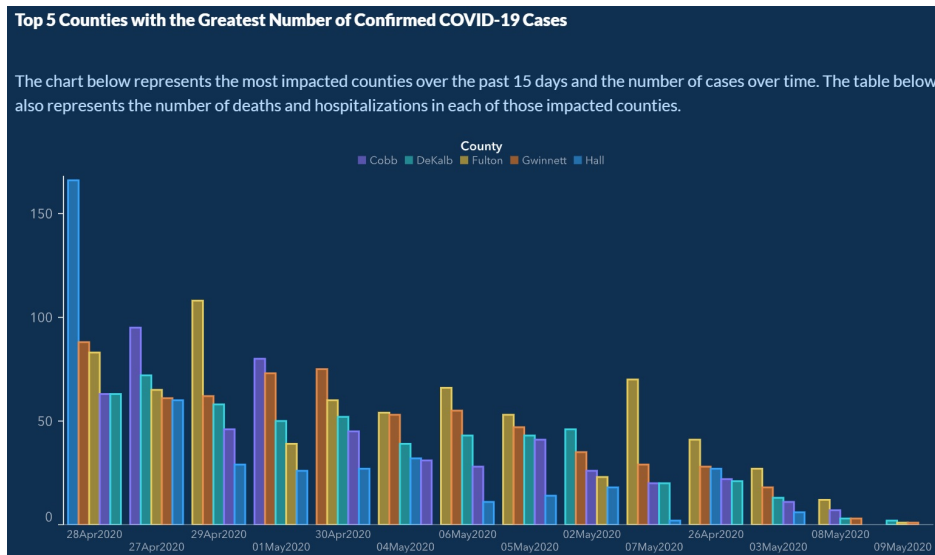


BST 270 Individual Project

In May 2020, the Georgia Department of Public Health posted the following plot to illustrate the number of confirmed COVID-19 cases in their hardest-hit counties over a two-week period. Health officials claimed that the plot provided evidence that COVID-19 cases were decreasing and made the argument for reopening the state.



The plot was heavily criticized by the statistical community and several media outlets for its deceptive portrayal of COVID-19 trends in Georgia. Whether the end result was due to malicious intent or simply poor judgment, it is incredibly irresponsible to publish data visualizations that obscure and distort the truth.

Data visualization is an incredibly powerful tool that can affect health policy decisions. Ensuring they are easy to interpret, and more importantly, showcase accurate insights from data is paramount for scientific transparency and the health of individuals. For this assignment you are tasked with reproducing COVID-19 visualizations and tables published by the [New York Times](#). Specifically, you will attempt to reproduce the following for January 12th, 2022:

1. New cases as a function of time with a rolling average plot - the first plot on the page (you don't need to recreate the colors or theme)
2. Table of cases, hospitalizations and deaths - the first table on the page
3. The county-level map for previous week ('Hot spots') - the second plot on the page (only the 'Hot Spots' plot)
4. Table of cases by state - the second table on the page (do not need to include per 100,000, 14-day change, or fully vaccinated columns)

Data for cases and deaths can be downloaded from this [NYT GitHub repository](#) (use `us-counties.csv`). Data for hospitalizations can be downloaded from [The COVID Tracking Project](#). The project must be submitted in the form of a Jupyter notebook or RMarkdown file and corresponding compiled/knitted PDF, with commented code and text interspersed, including a **brief critique of the reproducibility of each plot and table**. All project documents must be uploaded to a GitHub repository each student will create within the [reproducible data science organization](#). The repository must also include a README file describing the contents of the repository and how to reproduce all results. You should keep in mind the file and folder

structure we covered in class and make the reproducible process as automated as possible.

Tips:

- In R, you can extract the number of new cases from the case totals using the `lag` function. In this toy example, `cases` records the daily total/cumulative number of cases over a two-week period. By default, the `lag` function simply shifts the vector of cases back by one. The number of new cases on each day is then the difference between `cases` and `lag(cases)`.

```
cases = c(13, 15, 18, 22, 29, 39, 59, 61, 62, 67, 74, 89, 108, 122)
new_cases = cases - lag(cases)
new_cases
```

```
## [1] NA  2  3  4  7 10 20  2  1  5  7 15 19 14
```

- You can write your own function to calculate a seven-day rolling average, but the `zoo` package already provides the `rollmean` function. Below, the `k = 7` argument tells the function to use a rolling window of seven entries. `fill = NA` tells `rollmean` to return NA for days where the seven-day rolling average can't be calculated (e.g. on the first day, there are no days that come before, so the sliding window can't cover seven days). That way, `new_cases_7dayavg` will be the same length as `cases` and `new_cases`, which would come in handy if they all belonged to the same data frame.

```
library(zoo)
```

```
new_cases_7dayavg = rollmean(new_cases, k = 7, fill = NA)
new_cases_7dayavg
```

```
## [1]      NA      NA      NA      NA 6.857143 6.714286 7.000000 7.428571
## [9] 8.571429 9.857143 9.000000      NA      NA      NA
```

Tasks

Task #1 Create the new cases as a function of time with a rolling average plot - the first plot on the page (you don't need to recreate the colors or theme).

```
# read in data
```

```
setwd("/Users/raphaelkim/Downloads/270/src")
data=read.csv("../data/us-counties.csv")
allStates=read.csv("../data/all-states-history.csv")
```

```
# aggregate data by dates, summing up the cases along date axes
caseData=aggregate(data$cases, by=list(date=data$date), FUN=sum)
cases=caseData$x
```

```
# calculate the lag and 7 day average
```

```
new_cases = cases - lag(cases)
new_cases_7dayavg = rollmean(new_cases, k = 7, fill = NA)
```

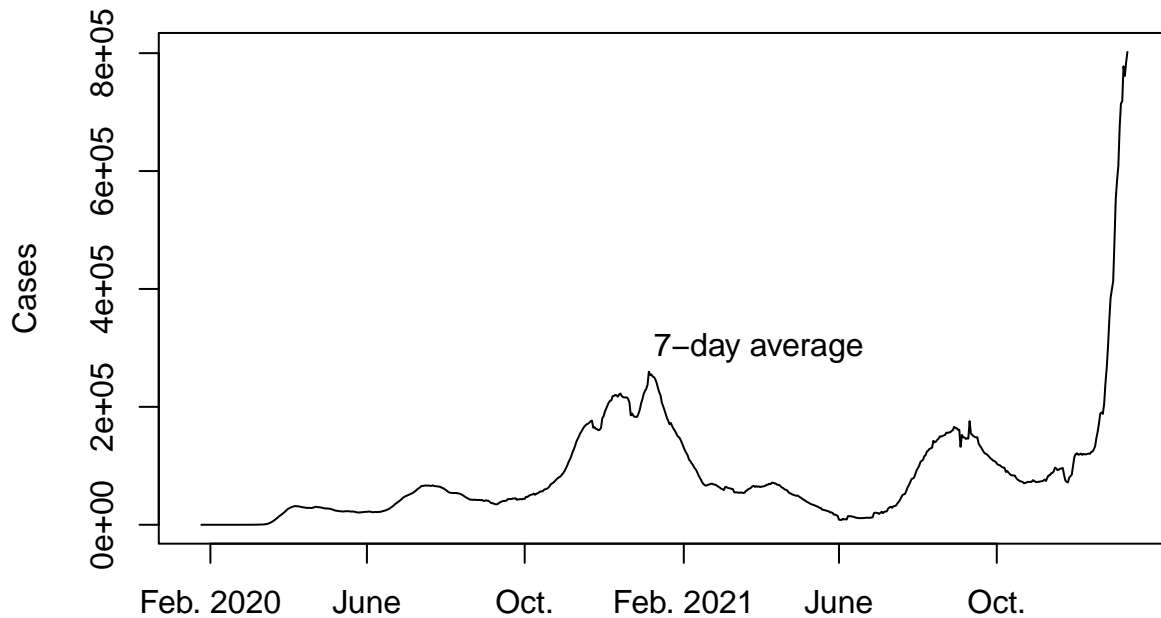
```
# plot it in the manner similar to below, with specific formatting in presentation
plot(new_cases_7dayavg, type='l', xaxt='n', xlab='', ylab='')
```

```
x=seq(1,724)
y=new_cases_7dayavg
```

```
labelIndices=c(which(caseData$date=="2020-02-01"), which(caseData$date=="2020-06-01"), which(caseData$date=="2020-10-01"),
               which(caseData$date=="2021-02-01"), which(caseData$date=="2021-06-01"), which(caseData$date=="2021-10-01"))
axis(1, at=labelIndices, labels=c("Feb. 2020", "June", "Oct.", "Feb. 2021", "June", "Oct."))
title(main="New Reported Cases All Time",
      xlab="", ylab="Cases")
```

```
text(x=340, y=c(259616.1+40000), pos=4, labels=c('7-day average'))
```

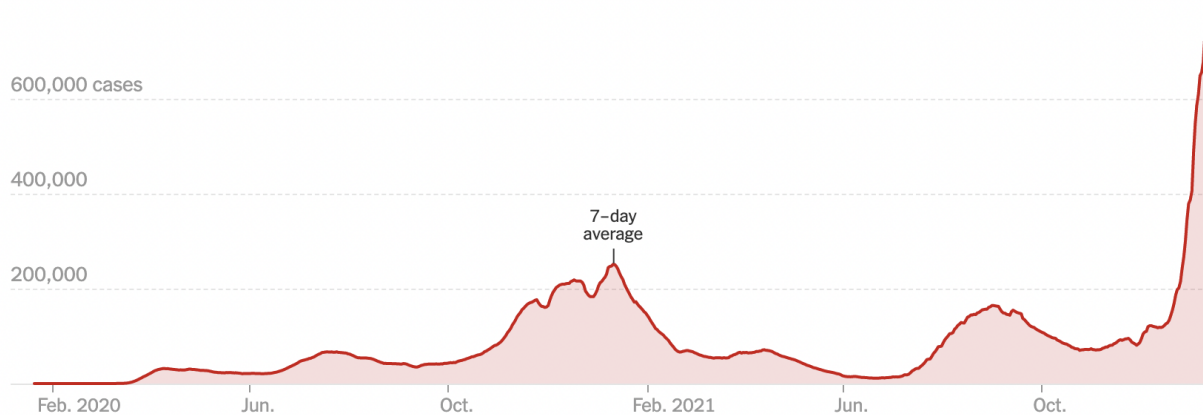
New Reported Cases All Time



New reported cases

All time

Last 90 days



Task #2 Create the table of cases, hospitalizations and deaths - the first table on the page, right below the figure you created in task #1. You don't need to include tests.

Cases first

```
# group by date, and calculate the 7 day averages as above
caseData <- data %>% group_by(date) %>% summarise(new_cases = sum(cases))
caseData$new_cases = caseData$new_cases - lag(caseData$new_cases)
caseData$new_cases = rollmean(caseData$new_cases, k = 7, fill = NA)
```

Table 1: Task 2

	Daily Avg. on Jan. 12	14-Day Change
Cases	802196.7	83.710990
Deaths	1689.0	-2.763385

```

dateStr="2022-01-12" # date of interest

# index corresponding to
offset=2
cases=c(
  caseData$new_cases[which(caseData$date==dateStr)-offset]
  , (caseData$new_cases[which(caseData$date==dateStr)-offset]-caseData$new_cases [(which(caseData$date==dateStr)-offset-14)])
)

cases

## [1] 802196.71429      83.71099

Death

# group by date, and calculate the 7 day averages as above
data2=data %>% drop_na(deaths)
deathData=aggregate(data2$deaths, by=list(date=data2$date), FUN=sum)
deaths=deathData$x
new_deaths = deaths - lag(deaths)

# similar to above but for deaths
deaths=c(
  new_deaths[which(caseData$date==dateStr)-offset],
  (new_deaths[which(deathData$date==dateStr)-offset]-new_deaths[(which(deathData$date==dateStr)-offset-14)])
)

deaths

## [1] 1689.000000    -2.763385

# automate the data presentation into a table.
t2=matrix(NA, nrow=2, ncol=2)
t2[1,]=cases
t2[2,]=deaths
t2=data.frame(t2)
rownames(t2)=c("Cases", "Deaths")
colnames(t2)=c("Daily Avg. on Jan. 12", "14-Day Change")

kable(t2, caption="Task 2")

```

	DAILY AVG. ON JAN. 12	14-DAY CHANGE
Cases	781,203	+159%
Tests	1,992,421	+43%
Hospitalized	145,005	+82%
Deaths	1,827	+51%

Task #3 Create the county-level map for previous week ('Hot spots') - the second plot on the page (only the 'Hot Spots' plot). You don't need to include state names and can use a different color palette.

```
#get past week data, which are the following dates
lastWeek=c("2021-01-12", "2021-01-11", "2021-01-10", "2021-01-09", "2021-01-08", "2021-01-07", "2021-01-06")
dataLastWeek=data[data$date %in% lastWeek,]

# similarly group them by county and compute 7 day averages.
countyData <- data %>% group_by(county) %>% summarise(caseCount = sum(cases))
countyData$new_cases = countyData$caseCount - lag(countyData$caseCount)
countyData$new_cases=rollmean(countyData$new_cases, 7, fill=NA)

# get correponding fips to map by that using usmap library
fipData=c()
for (i in (1:nrow(countyData)))
{
  county=countyData$county[i]
  fipData=c(fipData, data[data$county==county,]$fips[1])
}

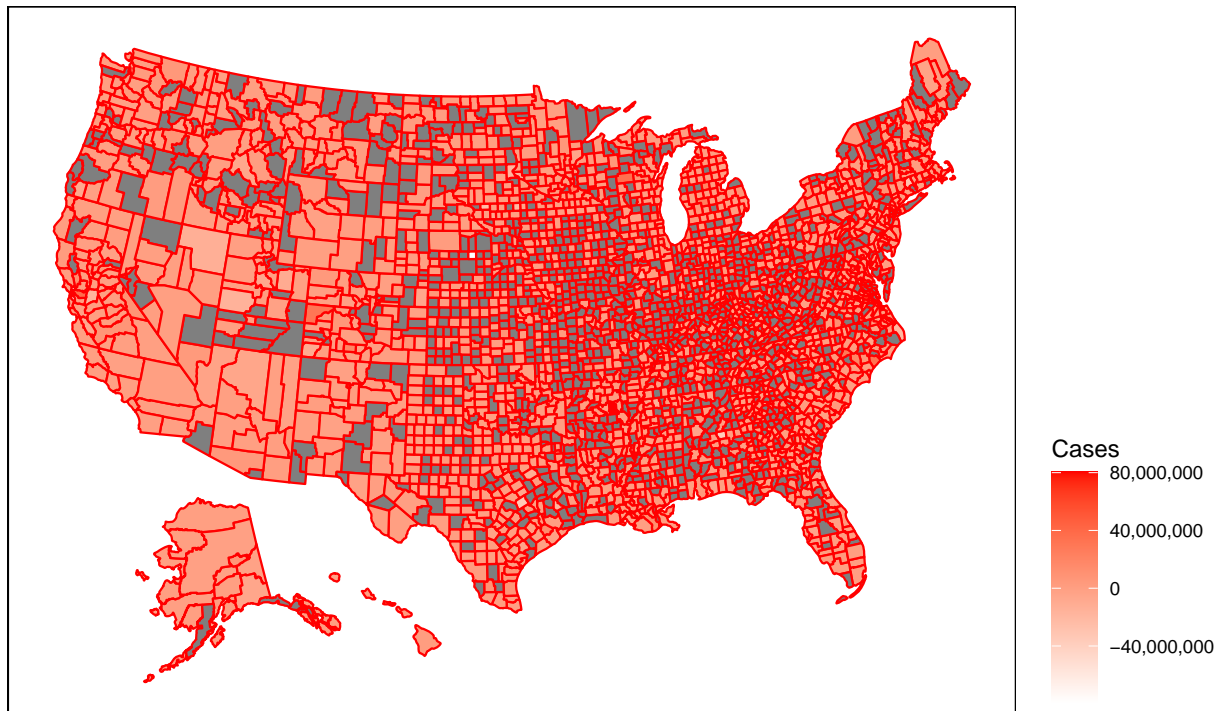
countyData$fips=fipData

# regroup by the identified fips
countyData = countyData %>% group_by(fips) %>% summarise(caseCount = new_cases)

## `summarise()` has grouped output by 'fips'. You can override using the `.groups` argument.

# plot based on fips
library(usmap)
plot_usmap(data = countyData, values = "caseCount", color = "red", labels=FALSE) +
  scale_fill_continuous( low = "white", high = "red",
                        name = "Cases", label = scales::comma
  ) +
  theme(legend.position = "right") +
  theme(panel.background = element_rect(colour = "black")) +
  labs(title = "Hot spots of Coronavirus - Daily Cases in Past Week")
```

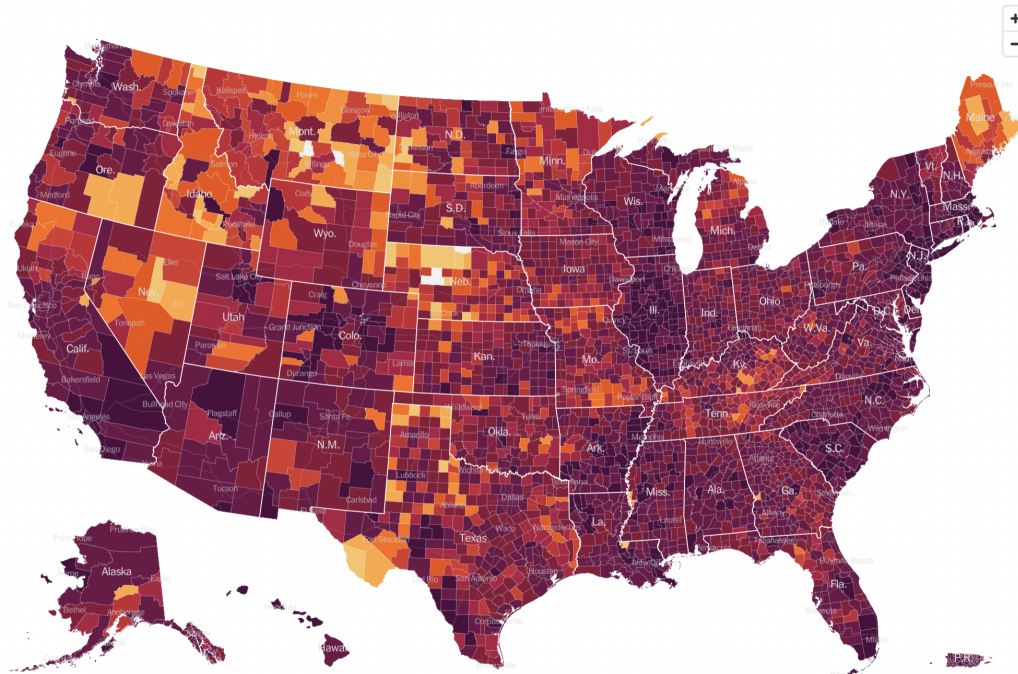
Hot spots of Coronavirus – Daily Cases in Past Week



Hot spots

AVERAGE DAILY CASES PER 100,000 PEOPLE IN PAST WEEK

10 30 50 70 100 250 FEW OR NO CASES



Task #4 Create the table of cases by state - the second table on the page (do not need to include per 100,000, 14-day change, or fully vaccinated columns).

```

# group by state and day
stateData = data %>%
  group_by(state, date) %>%
  summarize(caseCount = sum(cases))

## `summarise()` has grouped output by 'state'. You can override using the `.groups` argument.

# get lagged column to compute the differences
stateData$lagged=lag(stateData$caseCount)

stateData2=stateData %>%
  group_by(state, date) %>%
  summarize(ld=caseCount-lagged )

## `summarise()` has grouped output by 'state'. You can override using the `.groups` argument.

# now go through each state and compute the 7 day average for each state on 2021-01-12
allStatesNames=unique(stateData2$state)
vals=c()
date="2021-01-12"
states=c()
for (state in allStatesNames)
{
  stateDat=stateData2[stateData2$state==state,]
  stateDat$avgs=rollmean(stateDat$ld, 7, fill=NA)

  if (state=="American Samoa")
    vals=c(vals, 0)
  else
    vals=c(vals, stateDat$avgs[stateDat$date==date])
  #allStatesNames[1]
}












finalDat=data.frame(allStatesNames, vals)

colnames(finalDat)=c("state", "Cases Daily Average")
finalDat=rbind(c("United States", cases[1]), finalDat) # add on the US metric, computed above

# show in final df
kable(finalDat)

```

state	Cases Daily Average
United States	802196.714285714
Alabama	3320.14285714286
Alaska	256.714285714286
American Samoa	0
Arizona	8635.85714285714
Arkansas	2682.14285714286
California	42519.8571428571
Colorado	2311.71428571429
Connecticut	2489.71428571429
Delaware	746.285714285714
District of Columbia	290.428571428571
Florida	14116.4285714286
Georgia	9100
Guam	10.7142857142857
Hawaii	162.571428571429
Idaho	886.714285714286
Illinois	6015.71428571429
Indiana	4370.42857142857
Iowa	1309.28571428571
Kansas	1968.71428571429
Kentucky	3575.57142857143
Louisiana	3346
Maine	624.857142857143
Maryland	3020.28571428571
Massachusetts	5743.71428571429
Michigan	2939.71428571429
Minnesota	1659.71428571429
Mississippi	2074.85714285714
Missouri	3567.71428571429
Montana	436.714285714286
Nebraska	891.428571428571
Nevada	2047.85714285714
New Hampshire	764
New Jersey	6330.71428571429
New Mexico	1192.71428571429
New York	15925.2857142857
North Carolina	7788.57142857143
North Dakota	167.857142857143
Northern Mariana Islands	0.428571428571429
Ohio	7405.57142857143
Oklahoma	3922.57142857143
Oregon	1207
Pennsylvania	7331
Puerto Rico	555.285714285714
Rhode Island	975.571428571429
South Carolina	4541.57142857143
South Dakota	336.714285714286
Tennessee	5101.42857142857
Texas	22871.7142857143
Utah	2723.57142857143
Vermont	159.285714285714
Virgin Islands	17.7142857142857
Virginia	4959.57142857143
Washington	2449.42857142857
West Virginia	1250.14285714286
Wisconsin	2749.71428571429
Wyoming	323.142857142857

	CASES DAILY AVG.	PER 100,000	14-DAY CHANGE	HOSPITALIZED DAILY AVG.	PER 100,000	14-DAY CHANGE	DEATHS DAILY AVG.	PER 100,000	FULLY VACCINATED
United States	781,203	235	+159% 	145,005	44	+82%	1,827.2	0.55	63%
Rhode Island ›	5,349	505	+222% 	479	45	+64%	6.3	0.59	78%
New York ›	70,655	363	+69% 	12,933	66	+96%	164.7	0.85	73%
Massachusetts ›	23,793	345	+173% 	2,837	41	+103%	53.0	0.77	75%
New Jersey ›	29,097	328	+67% 	6,180	70	+105%	75.4	0.85	71%
Delaware ›	3,004	308	+184% 	745	76	+72%	13.0	1.34	65%
Florida ›	65,551	305	+116% 	10,526	49	+241%	39.6	0.18	64%
U.S. Virgin Islands ›	320	301	+221% 	22	21	+467%	0.1	0.13	51%
Vermont ›	1,746	280	+276% 	107	17	+82%	1.1	0.18	78%
Utah ›	8,939	279	+468% 	600	19	+43%	11.6	0.36	59%
Hawaii ›	3,868	273	+151% 	299	21	+179%	1.5	0.10	65%

Task #5 Provide a brief critique of the reproducibility of the figures and tables you created in tasks 1-4.

Task 1 was able to be reproduced easily, but I was not able to reproduce the rest of the tasks as easily. Later we learned the data used is on github, but it was somewhat challenging to follow the definitions and easily reproduce their numbers generated. Because of that, I would not rate the reproducibility very high since I was not able to recompute the 7 day averages easily from first principles.