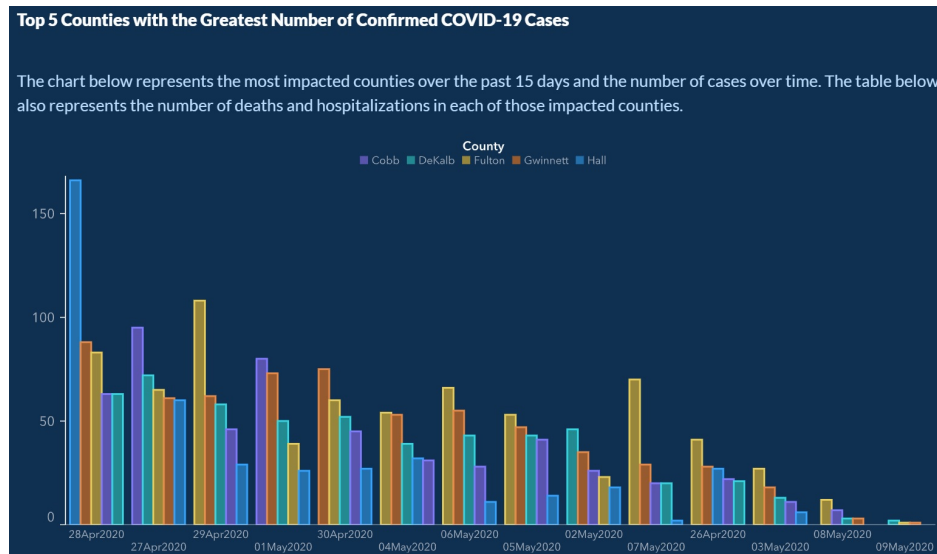


BST 270 Individual Project

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.

Critique: The dates on the x-axis were placed in the wrong order to show the descending trend in this plot. It is totally nonsense for a time-based plot.

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 17th, 2021:

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 or per capita 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:

- 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
```

- 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.

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

```
# read data for 2020 and 2021
```

```
c2020 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties-2020.csv")
```

```
## Rows: 884737 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr (3): county, state, fips
## dbl (2): cases, deaths
## date (1): date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
c2021 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties-2021.csv")
```

```
## Rows: 1185373 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr (3): county, state, fips
## dbl (2): cases, deaths
## date (1): date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
# merge datasets
counties <- rbind(c2020,c2021)
summary(counties)
```

```
##      date      county      state      fips
## Min.   :2020-01-21 Length:2070110 Length:2070110 Length:2070110
## 1st Qu.:2020-09-09 Class :character Class :character Class :character
## Median :2021-02-16 Mode  :character Mode  :character Mode  :character
## Mean   :2021-02-15
## 3rd Qu.:2021-07-25
## Max.   :2021-12-31
##
##      cases      deaths
## Min.   :      0 Min.   :      0.0
## 1st Qu.:    257 1st Qu.:      4.0
## Median :   1292 Median :     25.0
## Mean   :   7225 Mean   :    133.7
## 3rd Qu.:   4303 3rd Qu.:     80.0
## Max.   : 1697286 Max.   : 35382.0
##              NA's   :47231
```

```
# subset the data before Jan 18 2021
counties <- counties[counties$date <= "2021-01-23",]
summary(counties$date)
```

```
##      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
## "2020-01-21" "2020-06-14" "2020-08-29" "2020-08-27" "2020-11-11" "2021-01-23"
```

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)

```
# subset only date, cases, and deaths for plot1
subset <- counties[, c("date", "cases", "deaths")]

# detected NA
subset[is.na(subset)] <- 0

# combine cases and deaths for same date
subset <- subset %>%
  group_by(date) %>%
  summarise(sum_cases = sum(cases), sum_deaths = sum (deaths)) %>%
  mutate(days = date - first(date) + 1)
summary(subset)
```

```
##      date      sum_cases      sum_deaths      days
## Min.   :2020-01-21 Min.   :      1 Min.   :      0 Length:369
## 1st Qu.:2020-04-22 1st Qu.:  839336 1st Qu.:  47059 Class :difftime
## Median :2020-07-23 Median :  4050036 Median :144283 Mode  :numeric
## Mean   :2020-07-23 Mean   :  6101664 Mean   :149551
## 3rd Qu.:2020-10-23 3rd Qu.:  8565062 3rd Qu.:223953
## Max.   :2021-01-23 Max.   : 25050385 Max.   :417404
```

```

# create new_cases and
subset <- subset %>% mutate(new_cases = sum_cases - lag(sum_cases))

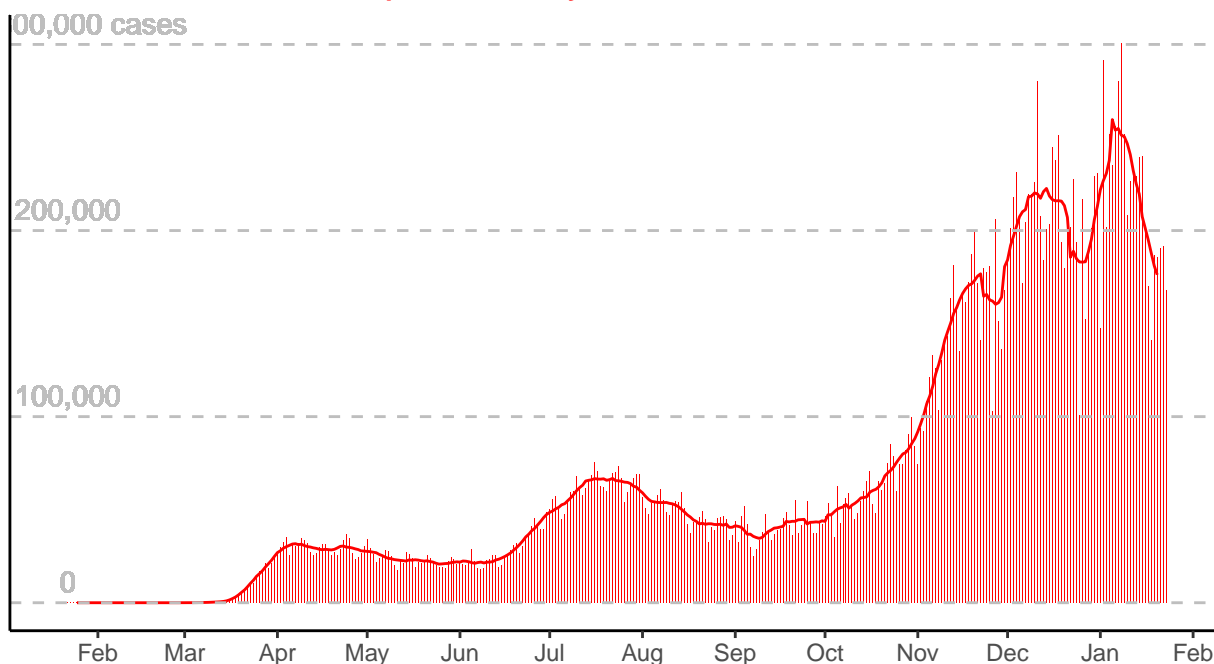
# rolling average for 7 days
subset$new_cases_7dayavg = rollmean(subset$new_cases, k = 7, fill = NA)

# Plot 1
ggplot(subset) +
  # bar plot for new cases
  geom_bar(aes(x=date, y=new_cases), stat="identity", fill="red",width=0.1,na.rm = TRUE, ) +
  # line plot for 7 days average
  geom_line(aes(x=date, y=new_cases_7dayavg), stat="identity", color="red",na.rm = TRUE) +
  labs(title = "Coronavirus in the U.S.: \n Lastest Map and Case Count",
        subtitle = "Updated January 18,2021,7:56 A.M. E.T.") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_classic() +
  # horizontal lines
  geom_hline(yintercept=0, linetype="dashed", color = "grey")+
  geom_hline(yintercept=100000, linetype="dashed", color = "grey")+
  geom_hline(yintercept=200000, linetype="dashed", color = "grey")+
  geom_hline(yintercept=300000, linetype="dashed", color = "grey")+
  geom_text(aes(date[2], 0, label = "0"),vjust= -0.5, color ="grey")+
  geom_text(aes(date[2], 100000, label = "100,000"),vjust= -0.5, color ="grey")+ geom_text(aes(date[2], 200000, label = "200,000"),vjust= -0.5, color ="grey")+
  geom_text(aes(date[10], 300000, label = "300,000 cases"),vjust= -0.5, color ="grey") +
  # theme adjustment
  theme(axis.text.y = element_blank(), axis.ticks.y=element_blank(),
        axis.title.x=element_blank(), axis.title.y=element_blank()) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  theme(plot.title = element_text(color="black", size=24, face="bold"))+
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(color="red", size=10, face="bold"))+
  theme(plot.subtitle = element_text(hjust = 0.5))

```

Coronavirus in the U.S.: Lastest Map and Case Count

Updated January 18, 2021, 7:56 A.M. E.T.



Critique: Plot 1 was reproduced successfully both in numbers and plot patterns. The key factor here is the method for calculating 7 day average in the lag function. The raw data was cleaned and cleared for case numbers.

2. Table of cases, hospitalizations and deaths - the first table on the page

Coronavirus in the U.S.: Latest Map and Case Count

Updated January 18, 2021, 7:56 A.M. E.T.

[Leer en español](#)



	TOTAL REPORTED	ON JAN. 17	14-DAY CHANGE
Cases	23.9 million+	169,641	+3% →
Deaths	397,612	1,730	+26% →
Hospitalized		124,387	+3% →

■ Day with reporting anomaly. Hospitalization data from the Covid Tracking Project; 14-day change trends use 7-day averages.

```
# new death and 7daysavg death
subset <- subset %>% mutate(new_deaths = sum_deaths - lag(sum_deaths))
subset$new_deaths_7dayavg = rollmean(subset$new_deaths, k = 7, fill = NA)

# table 1
totalrep_case <- subset$sum_cases[363]
# nuance from the picture because of the updated data from that time
jan17_case <- subset$new_cases[363]
totalrep_death <- subset$sum_deaths[363]
jan17_death <- subset$new_deaths[363]
change14_case <- (subset$new_cases_7dayavg[363]-subset$new_cases_7dayavg[350])/subset$new_cases_7dayavg[350]
change14_death <- (subset$new_deaths_7dayavg[363]-subset$new_deaths_7dayavg[350])/subset$new_deaths_7dayavg[350]

#create table
tab = matrix(c(totalrep_case,jan17_case,change14_case,totalrep_death,jan17_death,change14_death), ncol=6)
colnames(tab) = c('TOTAL REPORTED', 'ON JAN.17','14-DAY CHANGE')
rownames(tab) = c('Cases','Deaths')
format(tab, scientific = FALSE)
```

```
##          TOTAL REPORTED      ON JAN.17      14-DAY CHANGE
## Cases  "23986856.00000000" " 170094.00000000" "      -0.16428485"
## Deaths " 397624.00000000" "   1730.00000000" "       0.09933707"
```

```
round(tab,0)
```

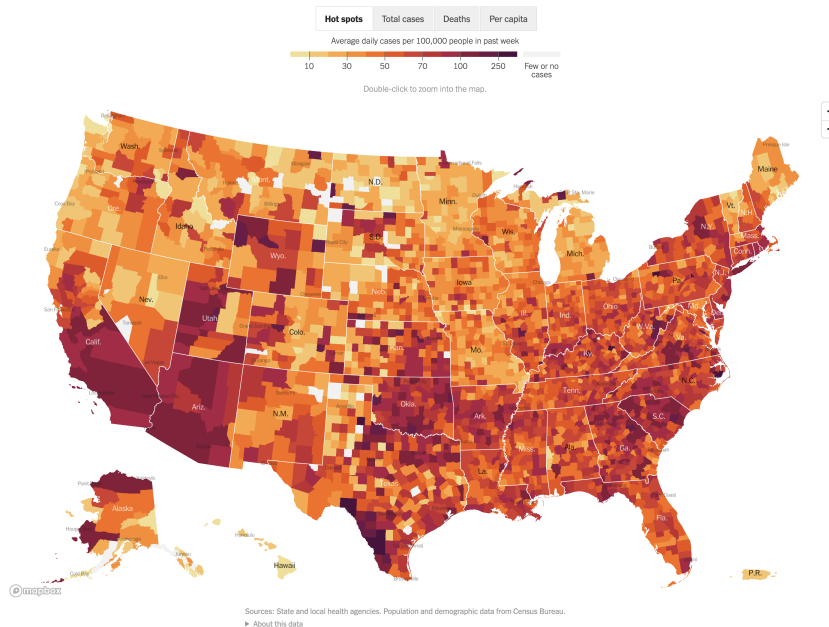
```
##          TOTAL REPORTED ON JAN.17 14-DAY CHANGE
## Cases      23986856      170094      0
## Deaths     397624       1730      0
```

```
tab <- as.data.frame(tab)
print(tab)
```

```
##          TOTAL REPORTED ON JAN.17 14-DAY CHANGE
## Cases      23986856      170094    -0.16428485
## Deaths     397624       1730     0.09933707
```

Critique: Table 1 was reproduced partially successfully. The number of cases and death of “total reported” and “on Jan.17” are the same as the New York Times. The nuance is reasonable because the data I collected was much later than the screenshot shown. Thus, it may include more missed cases which didn’t confirm at that time. However, the 14-day change numbers are wrong. I searched and tried more methods about this topic and still couldn’t match the number. The main reason should be the wrong mathematics calculation methods. This is the bad part New York Times needs to improve they didn’t give a detailed introduction about the methods and software they used, which made it non-reproducible.

3. The county-level map for previous week (‘Hot spots’) - the second plot on the page (only the ‘Hot Spots’ plot)



```
# data for counties
subset2 <- counties %>%
  group_by(fips,date) %>%
  filter(date >= "2021-01-07" & date <= "2021-01-23") %>%
  summarise(sum_cases = sum(cases)) %>%
  mutate(new_cases = sum_cases - lag(sum_cases)) %>% mutate(new_cases_7dayavg = rollmean(new_cases, k = 7
```

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

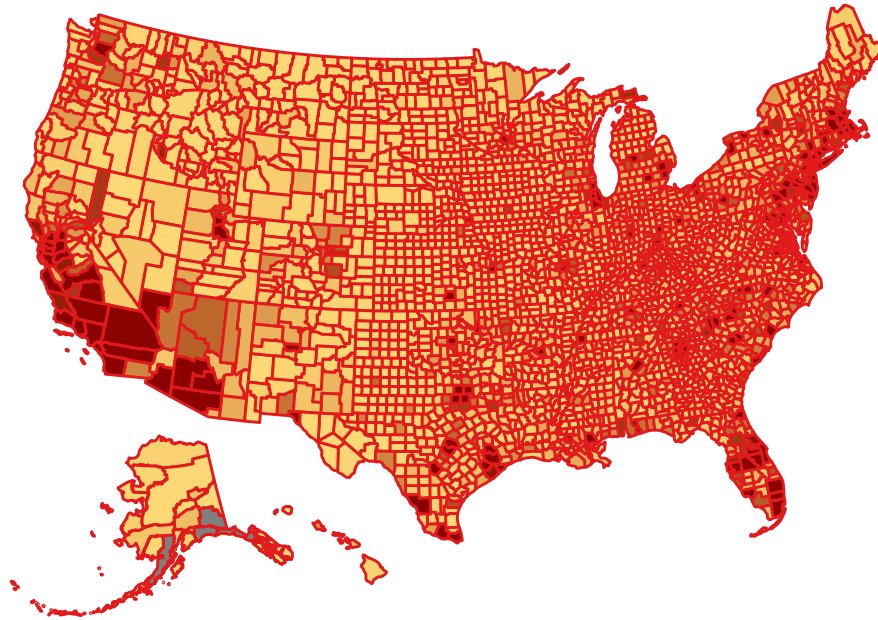
```
# check for distribution
# hist(subset2$new_cases_7dayavg)
# wired data
subset3 <- subset2[subset2$date == "2021-01-17",]
# do some adjustment
subset3$new_cases_7dayavg[subset3$new_cases_7dayavg > 250] <- 250
subset3$new_cases_7dayavg[subset3$new_cases_7dayavg < 0 ] <- 0
subset3 <- na.omit(subset3)

#usmap
library(usmap)
plot_usmap(data=subset3, value="new_cases_7dayavg",color="#E31A1C") +
  scale_fill_continuous(low="#FED976", high="darkred",name = "Average daily cases per 100,000 people in
  labs(title = "Hot Spots") +
  theme(legend.position = "top")
```

Hot Spots

Average daily cases per 100,000 people in past week

0 50 100 150 200 250



```
#tried different colors
#"ggthemes::Red-Gold", 30
#library("RColorBrewer")
#brewer.pal(n = 8, name = "YlOrRd")
## per 100,000 is the population size
```











Critique: As we matched in the last table, the numbers for new cases in 7 days average are

correct. However, this map is not the same as New York Times. The filtered data has wired distribution that both included minus and really big numbers for counties. I tried to do some adjustments to find the maps and methods they may use for selection. But it didn't work. There are a lot of possible reasons. We used R to deal with the data and we don't know what kind of software they used. Because some internal functions and visualization methods are different and will cause differences here. Also, the detailed methods for data cleaning and selection are unknown, which made it impossible to reproduce.

4. Table of cases by state - the second table on the page (do not need to include per 100,000 or per capita columns)

Cases and deaths by state and county

This table is sorted by places with the most cases per 100,000 residents in the last seven days. Charts are colored to reveal when outbreaks emerged.

Cases		Deaths		Search counties			
		TOTAL CASES	PER 100,000	DAILY AVG. IN LAST 7 DAYS	▼ PER 100,000	WEEKLY CASES PER CAPITA	
						FEWER	MORE
+ Arizona	MAP »	673,882	9,258	7,905	109		
+ California	MAP »	3,006,583	7,609	39,580	100		
+ South Carolina	MAP »	388,184	7,539	4,808	93		
+ Rhode Island	MAP »	104,443	9,859	976	92		
+ Oklahoma	MAP »	354,979	8,971	3,374	85		
+ Georgia	MAP »	791,322	7,453	8,457	80		
+ Utah	MAP »	323,837	10,101	2,548	79		
+ Texas	MAP »	2,127,334	7,337	22,782	79		
+ New York	MAP »	1,242,818	6,389	15,281	79		
+ Massachusetts	MAP »	470,140	6,821	5,336	77		

```
# get the sum_cases
```

```
subset4 <- counties %>% filter(date == "2021-01-17") %>%
  group_by(state) %>%
  summarise(sum_cases = sum(cases))
```

```
# get the daily average
```

```
subset5 <- counties %>% group_by(state,date) %>%
  summarise(sum_cases = sum(cases)) %>%
  filter(date >= "2021-01-07" & date <= "2021-01-23") %>%
  mutate(new_cases = sum_cases - lag(sum_cases)) %>% mutate(new_cases_7dayavg = rollmean(new_cases, k = 7)
  filter(date == "2021-01-17") %>%
  subset(select = c(state, new_cases_7dayavg))
```

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

```
# merge to the final subset
subset4 <- merge(subset4, subset5, by="state")

# filter the target states and round
subset6 <- subset4 %>%
  filter(state %in% c("Arizona", "California", "South Carolina", "Rhode Island", "Oklahoma", "Georgia", "Utah"))

colnames(subset6) <- c('', 'TOTAL CASES', 'DAILY AVG. IN LAST 7 DAYS')
# final table
# adjust the order
subset6[c(1,2,8,7,6,3,10,9,5,4),]
```

##		TOTAL CASES	DAILY AVG. IN LAST 7 DAYS
## 1	Arizona	673882	7146
## 2	California	3006966	33250
## 8	South Carolina	388184	4924
## 7	Rhode Island	104443	770
## 6	Oklahoma	354979	2714
## 3	Georgia	791322	7780
## 10	Utah	323837	1938
## 9	Texas	2125391	20904
## 5	New York	1242818	14364
## 4	Massachusetts	470140	4656

Critique: Table 2 was reproduced successfully! Both “total cases” and “daily average in last 7 days” are correct for these states from New York Times. Nuance is caused maybe because more data were confirmed after that time and can be ignored. But I couldn’t figure out why in this order for states. Descending for case numbers is the most natural and understandable way.