# CSE 544

# Probability and Statistics for Data Scientists

# Mini Project

## Group Members



| Name | SBU ID |
|---|---|
| Aayushi Nirmal | 113504530 |
| Drushti Mewada | 113276901 |
| Karan Dipesh Gada | 113082700 |
| Tanishq Sandeep Mehra | 112078038 |

**Mandatory task 1**

Cleaning Task:

Following are the steps we performed in this task.

1. The confirmed and deaths for the two states are in a cumulative fashion. Our first step was to process the data and get per-day statistics of confirmed and deaths for the two states in each day.
   Python file: “cleaning_part1.py”

2. After Step 1, there were existence of incorrect negative values in certain records of the table. To keep track of the negative instances, we marked it with NaN first and then replaced it with mean of the neighboring valid data points to make it meaningful to our operations.
   Python file: “cleaning_part2.py”

3. We also performed outlier detection using Tukey’s Outlier Detection the processed data from Step 2.

Output: **Using tukey's outliers detection to detect outliers in the covid data we have**

*We found **104** outliers* Since the outliers hold a significant importance in our data, it might affect our further analysis and testing so we are avoiding the removal of outliers

Python file: “cleaning_part3.ipynb”

Part 3 Output:

```python
from sklearn.preprocessing import MinMaxScaler
import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
plt.style.use('ggplot')
```

```python
covid_data = pd.read_csv('12_final_processed.csv')
```

**Using tukey's outliers detection to detect outliers in the covid data we have**

*We found **104** outliers* Since the outliers hold a significant importance in our data, it might affect our further analysis and testing so we are avoiding the removal of outliers

```python
def tukeys_outliers(df, col_arr):
    """
  We take all the columns of dataframe and find 1st and 3rd quartile and apply tukeys rule to find the outliers
    """
    outlier_indices = []

    for col in col_arr:
        # 1st quartile (25%)
        Q1 = np.percentile(df[col], 25)
        # 3rd quartile (75%)
        Q3 = np.percentile(df[col], 75)
        # Interquartile range (IQR)
        IQR = Q3 - Q1
        outlier_cal = 1.5 * IQR

        # Determine a list of indices of outliers for feature col
        outlier_list_col = df[(df[col] < Q1 - outlier_cal) | (df[col] > Q3 + outlier_cal)].index
        # append the found outlier indices for col to the list of outlier indices
        outlier_indices.extend(outlier_list_col)
#         print(df[(df[col] < Q1 - outlier_cal) | (df[col] > Q3 + outlier_cal)])
#         print("outliers in "+str(col) + " : " + str(len(outlier_indices)))
    print(set(outlier_indices), len(set(outlier_indices)))
    df.iloc[list(set(outlier_indices))].to_csv('outlier_data.csv',index=False)
    # select observations containing more than 2 outliers
    outlier_indices = Counter(outlier_indices)
    multiple_outliers = list(k for k, v in outlier_indices.items())

    return multiple_outliers
```

```python
col_arr = ['MI confirmed','MN confirmed','MI deaths','MN deaths']
outliers = tukeys_outliers(covid_data, features)
```

```
{79, 83, 84, 85, 90, 92, 94, 97, 101, 135, 267, 268, 269, 271, 274, 276, 277, 280, 281, 282, 284, 285, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298,
```

## Question 2A)

Outputs:

Python file: “ques2_a.ipynb”

## EWMA Predictions :

Predictions are for day 22 (value at index 21) to day 28 (value at index 27) i.e. last week of August.

```
================== EWMA ====================
```

```
MI cases predictions:

Alpha = 0.5 :
Days  Values
21    296
22    861
23    812
24    845
25    898
26    870
27    866
dtype: int64

Alpha = 0.8 :
Days  Values
21     111
22    1163
23     843
24     871
25     935
26     861
27     862
dtype: int64

MN cases predictions:

Alpha = 0.5 :
Days  Values
21    715
22    724
23    720
24    717
25    563
26    546
27    850
dtype: int64

Alpha = 0.8 :
Days  Values
21     791
22     745
23     722
24     715
25     470
26     517
27    1026
dtype: int64
```

```
dtype: int64

MI deaths predictions:

Alpha = 0.5 :
Days  Values
21      6
22     13
23      8
24      6
25     13
26      9
27     12
dtype: int64

Alpha = 0.8 :
Days  Values
21      2
22     17
23      6
24      4
25     17
26      8
27     14
dtype: int64

MN deaths predictions:

Alpha = 0.5 :
Days  Values
21      8
22      8
23      7
24      5
25      6
26     10
27     11
dtype: int64

Alpha = 0.8 :
Days  Values
21      8
22      8
23      6
24      4
25      7
26     12
27     12
dtype: int64
```

## AR Predictions :

Predictions in the output list are for the last week of August, starting from 22 August to 28 August.

================== AR ====================

```
print(ds_MN_deaths_AR_5)
```

```
MI cases predictions:

p = 3 :
[545, 603, 805, 719, 719, 660, 671]

p = 5 :
[400, 450, 674, 528, 548, 508, 587]

MN cases predictions:

p = 3 :
[713, 614, 550, 600, 662, 696, 672]

p = 5 :
[749, 704, 572, 547, 556, 655, 702]

MI deaths predictions:

p = 3 :
[10, 10, 12, 10, 10, 10, 10]

p = 5 :
[17, 8, 14, 10, 13, 8, 12]

MN deaths predictions:

p = 3 :
[12, 8, 7, 9, 8, 7, 8]

p = 5 :
[11, 8, 7, 9, 9, 7, 8]
```

# MSE values :

```
================= Errors - MSE =================

MSE - MI cases:

Alpha = 0.5 :
187186.2857142857

Alpha = 0.8 :
272754.28571428574

MSE - MN cases:

Alpha = 0.5 :
66590.0

Alpha = 0.8 :
76851.28571428571

MSE - MI deaths:

Alpha = 0.5 :
97.28571428571429

Alpha = 0.8 :
153.14285714285714

MSE - MN deaths:

Alpha = 0.5 :
20.571428571428573

Alpha = 0.8 :
19.714285714285715
```

## MAPE values :

```
================= Errors - MAPE =================

MAPE - MI cases:

Alpha = 0.5 :
39.18825711506203

Alpha = 0.8 :
103.06784124336295

MAPE - MN cases:

Alpha = 0.5 :
19.365797899709953

Alpha = 0.8 :
0.0

MAPE - MI deaths:

Alpha = 0.5 :
68.8949938949939

Alpha = 0.8 :
107.72308923569427

MAPE - MN deaths:

Alpha = 0.5 :
62.23716759431045

Alpha = 0.8 :
51.785714285714285
```

Question 2 b)
i)

```
from collections import Counter
from scipy.stats import gamma
from sklearn.preprocessing import MinMaxScaler
import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
from datetime import date
plt.style.use('ggplot')
```

```
[3] from google.colab import drive
    drive.mount('/content/gdrive')
    import os
    os.chdir('/content/gdrive/MyDrive/CSE544-ProbStats/Sample-Datasets/MiniProject')

    Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

```
[4] df = pd.read_csv('12_final_processed.csv')
```

```
[5] #Filtering out data for feb 21 and march 21
    feb_21 = df[(df['Date'] >= '2021-02-01') & (df['Date'] <= '2021-02-28')]
    march_21 = df[(df['Date'] >= '2021-03-01') & (df['Date'] <= '2021-03-31')]
```

```
[6] #Calculating the mean for each state/column for feb 21 and march 21

    feb_21_mean_death_MI = feb_21['MI deaths'].mean()
    feb_21_mean_cases_MI = feb_21['MI confirmed'].mean()


    march_21_mean_death_MI = march_21['MI deaths'].mean()
    march_21_mean_cases_MI = march_21['MI confirmed'].mean()


    feb_21_mean_death_MN = feb_21['MN deaths'].mean()
    feb_21_mean_cases_MN = feb_21['MN confirmed'].mean()
```

```
[6]
    march_21_mean_death_MN = march_21['MN deaths'].mean()
    march_21_mean_cases_MN = march_21['MN confirmed'].mean()

    #calculating corrected variance for use in tests

    def variance(col_data):
        sq_sum = 0
        mean_col_data = col_data.mean()
        n = len(col_data)
        for i in col_data:
            sq_sum = sq_sum + (i -mean_col_data)*(i-mean_col_data)
        return sq_sum/(n-1)
```

**Summary of the results that we run below:**

We *accept* the NULL hypothesis for deaths in MN for two population Wald's test, z-test, t-test and unpaired t-test

We *accept* the NULL hypothesis for cases in MN for two sample unpaired t-test

We *accept* the NULL hypothesis for death in MI for two sample unpaired t-test

We *reject* the NULL hypothesis for all the other cases ii)

**Wald's one Sample testing for confirmed cases and deaths in MI and MN**

*NULL hypothesis* H0: mean of confirmed deaths/cases for feb 21 = mean of deaths/cases for march 21

*Alternate hypothesis* H1: mean of confirmed deaths/cases for feb 21 != mean of deaths/cases for march 21

[7]

```
# walds one sample testing for deaths and confirmed cases for both states

def walds_1_testing(march_21_mean,feb_21_mean,march_21):
  w_1_numerator = march_21_mean - feb_21_mean
  w_1_denominator = np.sqrt(march_21_mean/len(march_21))
  return np.abs(w_1_numerator/w_1_denominator)



#for death in MI
w_1_result_death_MI = walds_1_testing(march_21_mean_death_MI,feb_21_mean_death_MI,march_21)
if(w_1_result_death_MI>1.96):
  print("Walds 1 sample testing for mean of death in MI is w="+str(w_1_result_death_MI) +" which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis")
else:
  print("Walds 1 sample testing for mean of death in MI is  w="+str(w_1_result_death_MI)+ " which is less than z-alpha/2 = 1.96 so we accept the NULL hypothesis")

#for cases in MI
w_1_result_cases_MI = walds_1_testing(march_21_mean_cases_MI,feb_21_mean_cases_MI,march_21)
if(w_1_result_cases_MI>1.96):
  print("Walds 1 sample testing for mean of cases in MI is w="+str(w_1_result_cases_MI) +" which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis");
else:
  print("Walds 1 sample testing for mean of cases in MI is w="+str(w_1_result_cases_MI)+ " which is less than z-alpha/2 = 1.96 so we accept the NULL hypothesis")



#for death in MN
w_1_result_death_MN = walds_1_testing(march_21_mean_death_MN,feb_21_mean_death_MN,march_21)
if(w_1_result_death_MN>1.96):
  print("\nWalds 1 sample testing for mean of death in MN is w="+str(w_1_result_death_MN) +" which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis")
else:
  print("\nWalds 1 sample testing for mean of death in MN is  w="+str(w_1_result_death_MN)+ " which is less than z-alpha/2 = 1.96 so we accept the NULL hypothesis")

#for cases in MN
w_1_result_cases_MN = walds_1_testing(march_21_mean_cases_MN,feb_21_mean_cases_MN,march_21)
if(w_1_result_cases_MN>1.96):
  print("Walds 1 sample testing for mean of cases in MN is w="+str(w_1_result_cases_MN) +" which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis");
else:
  print("Walds 1 sample testing for mean of cases in MN is w="+str(w_1_result_cases_MN)+ " which is less than z-alpha/2 = 1.96 so we accept the NULL hypothesis")
```

```
Walds 1 sample testing for mean of death in MI is w=19.18673565170543 which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis
Walds 1 sample testing for mean of cases in MI is w=193.96139177500143 which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis

Walds 1 sample testing for mean of death in MN is w=2.4912289212760137 which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis
Walds 1 sample testing for mean of cases in MN is w=51.90948310795567 which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis
```

Output:

```
Walds 1 sample testing for mean of death in MI is w=19.18673565170543
which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis
Walds 1 sample testing for mean of cases in MI is w=193.96139177500143
which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis

Walds 1 sample testing for mean of death in MN is w=2.4912289212760137
which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis
Walds 1 sample testing for mean of cases in MN is w=51.90948310795567
which is greater than z-alpha/2 = 1.96 so we reject the NULL hypothesis
```

iii)

**Wald's two population testing for confirmed cases and deaths in MI and MN**

*NULL hypothesis* H0: mean of confirmed deaths/cases for feb 21 = mean of deaths/cases for march 21

*Alternate hypothesis* H1: mean of confirmed deaths/cases for feb 21 != mean of deaths/cases for march 21

```
##walds 2 population testing for deaths and confirmed cases for both states

def walds_2_testing(march_21_mean,feb_21_mean,march_21, feb_21):
    #using values of both months for calculating standard error
  se = np.sqrt((march_21_mean/len(march_21)) + (feb_21_mean/len(feb_21)))
  w_2_result = (march_21_mean - feb_21_mean)/se
  return np.abs(w_2_result)


#for death calculation in MI
w_2_result_death_MI = walds_2_testing(march_21_mean_death_MI,feb_21_mean_death_MI,march_21, feb_21)
if(w_2_result_death_MI>1.96):
  print("walds 2 sample testing for mean of death in MI is w="+str(w_2_result_death_MI) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
else:
  print("walds 2 sample testing for mean of death in MI is w="+str(w_2_result_death_MI)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")



#for cases calculation in MI
w_2_result_cases_MI = walds_2_testing(march_21_mean_cases_MI,feb_21_mean_cases_MI,march_21, feb_21)
if(w_2_result_cases_MI>1.96):
  print("walds 2 sample testing for mean of cases in MI is w="+str(w_2_result_cases_MI) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
else:
  print("walds 2 sample testing for mean of cases in MI is w="+str(w_2_result_cases_MI)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")


#for death calculation in MN
w_2_result_death_MN = walds_2_testing(march_21_mean_death_MN,feb_21_mean_death_MN,march_21, feb_21)
if(w_2_result_death_MN>1.96):
  print("\nwalds 2 sample testing for mean of death in MN is w="+str(w_2_result_death_MN) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
else:
  print("\nwalds 2 sample testing for mean of death in MN is w="+str(w_2_result_death_MN)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")



#for cases calculation in MN
w_2_result_cases_MN = walds_2_testing(march_21_mean_cases_MN,feb_21_mean_cases_MN,march_21, feb_21)
if(w_2_result_cases_MN>1.96):
  print("walds 2 sample testing for mean of cases in MN is w="+str(w_2_result_cases_MN) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
else:
  print("walds 2 sample testing for mean of cases in MN is w="+str(w_2_result_cases_MN)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")
```

```
walds 2 sample testing for mean of death in MI is w=11.143749021603197 which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis
walds 2 sample testing for mean of cases in MI is w=162.15637581729627 which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis

walds 2 sample testing for mean of death in MN is w=1.7783708657644213 which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis
walds 2 sample testing for mean of cases in MN is w=38.69473932283188 which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis
```

Output:

```
walds 2 sample testing for mean of death in MI is w=11.143749021603197
which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis
walds 2 sample testing for mean of cases in MI is w=162.15637581729627
which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis

walds 2 sample testing for mean of death in MN is w=1.7783708657644213
which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis
walds 2 sample testing for mean of cases in MN is w=38.69473932283188
which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis
```

iv)

**z-testing for confirmed cases and deaths in MI and MN**

*NULL hypothesis* H0: mean of confirmed deaths/cases for feb 21 = mean of deaths/cases for march 21

*Alternate hypothesis* H1: mean of confirmed deaths/cases for feb 21 != mean of deaths/cases for march 21

```
[9] #z testing for deaths and confirmed cases for both states

    def z_test(march_21_mean, feb_21_mean, col_name):
        z_num = march_21_mean - feb_21_mean
        z_den = np.sqrt(variance(df[[col_name]].values)/(len(df)))

        z_result = np.abs(z_num/z_den)
        return z_result


    #for death in MI
    z_result_death_MI = z_test(march_21_mean_death_MI, feb_21_mean_death_MI, 'MI deaths')
    if(z_result_death_MI>1.96):
      print("z-test for mean of death in MI is w="+str(z_result_death_MI) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
    else:
      print("z-test for mean of death in MI is  w="+str(z_result_death_MI)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")

    #for cases in MI
    z_result_cases_MI = z_test(march_21_mean_cases_MI, feb_21_mean_cases_MI, 'MI confirmed')
    if(z_result_cases_MI>1.96):
      print("z-test for mean of cases in MI is w="+str(z_result_cases_MI) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
    else:
      print("z-test for mean of cases in MI is w="+str(z_result_cases_MI)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")

    #for death in MN
    z_result_death_MN = z_test(march_21_mean_death_MN, feb_21_mean_death_MN, 'MN deaths')
    if(z_result_death_MN>1.96):
      print("z-test for mean of death in MN is w="+str(z_result_death_MN) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
    else:
      print("\nz-test for mean of death in MN is  w="+str(z_result_death_MN)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")

    #for cases in MN
    z_result_cases_MN = z_test(march_21_mean_cases_MN, feb_21_mean_cases_MN, 'MN confirmed')
    if(z_result_cases_MN>1.96):
      print("z-test for mean of cases in MN is w="+str(z_result_cases_MN) +" which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis")
    else:
      print("z-test for mean of cases in MN is w="+str(z_result_cases_MN)+ " which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis")
```

```
z-test for mean of death in MI is w=[5.66552914] which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis
z-test for mean of cases in MI is w=[16.52716098] which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis

z-test for mean of death in MN is  w=[1.47842417] which is less than z_alpha/2 = 1.96 so accept the NULL hypothesis
z-test for mean of cases in MN is w=[3.54678737] which is greater than z_alpha/2 = 1.96 so reject the NULL hypothesis
```

**Assumptions in z-test/ Is z-test applicable?**

z test only works if either the data is large or normally distributed Here, the data points are greater than 30. So, we can say that z-test is applicable eventhough data is not normally distributed.

Output:

```
z-test for mean of death in MI is w=[5.66552914] which is greater than
z_alpha/2 = 1.96 so reject the NULL hypothesis
z-test for mean of cases in MI is w=[16.52716098] which is greater than
z_alpha/2 = 1.96 so reject the NULL hypothesis

z-test for mean of death in MN is  w=[1.47842417] which is less than
z_alpha/2 = 1.96 so accept the NULL hypothesis
z-test for mean of cases in MN is w=[3.54678737] which is greater than
z_alpha/2 = 1.96 so reject the NULL hypothesis
```

**Assumptions in z-test/ Is z-test applicable?**

z test only works if either the data is large or normally distributed Here, the data points are greater than 30. So, we can say that z-test is applicable eventhough data is not normally distributed.

v)

**T one sample testing for confirmed cases and deaths in MI and MN**

*NULL hypothesis* H0: mean of confirmed deaths/cases for feb 21 = mean of deaths/cases for march 21

*Alternate hypothesis* H1: mean of confirmed deaths/cases for feb 21 != mean of deaths/cases for march 21

```
[ ]
# T one sample testing for deaths and confirmed cases for both states

def T_1_sample(col,march_21_mean,feb_21_mean,march_21):
  t_1_num = march_21_mean - feb_21_mean
  t_1_den = np.sqrt(variance(march_21[[col]].values)/len(march_21))
  return np.abs(t_1_num/t_1_den)

# for deaths in MI
t_1_result_death_MI =T_1_sample('MI deaths',march_21_mean_death_MI,feb_21_mean_death_MI,march_21)
if(t_1_result_death_MI>2.3596):
  print("T-Test 1 sample testing for mean of death in MI is T1="+str(t_1_result_death_MI) +" which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis")
else:
  print("T-Test 1 sample testing for mean of death in MI is  T1="+str(t_1_result_death_MI)+ " which is less than t(n-1,alpha/2) = 2.3596 so accept the NULL hypothesis")


# for cases in MI
t_1_result_cases_MI =T_1_sample('MI confirmed',march_21_mean_cases_MI,feb_21_mean_cases_MI,march_21)
if(t_1_result_cases_MI>2.3596):
  print("T-Test 1 sample testing for mean of cases in MI is T1="+str(t_1_result_cases_MI) +" which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis")
else:
  print("T-Test 1 sample testing for mean of cases in MI is T1="+str(t_1_result_cases_MI)+ " which is less than t(n-1,alpha/2) = 2.3596 so accept the NULL hypothesis")

# for deaths in MN
t_1_result_death_MN =T_1_sample('MN deaths',march_21_mean_death_MN,feb_21_mean_death_MN,march_21)
if(t_1_result_death_MN>2.3596):
  print("T-Test 1 sample testing for mean of death in MN is T1="+str(t_1_result_death_MN) +" which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis")
else:
  print("\nT-Test 1 sample testing for mean of death in MN is  T1="+str(t_1_result_death_MN)+ " which is less than t(n-1,alpha/2) = 2.3596 so accept the NULL hypothesis")


# for cases in MN
t_1_result_cases_MN =T_1_sample('MN confirmed',march_21_mean_cases_MN,feb_21_mean_cases_MN,march_21)
if(t_1_result_cases_MN>2.3596):
  print("T-Test 1 sample testing for mean of cases in MN is T1="+str(t_1_result_cases_MN) +" which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis")
else:
  print("T-Test 1 sample testing for mean of cases in MN is T1="+str(t_1_result_cases_MN)+ " which is less than t(n-1,alpha/2) = 2.3596 so accept the NULL hypothesis")
```

```
T-Test 1 sample testing for mean of death in MI is T1=[5.28586887] which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis
T-Test 1 sample testing for mean of cases in MI is T1=[4.86763697] which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis

T-Test 1 sample testing for mean of death in MN is  T1=[0.35109704] which is less than t(n-1,alpha/2) = 2.3596 so accept the NULL hypothesis
T-Test 1 sample testing for mean of cases in MN is T1=[4.63262675] which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis
```

**Assumptions/Is t-test applicable?**

For this course T-test assumes that the data is normally distributed. But here we do not have normally distributed data so it is not a right choice to apply t-test

Output:

```
T-Test 1 sample testing for mean of death in MI is T1=[5.28586887] which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis
T-Test 1 sample testing for mean of cases in MI is T1=[4.86763697] which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis

T-Test 1 sample testing for mean of death in MN is  T1=[0.35109704] which is less than t(n-1,alpha/2) = 2.3596 so accept the NULL hypothesis
T-Test 1 sample testing for mean of cases in MN is T1=[4.63262675] which is greater than t(n-1,alpha/2) = 2.3596 so reject the NULL hypothesis
```

**Assumptions/Is t-test applicable?**

For this course T-test assumes that the data is normally distributed. But here we do not have normally distributed data so it is not a right choice to apply t-test

vi)

**T two sample testing for confirmed cases and deaths in MI and MN**

*NULL hypothesis* H0: mean of confirmed deaths/cases for feb 21 = mean of deaths/cases for march 21

*Alternate hypothesis* H1: mean of confirmed deaths/cases for feb 21 != mean of deaths/cases for march 21

```
[ ] # Unpaired T two sample testing for deaths and confirmed cases for both states.
    #Here we consider both samples so m=31 and n =28 so threshold will be t(n+m-2, alpha/2)

    def unpaired_T(feb_21_mean, march_21_mean, col):
        T2_num = feb_21_mean - march_21_mean

        feb_21_var = variance(feb_21[[col]].values)
        march_21_var = variance(march_21[[col]].values)
        T2_den = np.sqrt(march_21_var/len(march_21) + feb_21_var/len(feb_21))

        #T test unpaired result
        return np.abs(T2_num/T2_den)

    # T 2 sample test for deaths in MI
    T2_death_MI = unpaired_T(feb_21_mean_death_MI, march_21_mean_death_MI, 'MI deaths')
    if(T2_death_MI>2.3022):
      print("T two sample unpaired  testing for mean of death in MI is T="+str(T2_death_MI) +" which is greater than t(57,alpha/2) = 2.3022 so reject the NULL hypothesis")
    else:
      print("T two sample unpaired  testing for mean of death in MI is T="+str(T2_death_MI)+ " which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis")


    # T 2 sample test for cases in MI
    T2_cases_MI = unpaired_T(feb_21_mean_cases_MI, march_21_mean_cases_MI, 'MI confirmed')
    if(T2_cases_MI>2.3022):
      print("T two sample unpaired  testing for mean of cases in MI is T="+str(T2_cases_MI) +" which is greater than t(57,alpha/2) = 2.3022 so reject the NULL hypothesis")
    else:
      print("T two sample unpaired  testing for mean of cases in MI is T="+str(T2_cases_MI)+ " which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis")

    # T 2 sample test for deaths in MN
    T2_death_MN = unpaired_T(feb_21_mean_death_MN, march_21_mean_death_MN, 'MI deaths')
    if(T2_death_MN>2.3022):
      print("T two sample unpaired  testing for mean of death in MN is T="+str(T2_death_MN) +" which is greater than t(57,alpha/2) = 2.3022 so reject the NULL hypothesis")
    else:
      print("\nT two sample unpaired  testing for mean of death in MN is T="+str(T2_death_MN)+ " which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis")


    # T 2 sample test for cases in MN
    T2_cases_MN = unpaired_T(feb_21_mean_cases_MN, march_21_mean_cases_MN, 'MI confirmed')
    if(T2_cases_MN>2.3022):
      print("T two sample unpaired  testing for mean of cases in MN is T="+str(T2_cases_MN) +" which is greater than t(57,alpha/2) = 2.3022 so reject the NULL hypothesis")
    else:
      print("T two sample unpaired  testing for mean of cases in MN is T="+str(T2_cases_MN)+ " which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis")

    T two sample unpaired  testing for mean of death in MI is T=[2.18521476] which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis
    T two sample unpaired  testing for mean of cases in MI is T=[4.67047039] which is greater than t(57,alpha/2) = 2.3022 so reject the NULL hypothesis

    T two sample unpaired  testing for mean of death in MN is T=[0.21816034] which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis
    T two sample unpaired  testing for mean of cases in MN is T=[0.7358662] which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis
```

Output:

```
T two sample unpaired  testing for mean of death in MI is T=[2.18521476] which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis
T two sample unpaired  testing for mean of cases in MI is T=[4.67047039] which is greater than t(57,alpha/2) = 2.3022 so reject the NULL hypothesis

T two sample unpaired  testing for mean of death in MN is T=[0.21816034] which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis
T two sample unpaired  testing for mean of cases in MN is T=[0.7358662] which is less than t(57,alpha/2) = 2.3022 so accept the NULL hypothesis
```

## Question 2c

**KS 1 Sample Test for Confirmed Cases**

Considering State 1 distribution to be Poisson

```
The MME values :  [4245.369565217391]
Since Max distance(0.641214) > 0.05, we reject Null Hypothesis=>(MN
confirmed has same distribution as MI confirmed having true distribution
of poisson)
```

Considering State 1 distribution to be Geometric

```
The MME values :  [0.00023555075350637778]
Since Max distance(0.122759) > 0.05, we reject Null Hypothesis=>(MN
confirmed has same distribution as MI confirmed having true distribution
of geometric)
```

Considering State 1 distribution to be Binomial

```
The MME values :  [-1.3237902989862356, -3206.9804171163014]
Since Max distance(1.000000) > 0.05, we reject Null Hypothesis=>(MN
confirmed has same distribution as MI confirmed having true distribution
of binomial)
```

**Note:** The MME values for binomial distribution are negative and while calculating CDF for values of State1, the CDF was always 1

**KS 1 Sample Test for Deaths**

Considering State 1 distribution to be Poisson

```
The MME values :  [66.07608695652173]
Since Max distance(0.637416) > 0.05, we reject Null Hypothesis=>(MN deaths
has same distribution as MI deaths having true distribution of poisson)
```

Considering State 1 distribution to be Geometric

```
The MME values :  [0.015134068103306466]
Since Max distance(0.213130) > 0.05, we reject Null Hypothesis=>(MN deaths
has same distribution as MI deaths having true distribution of geometric)
```

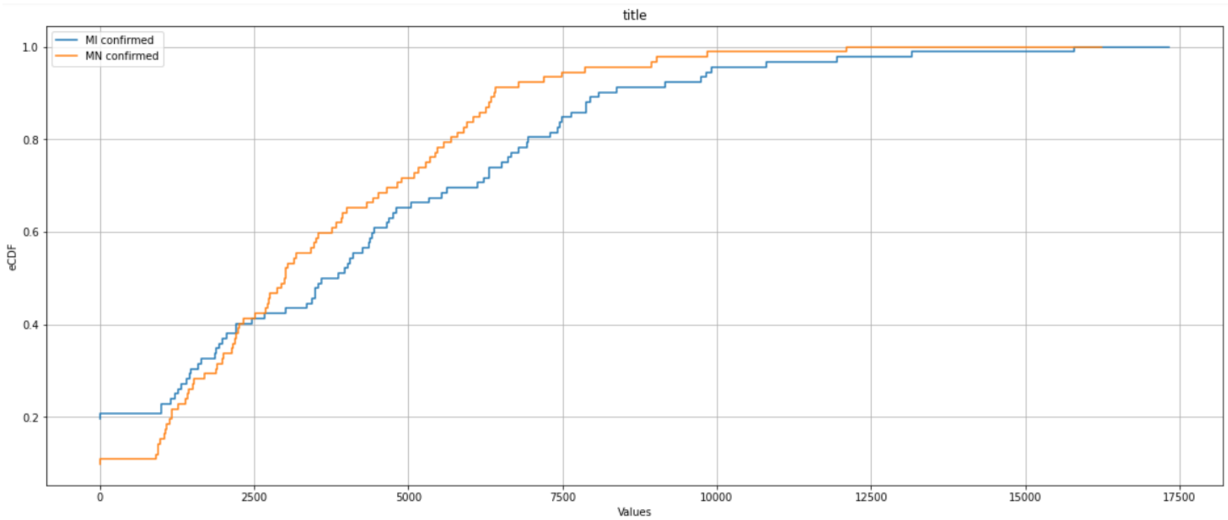
Considering State 1 distribution to be Binomial

```
The MME values :  [-0.9279563159766825, -71.20603181301271]
Since Max distance(1.000000) > 0.05, we reject Null Hypothesis=>(MN deaths
has same distribution as MI deaths having true distribution of binomial)
```

**Note:** The MME values for binomial distribution are negative and while calculating CDF for values of State1, the CDF was always 1

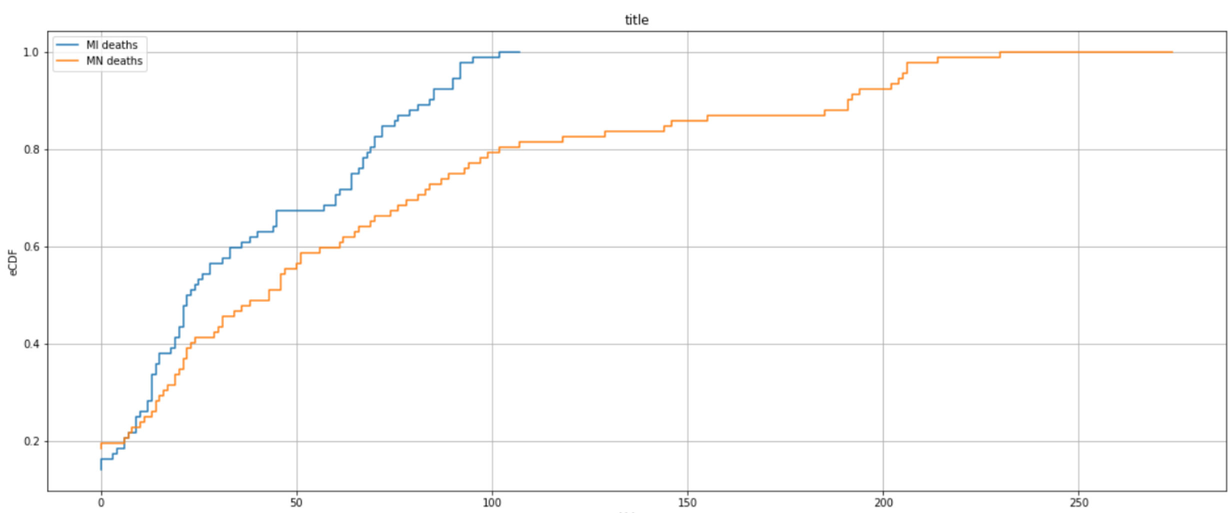
**KS 2 Sample Test for Daily Cases**

```
Since Max distance(0.173913) > 0.05, we reject Null Hypothesis=>(MN
confirmed has same distribution as MI confirmed)
```




**KS 2 Sample Test for Deaths**

```
Since Max distance(0.217391) > 0.05, we reject Null Hypothesis=>(MN deaths
has same distribution as MI deaths)
```




**Permutation Test for Daily Cases with 1000 permutations**

```
The p-value is  0.224
Since the p-value > 0.05, we accept the Null Hypothesis=>(MI confirmed has
same distribution as MN confirmed)
```

**Permutation Test for Deaths with 1000 permutations**

```
The p-value is  0.001
Since the p-value <= 0.05, we reject the Null Hypothesis=>(MI deaths has same distribution as MN deaths)
```

**Note:** Seeing the graphs for KS 2 Sample test we can say that MI confirmed and MN confirmed have a similar looking distribution but strict threshold made us reject null hypothesis=>MI and MN confirmed cases have same distribution. However, P-test gave us the rest that MI confirmed and MN confirmed have similar distribution.

Question 20

Poisson distribution $P_X(x) = \dfrac{e^{-\lambda} \cdot \lambda^{x}}{x!}$ ; $x \geq 0$

Exponential distribution [Prior] $f(\lambda) = k e^{-k\lambda}$; $\lambda \sim Exp(k)$

Given,

$D_0 = \{X_1, \ldots \ldots X_n\} \sim Poisson(\lambda)$
[First 28 days]

where $\lambda \sim Exp(k)$

Given, mean of prior $= \beta = \lambda_{MME}$

$\Rightarrow E[f(\lambda)] = \dfrac{1}{k} = \beta = \lambda_{MME} = \dfrac{n}{\sum_{i=1}^{n} X_i}$

$\therefore k = \dfrac{1}{\beta} = \dfrac{\sum_{i=1}^{n} x_i}{n}$

Using fifth week data to calculate posterior 1

$D_1 = \{X_{n+1} \ldots \ldots X_m\}$
[Fifth week] $m = 7$

$f(\lambda / D_1) \propto L(\lambda) \cdot f(\lambda)$

$\propto \prod_{i=n+1}^{m} \dfrac{e^{-\lambda} \cdot \lambda^{x_i}}{x_i!} \cdot \dfrac{1}{\beta} e^{-\lambda/\beta}$

$$f(\lambda/D_1) \;\alpha\; \frac{e^{-(m-(n+1))\lambda} \cdot \lambda^{\Sigma x_i}}{\prod_{i=n+1}^{m} x_i!} \cdot \frac{1}{\beta} e^{-\lambda/\beta}$$

$$\alpha \; \frac{e^{-\lambda(m-(n+1) + 1/\beta)} \cdot \lambda^{\sum_{n+1}^{m} x_i}}{\beta \cdot \sum_{n+1}^{m} x_i!}$$

Removing constants out of the proportionality, we get

$$f(\lambda/D_1) \;\alpha\; e^{-\lambda(m-n-1+1/\beta)} \cdot \lambda^{\Sigma x_i}$$

which is equivalent to a gamma distribution, where

Gamma distribution : $f(\lambda) \;\alpha\; \lambda^{\alpha'-1} \cdot e^{-\beta'\lambda}$

$$\alpha' = \sum_{i=n+1}^{m} x_i + 1$$

$$\beta' = m-n-1 + 1/\beta = m-n-1 + \frac{n}{\sum_{i=1}^{n} x_i}$$

Using sixth week data to calculate posterior 2

$$D_2 = \{ x_{m+1} \ldots x_l \}$$

$$f(\lambda \mid D_2) \propto \frac{e^{-(l-(m+1))\lambda} \cdot \lambda^{\sum_{m+1}^{l} x_i}}{\prod_{i=m+1}^{l} x_i!} \cdot e^{-\lambda(m-n-1+1/\beta)} \cdot \lambda^{\sum_{n+1}^{m} x_i}$$

$$\propto e^{-\lambda(l-m-1+m-n-1+1/\beta)} \cdot \lambda^{\sum_{n+1}^{l} x_i}$$

$$\propto e^{-\lambda(l-n-2+1/\beta)} \cdot \lambda^{\sum_{n+1}^{l} x_i}$$

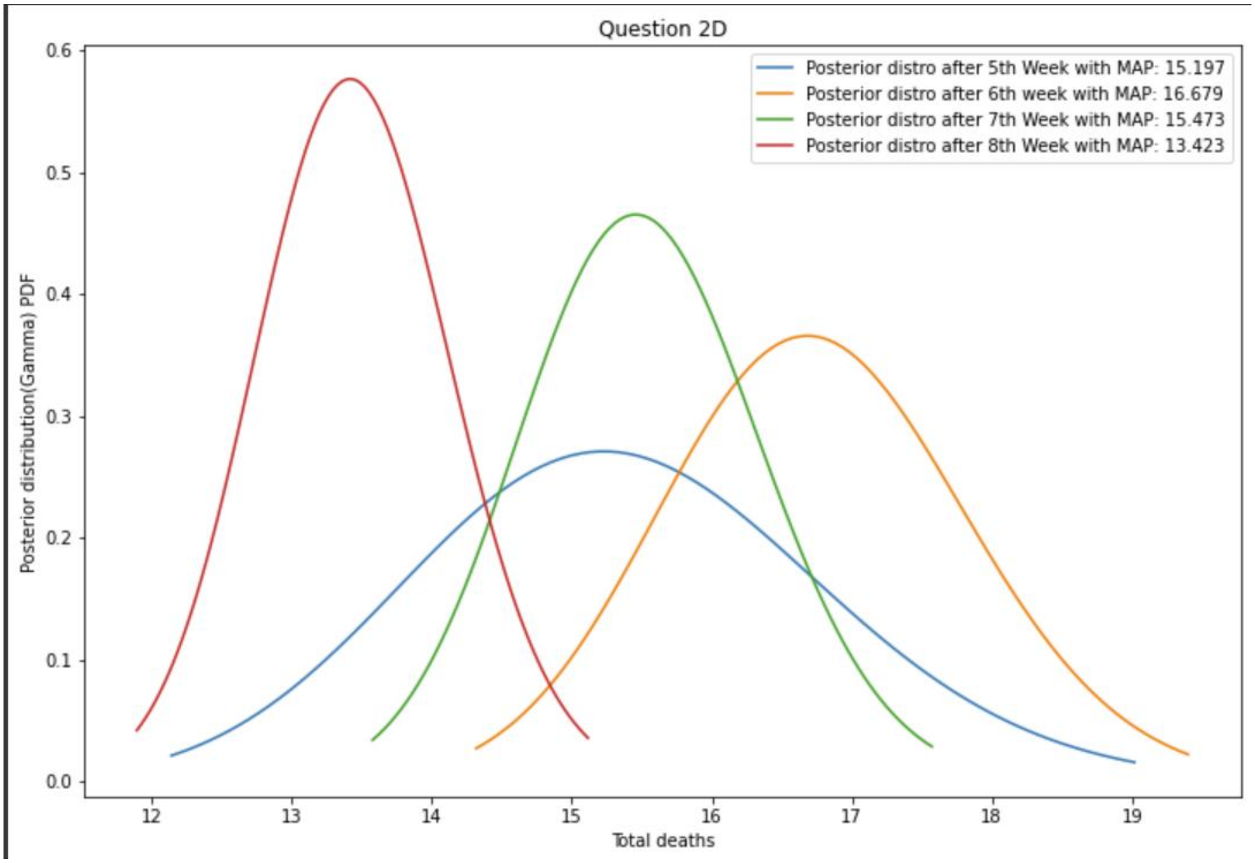
which is a conjugate prior

and hence posterior distro week 7 and 8 are conjugate priors.

Question 2D Code and Output:

```
import matplotlib.pyplot as plt
from scipy.stats import gamma

deaths_data = np.array(data["Total deaths"])
lambda_mme = np.sum(deaths_data[:28])/len(deaths_data[:28])
plt.figure(figsize=(12,8))

def plot_posterior_distributions(alpha, beta, label):
  #Reference : https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gamma.html
  x_values = np.linspace(gamma.ppf(0.01, alpha, scale=1/beta),
                         gamma.ppf(0.99, alpha, scale=1/beta), 100)
  y_values = gamma.pdf(x_values, alpha, scale=1/beta)
  plt.title("Question 2D")

  map_index = np.argmax(y_values)
  map_value = x_values[map_index]
  label= "Posterior distro " + label + " with MAP: " + str(round(map_value,3))

  plt.xlabel("Total deaths")
  plt.ylabel("Posterior distribution(Gamma) PDF")
  # print(map_index)
  plt.plot(x_values,y_values , label=label)
  plt.legend()

#First posterior distribution is a Gamma distribution
first_posterior_data = deaths_data[28:35]
#The first posterior distribution is a conjugate prior of the second posterior
# and hence taking all the data from fifth week
second_posterior_data = deaths_data[28:42]
#Similarly....
third_posterior_data = deaths_data[28:49]
fourth_posterior_data = deaths_data[28:56]
plot_posterior_distributions(np.sum(first_posterior_data) +1, len(first_posterior_data)+ (1/lambda_mme), "afte
plot_posterior_distributions(np.sum(second_posterior_data) +1, len(second_posterior_data)+ (1/lambda_mme), "af
plot_posterior_distributions(np.sum(third_posterior_data) +1, len(third_posterior_data)+ (1/lambda_mme), "afte
plot_posterior_distributions(np.sum(fourth_posterior_data) +1, len(fourth_posterior_data)+ (1/lambda_mme), "af
```

```
import numpy as np
# from numpy import genfromtxt
import pandas as pd
pd.options.mode.chained_assignment = None
import datetime

df = pd.read_csv('12_final_processed.csv', delimiter=',', parse_dates=True)

df["Date"] = pd.to_datetime(df["Date"])

def processDateStringToDate(date_str):
  return datetime.datetime.strptime(date_str, '%Y-%m-%d')

def processDataInDateRange(df, start_date_str, end_date_str):
  start_date = processDateStringToDate(start_date_str)
  end_date = processDateStringToDate(end_date_str)

  mask = (df['Date'] >= start_date) & (df['Date'] <= end_date)
  df_subdata = df.loc[mask]
  df_subdata["Total deaths"] = df_subdata["MN deaths"] + df_subdata["MI deaths"]
  df_subdata["Total confirmed"] = df_subdata["MN confirmed"] + df_subdata["MI confirmed"]
  df_subdata = df_subdata.reset_index()
  return df_subdata



start_date_str = '2020-06-01'
end_date_str = '2020-07-26'
data = processDataInDateRange(df, start_date_str, end_date_str)
data.to_csv('BI_2D_data.csv')
# print(data)
```



Question 2D
Posterior distro after 5th Week with MAP: 15.197
Posterior distro after 6th week with MAP: 16.679
Posterior distro after 7th Week with MAP: 15.473
Posterior distro after 8th Week with MAP: 13.423
Posterior distribution(Gamma) PDF
Total deaths
0.0
0.1
0.2
0.3
0.4
0.5
0.6
12
13
14
15
16
17
18
19

## Exploratory task 1:

Data used: https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=pet&s=wkjupus2&f=w

**Inference:**

The US weekly product supplied of kerosene-Type jet fuel has a significant dip from the month of January 2020 to August 2020.
We can infer that since covid-19 struck very bad and the cases were at peak in the US, the frequency in flights reduced or maybe when lockdown was imposed and flights were kept off for some amount of time, the demand for jet-Type fuel was reduced which indeed affected the supply for the same.

**The method:**

- We apply Pearson's correlation taking US_confirmed cases as X and Weekly U.S. Product Supplied of Kerosene-Type Jet Fuel (Thousand Barrels per Day) as Y from the month of January to August.

- Now since our US cases data was daily data and kerosene data was weekly data, we converted the daily data to weekly data.

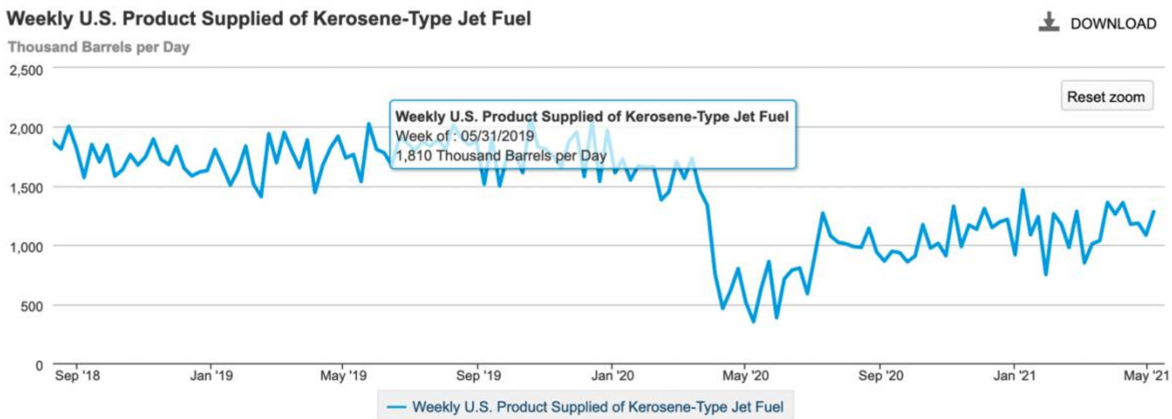
**The result:**

We find that the Correlation between all states and Weekly U.S. Product Supplied of Kerosene-Type Jet Fuel Thousand Barrels per Day for the period of march to july which turns out to be:
**-0.9394373097595258**

This means they are negatively related. That is when the cases were increasing the supply was decreased. As per our observations.

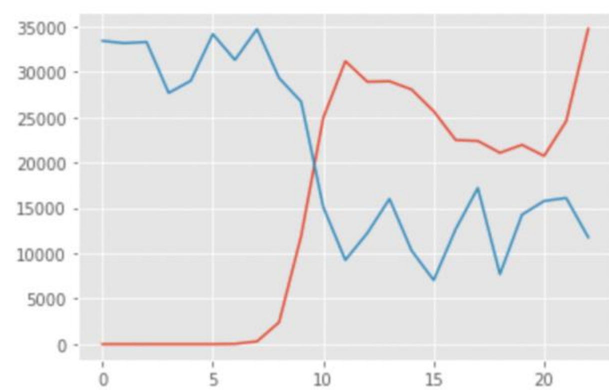
**Supporting evidence:**

While observing the data, we found a dip in Jet-Type fuel supply as follows:




For cases and Jet-Type fuel supply for Jan-Aug 2020

```
[<matplotlib.lines.Line2D at 0x7fb3589db040>]
```



## Exploratory Task 2

Dataset: “LA-2020-21.csv”

Processed from Link: https://www.epa.gov/outdoor-air-quality-data/air-quality-index-daily-values-report

**The inference:** The Ozone Air Quality Index (AQI) Value for the city of Los Angeles has impacted by the COVID peak between the duration of November, 2020 and March, 2021.

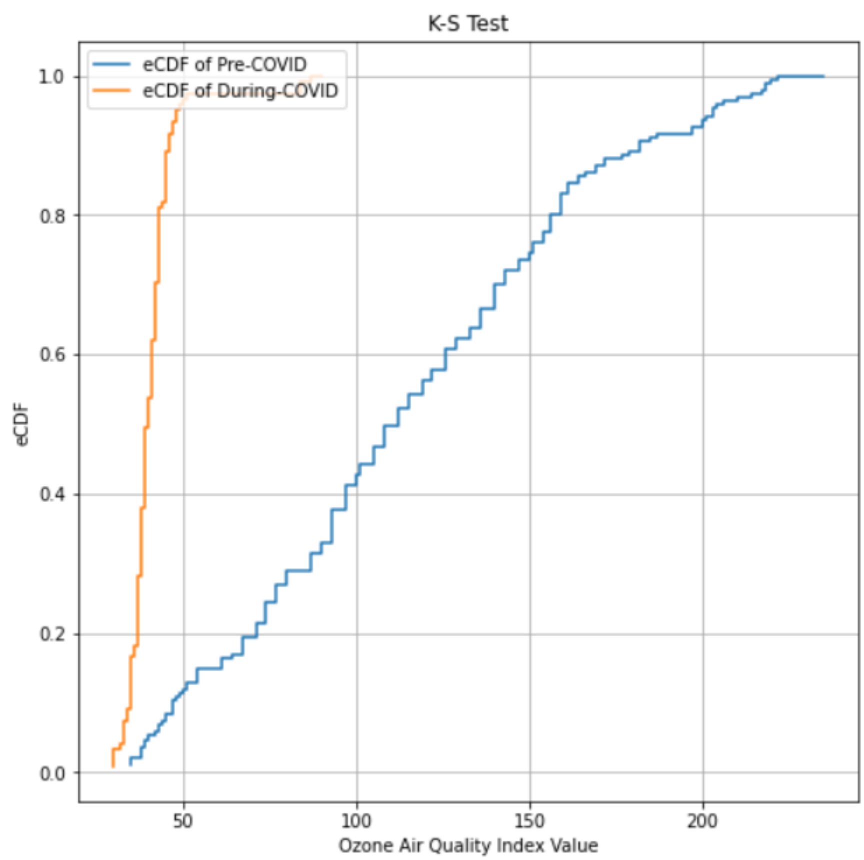
We use KS-Test to test whether the distribution of AQI Value for Los Angeles just before prime of COVID19 (April, 2020 to November, 2020) and during prime of COVID (November, 2020 to March, 2021) is different or same.

Note: We have only considered pre COVID period as certain months and not the entire historical data as the historical AQI values might be influenced by a lot of factors. We assume the AQI Value for a period of 12 months should have a similar distribution unless impacted by some factor (here, COVID19).

**The method:** We use KS 2 Sample test to check if the difference in the eCDF of the 2 distibution in question have significant difference.

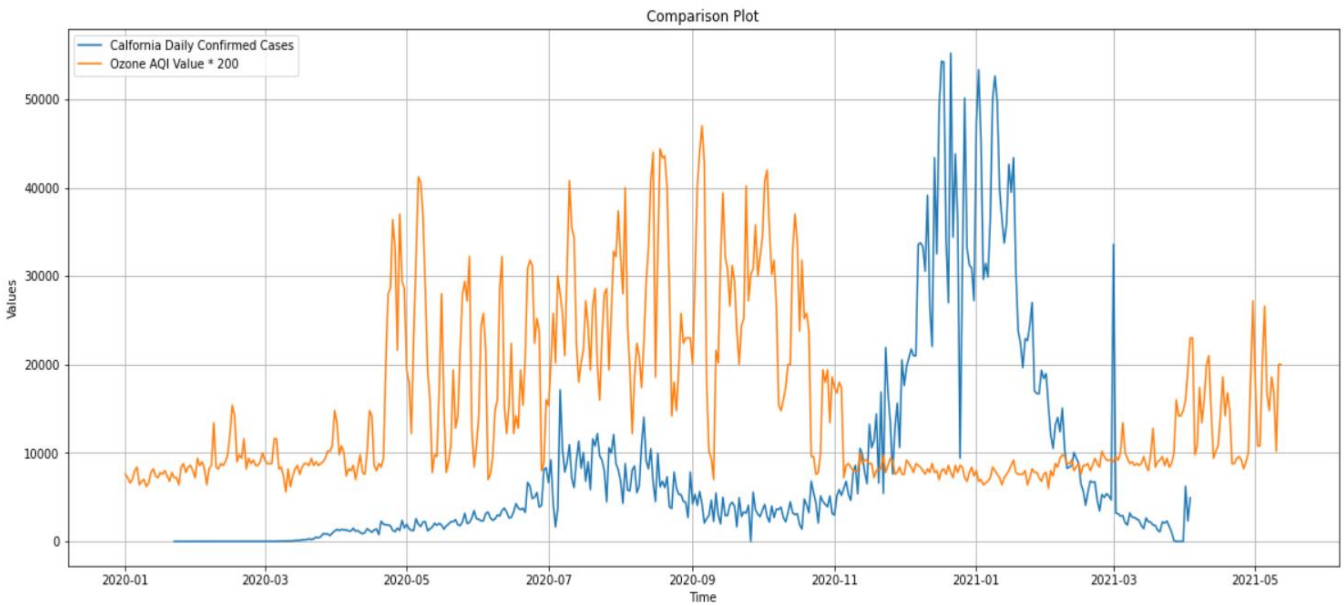
**Results:**

```
Since Max distance(0.847539) > 0.05, we reject Null Hypothesis (The Los Angeles
Ozone AQI Value has same distribution pre and during COVID)
```



**Supporting Evidence:**

The graph below is for time range of January, 2020 and May, 2021. We can visualize the data and infer that when the prime of COVID19 started in California the Ozone AQI Value of Los Angeles dropped. This can be because the people panicked and were suggested to stay indoors.



## Exploratory Part 3

**Inference:**
Type jet fuel has a significant dip from the month  of January 2020 to August 2020.
From inference 1, we understand that the supply of kerosene type jet fuel is correlated with covid. Now we want to check if that is the case for all types of fuel. So, we try to predict the the overall supply of petroleum products from the trend in supply of kerosene type fuel and covid cases.

**The method:**

We apply Multiple Linear Regression where x1 and x2 are covid cases in US and supply of kerosene type fuel. And the value to be predicted i.e. Y is overall supply of petroleum products.

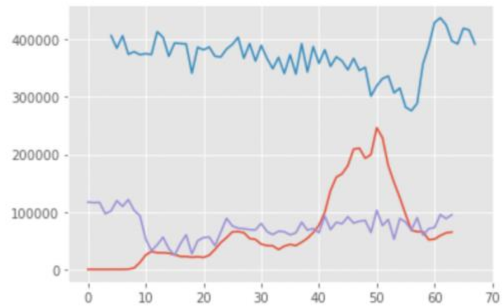
**The result:**

The MSE calculated on trained data is 0.0000421.
Thus, the prediction is pretty good and our inference seems correct.

**Supporting evidence:**

We plot the 3 data and can see the correlation.

```
In [370]: plt.plot(wdf['All states'])
          plt.plot(df2['Weekly U.S. Product Supplied of Petroleum Products Thousand Barrels per Day']*20)
          plt.plot(jet_df2['Weekly U.S. Product Supplied of Kerosene-Type Jet Fuel Thousand Barrels per Day']*70)

Out[370]: [<matplotlib.lines.Line2D at 0x7fb33c5de070>]
```




```
In [ ]:
```