

Stage2_Task4_Cases

March 14, 2023

0.1 Plot weekly trends (new cases) for the top 3 infected counties. Show plots by raw values and log normalized values. Describe what is causing them and what were the peaks. Do the counties follow state pattern

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

```
[2]: #I have selected the Alabama state for analysis
selected_state = "AL"
selected_county_1 = "Jefferson County "
selected_county_2 = "Madison County "
selected_county_3 = "Mobile County "
# 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 and
      ↪merging will be easy with enrichment data.
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]: cases_selected_state = cases_transpose[cases_transpose["State"] ==
      ↪selected_state]
cases_selected_state
```

```
[4]:   countyFIPS   County Name State StateFIPS   Date \
1      1001   Autauga County    AL         1  2020-01-22
2      1003   Baldwin County    AL         1  2020-01-22
3      1005   Barbour County    AL         1  2020-01-22
4      1007     Bibb County    AL         1  2020-01-22
5      1009   Blount County    AL         1  2020-01-22
...
3480433   1125  Tuscaloosa County    AL         1  2023-01-16
3480434   1127     Walker County    AL         1  2023-01-16
3480435   1129  Washington County    AL         1  2023-01-16
3480436   1131     Wilcox County    AL         1  2023-01-16
3480437   1133   Winston County    AL         1  2023-01-16

      Number of Cases
1                  0
2                  0
3                  0
4                  0
5                  0
```

```
...
3480433      68860
3480434      23425
3480435       4309
3480436       3569
3480437       9200
```

[73097 rows x 6 columns]

```
[5]: cases_selected_county = cases_selected_state[cases_selected_state["County_
↳Name"] == selected_county_1].reset_index()
del cases_selected_county[cases_selected_county.columns[0]]

cases_selected_county
```

```
[5]:      countyFIPS      County Name State  StateFIPS      Date \
0          1073  Jefferson County    AL          1  2020-01-22
1          1073  Jefferson County    AL          1  2020-01-23
2          1073  Jefferson County    AL          1  2020-01-24
3          1073  Jefferson County    AL          1  2020-01-25
4          1073  Jefferson County    AL          1  2020-01-26
...
1086        1073  Jefferson County    AL          1  2023-01-12
1087        1073  Jefferson County    AL          1  2023-01-13
1088        1073  Jefferson County    AL          1  2023-01-14
1089        1073  Jefferson County    AL          1  2023-01-15
1090        1073  Jefferson County    AL          1  2023-01-16
```

```
      Number of Cases
0              0
1              0
2              0
3              0
4              0
...
1086          229633
1087          229633
1088          229633
1089          229633
1090          229633
```

[1091 rows x 6 columns]

```
[6]: #For the selected state Alabama summing the deaths per day of all the counties.
cases_selected_county_daily = cases_selected_county.groupby('Date')['Number of_
↳Cases'].sum()
cases_selected_county_daily
```

```
[6]: Date
2020-01-22      0
2020-01-23      0
2020-01-24      0
2020-01-25      0
2020-01-26      0
...
2023-01-12    229633
2023-01-13    229633
2023-01-14    229633
2023-01-15    229633
2023-01-16    229633
Name: Number of Cases, Length: 1091, dtype: int64
```

```
[7]: #Finding out the new cases per day.
new_cases_selected_county_daily = cases_selected_county_daily.diff().
↳reset_index()
new_cases_selected_county_daily
```

```
[7]:      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
...
1086  2023-01-12    1133.0
1087  2023-01-13      0.0
1088  2023-01-14      0.0
1089  2023-01-15      0.0
1090  2023-01-16      0.0
```

[1091 rows x 2 columns]

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

```
[8]:      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

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

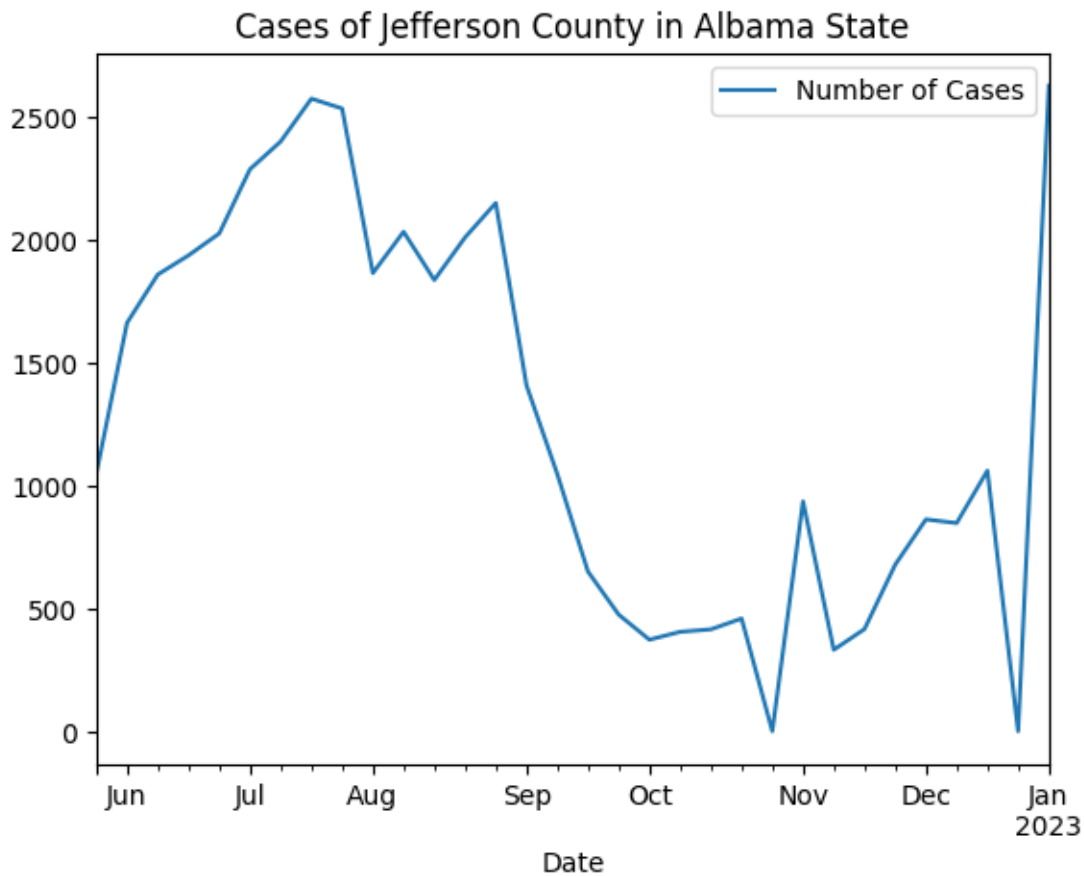
```
[9]:
```

	Date	Number of Cases
0	2022-05-30	1047.0
1	2022-06-06	1660.0
2	2022-06-13	1856.0
3	2022-06-20	1934.0
4	2022-06-27	2023.0
5	2022-07-04	2284.0
6	2022-07-11	2397.0
7	2022-07-18	2571.0
8	2022-07-25	2531.0
9	2022-08-01	1862.0
10	2022-08-08	2030.0
11	2022-08-15	1834.0
12	2022-08-22	2007.0
13	2022-08-29	2147.0
14	2022-09-05	1406.0
15	2022-09-12	1046.0
16	2022-09-19	649.0
17	2022-09-26	474.0
18	2022-10-03	372.0
19	2022-10-10	404.0
20	2022-10-17	414.0
21	2022-10-24	458.0
22	2022-10-31	0.0
23	2022-11-07	934.0
24	2022-11-14	331.0
25	2022-11-21	415.0
26	2022-11-28	677.0
27	2022-12-05	861.0
28	2022-12-12	846.0

29	2022-12-19	1059.0
30	2022-12-26	0.0
31	2023-01-02	2624.0

```
[10]: weekly_cases_selected_county_given_range.plot(x='Date', y='Number of Cases',
→title = 'Cases of Jefferson County in Albama State')
```

```
[10]: <AxesSubplot: title={'center': 'Cases of Jefferson County in Albama State'},
xlabel='Date'>
```



0.1.1 Week starting with 2023-01-02 has a peak of cases in jefferson county with value 2624, cause can be new year long weekend and party which made lots of people to gather. Other spike is in July, cause can be independance day long weekend, people might have gathered to celebrate.

```
[11]: weekly_cases_selected_county_given_range_max_normalised =
→weekly_cases_selected_county_given_range.copy()
for column in weekly_cases_selected_county_given_range_max_normalised.columns:
```

```

weekly_cases_selected_county_given_range_max_normalised['Number of Cases']_
↪= np.log(weekly_cases_selected_county_given_range_max_normalised['Number of_
↪Cases'] + 1 ) / np.
↪log(weekly_cases_selected_county_given_range_max_normalised['Number of_
↪Cases'].max() + 1)
display(weekly_cases_selected_county_given_range_max_normalised)

```

	Date	Number of Cases
0	2022-05-30	0.913318
1	2022-06-06	0.957445
2	2022-06-13	0.967933
3	2022-06-20	0.971783
4	2022-06-27	0.975978
5	2022-07-04	0.987234
6	2022-07-11	0.991689
7	2022-07-18	0.998130
8	2022-07-25	0.996691
9	2022-08-01	0.968235
10	2022-08-08	0.976300
11	2022-08-15	0.966816
12	2022-08-22	0.975239
13	2022-08-29	0.981507
14	2022-09-05	0.941698
15	2022-09-12	0.913225
16	2022-09-19	0.866076
17	2022-09-26	0.834192
18	2022-10-03	0.809129
19	2022-10-10	0.817712
20	2022-10-17	0.820245
21	2022-10-24	0.830666
22	2022-10-31	0.000000
23	2022-11-07	0.902174
24	2022-11-14	0.796899
25	2022-11-21	0.820495
26	2022-11-28	0.870310
27	2022-12-05	0.894181
28	2022-12-12	0.892449
29	2022-12-19	0.914425
30	2022-12-26	0.000000
31	2023-01-02	1.000000

```

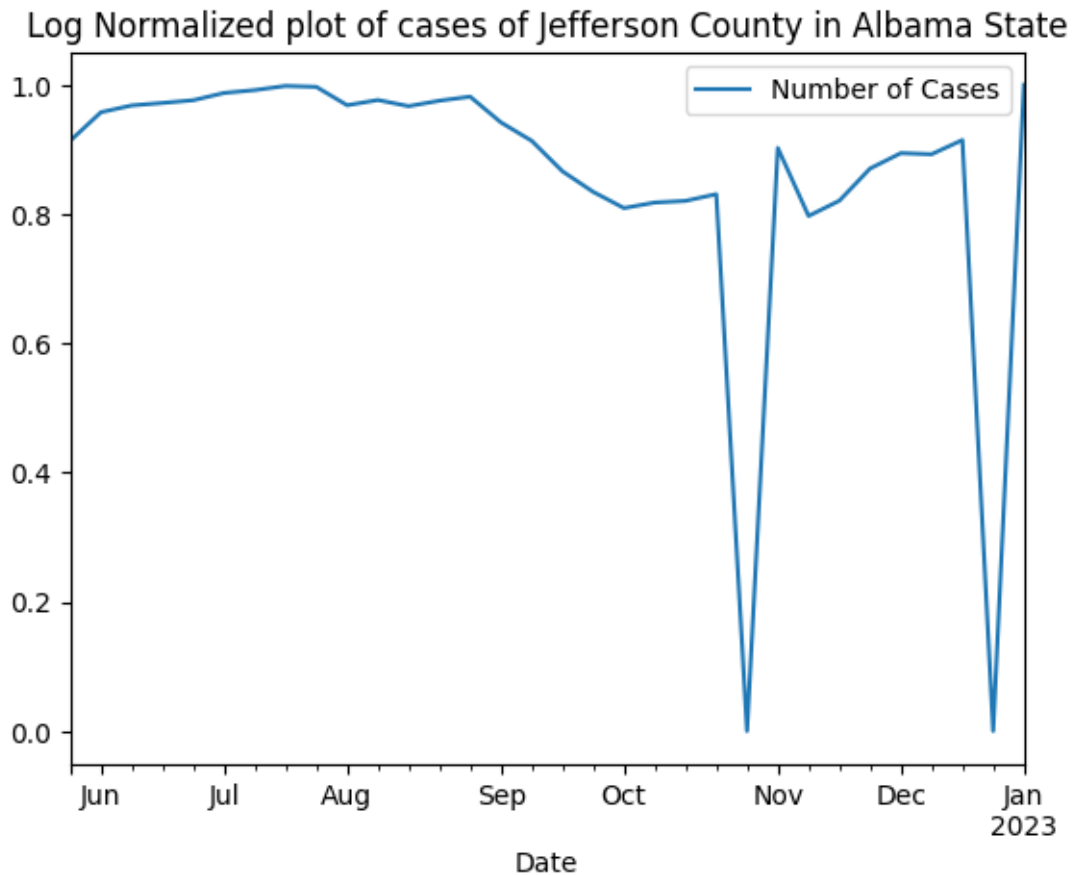
[12]: weekly_cases_selected_county_given_range_max_normalised.plot(x='Date',_
↪y='Number of Cases', title = 'Log Normalized plot of cases of Jefferson_
↪County in Alabama State')

```

```

[12]: <AxesSubplot: title={'center': 'Log Normalized plot of cases of Jefferson County
in Alabama State'}, xlabel='Date'>

```



```
[13]: cases_selected_county_2 = cases_selected_state[cases_selected_state["County_
↪Name"] == selected_county_2].reset_index()
del cases_selected_county_2[cases_selected_county_2.columns[0]]
cases_selected_county_2
```

```
[13]:
```

	countyFIPS	County Name	State	StateFIPS	Date \
0	1089	Madison County	AL	1	2020-01-22
1	1089	Madison County	AL	1	2020-01-23
2	1089	Madison County	AL	1	2020-01-24
3	1089	Madison County	AL	1	2020-01-25
4	1089	Madison County	AL	1	2020-01-26
...
1086	1089	Madison County	AL	1	2023-01-12
1087	1089	Madison County	AL	1	2023-01-13
1088	1089	Madison County	AL	1	2023-01-14
1089	1089	Madison County	AL	1	2023-01-15
1090	1089	Madison County	AL	1	2023-01-16

Number of Cases

0	0
1	0
2	0
3	0
4	0
...	...
1086	113105
1087	113105
1088	113105
1089	113105
1090	113105

[1091 rows x 6 columns]

```
[14]: #For the selected state Alabama summing the deaths per day of all the counties.
cases_selected_county_daily_2 = cases_selected_county_2.groupby('Date')['Number_
↳of Cases'].sum()
cases_selected_county_daily_2
```

```
[14]: Date
2020-01-22      0
2020-01-23      0
2020-01-24      0
2020-01-25      0
2020-01-26      0
...
2023-01-12    113105
2023-01-13    113105
2023-01-14    113105
2023-01-15    113105
2023-01-16    113105
Name: Number of Cases, Length: 1091, dtype: int64
```

```
[15]: #Finding out the new cases per day.
new_cases_selected_county_daily_2 = cases_selected_county_daily_2.diff().
↳reset_index()
new_cases_selected_county_daily_2
```

```
[15]:
```

	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
...
1086	2023-01-12	626.0
1087	2023-01-13	0.0

```

1088 2023-01-14      0.0
1089 2023-01-15      0.0
1090 2023-01-16      0.0

```

[1091 rows x 2 columns]

```

[16]: #Converting the daily to weekly analysis and finding weekly.
weekly_cases_selected_county_2 = new_cases_selected_county_daily_2.copy()
weekly_cases_selected_county_2['Date'] = pd.
    ↳to_datetime(weekly_cases_selected_county_2['Date']) - pd.to_timedelta(7,
    ↳unit='d')
weekly_cases_selected_county_2 = weekly_cases_selected_county_2.groupby([pd.
    ↳Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
weekly_cases_selected_county_2 = weekly_cases_selected_county_2.reset_index()
weekly_cases_selected_county_2.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]: #considering the given range of dates starting from monday. and weekly analysis
    ↳from monday to sunday.
weekly_cases_selected_county_given_range_2 =
    ↳weekly_cases_selected_county_2[(weekly_cases_selected_county_2["Date"] >=
    ↳'2022-05-29') & (weekly_cases_selected_county_2["Date"] <= '2023-01-02')]
weekly_cases_selected_county_given_range_2 =
    ↳weekly_cases_selected_county_given_range_2.sort_values(by=['Date']).
    ↳reset_index(drop=True)
weekly_cases_selected_county_given_range_2['Date'] =
    ↳weekly_cases_selected_county_given_range_2['Date'] + pd.to_timedelta(1,
    ↳unit='d')
weekly_cases_selected_county_given_range_2

```

```

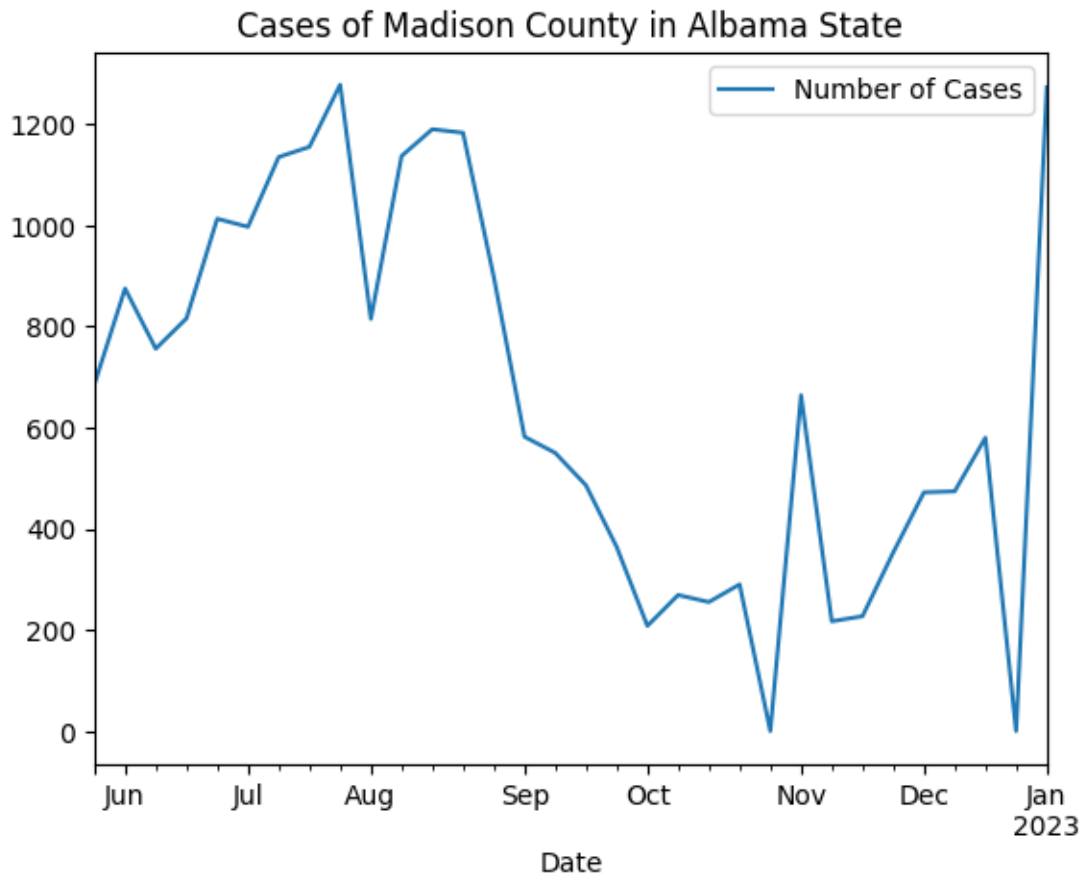
[17]:      Date  Number of Cases
0 2022-05-30      685.0
1 2022-06-06      875.0
2 2022-06-13      756.0
3 2022-06-20      816.0
4 2022-06-27     1013.0
5 2022-07-04      997.0
6 2022-07-11     1135.0
7 2022-07-18     1155.0
8 2022-07-25     1278.0
9 2022-08-01      815.0

```

10	2022-08-08	1137.0
11	2022-08-15	1190.0
12	2022-08-22	1183.0
13	2022-08-29	898.0
14	2022-09-05	582.0
15	2022-09-12	550.0
16	2022-09-19	486.0
17	2022-09-26	364.0
18	2022-10-03	208.0
19	2022-10-10	269.0
20	2022-10-17	255.0
21	2022-10-24	290.0
22	2022-10-31	0.0
23	2022-11-07	664.0
24	2022-11-14	217.0
25	2022-11-21	227.0
26	2022-11-28	353.0
27	2022-12-05	472.0
28	2022-12-12	474.0
29	2022-12-19	580.0
30	2022-12-26	0.0
31	2023-01-02	1273.0

```
[18]: weekly_cases_selected_county_given_range_2.plot(x='Date', y='Number of Cases',
↪title = 'Cases of Madison County in Albama State')
```

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



0.1.2 Week starting with 2022-07-25 has peak of cases in Madison county with value 1278, even though peak is in last week of July, we can see higher trend from start of July. This can be because of independence day holiday. Other spike is in January first week which might be because of new year celebrations.

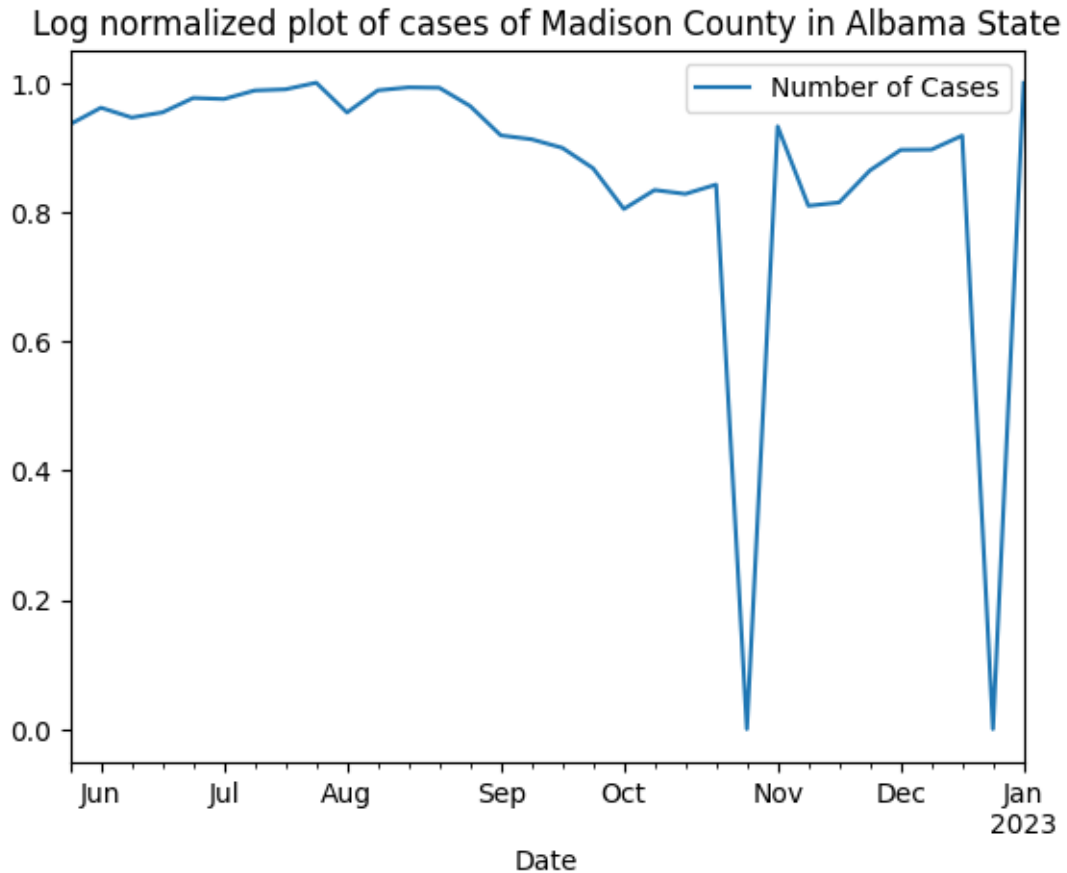
```
[19]: weekly_cases_selected_county_given_range_2_max_normalised = _
→weekly_cases_selected_county_given_range_2.copy()
for column in weekly_cases_selected_county_given_range_2_max_normalised.columns:
    weekly_cases_selected_county_given_range_2_max_normalised['Number of _
→Cases'] = np.
→log(weekly_cases_selected_county_given_range_2_max_normalised['Number of _
→Cases'] + 1 ) / np.
→log(weekly_cases_selected_county_given_range_2_max_normalised['Number of _
→Cases'].max() + 1)
display(weekly_cases_selected_county_given_range_2_max_normalised)
```

	Date	Number of Cases
0	2022-05-30	0.935777
1	2022-06-06	0.961324

2	2022-06-13	0.946122
3	2022-06-20	0.954084
4	2022-06-27	0.976397
5	2022-07-04	0.974766
6	2022-07-11	0.987995
7	2022-07-18	0.989768
8	2022-07-25	1.000000
9	2022-08-01	0.953956
10	2022-08-08	0.988174
11	2022-08-15	0.992794
12	2022-08-22	0.992197
13	2022-08-29	0.964006
14	2022-09-05	0.918522
15	2022-09-12	0.912487
16	2022-09-19	0.899197
17	2022-09-26	0.867674
18	2022-10-03	0.804697
19	2022-10-10	0.833964
20	2022-10-17	0.827928
21	2022-10-24	0.842413
22	2022-10-31	0.000000
23	2022-11-07	0.932495
24	2022-11-14	0.809557
25	2022-11-21	0.814708
26	2022-11-28	0.864288
27	2022-12-05	0.896040
28	2022-12-12	0.896497
29	2022-12-19	0.918156
30	2022-12-26	0.000000
31	2023-01-02	0.999605

```
[20]: weekly_cases_selected_county_given_range_2_max_normalised.plot(x='Date',
    ↪y='Number of Cases', title = 'Log normalized plot of cases of Madison County
    ↪in Alabama State')
```

```
[20]: <AxesSubplot: title={'center': 'Log normalized plot of cases of Madison County
    in Alabama State'}, xlabel='Date'>
```



```
[21]: cases_selected_county_3 = cases_selected_state[cases_selected_state["County_
↪Name"] == selected_county_3].reset_index()
del cases_selected_county_3[cases_selected_county_3.columns[0]]
cases_selected_county_3
```

```
[21]:
```

	countyFIPS	County Name	State	StateFIPS	Date	Number of Cases
0	1097	Mobile County	AL	1	2020-01-22	0
1	1097	Mobile County	AL	1	2020-01-23	0
2	1097	Mobile County	AL	1	2020-01-24	0
3	1097	Mobile County	AL	1	2020-01-25	0
4	1097	Mobile County	AL	1	2020-01-26	0
...
1086	1097	Mobile County	AL	1	2023-01-12	132438
1087	1097	Mobile County	AL	1	2023-01-13	132438
1088	1097	Mobile County	AL	1	2023-01-14	132438
1089	1097	Mobile County	AL	1	2023-01-15	132438
1090	1097	Mobile County	AL	1	2023-01-16	132438

```
[1091 rows x 6 columns]
```

```
[22]: #For the selected state Albama summing the deaths per day of all the counties.
cases_selected_county_daily_3 = cases_selected_county_3.groupby('Date')['Number_
↳of Cases'].sum()
cases_selected_county_daily_3
```

```
[22]: Date
2020-01-22      0
2020-01-23      0
2020-01-24      0
2020-01-25      0
2020-01-26      0
...
2023-01-12    132438
2023-01-13    132438
2023-01-14    132438
2023-01-15    132438
2023-01-16    132438
Name: Number of Cases, Length: 1091, dtype: int64
```

```
[23]: #Finding out the new cases per day.
new_cases_selected_county_daily_3 = cases_selected_county_daily_3.diff().
↳reset_index()
new_cases_selected_county_daily_3
```

```
[23]:      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
...
1086 2023-01-12    590.0
1087 2023-01-13      0.0
1088 2023-01-14      0.0
1089 2023-01-15      0.0
1090 2023-01-16      0.0
```

[1091 rows x 2 columns]

```
[24]: #Converting the daily to weekly analysis and finding weekly.
weekly_cases_selected_county_3 = new_cases_selected_county_daily_3.copy()
weekly_cases_selected_county_3['Date'] = pd.
↳to_datetime(weekly_cases_selected_county_3['Date']) - pd.to_timedelta(7,
↳unit='d')
weekly_cases_selected_county_3 = weekly_cases_selected_county_3.groupby([pd.
↳Grouper(key='Date', freq='W-SUN')])['Number of Cases'].sum()
weekly_cases_selected_county_3 = weekly_cases_selected_county_3.reset_index()
```

```
weekly_cases_selected_county_3.head()
```

```
[24]:
```

	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

```
[25]: #considering the given range of dates starting from monday. and weekly analysis  
      ↳from monday to sunday.  
weekly_cases_selected_county_given_range_3 =  
      ↳weekly_cases_selected_county_3[(weekly_cases_selected_county_3["Date"] >=  
      ↳'2022-05-29') & (weekly_cases_selected_county_3["Date"] <= '2023-01-02')]  
weekly_cases_selected_county_given_range_3 =  
      ↳weekly_cases_selected_county_given_range_3.sort_values(by=['Date']).  
      ↳reset_index(drop=True)  
weekly_cases_selected_county_given_range_3['Date'] =  
      ↳weekly_cases_selected_county_given_range_3['Date'] + pd.to_timedelta(1,  
      ↳unit='d')  
weekly_cases_selected_county_given_range_3
```

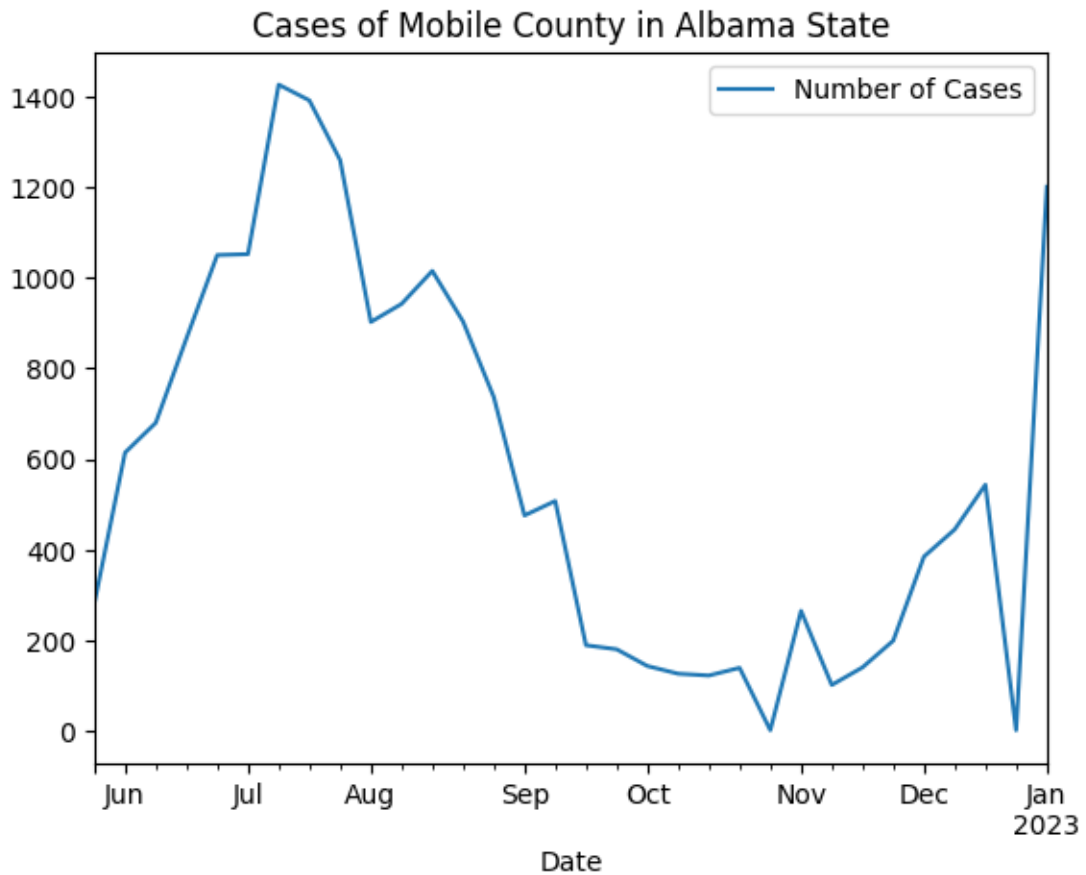
```
[25]:
```

	Date	Number of Cases
0	2022-05-30	280.0
1	2022-06-06	614.0
2	2022-06-13	680.0
3	2022-06-20	867.0
4	2022-06-27	1051.0
5	2022-07-04	1053.0
6	2022-07-11	1428.0
7	2022-07-18	1393.0
8	2022-07-25	1261.0
9	2022-08-01	903.0
10	2022-08-08	943.0
11	2022-08-15	1016.0
12	2022-08-22	904.0
13	2022-08-29	737.0
14	2022-09-05	475.0
15	2022-09-12	507.0
16	2022-09-19	188.0
17	2022-09-26	179.0
18	2022-10-03	142.0
19	2022-10-10	125.0
20	2022-10-17	121.0
21	2022-10-24	138.0
22	2022-10-31	0.0
23	2022-11-07	264.0

24	2022-11-14	100.0
25	2022-11-21	139.0
26	2022-11-28	198.0
27	2022-12-05	384.0
28	2022-12-12	444.0
29	2022-12-19	543.0
30	2022-12-26	0.0
31	2023-01-02	1202.0

```
[26]: weekly_cases_selected_county_given_range_3.plot(x='Date', y='Number of Cases',
→title = 'Cases of Mobile County in Albama State')
```

```
[26]: <AxesSubplot: title={'center': 'Cases of Mobile County in Albama State'},
xlabel='Date'>
```



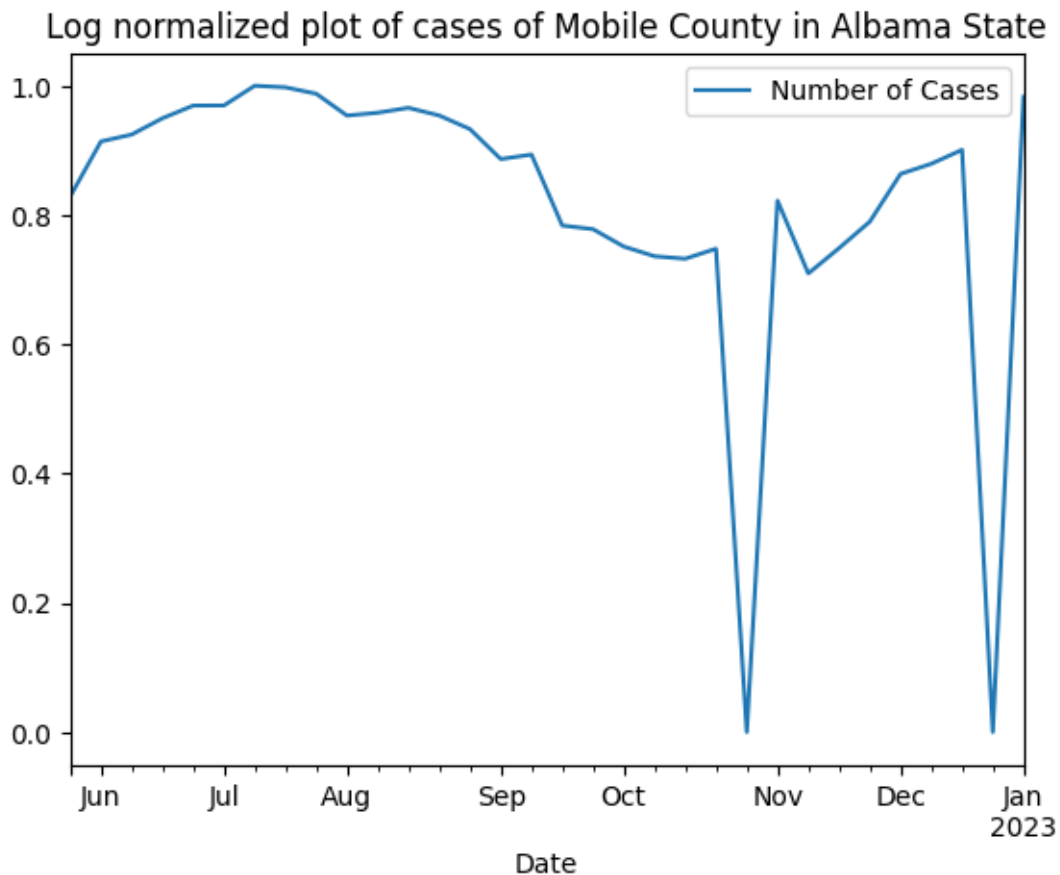
0.1.3 Week starting with 2022-07-11 has peak of cases in mobile county with value 1428, cause can be independence day long weekend holiday, people might have gathered to celebrate.

```
[27]: weekly_cases_selected_county_given_range_3_max_normalised =   
      ↪ weekly_cases_selected_county_given_range_3.copy()   
      for column in weekly_cases_selected_county_given_range_3_max_normalised.columns:   
          weekly_cases_selected_county_given_range_3_max_normalised['Number of   
      ↪ Cases'] = np.   
      ↪ log(w weekly_cases_selected_county_given_range_3_max_normalised['Number of   
      ↪ Cases'] + 1 ) / np.   
      ↪ log(w weekly_cases_selected_county_given_range_3_max_normalised['Number of   
      ↪ Cases'].max() + 1)   
      display(w weekly_cases_selected_county_given_range_3_max_normalised)
```

	Date	Number of Cases
0	2022-05-30	0.828735
1	2022-06-06	0.913757
2	2022-06-13	0.924463
3	2022-06-20	0.949629
4	2022-06-27	0.969263
5	2022-07-04	0.969455
6	2022-07-11	1.000000
7	2022-07-18	0.997536
8	2022-07-25	0.987607
9	2022-08-01	0.953801
10	2022-08-08	0.958233
11	2022-08-15	0.965827
12	2022-08-22	0.953914
13	2022-08-29	0.932849
14	2022-09-05	0.886494
15	2022-09-12	0.893466
16	2022-09-19	0.783694
17	2022-09-26	0.778055
18	2022-10-03	0.751157
19	2022-10-10	0.736147
20	2022-10-17	0.732295
21	2022-10-24	0.747806
22	2022-10-31	0.000000
23	2022-11-07	0.822165
24	2022-11-14	0.709536
25	2022-11-21	0.748653
26	2022-11-28	0.789630
27	2022-12-05	0.863520
28	2022-12-12	0.879242
29	2022-12-19	0.900767
30	2022-12-26	0.000000
31	2023-01-02	0.982804

```
[28]: weekly_cases_selected_county_given_range_3_max_normalised.plot(x='Date',
    ↳y='Number of Cases', title = 'Log normalized plot of cases of Mobile County
    ↳in Alabama State')
```

```
[28]: <AxesSubplot: title={'center': 'Log normalized plot of cases of Mobile County in
Alabama State'}, xlabel='Date'>
```



```
[29]: #For the selected state Alabama summing the deaths per day of all the counties.
cases_selected_state_daily = cases_selected_state.groupby('Date')['Number of
    ↳Cases'].sum()
cases_selected_state_daily.head()
```

```
[29]: 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
```

```
[30]: #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()
```

```
[30]:
```

	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

```
[31]: #Converting the daily to weekly analysis and finding the mean weekly.
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()
```

```
[31]:
```

	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

```
[32]: #considering the given range of dates starting from monday. and weekly analysis
    ↳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
```

```
[32]:
```

	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

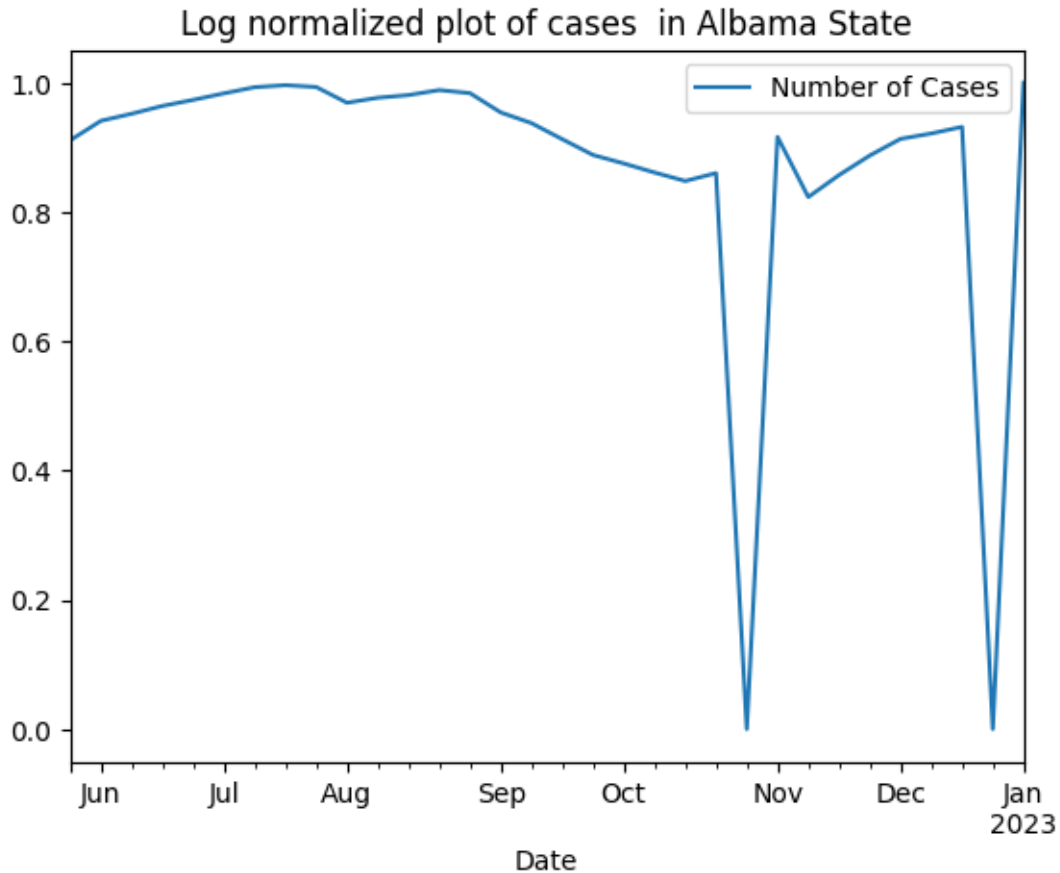
```
[33]: #getting the normalised values for state.
weekly_cases_selected_state_given_range_max_normalised =
    ↪weekly_cases_sum_selected_state_given_range.copy()
for column in weekly_cases_selected_state_given_range_max_normalised.columns:
    weekly_cases_selected_state_given_range_max_normalised['Number of Cases'] =
    ↪np.log(weekly_cases_selected_state_given_range_max_normalised['Number of
    ↪Cases'] + 1 ) / np.
    ↪log(weekly_cases_selected_state_given_range_max_normalised['Number of
    ↪Cases'].max() + 1)
display(weekly_cases_selected_state_given_range_max_normalised)
```

	Date	Number of Cases
0	2022-05-30	0.910949
1	2022-06-06	0.941026
2	2022-06-13	0.951989
3	2022-06-20	0.964080
4	2022-06-27	0.973428
5	2022-07-04	0.983511
6	2022-07-11	0.993074

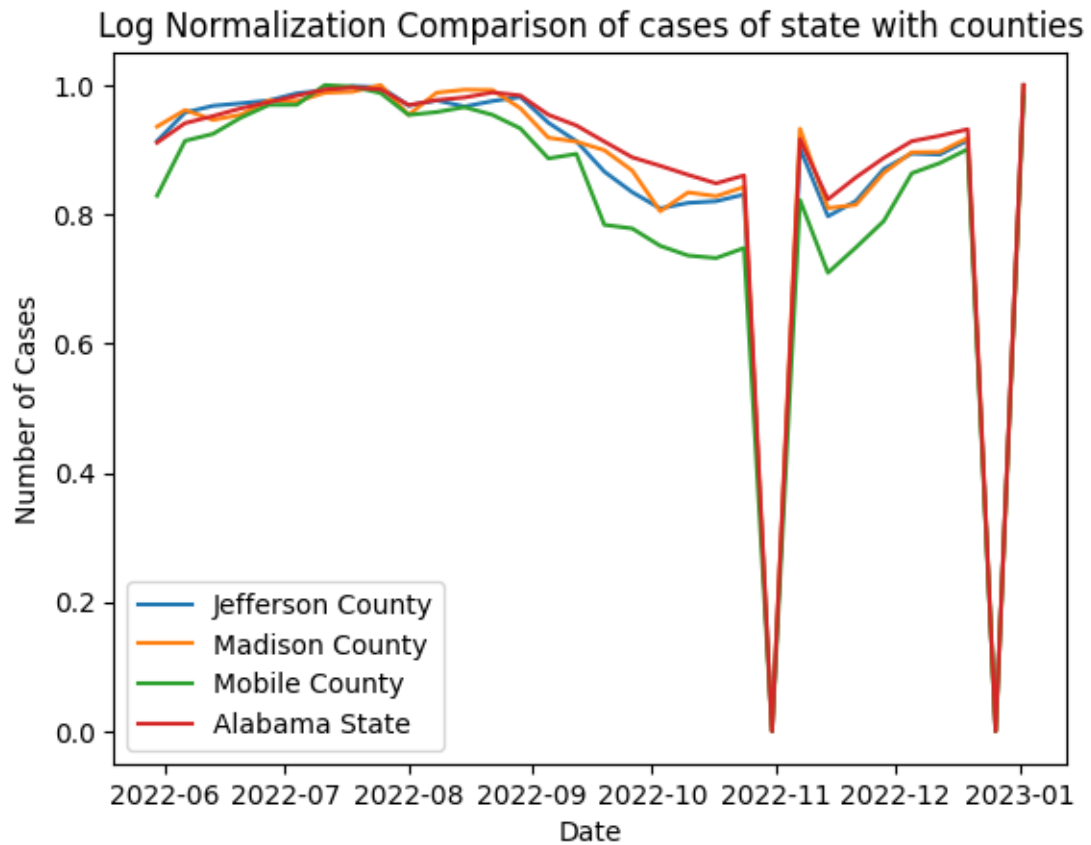
7	2022-07-18	0.996185
8	2022-07-25	0.993353
9	2022-08-01	0.968982
10	2022-08-08	0.976695
11	2022-08-15	0.980861
12	2022-08-22	0.988601
13	2022-08-29	0.983825
14	2022-09-05	0.954013
15	2022-09-12	0.937467
16	2022-09-19	0.912619
17	2022-09-26	0.888192
18	2022-10-03	0.875065
19	2022-10-10	0.861034
20	2022-10-17	0.847928
21	2022-10-24	0.860143
22	2022-10-31	0.000000
23	2022-11-07	0.916211
24	2022-11-14	0.823099
25	2022-11-21	0.857097
26	2022-11-28	0.887343
27	2022-12-05	0.913118
28	2022-12-12	0.921471
29	2022-12-19	0.931676
30	2022-12-26	0.000000
31	2023-01-02	1.000000

```
[34]: weekly_cases_selected_state_given_range_max_normalised.plot(x='Date', y='Number_
      ↪of Cases', title = 'Log normalized plot of cases in Albama State')
```

```
[34]: <AxesSubplot: title={'center': 'Log normalized plot of cases in Albama State'},
      xlabel='Date'>
```



```
[35]: plt.plot(weekly_cases_selected_county_given_range_max_normalised['Date'],  
            ↪weekly_cases_selected_county_given_range_max_normalised['Number of Cases'],  
            ↪label='Jefferson County')  
plt.plot(weekly_cases_selected_county_given_range_2_max_normalised['Date'],  
            ↪weekly_cases_selected_county_given_range_2_max_normalised['Number of  
            ↪Cases'], label='Madison County')  
plt.plot(weekly_cases_selected_county_given_range_3_max_normalised['Date'],  
            ↪weekly_cases_selected_county_given_range_3_max_normalised['Number of  
            ↪Cases'], label='Mobile County')  
plt.plot(weekly_cases_selected_state_given_range_max_normalised['Date'],  
            ↪weekly_cases_selected_state_given_range_max_normalised['Number of Cases'],  
            ↪label='Alabama State')  
plt.title('Log Normalization Comparison of cases of state with counties')  
plt.xlabel('Date')  
plt.ylabel('Number of Cases')  
plt.legend()  
plt.show()
```



0.1.4 Yes all three Counties(Jefferson County, Madison County, Mobile County) are following the State pattern(Alabama State).