In [268…

```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from sklearn import datasets, linear_model, metrics
import seaborn as sns
plt.style.use('seaborn')
%matplotlib notebook
import statsmodels.api as sm
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
from datetime import datetime
from statsmodels.formula.api import ols
```

In [269…

```
#This code reads in the dataset
df = pd.read_csv('covid.csv')
```

In [270…

```
#This code is used to identify the datatypes, submission_date is a string object, needs
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46920 entries, 0 to 46919
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   submission_date  46920 non-null  object
 1   state            46920 non-null  object
 2   tot_cases        46920 non-null  int64 
 3   conf_cases       46920 non-null  int64 
 4   prob_cases       46920 non-null  int64 
 5   new_case         46920 non-null  int64 
 6   pnew_case        46920 non-null  int64 
 7   tot_death        46920 non-null  int64 
 8   conf_death       46920 non-null  int64 
 9   prob_death       46920 non-null  int64 
 10  new_death        46920 non-null  int64 
 11  pnew_death       46920 non-null  int64 
 12  created_at       46920 non-null  object
 13  consent_cases    42228 non-null  object
 14  consent_deaths   43792 non-null  object
dtypes: int64(10), object(5)
memory usage: 5.4+ MB
```

In [271…

```
#This code transforms submission_date to datetime.
df['submission_date'] = pd.to_datetime(df['submission_date'])
```

In [272…

```
# This confirms submission is of datetime type
df.info()
```

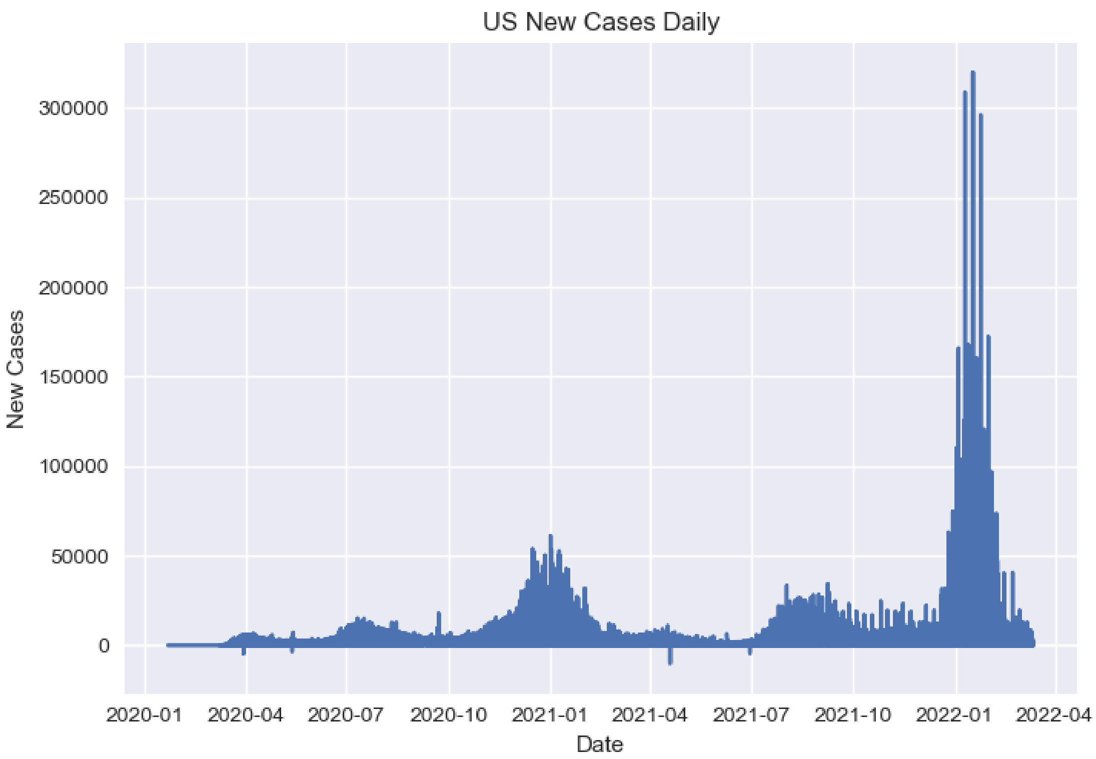
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46920 entries, 0 to 46919
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   submission_date  46920 non-null  datetime64[ns]
 1   state            46920 non-null  object
 2   tot_cases        46920 non-null  int64
 3   conf_cases       46920 non-null  int64
 4   prob_cases       46920 non-null  int64
 5   new_case         46920 non-null  int64
 6   pnew_case        46920 non-null  int64
 7   tot_death        46920 non-null  int64
 8   conf_death       46920 non-null  int64
 9   prob_death       46920 non-null  int64
 10  new_death        46920 non-null  int64
 11  pnew_death       46920 non-null  int64
 12  created_at       46920 non-null  object
 13  consent_cases    42228 non-null  object
 14  consent_deaths   43792 non-null  object
dtypes: datetime64[ns](1), int64(10), object(4)
memory usage: 5.4+ MB
```

In [273…

```
#df is 14 columns, the study doesn't require all the columns. this filters data to a ne
df1 = df.filter(['submission_date','state','tot_cases','new_case'])
```

In [288…

```
#This is a plot of the entire dataset. As one can see, the plot is not smooth at all, t
#to smooth.
plt.plot(df['submission_date'],df['new_case'])
plt.xlabel('Date')
plt.ylabel('New Cases')
plt.title('US New Cases Daily');
```



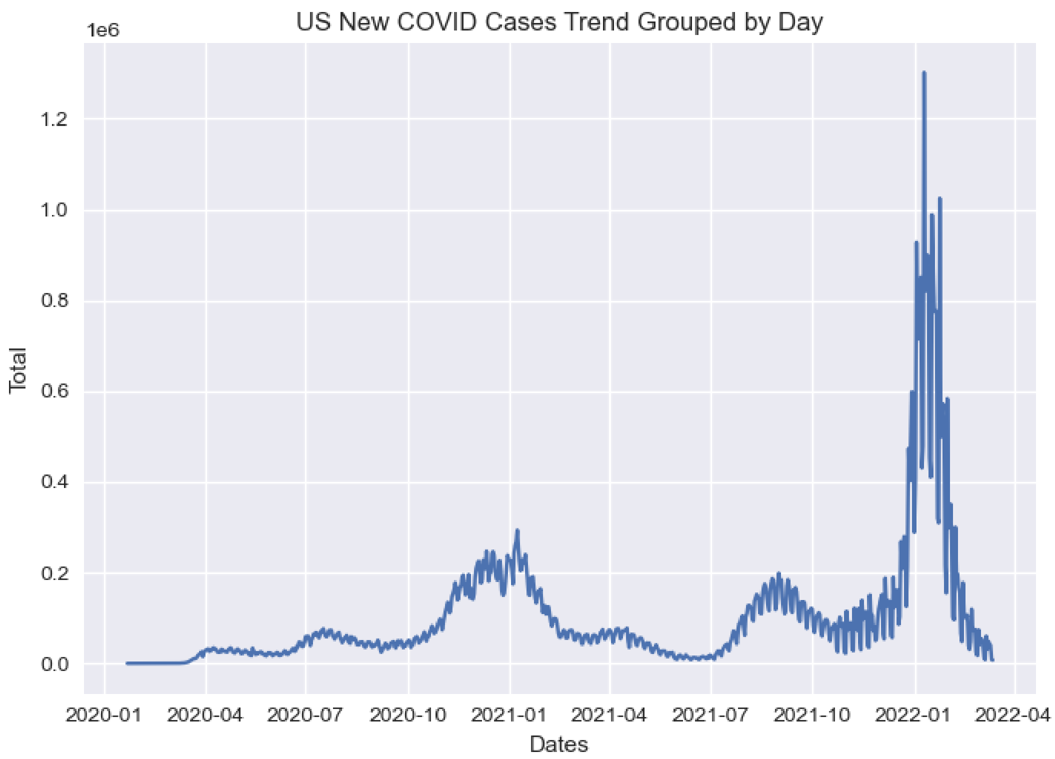
In [289…

```
#This creates a reduced sized dataframe to group by single dates. At present, each of t
#single date. Need to reduce to just a single day, this code groups observations and su
date_trend = df1.groupby('submission_date')['new_case'].sum().reset_index()
date_trend.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 782 entries, 0 to 781
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   submission_date  782 non-null    datetime64[ns]
 1   new_case         782 non-null    int64         
dtypes: datetime64[ns](1), int64(1)
memory usage: 12.3 KB
```

In [291…

```
#This plot is grouped by day. One can see that the data is smoother than plotting each
# as in the previous chart.
plt.plot(date_trend['submission_date'],date_trend['new_case'])
plt.xlabel('Dates')
plt.ylabel('Total')
plt.title('US New COVID Cases Trend Grouped by Day');
```



In [292…

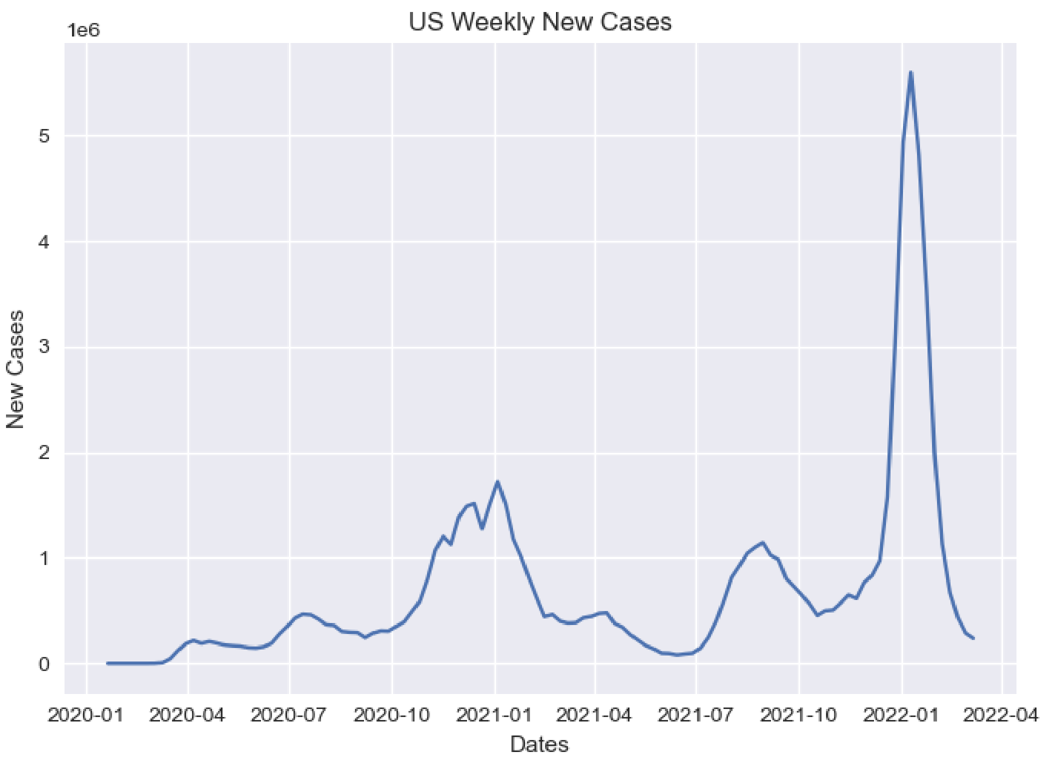
```
df1['week'] = df1['submission_date'] - pd.to_timedelta(arg=df1['submission_date'].dt.we
```

In [325…

```
wk_group = df1.groupby('week')['new_case'].sum().reset_index()
```

In [326…

```
#This chart is new US cases grouped by week, now we're starting to see much more smooth
#observe the trend.
plt.plot(wk_group['week'],wk_group['new_case'])
plt.xlabel('Dates')
plt.ylabel('New Cases')
plt.title('US Weekly New Cases');
```



In [296…

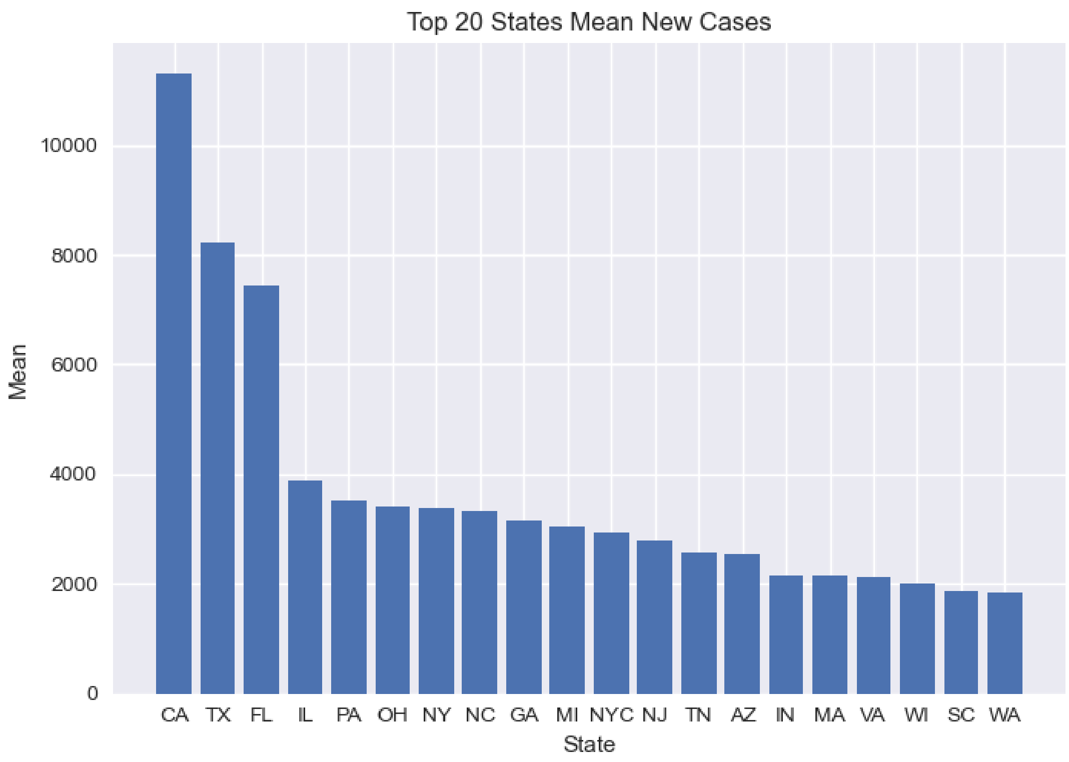
```
new = pd.DataFrame().assign(state = st_sum['state'],mean = st_mean['new_case'],median=s
```

In [298…

```
s_mean = df.groupby('state')[['new_case']].mean().reset_index().nlargest(20,'new_case')
s_median = df.groupby('state')[['new_case']].median().nlargest(20,'new_case').reset_ind
s_sum = df.groupby('state')[['new_case']].sum().nlargest(20,'new_case').reset_index()
```

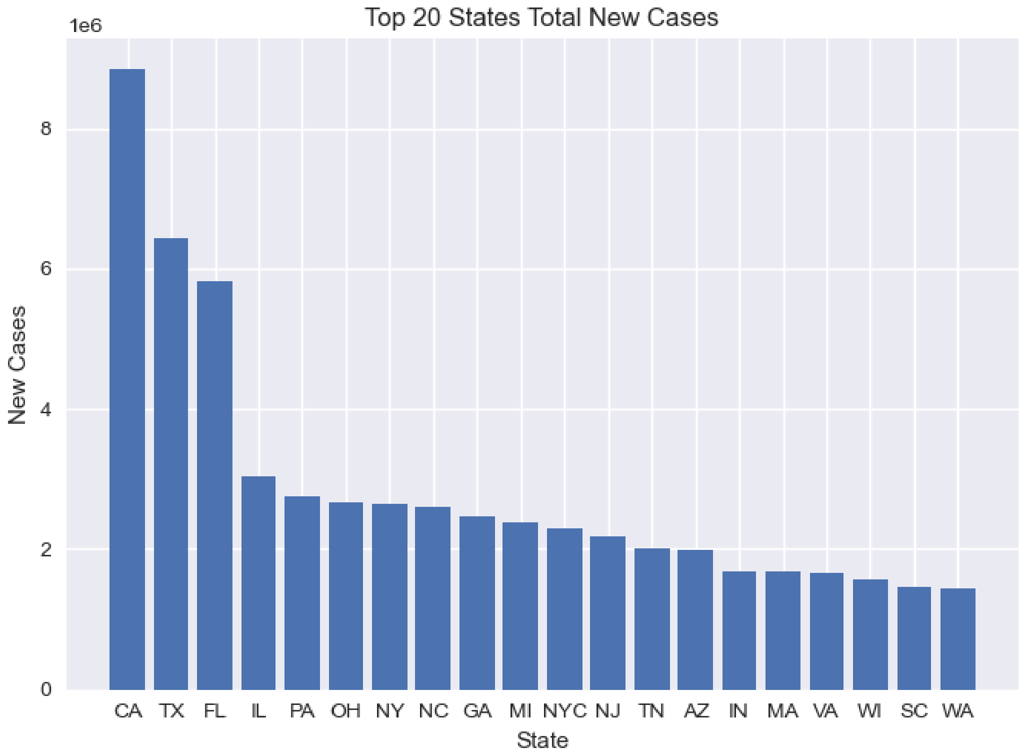
In [301…

```
#This is a bar chart of the top 20 states by average. One can see that the top 3 states
#which are highly populace states. Further research could explore a correlation between
plt.bar(s_mean['state'],s_mean['new_case'])
plt.xlabel('State')
plt.ylabel('Mean')
plt.title('Top 20 States Mean New Cases');
```



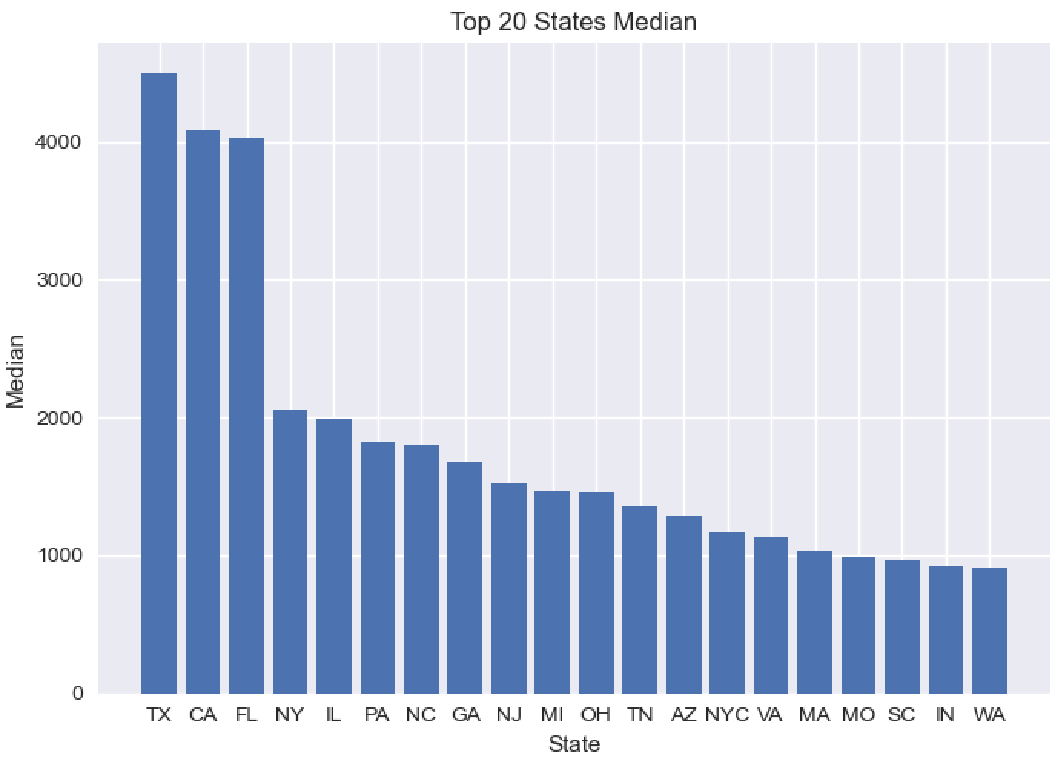
In [304…

```
#This chart is top 20 US states total new cases. Again, we see the highly populated sta
#chart.
plt.bar(s_sum['state'],s_sum['new_case'])
plt.xlabel('State')
plt.ylabel('New Cases')
plt.title('Top 20 States Total New Cases');
```



In [308…

```python
#This chart depicts the US states median new cases. The median of the dataset is much l
#the distribution is left-skewed.
plt.bar(s_median['state'],s_median['new_case'])
plt.xlabel('State')
plt.ylabel('Median')
plt.title('Top 20 States Median');
```



In [327…

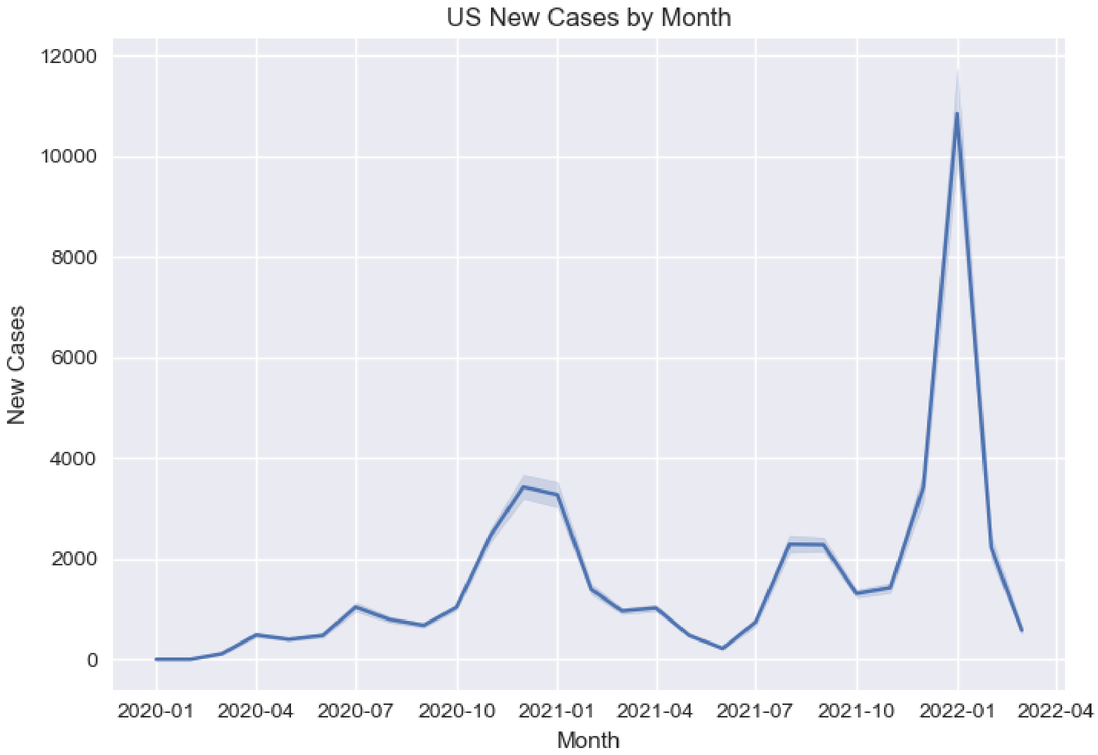
```
#This code creates the column "month" that will be used for the monthly trend chart.
df1['month'] = df1['submission_date'] + pd.offsets.MonthEnd(0) - pd.offsets.MonthBegin(
df1.head()
```

Out[327…

	submission_date	state	tot_cases	new_case	week	month
0	2022-03-13	WV	494875	0	2022-03-07	2022-03-01
1	2022-03-13	NYC	2287411	1900	2022-03-07	2022-03-01
2	2022-03-13	CO	1325063	0	2022-03-07	2022-03-01
3	2022-03-13	IA	756778	0	2022-03-07	2022-03-01
4	2022-03-13	RMI	4	0	2022-03-07	2022-03-01

In [314…

```
#The chart beow is a depiction of US new cases by month. With this visual, we're really
#dataset.
sns.lineplot(x = df1['month'],y=df1['new_case'])
plt.xlabel('Month')
plt.ylabel('New Cases')
plt.title('US New Cases by Month');
```



In [315…

```
#This code creates the dataframe that will become the top 3 US new cases.
top_3 = df1.groupby(['month','state'])['new_case'].sum().reset_index()
```

In [316…

```
top3 = top_3.nlargest(3,'new_case')
toplist = list(top3['state'])
```

In [320…

```
top = top_3[top_3['state'].isin(toplist)]
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20 entries, 5 to 56
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   state   20 non-null     object 
 1   mean    20 non-null     float64
 2   median  20 non-null     float64
 3   tot     20 non-null     int64  
dtypes: float64(2), int64(1), object(1)
memory usage: 800.0+ bytes
```

In [343…

```
#t = df1.groupby(['month','state'])['new_case'].sum().reset_index()
#t20 = t[t['state'].isin(toplist20)]
t20 = t20.groupby('month')[['new_case']].sum().reset_index()
t20.head()
```

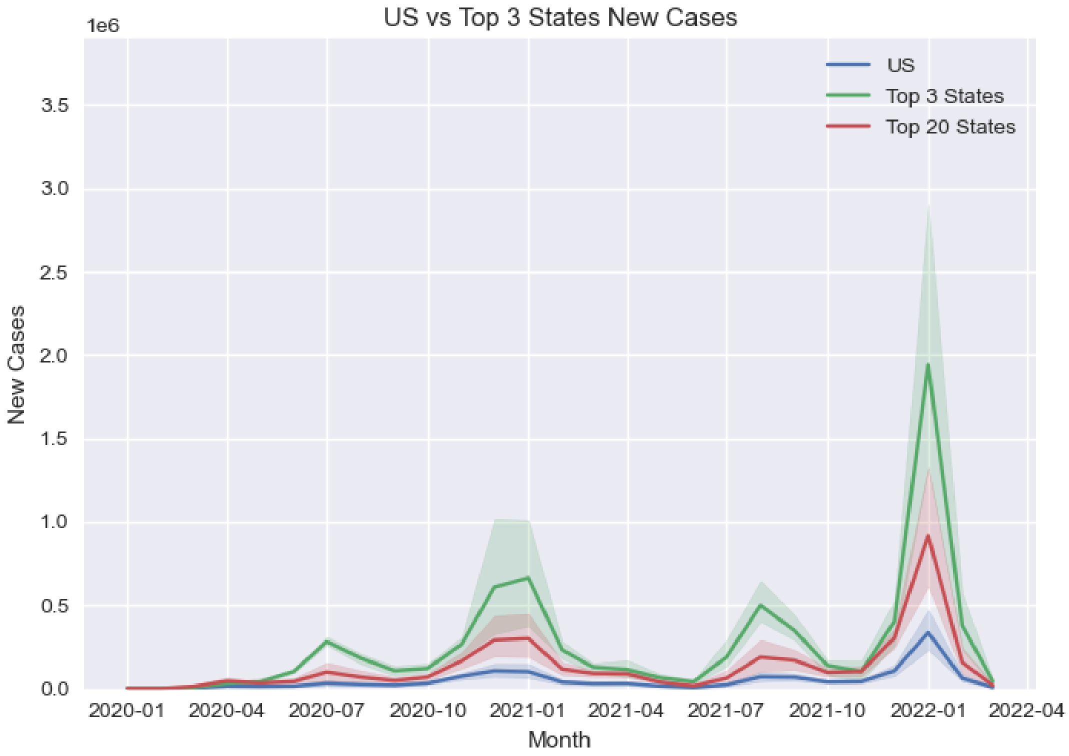
Out[343…

	month	new_case
0	2020-01-01	3

	month	new_case
1	2020-02-01	36
2	2020-03-01	161898
3	2020-04-01	595369
4	2020-05-01	431893

In [353…

```
#This chart is a visual comparison of the US vs Top 3 States vs Top 20 states new cases
#a large number of new cases in this time period.
sns.lineplot(x = top_3['month'],y=top_3['new_case'],label = 'US')
sns.lineplot(x = top['month'],y=top['new_case'],label='Top 3 States')
sns.lineplot(x = top_20['month'],y=top_20['new_case'],label='Top 20 States')
plt.ylabel('New Cases')
plt.xlabel('Month')
plt.title('US vs Top 3 States New Cases')
plt.ylim(0,3900000)
plt.legend();
```



In [123…

```
top20 = top_3.nlargest(20,'new_case')
toplist20 = list(top20['state'])
```

In [350…

```
top_20 = top_3[top_3['state'].isin(toplist20)]
```

In [354…

```
#These next several blocks of code create the rolling average dataframe for the visual
```

In [160…

```
sma = df1
sma.head()
```

Out[160…

	submission_date	state	tot_cases	new_case	month	weekly_rolling
0	2022-03-13	WV	494875	0	2022-03-01	NaN
1	2022-03-13	NYC	2287411	1900	2022-03-01	NaN
2	2022-03-13	CO	1325063	0	2022-03-01	NaN
3	2022-03-13	IA	756778	0	2022-03-01	NaN
4	2022-03-13	RMI	4	0	2022-03-01	NaN

In [161…

```
sma['weekly_rolling'] = sma.new_case.rolling(30).mean()
```

In [162…

```
sma = sma.dropna()
sma.head()
```

Out[162…

	submission_date	state	tot_cases	new_case	month	weekly_rolling
29	2022-03-13	VT	105475	0	2022-03-01	70.966667
30	2022-03-13	MI	2371788	0	2022-03-01	70.966667
31	2022-03-13	MO	1403268	0	2022-03-01	7.633333
32	2022-03-13	WA	1437914	0	2022-03-01	7.633333
33	2022-03-13	VI	15556	0	2022-03-01	7.633333

In [169…

```
sma1 = df1.groupby('submission_date')['new_case'].sum().reset_index()
sma1.head()
```

Out[169…

	submission_date	new_case
0	2020-01-22	0
1	2020-01-23	1
2	2020-01-24	1
3	2020-01-25	0
4	2020-01-26	1

In [170…

```
sma1['7-day'] = sma1.new_case.rolling(7).mean()
sma1.head()
```

Out[170…

	submission_date	new_case	7-day
0	2020-01-22	0	NaN

	submission_date	new_case	7-day
1	2020-01-23	1	NaN
2	2020-01-24	1	NaN
3	2020-01-25	0	NaN
4	2020-01-26	1	NaN

In [171…

```python
sma1 = sma1.dropna()
sma1.head()
```

Out[171…

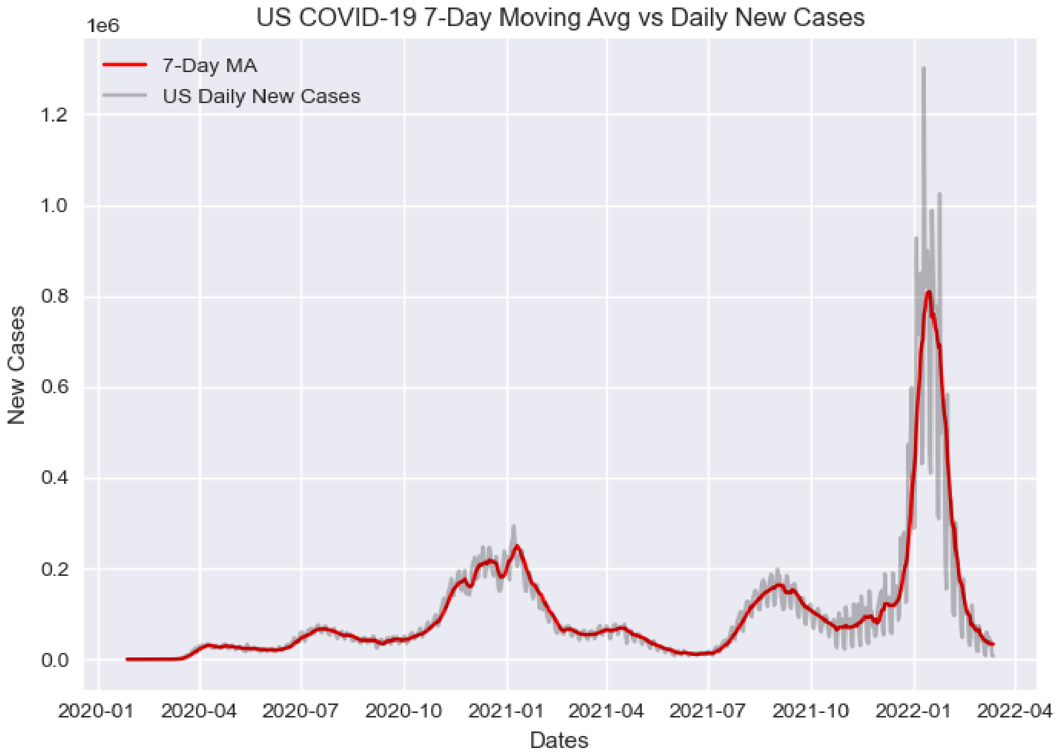
	submission_date	new_case	7-day
6	2020-01-28	0	0.428571
7	2020-01-29	0	0.428571
8	2020-01-30	0	0.285714
9	2020-01-31	1	0.285714
10	2020-02-01	1	0.428571

In [183…

```python
#The below graph is a depiction of the seven day moving average of new cases of COVID-1
#Jan 20 - Mar 22 in the USA. The new cases were grouped by day and summed. The chart sh
#grey, with the smoothed 7-day rolling average in red. The COVID-19 trend in the US is 
#The viewer can see that there were very few new cases in Jan 2020, then we can see the
#around the time that summer fun was ending. Fall to winter of 2020 saw a triple top pe
#many Americans were through with the lockdowns and chose enjoying the Holidays over sa
#Spring 2021 when the vaccine became available, but another huge peak in Sep-Oct 2021 w
#his was followed by a drop-off, then record setting new cases with the onset of the Om
#drop off as many citizens got first time and booster vaccines.
```

In [182…

```python
plt.plot(sma1['submission_date'],sma1['7-day'],label='7-Day MA',color='red')
plt.plot(sma1['submission_date'],sma1['new_case'],label='US Daily New Cases',alpha=0.25
plt.ylabel('New Cases')
plt.xlabel('Dates')
plt.title('US COVID-19 7-Day Moving Avg vs Daily New Cases')
plt.legend();
```


US COVID-19 7-Day Moving Avg vs Daily New Cases
1e6
7-Day MA
US Daily New Cases
1.2
1.0
0.8
0.6
0.4
0.2
0.0
New Cases
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
2021-10
2022-01
2022-04
Dates

```
In [ ]:
```