BST 270 Individual Project

Evan Goldberg

In this project we attempt to reproduce the 4 following figures/tables.

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)

We use two data files. First we use data from us-counties.csv which can be accessed in the NYT GitHub repository as well as in our repository. This will be our primary source of data for this project. We also use data from The COVID Tracking Project for hospitalization data.

```
us_counties <- read.csv("us-counties.csv")
cases <- read.csv("cases.csv")
```

Figure 1.

Our data is made up of dates and the number of cases that have been reported in a given county up to that date. While we will have a use for that later, we actually need to wrangle our data in order to get new cases for a given day. We will then take these new cases per day in order to calculate a 7-day rolling average. The steps to do this are shown below.

```
#initial data cleaning, drop missing values and group by date so that there is
#one entry of cases per date
cases_per_day <- us_counties[,c("date", "cases")] %>%
  drop_na() %>%
  group_by(date) %>%
  summarise(cases = sum(cases))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#make the date variable readable in R
cases_per_day$date <- as.Date(cases_per_day$date, "%m/%d/%y")

#order data chronologically
cases_per_day <- cases_per_day[order(cases_per_day$date),]

#create a column for new cases by subtracting the previous day's total number
#of cases from the current day's
cases_per_day$new_cases <- cases_per_day$cases - lag(cases_per_day$cases)

#create a rolling 7 day average which is the average number of new cases over
#the previous 7 days
```

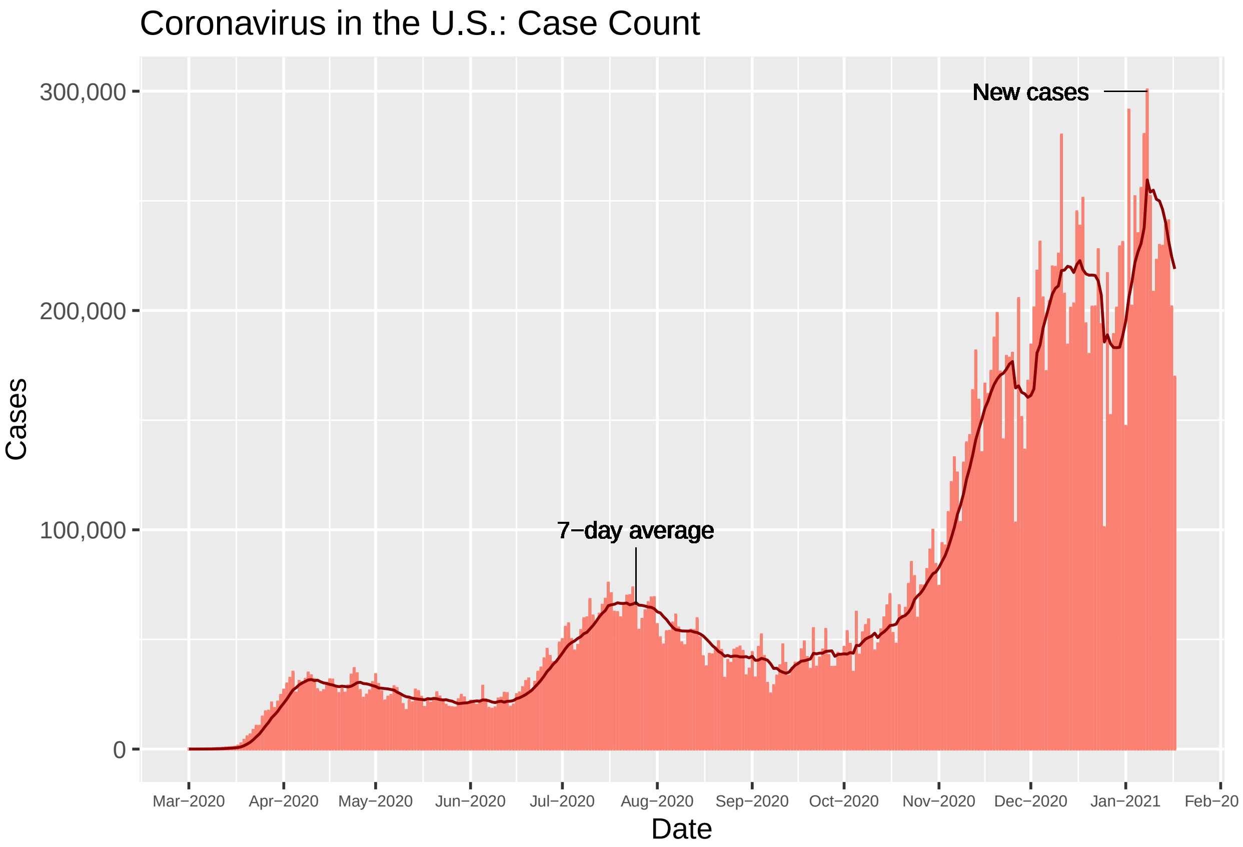
```
cases_per_day$new_cases_7dayavg <- round(rollmean(cases_per_day$new_cases,
                                                  k = 7, fill = NA))
cases_per_day$new_cases_7dayavg <- lag(cases_per_day$new_cases_7dayavg, n = 3)

#drop rows before March 1st 2020 and after January 17th 2021
cases_per_day <- cases_per_day %>% filter(date >= as.Date("2020-03-01") & date <= as.Date("2021-01-17"))

#replace the entry of March 1st to represent the total number of cases up until
#March 1st
cases_per_day$new_cases[1] <- cases_per_day$cases[1]
```

Now that we have performed the necessary data wrangling, we can use ggplot to create an appropriate plot.

```
ggplot(cases_per_day)+
  geom_col(aes(x = date, y= new_cases), #create the columns for new cases
           color = "salmon",
           alpha = 0.75,
           width = 0.01)+
  geom_line(aes(x = date, y= new_cases_7dayavg),
            color = "darkred")+ #creates the line for 7-day average
  scale_x_date(date_breaks = "1 month",
               date_labels = "%b-%Y")+
  scale_y_continuous(breaks = c(0, 100000, 200000, 300000),
                     labels = c("0", "100,000", "200,000", "300,000"))+
  xlab("Date")+
  ylab("Cases")+
  theme(axis.text.x = element_text(size = 6))+
  #create labels to differentiate the columns and line plots
  geom_text(
    label="7-day average",
    x=as.Date("2020-07-25"),
    y=100000,
    size = 3
    )+
  geom_segment(aes(x = as.Date("2020-07-25"),
                   y = 66000,
                   xend = as.Date("2020-07-25"),
                   yend = 92000),
               size = 0.2)+
  theme(axis.text.x = element_text(size = 6))+
  geom_text(
    label="New cases",
    x=as.Date("2020-12-01"),
    y=300000,
    size = 3
    )+
  geom_segment(aes(x = as.Date("2020-12-25"),
                   y = 300000,
                   xend = as.Date("2021-01-08"),
                   yend = 300000),
               size = 0.2)+
  ggtitle("Coronavirus in the U.S.: Case Count")
```



This figure was fairly easy to reproduce and looks very similar to the original plot in the New York Times article, however there were some aspects of the original figure that are not explained very well. Specifically how they derived their data for the first week of 7 day averages is not super clear. They do not explain whether they use data from prior dates that are not shown, or whether they derive the values differently for the first week. Simply using data from the last week of February yields values slightly below their values, however it does not appear to be significantly different, and thus does not drastically change our understanding of the figure. Other than this slight discrepancy, all other values for new cases and 7-day averages match what the paper reports. The variable names are clear and the data is well kept and easy to understand. Overall I would say this figure does have fairly good reproducibility

Table 2.

In order to create this table, we need to perform similar steps to what we did for Figure 1, but this time with death and hospitalization data. Our us-counties.csv file contains death data in the same format as the cases data so it should be very easy to form the same types of manipulations. However, we have to get our hospitalization data from elsewhere, specifically from The COVID Tracking Project. We have already downloaded the data and it can be accessed from the cases.csv in our repository. We perform similar manipulations, however it is important to note that our hospitalization data is in the form of hospitalizations for a given day rather than total cases. This actually saves us a step but it is important to recognize the difference.

In order to create the desired table we want to obtain the following information.

We want the total number of cases and deaths as of January 17th, 2021. We also want the number of new cases, new deaths, and current hospitalizations on January 17th, 2021. Finally, we want the 14-day change percentage for cases, deaths, and hospitalizations. This is calculated using the 7 day average instead of just new cases so it is slightly more complicated, however it is still quite doable. This will tell us the percent change from January 4th, 2021 to January 17th, 2021. The steps to do this are shown below.

```
#repeat process but with deaths instead of cases
deaths_per_day <- us_counties[,c("date", "deaths")] %>%
  drop_na() %>%
  group_by(date) %>%
  summarise(deaths = sum(deaths))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
deaths_per_day$date <- as.Date(deaths_per_day$date, "%m/%d/%y")

deaths_per_day <- deaths_per_day[order(deaths_per_day$date),]

deaths_per_day$new_deaths <- deaths_per_day$deaths - lag(deaths_per_day$deaths)

deaths_per_day$new_deaths_7dayavg <- round(rollmean(deaths_per_day$new_deaths,
                                                 k = 7, fill = NA))
deaths_per_day$new_deaths_7dayavg <- lag(deaths_per_day$new_deaths_7dayavg, n = 3)

deaths_per_day <- deaths_per_day %>% filter(date >= as.Date("2020-03-01") & date <= as.Date("2021-01-17

#repeat process but with hospitalizations
#note that we can skip the step where we generate new cases since we already
#have that information
hospitalized_per_day <- cases[,c("date", "hospitalizedCurrently")] %>%
  drop_na() %>%
  group_by(date) %>%
  summarise(hospitalized = sum(hospitalizedCurrently))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
hospitalized_per_day$date <- as.Date(hospitalized_per_day$date, "%m/%d/%y")

hospitalized_per_day <- hospitalized_per_day[order(hospitalized_per_day$date),]

hospitalized_per_day$new_hospitalized_7dayavg <- round(rollmean(hospitalized_per_day$hospitalized,
                                                               k = 7, fill = NA))
hospitalized_per_day$new_hospitalized_7dayavg <- lag(hospitalized_per_day$new_hospitalized_7dayavg, n =

hospitalized_per_day <- hospitalized_per_day %>% filter(date >= as.Date("2020-03-01") & date <= as.Date
```

We combine our three individual datasets into one for easy accessibility (we call it data__per__day to match the 3 datasets that we are combining). We then hand pick the values in this table that correspond to our desired values for the table. All the data we desire is in the last row of the data set.

```
#merges cases, deaths, and hostpitalized per day into one data set for easier
#use
data_per_day <- merge(cases_per_day,
                      merge(deaths_per_day,
                            hospitalized_per_day,
                            by = "date"),
                      by = "date")

#create a 14-day change column for cases, deaths, and hospitalizations
data_per_day$new_cases_14daychange <- round((data_per_day$new_cases_7dayavg - lag(data_per_day$new_case
data_per_day$new_deaths_14daychange <- round((data_per_day$new_deaths_7dayavg - lag(data_per_day$new_dea

data_per_day$new_hospitalized_14daychange <- round((data_per_day$new_hospitalized_7dayavg - lag(data_per

#create a basic table for cases, deaths, and hospitalizations
table <- setNames(data.frame(matrix(ncol = 3, nrow = 0), stringsAsFactors = F), c("Total Reported", "On

#take the total values, current values, and 14 day change values for cases, deaths, and hospitalization
table <- rbind(table, c(data_per_day$cases[nrow(data_per_day)], data_per_day$new_cases[nrow(data_per_day

table <- rbind(table, c(data_per_day$death[nrow(data_per_day)], data_per_day$new_deaths[nrow(data_per_da

table <- rbind(table, c(NA, data_per_day$hospitalized[nrow(data_per_day)], data_per_day$new_hospitalized

#add row and re-add column names that were lost during rbind
rownames(table)<-c("Cases", "Deaths", "Hospitalized")
colnames(table)<-c("Total Reported", "On Jan. 17", "14-Day Change")

table
```

```
##              Total Reported On Jan. 17 14-Day Change
## Cases              23983607     169641          0.03
## Deaths               397612       1730          0.26
## Hospitalized             NA     124387          0.03
```

Like the associated figure, this table was fairly easy to reproduce. The first two columns are very easy to create since they simply require the total number reported as well as the daily number reported which are both accessible with very little data wrangling. The last column was a bit trickier since although they do mention how they calculated it, it's essentially hidden in the fine print.

Additionally, it is important to recognize the slight difference between how hospitalizations are defined compared to cases and deaths. Our value of 124387 is the number of people hospitalized on January 17th, 2021, not the number of NEW hospitalization cases. This also means that our 14-day change is defined slightly differently. For hospitalizations, it is the change in total hospitalizations. This is not explained super well in the article which detracts slightly from its reproducibility.

Obviously, there are some slight cosmetic differences between our table and the table in the article. Our table does not include the arrows which appear to signify the current trend. The table in the article also highlights the Cases information in red and rewords the total number of cases reported slightly. Since all of our numbers do indeed match the paper, despite the slight discrepancy with hospitalization, this table can be considered reproducible and the steps to do so are fairly straightforward.

Table 4.

In order to create this table, we return back to our us-counties.csv file. We will perform many of the same steps as the work we did for Figure 1., however this time we will also be grouping by states. The values we care about are the total number of cases as of January 17th, 2021 for each state, as well as the daily average in the last 7 days as of January 17th, 2021 for each state.

```
#initial data cleaning, drop missing values and group by date and state so that there is
#one entry of cases per date for each state
state_cases_per_day <- us_counties[,c("date", "state", "cases")] %>%
  drop_na() %>%
```

```r
  group_by(state,date) %>%
  summarise(cases = sum(cases))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

```r
#make the date variable readable in R
state_cases_per_day$date <- as.Date(state_cases_per_day$date, "%m/%d/%y")

#order data chronologically within each state
state_cases_per_day <- state_cases_per_day[order(state_cases_per_day$state, state_cases_per_day$date),]

#create a column for new cases by subtracting the previous day's total number
#of cases from the current day's
#this will cause us to have incorrect information for the first day of the
#pandemic for each state, however can easily address this afterwards
#we will need this information for our 7 day average
state_cases_per_day$new_cases <- state_cases_per_day$cases - lag(state_cases_per_day$cases)

#the code below replaces the first entry of new cases for each state with the
#first entry of total cases for each state like it is done for Figure 1.
for(i in unique(state_cases_per_day$state))
{
  state_cases_per_day$new_cases[which(state_cases_per_day$state==i)[1]] <- state_cases_per_day$cases[whi
}

#we will need to create a temporary dataset to store each state's data
#individually in order to create accurate 7 day averages
state_cases_per_day$new_cases_7dayavg <- 0 #initialize the column

for(i in unique(state_cases_per_day$state))
{
  #create a temporary list of new cases for the given state
  state_new_cases_temp <- state_cases_per_day$new_cases[which(state_cases_per_day$state==i)]

  #calculate the 7 day average and shift them so they align properly
  state_new_cases_7dayavg_temp <- round(rollmean(state_new_cases_temp,
                                                  k = 7, fill = NA))
  state_new_cases_7dayavg_temp <- lag(state_new_cases_7dayavg_temp, n = 3)

  #link the 7 day average data to the corresponding state and day
  for(j in which(state_cases_per_day$state==i))
      {
        state_cases_per_day$new_cases_7dayavg[j]<-state_new_cases_7dayavg_temp[j-which(state_cases_per_
      }
}

#create a new dataset that only contains values corresponding to January 17th, 2021
state_cases_final <- state_cases_per_day %>% filter(date == as.Date("2021-01-17"))
```

Now that we have wrangled our data, we can drop the unnecessary columns in order to create our table.

```r
table2 <- state_cases_final[c("state", "cases", "new_cases_7dayavg")]
colnames(table2) <- c("state", "total cases", "daily average in last 7 days")
table2
```

```
## # A tibble: 55 x 3
## # Groups:   state [55]
##    state                `total cases` `daily average in last 7 days`
##    <chr>                        <int>                          <dbl>
##  1 Alabama                     422598                           2957
##  2 Alaska                       51630                            242
##  3 Arizona                     673882                           7905
##  4 Arkansas                    271154                           2297
##  5 California                 3006583                          39580
##  6 Colorado                    376921                           1986
##  7 Connecticut                 223422                           2490
##  8 Delaware                     70294                            717
##  9 District of Columbia         33851                            294
## 10 Florida                    1571271                          13467
## # ... with 45 more rows
```

Like the previous table, this table was fairly easy to reproduce. The hardest part was sorting the data in order to properly calculate the daily average in the last 7 days. We can see that our numbers match perfectly with the numbers reported in the New York Times Article. We have already worked with this data in previous sections so we know it is easy to work with and the variables all make sense and are named properly. I consider this table to be reproducible, and the steps to do so were relatively simple.

Conclusion

Based on my experience working on this project. I would say that the tables and figures presented in the New York Times article are definitely reproducible. The data is easily accessible and easy to understand since it is sorted well and the columns are all properly named. Although there were some steps that were not so clear, particularly with how the 7 day averages were calculated and information regarding hospitalizations, I did not find any part of the reproducing process to be particularly difficult or undoable.

In the end all our values matched the values presented in the figures and tables which is always a good sign. Obviously extra steps would be needed in or to cosmetically reproduce their figures and tables perfectly, but for the scope of this project, that is not necessary. I would give the article an A for reproduceability due to the accessibility of the data used and the overall clear explanations on how values are calculated. The slight lack of clarity for the 7 day averages and hospitalization definitions holds the paper back from the coveted A+ grade.