

Stage2_task1_Deaths

March 14, 2023

0.1 Generate weekly statistics (mean, median, mode) for number of new deaths across a specific state.

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

```
[2]: #I have selected the Alabama State
selected_state = "AL"
#Reading 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]: #For the selected state Alabama 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()
```

```
[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 Deaths, dtype: int64
```

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

```
new_deaths_selected_state_daily.head()
```

```
[6]:
```

	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

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

```
[7]:
```

	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

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

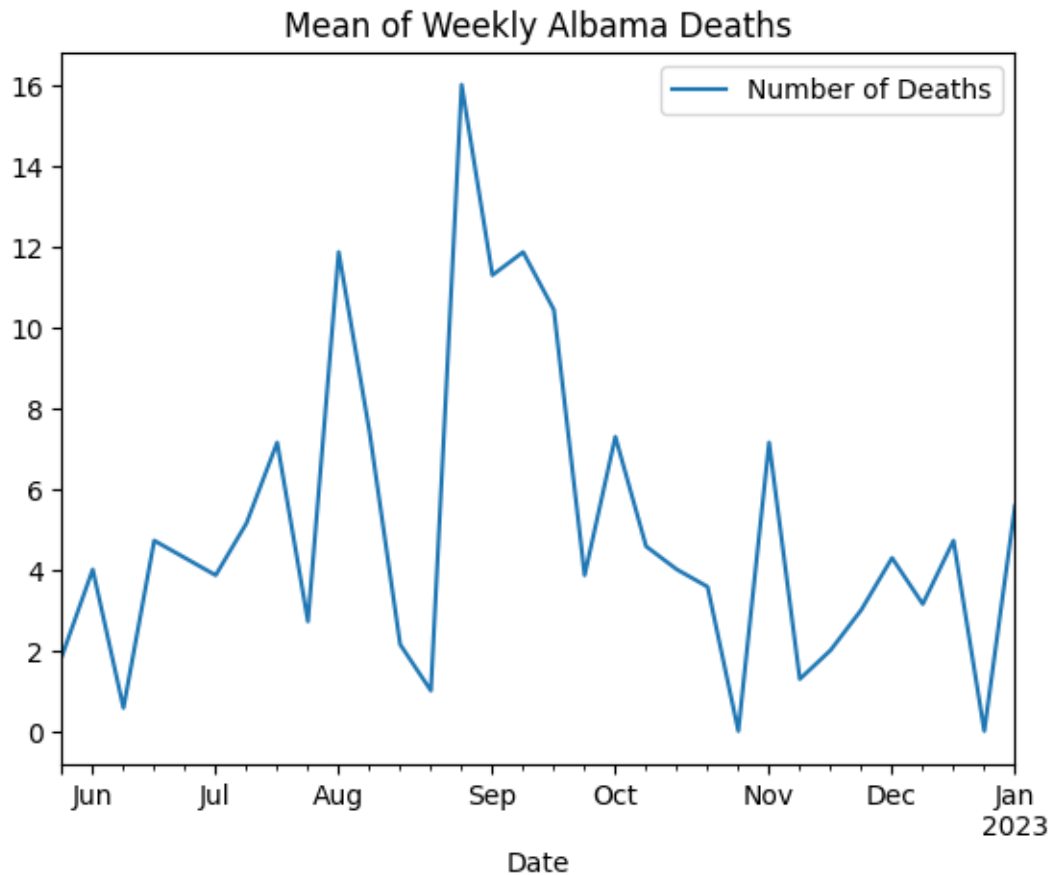
```
[8]:
```

	Date	Number of Deaths
0	2022-05-30	1.857143
1	2022-06-06	4.000000
2	2022-06-13	0.571429
3	2022-06-20	4.714286

4	2022-06-27	4.285714
5	2022-07-04	3.857143
6	2022-07-11	5.142857
7	2022-07-18	7.142857
8	2022-07-25	2.714286
9	2022-08-01	11.857143
10	2022-08-08	7.428571
11	2022-08-15	2.142857
12	2022-08-22	1.000000
13	2022-08-29	16.000000
14	2022-09-05	11.285714
15	2022-09-12	11.857143
16	2022-09-19	10.428571
17	2022-09-26	3.857143
18	2022-10-03	7.285714
19	2022-10-10	4.571429
20	2022-10-17	4.000000
21	2022-10-24	3.571429
22	2022-10-31	0.000000
23	2022-11-07	7.142857
24	2022-11-14	1.285714
25	2022-11-21	2.000000
26	2022-11-28	3.000000
27	2022-12-05	4.285714
28	2022-12-12	3.142857
29	2022-12-19	4.714286
30	2022-12-26	0.000000
31	2023-01-02	5.571429

```
[9]: #Plotting the mean graph
weekly_deaths_mean_selected_state_given_range.plot(x='Date', y='Number of_
→Deaths', title = 'Mean of Weekly Albama Deaths')
```

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



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

```
[10]:      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
```

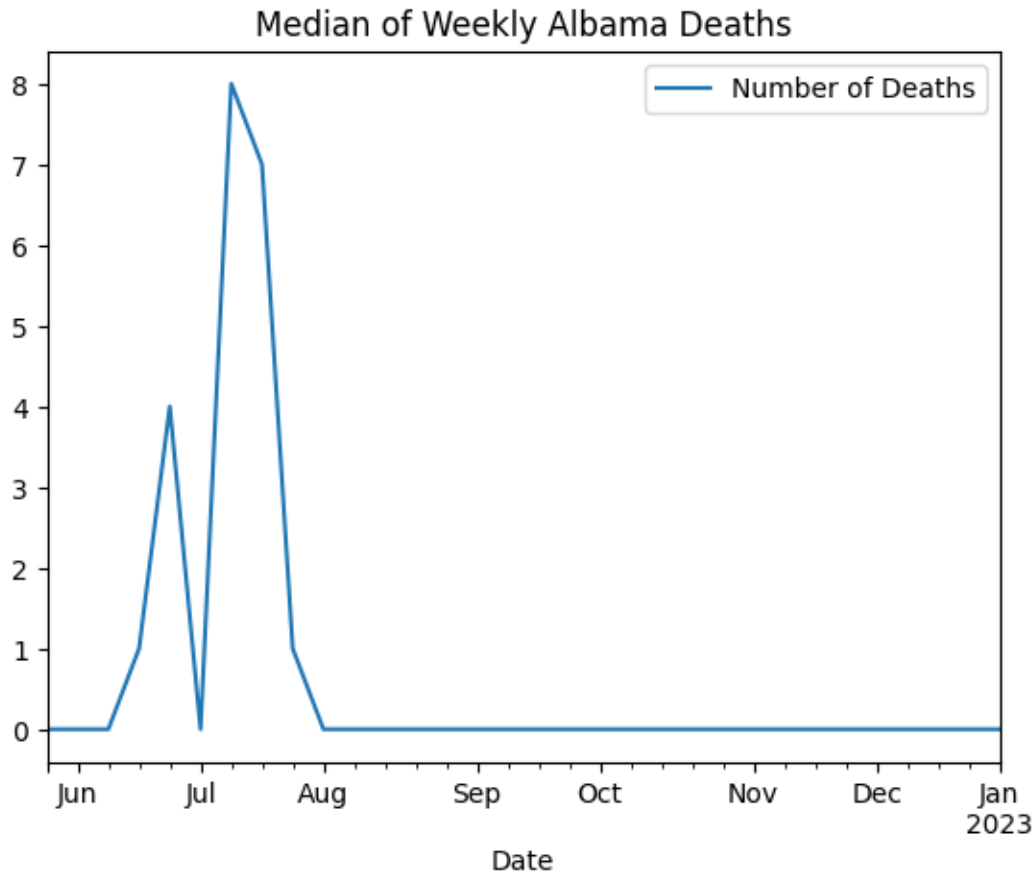
```
[11]: #considering the given range of dates starting from monday. and weekly analysis
      ↪from monday to sunday.
weekly_deaths_median_selected_state_given_range =
      ↪weekly_deaths_median_selected_state[(weekly_deaths_median_selected_state["Date"]
      ↪>= '2022-05-29') & (weekly_deaths_median_selected_state["Date"]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

```
[11]:
```

	Date	Number of Deaths
0	2022-05-30	0.0
1	2022-06-06	0.0
2	2022-06-13	0.0
3	2022-06-20	1.0
4	2022-06-27	4.0
5	2022-07-04	0.0
6	2022-07-11	8.0
7	2022-07-18	7.0
8	2022-07-25	1.0
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_deaths_median_selected_state_given_range.plot(x='Date', y='Number of Deaths', title = 'Median of Weekly Alabama Deaths')
```

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



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

```
[13]:      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
```

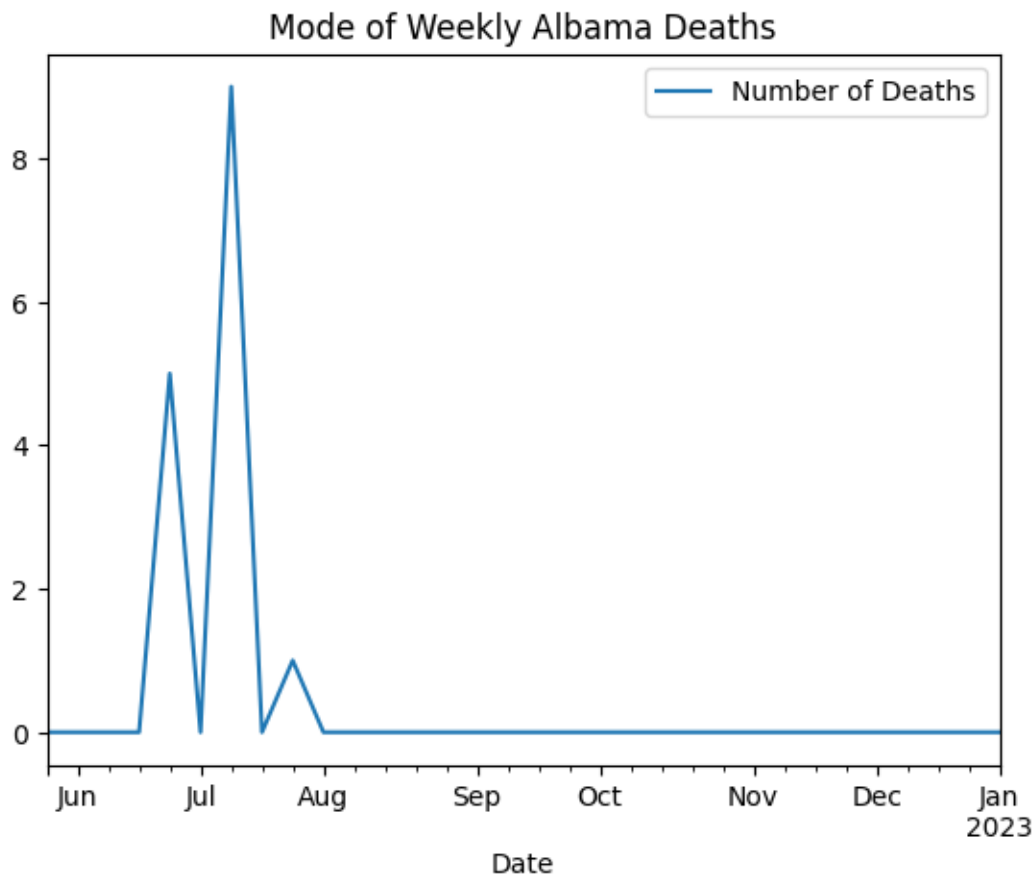
```
[14]: #considering the given range of dates starting from monday. and weekly analysis
      ↳from monday to sunday.
weekly_deaths_mode_selected_state_given_range =
      ↳weekly_deaths_mode_selected_state[weekly_deaths_mode_selected_state["Date"]
      ↳>= '2022-05-29') & (weekly_deaths_mode_selected_state["Date"]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

```
[14]:      Date  Number of Deaths
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          5.0
5 2022-07-04          0.0
6 2022-07-11          9.0
7 2022-07-18          0.0
8 2022-07-25          1.0
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]: #Plotting the mode graph
weekly_deaths_mode_selected_state_given_range.plot(x='Date', y='Number of
↳Deaths', title = 'Mode of Weekly Albama Deaths')
```

```
[15]: <AxesSubplot: title={'center': 'Mode of Weekly Albama Deaths'}, xlabel='Date'>
```



```
[16]: #plotting the mean, median and mode graphs.
plt.plot(weekly_deaths_mean_selected_state_given_range['Date'],
↳weekly_deaths_mean_selected_state_given_range['Number of Deaths'],
↳label='Mean of Alabama State')
```

```

plt.plot(weekly_deaths_median_selected_state_given_range['Date'],  

↪weekly_deaths_median_selected_state_given_range['Number of Deaths'],  

↪label='Median of Alabama State')
plt.plot(weekly_deaths_mode_selected_state_given_range['Date'],  

↪weekly_deaths_mode_selected_state_given_range['Number of Deaths'],  

↪label='Mode of Alabama State')
plt.title('Mean,Median and Mode of Deaths in Alabama State')
plt.xlabel('Date')
plt.ylabel('Number of Deaths')
plt.legend()
plt.show()

```

