

Interactive Mapping of COVID-19 Data

2024-03-10

```
library(data.table)
library(dplyr)
library(dplyr)
library(tidyr)
library(plotly)
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.

```
# Read us-states.csv
us_df = fread("us-states.csv")
```

```
# Show some values from dataframe
head(us_df, 10)
```

	date	state	fips	cases	deaths
	<IDate>	<chr>	<int>	<int>	<int>
	2020-01-21	Washington	53	1	0
	2020-01-22	Washington	53	1	0
	2020-01-23	Washington	53	1	0
	2020-01-24	Illinois	17	1	0
	2020-01-24	Washington	53	1	0
	2020-01-25	California	6	1	0
	2020-01-25	Illinois	17	1	0
	2020-01-25	Washington	53	1	0
	2020-01-26	Arizona	4	1	0
	2020-01-26	California	6	2	0

1-10 of 10 rows

```
# Create new dataframe with new cases per month for each state
```

The cases column in us states dataset appears to be cumulative since the cases keeps adding up to the point in time.

```
us_df$date <- as.Date(us_df$date)
us_states_cases <- us_df %>%
  mutate(year_month = format(date, "%Y-%m")) %>%
  group_by(state, year_month) %>%
  summarise(cases_per_month = max(cases) - min(cases), .groups = "drop")

# us_states_cases$month <- format(as.Date(paste0(us_states_cases$month, "-01")), "%b-%Y")
head(us_states_cases, 10)
```

state <chr>	year_month <chr>	cases_per_month <int>
Alabama	2020-03	993
Alabama	2020-04	5960
Alabama	2020-05	10658
Alabama	2020-06	19511
Alabama	2020-07	48761
Alabama	2020-08	36709
Alabama	2020-09	27085
Alabama	2020-10	36541
Alabama	2020-11	55539
Alabama	2020-12	108326

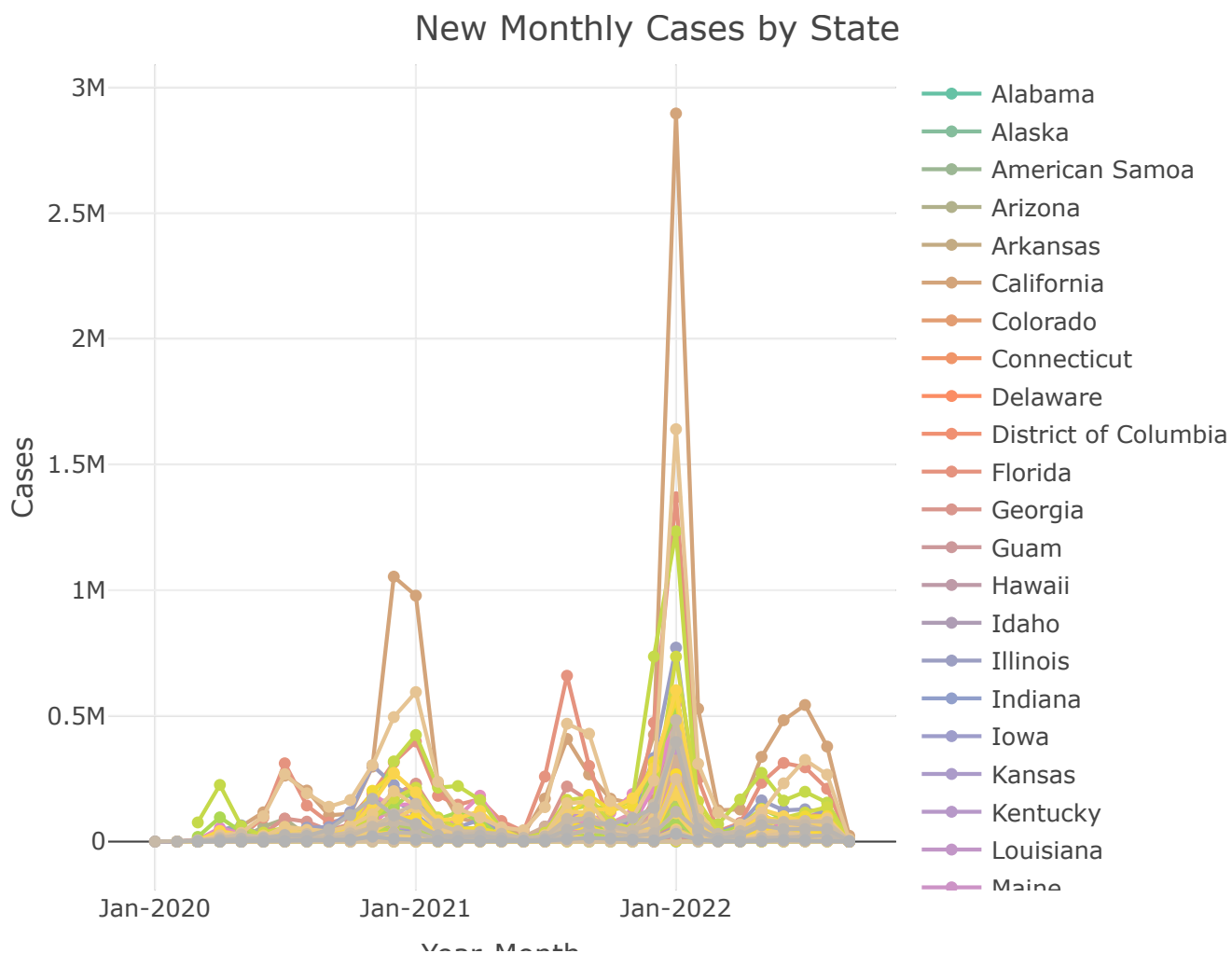
1-10 of 10 rows

```
# Using previous dataframe plot new monthly cases in states, group by states
# The resulting plot is busy, use interactive plotly capabilities to limit number
# of displayed states
```

```
us_states_cases_new <- us_states_cases %>%
  mutate(date = as.Date(paste0(year_month, "-01")))

plot_ly(us_states_cases_new, x = ~date, y = ~cases_per_month, color = ~state,
  type = "scatter", mode = "lines+markers") %>%
  layout(title = "New Monthly Cases by State",
    xaxis = list(title = "Year-Month", tickformat = "%b-%Y"),
    yaxis = list(title = "Cases"))
```

```
## 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
```

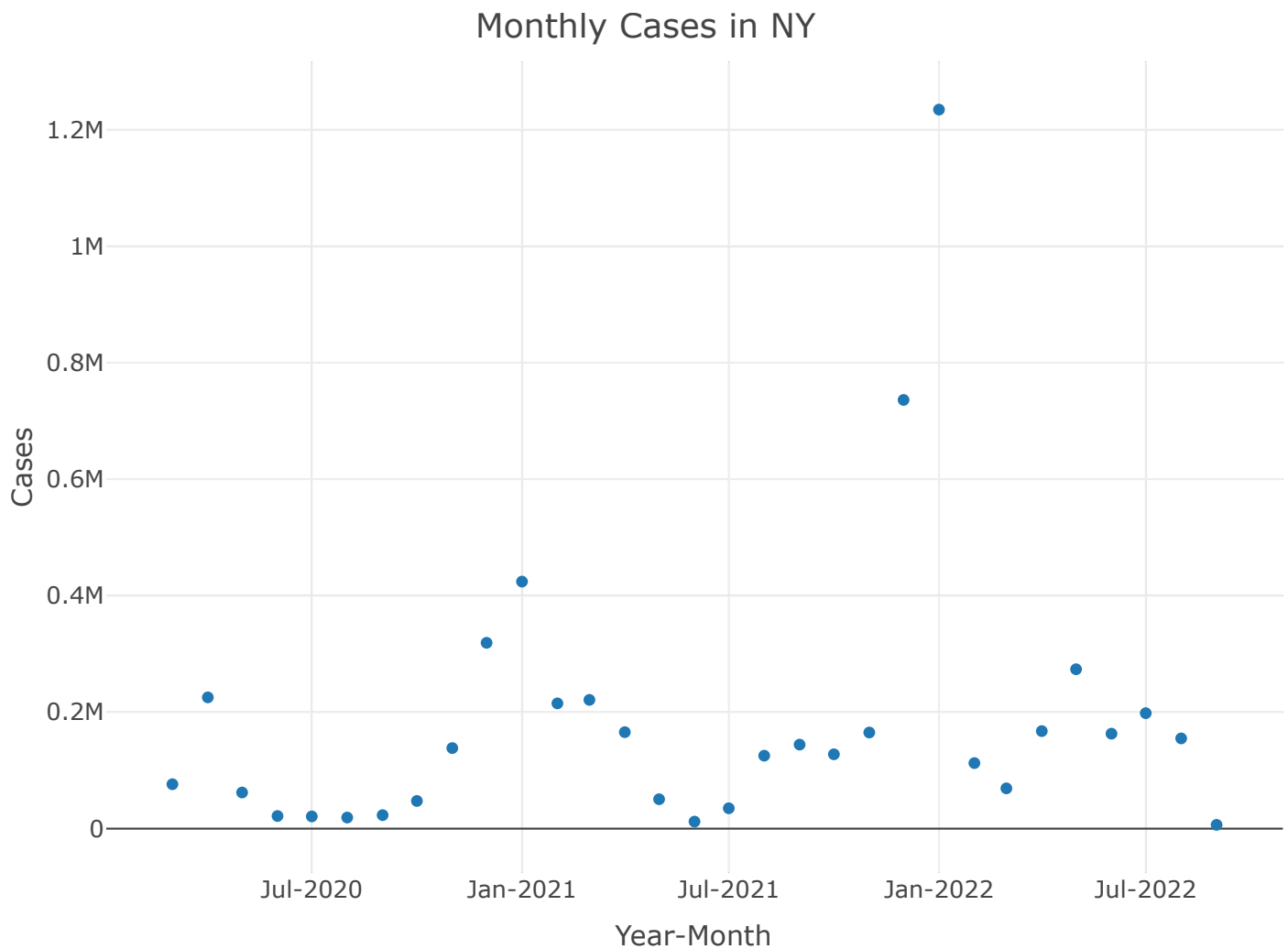


year-month

```
# Plot new monthly cases only in NY state
```

```
ny_cases <- us_states_cases_new %>%  
  filter(state == "New York")
```

```
plot_ly(ny_cases, x = ~date, y = ~cases_per_month, type = "scatter", mode = "markers") %>%  
  layout(title = "Monthly Cases in NY",  
    xaxis = list(title = "Year-Month", tickformat = "%b-%Y"),  
    yaxis = list(title = "Cases"))
```



```
# Found the year-month with highest cases in NY state

highest_ny_cases <- us_df %>%
  filter(state == "New York") %>%
  mutate(year_month = format(date, "%Y-%m")) %>%
  group_by(year_month, state, fips) %>%
  summarise(cases_cum = cumsum(cases),
            deaths_cum = cumsum(deaths),
            cases = max(cases) - min(cases), .groups = "drop_last") %>%
  arrange(desc(cases)) %>%
  select(state, year_month, fips, cases_cum, deaths_cum, cases)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
## always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
highest_ny <- head(highest_ny_cases, 1)
highest_ny
```

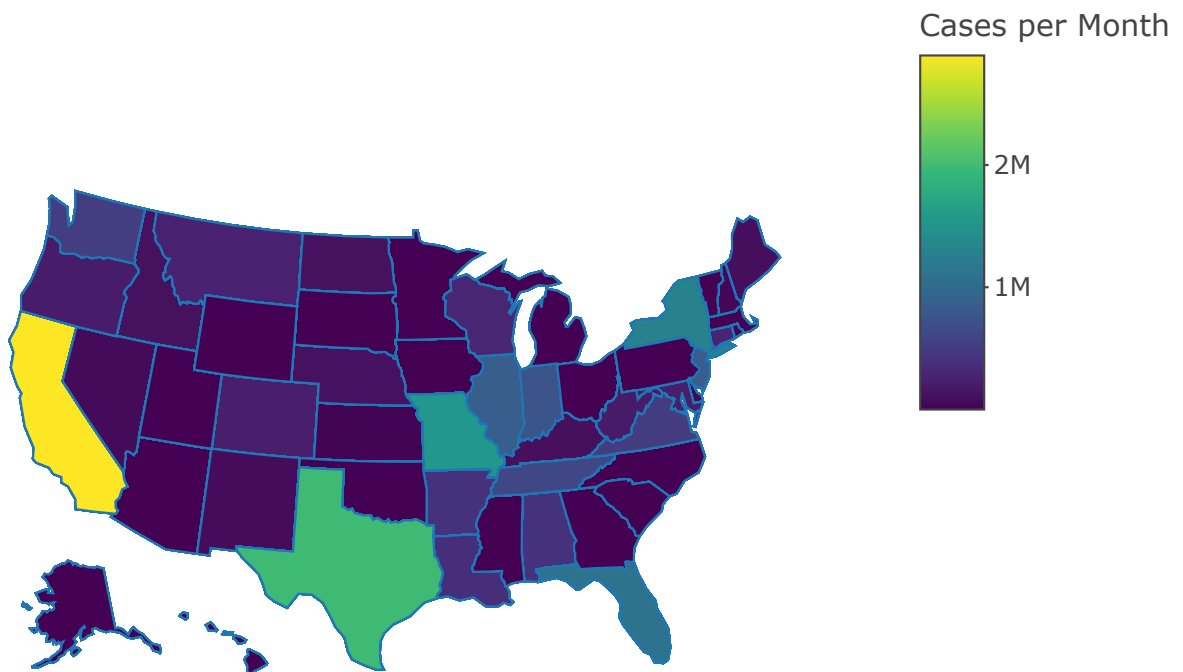
state <chr>	year_month <chr>	fips <int>	cases_cum <int>	deaths_cum <int>	cases <int>
New York	2022-01	36	3555047	59072	1234485
1 row					

```
# Plot new cases in determined above year-month
# using USA state map, color each state by number of cases

us_states_cases$code = state.abb[match(us_states_cases$state, state.name)]

plot_ly(
  type = "choropleth",
  locations = us_states_cases$code,
  locationmode = "USA-states",
  z = us_states_cases$cases_per_month*200,
  zmin = 0,
  zmax = max(us_states_cases$cases_per_month),
  text = us_states_cases$state) %>%
  colorbar(title = "Cases per Month", tickformat = ",", tickvals = c(1000000, 2000000), ticktext = c("1M", "2M")) %>%
  layout(
    title = "COVID-19 Cases by State",
    geo = list(scope = "usa", projection = list(type = "albers usa"), lakecolor = toRGB("white"))
  )
```

COVID-19 Cases by State

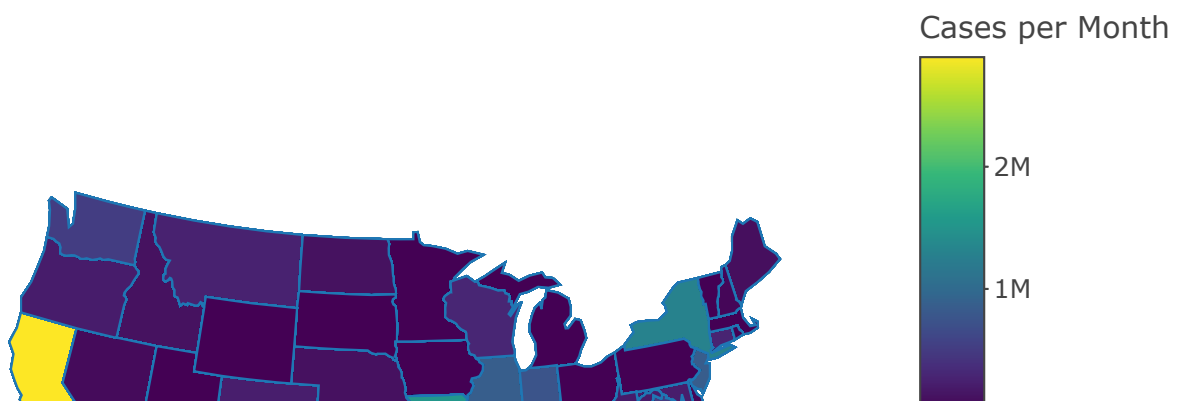


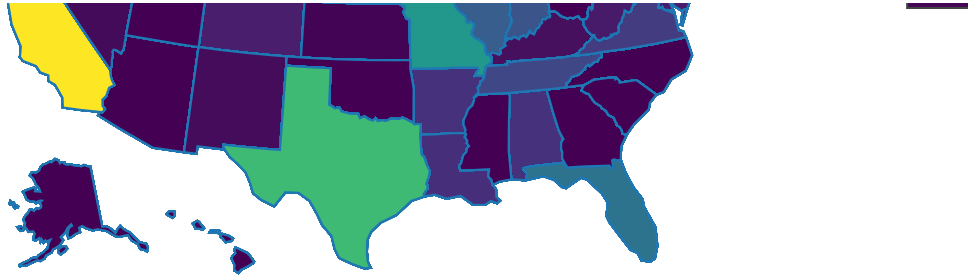
```
#Add animation capability
```

```
plot_ly(
  type = "choropleth",
  locations = us_states_cases$code,
  locationmode = "USA-states",
  z = us_states_cases$cases_per_month*200,
  zmin = 0,
  zmax = max(us_states_cases$cases_per_month),
  text = us_states_cases$state,
  animation_frame = us_states_cases$year_month,
  colorbar = list(title = "Cases per Month", tickformat = ",", tickvals = c(1000000,
2000000), ticktext = c("1M", "2M"))
) %>%
  layout(
    title = "COVID-19 Cases by State",
    geo = list(scope = "usa", projection = list(type = "albers usa"), lakecolor = toR
GB("white"))
  )
```

```
## Warning: 'choropleth' objects don't have these attributes: 'animation_frame'
## Valid attributes include:
## 'autocolorscale', 'coloraxis', 'colorbar', 'colorscale', 'customdata', 'customdata
src', 'featureidkey', 'geo', 'geojson', 'hoverinfo', 'hoverinfosrc', 'hoverlabel', 'h
overtemplate', 'hovertemplatesrc', 'hovertext', 'hovertextsrc', 'ids', 'idssrc', 'leg
endgroup', 'legendgrouptitle', 'legendrank', 'locationmode', 'locations', 'locationss
rc', 'marker', 'meta', 'metasrc', 'name', 'reversescale', 'selected', 'selectedpoint
s', 'showlegend', 'showscale', 'stream', 'text', 'textsrc', 'transforms', 'type', 'ui
d', 'uirevision', 'unselected', 'visible', 'z', 'zauto', 'zmax', 'zmid', 'zmin', 'zsr
c', 'key', 'set', 'frame', 'transforms', '_isNestedKey', '_isSimpleKey', '_isGraticul
e', '_bbox'
```

COVID-19 Cases by State





An animated plot can be useful when we want to see cases evolving with time in different states in USA. This can benefit to see the progression of cases over time, with the ability to identify patterns that may be difficult from the static plots. Whereas, the static plot, is useful when we want to show the data at a specific point in time. It may be effective when we just need a snapshot of the data at specific instance. This is useful when we want to show the states affected the most.