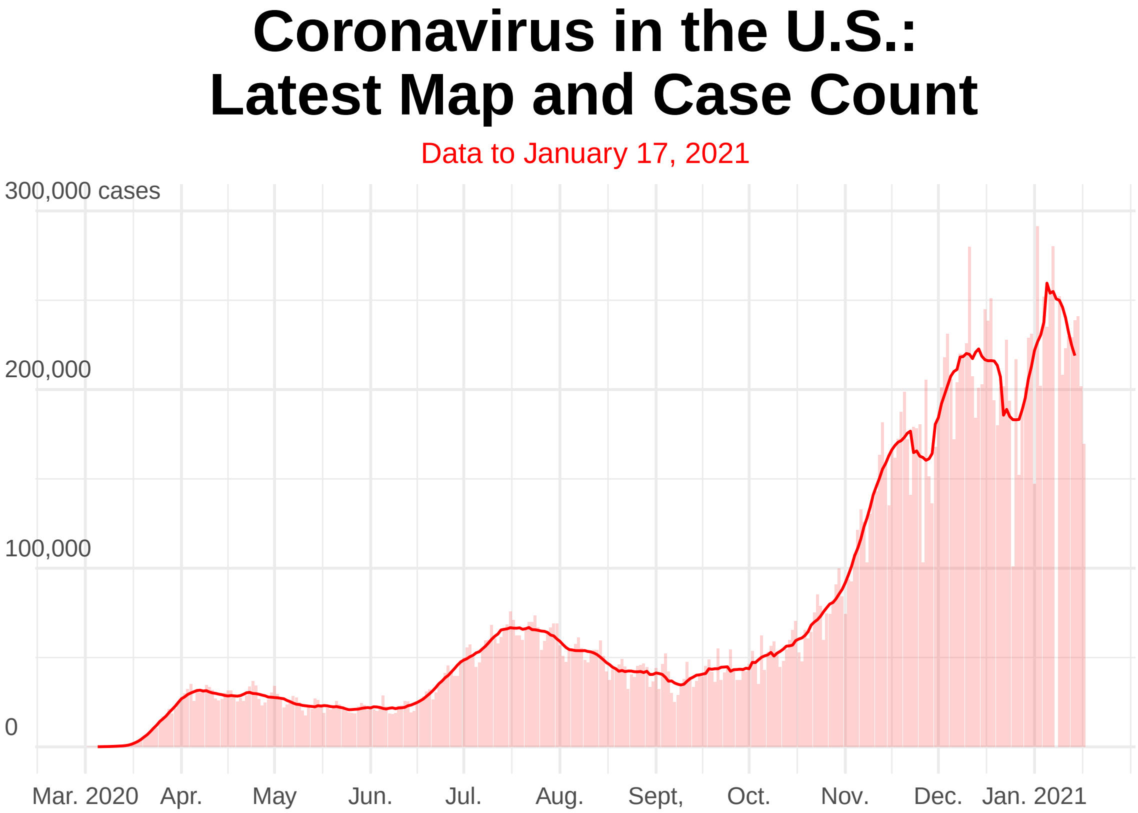
Figure 1: Coronavirus in the U.S.: Latest Map and Case Count



The figure above is reproduced from the New York Times COVID-19 visualizations which shows daily new case counts from March 1, 2020 to January 17, 2021. Additionally, the figure shows a 7-day running average of new cases denoted by the red line. The data is available on **NYT Github repository** under us-counties.csv. Code used to reproduce this figure and tables below can be found at the bottom of this document and the BST270_FinalProject.Rmd file in the Github repository.

Critique: The data is publicly available and easy to assess (link above). The us-counties.csv file doesn't have a specific code book but you can figure out what the variables were by reading the ReadMe file in the NYT repository. The ReadMe file does a good job at describing data sources, management and storage. The file names I would say were not to hard to understand and the website does a good job at trying to describe the files and data collected. The variable names were a bit deceiving in that I wasn't sure if they were new or total cases/deaths at first until you read the ReadMe file. No software or toy example were available. Overall, we were able to reproduce the figure very closely. I could not replicate the yellow bars because no specific dates were given for reporting anomalies. In conclusion, I thought it was a nice, intuitive and reproducible figure (speaking as a data scientist).

Table 1: Cases, Deaths and Hospitalized

	Total Reported	On Jan. 17	14-Day Change
Cases	23,983,607	169,641	+3%
Deaths	397,612	1,730	+26%
Hospitalized		124,387	+3%

Table 1 shows the cases, deaths and hospitalizations complimenting the figure above. Note that the 14-day percent change is computed using the 7-day running average of new cases.

Critique: The data for cases and deaths were available from the same data set as the figure above (us_counties.csv) and **hospitalized** (all_states_history.csv) data was provided from the COVID Tracking Project website. Both data sets needed for the table were publicly available and easy to assess.

The all_states_history.csv file doesn't have a specific code book but you can figure out what the variables were by reading the website definitions. I did not find a file describing data sources, management and storage. There seems to only be one file name that holds all the data collected (can get it by state too) which is really nice and easy to use. The variable names were descriptive and more details about them could be found on the website definitions. No software or toy example were available. Overall, we were able to reproduce the table.

Table 2 : Cases by State

State	Total Cases	Daily Avg. in Last 7 Days
Arizona	673,882	7,905
California	3,006,583	39,580
South Carolina	388,184	4,808
Rhode Island	104,443	976
Oklahoma	354,979	3,374
Georgia	791,322	8,457
Utah	323,837	2,548
Texas	2,127,334	22,782
New York	1,242,818	15,281
Massachusetts	470,140	5,336

Table 2 shows information on the number of total new cases in the last seven days (January 11-17, 2021) and daily average of new cases in the same time span for ten states. The states are ordered by the most cases per 100,000 residents, a measure not shown in this table.

Critique: The data for this table is publicly available and easy to assess (us_counties.csv, link above). The data information is the same as the figure so I will not repeat. The table was easy to recreate once you had aggregated the cases by state and date then computed the new case and mean for the only the seven days needed. No software or toy example were available. Overall, we were able to reproduce the table.

Code to Reproduce Figure and Tables

Reproduce Figure 1

```r
# Load libraries
library(ggplot2)
library(tidyverse)
library(stringr)
library(zoo)
library(lubridate)
library(kableExtra)
library(data.table)
```

```
#import data needed to reproduce plots
us_counties <-
  #data for deaths and cases from NYT GitHub repository
  data.table::fread("https://github.com/nytimes/covid-19-data/raw/master/us-counties.csv")

all_states_history <-
  #data for hospitalizations in The COVID Tracking Project website
  data.table::fread("https://covidtracking.com/data/download/all-states-history.csv")

#subset both data sets from March 1, 2020 to January 17, 2021
newcases_data <-
  subset(us_counties,
         date >= as.Date("2020-03-01") & date < as.Date("2021-01-18"))
hosp_data <-
  subset(all_states_history,
         date >= as.Date("2020-03-01") & date < as.Date("2021-01-18"))

#aggregate new cases by day
cases_byday <-
  aggregate(cases ~ date, data = newcases_data, sum, na.rm = TRUE)

#compute new cases by day using lag function
cases_byday$new_cases <-
  cases_byday$cases - lag(cases_byday$cases)

#compute a 7-day rolling average of new cases
cases_byday <- cases_byday %>%
  dplyr::arrange(desc(date)) %>%
  dplyr::mutate(cases_07da = zoo::rollmean(new_cases, k = 7, fill = NA))

#this will limit the plot dates
min <- as.Date("2020-3-1")
max <- as.Date("2021-1-17")

#create plot of new case counts with 7- day rolling average
ggplot(cases_byday, aes(date, new_cases)) +
  geom_bar(stat = "identity", fill = "#FF6666", alpha = 0.3) + # creates the bar plot for daily new cas
  theme_minimal()  + geom_line(aes(date, cases_07da), color = "red")  + # creates line for 7 day rollin
  scale_x_date(  # creates the ticks on the x-axis
    breaks = as.Date(
      c(
        "2020-03-01",
        "2020-04-01",
        "2020-05-01",
        "2020-06-01",
        "2020-07-01",
        "2020-08-01",
        "2020-09-01",
        "2020-10-01",
        "2020-11-01",
        "2020-12-01",
        "2021-01-01"
      )
    ),
    labels = c(  #labels for the ticks on the x-axis
      "Mar. 2020",
      "Apr.",
      "May",
      "Jun.",
      "Jul.",
      "Aug.",
      "Sept,",
      "Oct.",
      "Nov.",
      "Dec.",
      "Jan. 2021"
    )
  ) +
  scale_y_continuous( # creates the ticks for the y-axis
    limits = c(0, 300000), #limit for the y-axis
    breaks = c(0, 100000, 200000, 300000), # breaks and labels for the ticks on the y-axis
    labels = c("0", "100,000", "200,000", "300,000 cases")
  ) +
  ylab("") + xlab("") + labs(title = "Coronavirus in the U.S.:\n Latest Map and Case Count", subtitle =
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 22),
    plot.subtitle = element_text(hjust = 0.5, color = "red"),
    axis.text.y = element_text( #moves y-axis text closer to the plot
      vjust = -0.7,
      hjust = 0,
      margin = margin(l = 20, r = -50)
    )
  )
```

Reproduce Table 1

```
#create a blank table
table1 <- data.frame(matrix(NA, 3, 3))

#first value in table is extracted from the number of total reported cases as of January 17, 2021
table1[1, 1] <-
  as.numeric(cases_byday[which(cases_byday$date == as.Date("2021-01-17")), "cases"])

#extracts the number of new cases on January 17, 2021 from data set previously created
table1[1, 2] <-
  as.numeric(cases_byday[which(cases_byday$date == as.Date("2021-01-17")), "new_cases"])

## need to compute new deaths (same code as new cases above)
#aggregate deaths by day
deaths_byday <-
  aggregate(deaths ~ date, data = newcases_data, sum, na.rm = TRUE)

#compute new deaths by day
deaths_byday$new_deaths <-
  deaths_byday$deaths - lag(deaths_byday$deaths)
```

```r
#compute a 7-day rolling average for deaths
deaths_byday <- deaths_byday %>%
  dplyr::arrange(desc(date)) %>%
  dplyr::mutate(deaths_07da = zoo::rollmean(new_deaths, k = 7, fill = NA))

#extract the number of total reported deaths as of January 17, 2021
table1[2, 1] <-
  deaths_byday[which(deaths_byday$date == as.Date("2021-01-17")), "deaths"]
#extracts the number of new deaths on Jan 17, 2021
table1[2, 2] <-
  deaths_byday[which(deaths_byday$date == as.Date("2021-01-17")), "new_deaths"]

## need to compute new hospitalized data

#first aggregate hospitalizations by day
hosp_byday <-
  aggregate(hospitalizedCurrently ~ date,
            data = hosp_data,
            sum,
            na.rm = TRUE)

#compute a 7-day rolling average for hospitalized
hosp_byday <- hosp_byday %>%
  dplyr::arrange(desc(date)) %>%
  dplyr::mutate(hosp_07da = zoo::rollmean(hospitalizedCurrently, k = 7, fill = NA))

#extract the number of new hospitalized on January 17, 2021
table1[3, 2] <-
  hosp_byday[which(hosp_byday$date == as.Date("2021-01-17")), "hospitalizedCurrently"]


##need to compute 14 day percent change (this requires percent change using the 7 day average variable)
# (Today average # - 14 days ago average # )/ 14 days ago average # *100
# for cases
for (i in 1:nrow(cases_byday)) {
  cases_byday$trend14days[i] <-
    round((cases_byday$cases_07da[i] - cases_byday$cases_07da[i + 14]) /
            cases_byday$cases_07da[i + 14] * 100)
}

# adding plus and percent sign like in table
cases_byday$trend14days <-
  ifelse(
    cases_byday$trend14days > 0 ,
    str_c("+", cases_byday$trend14days, "%"),
    str_c(cases_byday$trend14days, "%")
  )

#for deaths
for (i in 1:nrow(deaths_byday)) {
  deaths_byday$trend14days[i] <-
    round((deaths_byday$deaths_07da[i] - deaths_byday$deaths_07da[i + 14]) /
            deaths_byday$deaths_07da[i + 14] * 100)
}

# adding plus and percent sign like in table
deaths_byday$trend14days <-
  ifelse(
    deaths_byday$trend14days > 0 ,
    str_c("+", deaths_byday$trend14days, "%"),
    str_c(deaths_byday$trend14days, "%")
  )

#for hospitalized
for (i in 1:nrow(hosp_byday)) {
  hosp_byday$trend14days[i] <-
    round((hosp_byday$hosp_07da[i] - hosp_byday$hosp_07da[i + 14]) /
            hosp_byday$hosp_07da[i + 14] * 100)
}
# adding plus and percent sign like in table
hosp_byday$trend14days <-
  ifelse(
    hosp_byday$trend14days > 0 ,
    str_c("+", hosp_byday$trend14days, "%"),
    str_c(hosp_byday$trend14days, "%")
  )

# adding the 14 day change values for most recent day available into the table
table1[1, 3] <-
  cases_byday$trend14days[min(which(!is.na(cases_byday$trend14days)))]

table1[2, 3] <-
  deaths_byday$trend14days[min(which(!is.na(deaths_byday$trend14days)))]

table1[3, 3] <-
  hosp_byday$trend14days[min(which(!is.na(hosp_byday$trend14days)))]

#fixing table with proper column and row names
colnames(table1) <-
  c("Total Reported", "On Jan. 17", "14-Day Change")
rownames(table1) <-
  c("Cases", "Deaths", "Hospitalized")
#changing values to numeric
table1$`Total Reported` <-
  as.numeric(table1$`Total Reported`)

#creating table
opts <- options(knitr.kable.NA = "")
kable(table1,
      "latex",
      format.args = list(big.mark = ","),
      booktabs = T) %>%
  kable_styling(position = "center")
```

Reproduce Table 2

```r
# subset to only the 7 days of interest (January 10-17, 2021)
last7days_cases <-
  subset(us_counties,
         date <= as.Date("2021-01-17") & date >= as.Date("2021-01-10"))

#aggregate new cases by date
cases_bydatestate <-
  aggregate(cases ~ date + state, data = last7days_cases, sum, na.rm = TRUE)

# compute new cases
cases_bydatestate$new_cases <-
  cases_bydatestate$cases - lag(cases_bydatestate$cases)

# do not need the information for this day, only needed to compute the new cases for Jan. 11
new_df_cases <-
  subset(cases_bydatestate, date != as.Date("2021-01-10"))

#create empty table 2
table2 <- data.frame(matrix(NA, 55, 3))

# this fills the table with the number of total cases in each state
table2[1:55, 1:2] <-
  cases_bydatestate[which(cases_bydatestate$date == as.Date("2021-01-17")), c("state", "cases")]
# this computes the daily average number of cases for the last 7 days
table2[1:55, 3] <-
  round(aggregate(new_cases ~ state, data = new_df_cases, mean, na.rm = TRUE)[2])

#rename column names of table 2
colnames(table2) <-
  c("State", "Total Cases", "Daily Avg. in Last 7 Days")

# extract the top ten states that have the most cases per 100,000 residents (not computed by us just fo
kable(
  table2[c(3, 5, 44, 43, 39, 11, 48, 47, 34, 23), ],
  booktabs = T,
  format.args = list(big.mark = ","),
  row.names = F
) %>%
  kable_styling(position = "center")
```