```
#LIBRARY'S
library(tidyverse)
library(plotly)
library(data.table)
library(ggplot2)
library(maps)
library(dplyr)
library(tidyr)
library(lubridate)
```

Covid-19 Dataset

Download us-states.csv (https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-states.csv) from https://github.com/nytimes/covid-19-data/ (https://github.com/nytimes/covid-19-data/). README.md (https://github.com/nytimes/covid-19-data/blob/master/README.md) for details on file content.

```
data1 = fread("us-states.csv")
```

```
head(data1)
```

```
##          date      state fips cases deaths
## 1: 2020-01-21 Washington   53     1      0
## 2: 2020-01-22 Washington   53     1      0
## 3: 2020-01-23 Washington   53     1      0
## 4: 2020-01-24   Illinois   17     1      0
## 5: 2020-01-24 Washington   53     1      0
## 6: 2020-01-25 California    6     1      0
```

```
data1$date = as.Date(data1$date)
data_us = data1 %>%
  group_by(state, year_month = format(date, "%Y-%m")) %>%
  summarise(fips = max(fips), cases_cum = max(cases), deaths_cum = max(deaths), date=
min(date)) %>%
  mutate(cases=cases_cum-lag(cases_cum,default=0))
```

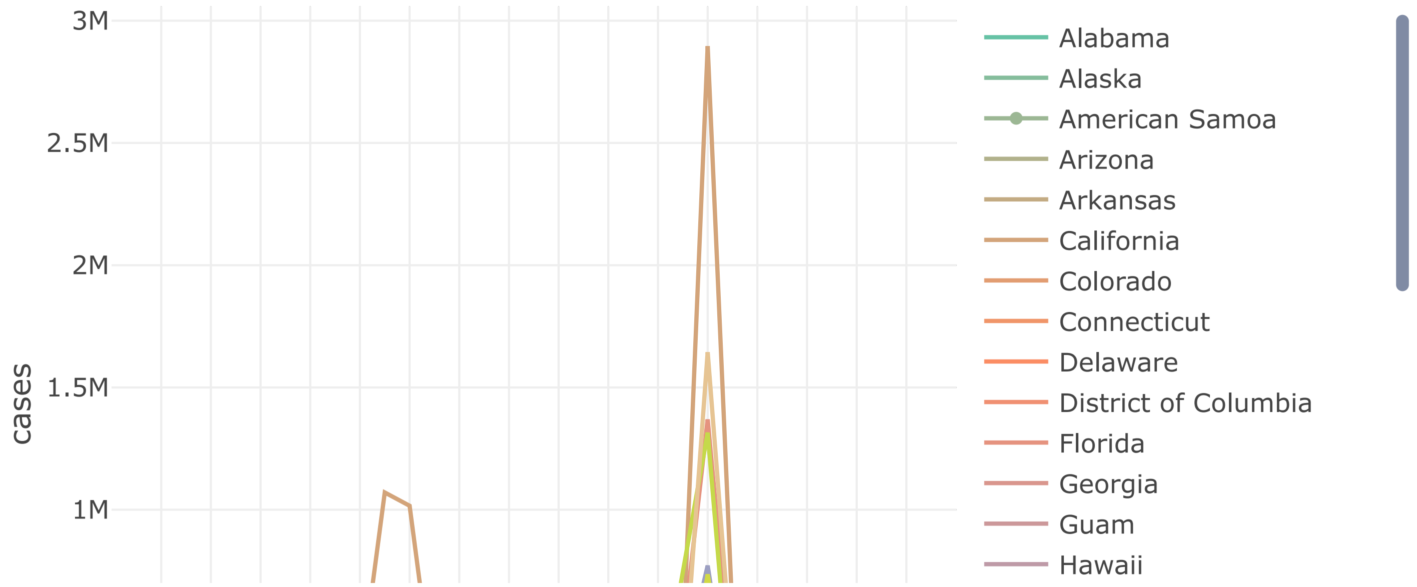
```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

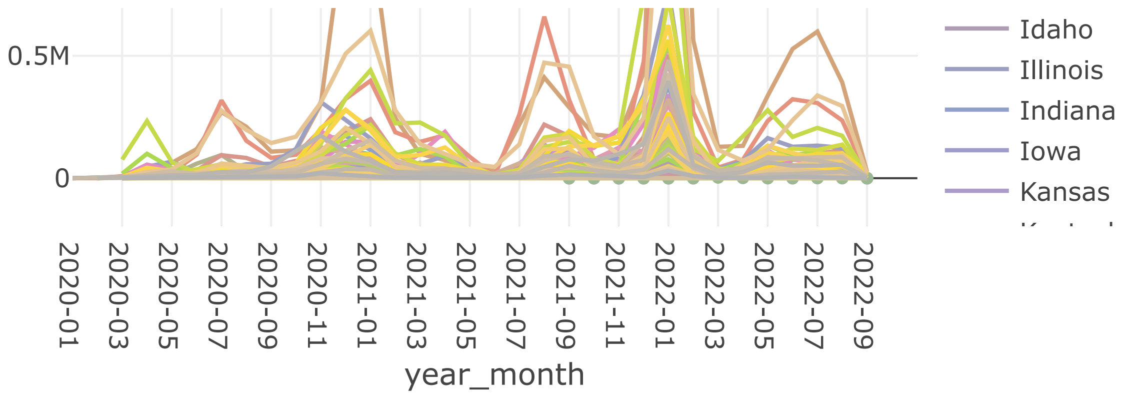
```
data_us
```

```
## # A tibble: 1,732 × 7
## # Groups:   state [56]
##    state   year_month  fips cases_cum deaths_cum date         cases
##    <chr>   <chr>      <int>     <int>      <int> <date>       <int>
##  1 Alabama 2020-03        1       999         14 2020-03-13    999
##  2 Alabama 2020-04        1      7068        272 2020-04-01   6069
##  3 Alabama 2020-05        1     17952        630 2020-05-01  10884
##  4 Alabama 2020-06        1     38045        950 2020-06-01  20093
##  5 Alabama 2020-07        1     87723       1580 2020-07-01  49678
##  6 Alabama 2020-08        1    126058       2182 2020-08-01  38335
##  7 Alabama 2020-09        1    154701       2540 2020-09-01  28643
##  8 Alabama 2020-10        1    192285       2967 2020-10-01  37584
##  9 Alabama 2020-11        1    249524       3578 2020-11-01  57239
## 10 Alabama 2020-12        1    361226       4827 2020-12-01 111702
## # ℹ 1,722 more rows
```

```
state_plot = data_us %>%
  plot_ly(x = ~year_month, y = ~cases, color = ~state, type = 'scatter',mode = 'Pat
h')
state_plot
```

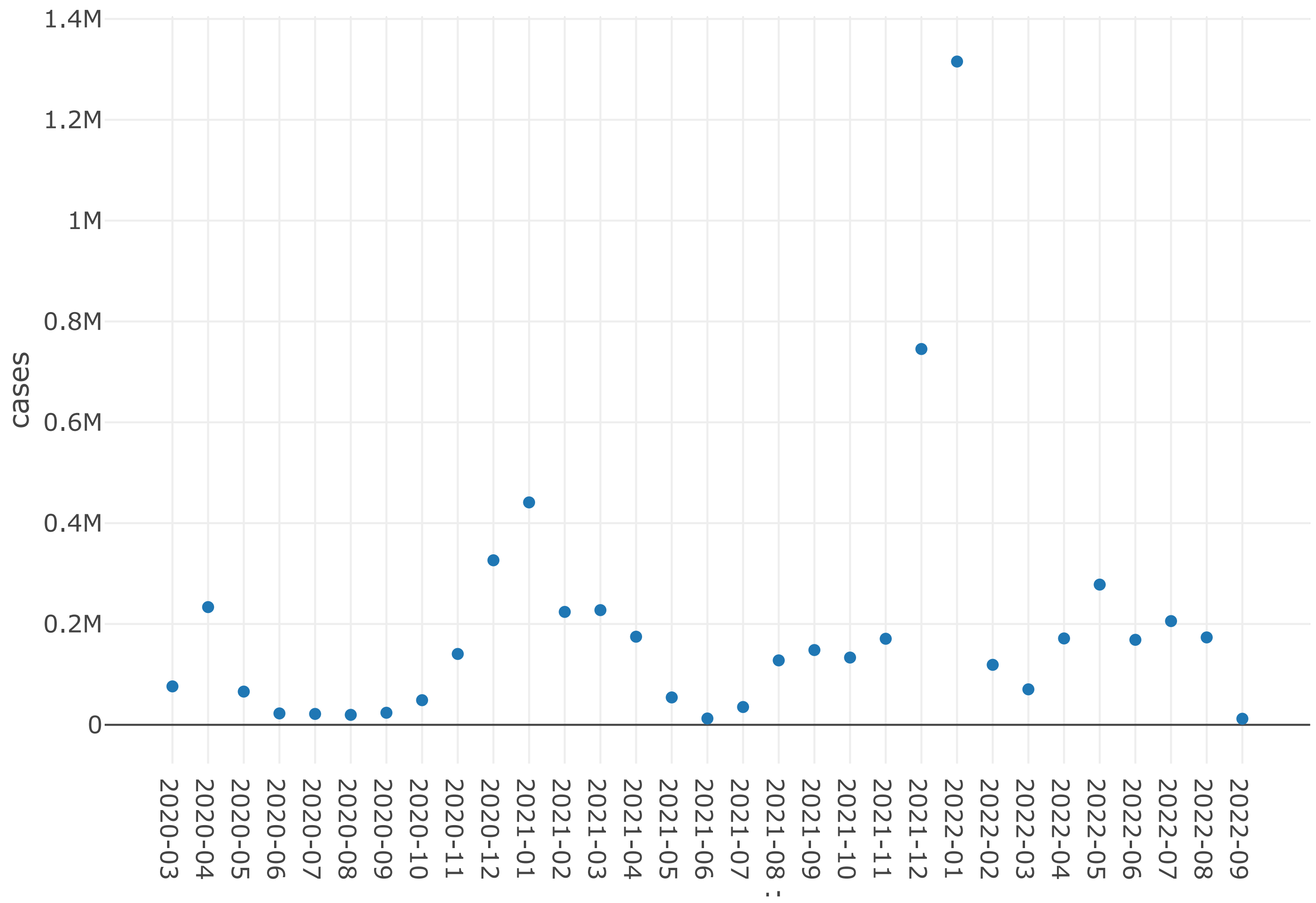
```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for p
alette Set2 is 8
## Returning the palette you asked for with that many colors

## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for p
alette Set2 is 8
## Returning the palette you asked for with that many colors
```





```
ny_data = data_us %>% filter(state == "New York")
ny_scatter.plot = ny_data %>%
  plot_ly(x = ~year_month, y = ~cases, type = 'scatter')
ny_scatter.plot
```

```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```



year_montn

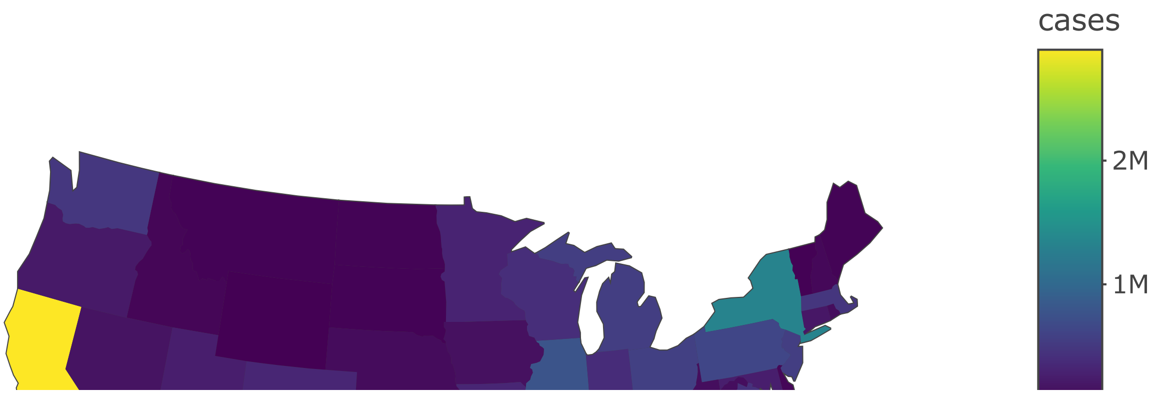
```
highest_cases = ny_data[which.max(ny_data$cases), ]
highest_cases
```

```
## # A tibble: 1 × 7
## # Groups:   state [1]
##   state    year_month  fips cases_cum deaths_cum date         cases
##   <chr>    <chr>      <int>     <int>      <int> <date>       <int>
## 1 New York 2022-01       36   4789532      64247 2022-01-01 1315562
```

```
g = list(
  scope = "usa",
  projection = list(type = 'albers usa'),
  lakecolor = toRGB('white'))

us_data_filtered = data_us[data_us$state %in% state.name, ]

us_data_filtered$state_short.name <- state.abb[match(us_data_filtered$state, state.na
me)]
dummy = us_data_filtered %>% group_by(state, state_short.name) %>% summarise(cases =
max(cases))
```

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

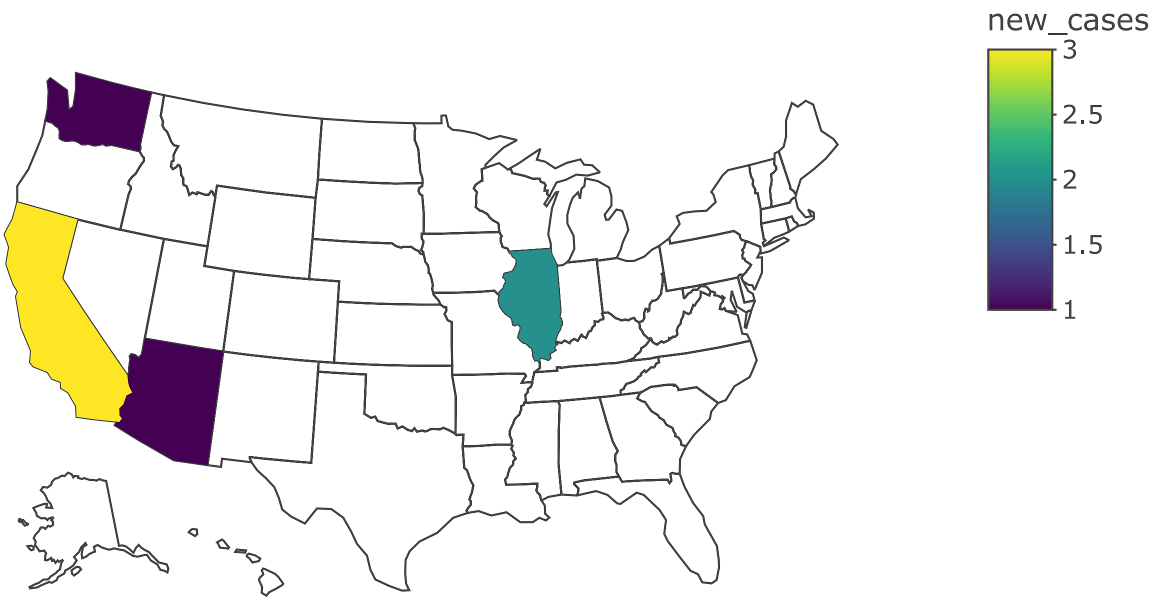
```
plot_geo(data = dummy) %>%
  add_trace(
    z = ~cases, text = ~state,  span = I(0), locations = ~state_short.name, locationm
ode = 'USA-states') %>%
  layout(geo = g)
```

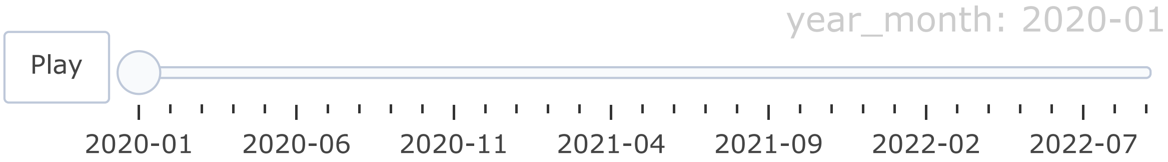


```
dummy = us_data_filtered %>% group_by(state, state_short.name, year_month) %>% summar
ise(new_cases = max(cases))
```

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

```
plot_geo(data = dummy) %>%
  add_trace(
    z = ~new_cases, text = ~state,  span = I(0), locations = ~state_short.name, locat
ionmode = 'USA-states', frame = ~year_month) %>%
  layout(geo = g)
```




year_month: 2020-01
Play
2020-01
2020-06
2020-11
2021-04
2021-09
2022-02
2022-07