Stage2_Task2_Cases-

March 14, 2023

0.1 Compare the data against 3 other states. Normalize by population, use a normalization factor which is able to identify cases, for example try per 10,000 or 100,000 (this depends on the population). Plot the values across the weeks in a line plot for the 3 states in a single graph. Describe why the rates differ across these states in the notebook. Identify the peaks, are they consistent with the US pattern?

[1]:
```
import pandas as pd
import numpy as np
import statistics
import matplotlib.pyplot as plt
```

[2]:
```
#I have selected the Albama,Newjersey,Florida and NorthCarolina states for␣
 ↪analysis.
selected_state_AL = "AL"
selected_state_NJ = "NJ"
selected_state_MD = "MD"
selected_state_NC = "NC"
#reading the confirmed data
cases = pd.read_csv("../data/covid_confirmed_usafacts.csv")
cases.head()
```

[2]:
```
   countyFIPS            County Name State  StateFIPS  2020-01-22  2020-01-23  \
0           0  Statewide Unallocated    AL          1           0           0
1        1001         Autauga County    AL          1           0           0
2        1003         Baldwin County    AL          1           0           0
3        1005         Barbour County    AL          1           0           0
4        1007            Bibb County    AL          1           0           0

   2020-01-24  2020-01-25  2020-01-26  2020-01-27  …  2023-01-07  \
0           0           0           0           0  …           0
1           0           0           0           0  …       19205
2           0           0           0           0  …       68182
3           0           0           0           0  …        7120
4           0           0           0           0  …        7808

   2023-01-08  2023-01-09  2023-01-10  2023-01-11  2023-01-12  2023-01-13  \
0           0           0           0           0           0           0
```

```
1       19205       19205       19205       19205       19318       19318
2       68182       68182       68182       68182       68518       68518
3        7120        7120        7120        7120        7188        7188
4        7808        7808        7808        7808        7855        7855

   2023-01-14  2023-01-15  2023-01-16
0           0           0           0
1       19318       19318       19318
2       68518       68518       68518
3        7188        7188        7188
4        7855        7855        7855

[5 rows x 1095 columns]
```

[3]:
```
# using the melt function so that we get the all the dates in one column.
cases_transpose = pd.melt(frame = cases, id_vars=('countyFIPS','County␣
 ↪Name','State','StateFIPS'),var_name=["Date"],value_name='Number of Cases')
cases_transpose = cases_transpose[cases_transpose['countyFIPS'] != 0]
cases_transpose.head()
```

[3]:
```
   countyFIPS      County Name State  StateFIPS        Date  Number of Cases
1        1001   Autauga County    AL          1  2020-01-22                0
2        1003   Baldwin County    AL          1  2020-01-22                0
3        1005   Barbour County    AL          1  2020-01-22                0
4        1007      Bibb County    AL          1  2020-01-22                0
5        1009    Blount County    AL          1  2020-01-22                0
```

[4]:
```
#Filtering the data for Albama State.
cases_selected_state = cases_transpose[cases_transpose["State"] ==␣
 ↪selected_state_AL]
cases_selected_state.head()
```

[4]:
```
   countyFIPS      County Name State  StateFIPS        Date  Number of Cases
1        1001   Autauga County    AL          1  2020-01-22                0
2        1003   Baldwin County    AL          1  2020-01-22                0
3        1005   Barbour County    AL          1  2020-01-22                0
4        1007      Bibb County    AL          1  2020-01-22                0
5        1009    Blount County    AL          1  2020-01-22                0
```

[5]:
```
#For the selected state Albama summing the cases per day of all the counties.
cases_selected_state_daily = cases_selected_state.groupby('Date')['Number of␣
 ↪Cases'].sum()
cases_selected_state_daily.head()
```

[5]:
```
Date
2020-01-22    0
2020-01-23    0
```

```
2020-01-24    0
2020-01-25    0
2020-01-26    0
Name: Number of Cases, dtype: int64
```

[6]:
```
#Finding out the new cases per day.
new_cases_selected_state_daily = cases_selected_state_daily.diff().reset_index()
new_cases_selected_state_daily.head()
```

[6]:
```
         Date  Number of Cases
0  2020-01-22              NaN
1  2020-01-23              0.0
2  2020-01-24              0.0
3  2020-01-25              0.0
4  2020-01-26              0.0
```

[7]:
```
#Converting the daily to weekly analysis and finding the mean weekly.
weekly_cases_mean_selected_state = new_cases_selected_state_daily.copy()
weekly_cases_mean_selected_state['Date'] = pd.
 ↪to_datetime(weekly_cases_mean_selected_state['Date']) - pd.to_timedelta(7,␣
 ↪unit='d')
weekly_cases_mean_selected_state = weekly_cases_mean_selected_state.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Cases'].mean()
weekly_cases_mean_selected_state = weekly_cases_mean_selected_state.
 ↪reset_index()
weekly_cases_mean_selected_state.head()
```

[7]:
```
        Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

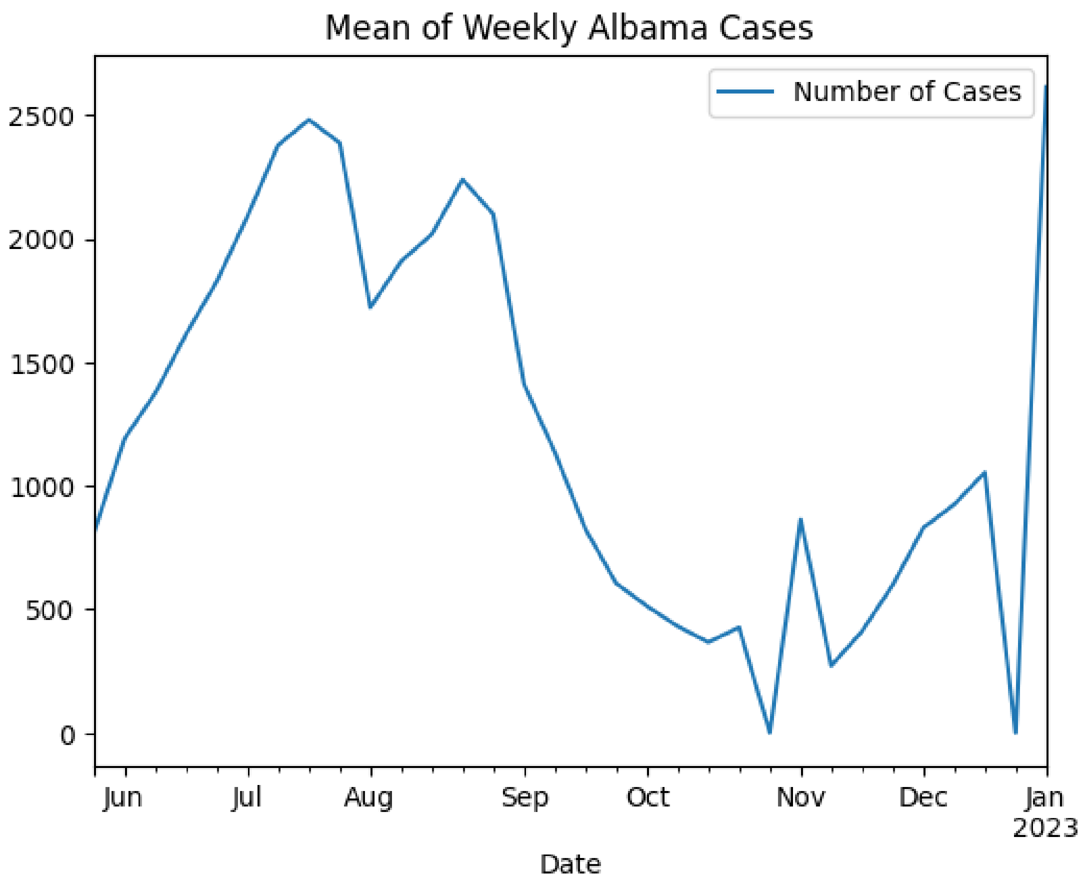
[8]:
```
#Adding one day so that we get weekly analysis from monday to sunday and weekly␣
 ↪analsis from monday to sunday.
weekly_cases_mean_selected_state_given_range =␣
 ↪weekly_cases_mean_selected_state[(weekly_cases_mean_selected_state["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mean_selected_state["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mean_selected_state_given_range =␣
 ↪weekly_cases_mean_selected_state_given_range.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_mean_selected_state_given_range['Date'] =␣
 ↪weekly_cases_mean_selected_state_given_range['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
```

```
weekly_cases_mean_selected_state_given_range
```

[8]:
```
         Date  Number of Cases
0  2022-05-30       806.857143
1  2022-06-06      1190.285714
2  2022-06-13      1374.285714
3  2022-06-20      1612.428571
4  2022-06-27      1826.142857
5  2022-07-04      2090.428571
6  2022-07-11      2378.428571
7  2022-07-18      2480.857143
8  2022-07-25      2387.428571
9  2022-08-01      1721.000000
10 2022-08-08      1907.714286
11 2022-08-15      2017.285714
12 2022-08-22      2238.857143
13 2022-08-29      2099.285714
14 2022-09-05      1411.428571
15 2022-09-12      1136.285714
16 2022-09-19       824.285714
17 2022-09-26       604.428571
18 2022-10-03       512.714286
19 2022-10-10       430.714286
20 2022-10-17       366.571429
21 2022-10-24       426.000000
22 2022-10-31         0.000000
23 2022-11-07       863.142857
24 2022-11-14       271.142857
25 2022-11-21       410.285714
26 2022-11-28       598.000000
27 2022-12-05       829.571429
28 2022-12-12       923.571429
29 2022-12-19      1053.857143
30 2022-12-26         0.000000
31 2023-01-02      2612.857143
```

[9]:
```
#Plotting the mean graph for albama state.
weekly_cases_mean_selected_state_given_range.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Mean of Weekly Albama Cases')
```

[9]:
```
<AxesSubplot: title={'center': 'Mean of Weekly Albama Cases'}, xlabel='Date'>
```



```
[10]: #Converting the daily to weekly analysis and finding the median weekly.
      weekly_cases_median_selected_state = new_cases_selected_state_daily.copy()
      weekly_cases_median_selected_state['Date'] = pd.
       ↪to_datetime(weekly_cases_median_selected_state['Date']) - pd.to_timedelta(7,␣
       ↪unit='d')
      weekly_cases_median_selected_state = weekly_cases_median_selected_state.
       ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].median()
      weekly_cases_median_selected_state = weekly_cases_median_selected_state.
       ↪reset_index()
      weekly_cases_median_selected_state.head()
```

```
[10]:         Date  Number of Cases
      0 2020-01-19              0.0
      1 2020-01-26              0.0
      2 2020-02-02              0.0
      3 2020-02-09              0.0
      4 2020-02-16              0.0
```

[11]:

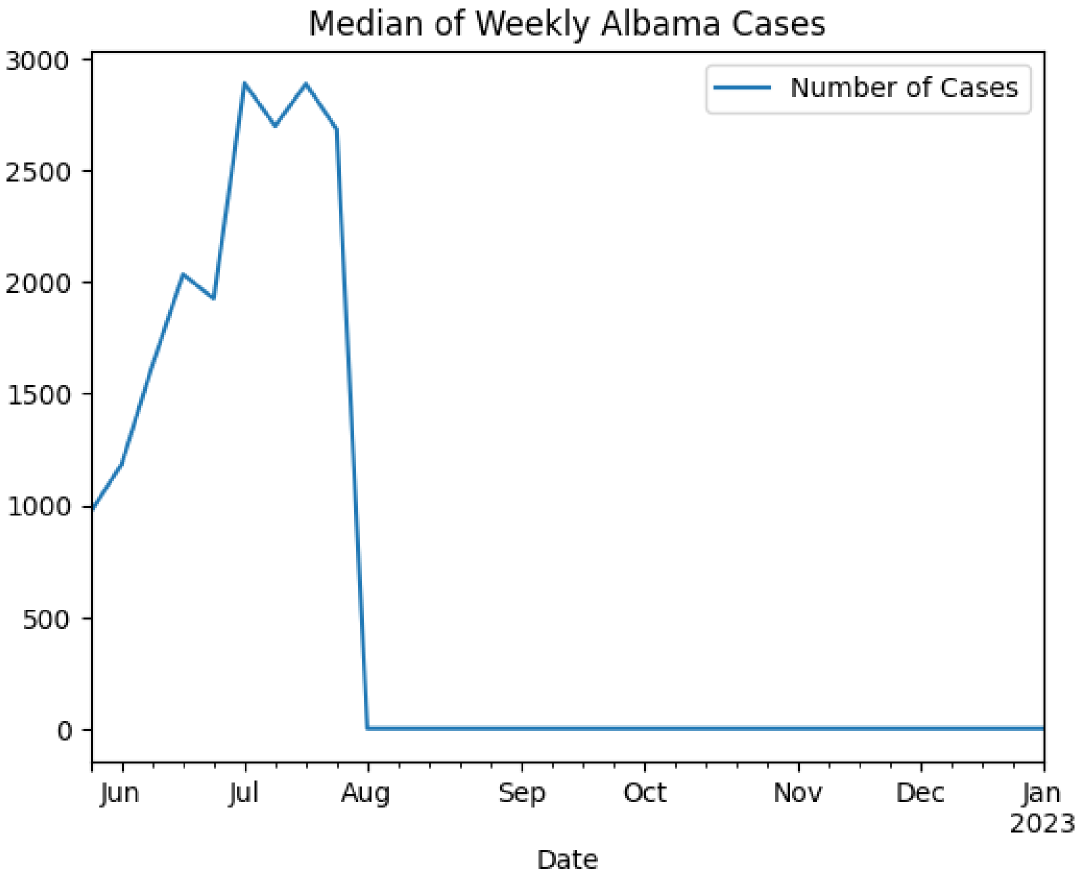
```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_median_selected_state_given_range =␣
 ↪weekly_cases_median_selected_state[(weekly_cases_median_selected_state["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_median_selected_state["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_median_selected_state_given_range =␣
 ↪weekly_cases_median_selected_state_given_range.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_median_selected_state_given_range['Date'] =␣
 ↪weekly_cases_median_selected_state_given_range['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_median_selected_state_given_range
```

[11]:

```
         Date  Number of Cases
0  2022-05-30            971.0
1  2022-06-06           1183.0
2  2022-06-13           1625.0
3  2022-06-20           2035.0
4  2022-06-27           1926.0
5  2022-07-04           2892.0
6  2022-07-11           2699.0
7  2022-07-18           2889.0
8  2022-07-25           2685.0
9  2022-08-01              0.0
10 2022-08-08              0.0
11 2022-08-15              0.0
12 2022-08-22              0.0
13 2022-08-29              0.0
14 2022-09-05              0.0
15 2022-09-12              0.0
16 2022-09-19              0.0
17 2022-09-26              0.0
18 2022-10-03              0.0
19 2022-10-10              0.0
20 2022-10-17              0.0
21 2022-10-24              0.0
22 2022-10-31              0.0
23 2022-11-07              0.0
24 2022-11-14              0.0
25 2022-11-21              0.0
26 2022-11-28              0.0
27 2022-12-05              0.0
28 2022-12-12              0.0
29 2022-12-19              0.0
30 2022-12-26              0.0
31 2023-01-02              0.0
```

[12]:
```
#Plotting the median graph
weekly_cases_median_selected_state_given_range.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Median of Weekly Albama Cases')
```

[12]:
```
<AxesSubplot: title={'center': 'Median of Weekly Albama Cases'}, xlabel='Date'>
```



[13]:
```
#Converting the daily to weekly analysis and finding the mode weekly.
weekly_cases_mode_selected_state = new_cases_selected_state_daily.copy()
weekly_cases_mode_selected_state['Date'] = pd.
 ↪to_datetime(weekly_cases_mode_selected_state['Date']) - pd.to_timedelta(7,␣
 ↪unit='d')
weekly_cases_mode_selected_state = weekly_cases_mode_selected_state.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Cases'].apply(statistics.mode)
weekly_cases_mode_selected_state = weekly_cases_mode_selected_state.
 ↪reset_index()
weekly_cases_mode_selected_state.head()
```

[13]:
```
        Date  Number of Cases
0 2020-01-19              0.0
```

```
1 2020-01-26             0.0
2 2020-02-02             0.0
3 2020-02-09             0.0
4 2020-02-16             0.0
```

[14]:

```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_mode_selected_state_given_range =␣
 ↪weekly_cases_mode_selected_state[(weekly_cases_mode_selected_state["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mode_selected_state["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mode_selected_state_given_range =␣
 ↪weekly_cases_mode_selected_state_given_range.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_mode_selected_state_given_range['Date'] =␣
 ↪weekly_cases_mode_selected_state_given_range['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_mode_selected_state_given_range
```

[14]:

```
         Date  Number of Cases
0  2022-05-30              0.0
1  2022-06-06              0.0
2  2022-06-13              0.0
3  2022-06-20              0.0
4  2022-06-27              0.0
5  2022-07-04              0.0
6  2022-07-11              0.0
7  2022-07-18              0.0
8  2022-07-25              0.0
9  2022-08-01              0.0
10 2022-08-08              0.0
11 2022-08-15              0.0
12 2022-08-22              0.0
13 2022-08-29              0.0
14 2022-09-05              0.0
15 2022-09-12              0.0
16 2022-09-19              0.0
17 2022-09-26              0.0
18 2022-10-03              0.0
19 2022-10-10              0.0
20 2022-10-17              0.0
21 2022-10-24              0.0
22 2022-10-31              0.0
23 2022-11-07              0.0
24 2022-11-14              0.0
25 2022-11-21              0.0
26 2022-11-28              0.0
```

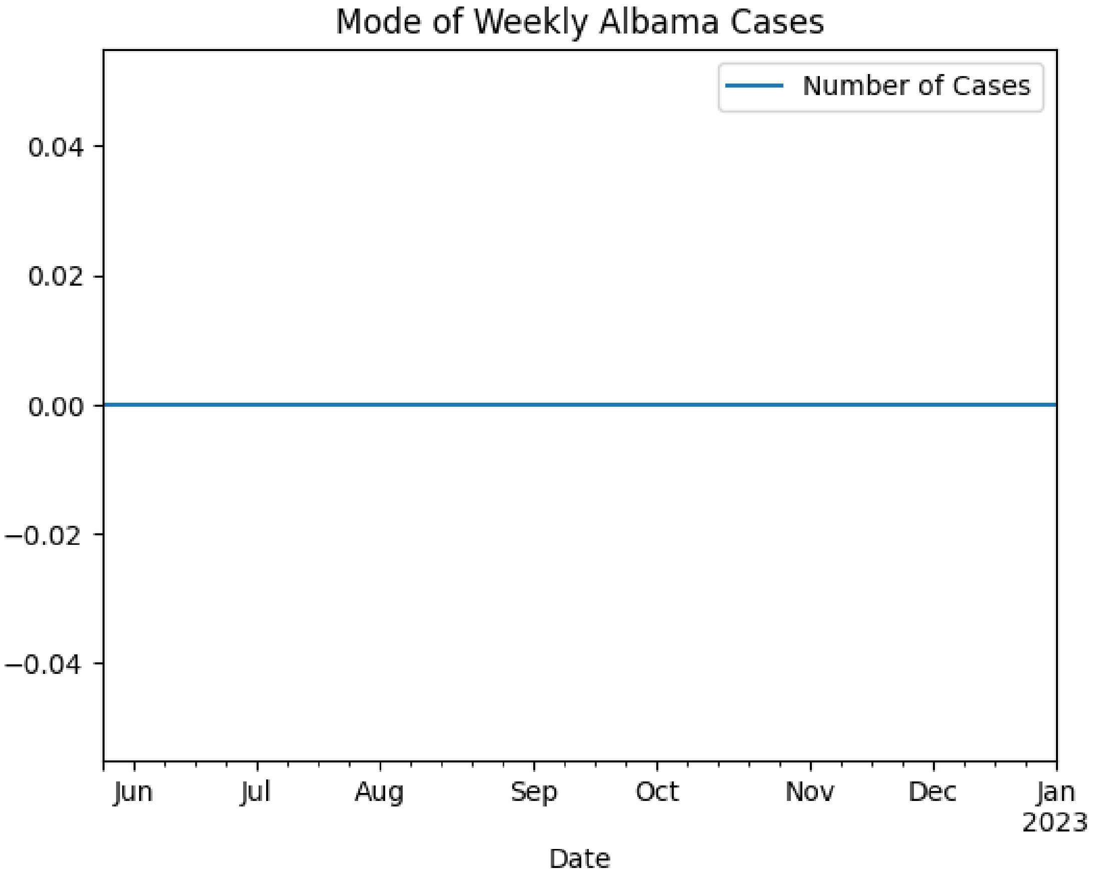
```
27 2022-12-05             0.0
28 2022-12-12             0.0
29 2022-12-19             0.0
30 2022-12-26             0.0
31 2023-01-02             0.0
```

[15]:

```
#Plottinf the mode graph
weekly_cases_mode_selected_state_given_range.plot(x='Date', y='Number of␣
↪Cases', title = 'Mode of Weekly Albama Cases')
```

[15]:

```
<AxesSubplot: title={'center': 'Mode of Weekly Albama Cases'}, xlabel='Date'>
```



[16]:

```
#Converting the daily to weekly analysis and finding the weekly sum of cases.
weekly_cases_sum_selected_state = new_cases_selected_state_daily.copy()
weekly_cases_sum_selected_state['Date'] = pd.
↪to_datetime(weekly_cases_sum_selected_state['Date']) - pd.to_timedelta(7,␣
↪unit='d')
weekly_cases_sum_selected_state = weekly_cases_sum_selected_state.groupby([pd.
↪Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
```

```
weekly_cases_sum_selected_state = weekly_cases_sum_selected_state.reset_index()
weekly_cases_sum_selected_state.head()
```

[16]:

```
         Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

[17]:

```
#Adding one day so that we get weekly analysis from monday to sunday and weekly␣
 ↪analsis from monday to sunday.
weekly_cases_sum_selected_state_given_range =␣
 ↪weekly_cases_sum_selected_state[(weekly_cases_sum_selected_state["Date"] >=␣
 ↪'2022-05-29') & (weekly_cases_sum_selected_state["Date"] <= '2023-01-02')]
weekly_cases_sum_selected_state_given_range =␣
 ↪weekly_cases_sum_selected_state_given_range.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_sum_selected_state_given_range['Date'] =␣
 ↪weekly_cases_sum_selected_state_given_range['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_sum_selected_state_given_range
```

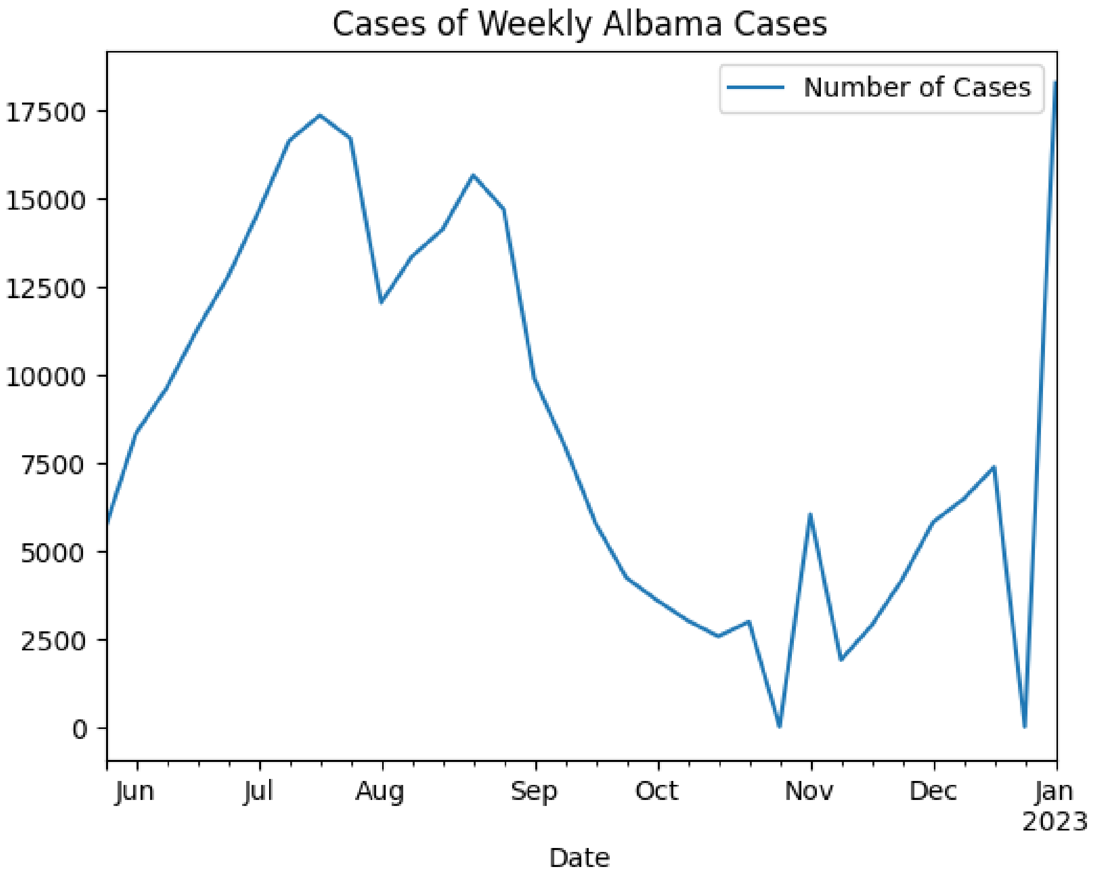
[17]:

```
          Date  Number of Cases
0   2022-05-30           5648.0
1   2022-06-06           8332.0
2   2022-06-13           9620.0
3   2022-06-20          11287.0
4   2022-06-27          12783.0
5   2022-07-04          14633.0
6   2022-07-11          16649.0
7   2022-07-18          17366.0
8   2022-07-25          16712.0
9   2022-08-01          12047.0
10  2022-08-08          13354.0
11  2022-08-15          14121.0
12  2022-08-22          15672.0
13  2022-08-29          14695.0
14  2022-09-05           9880.0
15  2022-09-12           7954.0
16  2022-09-19           5770.0
17  2022-09-26           4231.0
18  2022-10-03           3589.0
19  2022-10-10           3015.0
20  2022-10-17           2566.0
21  2022-10-24           2982.0
22  2022-10-31              0.0
```

```
23 2022-11-07           6042.0
24 2022-11-14           1898.0
25 2022-11-21           2872.0
26 2022-11-28           4186.0
27 2022-12-05           5807.0
28 2022-12-12           6465.0
29 2022-12-19           7377.0
30 2022-12-26              0.0
31 2023-01-02          18290.0
```

[18]:
```
weekly_cases_sum_selected_state_given_range.plot(x='Date', y='Number of Cases',␣
 ↪title = 'Cases of Weekly Albama Cases')
```

[18]:
```
<AxesSubplot: title={'center': 'Cases of Weekly Albama Cases'}, xlabel='Date'>
```



0.1.1 Week starting 2022-07-18 has a peak of cases in alabama state. As 4th of july was independence day so the cases got increase for the month of july.

[19]:

```
#reading the population data
population = pd.read_csv("../data/covid_county_population_usafacts.csv")
population.head()
```

[19]:

```
   countyFIPS            County Name State  population
0           0  Statewide Unallocated    AL           0
1        1001         Autauga County    AL       55869
2        1003         Baldwin County    AL      223234
3        1005         Barbour County    AL       24686
4        1007            Bibb County    AL       22394
```

[20]:

```
#Filtering the population for albama state
population_selected_state = population[population["State"] == selected_state_AL]
population_selected_state
```

[20]:

```
    countyFIPS            County Name State  population
0            0  Statewide Unallocated    AL           0
1         1001         Autauga County    AL       55869
2         1003         Baldwin County    AL      223234
3         1005         Barbour County    AL       24686
4         1007            Bibb County    AL       22394
..         …                    …   …         …
63        1125      Tuscaloosa County    AL      209355
64        1127          Walker County    AL       63521
65        1129      Washington County    AL       16326
66        1131          Wilcox County    AL       10373
67        1133         Winston County    AL       23629

[68 rows x 4 columns]
```

[21]:

```
#For the Albama state summing the population.
population_selected_state = population_selected_state.
 ↪groupby('State')['population'].sum()
population_selected_state
```

[21]:

```
State
AL    4903185
Name: population, dtype: int64
```

[22]:

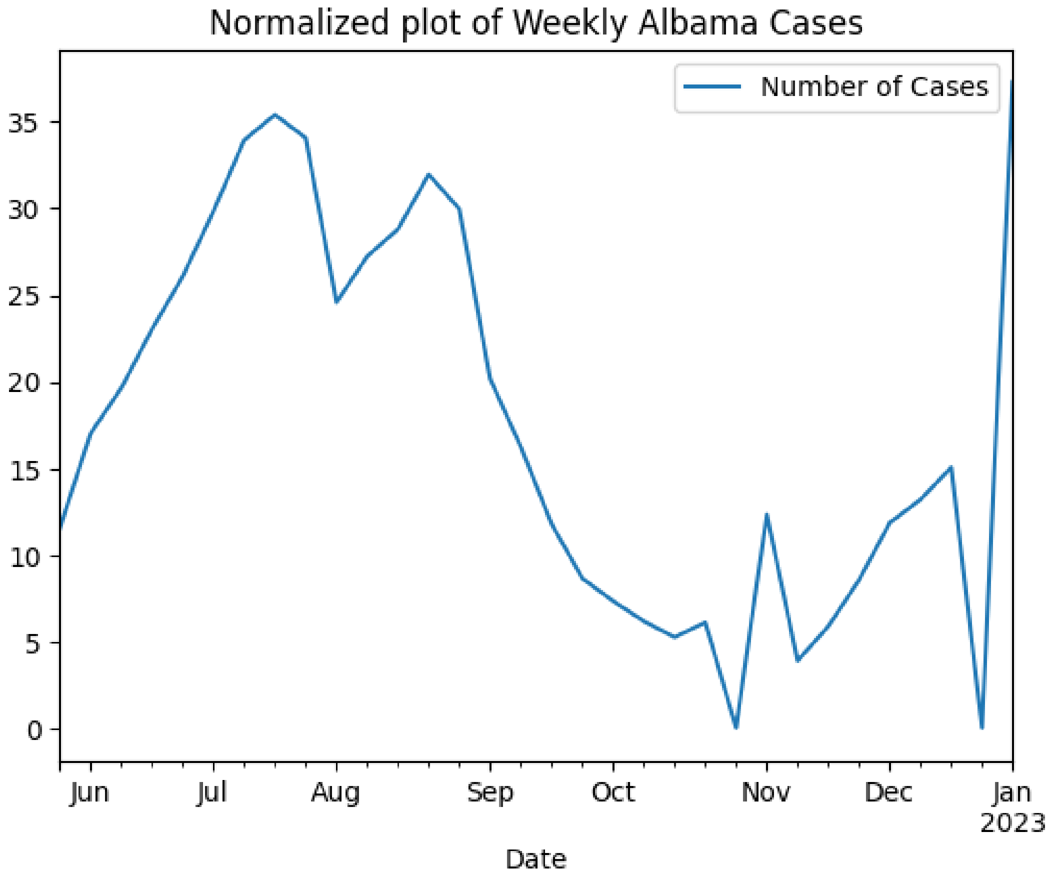
```
#normalizing by population and using normalization factor to identify cases
normalized_weekly_cases_sum_selected_state_given_range =␣
 ↪weekly_cases_sum_selected_state_given_range.copy()
```

```
normalized_weekly_cases_sum_selected_state_given_range['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_selected_state_given_range['Number of Cases'].
 ↪mul(10000)
normalized_weekly_cases_sum_selected_state_given_range['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_selected_state_given_range['Number of Cases'].
 ↪div(4903185)
normalized_weekly_cases_sum_selected_state_given_range
```

```
[22]:          Date  Number of Cases
      0  2022-05-30        11.519043
      1  2022-06-06        16.993036
      2  2022-06-13        19.619900
      3  2022-06-20        23.019731
      4  2022-06-27        26.070809
      5  2022-07-04        29.843867
      6  2022-07-11        33.955480
      7  2022-07-18        35.417795
      8  2022-07-25        34.083968
      9  2022-08-01        24.569744
      10 2022-08-08        27.235358
      11 2022-08-15        28.799648
      12 2022-08-22        31.962898
      13 2022-08-29        29.970315
      14 2022-09-05        20.150168
      15 2022-09-12        16.222109
      16 2022-09-19        11.767861
      17 2022-09-26         8.629085
      18 2022-10-03         7.319732
      19 2022-10-10         6.149064
      20 2022-10-17         5.233333
      21 2022-10-24         6.081761
      22 2022-10-31         0.000000
      23 2022-11-07        12.322603
      24 2022-11-14         3.870953
      25 2022-11-21         5.857417
      26 2022-11-28         8.537308
      27 2022-12-05        11.843322
      28 2022-12-12        13.185307
      29 2022-12-19        15.045323
      30 2022-12-26         0.000000
      31 2023-01-02        37.302284
```

```
[23]: normalized_weekly_cases_sum_selected_state_given_range.plot(x='Date', y='Number␣
       ↪of Cases', title = 'Normalized plot of Weekly Albama Cases')
```

```
[23]: <AxesSubplot: title={'center': 'Normalized plot of Weekly Albama Cases'},
      xlabel='Date'>
```



[24]:
```
cases_selected_state_NJ = cases_transpose[cases_transpose["State"] ==␣
 ↪selected_state_NJ].reset_index()
del cases_selected_state_NJ[cases_selected_state_NJ.columns[0]]
cases_selected_state_NJ.head()
```

[24]:
```
   countyFIPS          County Name State  StateFIPS        Date  \
0       34001     Atlantic County     NJ         34  2020-01-22
1       34003       Bergen County     NJ         34  2020-01-22
2       34005   Burlington County     NJ         34  2020-01-22
3       34007       Camden County     NJ         34  2020-01-22
4       34009     Cape May County     NJ         34  2020-01-22

   Number of Cases
0                0
1                0
2                0
3                0
4                0
```

[25]:
```
#For the selected state Albama summing the cases per day of all the counties.
cases_selected_state_daily_NJ = cases_selected_state_NJ.groupby('Date')['Number␣
 ↪of Cases'].sum()
cases_selected_state_daily_NJ.head()
```

[25]:
```
Date
2020-01-22    0
2020-01-23    0
2020-01-24    0
2020-01-25    0
2020-01-26    0
Name: Number of Cases, dtype: int64
```

[26]:
```
#Finding out the new cases per day.
new_cases_selected_state_daily_NJ = cases_selected_state_daily_NJ.diff().
 ↪reset_index()
new_cases_selected_state_daily_NJ.head()
```

[26]:
```
         Date  Number of Cases
0  2020-01-22              NaN
1  2020-01-23              0.0
2  2020-01-24              0.0
3  2020-01-25              0.0
4  2020-01-26              0.0
```

[27]:
```
#Converting the daily to weekly analysis and finding the mean weekly.
weekly_cases_mean_selected_state_NJ = new_cases_selected_state_daily_NJ.copy()
weekly_cases_mean_selected_state_NJ['Date'] = pd.
 ↪to_datetime(weekly_cases_mean_selected_state_NJ['Date']) - pd.
 ↪to_timedelta(7, unit='d')
weekly_cases_mean_selected_state_NJ = weekly_cases_mean_selected_state_NJ.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].mean()
weekly_cases_mean_selected_state_NJ = weekly_cases_mean_selected_state_NJ.
 ↪reset_index()
weekly_cases_mean_selected_state_NJ.head()
```

[27]:
```
        Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

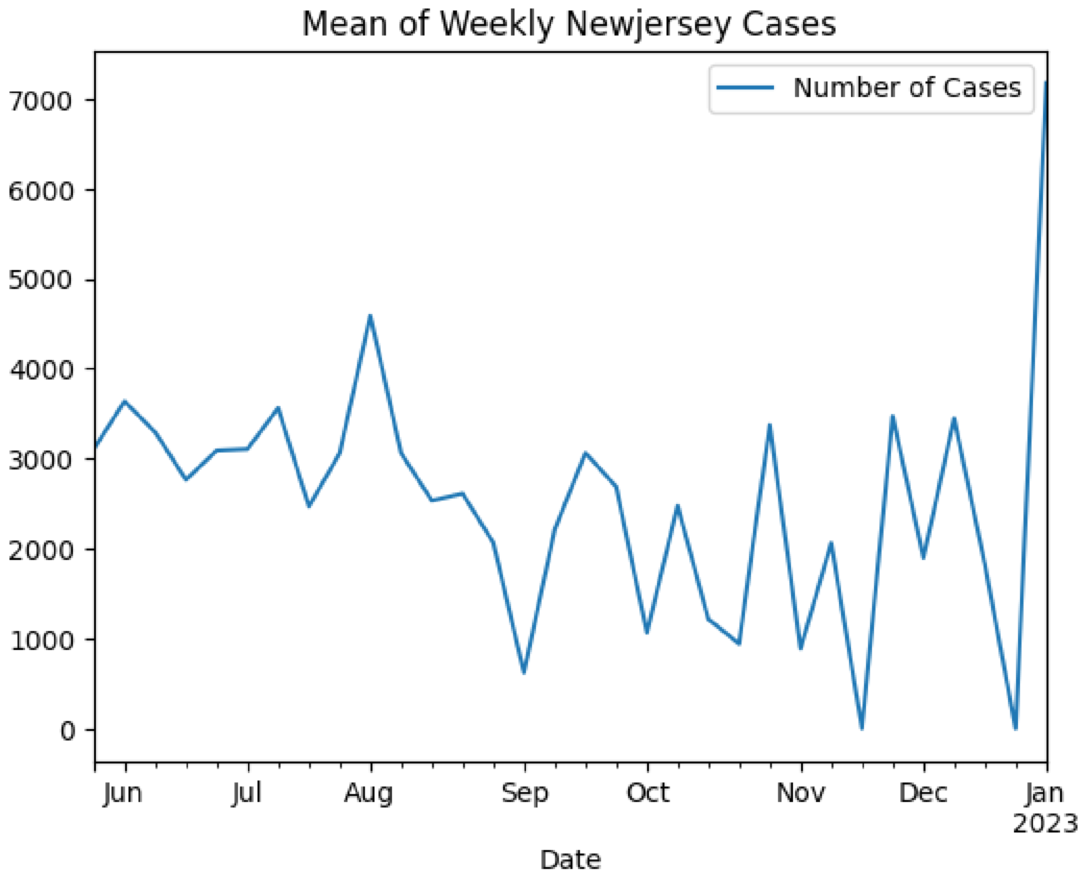
[28]:
```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
```

```
weekly_cases_mean_selected_state_given_range_NJ =␣
 ↪weekly_cases_mean_selected_state_NJ[(weekly_cases_mean_selected_state_NJ["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mean_selected_state_NJ["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mean_selected_state_given_range_NJ =␣
 ↪weekly_cases_mean_selected_state_given_range_NJ.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_mean_selected_state_given_range_NJ['Date'] =␣
 ↪weekly_cases_mean_selected_state_given_range_NJ['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_mean_selected_state_given_range_NJ
```

```
[28]:          Date  Number of Cases
      0  2022-05-30      3105.000000
      1  2022-06-06      3634.714286
      2  2022-06-13      3291.285714
      3  2022-06-20      2762.142857
      4  2022-06-27      3089.142857
      5  2022-07-04      3104.571429
      6  2022-07-11      3560.571429
      7  2022-07-18      2464.714286
      8  2022-07-25      3058.285714
      9  2022-08-01      4587.142857
      10 2022-08-08      3059.285714
      11 2022-08-15      2530.571429
      12 2022-08-22      2608.428571
      13 2022-08-29      2065.428571
      14 2022-09-05       616.571429
      15 2022-09-12      2209.285714
      16 2022-09-19      3062.000000
      17 2022-09-26      2686.000000
      18 2022-10-03      1059.142857
      19 2022-10-10      2474.285714
      20 2022-10-17      1213.714286
      21 2022-10-24       940.428571
      22 2022-10-31      3372.000000
      23 2022-11-07       885.000000
      24 2022-11-14      2060.000000
      25 2022-11-21         0.000000
      26 2022-11-28      3470.285714
      27 2022-12-05      1893.428571
      28 2022-12-12      3445.428571
      29 2022-12-19      1821.000000
      30 2022-12-26         0.000000
      31 2023-01-02      7173.142857
```

```
[29]: #Plotting the mean graph
      weekly_cases_mean_selected_state_given_range_NJ.plot(x='Date', y='Number of␣
       ↪Cases', title = 'Mean of Weekly Newjersey Cases')
```

```
[29]: <AxesSubplot: title={'center': 'Mean of Weekly Newjersey Cases'}, xlabel='Date'>
```



```
[30]: #Converting the daily to weekly analysis and finding the median weekly.
      weekly_cases_median_selected_state_NJ = new_cases_selected_state_daily_NJ.copy()
      weekly_cases_median_selected_state_NJ['Date'] = pd.
       ↪to_datetime(weekly_cases_median_selected_state_NJ['Date']) - pd.
       ↪to_timedelta(7, unit='d')
      weekly_cases_median_selected_state_NJ = weekly_cases_median_selected_state_NJ.
       ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].median()
      weekly_cases_median_selected_state_NJ = weekly_cases_median_selected_state_NJ.
       ↪reset_index()
      weekly_cases_median_selected_state_NJ.head()
```

```
[30]:         Date  Number of Cases
      0 2020-01-19              0.0
```

```
1 2020-01-26             0.0
2 2020-02-02             0.0
3 2020-02-09             0.0
4 2020-02-16             0.0
```

[31]:

```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_median_selected_state_given_range_NJ =␣
 ↪weekly_cases_median_selected_state_NJ[(weekly_cases_median_selected_state_NJ["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_median_selected_state_NJ["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_median_selected_state_given_range_NJ =␣
 ↪weekly_cases_median_selected_state_given_range_NJ.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_median_selected_state_given_range_NJ['Date'] =␣
 ↪weekly_cases_median_selected_state_given_range_NJ['Date'] + pd.
 ↪to_timedelta(1, unit='d')
weekly_cases_median_selected_state_given_range_NJ
```

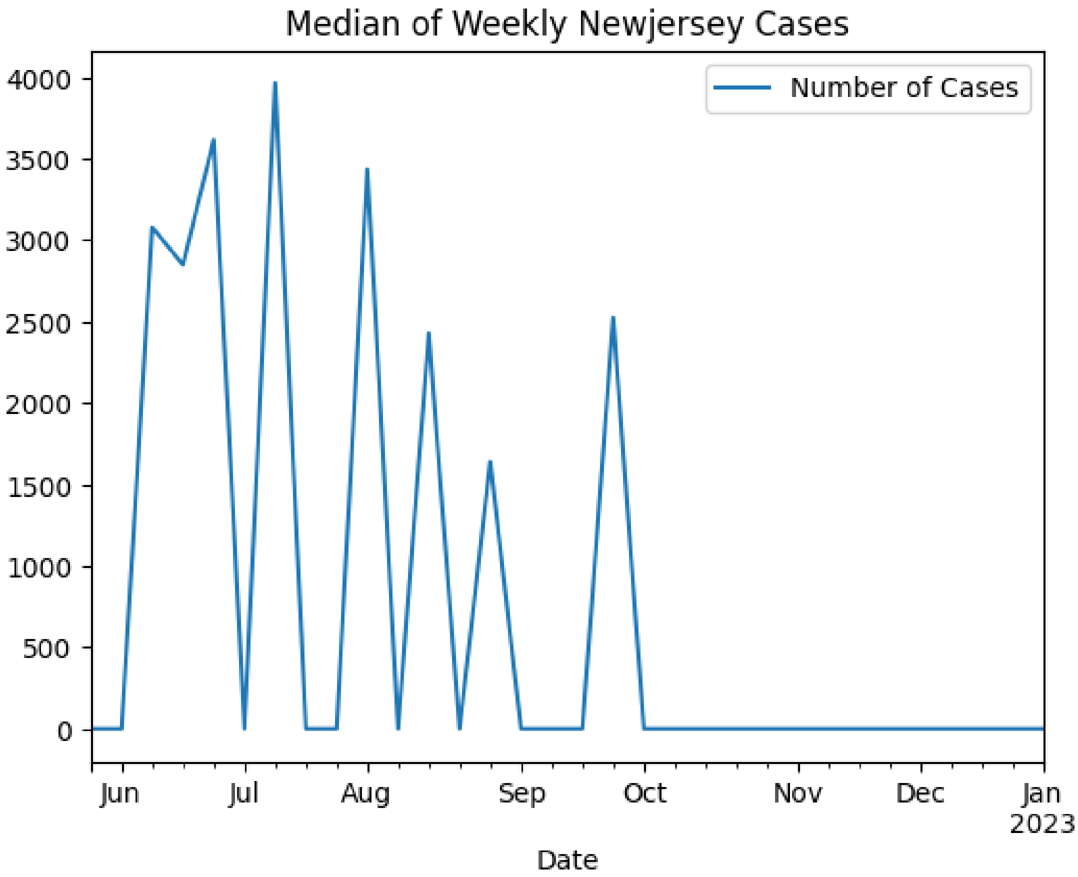
[31]:

```
          Date  Number of Cases
0   2022-05-30              0.0
1   2022-06-06              0.0
2   2022-06-13           3079.0
3   2022-06-20           2849.0
4   2022-06-27           3617.0
5   2022-07-04              0.0
6   2022-07-11           3965.0
7   2022-07-18              0.0
8   2022-07-25              0.0
9   2022-08-01           3435.0
10  2022-08-08              0.0
11  2022-08-15           2429.0
12  2022-08-22              0.0
13  2022-08-29           1641.0
14  2022-09-05              0.0
15  2022-09-12              0.0
16  2022-09-19              0.0
17  2022-09-26           2524.0
18  2022-10-03              0.0
19  2022-10-10              0.0
20  2022-10-17              0.0
21  2022-10-24              0.0
22  2022-10-31              0.0
23  2022-11-07              0.0
24  2022-11-14              0.0
25  2022-11-21              0.0
26  2022-11-28              0.0
```

```
27 2022-12-05             0.0
28 2022-12-12             0.0
29 2022-12-19             0.0
30 2022-12-26             0.0
31 2023-01-02             0.0
```

[32]:
```
#Plotting the median graph
weekly_cases_median_selected_state_given_range_NJ.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Median of Weekly Newjersey Cases')
```

[32]:
```
<AxesSubplot: title={'center': 'Median of Weekly Newjersey Cases'},
xlabel='Date'>
```



[33]:
```
#Converting the daily to weekly analysis and finding the mode weekly.
weekly_cases_mode_selected_state_NJ = new_cases_selected_state_daily_NJ.copy()
weekly_cases_mode_selected_state_NJ['Date'] = pd.
 ↪to_datetime(weekly_cases_mode_selected_state_NJ['Date']) - pd.
 ↪to_timedelta(7, unit='d')
```

```
weekly_cases_mode_selected_state_NJ = weekly_cases_mode_selected_state_NJ.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].
 ↪apply(statistics.mode)
weekly_cases_mode_selected_state_NJ = weekly_cases_mode_selected_state_NJ.
 ↪reset_index()
weekly_cases_mode_selected_state_NJ.head()
```

[33]:

```
        Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

[34]:

```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_mode_selected_state_given_range_NJ =␣
 ↪weekly_cases_mode_selected_state_NJ[(weekly_cases_mode_selected_state_NJ["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mode_selected_state_NJ["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mode_selected_state_given_range_NJ =␣
 ↪weekly_cases_mode_selected_state_given_range_NJ.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_mode_selected_state_given_range_NJ['Date'] =␣
 ↪weekly_cases_mode_selected_state_given_range_NJ['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_mode_selected_state_given_range_NJ
```

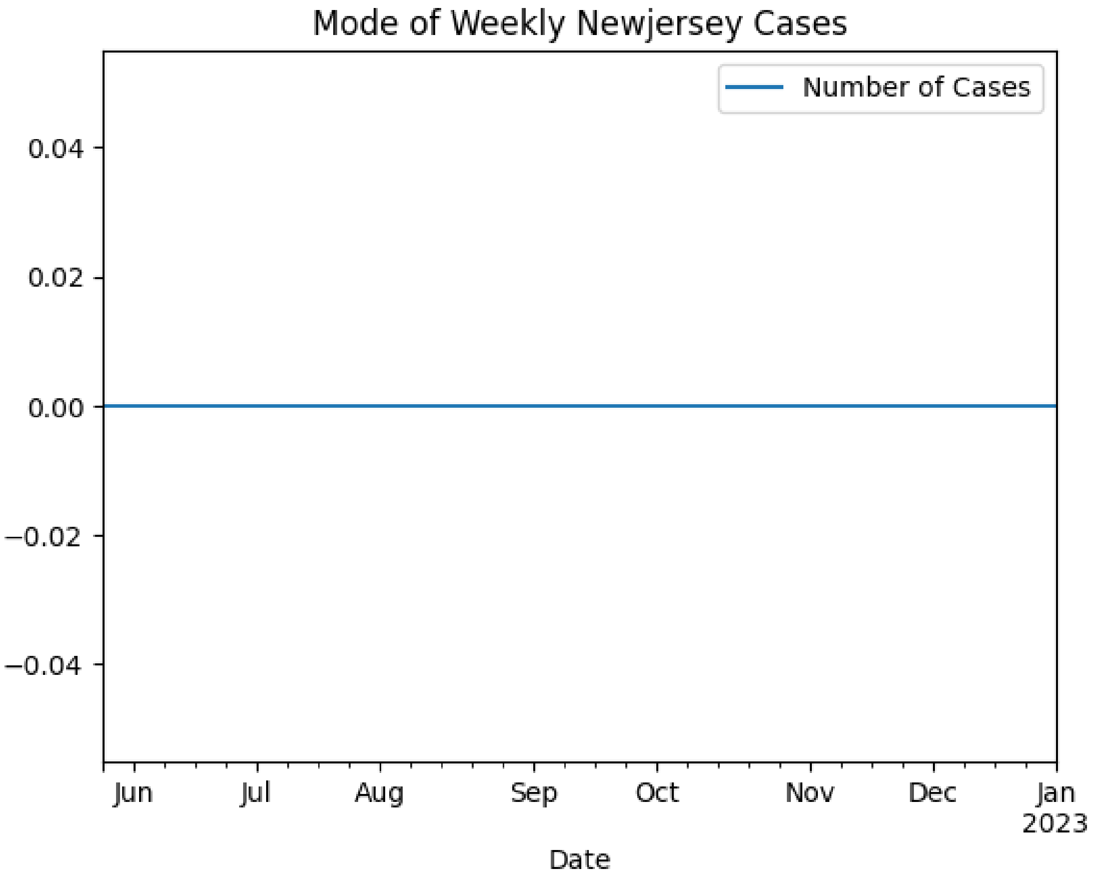
[34]:

```
         Date  Number of Cases
0  2022-05-30              0.0
1  2022-06-06              0.0
2  2022-06-13              0.0
3  2022-06-20              0.0
4  2022-06-27              0.0
5  2022-07-04              0.0
6  2022-07-11              0.0
7  2022-07-18              0.0
8  2022-07-25              0.0
9  2022-08-01              0.0
10 2022-08-08              0.0
11 2022-08-15              0.0
12 2022-08-22              0.0
13 2022-08-29              0.0
14 2022-09-05              0.0
15 2022-09-12              0.0
16 2022-09-19              0.0
17 2022-09-26              0.0
```

```
18 2022-10-03               0.0
19 2022-10-10               0.0
20 2022-10-17               0.0
21 2022-10-24               0.0
22 2022-10-31               0.0
23 2022-11-07               0.0
24 2022-11-14               0.0
25 2022-11-21               0.0
26 2022-11-28               0.0
27 2022-12-05               0.0
28 2022-12-12               0.0
29 2022-12-19               0.0
30 2022-12-26               0.0
31 2023-01-02               0.0
```

```
[35]: #Plottinf the mode graph
      weekly_cases_mode_selected_state_given_range_NJ.plot(x='Date', y='Number of␣
       ↪Cases', title = 'Mode of Weekly Newjersey Cases')
```

```
[35]: <AxesSubplot: title={'center': 'Mode of Weekly Newjersey Cases'}, xlabel='Date'>
```



```
[36]: #Converting the daily to weekly analysis and finding the weekly sum of cases.
      weekly_cases_sum_selected_state_NJ = new_cases_selected_state_daily_NJ.copy()
      weekly_cases_sum_selected_state_NJ['Date'] = pd.
       ↪to_datetime(weekly_cases_sum_selected_state_NJ['Date']) - pd.to_timedelta(7,␣
       ↪unit='d')
      weekly_cases_sum_selected_state_NJ = weekly_cases_sum_selected_state_NJ.
       ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
      weekly_cases_sum_selected_state_NJ = weekly_cases_sum_selected_state_NJ.
       ↪reset_index()
      weekly_cases_sum_selected_state_NJ.head()
```

```
[36]:         Date  Number of Cases
      0 2020-01-19              0.0
      1 2020-01-26              0.0
      2 2020-02-02              0.0
      3 2020-02-09              0.0
      4 2020-02-16              0.0
```

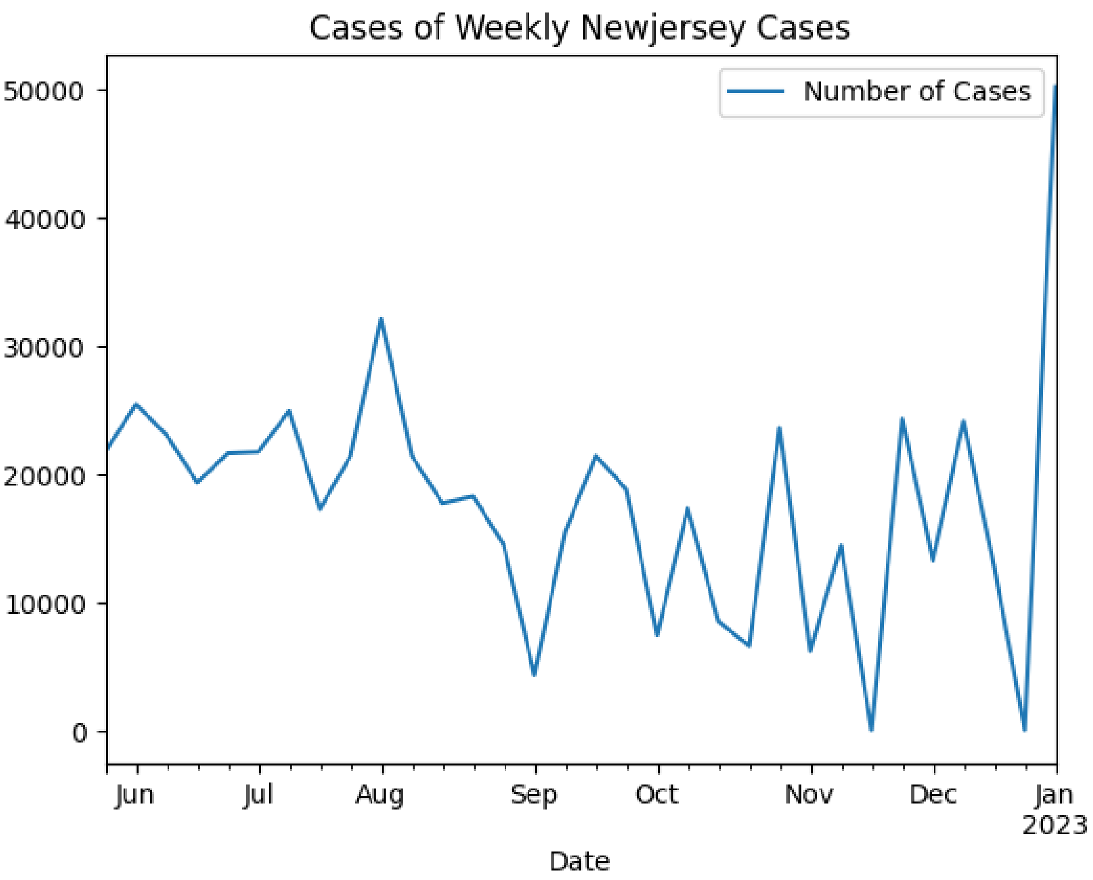
```
[37]: #Adding one day so that we get weekly analysis from monday to sunday and weekly␣
       ↪analsis from monday to sunday.
      weekly_cases_sum_selected_state_given_range_NJ =␣
       ↪weekly_cases_sum_selected_state_NJ[(weekly_cases_sum_selected_state_NJ["Date"]␣
       ↪>= '2022-05-29') & (weekly_cases_sum_selected_state["Date"] <= '2023-01-02')]
      weekly_cases_sum_selected_state_given_range_NJ =␣
       ↪weekly_cases_sum_selected_state_given_range_NJ.sort_values(by=['Date']).
       ↪reset_index(drop=True)
      weekly_cases_sum_selected_state_given_range_NJ['Date'] =␣
       ↪weekly_cases_sum_selected_state_given_range_NJ['Date'] + pd.to_timedelta(1,␣
       ↪unit='d')
      weekly_cases_sum_selected_state_given_range_NJ
```

```
[37]:          Date  Number of Cases
      0   2022-05-30          21735.0
      1   2022-06-06          25443.0
      2   2022-06-13          23039.0
      3   2022-06-20          19335.0
      4   2022-06-27          21624.0
      5   2022-07-04          21732.0
      6   2022-07-11          24924.0
      7   2022-07-18          17253.0
      8   2022-07-25          21408.0
      9   2022-08-01          32110.0
      10  2022-08-08          21415.0
      11  2022-08-15          17714.0
      12  2022-08-22          18259.0
```

```
13 2022-08-29          14458.0
14 2022-09-05           4316.0
15 2022-09-12          15465.0
16 2022-09-19          21434.0
17 2022-09-26          18802.0
18 2022-10-03           7414.0
19 2022-10-10          17320.0
20 2022-10-17           8496.0
21 2022-10-24           6583.0
22 2022-10-31          23604.0
23 2022-11-07           6195.0
24 2022-11-14          14420.0
25 2022-11-21              0.0
26 2022-11-28          24292.0
27 2022-12-05          13254.0
28 2022-12-12          24118.0
29 2022-12-19          12747.0
30 2022-12-26              0.0
31 2023-01-02          50212.0
```

[38]:
```
weekly_cases_sum_selected_state_given_range_NJ.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Cases of Weekly Newjersey Cases')
```

[38]:
```
<AxesSubplot: title={'center': 'Cases of Weekly Newjersey Cases'},
xlabel='Date'>
```



0.1.2 Week starting with 2023-01-02 has a peak of cases in newjersey state with value 50212. Due to long vacation of christmas and new year the cases have incread as many people might have gathered for celebrations.

[39]:

```
#Filtering the population for NEWJERSEY state
population_selected_state_NJ = population[population["State"] ==␣
 ↪selected_state_NJ].reset_index()
del population_selected_state_NJ[population_selected_state_NJ.columns[0]]
population_selected_state_NJ
```

[39]:

```
   countyFIPS            County Name State  population
0           0  Statewide Unallocated    NJ           0
1       34001        Atlantic County    NJ      263670
2       34003          Bergen County    NJ      932202
3       34005      Burlington County    NJ      445349
4       34007          Camden County    NJ      506471
5       34009        Cape May County    NJ       92039
6       34011      Cumberland County    NJ      149527
7       34013           Essex County    NJ      798975
```

```
8         34015      Gloucester County    NJ      291636
9         34017          Hudson County    NJ      672391
10        34019       Hunterdon County    NJ      124371
11        34021          Mercer County    NJ      367430
12        34023       Middlesex County    NJ      825062
13        34025        Monmouth County    NJ      618795
14        34027          Morris County    NJ      491845
15        34029           Ocean County    NJ      607186
16        34031         Passaic County    NJ      501826
17        34033           Salem County    NJ       62385
18        34035        Somerset County    NJ      328934
19        34037          Sussex County    NJ      140488
20        34039           Union County    NJ      556341
21        34041          Warren County    NJ      105267
```

[40]:

```
#For the Albama state summing the population.
population_selected_state_NJ = population_selected_state_NJ.
 ↪groupby('State')['population'].sum()
population_selected_state_NJ
```

[40]:

```
State
NJ    8882190
Name: population, dtype: int64
```

[41]:

```
#normalizing by population and using normalization factor to identify cases
normalized_weekly_cases_sum_selected_state_given_range_NJ =␣
 ↪weekly_cases_sum_selected_state_given_range_NJ.copy()
normalized_weekly_cases_sum_selected_state_given_range_NJ['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_NJ['Number of Cases'].
 ↪mul(10000)
normalized_weekly_cases_sum_selected_state_given_range_NJ['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_NJ['Number of Cases'].
 ↪div(8882190)
normalized_weekly_cases_sum_selected_state_given_range_NJ
```

[41]:

```
         Date  Number of Cases
0  2022-05-30        24.470316
1  2022-06-06        28.644963
2  2022-06-13        25.938423
3  2022-06-20        21.768280
4  2022-06-27        24.345347
5  2022-07-04        24.466939
6  2022-07-11        28.060647
7  2022-07-18        19.424264
8  2022-07-25        24.102164
9  2022-08-01        36.150994
10 2022-08-08        24.110045
```

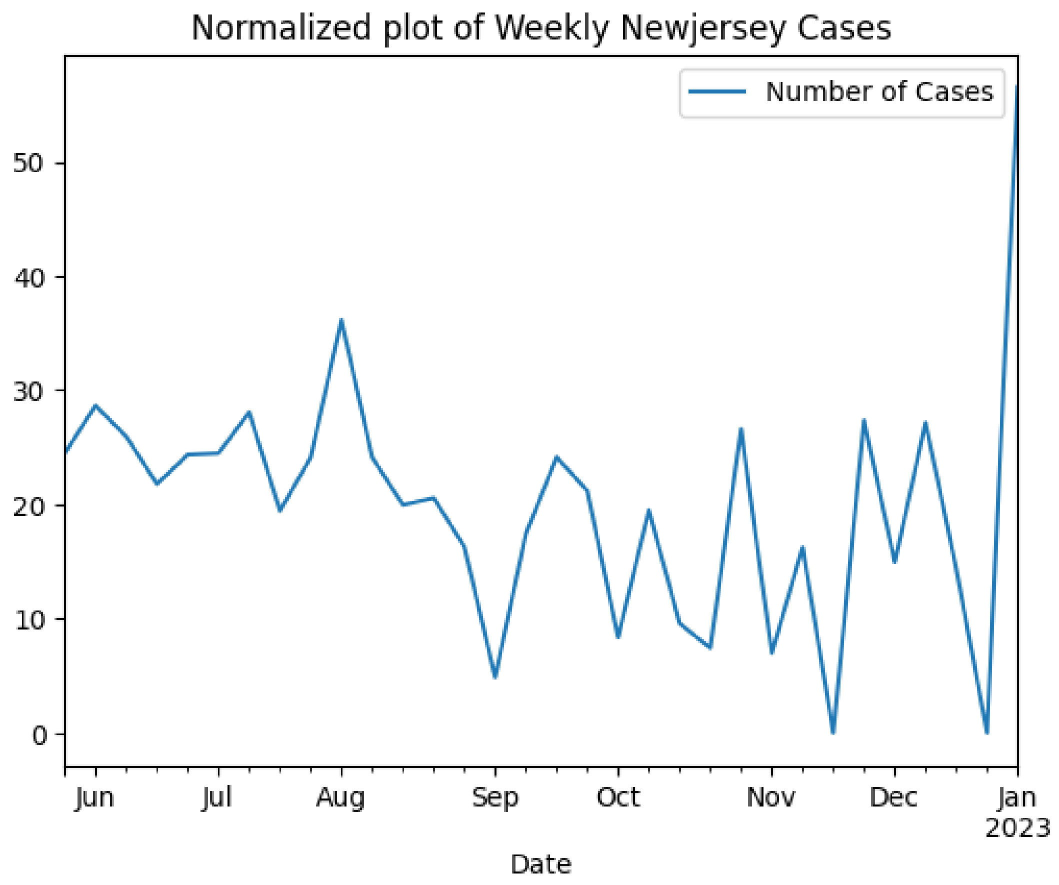
```
11 2022-08-15        19.943280
12 2022-08-22        20.556867
13 2022-08-29        16.277517
14 2022-09-05         4.859162
15 2022-09-12        17.411247
16 2022-09-19        24.131436
17 2022-09-26        21.168203
18 2022-10-03         8.347041
19 2022-10-10        19.499695
20 2022-10-17         9.565209
21 2022-10-24         7.411460
22 2022-10-31        26.574527
23 2022-11-07         6.974631
24 2022-11-14        16.234735
25 2022-11-21         0.000000
26 2022-11-28        27.349111
27 2022-12-05        14.921996
28 2022-12-12        27.153213
29 2022-12-19        14.351190
30 2022-12-26         0.000000
31 2023-01-02        56.531103
```

[42]:

```
normalized_weekly_cases_sum_selected_state_given_range_NJ.plot(x='Date',␣
 ↪y='Number of Cases', title = 'Normalized plot of Weekly Newjersey Cases')
```

[42]:

```
<AxesSubplot: title={'center': 'Normalized plot of Weekly Newjersey Cases'},
xlabel='Date'>
```



```
[43]: cases_selected_state_MD = cases_transpose[cases_transpose["State"] ==␣
       ↪selected_state_MD].reset_index()
      del cases_selected_state_MD[cases_selected_state_MD.columns[0]]
      cases_selected_state_MD.head()
```

```
[43]:    countyFIPS           County Name State  StateFIPS        Date  \
      0       24001       Allegany County    MD         24  2020-01-22
      1       24003  Anne Arundel County    MD         24  2020-01-22
      2       24005      Baltimore County    MD         24  2020-01-22
      3       24009        Calvert County    MD         24  2020-01-22
      4       24011       Caroline County    MD         24  2020-01-22

         Number of Cases
      0                0
      1                0
      2                0
      3                0
      4                0
```

[44]:

```
#For the selected state Albama summing the deaths per day of all the counties.
cases_selected_state_daily_MD = cases_selected_state_MD.groupby('Date')['Number␣
 ↪of Cases'].sum()
cases_selected_state_daily_MD.head()
```

[44]:

```
Date
2020-01-22    0
2020-01-23    0
2020-01-24    0
2020-01-25    0
2020-01-26    0
Name: Number of Cases, dtype: int64
```

[45]:

```
#Finding out the new cases per day.
new_cases_selected_state_daily_MD = cases_selected_state_daily_MD.diff().
 ↪reset_index()
new_cases_selected_state_daily_MD.head()
```

[45]:

```
         Date  Number of Cases
0  2020-01-22              NaN
1  2020-01-23              0.0
2  2020-01-24              0.0
3  2020-01-25              0.0
4  2020-01-26              0.0
```

[46]:

```
#Converting the daily to weekly analysis and finding the mean weekly.
weekly_cases_mean_selected_state_MD = new_cases_selected_state_daily_MD.copy()
weekly_cases_mean_selected_state_MD['Date'] = pd.
 ↪to_datetime(weekly_cases_mean_selected_state_MD['Date']) - pd.
 ↪to_timedelta(7, unit='d')
weekly_cases_mean_selected_state_MD = weekly_cases_mean_selected_state_MD.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].mean()
weekly_cases_mean_selected_state_MD = weekly_cases_mean_selected_state_MD.
 ↪reset_index()
weekly_cases_mean_selected_state_MD.head()
```

[46]:

```
        Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

[47]:

```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
```

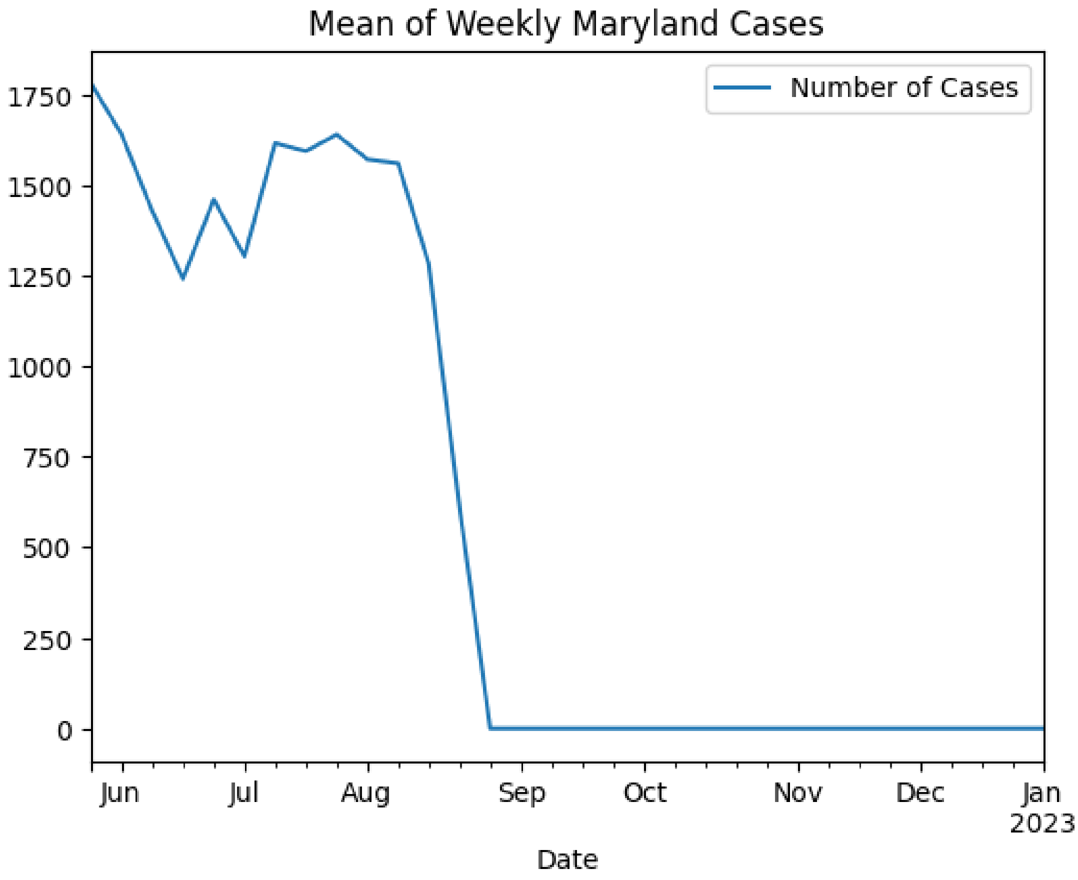
```
weekly_cases_mean_selected_state_given_range_MD =␣
 ↪weekly_cases_mean_selected_state_MD[(weekly_cases_mean_selected_state_MD["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mean_selected_state_MD["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mean_selected_state_given_range_MD =␣
 ↪weekly_cases_mean_selected_state_given_range_MD.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_mean_selected_state_given_range_MD['Date'] =␣
 ↪weekly_cases_mean_selected_state_given_range_MD['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_mean_selected_state_given_range_MD
```

[47]:

```
         Date  Number of Cases
0  2022-05-30      1782.714286
1  2022-06-06      1641.142857
2  2022-06-13      1429.428571
3  2022-06-20      1241.571429
4  2022-06-27      1461.285714
5  2022-07-04      1303.857143
6  2022-07-11      1616.714286
7  2022-07-18      1594.428571
8  2022-07-25      1640.428571
9  2022-08-01      1572.142857
10 2022-08-08      1561.428571
11 2022-08-15      1282.428571
12 2022-08-22       609.714286
13 2022-08-29         0.000000
14 2022-09-05         0.000000
15 2022-09-12         0.000000
16 2022-09-19         0.000000
17 2022-09-26         0.000000
18 2022-10-03         0.000000
19 2022-10-10         0.000000
20 2022-10-17         0.000000
21 2022-10-24         0.000000
22 2022-10-31         0.000000
23 2022-11-07         0.000000
24 2022-11-14         0.000000
25 2022-11-21         0.000000
26 2022-11-28         0.000000
27 2022-12-05         0.000000
28 2022-12-12         0.000000
29 2022-12-19         0.000000
30 2022-12-26         0.000000
31 2023-01-02         0.000000
```

```
[48]: #Plotting the mean graph
      weekly_cases_mean_selected_state_given_range_MD.plot(x='Date', y='Number of␣
       ↪Cases', title = 'Mean of Weekly Maryland Cases')
```

```
[48]: <AxesSubplot: title={'center': 'Mean of Weekly Maryland Cases'}, xlabel='Date'>
```



```
[49]: #Converting the daily to weekly analysis and finding the median weekly.
      weekly_cases_median_selected_state_MD = new_cases_selected_state_daily_MD.copy()
      weekly_cases_median_selected_state_MD['Date'] = pd.
       ↪to_datetime(weekly_cases_median_selected_state_MD['Date']) - pd.
       ↪to_timedelta(7, unit='d')
      weekly_cases_median_selected_state_MD = weekly_cases_median_selected_state_MD.
       ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].median()
      weekly_cases_median_selected_state_MD = weekly_cases_median_selected_state_MD.
       ↪reset_index()
      weekly_cases_median_selected_state_MD.head()
```

```
[49]:         Date  Number of Cases
      0 2020-01-19              0.0
```

```
1 2020-01-26             0.0
2 2020-02-02             0.0
3 2020-02-09             0.0
4 2020-02-16             0.0
```

[50]:

```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_median_selected_state_given_range_MD =␣
 ↪weekly_cases_median_selected_state_MD[(weekly_cases_median_selected_state_MD["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_median_selected_state_MD["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_median_selected_state_given_range_MD =␣
 ↪weekly_cases_median_selected_state_given_range_MD.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_median_selected_state_given_range_MD['Date'] =␣
 ↪weekly_cases_median_selected_state_given_range_MD['Date'] + pd.
 ↪to_timedelta(1, unit='d')
weekly_cases_median_selected_state_given_range_MD
```

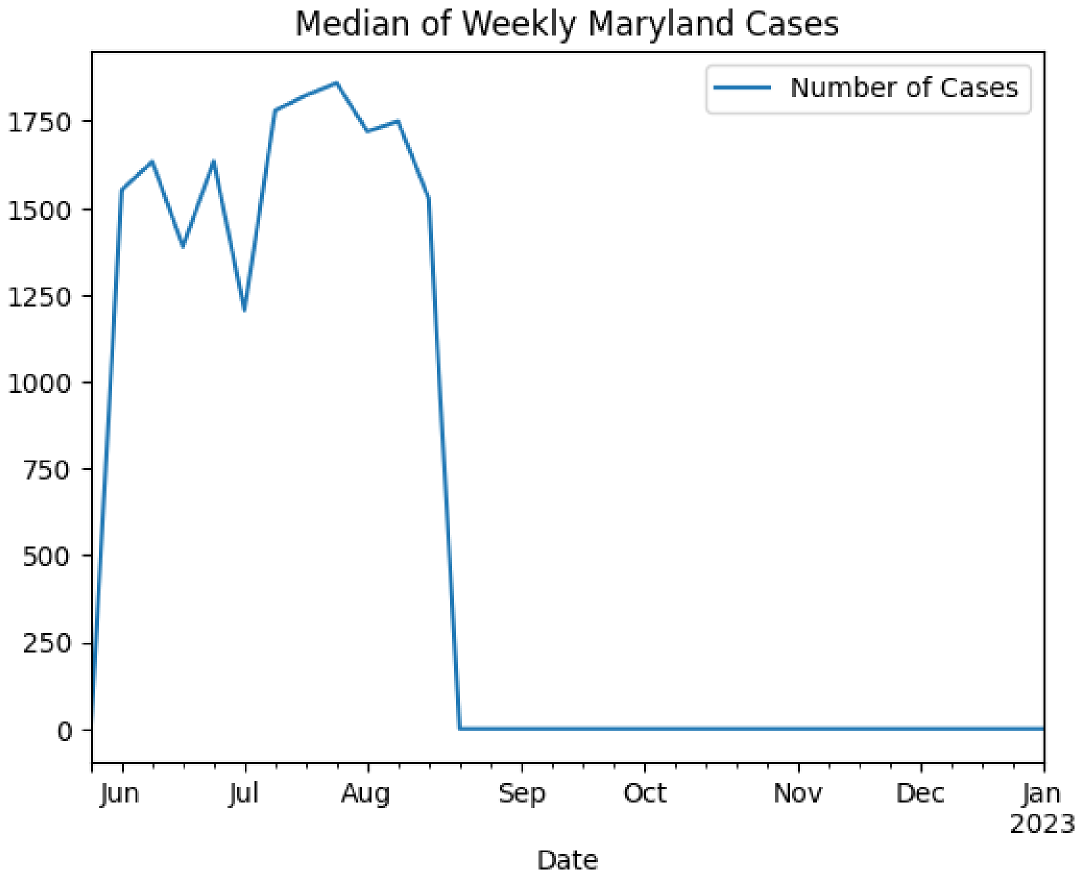
[50]:

```
         Date  Number of Cases
0  2022-05-30              0.0
1  2022-06-06           1551.0
2  2022-06-13           1633.0
3  2022-06-20           1388.0
4  2022-06-27           1634.0
5  2022-07-04           1205.0
6  2022-07-11           1780.0
7  2022-07-18           1824.0
8  2022-07-25           1859.0
9  2022-08-01           1720.0
10 2022-08-08           1749.0
11 2022-08-15           1526.0
12 2022-08-22              0.0
13 2022-08-29              0.0
14 2022-09-05              0.0
15 2022-09-12              0.0
16 2022-09-19              0.0
17 2022-09-26              0.0
18 2022-10-03              0.0
19 2022-10-10              0.0
20 2022-10-17              0.0
21 2022-10-24              0.0
22 2022-10-31              0.0
23 2022-11-07              0.0
24 2022-11-14              0.0
25 2022-11-21              0.0
26 2022-11-28              0.0
```

```
27 2022-12-05              0.0
28 2022-12-12              0.0
29 2022-12-19              0.0
30 2022-12-26              0.0
31 2023-01-02              0.0
```

```
[51]: #Plotting the median graph
weekly_cases_median_selected_state_given_range_MD.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Median of Weekly Maryland Cases')
```

```
[51]: <AxesSubplot: title={'center': 'Median of Weekly Maryland Cases'},
xlabel='Date'>
```



```
[52]: #Converting the daily to weekly analysis and finding the mode weekly.
weekly_cases_mode_selected_state_MD = new_cases_selected_state_daily_MD.copy()
weekly_cases_mode_selected_state_MD['Date'] = pd.
 ↪to_datetime(weekly_cases_mode_selected_state_MD['Date']) - pd.
 ↪to_timedelta(7, unit='d')
```

```
weekly_cases_mode_selected_state_MD = weekly_cases_mode_selected_state_MD.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].
 ↪apply(statistics.mode)
weekly_cases_mode_selected_state_MD = weekly_cases_mode_selected_state_MD.
 ↪reset_index()
weekly_cases_mode_selected_state_MD.head()
```

[52]:

```
        Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

[53]:

```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_mode_selected_state_given_range_MD =␣
 ↪weekly_cases_mode_selected_state_MD[(weekly_cases_mode_selected_state_MD["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mode_selected_state_MD["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mode_selected_state_given_range_MD =␣
 ↪weekly_cases_mode_selected_state_given_range_MD.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_cases_mode_selected_state_given_range_MD['Date'] =␣
 ↪weekly_cases_mode_selected_state_given_range_MD['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_mode_selected_state_given_range_MD
```

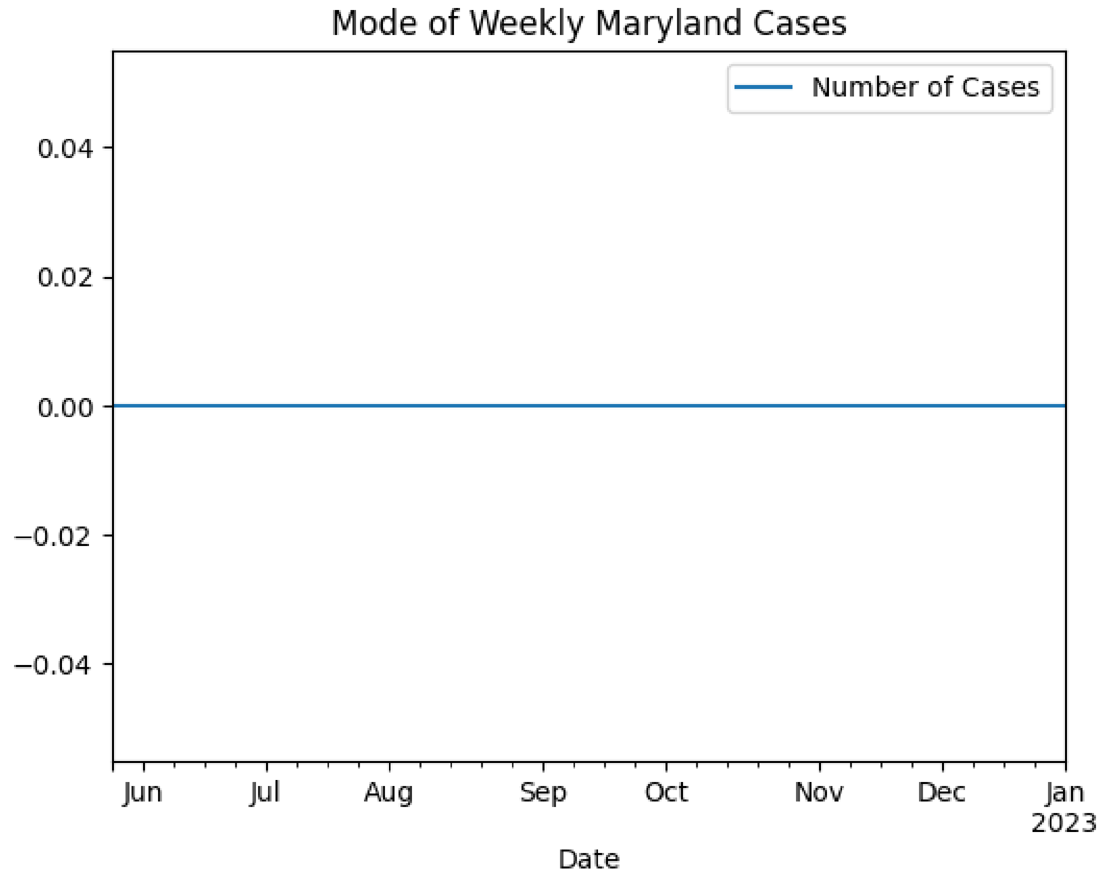
[53]:

```
         Date  Number of Cases
0  2022-05-30              0.0
1  2022-06-06              0.0
2  2022-06-13              0.0
3  2022-06-20              0.0
4  2022-06-27              0.0
5  2022-07-04              0.0
6  2022-07-11              0.0
7  2022-07-18              0.0
8  2022-07-25              0.0
9  2022-08-01              0.0
10 2022-08-08              0.0
11 2022-08-15              0.0
12 2022-08-22              0.0
13 2022-08-29              0.0
14 2022-09-05              0.0
15 2022-09-12              0.0
16 2022-09-19              0.0
17 2022-09-26              0.0
```

```
18 2022-10-03              0.0
19 2022-10-10              0.0
20 2022-10-17              0.0
21 2022-10-24              0.0
22 2022-10-31              0.0
23 2022-11-07              0.0
24 2022-11-14              0.0
25 2022-11-21              0.0
26 2022-11-28              0.0
27 2022-12-05              0.0
28 2022-12-12              0.0
29 2022-12-19              0.0
30 2022-12-26              0.0
31 2023-01-02              0.0
```

```
[54]: #Plottinf the mode graph
      weekly_cases_mode_selected_state_given_range_MD.plot(x='Date', y='Number of␣
       ↪Cases', title = 'Mode of Weekly Maryland Cases')
```

```
[54]: <AxesSubplot: title={'center': 'Mode of Weekly Maryland Cases'}, xlabel='Date'>
```



```
[55]: #Converting the daily to weekly analysis and finding the weekly sum of cases.
      weekly_cases_sum_selected_state_MD = new_cases_selected_state_daily_MD.copy()
      weekly_cases_sum_selected_state_MD['Date'] = pd.
       ↪to_datetime(weekly_cases_sum_selected_state_MD['Date']) - pd.to_timedelta(7,␣
       ↪unit='d')
      weekly_cases_sum_selected_state_MD = weekly_cases_sum_selected_state_MD.
       ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
      weekly_cases_sum_selected_state_MD = weekly_cases_sum_selected_state_MD.
       ↪reset_index()
      weekly_cases_sum_selected_state_MD.head()
```

```
[55]:         Date  Number of Cases
      0 2020-01-19              0.0
      1 2020-01-26              0.0
      2 2020-02-02              0.0
      3 2020-02-09              0.0
      4 2020-02-16              0.0
```

```
[56]: #Adding one day so that we get weekly analysis from monday to sunday and weekly␣
       ↪analsis from monday to sunday.
      weekly_cases_sum_selected_state_given_range_MD =␣
       ↪weekly_cases_sum_selected_state_MD[(weekly_cases_sum_selected_state_MD["Date"]␣
       ↪>= '2022-05-29') & (weekly_cases_sum_selected_state_MD["Date"] <=␣
       ↪'2023-01-02')]
      weekly_cases_sum_selected_state_given_range_MD =␣
       ↪weekly_cases_sum_selected_state_given_range_MD.sort_values(by=['Date']).
       ↪reset_index(drop=True)
      weekly_cases_sum_selected_state_given_range_MD['Date'] =␣
       ↪weekly_cases_sum_selected_state_given_range_MD['Date'] + pd.to_timedelta(1,␣
       ↪unit='d')
      weekly_cases_sum_selected_state_given_range_MD
```

```
[56]:          Date  Number of Cases
      0   2022-05-30          12479.0
      1   2022-06-06          11488.0
      2   2022-06-13          10006.0
      3   2022-06-20           8691.0
      4   2022-06-27          10229.0
      5   2022-07-04           9127.0
      6   2022-07-11          11317.0
      7   2022-07-18          11161.0
      8   2022-07-25          11483.0
      9   2022-08-01          11005.0
      10  2022-08-08          10930.0
      11  2022-08-15           8977.0
```

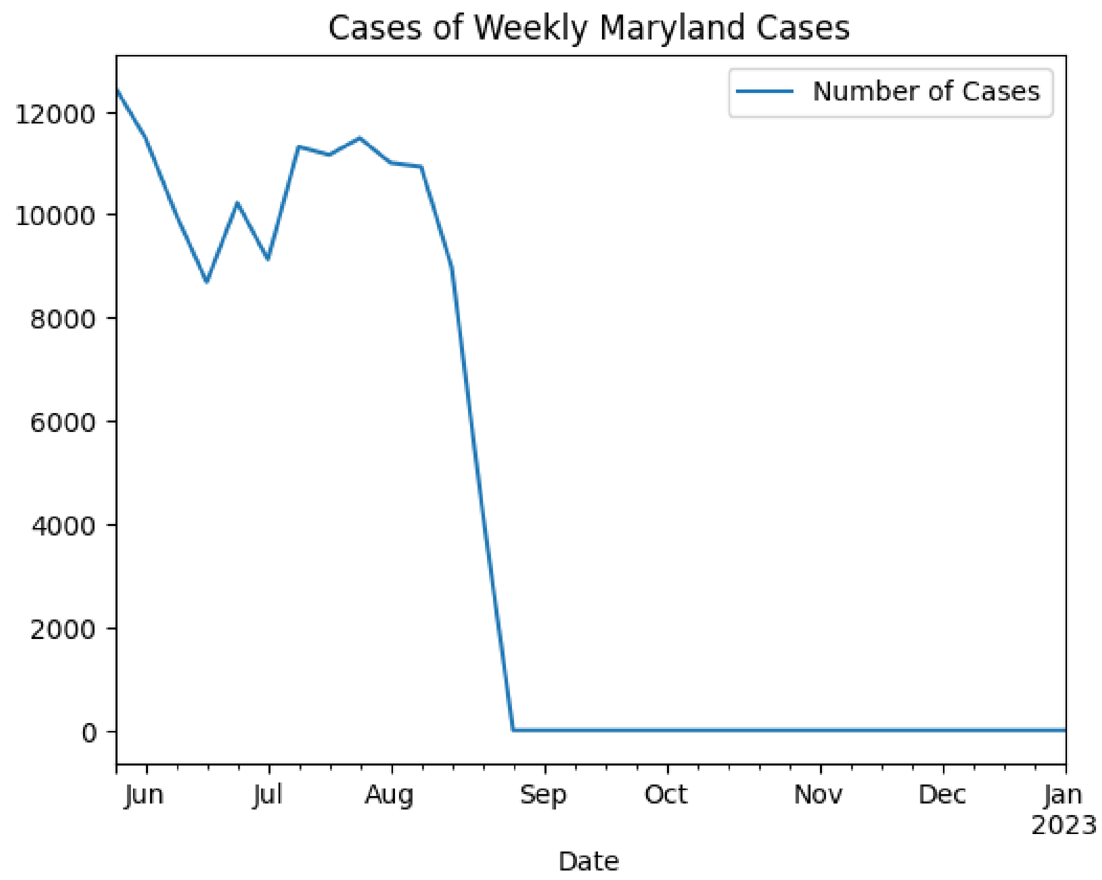
```
12 2022-08-22          4268.0
13 2022-08-29             0.0
14 2022-09-05             0.0
15 2022-09-12             0.0
16 2022-09-19             0.0
17 2022-09-26             0.0
18 2022-10-03             0.0
19 2022-10-10             0.0
20 2022-10-17             0.0
21 2022-10-24             0.0
22 2022-10-31             0.0
23 2022-11-07             0.0
24 2022-11-14             0.0
25 2022-11-21             0.0
26 2022-11-28             0.0
27 2022-12-05             0.0
28 2022-12-12             0.0
29 2022-12-19             0.0
30 2022-12-26             0.0
31 2023-01-02             0.0
```

[57]:

```
weekly_cases_sum_selected_state_given_range_MD.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Cases of Weekly Maryland Cases')
```

[57]:

```
<AxesSubplot: title={'center': 'Cases of Weekly Maryland Cases'}, xlabel='Date'>
```



0.1.3 Week starting with 2022-05-30 has a peak of cases in Maryland State with value 12479. As it was a holiday so is the reason it got spike in cases.

```
[58]: #Filtering the population for maryland state
population_selected_state_MD = population[population["State"] ==␣
 ↪selected_state_MD].reset_index()
del population_selected_state_MD[population_selected_state_MD.columns[0]]
population_selected_state_MD
```

```
[58]:     countyFIPS             County Name State  population
      0            0  Statewide Unallocated    MD           0
      1        24001         Allegany County    MD       70416
      2        24003     Anne Arundel County    MD      579234
      3        24005        Baltimore County    MD      827370
      4        24009          Calvert County    MD       92525
      5        24011        Caroline County    MD       33406
      6        24013          Carroll County    MD      168447
      7        24015            Cecil County    MD      102855
      8        24017          Charles County    MD      163257
```

```
9         24019        Dorchester County    MD       31929
10        24021         Frederick County    MD      259547
11        24023           Garrett County    MD       29014
12        24025           Harford County    MD      255441
13        24027            Howard County    MD      325690
14        24029              Kent County    MD       19422
15        24031        Montgomery County    MD     1050688
16        24033   Prince George's County    MD      909327
17        24035      Queen Anne's County    MD       50381
18        24037        St. Mary's County    MD      113510
19        24039          Somerset County    MD       25616
20        24041            Talbot County    MD       37181
21        24043        Washington County    MD      151049
22        24045          Wicomico County    MD      103609
23        24047         Worcester County    MD       52276
24        24510           Baltimore City    MD      593490
```

[59]:
```
#For the Albama state summing the population.
population_selected_state_MD = population_selected_state_MD.
 ↪groupby('State')['population'].sum()
population_selected_state_MD
```

[59]:
```
State
MD    6045680
Name: population, dtype: int64
```

[60]:
```
#normalizing by population and using normalization factor to identify cases
normalized_weekly_cases_sum_selected_state_given_range_MD =␣
 ↪weekly_cases_sum_selected_state_given_range_MD.copy()
normalized_weekly_cases_sum_selected_state_given_range_MD['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_MD['Number of Cases'].
 ↪mul(10000)
normalized_weekly_cases_sum_selected_state_given_range_MD['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_MD['Number of Cases'].
 ↪div(6045680)
normalized_weekly_cases_sum_selected_state_given_range_MD
```

[60]:
```
         Date  Number of Cases
0  2022-05-30        20.641185
1  2022-06-06        19.001998
2  2022-06-13        16.550661
3  2022-06-20        14.375554
4  2022-06-27        16.919519
5  2022-07-04        15.096730
6  2022-07-11        18.719152
7  2022-07-18        18.461116
8  2022-07-25        18.993728
```

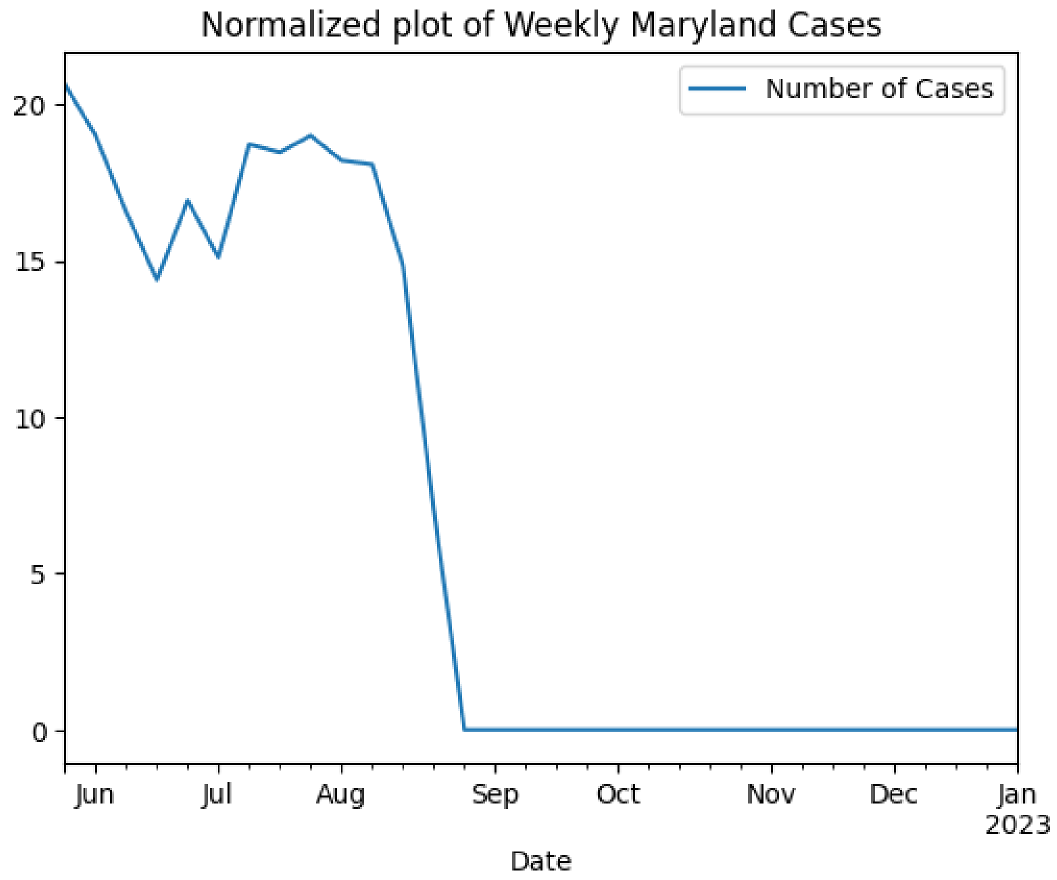
```
9   2022-08-01         18.203081
10  2022-08-08         18.079025
11  2022-08-15         14.848619
12  2022-08-22          7.059586
13  2022-08-29          0.000000
14  2022-09-05          0.000000
15  2022-09-12          0.000000
16  2022-09-19          0.000000
17  2022-09-26          0.000000
18  2022-10-03          0.000000
19  2022-10-10          0.000000
20  2022-10-17          0.000000
21  2022-10-24          0.000000
22  2022-10-31          0.000000
23  2022-11-07          0.000000
24  2022-11-14          0.000000
25  2022-11-21          0.000000
26  2022-11-28          0.000000
27  2022-12-05          0.000000
28  2022-12-12          0.000000
29  2022-12-19          0.000000
30  2022-12-26          0.000000
31  2023-01-02          0.000000
```

[61]:

```
normalized_weekly_cases_sum_selected_state_given_range_MD.plot(x='Date',␣
 ↪y='Number of Cases', title = 'Normalized plot of Weekly Maryland Cases')
```

[61]:

```
<AxesSubplot: title={'center': 'Normalized plot of Weekly Maryland Cases'},
xlabel='Date'>
```



```
[62]: cases_selected_state_NC = cases_transpose[cases_transpose["State"] ==␣
       ↪selected_state_NC].reset_index()
      del cases_selected_state_NC[cases_selected_state_NC.columns[0]]
      cases_selected_state_NC.head()
```

```
[62]:    countyFIPS        County Name State  StateFIPS        Date  Number of Cases
      0       37001   Alamance County    NC         37  2020-01-22                0
      1       37003  Alexander County    NC         37  2020-01-22                0
      2       37005  Alleghany County    NC         37  2020-01-22                0
      3       37007      Anson County    NC         37  2020-01-22                0
      4       37009       Ashe County    NC         37  2020-01-22                0
```

```
[63]: #For the selected state Albama summing the deaths per day of all the counties.
      cases_selected_state_daily_NC = cases_selected_state_NC.groupby('Date')['Number␣
       ↪of Cases'].sum()
      cases_selected_state_daily_NC.head()
```

```
[63]: Date
      2020-01-22    0
```

```
2020-01-23    0
2020-01-24    0
2020-01-25    0
2020-01-26    0
Name: Number of Cases, dtype: int64
```

[64]:
```
#Finding out the new cases per day.
new_cases_selected_state_daily_NC = cases_selected_state_daily_NC.diff().
 ↪reset_index()
new_cases_selected_state_daily_NC.head()
```

[64]:
```
        Date  Number of Cases
0 2020-01-22              NaN
1 2020-01-23              0.0
2 2020-01-24              0.0
3 2020-01-25              0.0
4 2020-01-26              0.0
```

[65]:
```
#Converting the daily to weekly analysis and finding the mean weekly.
weekly_cases_mean_selected_state_NC = new_cases_selected_state_daily_NC.copy()
weekly_cases_mean_selected_state_NC['Date'] = pd.
 ↪to_datetime(weekly_cases_mean_selected_state_NC['Date']) - pd.
 ↪to_timedelta(7, unit='d')
weekly_cases_mean_selected_state_NC = weekly_cases_mean_selected_state_NC.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].mean()
weekly_cases_mean_selected_state_NC = weekly_cases_mean_selected_state_NC.
 ↪reset_index()
weekly_cases_mean_selected_state_NC.head()
```

[65]:
```
        Date  Number of Cases
0 2020-01-19              0.0
1 2020-01-26              0.0
2 2020-02-02              0.0
3 2020-02-09              0.0
4 2020-02-16              0.0
```

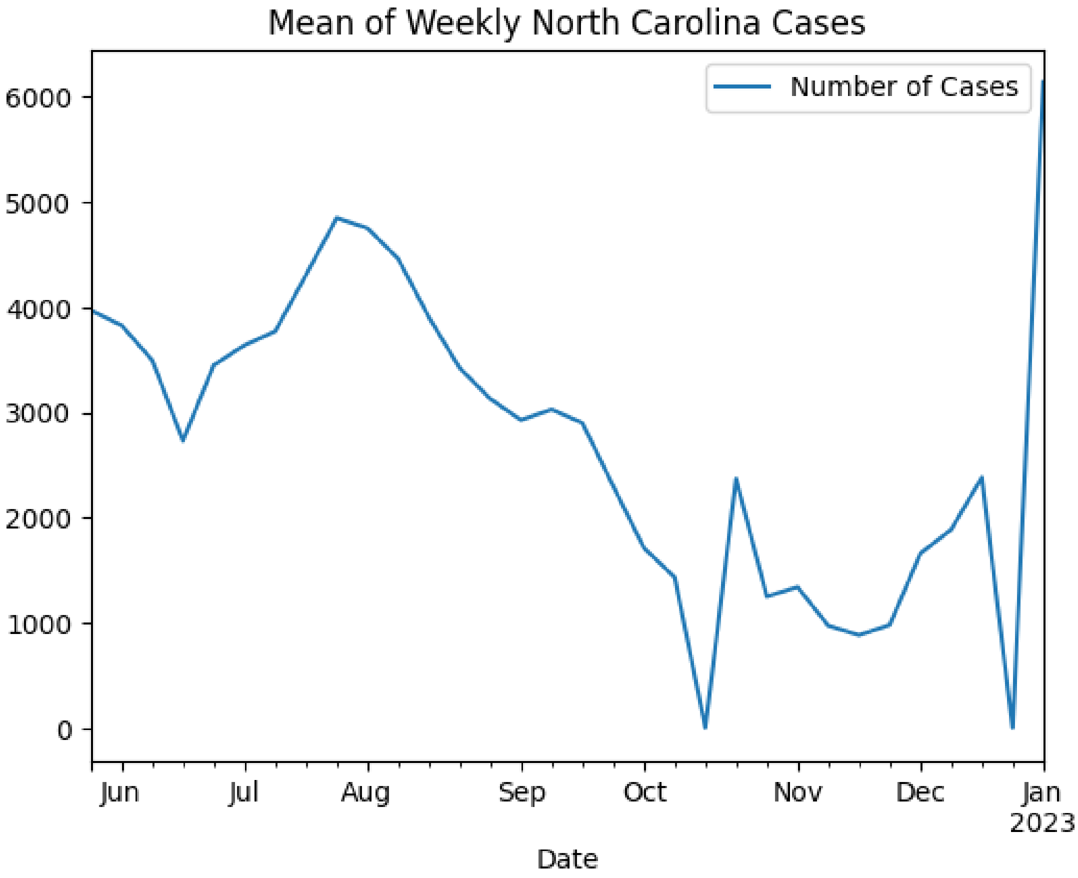
[66]:
```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_cases_mean_selected_state_given_range_NC =␣
 ↪weekly_cases_mean_selected_state_NC[(weekly_cases_mean_selected_state_NC["Date"]␣
 ↪>= '2022-05-29') & (weekly_cases_mean_selected_state_NC["Date"] <=␣
 ↪'2023-01-02')]
weekly_cases_mean_selected_state_given_range_NC =␣
 ↪weekly_cases_mean_selected_state_given_range_NC.sort_values(by=['Date']).
 ↪reset_index(drop=True)
```

```
weekly_cases_mean_selected_state_given_range_NC['Date'] =␣
 ↪weekly_cases_mean_selected_state_given_range_NC['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_cases_mean_selected_state_given_range_NC
```

```
[66]:          Date  Number of Cases
      0  2022-05-30      3970.000000
      1  2022-06-06      3827.142857
      2  2022-06-13      3494.857143
      3  2022-06-20      2728.285714
      4  2022-06-27      3448.714286
      5  2022-07-04      3637.428571
      6  2022-07-11      3769.571429
      7  2022-07-18      4304.428571
      8  2022-07-25      4847.428571
      9  2022-08-01      4752.142857
      10 2022-08-08      4461.428571
      11 2022-08-15      3905.571429
      12 2022-08-22      3422.714286
      13 2022-08-29      3128.285714
      14 2022-09-05      2925.857143
      15 2022-09-12      3027.000000
      16 2022-09-19      2897.714286
      17 2022-09-26      2298.714286
      18 2022-10-03      1710.142857
      19 2022-10-10      1431.571429
      20 2022-10-17         0.000000
      21 2022-10-24      2368.857143
      22 2022-10-31      1247.857143
      23 2022-11-07      1337.857143
      24 2022-11-14       970.714286
      25 2022-11-21       880.857143
      26 2022-11-28       977.428571
      27 2022-12-05      1657.857143
      28 2022-12-12      1884.285714
      29 2022-12-19      2382.285714
      30 2022-12-26         0.000000
      31 2023-01-02      6140.285714
```

```
[67]: weekly_cases_mean_selected_state_given_range_NC.plot(x='Date', y='Number of␣
       ↪Cases', title = 'Mean of Weekly North Carolina Cases')
```

```
[67]: <AxesSubplot: title={'center': 'Mean of Weekly North Carolina Cases'},
      xlabel='Date'>
```



```
[68]: #Converting the daily to weekly analysis and finding the median weekly.
weekly_cases_median_selected_state_NC = new_cases_selected_state_daily_NC.copy()
weekly_cases_median_selected_state_NC['Date'] = pd.
 ↪to_datetime(weekly_cases_median_selected_state_NC['Date']) - pd.
 ↪to_timedelta(7, unit='d')
weekly_cases_median_selected_state_NC = weekly_cases_median_selected_state_NC.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].median()
weekly_cases_median_selected_state_NC = weekly_cases_median_selected_state_NC.
 ↪reset_index()
weekly_cases_median_selected_state_NC.head()
```

```
[68]:         Date  Number of Cases
      0 2020-01-19              0.0
      1 2020-01-26              0.0
      2 2020-02-02              0.0
      3 2020-02-09              0.0
      4 2020-02-16              0.0
```

```
[69]: #considering the given range of dates starting from monday. and weekly analsis␣
       ↪from monday to sunday.
      weekly_cases_median_selected_state_given_range_NC =␣
       ↪weekly_cases_median_selected_state_NC[(weekly_cases_median_selected_state_NC["Date"]␣
       ↪>= '2022-05-29') & (weekly_cases_median_selected_state_NC["Date"] <=␣
       ↪'2023-01-02')]
      weekly_cases_median_selected_state_given_range_NC =␣
       ↪weekly_cases_median_selected_state_given_range_NC.sort_values(by=['Date']).
       ↪reset_index(drop=True)
      weekly_cases_median_selected_state_given_range_NC['Date'] =␣
       ↪weekly_cases_median_selected_state_given_range_NC['Date'] + pd.
       ↪to_timedelta(1, unit='d')
      weekly_cases_median_selected_state_given_range_NC
```

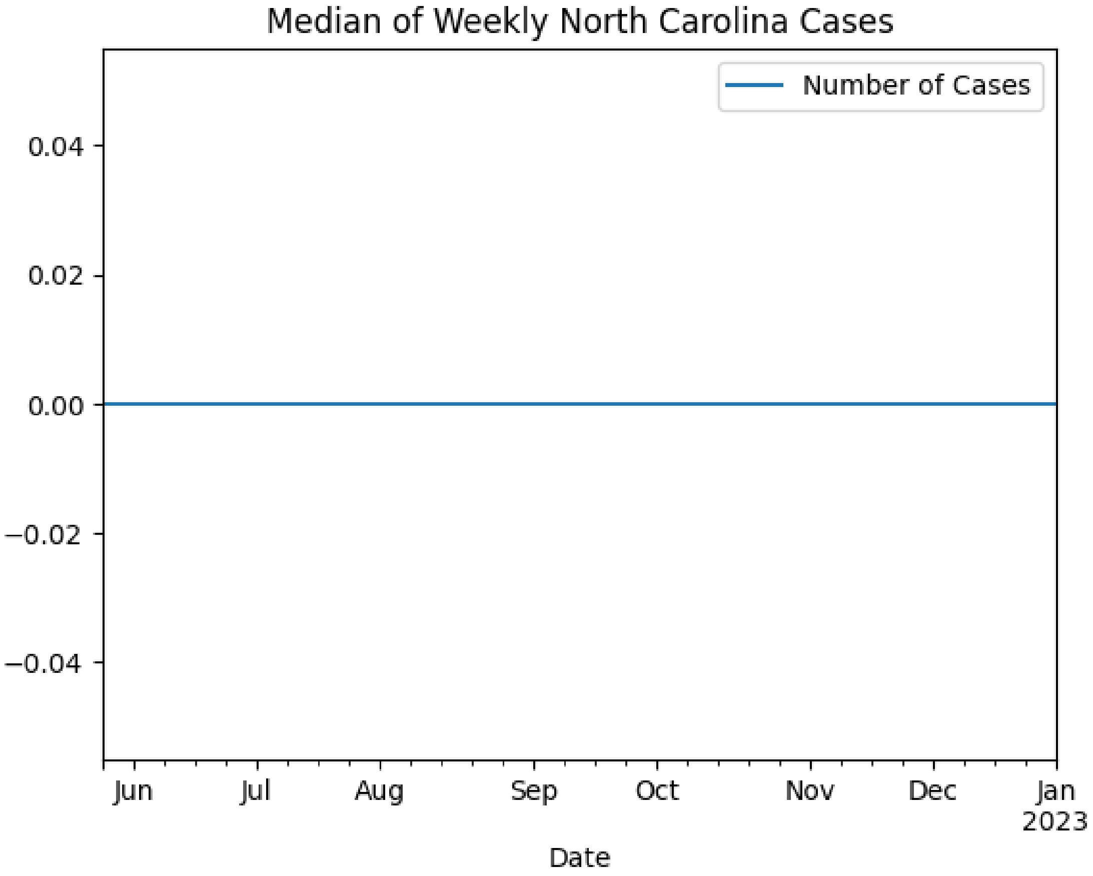
```
[69]:          Date  Number of Cases
      0  2022-05-30              0.0
      1  2022-06-06              0.0
      2  2022-06-13              0.0
      3  2022-06-20              0.0
      4  2022-06-27              0.0
      5  2022-07-04              0.0
      6  2022-07-11              0.0
      7  2022-07-18              0.0
      8  2022-07-25              0.0
      9  2022-08-01              0.0
      10 2022-08-08              0.0
      11 2022-08-15              0.0
      12 2022-08-22              0.0
      13 2022-08-29              0.0
      14 2022-09-05              0.0
      15 2022-09-12              0.0
      16 2022-09-19              0.0
      17 2022-09-26              0.0
      18 2022-10-03              0.0
      19 2022-10-10              0.0
      20 2022-10-17              0.0
      21 2022-10-24              0.0
      22 2022-10-31              0.0
      23 2022-11-07              0.0
      24 2022-11-14              0.0
      25 2022-11-21              0.0
      26 2022-11-28              0.0
      27 2022-12-05              0.0
      28 2022-12-12              0.0
      29 2022-12-19              0.0
      30 2022-12-26              0.0
      31 2023-01-02              0.0
```

[70]:

```
weekly_cases_median_selected_state_given_range_NC.plot(x='Date', y='Number of␣
↪Cases', title = 'Median of Weekly North Carolina Cases')
```

[70]:

```
<AxesSubplot: title={'center': 'Median of Weekly North Carolina Cases'},
xlabel='Date'>
```



[71]:

```
#Converting the daily to weekly analysis and finding the mode weekly.
weekly_cases_mode_selected_state_NC = new_cases_selected_state_daily_NC.copy()
weekly_cases_mode_selected_state_NC['Date'] = pd.
↪to_datetime(weekly_cases_mode_selected_state_NC['Date']) - pd.
↪to_timedelta(7, unit='d')
weekly_cases_mode_selected_state_NC = weekly_cases_mode_selected_state_NC.
↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].
↪apply(statistics.mode)
weekly_cases_mode_selected_state_NC = weekly_cases_mode_selected_state_NC.
↪reset_index()
weekly_cases_mode_selected_state_NC.head()
```

```
[71]:         Date  Number of Cases
      0 2020-01-19              0.0
      1 2020-01-26              0.0
      2 2020-02-02              0.0
      3 2020-02-09              0.0
      4 2020-02-16              0.0
```

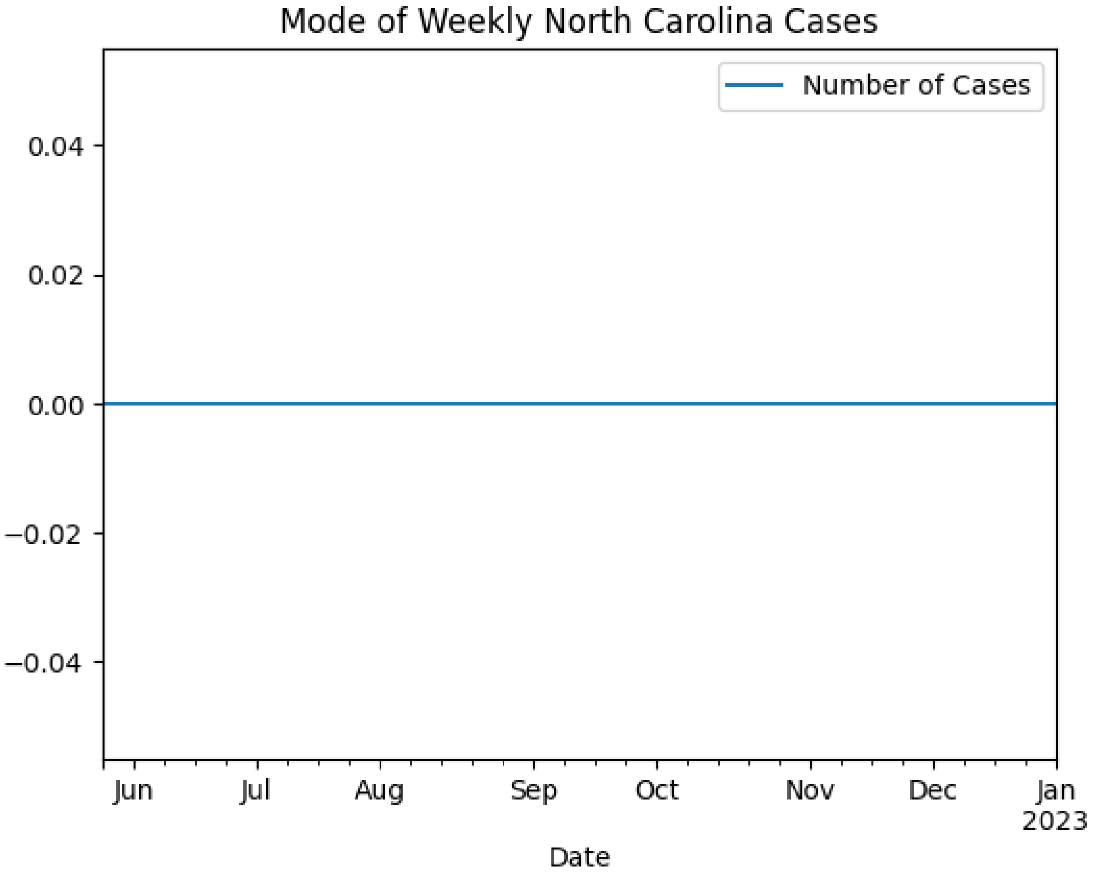
```
[72]: #considering the given range of dates starting from monday. and weekly analsis␣
       ↪from monday to sunday.
      weekly_cases_mode_selected_state_given_range_NC =␣
       ↪weekly_cases_mode_selected_state_NC[(weekly_cases_mode_selected_state_NC["Date"]␣
       ↪>= '2022-05-29') & (weekly_cases_mode_selected_state_NC["Date"] <=␣
       ↪'2023-01-02')]
      weekly_cases_mode_selected_state_given_range_NC =␣
       ↪weekly_cases_mode_selected_state_given_range_NC.sort_values(by=['Date']).
       ↪reset_index(drop=True)
      weekly_cases_mode_selected_state_given_range_NC['Date'] =␣
       ↪weekly_cases_mode_selected_state_given_range_NC['Date'] + pd.to_timedelta(1,␣
       ↪unit='d')
      weekly_cases_mode_selected_state_given_range_NC
```

```
[72]:          Date  Number of Cases
      0  2022-05-30              0.0
      1  2022-06-06              0.0
      2  2022-06-13              0.0
      3  2022-06-20              0.0
      4  2022-06-27              0.0
      5  2022-07-04              0.0
      6  2022-07-11              0.0
      7  2022-07-18              0.0
      8  2022-07-25              0.0
      9  2022-08-01              0.0
      10 2022-08-08              0.0
      11 2022-08-15              0.0
      12 2022-08-22              0.0
      13 2022-08-29              0.0
      14 2022-09-05              0.0
      15 2022-09-12              0.0
      16 2022-09-19              0.0
      17 2022-09-26              0.0
      18 2022-10-03              0.0
      19 2022-10-10              0.0
      20 2022-10-17              0.0
      21 2022-10-24              0.0
      22 2022-10-31              0.0
      23 2022-11-07              0.0
      24 2022-11-14              0.0
```

```
25 2022-11-21              0.0
26 2022-11-28              0.0
27 2022-12-05              0.0
28 2022-12-12              0.0
29 2022-12-19              0.0
30 2022-12-26              0.0
31 2023-01-02              0.0
```

[73]:
```
weekly_cases_mode_selected_state_given_range_NC.plot(x='Date', y='Number of␣
↪Cases', title = 'Mode of Weekly North Carolina Cases')
```

[73]:
```
<AxesSubplot: title={'center': 'Mode of Weekly North Carolina Cases'},
xlabel='Date'>
```



[74]:
```
#Converting the daily to weekly analysis and finding the weekly sum of cases.
weekly_cases_sum_selected_state_NC = new_cases_selected_state_daily_NC.copy()
weekly_cases_sum_selected_state_NC['Date'] = pd.
↪to_datetime(weekly_cases_sum_selected_state_NC['Date']) - pd.to_timedelta(7,␣
↪unit='d')
```

```
weekly_cases_sum_selected_state_NC = weekly_cases_sum_selected_state_NC.
 ↪groupby([pd.Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
weekly_cases_sum_selected_state_NC = weekly_cases_sum_selected_state_NC.
 ↪reset_index()
weekly_cases_sum_selected_state_NC.head()
```

```
[74]:         Date  Number of Cases
      0 2020-01-19              0.0
      1 2020-01-26              0.0
      2 2020-02-02              0.0
      3 2020-02-09              0.0
      4 2020-02-16              0.0
```

```
[75]: #Adding one day so that we get weekly analysis from monday to sunday and weekly␣
       ↪analsis from monday to sunday.
      weekly_cases_sum_selected_state_given_range_NC =␣
       ↪weekly_cases_sum_selected_state_NC[(weekly_cases_sum_selected_state_NC["Date"]␣
       ↪>= '2022-05-29') & (weekly_cases_sum_selected_state_NC["Date"] <=␣
       ↪'2023-01-02')]
      weekly_cases_sum_selected_state_given_range_NC =␣
       ↪weekly_cases_sum_selected_state_given_range_NC.sort_values(by=['Date']).
       ↪reset_index(drop=True)
      weekly_cases_sum_selected_state_given_range_NC['Date'] =␣
       ↪weekly_cases_sum_selected_state_given_range_NC['Date'] + pd.to_timedelta(1,␣
       ↪unit='d')
      weekly_cases_sum_selected_state_given_range_NC
```

```
[75]:          Date  Number of Cases
      0  2022-05-30          27790.0
      1  2022-06-06          26790.0
      2  2022-06-13          24464.0
      3  2022-06-20          19098.0
      4  2022-06-27          24141.0
      5  2022-07-04          25462.0
      6  2022-07-11          26387.0
      7  2022-07-18          30131.0
      8  2022-07-25          33932.0
      9  2022-08-01          33265.0
      10 2022-08-08          31230.0
      11 2022-08-15          27339.0
      12 2022-08-22          23959.0
      13 2022-08-29          21898.0
      14 2022-09-05          20481.0
      15 2022-09-12          21189.0
      16 2022-09-19          20284.0
      17 2022-09-26          16091.0
      18 2022-10-03          11971.0
```

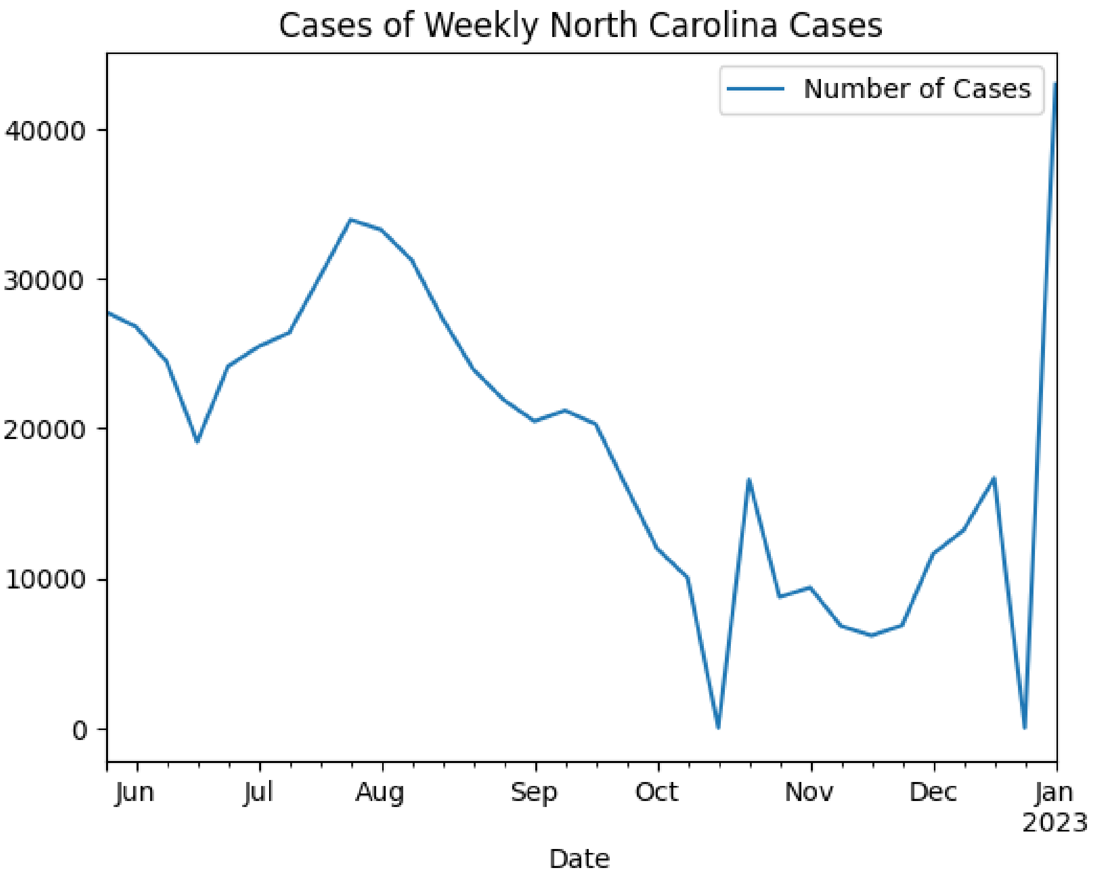
```
19 2022-10-10          10021.0
20 2022-10-17              0.0
21 2022-10-24          16582.0
22 2022-10-31           8735.0
23 2022-11-07           9365.0
24 2022-11-14           6795.0
25 2022-11-21           6166.0
26 2022-11-28           6842.0
27 2022-12-05          11605.0
28 2022-12-12          13190.0
29 2022-12-19          16676.0
30 2022-12-26              0.0
31 2023-01-02          42982.0
```

[76]:

```
weekly_cases_sum_selected_state_given_range_NC.plot(x='Date', y='Number of␣
↪Cases', title = 'Cases of Weekly North Carolina Cases')
```

[76]:

```
<AxesSubplot: title={'center': 'Cases of Weekly North Carolina Cases'},
xlabel='Date'>
```



0.1.4 Week starting with 2023-01-02 has a peak of cases in North Carolina with value 42982. the peak was beacuse of the new year and christmas holiday. Pepole may have gathered and had celebrations.

```
[77]: #Filtering the population for NorthCarolina state
      population_selected_state_NC = population[population["State"] ==␣
       ↪selected_state_NC].reset_index()
      del population_selected_state_NC[population_selected_state_NC.columns[0]]
      population_selected_state_NC
```

```
[77]:      countyFIPS             County Name State  population
      0             0  Statewide Unallocated    NC           0
      1         37001         Alamance County    NC      169509
      2         37003        Alexander County    NC       37497
      3         37005        Alleghany County    NC       11137
      4         37007            Anson County    NC       24446
      ..          …                      …    …           …
      96        37191            Wayne County    NC      123131
      97        37193           Wilkes County    NC       68412
      98        37195           Wilson County    NC       81801
      99        37197           Yadkin County    NC       37667
      100       37199           Yancey County    NC       18069

      [101 rows x 4 columns]
```

```
[78]: #For the Albama state summing the population.
      population_selected_state_NC = population_selected_state_NC.
       ↪groupby('State')['population'].sum()
      population_selected_state_NC
```

```
[78]: State
      NC    10488084
      Name: population, dtype: int64
```

```
[79]: #normalizing by population and using normalization factor to identify cases
      normalized_weekly_cases_sum_selected_state_given_range_NC =␣
       ↪weekly_cases_sum_selected_state_given_range_NC.copy()
      normalized_weekly_cases_sum_selected_state_given_range_NC['Number of Cases'] =␣
       ↪normalized_weekly_cases_sum_selected_state_given_range_NC['Number of Cases'].
       ↪mul(10000)
      normalized_weekly_cases_sum_selected_state_given_range_NC['Number of Cases'] =␣
       ↪normalized_weekly_cases_sum_selected_state_given_range_NC['Number of Cases'].
       ↪div(10488084)
      normalized_weekly_cases_sum_selected_state_given_range_NC
```

```
[79]:          Date  Number of Cases
      0  2022-05-30        26.496737
```

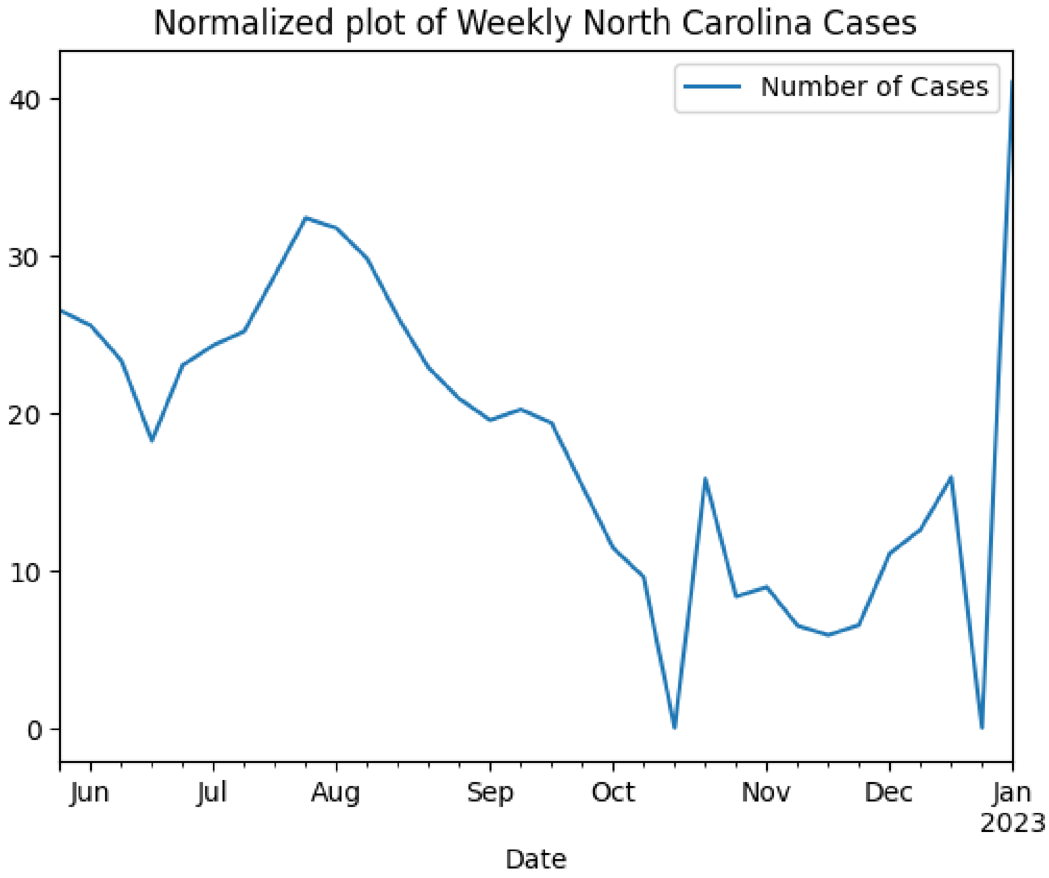
```
1   2022-06-06        25.543274
2   2022-06-13        23.325519
3   2022-06-20        18.209236
4   2022-06-27        23.017550
5   2022-07-04        24.277075
6   2022-07-11        25.159028
7   2022-07-18        28.728794
8   2022-07-25        32.352906
9   2022-08-01        31.716947
10  2022-08-08        29.776649
11  2022-08-15        26.066725
12  2022-08-22        22.844020
13  2022-08-29        20.878933
14  2022-09-05        19.527876
15  2022-09-12        20.202927
16  2022-09-19        19.340043
17  2022-09-26        15.342173
18  2022-10-03        11.413906
19  2022-10-10         9.554653
20  2022-10-17         0.000000
21  2022-10-24        15.810323
22  2022-10-31         8.328499
23  2022-11-07         8.929181
24  2022-11-14         6.478781
25  2022-11-21         5.879053
26  2022-11-28         6.523594
27  2022-12-05        11.064938
28  2022-12-12        12.576177
29  2022-12-19        15.899949
30  2022-12-26         0.000000
31  2023-01-02        40.981747
```

[80]:

```
normalized_weekly_cases_sum_selected_state_given_range_NC.plot(x='Date',␣
 ↪y='Number of Cases', title = 'Normalized plot of Weekly North Carolina␣
 ↪Cases')
```

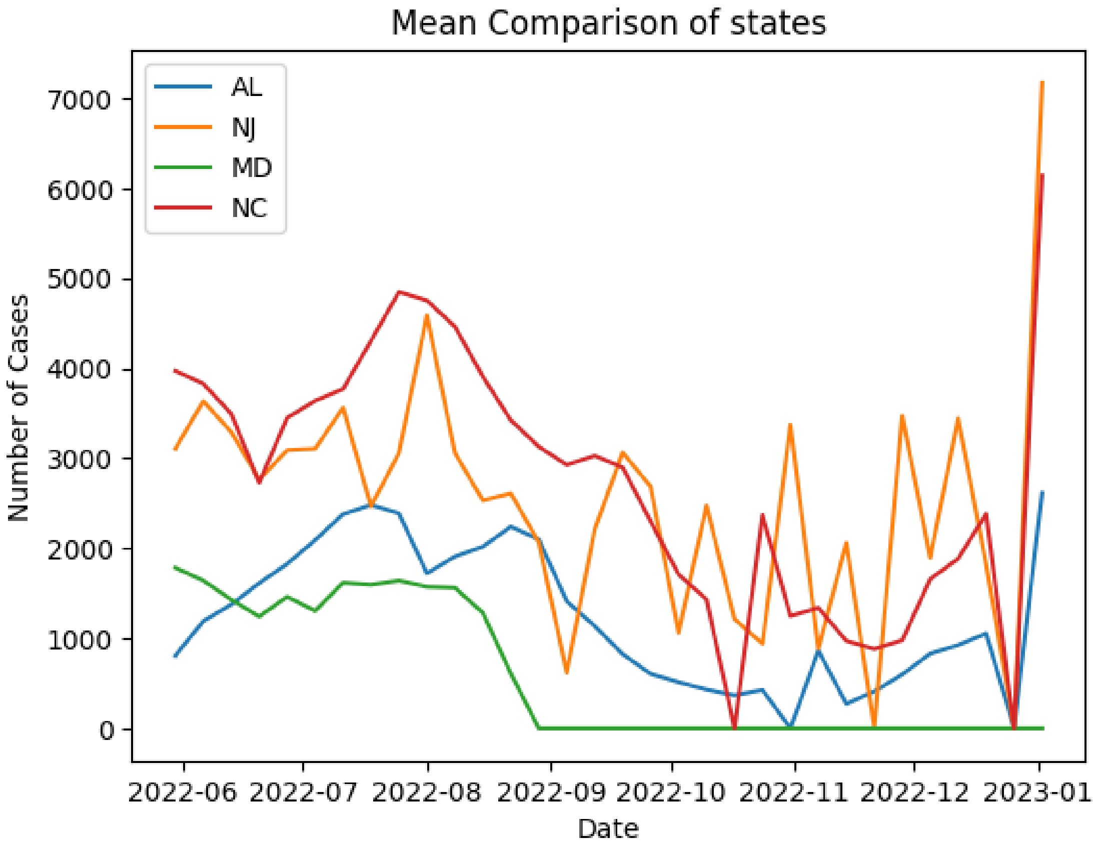
[80]:

```
<AxesSubplot: title={'center': 'Normalized plot of Weekly North Carolina
Cases'}, xlabel='Date'>
```



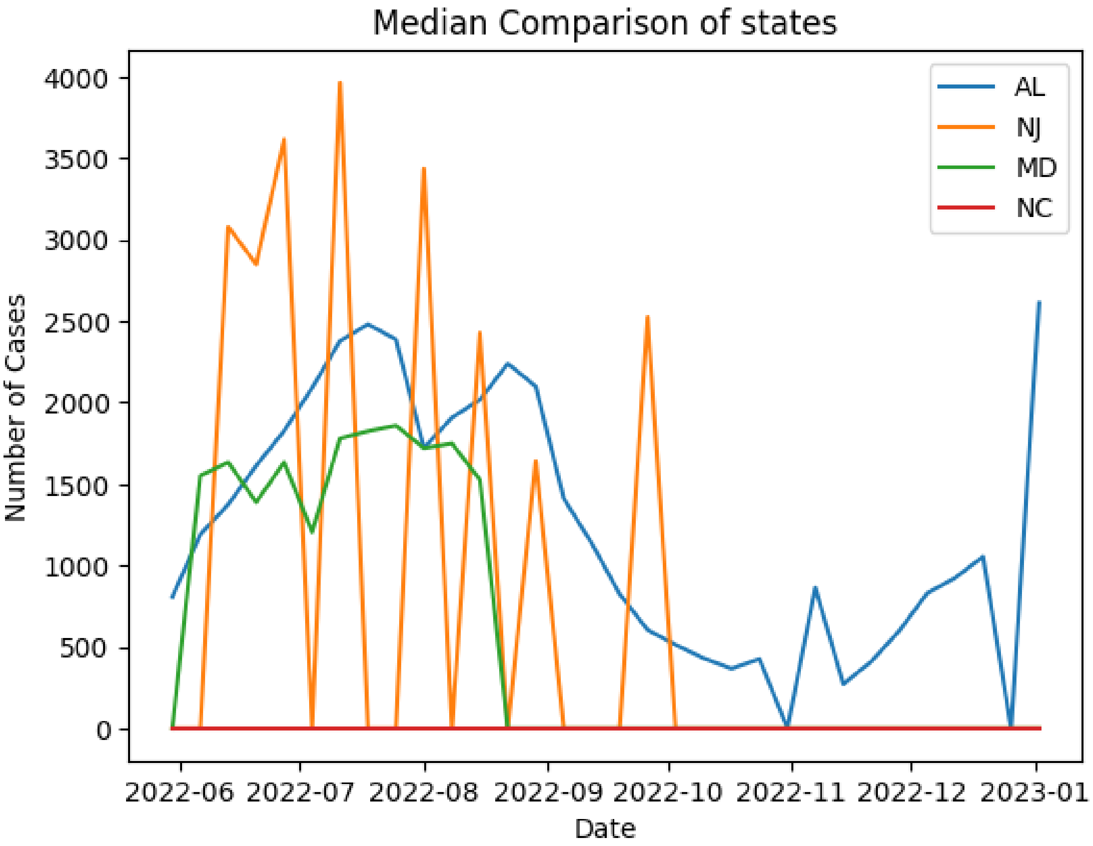
[81]:

```
#Mean comparision of three states with alabama state.
plt.plot(weekly_cases_mean_selected_state_given_range['Date'],␣
 ↪weekly_cases_mean_selected_state_given_range['Number of Cases'], label='AL')
plt.plot(weekly_cases_mean_selected_state_given_range_NJ['Date'],␣
 ↪weekly_cases_mean_selected_state_given_range_NJ['Number of Cases'],␣
 ↪label='NJ')
plt.plot(weekly_cases_mean_selected_state_given_range_MD['Date'],␣
 ↪weekly_cases_mean_selected_state_given_range_MD['Number of Cases'],␣
 ↪label='MD')
plt.plot(weekly_cases_mean_selected_state_given_range_NC['Date'],␣
 ↪weekly_cases_mean_selected_state_given_range_NC['Number of Cases'],␣
 ↪label='NC')
plt.title('Mean Comparison of states')
plt.xlabel('Date')
plt.ylabel('Number of Cases')
plt.legend()
plt.show()
```



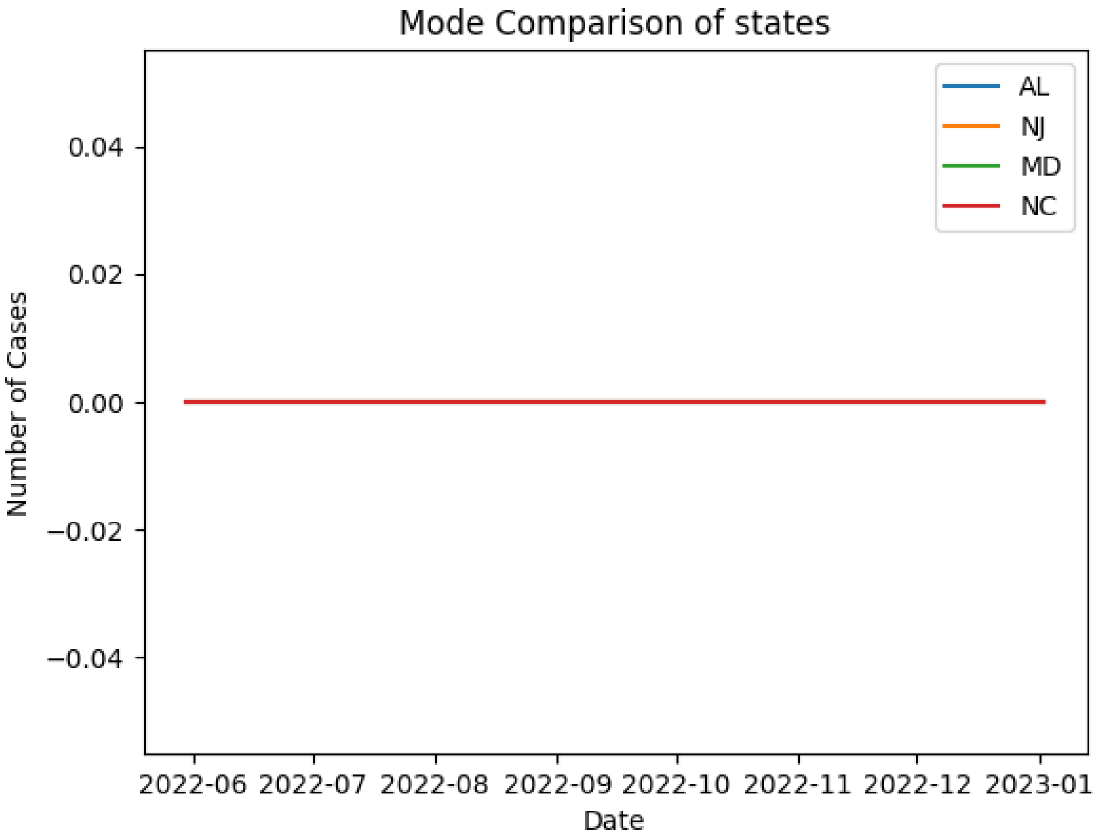
[82]:

```
#Median comparision of three states with alabama state.
plt.plot(weekly_cases_median_selected_state_given_range['Date'],␣
 ↪weekly_cases_mean_selected_state_given_range['Number of Cases'], label='AL')
plt.plot(weekly_cases_median_selected_state_given_range_NJ['Date'],␣
 ↪weekly_cases_median_selected_state_given_range_NJ['Number of Cases'],␣
 ↪label='NJ')
plt.plot(weekly_cases_median_selected_state_given_range_MD['Date'],␣
 ↪weekly_cases_median_selected_state_given_range_MD['Number of Cases'],␣
 ↪label='MD')
plt.plot(weekly_cases_median_selected_state_given_range_NC['Date'],␣
 ↪weekly_cases_median_selected_state_given_range_NC['Number of Cases'],␣
 ↪label='NC')
plt.title('Median Comparison of states')
plt.xlabel('Date')
plt.ylabel('Number of Cases')
plt.legend()
plt.show()
```



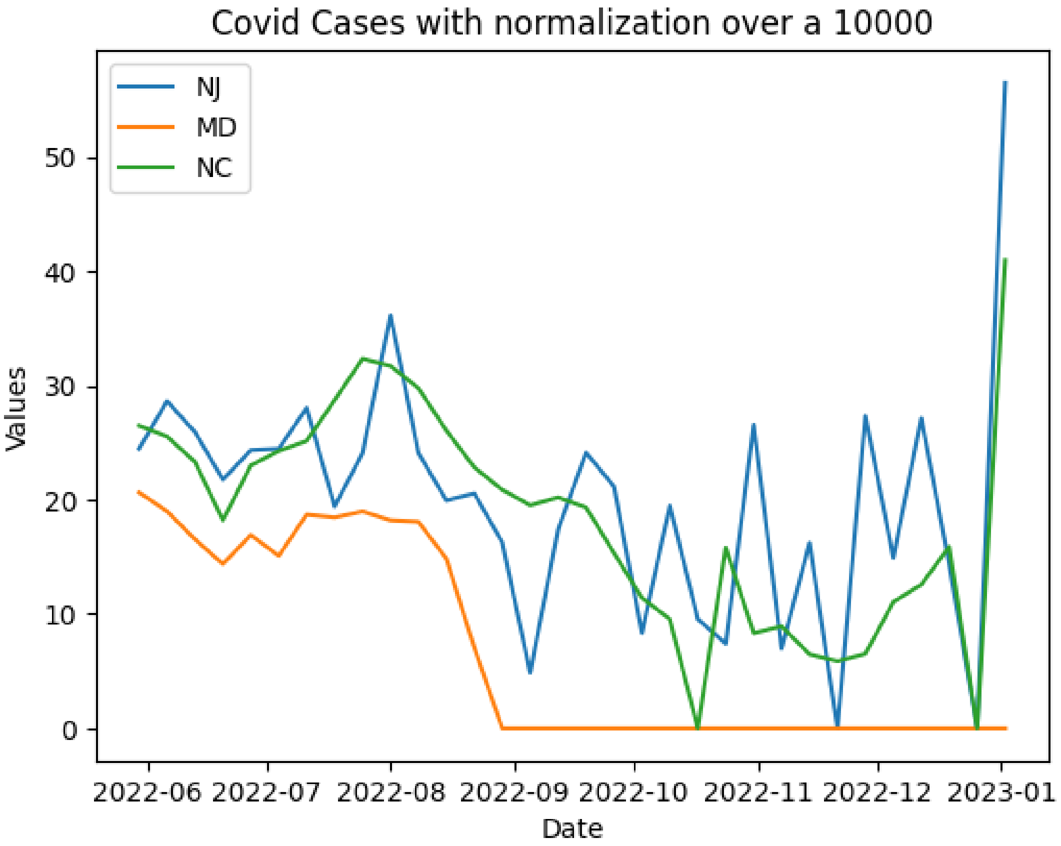
[83]:

```
#Mode comparision of three states with alabama state.
plt.plot(weekly_cases_mode_selected_state_given_range['Date'],␣
 ↪weekly_cases_mode_selected_state_given_range['Number of Cases'], label='AL')
plt.plot(weekly_cases_mode_selected_state_given_range_NJ['Date'],␣
 ↪weekly_cases_mode_selected_state_given_range_NJ['Number of Cases'],␣
 ↪label='NJ')
plt.plot(weekly_cases_mode_selected_state_given_range_MD['Date'],␣
 ↪weekly_cases_mode_selected_state_given_range_MD['Number of Cases'],␣
 ↪label='MD')
plt.plot(weekly_cases_mode_selected_state_given_range_NC['Date'],␣
 ↪weekly_cases_mode_selected_state_given_range_NC['Number of Cases'],␣
 ↪label='NC')
plt.title('Mode Comparison of states')
plt.xlabel('Date')
plt.ylabel('Number of Cases')
plt.legend()
plt.show()
```



[84]:

```
#Normalization plot of three states(Newjersey,Maryland,NorthCarolina).
plt.plot(normalized_weekly_cases_sum_selected_state_given_range_NJ['Date'],␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_NJ['Number of␣
 ↪Cases'], label='NJ')
plt.plot(normalized_weekly_cases_sum_selected_state_given_range_MD['Date'],␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_MD['Number of␣
 ↪Cases'], label='MD')
plt.plot(normalized_weekly_cases_sum_selected_state_given_range_NC['Date'],␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_NC['Number of␣
 ↪Cases'], label='NC')
plt.title('Covid Cases with normalization over a 10000')
plt.xlabel('Date')
plt.ylabel('Values')
plt.legend()
plt.show()
```



[85]:

```
#For the country summing the cases per day of all the counties.
cases_transpose = cases_transpose.groupby('Date')['Number of Cases'].sum()
cases_transpose.head()
```

[85]:

```
Date
2020-01-22    723
2020-01-23    734
2020-01-24    741
2020-01-25    751
2020-01-26    759
Name: Number of Cases, dtype: int64
```

[86]:

```
#Finding out the new cases per day.
new_cases_country_daily = cases_transpose.diff().reset_index()
new_cases_country_daily.head()
```

[86]:

```
         Date  Number of Cases
0  2020-01-22              NaN
1  2020-01-23             11.0
2  2020-01-24              7.0
```

```
3  2020-01-25             10.0
4  2020-01-26              8.0
```

[87]:
```
#Converting the daily to weekly analysis and finding the mean weekly.
weekly_cases_sum_country = new_cases_country_daily.copy()
weekly_cases_sum_country['Date'] = pd.
 ↪to_datetime(weekly_cases_sum_country['Date']) - pd.to_timedelta(7, unit='d')
weekly_cases_sum_country = weekly_cases_sum_country.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
weekly_cases_sum_country = weekly_cases_sum_country.reset_index()
weekly_cases_sum_country.head()
```

[87]:
```
        Date  Number of Cases
0 2020-01-19             36.0
1 2020-01-26           -727.0
2 2020-02-02             63.0
3 2020-02-09             49.0
4 2020-02-16            102.0
```

[88]:
```
#Adding one day so that we get weekly analysis from monday to sunday and weekly␣
 ↪analsis from monday to sunday.
weekly_cases_country_given_range =␣
 ↪weekly_cases_sum_country[(weekly_cases_sum_country["Date"] >= '2022-05-29')␣
 ↪& (weekly_cases_sum_country["Date"] <= '2023-01-02')]
weekly_cases_country_given_range = weekly_cases_country_given_range.
 ↪sort_values(by=['Date']).reset_index(drop=True)
weekly_cases_country_given_range['Date'] =␣
 ↪weekly_cases_country_given_range['Date'] + pd.to_timedelta(1, unit='d')
weekly_cases_country_given_range
```

[88]:
```
         Date  Number of Cases
0  2022-05-30         596616.0
1  2022-06-06         607086.0
2  2022-06-13         720738.0
3  2022-06-20         524811.0
4  2022-06-27         715190.0
5  2022-07-04         573480.0
6  2022-07-11         752889.0
7  2022-07-18         647246.0
8  2022-07-25         789033.0
9  2022-08-01         584134.0
10 2022-08-08         638027.0
11 2022-08-15         493742.0
12 2022-08-22         386887.0
13 2022-08-29         -26293.0
14 2022-09-05         510896.0
15 2022-09-12         265845.0
```

```
16 2022-09-19         360031.0
17 2022-09-26         227630.0
18 2022-10-03         245389.0
19 2022-10-10         229679.0
20 2022-10-17         177433.0
21 2022-10-24         217652.0
22 2022-10-31         220576.0
23 2022-11-07         196964.0
24 2022-11-14         222255.0
25 2022-11-21         210368.0
26 2022-11-28         310649.0
27 2022-12-05         320264.0
28 2022-12-12         326894.0
29 2022-12-19         299191.0
30 2022-12-26         241219.0
31 2023-01-02         397549.0
```

[89]:

```
#For the country summing the population.
population_country = population['population'].sum()
population_country
```

[89]:

```
328239523
```

[90]:

```
#normalizing by population and using normalization factor to identify cases
normalized_weekly_cases_sum_country_given_range =␣
 ↪weekly_cases_country_given_range.copy()
normalized_weekly_cases_sum_country_given_range['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_country_given_range['Number of Cases'].mul(10000)
normalized_weekly_cases_sum_country_given_range['Number of Cases'] =␣
 ↪normalized_weekly_cases_sum_country_given_range['Number of Cases'].
 ↪div(328239523)
normalized_weekly_cases_sum_country_given_range
```

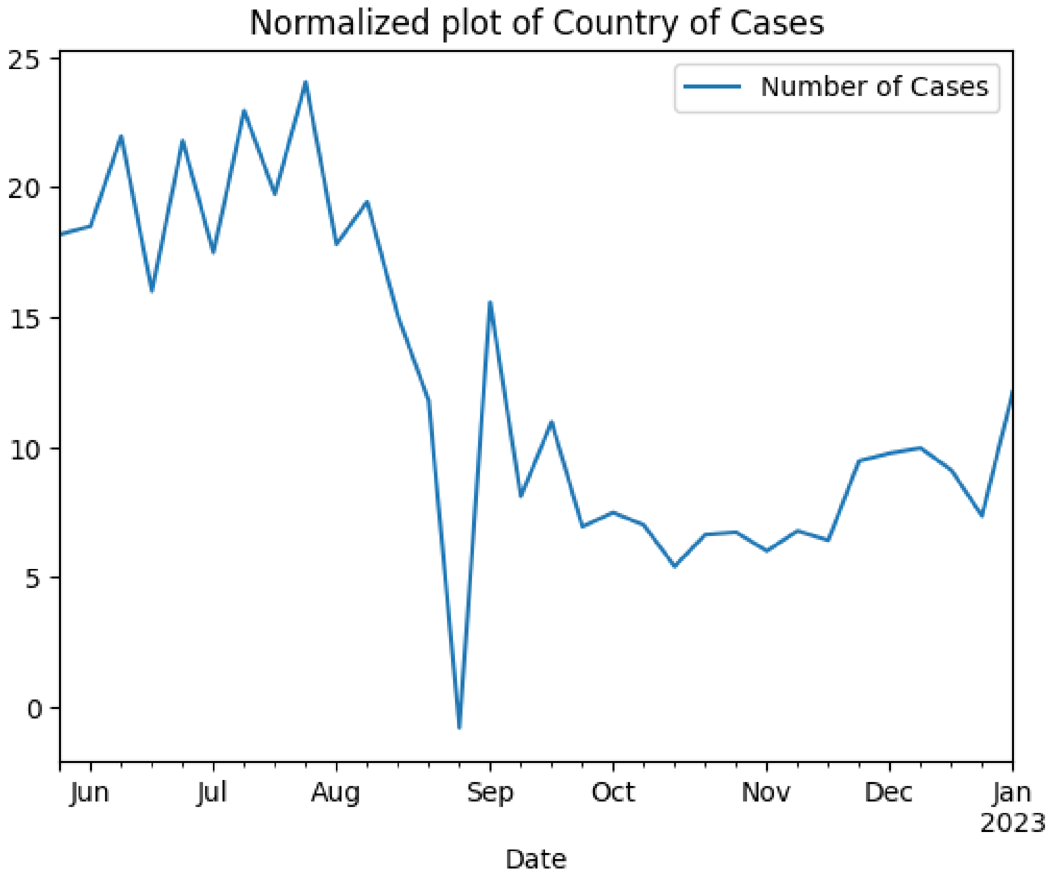
[90]:

```
          Date  Number of Cases
0  2022-05-30        18.176239
1  2022-06-06        18.495213
2  2022-06-13        21.957685
3  2022-06-20        15.988660
4  2022-06-27        21.788662
5  2022-07-04        17.471388
6  2022-07-11        22.937183
7  2022-07-18        19.718710
8  2022-07-25        24.038330
9  2022-08-01        17.795968
10 2022-08-08        19.437848
11 2022-08-15        15.042125
12 2022-08-22        11.786728
```

```
13 2022-08-29        -0.801031
14 2022-09-05        15.564731
15 2022-09-12         8.099116
16 2022-09-19        10.968545
17 2022-09-26         6.934875
18 2022-10-03         7.475913
19 2022-10-10         6.997299
20 2022-10-17         5.405595
21 2022-10-24         6.630889
22 2022-10-31         6.719971
23 2022-11-07         6.000618
24 2022-11-14         6.771122
25 2022-11-21         6.408978
26 2022-11-28         9.464095
27 2022-12-05         9.757021
28 2022-12-12         9.959008
29 2022-12-19         9.115021
30 2022-12-26         7.348871
31 2023-01-02        12.111552
```

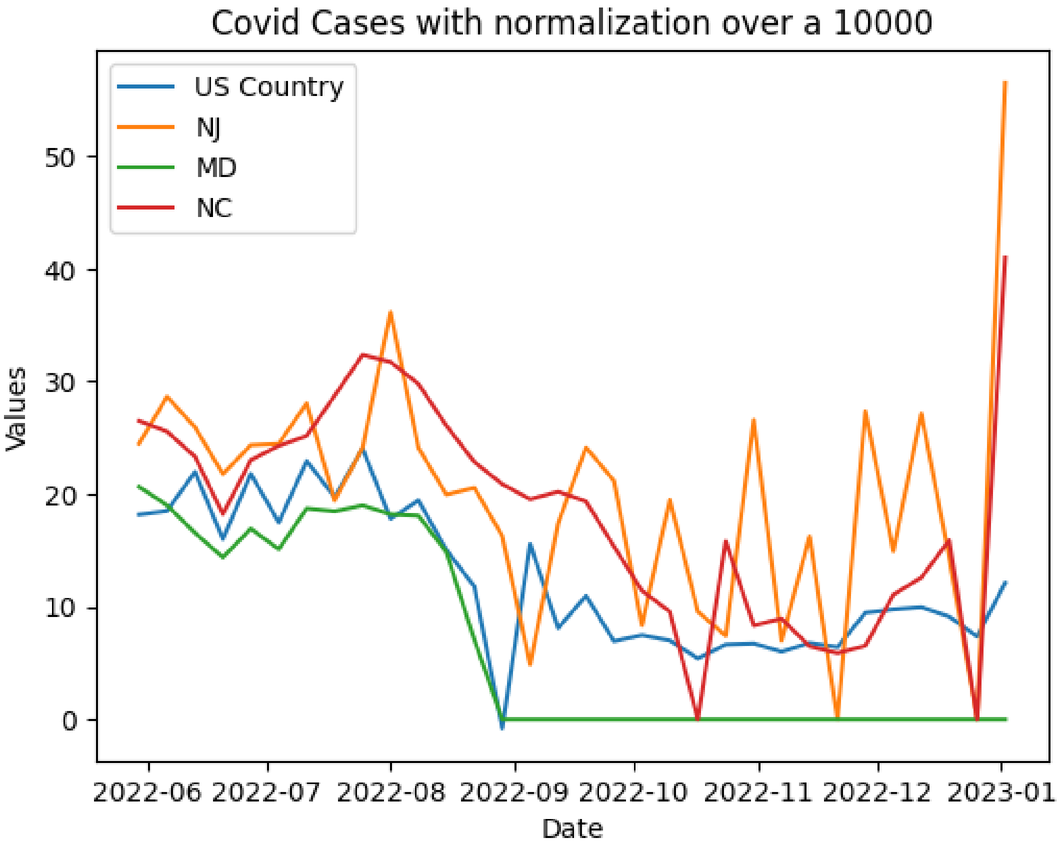
[91]:
```
normalized_weekly_cases_sum_country_given_range.plot(x='Date', y='Number of␣
 ↪Cases', title = 'Normalized plot of Country of Cases')
```

[91]:
```
<AxesSubplot: title={'center': 'Normalized plot of Country of Cases'},
xlabel='Date'>
```



[92]:

```
#Normalied Plot of US country and the three states.
plt.plot(normalized_weekly_cases_sum_country_given_range['Date'],␣
 ↪normalized_weekly_cases_sum_country_given_range['Number of Cases'],␣
 ↪label='US Country')
plt.plot(normalized_weekly_cases_sum_selected_state_given_range_NJ['Date'],␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_NJ['Number of␣
 ↪Cases'], label='NJ')
plt.plot(normalized_weekly_cases_sum_selected_state_given_range_MD['Date'],␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_MD['Number of␣
 ↪Cases'], label='MD')
plt.plot(normalized_weekly_cases_sum_selected_state_given_range_NC['Date'],␣
 ↪normalized_weekly_cases_sum_selected_state_given_range_NC['Number of␣
 ↪Cases'], label='NC')
plt.title('Covid Cases with normalization over a 10000')
plt.xlabel('Date')
plt.ylabel('Values')
plt.legend()
plt.show()
```



0.1.5 Here the two states(Newjersey(NJ)and North Carolina(NC) trends are following the country trends but Maryland(MD) state is not following the country trends.

0.1.6 Maryland shows zero cases from september 2022 this is because of non availability of data as I researched and found cases were not zero.

0.1.7 Nj has high rate of cases compared to NC might be because of more population.