COVID_BY_STATE

October 8, 2020

By Edison Gu

0.1 Part 1: Six COVID-19 Related Questions

- Which state's population is coronavirus most hazard to?
- How much testing has been done in each state? Which state needs more testing?
- What are the top 20 states that have the most cases over time?
- Is the recovery process of each state related to its governor's political affiliation?
- Does the spread of the coronavirus follow a similar pattern across different state?
- Which states are seeing a slow-down in growing cases in the past month?

0.2 Part 2: About The Dataset

- There are 54 columns in the original data table, they are all variational components. Among them, `date` is sequential, `state` is nominal, and rest of the varaiables are mostly numerical values.
- In `governor` table, `party` is a nominal / binary variable.

0.3 Part 3: Analysis Needed

- For data preparation, I will first choose a selection of variational components from the orignial data table that are related to answering the six questions above;
- I will then "expand" the data table vertically so that each `date` will have observations from all 55 `state` and territories. At the same time, I will replace all missing values with 0's;
  - This is done to "trick" the visualization API to include all states from the beginning, even though we don't have complete data from all states from the start;
- I wil also join `governor` and `population` tables to the main `DataFrame`;
  - To see the effect of governor and their political affiliation;
  - To calculate cases per million population, etc - this is to "standardize" the data;
- For analysis, I will be looking at total cases, deaths, and recovers for this question;
- Besides the cumulative counts, daily changes in cases, deaths also indicates how one state is trending;
  - I will calculate the 7-day rolling average for the daily new cases to "smoooth" the curve;
  - By selecting only the result for the last 14 days, we can see whether one state is trending up or down;

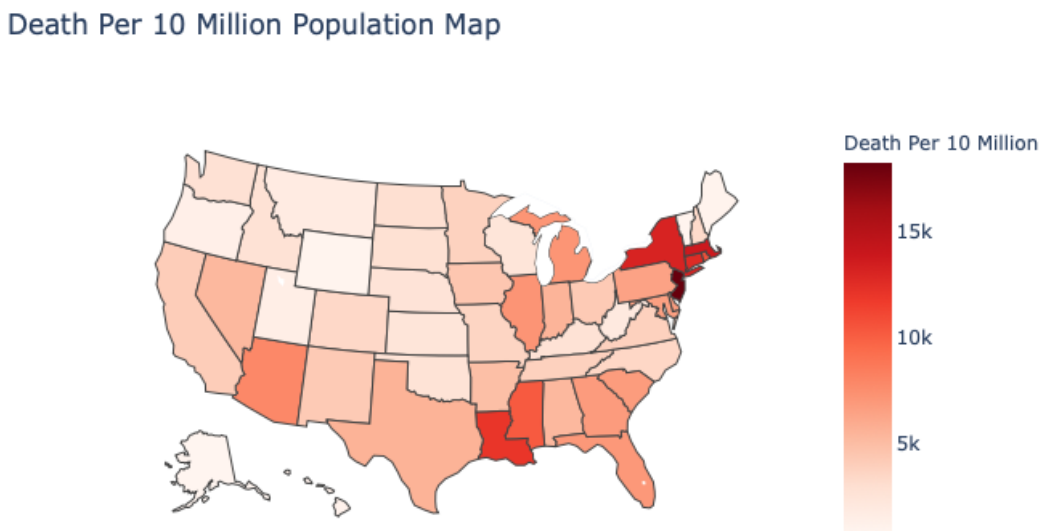
- I will also compare the presentage of death over cases - this will give us the mortality at each state;
  - A percentage will be calculated by dividing `death` over `positive`;
- To more explicitly compare between states, I will compute the fraction of cases each state contributed to the total cases in the US;
  - Eg. State total cases divided by US total cases;
- I will also join the covid daily table with the state population table to calculate cases/deaths per 1M population;
  - Eg. State total death divided by state population, then times 1 million;
- For testing capacity, I will focus on the positive rate;
  - This will be calculated by dividing `positive` by `totalTestResult`;
  - `totalTestIncrease` can tell us the increase in testing capacity through time.

0.4 Part 4: Design Process & Justification

0.4.1 Which State's Population Is Coronavirus Most Hazard To?

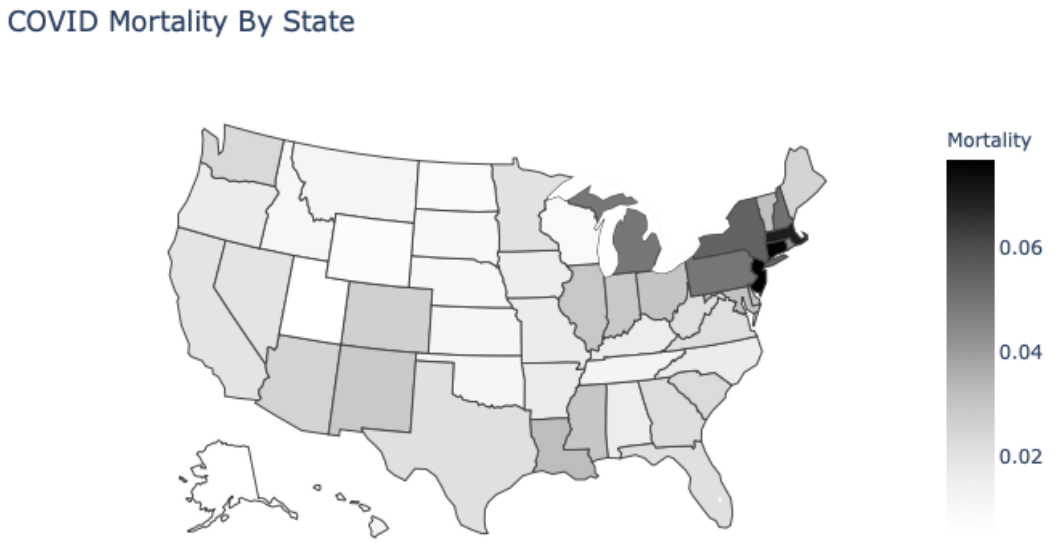
This question is aiming to pinpoint states that have the most COVID related deaths. Therefore, spacial location will be an important visual attribute.

- `State` channel will be mapped to the state location on the US map;
- `Death/10M` and `Mortality` channels will be mapped to the color intensity of each state's fill color.



Death Per 10 Million Population Map `State` location is selective, and `Death/10M` represented by the color intensity is ordered.

Through our visualization, it looks like Louisiana, New Jersey, New York among several other states have the highest number of deaths per 10 million of their state population. Coronavirus is more hazard to population in these states than others.



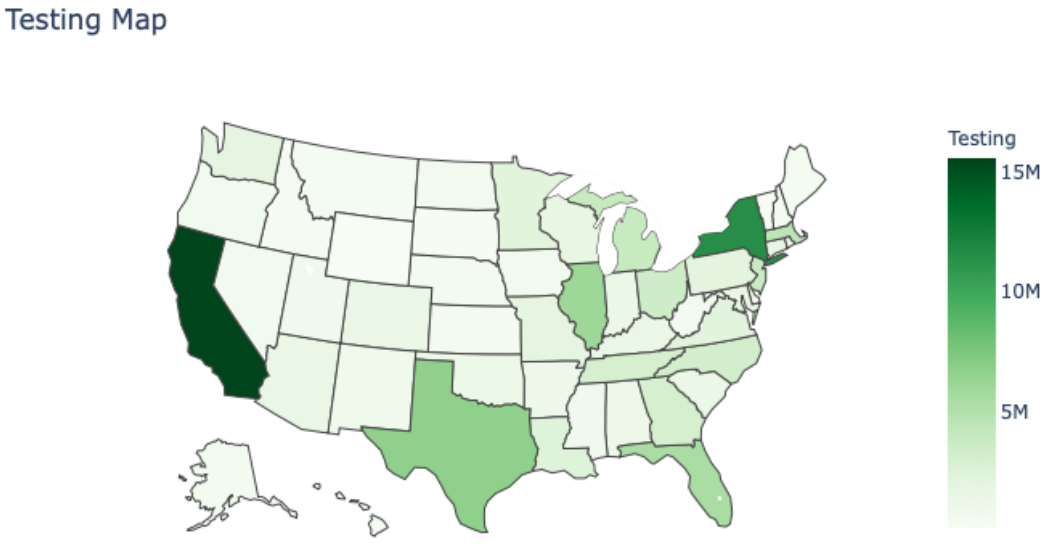
COVID Mortality By State `State` location is selective, and `Mortality` representated by the color intensity is ordered.

From this map, we can see that Connecticut, New Jersey, and Massachusettes have the highest mortality rate from COVID so far. This gives us a rough idea on the capability of the state treating the positive patients and patients at some states are more at risk than the others.

0.4.2 How Much Testing Has been Doen In Each State? Which State Needs More Testing?

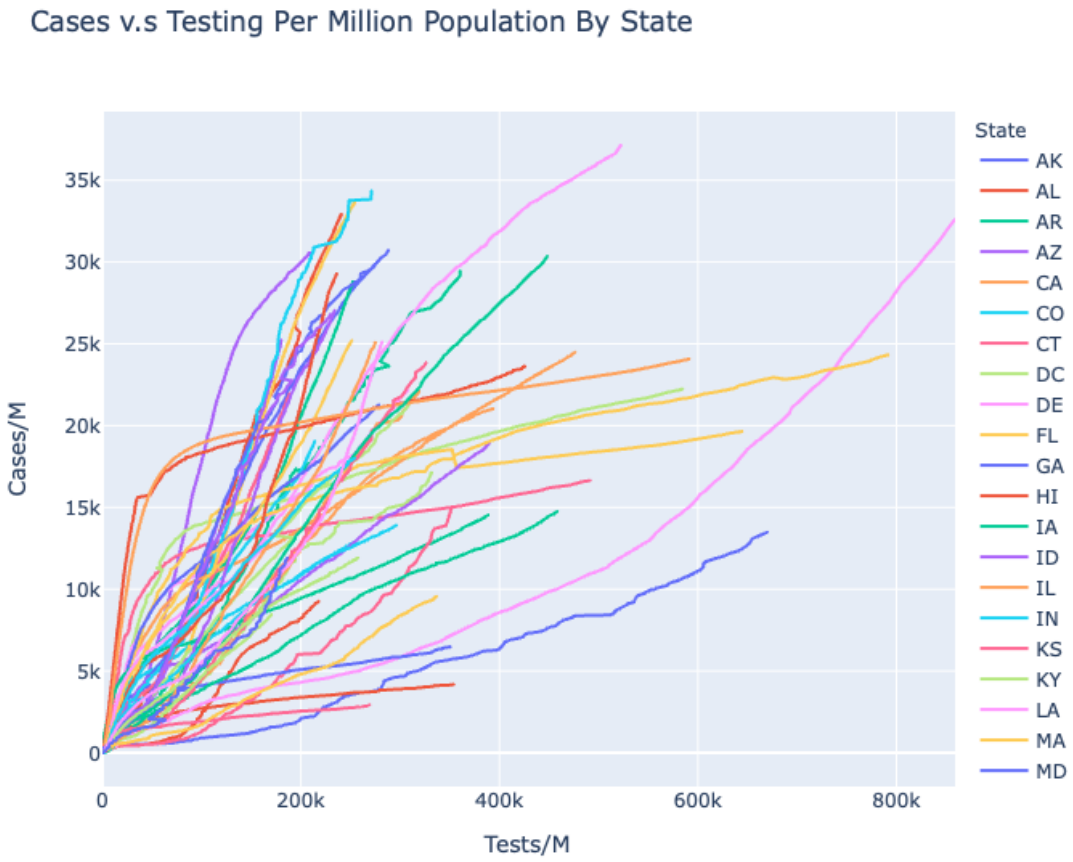
For this question, we will focus on the relationship between testing and confirmed cases.

- In the first graph:
  - `State` channel will be mapped to the state location on the US map;
  - `Testing` channels will be mapped to the color intensity of each state's fill color;
- In the second graph:
  - `Tests/M` and `Case/M` will be mapped to the `x` and `y` coordinate respectively;
  - `State` channel will be mapped to the color visual variable.



Testing Map `State` location is selective, and `Testing` represented by the color intensity is ordered.

Through our visualization, California, Texas, New York, and Illinois have conducted the most amount of testing. However, this does not necessarily means that they do not need more testing or vice versa as we will see in the next visualization.



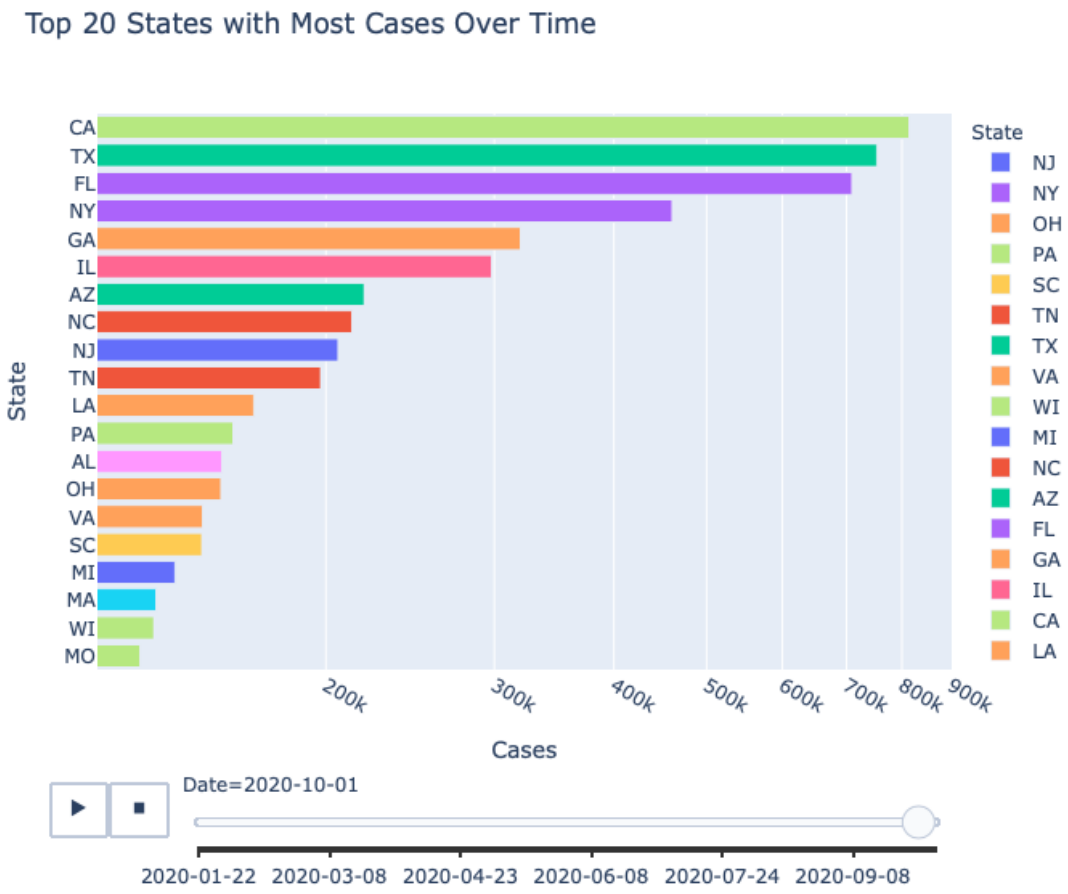
Cases v.s Testing Per Million Population By State `State` differentiated by color is selective, and so are the lines mapped out by `Cases/M` and `Test/M`.

We can see that states like NY, RI and MA, even with more testing, their cases are growing fairly slowly. On the other hand, states like ND, LA and AZ have much larger increase in confirmed cases with more testings. This means that cases are growing at these states and many cases are not identified through testing - they are the states who will need more testing.

0.4.3 What Are The Top 20 States That Have The Most Cases Over Time?

To answer this questions, I will use the following channels and marks:

- `State` is mapped to the `y` position of each mark, which is a bar. It is also mapped to colors;
- `Cases` is mapped to the `x` position or the length of each bar;
- Finally, `Date` as time is mapped to the slider, giving the ordering of top 20 states on each day.



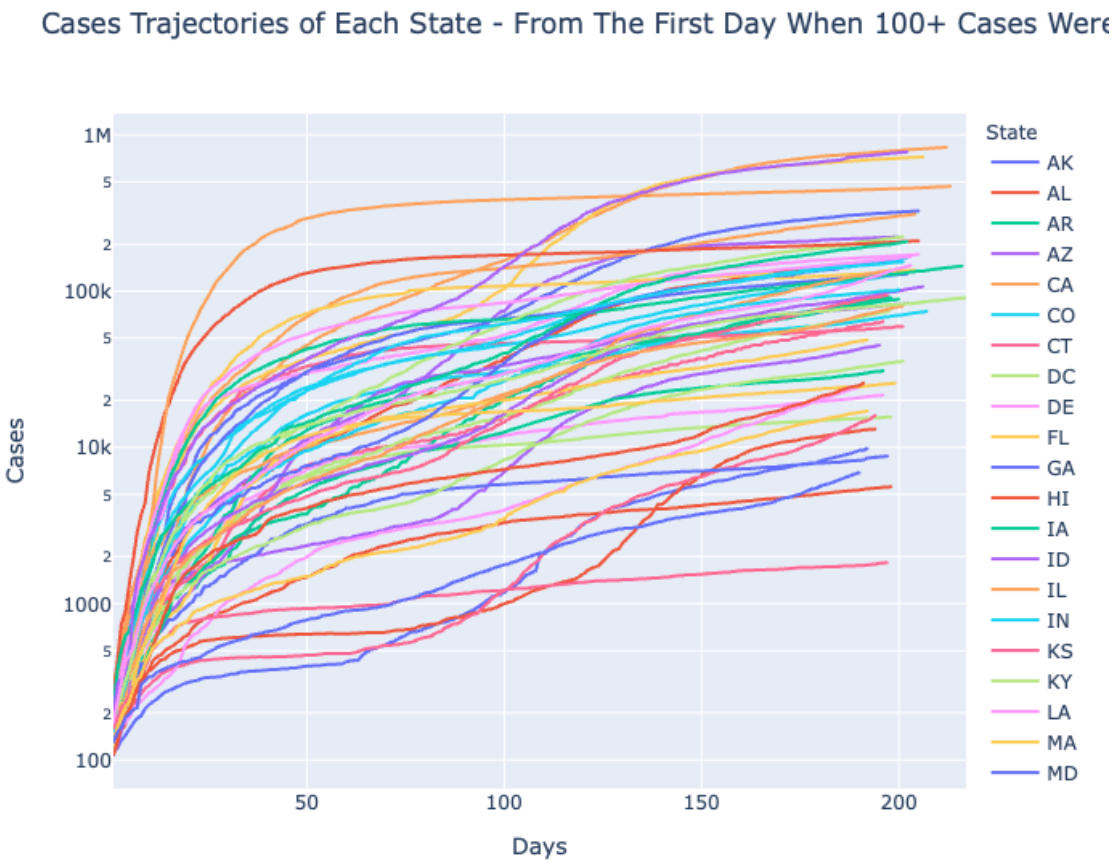
Top 20 States with Most Cases Over Time `State` on the `y` axis is ordered by the number of total cases. However, their colors are associative. `Cases` indicated by the length of the bar is quantitative, and `Date` is associative.

This graph below is interactive (see in Part 6), the top 20 states with the most cases are ranked from the highest to the lowest on each day.

0.4.4 Is The Recovery Process of Each State Related to Its Governor's Political Affiliation?

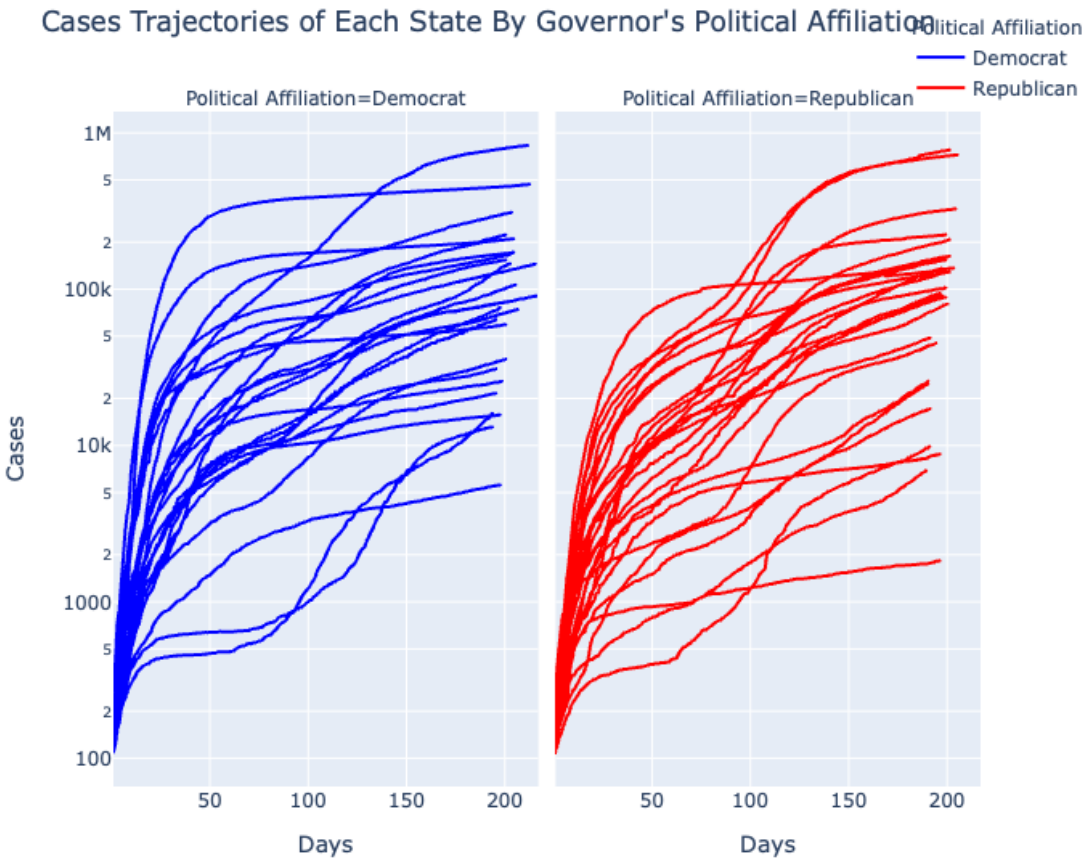
For this question, we will look at different states's cases over time to see the recovery process.

- `Cases` and `Days` after 100 cases were confirmed are mapped the `y` and `x` coordinate;
- `State` is mapped to colors.



Cases Trajectories of Each State - From The First Day When 100+ Cases Were Reported Lines on this graph is associative, so as the colors of lines representing different `state`.

From this graph, we see some states' have entered the recovery phases where cases are growing very slowly without rebound. On the other hand, some states' new cases are growing much faster. We will break them down by political party in the next graph.



Cases Trajectories of Each State By Governor's Political Affiliation Similar to the visualization above, but here `Political Affiliation` is mapped to color and has separated the lines into two graphs. Color here can be selective as there are only two colors and separated by left-and-right position. Plus, the underlying meaning of "Red" state and "Blue" state makes it easier for people to isolate interested groups.

From the graph below, we see that more democratic states have higher total cases - this might because the democratic states have more population. On the other hand, we can see that republican dominated states are having faster growing rate in cases, possibly due to prematured reopening. However, this difference is little.

0.4.5 Does The Spread of Coronavirus Follow A Similar Pattern Across Different States?

We are interested in the growth rate or the doubling rate across different states.

- `Cases` and `New Cases` map the lines which represents the growing pattern of each state;
- `State` is mapped to the colors to help differentiate between them.

Spreading Trajectories of Each State This grpah is motivated by this YouTube video.

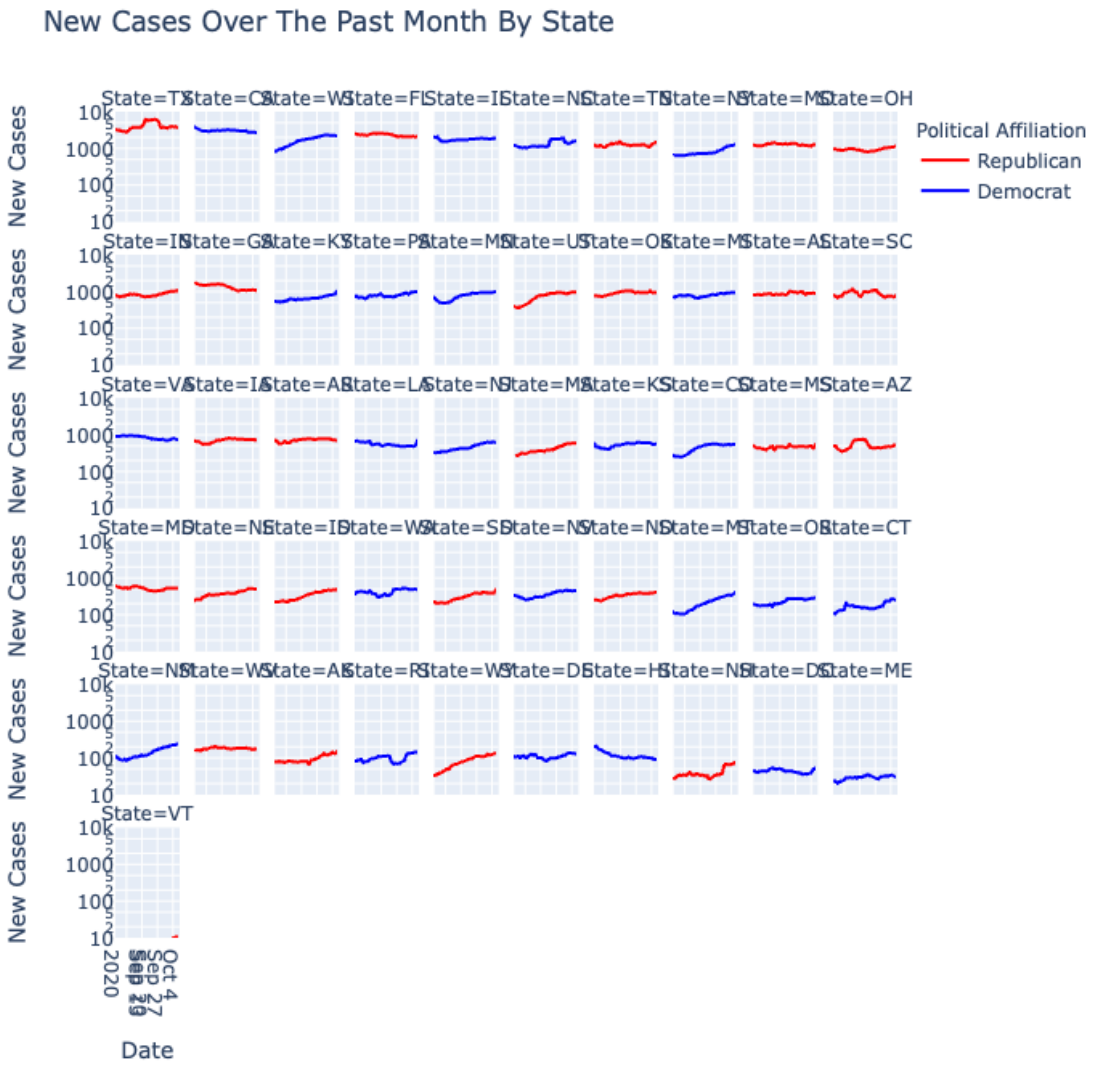
Lines and their colors are associative.

We can see that all states have been following a similar pattern in the spreading of coronavirus - all of them roughly follow the same diagonoal line. Though many states are starting to see a drop in new cases, signs of rebound are seen as the new cases are picking up again.

0.4.6 Which States Are Seeing A Slow-Down In Growing Cases In The Past Month?

We are going to look at the new cases and positive rate just for the past 30 days.

- `New Cases`, `Pos Rate` and `Date` are used to map the shape of the line;
- `State` are faceted into subplots; `-Political Affiliation` is mapped to color just to provide some additional insight.



New Cases Over The Past Month By State Lines are associative. Facets are ordered since their positions are determined by `New Cases` and `Pos Rate` from highest to lowest. Color here can be selective as explained in the previous graph (4.4.1).

We can see a quite even distribution between states that are slowing down and having increasing new cases. New cases in GA, HI, CA, etc are decreasing. However, majority of the states are experiencing an increase in new cases in the past 30 days. This behavior does not seem to be related with each state's governor's political affiliation.

Positive Rate Over The Past Month By State Similar as the graph above - some states' positive rate is increaseing, some are decreasing. However, we do see that more Red states than Blue states have higher positive rates in the past 30 days.

0.5 Part 5: Visualizations & Source Code

0.5.1 Set up

[1]:
```
import numpy as np
import pandas as pd
import plotly.graph_objects as go
import plotly.express as px

from datetime import timedelta
```

```
from datetime import datetime

pd.options.display.max_columns = None
pd.options.display.max_rows = None
```

0.5.2 Import Datasets

[2]:
```
# Import data
# https://covidtracking.com/data/api
covid = pd.read_csv('https://api.covidtracking.com/v1/states/daily.csv')
covid.date = pd.to_datetime(covid.date, format='%Y%m%d')

# https://github.com/nytimes/covid-19-data/tree/master/mask-use
mask = pd.read_csv(
    'https://raw.githubusercontent.com/nytimes/covid-19-data/master/mask-use/
 ↪mask-use-by-county.csv',
    dtype={'COUNTYFP': str}
)
# https://github.com/usdigitalresponse/covid-exit-strategy/tree/master/covid/
 ↪data
population = pd.read_csv(
    'https://raw.githubusercontent.com/usdigitalresponse/covid-exit-strategy/
 ↪master/covid/data/population.csv'
)
population.columns = ['full', 'population']

abbr = pd.read_json(
    'https://raw.githubusercontent.com/usdigitalresponse/covid-exit-strategy/
 ↪master/covid/data/us_state_abbreviations.json',
    orient='index'
).reset_index()
abbr.columns = ['state', 'full']

# https://www.kff.org/3fec844/
governor = pd.read_csv('governor.csv')
governor.columns = ['full', 'party']
```

0.5.3 Data Cleaning & Integration

[3]:
```
selected = [
    'date',
    'state',
    'fips',
```

```
    'death',
    'deathIncrease',

    'positive',
    'positiveIncrease',

    'totalTestResults',
    'totalTestResultsIncrease'
]

covid = covid.loc[:, selected]
covid.shape
```

[3]:
```
(12250, 9)
```

[4]:
```
states = covid.state.unique()
dates = covid.date.unique()

temp = np.array([])
for s in states:
    for d in dates:
        temp = np.append(temp, [s, d])

index = temp.reshape(-1, 2)

index = pd.DataFrame(index)
index.columns = ['state', 'date']
index.date = pd.to_datetime(index.date, format='%Y-%m-%d')

cov = pd.merge(covid, index, how='outer', on=['state', 'date']).fillna(0)
index.shape
```

[4]:
```
(14616, 2)
```

[5]:
```
# Add state governor party
state_gov = pd.merge(governor, abbr, on='full')[['state', 'party']]
cov = pd.merge(cov, state_gov, on='state')
```

[6]:
```
# Add state population
state_pop = pd.merge(population, abbr, on='full')[['state', 'population']]
cov = pd.merge(cov, state_pop, on='state')
```

[7]:
```
cov['mortality'] = cov.death / cov.positive
cov['pos_rate'] = cov.positive / cov.totalTestResults
cov['cases/M'] = round(cov.positive / cov.population * 1E6)
cov['tests/M'] = round(cov.totalTestResults / cov.population * 1E6)
cov['deaths/10M'] = round(cov.death / cov.population * 1E7)
```

[8]:
```
cov.head(5)
```

[8]:
```
        date state  fips  death  deathIncrease  positive  positiveIncrease  \
0 2020-10-08    AK   2.0   60.0            1.0    9996.0             135.0
1 2020-10-07    AK   2.0   59.0            1.0    9861.0             274.0
2 2020-10-06    AK   2.0   58.0            0.0    9587.0               0.0
3 2020-10-05    AK   2.0   58.0            0.0    9587.0             211.0
4 2020-10-04    AK   2.0   58.0            0.0    9376.0             189.0

   totalTestResults  totalTestResultsIncrease       party  population  \
0          491171.0                    1097.0  Republican      731545
1          490074.0                   10700.0  Republican      731545
2          479374.0                       0.0  Republican      731545
3          479374.0                    2556.0  Republican      731545
4          476818.0                    3562.0  Republican      731545

   mortality  pos_rate  cases/M   tests/M  deaths/10M
0   0.006002  0.020351  13664.0  671416.0       820.0
1   0.005983  0.020121  13480.0  669916.0       807.0
2   0.006050  0.019999  13105.0  655290.0       793.0
3   0.006050  0.019999  13105.0  655290.0       793.0
4   0.006186  0.019664  12817.0  651796.0       793.0
```

0.5.4 Visualizations

Death Per 10 Million Population Map

[9]:
```
df = cov[cov.date == cov.date.max()]

fig = go.Figure(data=go.Choropleth(
    locations=df.state,  # Spatial coordinates
    z=df['deaths/10M'],  # Data to be color-coded
    locationmode='USA-states',  # set of locations match entries in `locations`
    colorscale='Reds',
    colorbar_title="Death Per 10 Million",
))

fig.update_layout(
    title_text='Death Per 10 Million Population Map',
    geo_scope='usa',  # limite map scope to USA
)

fig.show()
```

COVID Mortality By State

[10]:
```
fig = go.Figure(data=go.Choropleth(
    locations=df.state,  # Spatial coordinates
    z=df.mortality,  # Data to be color-coded
    locationmode='USA-states',  # set of locations match entries in `locations`
    colorscale='Greys',
    colorbar_title="Mortality",
))

fig.update_layout(
    title_text='COVID Mortality By State',
    geo_scope='usa',  # limite map scope to USA
)
fig.show()
```

Testing Map

[11]:
```
df = cov[cov.date == cov.date.max()]


fig = go.Figure(data=go.Choropleth(
    locations=df.state,  # Spatial coordinates
    z=df['totalTestResults'],  # Data to be color-coded
    locationmode='USA-states',  # set of locations match entries in `locations`
    colorscale='Greens',
    colorbar_title="Testing",
))

fig.update_layout(
    title_text='Testing Map',
    geo_scope='usa',  # limite map scope to USA
)

fig.show()
```

Cases v.s Testing Per Million Population By State

[12]:
```
fig = px.line(cov, x="tests/M", y="cases/M", color="state",
              title='Cases v.s Testing Per Million Population By State',
              labels={'tests/M': 'Tests/M',
                      'cases/M': 'Cases/M',
                      'state': 'State'},
              height=600)
fig.show()
```

Top 20 States with Most Cases Over Time

[13]:
```
df = cov.sort_values(['date', 'positive']
                     ).reset_index().groupby('date').tail(20)
df['date_str'] = df.date.apply(lambda x: x.strftime("%Y-%m-%d"))
df = df.drop('index', axis=1).reset_index(drop=True)
```

[14]:
```
fig = px.bar(df, y="state", x="positive", animation_frame="date_str",
             color="state", hover_name="state",
             log_x=True, orientation='h', height=600,
             title='Top 20 States with Most Cases Over Time',
             labels={'positive': 'Cases',
                     'state': 'State',
                    'date_str':'Date'})

fig.update_layout(yaxis={'categoryorder': 'total ascending'},
                  transition={'duration': 4000,
                              "easing": "quad-in-out"})
fig.show()
```

Cases Trajectories of Each State - From The First Day When 100+ Cases Were Reported

[15]:
```
df = cov[cov.positive >= 100].groupby(by='state').min()
df = df.reset_index().loc[:, ['state', 'date']].rename(
    columns={"date": "day_one"})

df = pd.merge(cov, df, on='state')
df['n_days'] = (df.date - df.day_one)/np.timedelta64(1, 'D')
df = df[df.n_days > 0]
```

[16]:
```
fig = px.line(df, x="n_days", y="positive", color="state",
              title='Cases Trajectories of Each State - From The First Day When␣
 ↪100+ Cases Were Reported',
              labels={'n_days': 'Days',
                      'positive': 'Cases',
                      'state': 'State'},
              log_y=True, height=600)
fig.show()
```

Cases Trajectories of Each State By Governor's Political Affiliation

[17]:
```
df = df.sort_values(['party', 'state', 'n_days']).reset_index(drop=True)
fig = px.line(df, x="n_days", y="positive", line_group='state',
              facet_col="party", color="party",
              title='Cases Trajectories of Each State By Governor\'s Political␣
 ↪Affiliation',
              labels={'n_days': 'Days',
```

```
                        'positive': 'Cases',
                        'state': 'State',
                        'party': 'Political Affiliation'},
              log_y=True, height=600,
              color_discrete_map={
                  "Republican": "red",
                  "Democrat": "blue"
              })
fig.update_layout(legend={'xanchor': 'center',
                          'yanchor': 'bottom'})
fig.show()
```

Spreading Trajectories of Each State

[18]:
```
df = pd.DataFrame(cov)
df = df.sort_values(by=['state', 'date']).reset_index()
df['avg_cases'] = df.groupby('state').rolling(
    window=14).positiveIncrease.mean().reset_index(drop=True)
#df['avg_deaths'] = df.groupby('state').rolling(window=7).deathIncrease.mean().
 ↪reset_index(drop=True)

df = df[(df.avg_cases >= 1) & (df.positive >= 100)]
```

[19]:
```
fig = px.line(df, x="positive", y="avg_cases", color="state",
              title='Spreading Trajectories of Each State',
              labels={'avg_cases': 'New Cases (14-day Average)',
                      'positive': 'Cases',
                      'state': 'State'},
              log_x=True, log_y=True, height=600)
fig.show()
```

New Cases Over The Past Month By State

[20]:
```
df = pd.DataFrame(cov)
df = df.sort_values(by=['state', 'date']).reset_index()
df['avg_cases'] = df.groupby('state').rolling(
    window=7).positiveIncrease.mean().reset_index(drop=True)
#df['avg_deaths'] = df.groupby('state').rolling(window=7).deathIncrease.mean().
 ↪reset_index(drop=True)

df = df[(df.avg_cases >= 1) & (df.positive >= 100)]
df = df[df.date > df.date.max()-timedelta(days=30)]
df = df.sort_values(['date', 'avg_cases'],
                    ascending=False).reset_index(drop=True)
```

[21]:
```
fig = px.line(df, x="date", y="avg_cases", color='party', facet_col="state",␣
↪facet_col_wrap=10,
              title='New Cases Over The Past Month By State',
              labels={'avg_cases': 'New Cases',
                      'date': 'Date',
                      'state': 'State',
                      'party': 'Political Affiliation'},
              log_y=True, height=700, facet_row_spacing=0.04, range_y=[10,␣
↪10000],
              color_discrete_map={
                  "Republican": "red",
                  "Democrat": "blue"
              })
fig.show()
```

Positive Rate Over The Past Month By State

[22]:
```
df = pd.DataFrame(cov)
df = df[(df.positive >= 100)]
df = df[df.date > df.date.max()-timedelta(days=30)]
df = df.sort_values(['date', 'pos_rate'],
                    ascending=False).reset_index(drop=True)
```

[23]:
```
fig = px.line(df, x="date", y="pos_rate", color='party', facet_col="state",␣
↪facet_col_wrap=10,
              title='Positive Rate Over The Past Month By State',
              labels={'pos_rate': 'Pos Rate',
                      'date': 'Date',
                      'state': 'State',
                      'party': 'Political Affiliation'},
              height=700, facet_row_spacing=0.04,
              color_discrete_map={
                  "Republican": "red",
                  "Democrat": "blue"
              })
fig.show()
```