Thailand 15 101 2020-01-24  
## 4 Thailand <NA> Thailand 15 101 2020-01-25  
## 5 Thailand <NA> Thailand 15 101 2020-01-26  
## 6 Thailand <NA> Thailand 15 101 2020-01-27  
## # ... with 2 more variables: Confirmed\_Cases <dbl>, continent <chr>

* **Exercise 6:** (1/2 point) Find the 25 Country/State’s with the most Confirmed Cases (Hint: use group\_by, summarize and ungroup?)

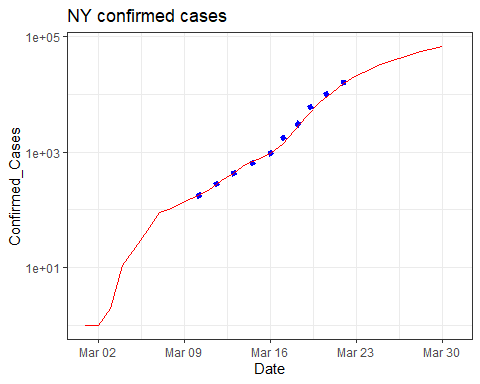
## # A tibble: 25 x 4  
## `Country/State` `Country/Region` continent ttl  
## <chr> <chr> <chr> <dbl>  
## 1 China\_Hubei China Asia 2894885  
## 2 Italy Italy Europe 497959  
## 3 Iran Iran Asia 252770  
## 4 Spain Spain Europe 186200  
## 5 Korea, South Korea, South Asia 181699  
## 6 Germany Germany Europe 160974  
## 7 France\_France France Europe 117724  
## 8 China\_Guangdong China Asia 67015  
## 9 US\_New York US Americas 64538  
## 10 China\_Henan China Asia 61811  
## 11 China\_Zhejiang China Asia 61207  
## 12 China\_Hunan China Asia 50256  
## 13 China\_Anhui China Asia 47647  
## 14 Switzerland Switzerland Europe 46413  
## 15 China\_Jiangxi China Asia 45181  
## 16 United Kingdom\_United Kingdom United Kingdom Europe 35081  
## 17 China\_Shandong China Asia 33603  
## 18 China\_Jiangsu China Asia 30272  
## 19 China\_Chongqing China Asia 28659  
## 20 Netherlands\_Netherlands Netherlands Europe 28030  
## 21 Cruise Ship\_Diamond Princess Cruise Ship Asia 26228  
## 22 China\_Sichuan China Asia 25719  
## 23 China\_Heilongjiang China Asia 21991  
## 24 Belgium Belgium Europe 21324  
## 25 China\_Beijing China Asia 20967

* **Exercise 7:** (1/2 point) Plot the number of cases over time for the top 25 country/states faceting by continent. Use appropriate scales for the axes.



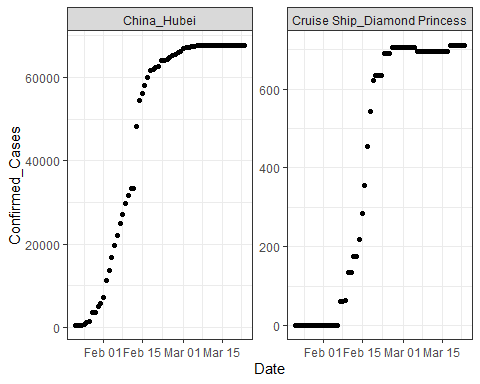
* **Exercise 8:** (1 point) Get the data for the state of NY from this csse dataset and compare the growth of cases from this dataset to the growth shown in the nytimes dataset used for exercises 1-3. Does the data match? If not, how is it different?

## # A tibble: 30 x 3  
## Date Confirmed\_Cases n  
## <date> <dbl> <dbl>  
## 1 2020-03-01 NA 1  
## 2 2020-03-02 NA 1  
## 3 2020-03-03 NA 2  
## 4 2020-03-04 NA 11  
## 5 2020-03-05 NA 22  
## 6 2020-03-06 NA 44  
## 7 2020-03-07 NA 89  
## 8 2020-03-08 NA 106  
## 9 2020-03-09 NA 142  
## 10 2020-03-10 173 173  
## 11 2020-03-11 220 217  
## 12 2020-03-12 328 326  
## 13 2020-03-13 421 421  
## 14 2020-03-14 525 610  
## 15 2020-03-15 732 732  
## 16 2020-03-16 967 950  
## 17 2020-03-17 1706 1374  
## 18 2020-03-18 2495 2382  
## 19 2020-03-19 5365 4152  
## 20 2020-03-20 8310 7102  
## 21 2020-03-21 11710 10356  
## 22 2020-03-22 15793 15168  
## 23 2020-03-23 15793 20875  
## 24 2020-03-24 NA 25666  
## 25 2020-03-25 NA 33067  
## 26 2020-03-26 NA 38988  
## 27 2020-03-27 NA 44636  
## 28 2020-03-28 NA 53364  
## 29 2020-03-29 NA 59568  
## 30 2020-03-30 NA 67216



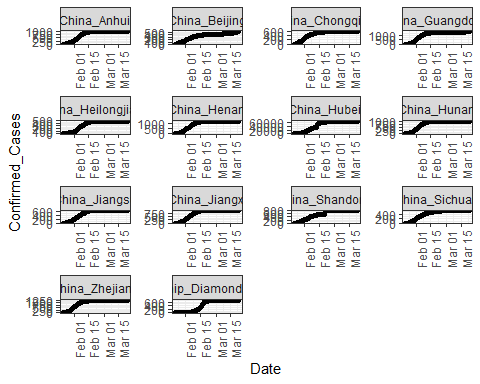
Let’s take a look at situations where the growth appears to have stopped. In particular, we start by visualizing the cases over time in the Hubei province of China and the Diamond Princess cruise ship.

Slowed\_cases <- plot\_data %>%   
 filter(`Country/State` %in% c("China\_Hubei", "Cruise Ship\_Diamond Princess"))   
Slowed\_cases %>%  
 ggplot(aes(x = Date, y = Confirmed\_Cases)) +  
 geom\_point() +  
 facet\_wrap(~ `Country/State`, scales = "free") #+



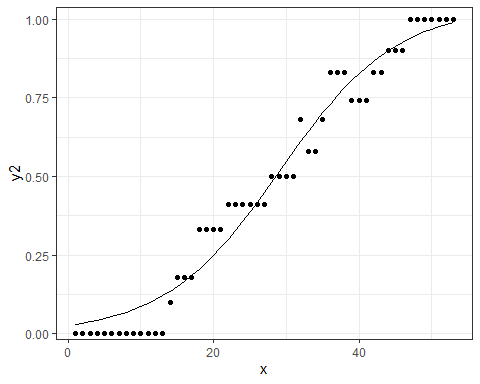
# scale\_y\_log10()

* **Exercise 9:** (1/2 point) Visualize cases from China and the Diamond Princess over time.



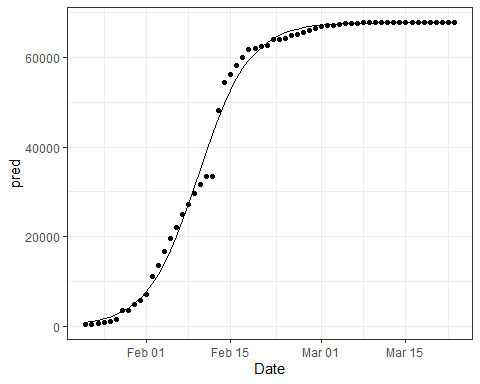
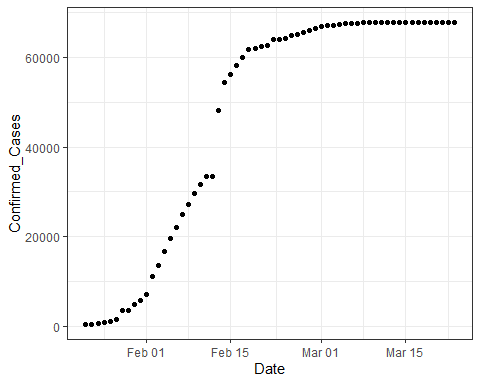
These graphs look like sigmoid functions. Here is a demo of fitting a sigmoid function.

# function needed for visualization purposes  
sigmoid = function(x, params) {  
 params[1] / (1 + exp(-params[2] \* (x - params[3])))  
}  
   
x = 1:53  
y = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0.1,0.18,0.18,0.18,0.33,0.33,0.33,0.33,0.41,  
 0.41,0.41,0.41,0.41,0.41,0.5,0.5,0.5,0.5,0.68,0.58,0.58,0.68,0.83,0.83,0.83,  
 0.74,0.74,0.74,0.83,0.83,0.9,0.9,0.9,1,1,1,1,1,1,1)  
df <- tibble(x = x, y = y)   
# fitting code  
fitmodel <- nls(y ~ a /(1 + exp(-b \* (x - c))), data = df,  
 start = list(a = 1, b = 0.5, c = 25))  
   
# visualization code  
# get the coefficients using the coef function  
params=coef(fitmodel)  
   
df$y2 <- sigmoid(x, params)  
df %>% ggplot(aes(x, y2)) + geom\_line() + geom\_point(y = y)



A more generalized sigmoid function is of the form

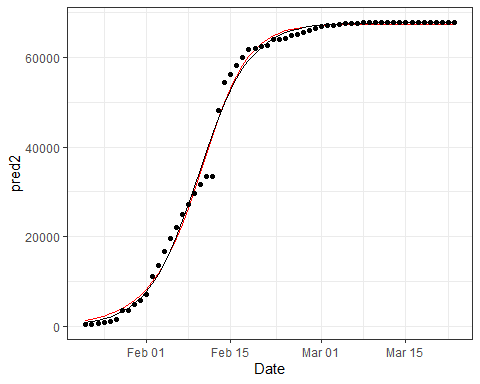
We will try using that form for the Hubei province and fitting a sigmoid function to that data. -**Exercise 10:** (1 point) Model the Hubei data with the simple form demonstrated above. Plot your results on top of the data points. Use broom::glance() and summary to look at model results.



##   
## Formula: Confirmed\_Cases ~ K/(1 + exp(-B \* (date\_int - t0)))  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## K 6.773e+04 3.839e+02 176.44 <2e-16 \*\*\*  
## B 2.348e-01 7.916e-03 29.66 <2e-16 \*\*\*  
## t0 1.830e+04 1.660e-01 110241.51 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2002 on 59 degrees of freedom  
##   
## Number of iterations to convergence: 6   
## Achieved convergence tolerance: 2.132e-06

## # A tibble: 1 x 8  
## sigma isConv finTol logLik AIC BIC deviance df.residual  
## <dbl> <lgl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>  
## 1 2002. TRUE 0.00000213 -558. 1123. 1132. 236424289. 59

-**Exercise 11:** (1 point) Now develop a second model and apply it to the same Hubei data. Graph your model results on top of the data and use summary and glance to look at your results. This model family will be



##   
## Formula: Confirmed\_Cases ~ K/((1 + exp(-B \* (date\_int - t0))))^(1/v)  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## K 6.743e+04 3.909e+02 172.490 < 2e-16 \*\*\*  
## B 2.799e-01 2.746e-02 10.196 1.50e-14 \*\*\*  
## t0 1.830e+04 1.018e+00 17983.849 < 2e-16 \*\*\*  
## v 1.490e+00 2.885e-01 5.164 3.09e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1976 on 58 degrees of freedom  
##   
## Number of iterations to convergence: 11   
## Achieved convergence tolerance: 9.751e-06

## # A tibble: 1 x 8  
## sigma isConv finTol logLik AIC BIC deviance df.residual  
## <dbl> <lgl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>  
## 1 1976. TRUE 0.00000975 -556. 1123. 1133. 226420388. 58

* **Exercise 12:** (2 points) Now use a map function and nested grouped data to develop models out of this more complicated model family for the Diamond Princess and each China provinces except Beijing. Add the model to the nested data. Use broom::tidy() to add the coefficients of the models to your dataset. Use pivot\_wider to make these coefficients into columns.

## # A tibble: 13 x 7  
## # Groups: Country/State, Province/State, Country/Region, Lat, Long, continent  
## # [13]  
## `Country/State` `Province/State` `Country/Region` Lat Long continent data   
## <chr> <chr> <chr> <dbl> <dbl> <chr> <lis>  
## 1 China\_Hubei Hubei China 31.0 112. Asia <tib~  
## 2 China\_Guangdong Guangdong China 23.3 113. Asia <tib~  
## 3 China\_Henan Henan China 33.9 114. Asia <tib~  
## 4 China\_Zhejiang Zhejiang China 29.2 120. Asia <tib~  
## 5 China\_Hunan Hunan China 27.6 112. Asia <tib~  
## 6 China\_Anhui Anhui China 31.8 117. Asia <tib~  
## 7 China\_Jiangxi Jiangxi China 27.6 116. Asia <tib~  
## 8 China\_Shandong Shandong China 36.3 118. Asia <tib~  
## 9 Cruise Ship\_Di~ Diamond Princess Cruise Ship 35.4 140. Asia <tib~  
## 10 China\_Jiangsu Jiangsu China 33.0 119. Asia <tib~  
## 11 China\_Chongqing Chongqing China 30.1 108. Asia <tib~  
## 12 China\_Sichuan Sichuan China 30.6 103. Asia <tib~  
## 13 China\_Heilongj~ Heilongjiang China 47.9 128. Asia <tib~

## # A tibble: 52 x 13  
## # Groups: Province/State, Lat, Long, Country/State, Country/Region, continent  
## # [13]  
## `Province/State` Lat Long term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Hubei 31.0 112. K 6.74e+4 391. 172. 2.52e- 80  
## 2 Hubei 31.0 112. B 2.80e-1 0.0275 10.2 1.50e- 14  
## 3 Hubei 31.0 112. t0 1.83e+4 1.02 17984. 2.37e-197  
## 4 Hubei 31.0 112. v 1.49e+0 0.288 5.16 3.09e- 6  
## 5 Guangdong 23.3 113. K 1.36e+3 2.57 529. 1.51e-108  
## 6 Guangdong 23.3 113. B 2.11e-1 0.00600 35.2 7.88e- 41  
## 7 Guangdong 23.3 113. t0 1.83e+4 1.38 13246. 1.19e-189  
## 8 Guangdong 23.3 113. v 2.86e-1 0.0640 4.48 3.59e- 5  
## 9 Henan 33.9 114. K 1.28e+3 1.71 745. 3.77e-117  
## 10 Henan 33.9 114. B 2.38e-1 0.00511 46.5 1.28e- 47  
## # ... with 42 more rows, and 5 more variables: `Country/State` <chr>,  
## # `Country/Region` <chr>, continent <chr>, data <list>, model <list>

## # A tibble: 13 x 8  
## `Province/State` Lat Long K B t0 v t0\_date   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <date>   
## 1 Hubei 31.0 112. 67430. 0.280 18304. 1.49 2020-02-11  
## 2 Guangdong 23.3 113. 1362. 0.211 18287. 0.286 2020-01-26  
## 3 Henan 33.9 114. 1276. 0.238 18292. 0.526 2020-01-31  
## 4 Zhejiang 29.2 120. 1228. 0.107 18259. 0.0262 2019-12-29  
## 5 Hunan 27.6 112. 1021. 0.234 18290. 0.449 2020-01-29  
## 6 Anhui 31.8 117. 993. 0.271 18296. 0.888 2020-02-03  
## 7 Jiangxi 27.6 116. 938. 0.260 18294. 0.636 2020-02-01  
## 8 Shandong 36.3 118. 784. 0.108 18272. 0.0721 2020-01-11  
## 9 Diamond Princess 35.4 140. 701. 0.632 18310. 2.48 2020-02-18  
## 10 Jiangsu 33.0 119. 635. 0.223 18293. 0.583 2020-02-01  
## 11 Chongqing 30.1 108. 579. 0.181 18283. 0.191 2020-01-22  
## 12 Sichuan 30.6 103. 544. 0.143 18239. 0.000387 2019-12-08  
## 13 Heilongjiang 47.9 128. 483. 0.241 18296. 0.642 2020-02-04

Now you would be free to look at the t0\_date in comparison to when the first n cases appeared in that province or look at K in relation to the total population of the province. If you find a pattern there, perhaps see if it applies to those datasets that have not yet flattened out.

-**Extra Credit** (1 point) Use any of the COVID19 data to analyze anything of particular interest to you. Explain what you are doing and why and what did you find out.

I hope you had fun with this timely dataset!!