Stage2_Task4_Deaths

March 14, 2023

0.1 Plot weekly trends (new deaths) 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 Albama state for analysis
selected_state = "AL"
selected_county_1 = "Jefferson County "
selected_county_2 = "Mobile County "
selected_county_3 = "Madison County "
#Readind the deaths data
deaths = pd.read_csv("../data/covid_deaths_usafacts.csv")
deaths.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	…	230	
2	0	0	0	0	…	719	
3	0	0	0	0	…	103	
4	0	0	0	0	…	108	

	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	230	230	230	230	230	230	
2	719	719	719	719	721	721	
3	103	103	103	103	103	103	
4	108	108	108	108	108	108	

```
   2023-01-14  2023-01-15  2023-01-16
0           0           0           0
1         230         230         230
2         721         721         721
3         103         103         103
4         108         108         108

[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.
deaths_transpose = pd.melt(frame= deaths, id_vars=('countyFIPS','County␣
 ↪Name','State','StateFIPS'), var_name=["Date"], value_name='Number of Deaths')
deaths_transpose = deaths_transpose[deaths_transpose['countyFIPS'] != 0]
deaths_transpose.head()
```

[3]:

	countyFIPS	County Name	State	StateFIPS	Date	Number of Deaths
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]:
```
deaths_selected_state = deaths_transpose[deaths_transpose["State"] ==␣
 ↪selected_state]
deaths_selected_state.head()
```

[4]:

	countyFIPS	County Name	State	StateFIPS	Date	Number of Deaths
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]:
```
deaths_selected_county = deaths_selected_state[deaths_selected_state["County␣
 ↪Name"] == selected_county_1].reset_index()
del deaths_selected_county[deaths_selected_county.columns[0]]
deaths_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 Deaths
0                    0
1                    0
2                    0
3                    0
4                    0
…                  …
1086              2495
1087              2495
1088              2495
1089              2495
1090              2495

[1091 rows x 6 columns]
```

[6]:
```
#For the selected state Albama summing the deaths per day of all the counties.
deaths_selected_county_daily = deaths_selected_county.groupby('Date')['Number␣
 ↪of Deaths'].sum()
deaths_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    2495
2023-01-13    2495
2023-01-14    2495
2023-01-15    2495
2023-01-16    2495
Name: Number of Deaths, Length: 1091, dtype: int64
```

[7]:
```
#Finding out the new cases per day.
new_deaths_selected_county_daily = deaths_selected_county_daily.diff().
 ↪reset_index()
new_deaths_selected_county_daily
```

[7]:
```
            Date  Number of Deaths
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                4.0
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_deaths_selected_county = new_deaths_selected_county_daily
weekly_deaths_selected_county['Date'] = pd.
 ↪to_datetime(weekly_deaths_selected_county['Date']) - pd.to_timedelta(7,␣
 ↪unit='d')
weekly_deaths_selected_county = weekly_deaths_selected_county.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Deaths'].sum()
weekly_deaths_selected_county = weekly_deaths_selected_county.reset_index()
weekly_deaths_selected_county.head()
```

[8]:
```
        Date  Number of Deaths
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
```

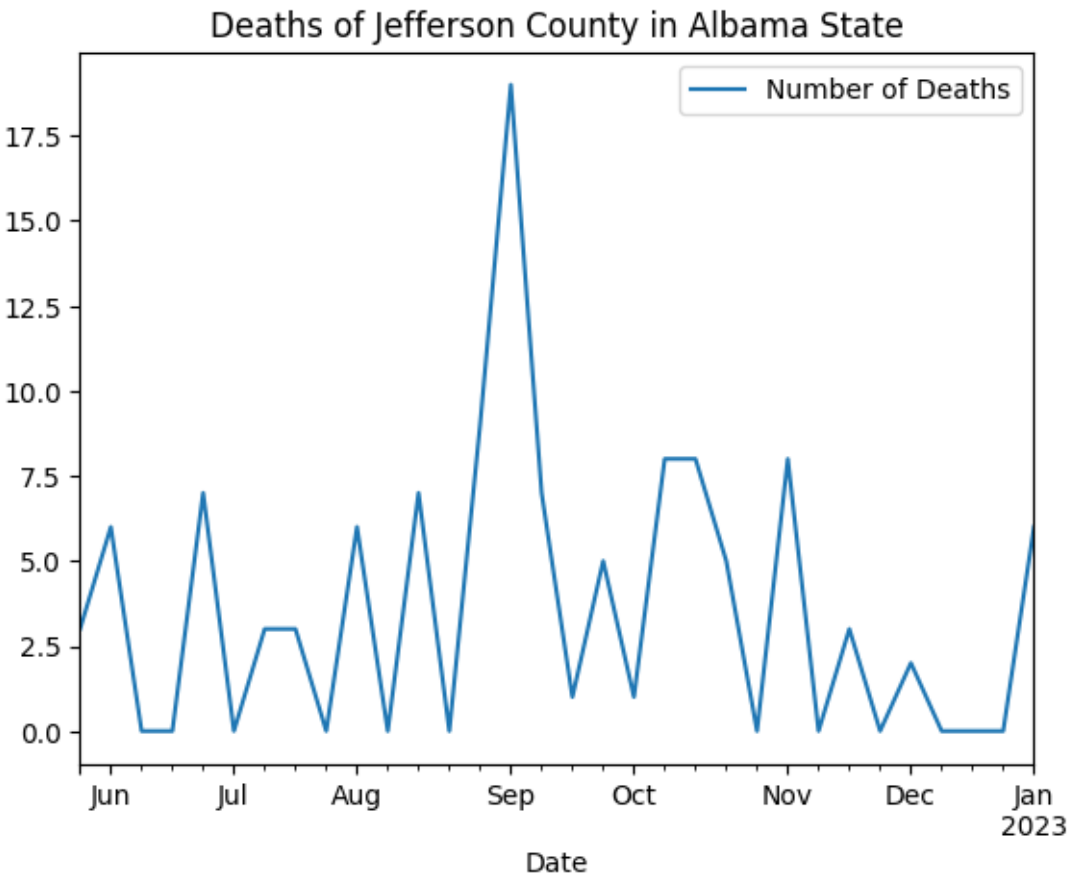
[9]:
```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_deaths_selected_county_given_range =␣
 ↪weekly_deaths_selected_county[(weekly_deaths_selected_county["Date"] >=␣
 ↪'2022-05-29') & (weekly_deaths_selected_county["Date"] <= '2023-01-02')]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

[9]:
```
          Date  Number of Deaths
0   2022-05-30                3.0
1   2022-06-06                6.0
2   2022-06-13                0.0
```

```
3  2022-06-20                0.0
4  2022-06-27                7.0
5  2022-07-04                0.0
6  2022-07-11                3.0
7  2022-07-18                3.0
8  2022-07-25                0.0
9  2022-08-01                6.0
10 2022-08-08                0.0
11 2022-08-15                7.0
12 2022-08-22                0.0
13 2022-08-29                9.0
14 2022-09-05               19.0
15 2022-09-12                7.0
16 2022-09-19                1.0
17 2022-09-26                5.0
18 2022-10-03                1.0
19 2022-10-10                8.0
20 2022-10-17                8.0
21 2022-10-24                5.0
22 2022-10-31                0.0
23 2022-11-07                8.0
24 2022-11-14                0.0
25 2022-11-21                3.0
26 2022-11-28                0.0
27 2022-12-05                2.0
28 2022-12-12                0.0
29 2022-12-19                0.0
30 2022-12-26                0.0
31 2023-01-02                6.0
```

[10]:
```
weekly_deaths_selected_county_given_range.plot(x='Date', y='Number of Deaths',␣
↪title = 'Deaths of Jefferson County in Albama State')
```

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



0.1.1 Week ending with 2022-09-05 has peak of deaths in jefferson county with value 19. As september 5th 2022 was a labour day and maybe people have gathered as it was long weeekend.

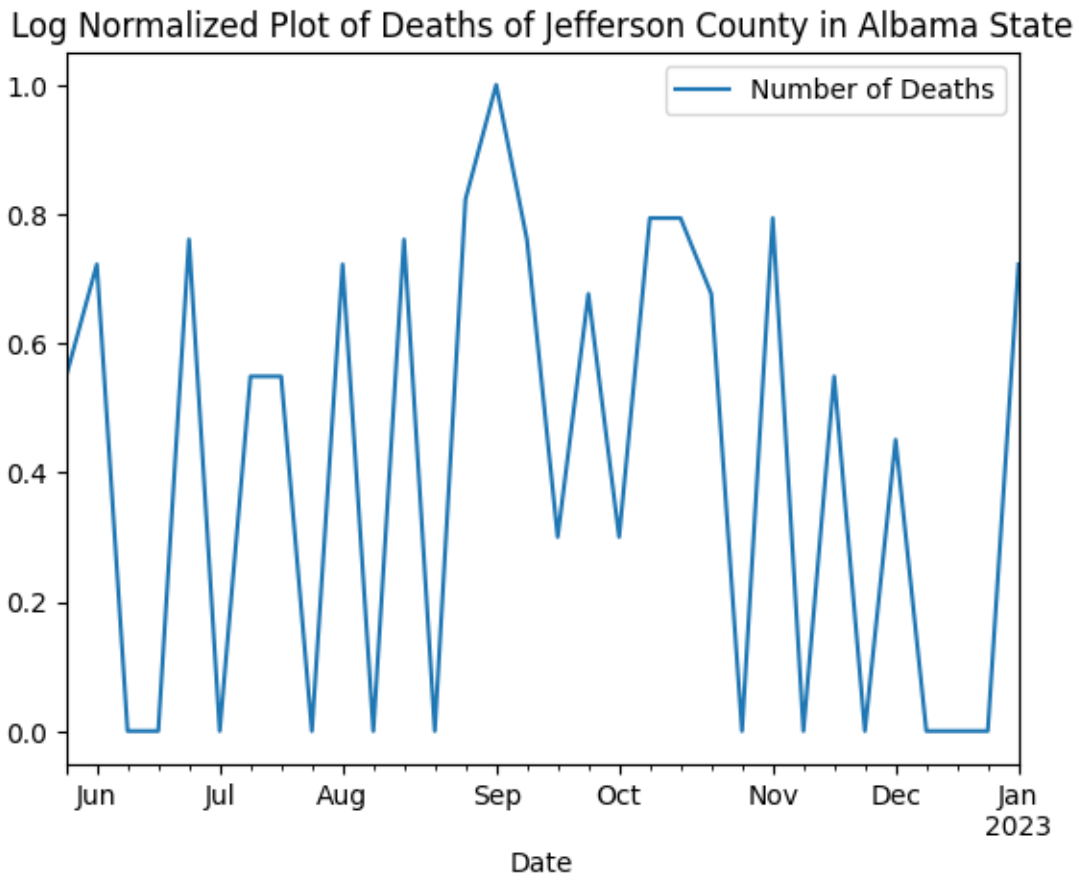
[11]:
```
weekly_deaths_selected_county_given_range_max_normalised =␣
 ↪weekly_deaths_selected_county_given_range.copy()
for column in weekly_deaths_selected_county_given_range_max_normalised.columns:
    weekly_deaths_selected_county_given_range_max_normalised['Number of␣
 ↪Deaths'] = np.
 ↪log(weekly_deaths_selected_county_given_range_max_normalised['Number of␣
 ↪Deaths'] + 1 ) / np.
 ↪log(weekly_deaths_selected_county_given_range_max_normalised['Number of␣
 ↪Deaths'].max() + 1)
display(weekly_deaths_selected_county_given_range_max_normalised)
```

	Date	Number of Deaths
0	2022-05-30	0.548690
1	2022-06-06	0.722082
2	2022-06-13	0.000000

```
3  2022-06-20          0.000000
4  2022-06-27          0.760549
5  2022-07-04          0.000000
6  2022-07-11          0.548690
7  2022-07-18          0.548690
8  2022-07-25          0.000000
9  2022-08-01          0.722082
10 2022-08-08          0.000000
11 2022-08-15          0.760549
12 2022-08-22          0.000000
13 2022-08-29          0.822626
14 2022-09-05          1.000000
15 2022-09-12          0.760549
16 2022-09-19          0.300274
17 2022-09-26          0.676361
18 2022-10-03          0.300274
19 2022-10-10          0.793648
20 2022-10-17          0.793648
21 2022-10-24          0.676361
22 2022-10-31          0.000000
23 2022-11-07          0.793648
24 2022-11-14          0.000000
25 2022-11-21          0.548690
26 2022-11-28          0.000000
27 2022-12-05          0.450724
28 2022-12-12          0.000000
29 2022-12-19          0.000000
30 2022-12-26          0.000000
31 2023-01-02          0.722082
```

[12]:
```
weekly_deaths_selected_county_given_range_max_normalised.plot(x='Date',␣
↪y='Number of Deaths', title = 'Log Normalized Plot of Deaths of Jefferson␣
↪County in Albama State')
```

[12]:
```
<AxesSubplot: title={'center': 'Log Normalized Plot of Deaths of Jefferson
County in Albama State'}, xlabel='Date'>
```



[13]:
```
deaths_selected_county_2 = deaths_selected_state[deaths_selected_state["County␣
↪Name"] == selected_county_2].reset_index()
del deaths_selected_county_2[deaths_selected_county_2.columns[0]]
deaths_selected_county_2
```

[13]:

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

```
0                    0
1                    0
2                    0
3                    0
4                    0
…                  …
1086              1765
1087              1765
1088              1765
1089              1765
1090              1765

[1091 rows x 6 columns]
```

[14]:
```
#For the selected state Albama summing the deaths per day of all the counties.
deaths_selected_county_daily_2 = deaths_selected_county_2.
 ↪groupby('Date')['Number of Deaths'].sum()
deaths_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    1765
2023-01-13    1765
2023-01-14    1765
2023-01-15    1765
2023-01-16    1765
Name: Number of Deaths, Length: 1091, dtype: int64
```

[15]:
```
#Finding out the new cases per day.
new_deaths_selected_county_daily_2 = deaths_selected_county_daily_2.diff().
 ↪reset_index()
new_deaths_selected_county_daily_2
```

[15]:
```
            Date  Number of Deaths
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               2.0
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_deaths_selected_county_2 = new_deaths_selected_county_daily_2
weekly_deaths_selected_county_2['Date'] = pd.
 ↪to_datetime(weekly_deaths_selected_county_2['Date']) - pd.to_timedelta(7,␣
 ↪unit='d')
weekly_deaths_selected_county_2 = weekly_deaths_selected_county_2.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Deaths'].sum()
weekly_deaths_selected_county_2 = weekly_deaths_selected_county_2.reset_index()
weekly_deaths_selected_county_2.head()
```

[16]:
```
        Date  Number of Deaths
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
```

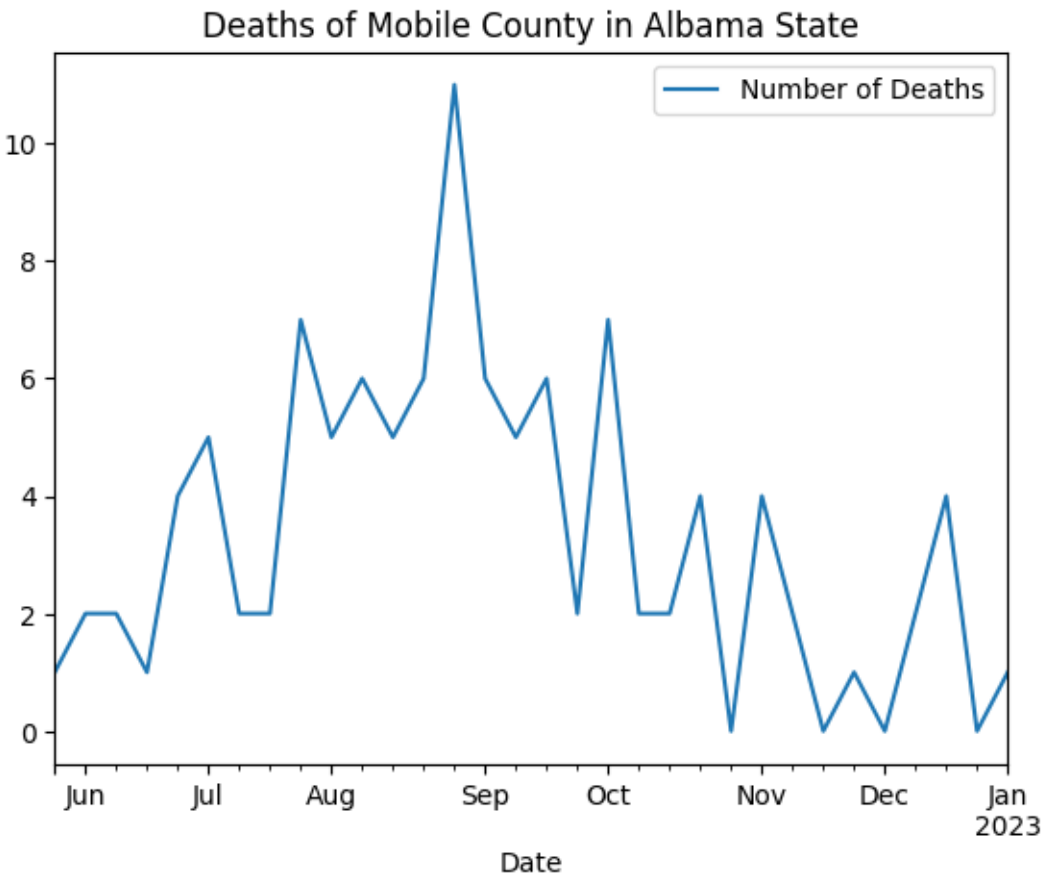
[17]:
```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_deaths_selected_county_given_range_2 =␣
 ↪weekly_deaths_selected_county_2[(weekly_deaths_selected_county_2["Date"] >=␣
 ↪'2022-05-29') & (weekly_deaths_selected_county_2["Date"] <= '2023-01-02')]
weekly_deaths_selected_county_given_range_2 =␣
 ↪weekly_deaths_selected_county_given_range_2.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_deaths_selected_county_given_range_2['Date'] =␣
 ↪weekly_deaths_selected_county_given_range_2['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_deaths_selected_county_given_range_2
```

[17]:
```
         Date  Number of Deaths
0  2022-05-30               1.0
1  2022-06-06               2.0
2  2022-06-13               2.0
3  2022-06-20               1.0
4  2022-06-27               4.0
5  2022-07-04               5.0
6  2022-07-11               2.0
7  2022-07-18               2.0
8  2022-07-25               7.0
9  2022-08-01               5.0
```

```
10 2022-08-08                6.0
11 2022-08-15                5.0
12 2022-08-22                6.0
13 2022-08-29               11.0
14 2022-09-05                6.0
15 2022-09-12                5.0
16 2022-09-19                6.0
17 2022-09-26                2.0
18 2022-10-03                7.0
19 2022-10-10                2.0
20 2022-10-17                2.0
21 2022-10-24                4.0
22 2022-10-31                0.0
23 2022-11-07                4.0
24 2022-11-14                2.0
25 2022-11-21                0.0
26 2022-11-28                1.0
27 2022-12-05                0.0
28 2022-12-12                2.0
29 2022-12-19                4.0
30 2022-12-26                0.0
31 2023-01-02                1.0
```

[18]:
```
weekly_deaths_selected_county_given_range_2.plot(x='Date', y='Number of␣
↪Deaths', title = 'Deaths of Mobile County in Albama State')
```

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



0.1.2 Week ending with 2022-08-29 has peak of deaths in mobile county with value 11. As september 5th 2022 was a labour day and maybe people have gathered as it was long weeekend.

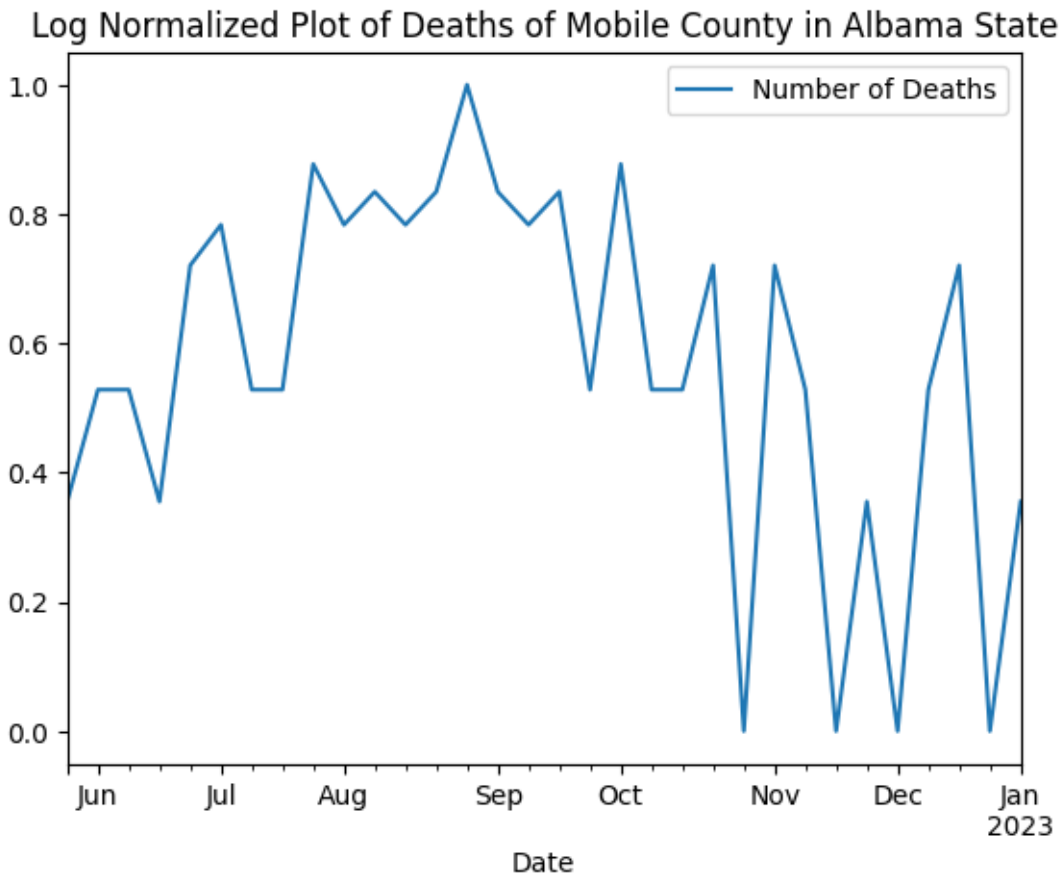
[19]:
```
weekly_deaths_selected_county_given_range_2_max_normalised =␣
 ↪weekly_deaths_selected_county_given_range_2.copy()
for column in weekly_deaths_selected_county_given_range_2_max_normalised.
 ↪columns:
    weekly_deaths_selected_county_given_range_2_max_normalised['Number of␣
 ↪Deaths'] = np.
 ↪log(weekly_deaths_selected_county_given_range_2_max_normalised['Number of␣
 ↪Deaths'] + 1 ) / np.
 ↪log(weekly_deaths_selected_county_given_range_2_max_normalised['Number of␣
 ↪Deaths'].max() + 1)
display(weekly_deaths_selected_county_given_range_2_max_normalised)
```

```
         Date  Number of Deaths
0  2022-05-30          0.354952
1  2022-06-06          0.528185
```

```
2  2022-06-13          0.528185
3  2022-06-20          0.354952
4  2022-06-27          0.720441
5  2022-07-04          0.783295
6  2022-07-11          0.528185
7  2022-07-18          0.528185
8  2022-07-25          0.877217
9  2022-08-01          0.783295
10 2022-08-08          0.834381
11 2022-08-15          0.783295
12 2022-08-22          0.834381
13 2022-08-29          1.000000
14 2022-09-05          0.834381
15 2022-09-12          0.783295
16 2022-09-19          0.834381
17 2022-09-26          0.528185
18 2022-10-03          0.877217
19 2022-10-10          0.528185
20 2022-10-17          0.528185
21 2022-10-24          0.720441
22 2022-10-31          0.000000
23 2022-11-07          0.720441
24 2022-11-14          0.528185
25 2022-11-21          0.000000
26 2022-11-28          0.354952
27 2022-12-05          0.000000
28 2022-12-12          0.528185
29 2022-12-19          0.720441
30 2022-12-26          0.000000
31 2023-01-02          0.354952
```

[20]:
```
weekly_deaths_selected_county_given_range_2_max_normalised.plot(x='Date',␣
↪y='Number of Deaths', title = 'Log Normalized Plot of Deaths of Mobile␣
↪County in Albama State')
```

[20]:
```
<AxesSubplot: title={'center': 'Log Normalized Plot of Deaths of Mobile County
in Albama State'}, xlabel='Date'>
```



[21]:
```
deaths_selected_county_3 = deaths_selected_state[deaths_selected_state["County␣
↪Name"] == selected_county_3].reset_index()
del deaths_selected_county_3[deaths_selected_county_3.columns[0]]
deaths_selected_county_3
```

[21]:

	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 Deaths

```
0                    0
1                    0
2                    0
3                    0
4                    0
…                  …
1086              1052
1087              1052
1088              1052
1089              1052
1090              1052

[1091 rows x 6 columns]
```

[22]:
```
#For the selected state Albama summing the deaths per day of all the counties.
deaths_selected_county_daily_3 = deaths_selected_county_3.
 ↪groupby('Date')['Number of Deaths'].sum()
deaths_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    1052
2023-01-13    1052
2023-01-14    1052
2023-01-15    1052
2023-01-16    1052
Name: Number of Deaths, Length: 1091, dtype: int64
```

[23]:
```
#Finding out the new cases per day.
new_deaths_selected_county_daily_3 = deaths_selected_county_daily_3.diff().
 ↪reset_index()
new_deaths_selected_county_daily_3
```

[23]:
```
            Date  Number of Deaths
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               4.0
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_deaths_selected_county_3 = new_deaths_selected_county_daily_3
weekly_deaths_selected_county_3['Date'] = pd.
 ↪to_datetime(weekly_deaths_selected_county_3['Date']) - pd.to_timedelta(7,␣
 ↪unit='d')
weekly_deaths_selected_county_3 = weekly_deaths_selected_county_3.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Deaths'].sum()
weekly_deaths_selected_county_3 = weekly_deaths_selected_county_3.reset_index()
weekly_deaths_selected_county_3.head()
```

[24]:
```
        Date  Number of Deaths
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
```

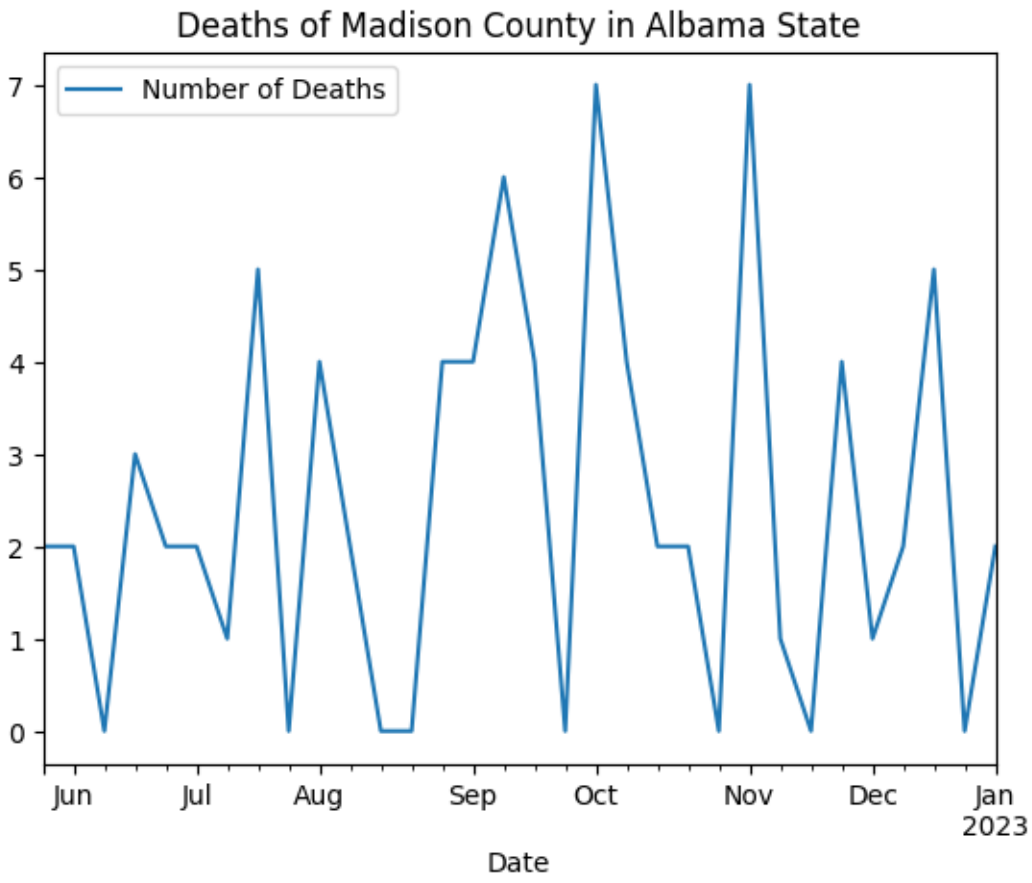
[25]:
```
#considering the given range of dates starting from monday. and weekly analsis␣
 ↪from monday to sunday.
weekly_deaths_selected_county_given_range_3 =␣
 ↪weekly_deaths_selected_county_3[(weekly_deaths_selected_county_3["Date"] >=␣
 ↪'2022-05-29') & (weekly_deaths_selected_county_3["Date"] <= '2023-01-02')]
weekly_deaths_selected_county_given_range_3 =␣
 ↪weekly_deaths_selected_county_given_range_3.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_deaths_selected_county_given_range_3['Date'] =␣
 ↪weekly_deaths_selected_county_given_range_3['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_deaths_selected_county_given_range_3
```

[25]:
```
         Date  Number of Deaths
0  2022-05-30               2.0
1  2022-06-06               2.0
2  2022-06-13               0.0
3  2022-06-20               3.0
4  2022-06-27               2.0
5  2022-07-04               2.0
6  2022-07-11               1.0
7  2022-07-18               5.0
8  2022-07-25               0.0
9  2022-08-01               4.0
```

```
10 2022-08-08              2.0
11 2022-08-15              0.0
12 2022-08-22              0.0
13 2022-08-29              4.0
14 2022-09-05              4.0
15 2022-09-12              6.0
16 2022-09-19              4.0
17 2022-09-26              0.0
18 2022-10-03              7.0
19 2022-10-10              4.0
20 2022-10-17              2.0
21 2022-10-24              2.0
22 2022-10-31              0.0
23 2022-11-07              7.0
24 2022-11-14              1.0
25 2022-11-21              0.0
26 2022-11-28              4.0
27 2022-12-05              1.0
28 2022-12-12              2.0
29 2022-12-19              5.0
30 2022-12-26              0.0
31 2023-01-02              2.0
```

[26]:
```
weekly_deaths_selected_county_given_range_3.plot(x='Date', y='Number of␣
↪Deaths', title = 'Deaths of Madison County in Albama State')
```

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



0.1.3 Week ending with 2022-10-03 and 2022-11-07 has peak value of deaths in madison county with value 7.as september 5th was state holidy may be cases increased there which reukted in deaths after two weeks.

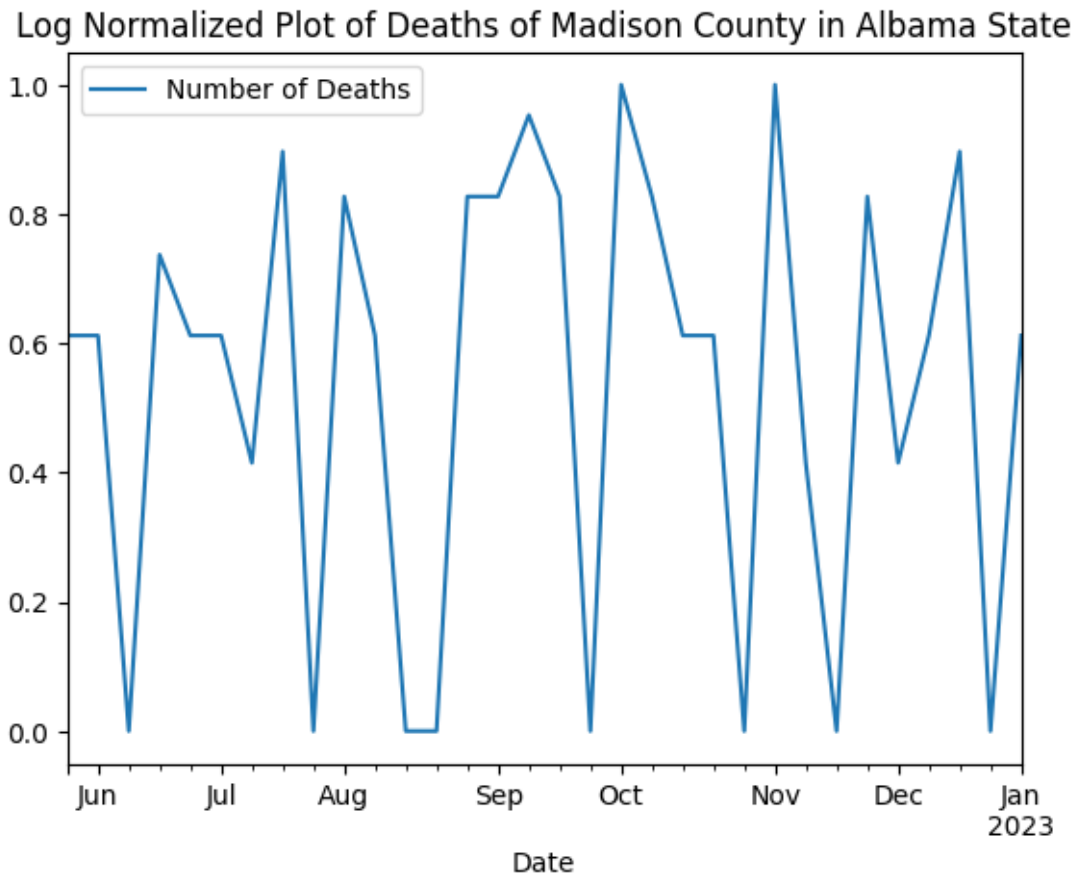
[27]:
```
weekly_deaths_selected_county_given_range_3_max_normalised =␣
 ↪weekly_deaths_selected_county_given_range_3.copy()
for column in weekly_deaths_selected_county_given_range_3_max_normalised.
 ↪columns:
    weekly_deaths_selected_county_given_range_3_max_normalised['Number of␣
 ↪Deaths'] = np.
 ↪log(weekly_deaths_selected_county_given_range_3_max_normalised['Number of␣
 ↪Deaths'] + 1 ) / np.
 ↪log(weekly_deaths_selected_county_given_range_3_max_normalised['Number of␣
 ↪Deaths'].max() + 1)
display(weekly_deaths_selected_county_given_range_3_max_normalised)
```

```
         Date  Number of Deaths
0  2022-05-30          0.611947
1  2022-06-06          0.611947
```

```
2  2022-06-13          0.000000
3  2022-06-20          0.736966
4  2022-06-27          0.611947
5  2022-07-04          0.611947
6  2022-07-11          0.415037
7  2022-07-18          0.896585
8  2022-07-25          0.000000
9  2022-08-01          0.826987
10 2022-08-08          0.611947
11 2022-08-15          0.000000
12 2022-08-22          0.000000
13 2022-08-29          0.826987
14 2022-09-05          0.826987
15 2022-09-12          0.952919
16 2022-09-19          0.826987
17 2022-09-26          0.000000
18 2022-10-03          1.000000
19 2022-10-10          0.826987
20 2022-10-17          0.611947
21 2022-10-24          0.611947
22 2022-10-31          0.000000
23 2022-11-07          1.000000
24 2022-11-14          0.415037
25 2022-11-21          0.000000
26 2022-11-28          0.826987
27 2022-12-05          0.415037
28 2022-12-12          0.611947
29 2022-12-19          0.896585
30 2022-12-26          0.000000
31 2023-01-02          0.611947
```

[28]:
```
weekly_deaths_selected_county_given_range_3_max_normalised.plot(x='Date',␣
↪y='Number of Deaths', title = 'Log Normalized Plot of Deaths of Madison␣
↪County in Albama State')
```

[28]:
```
<AxesSubplot: title={'center': 'Log Normalized Plot of Deaths of Madison County
in Albama State'}, xlabel='Date'>
```



[29]:
```
#For the selected state Albama summing the deaths per day of all the counties.
deaths_selected_state_daily = deaths_selected_state.groupby('Date')['Number of␣
↪Deaths'].sum()
deaths_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 Deaths, dtype: int64
```

[30]:
```
#Finding out the new cases per day.
new_deaths_selected_state_daily = deaths_selected_state_daily.diff().
↪reset_index()
new_deaths_selected_state_daily.head()
```

[30]:

	Date	Number of Deaths
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_deaths_sum_selected_state = new_deaths_selected_state_daily.copy()
weekly_deaths_sum_selected_state['Date'] = pd.
 ↪to_datetime(weekly_deaths_sum_selected_state['Date']) - pd.to_timedelta(7,␣
 ↪unit='d')
weekly_deaths_sum_selected_state = weekly_deaths_sum_selected_state.groupby([pd.
 ↪Grouper(key='Date', freq='W-SUN')])['Number of Deaths'].sum()
weekly_deaths_sum_selected_state = weekly_deaths_sum_selected_state.
 ↪reset_index()
weekly_deaths_sum_selected_state.head()
```

[31]:

	Date	Number of Deaths
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 analsis␣
 ↪from monday to sunday.
weekly_deaths_sum_selected_state_given_range =␣
 ↪weekly_deaths_sum_selected_state[(weekly_deaths_sum_selected_state["Date"]␣
 ↪>= '2022-05-29') & (weekly_deaths_sum_selected_state["Date"] <=␣
 ↪'2023-01-02')]
weekly_deaths_sum_selected_state_given_range =␣
 ↪weekly_deaths_sum_selected_state_given_range.sort_values(by=['Date']).
 ↪reset_index(drop=True)
weekly_deaths_sum_selected_state_given_range['Date'] =␣
 ↪weekly_deaths_sum_selected_state_given_range['Date'] + pd.to_timedelta(1,␣
 ↪unit='d')
weekly_deaths_sum_selected_state_given_range
```

[32]:

	Date	Number of Deaths
0	2022-05-30	13.0
1	2022-06-06	28.0
2	2022-06-13	4.0
3	2022-06-20	33.0
4	2022-06-27	30.0
5	2022-07-04	27.0
6	2022-07-11	36.0

```
7  2022-07-18              50.0
8  2022-07-25              19.0
9  2022-08-01              83.0
10 2022-08-08              52.0
11 2022-08-15              15.0
12 2022-08-22               7.0
13 2022-08-29             112.0
14 2022-09-05              79.0
15 2022-09-12              83.0
16 2022-09-19              73.0
17 2022-09-26              27.0
18 2022-10-03              51.0
19 2022-10-10              32.0
20 2022-10-17              28.0
21 2022-10-24              25.0
22 2022-10-31               0.0
23 2022-11-07              50.0
24 2022-11-14               9.0
25 2022-11-21              14.0
26 2022-11-28              21.0
27 2022-12-05              30.0
28 2022-12-12              22.0
29 2022-12-19              33.0
30 2022-12-26               0.0
31 2023-01-02              39.0
```

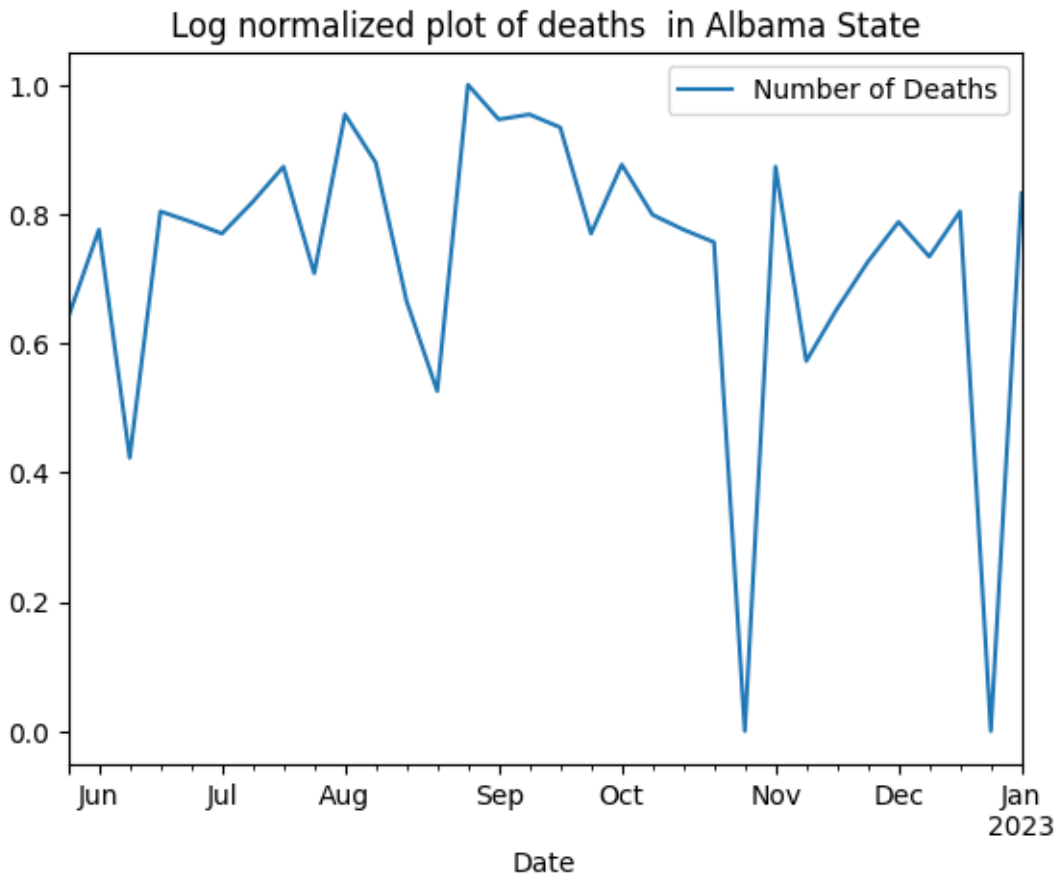
[33]:
```
weekly_deaths_selected_state_given_range_max_normalised =␣
 ↪weekly_deaths_sum_selected_state_given_range.copy()
for column in weekly_deaths_selected_state_given_range_max_normalised.columns:
    weekly_deaths_selected_state_given_range_max_normalised['Number of Deaths']␣
 ↪= np.log(weekly_deaths_selected_state_given_range_max_normalised['Number of␣
 ↪Deaths'] + 1 ) / np.
 ↪log(weekly_deaths_selected_state_given_range_max_normalised['Number of␣
 ↪Deaths'].max() + 1)
display(weekly_deaths_selected_state_given_range_max_normalised)
```

```
         Date  Number of Deaths
0  2022-05-30          0.639925
1  2022-06-06          0.775931
2  2022-06-13          0.422717
3  2022-06-20          0.804006
4  2022-06-27          0.787769
5  2022-07-04          0.769664
6  2022-07-11          0.818711
7  2022-07-18          0.873193
8  2022-07-25          0.708141
9  2022-08-01          0.954022
```

```
10 2022-08-08          0.879587
11 2022-08-15          0.665843
12 2022-08-22          0.525940
13 2022-08-29          1.000000
14 2022-09-05          0.946315
15 2022-09-12          0.954022
16 2022-09-19          0.933915
17 2022-09-26          0.769664
18 2022-10-03          0.876424
19 2022-10-10          0.798779
20 2022-10-17          0.775931
21 2022-10-24          0.756337
22 2022-10-31          0.000000
23 2022-11-07          0.873193
24 2022-11-14          0.572476
25 2022-11-21          0.653375
26 2022-11-28          0.725836
27 2022-12-05          0.787769
28 2022-12-12          0.734015
29 2022-12-19          0.804006
30 2022-12-26          0.000000
31 2023-01-02          0.832137
```

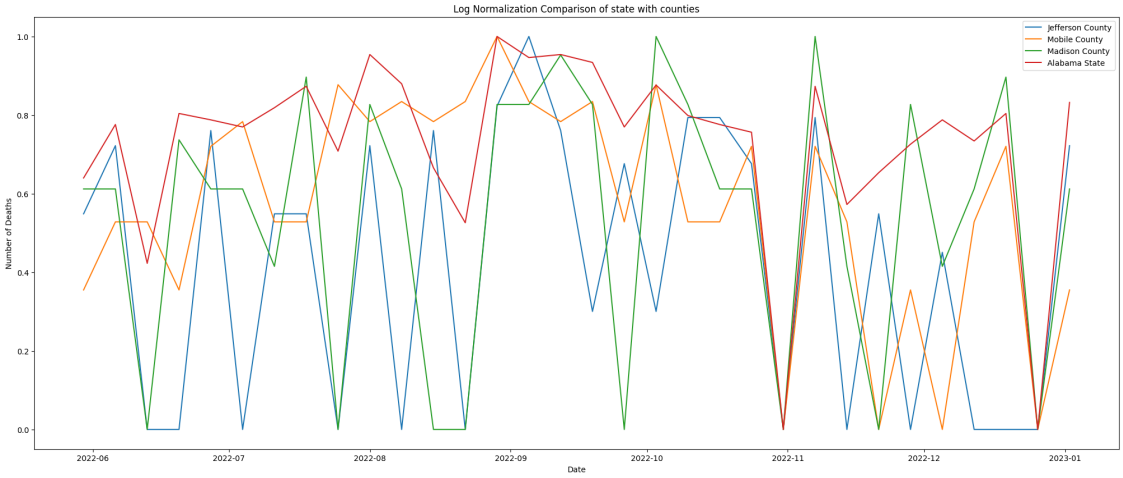
[34]:
```
weekly_deaths_selected_state_given_range_max_normalised.plot(x='Date',␣
↪y='Number of Deaths', title = 'Log normalized plot of deaths  in Albama␣
↪State')
```

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



[35]:
```
plt.figure(figsize=(25,10))
plt.plot(weekly_deaths_selected_county_given_range_max_normalised['Date'],␣
 ↪weekly_deaths_selected_county_given_range_max_normalised['Number of␣
 ↪Deaths'], label='Jefferson County')
plt.plot(weekly_deaths_selected_county_given_range_2_max_normalised['Date'],␣
 ↪weekly_deaths_selected_county_given_range_2_max_normalised['Number of␣
 ↪Deaths'], label='Mobile County')
plt.plot(weekly_deaths_selected_county_given_range_3_max_normalised['Date'],␣
 ↪weekly_deaths_selected_county_given_range_3_max_normalised['Number of␣
 ↪Deaths'], label='Madison County')
plt.plot(weekly_deaths_selected_state_given_range_max_normalised['Date'],␣
 ↪weekly_deaths_selected_state_given_range_max_normalised['Number of Deaths'],␣
 ↪label='Alabama State')
plt.title('Log Normalization Comparison of state with counties')
plt.xlabel('Date')
plt.ylabel('Number of Deaths')
plt.legend()

plt.show()
```



0.1.4 The three counties(Jefferson,Mobile,Madison County) are following state(Alabama State) patterns but not so closely.