Cleaning COVID-19 Data from Johns Hopkins

Bill Kayser

Contents

Initialize the Notebook loading the following required libraries.

```
library(tidyverse)
library(lubridate)
library(scales)
library(wpp2019)
theme_set(theme_light())
```

Load Datasets

We are pulling in data from the Johns Hopkins COVID-19 repository as well as world population data from wpp2019 and US Census information on county populations

COVID-19 Data

Let's update the CV data directly from the Johns Hopkins git repo. The first thing to do is check for any daily updates by updating the git submodule:

```
system('git submodule update --remote COVID-19')
```

Load the timeseries data. We'll need to convert the table from a wide format where the time series is in columns to a narrow format where each observation is a single datapoint for a day.

This function will process the different timeseries files uniformly.

```
read_and_clean <- function(infile) {
  # Read the data
  cv.wide <- read.csv(infile)
  if ('Country.Region' %in% names(cv.wide)) {
    # This is the countries data so column names need to be adjusted
    cv.wide <- rename(cv.wide,
                      Province_State=Province.State,
                      Country_Region=Country.Region,
                      Long_=Long) %>%
      mutate(Admin2=NA,
             Combined_Key=str_c(Country_Region, ", ", Province_State))
  } else {
    # There are two counties without FIPS codes for Kansas City, and "Dukes and Nantucket".  I think th
    cv.wide <- filter(cv.wide,
                      !is.na(FIPS))
```

```
  }
  # Identify the date columns so we can gather them
  datecols <- grep('^X', names(cv.wide))

  # Gather the date columns into a single pair of columns, "date" and "count"
  # This converts the data from a wide format to a narrow format more amenable to
  # graphing and reshaping
  gather(cv.wide, key='date', value='count', datecols) %>%
    select(State=Province_State, starts_with('FIPS'), County=Admin2, Country=Country_Region, Lat, Long=
    mutate(Date=mdy(str_sub(Date, 2))) %>%
    as_tibble()
}
```

Load US Timeseries Data

```
cases <- read_and_clean('COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confi
deaths <- read_and_clean('COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deat
```

Load International Timeseries Data

```
countries.cases <- read_and_clean('COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_series_cov
countries.deaths <- read_and_clean('COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_series_co
```

Load Census Data

This is how you could get data directly using the US Census bureau APIs but the data appears to be too fine grained. We can use this later if we need more detailed census information.

```
get_estimates(product="population", key=Sys.getenv('CENSUS_KEY'))
```

This is just the static datafile of summary census statistics. Good enough for now.

```
regions <- c('Northeast', 'Midwest', 'South', 'West')
divisions <- c('N Atlantic', 'Mid Atlantic', 'E North Ctrl', 'W North Ctrl', 'S Atlantic', 'E South Ctr

census_data <- read.csv('https://www2.census.gov/programs-surveys/popest/datasets/2010-2019/counties/to
  mutate(Region = as.factor(regions[REGION]),
         Division = as.factor(divisions[DIVISION]),
         FIPS=as.integer(str_c(STATE,sprintf('%0.3i', COUNTY)))) %>%
  select(Region,
         State=STNAME,
         Division,
         FIPS,
         CountyCode=COUNTY,
         CTYNAME,
         Population=POPESTIMATE2019,
         Annual.Deaths=DEATHS2019) %>%
   as_tibble()

counties <- filter(census_data, CountyCode > 0) %>% select(-State, -CTYNAME)
states <- filter(census_data, CountyCode == 0) %>% select(-CountyCode)
```

Prepare Data

Join Tables

Join the US confirmed cases to deaths and census population data. Create a single table, `cvdata.us`.

```
cvdata.us <- left_join(cases,
                       select(deaths, Key, Date, Count),
                       by=c('Key', 'Date')) %>%
  rename(Cases = Count.x,
         Deaths = Count.y) %>%
  left_join(counties, by='FIPS') %>%
  filter()
```

Join the cases and deaths for country data into a single table, `cvdata.i18n`.

```
cvdata.i18n <- left_join(countries.cases,
                         select(countries.deaths, Key, Date, Deaths=Count),
                         by=c('Key', 'Date')) %>%
  rename(Cases=Count)
```

Add in First Derivative

A lot of sources present "New Cases" but I don't have that data yet so I can calculate the day to day change. I don't think this is the same as new cases because the day to day change presumably factors in deaths and recoveries.

It will be interesting to study not just the day to day numbers, but the change in the numbers from day to day. I will manually calculate this first derivative and store the values in colums appended with `.Diff`.

I can accurately calculate "New Deaths".

```
# sort by locale the date
cvdata.us <- arrange(cvdata.us, Key, Date)
boundaries <- which(cvdata.us$Key[2:nrow(cvdata.us)] != cvdata.us$Key[1:nrow(cvdata.us)-1])

# Calculate differential
X1 <- c(cvdata.us$Cases)
X0 <- c(0, cvdata.us$Cases)[1:length(X1)]
diff <- X1 - X0

diff[boundaries + 1] <- 0

cvdata.us$Cases.Diff <- diff

X1 <- c(cvdata.us$Deaths)
X0 <- c(0, cvdata.us$Deaths)[1:length(X1)]
diff <- X1 - X0
diff[boundaries + 1] <- 0

cvdata.us$Deaths.Diff <- diff
```

Derive Additional Tables

Group data by state and save into table `cvdata.us.by_state`.

```
cvdata.us.by_state <- cvdata.us %>%
  group_by(State, Date) %>%
```

```
  summarize(Cases = sum(Cases),
            Cases.Diff = sum(Cases.Diff),
            Deaths = sum(Deaths),
            Deaths.Diff = sum(Deaths.Diff)) %>%
  ungroup(State, Date) %>%
  left_join(states, by=c('State')) %>% mutate(State = as.factor(State)) %>%
  filter(!is.na(Region))
```

```
## Warning: Column `State` joining factors with different levels, coercing to
## character vector
```

Create a single timeseries table just for Italy called `italy`.

```
italy <- cvdata.i18n %>%
  filter(Country == 'Italy') %>%
  select(-State, -County, -Key, -Lat, -Long, -Country)
```

Data Exploration

For the US Data I'm just showing a few states so the plots are more readable. This is just preliminary exploration as I get to know the data and clean it for other purposes.

Let's first set up some parameters for the graph's X axis.

Top Counties Infected

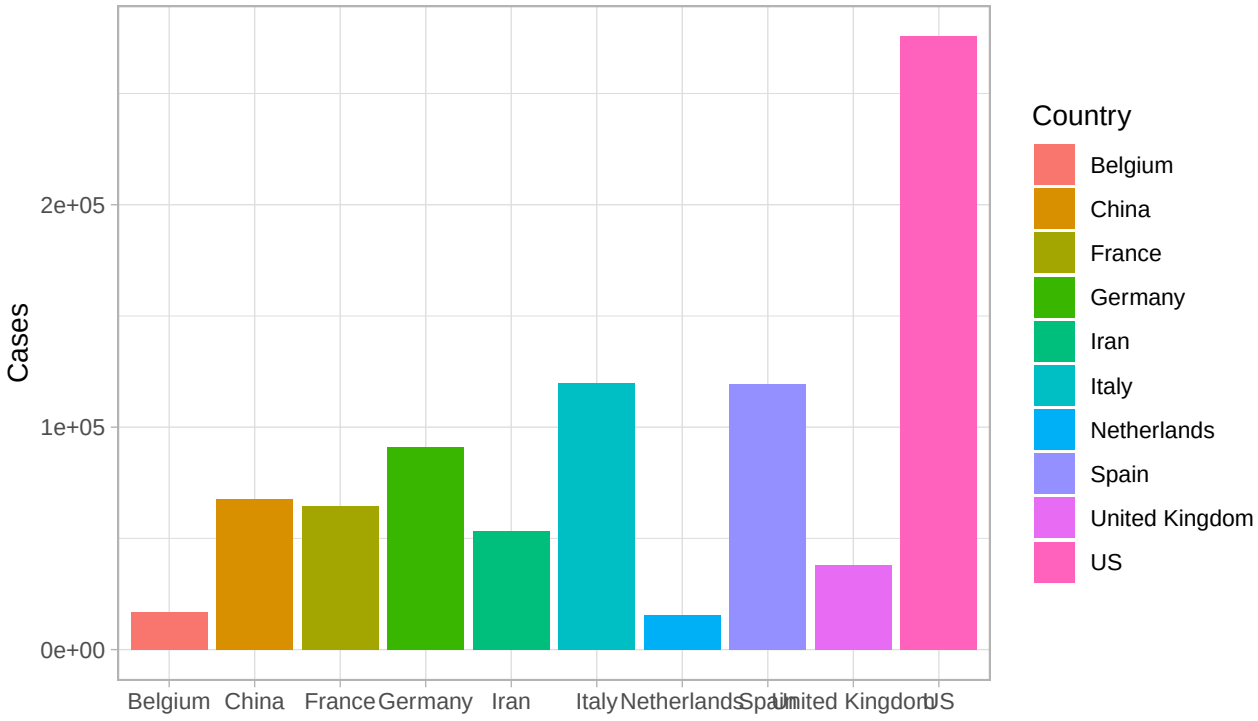
```
## Selecting by Deaths
```

Key	Cases.Per.100K	Deaths
New York City, New York, US	3509	1584
Rockland, New York, US	1316	42
Westchester, New York, US	1277	67
Orleans, Louisiana, US	891	148
Nassau, New York, US	886	95
Suffolk, New York, US	688	84
St. John the Baptist, Louisiana, US	677	17
Dougherty, Georgia, US	637	30
Orange, New York, US	623	30
Jefferson, Louisiana, US	577	85
Bergen, New Jersey, US	522	132
Union, New Jersey, US	447	45
Passaic, New Jersey, US	442	34
Hudson, New Jersey, US	422	59
Essex, New Jersey, US	384	118
Wayne, Michigan, US	348	223
Fairfield, Connecticut, US	288	75
Monmouth, New Jersey, US	282	48
Ocean, New Jersey, US	278	45
Suffolk, Massachusetts, US	272	21
Morris, New Jersey, US	264	43
Middlesex, New Jersey, US	258	56
Oakland, Michigan, US	202	136
Somerset, New Jersey, US	195	23
Macomb, Michigan, US	178	65

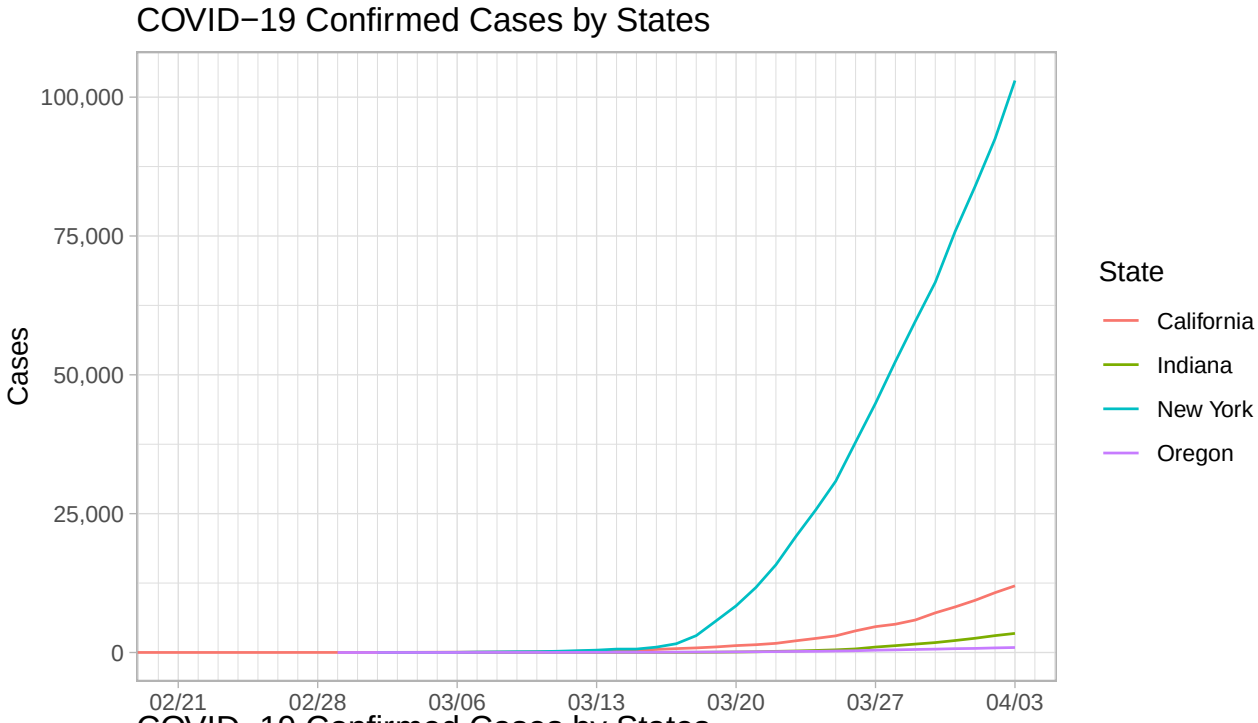
Key	Cases.Per.100K	Deaths
Snohomish, Washington, US	167	42
Essex, Massachusetts, US	157	23
Marion, Indiana, US	148	33
Norfolk, Massachusetts, US	148	24
Hampden, Massachusetts, US	142	27
Middlesex, Massachusetts, US	137	38
King, Washington, US	124	186
Miami-Dade, Florida, US	124	25
Cook, Illinois, US	119	141
Milwaukee, Wisconsin, US	108	24
New Haven, Connecticut, US	104	18
Weld, Colorado, US	101	16
Fulton, Georgia, US	83	23
Broward, Florida, US	82	22
Erie, New York, US	80	19
Hartford, Connecticut, US	76	18
Palm Beach, Florida, US	57	29
Clark, Nevada, US	56	39
DuPage, Illinois, US	54	16
Santa Clara, California, US	53	36
Cobb, Georgia, US	50	18
El Paso, Colorado, US	47	16
Los Angeles, California, US	45	89
Dallas, Texas, US	35	17
San Diego, California, US	29	16

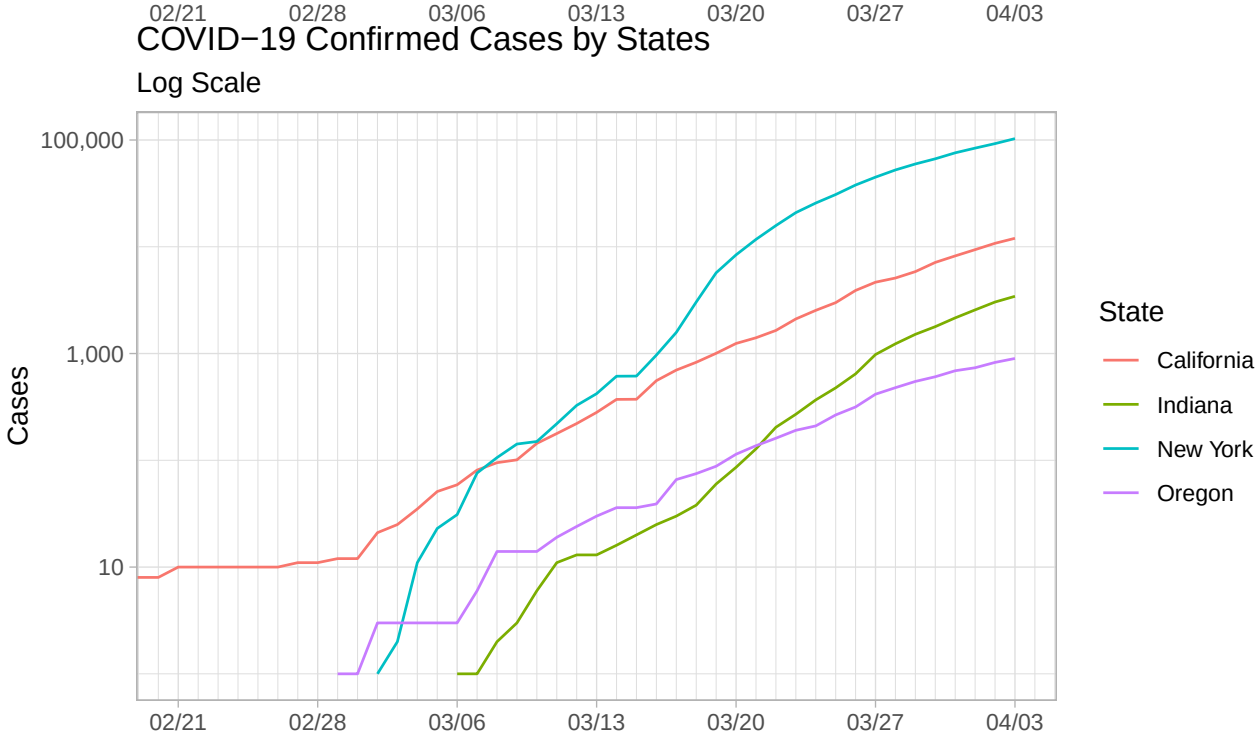
Top Countries infected

```
## Selecting by Deaths
```

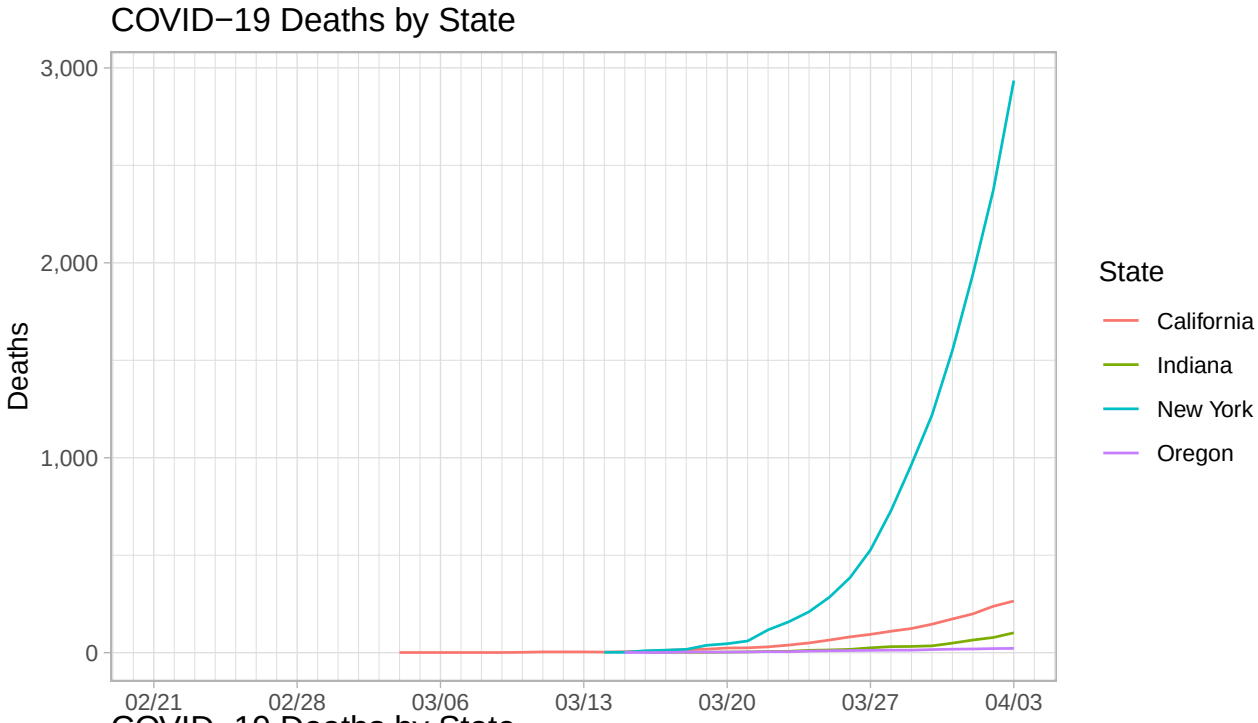


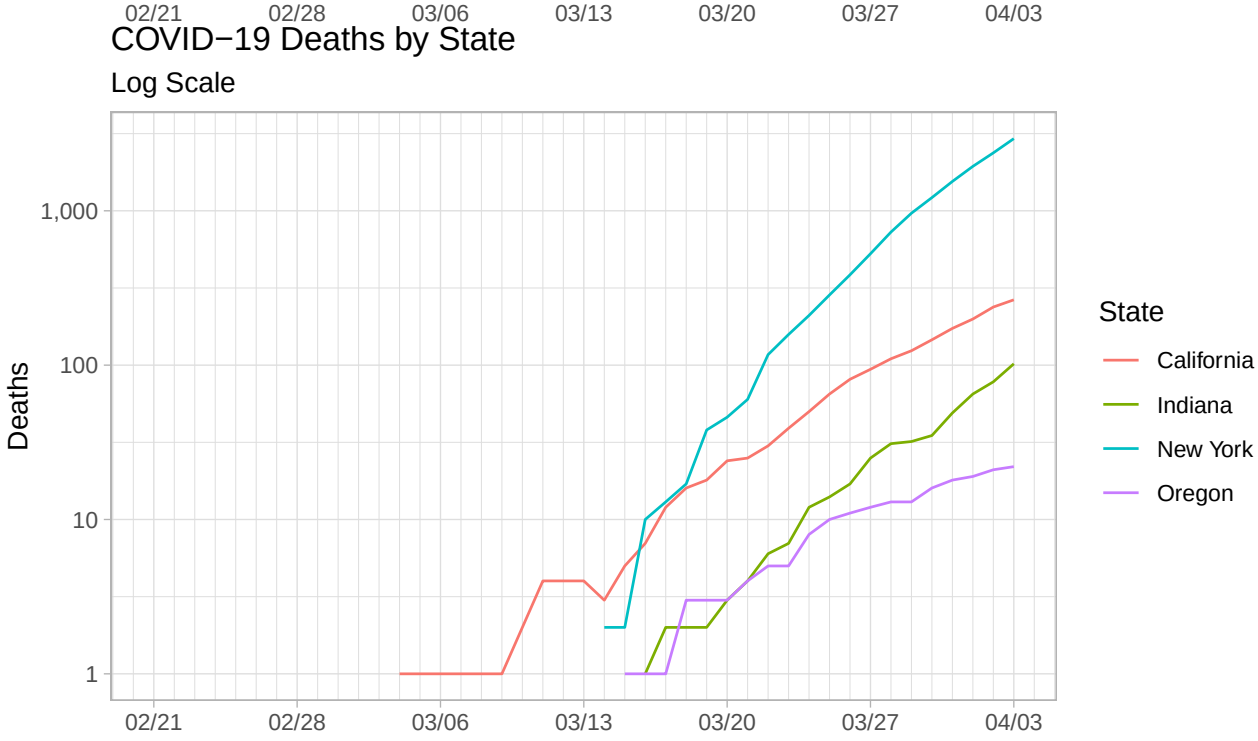
Confirmed Cases Grouped by State





Deaths, grouped by State.



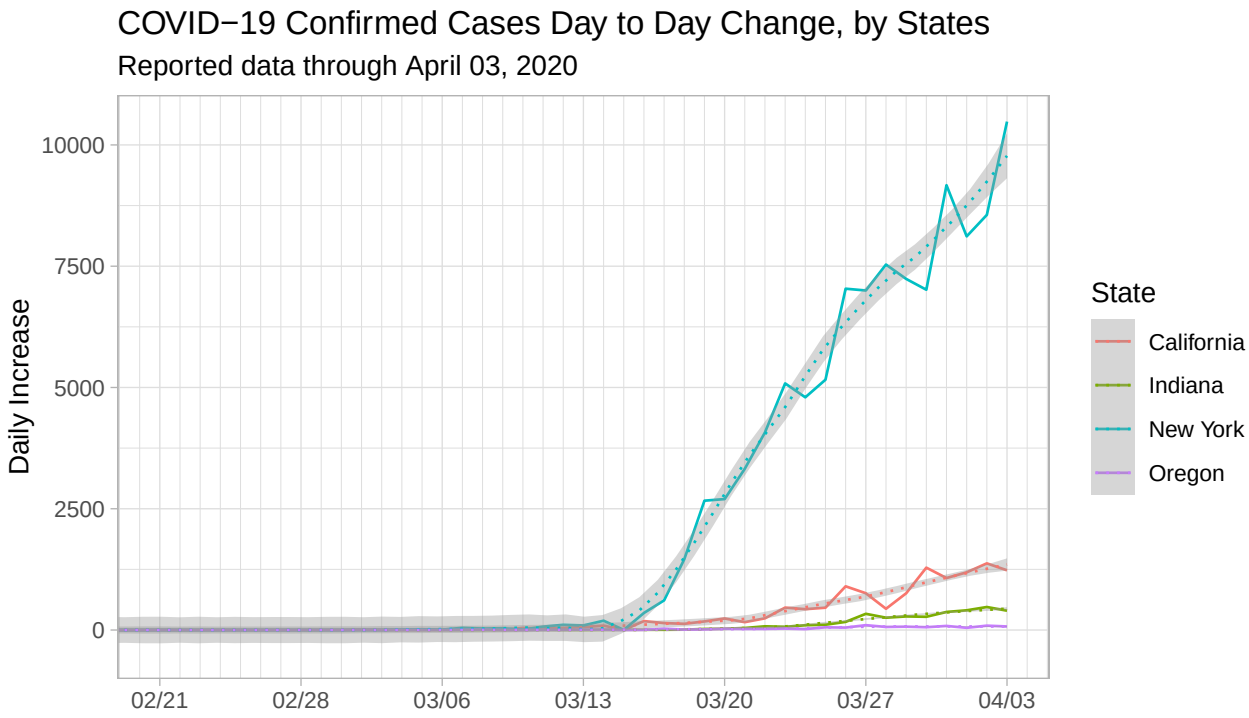


Change in Confirmed Cases

This plot shows the day to day change in confirmed cases. This is important because it will tell us when we hit that peak in infections, the top of the curve that we are trying to 'flatten'. This peak occurs when the line crosses over the X axis into negative territory.

Right now you can see New York is a long way from that but the LOESS trend line shows it starting to trend downward.

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

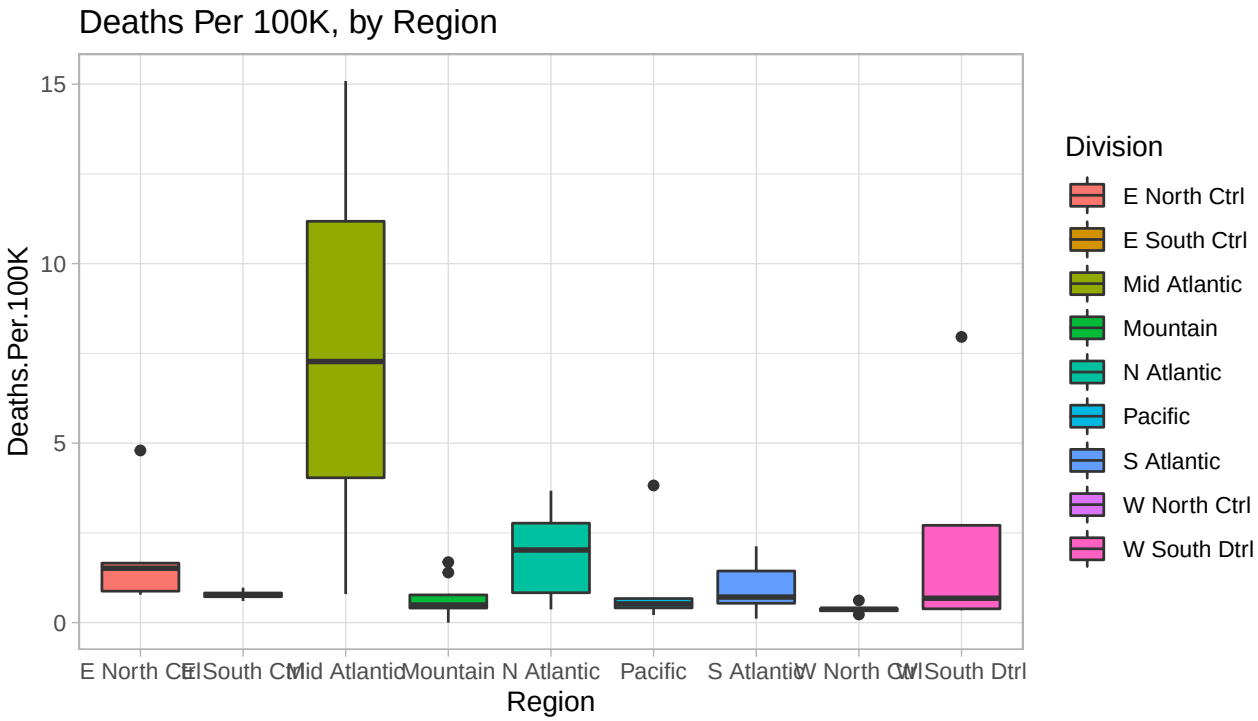


Infections Per Capita

Let's chart the current confirmed cases and deaths as a percentage of population.

```
## # A tibble: 51 x 3
##    State                Infections.Per.100_000 Deaths.Per.100_000
##    <fct>                                 <dbl>              <dbl>
##  1 New York                                529               15.1
##  2 New Jersey                              337                7.3
##  3 Louisiana                               221                8
##  4 Massachusetts                           151                2.8
##  5 Connecticut                             138                3.7
##  6 Michigan                                128                4.8
##  7 District of Columbia                    107                2.1
##  8 Washington                               90                3.8
##  9 Illinois                                 70                1.7
## 10 Pennsylvania                             67                0.8
## # ... with 41 more rows
```

Infection Rate by Region



Save the Data