In [3]:
```
import pandas as pd
import seaborn as sns
sns.set()
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
from scipy import stats
import sys
from numpy.linalg import inv
import math
from scipy.stats import gamma
```

In [4]:
```
#Reading COVID data from API to ensure we always have the latest availab
le data
# url = "https://covidtracking.com/api/v1/states/daily.csv"
full_data = pd.read_csv("daily.csv")
```

In [5]:
```
#We henceforth for the purpose of this project, will be working on the c
ombined COVID data of two states -
#Texas and New Mexico.
#We do realize these are two different states - we aim to combine the da
ta for these two and do some analysis and find meaningful inferences.
states = ['TX','NM']
texas_newmexico_data = full_data[full_data.state.isin(states)]
```

In [6]:
```
#Adding up rows for texas and new mexico
texas_newmexico_data = texas_newmexico_data.groupby(['date']).sum().rese
t_index()
```

In [7]:
```
#Data Cleaning - Filling null values with 0 as we assume no data was rec
orded/reported on those days.
#Why we impute with 0 and not with mean, median is better explained in t
he conclusion towards the end of this task.
texas_newmexico_data = texas_newmexico_data.fillna(0)
# texas_newmexico_data
```

Data Cleanup

Tukey's method of detecting outliers

In [8]:
```
# Tukey Method to detect outliers
n = 1 #In this case, we considered outliers as rows that have at least o
ne outlier numerical value.
indices = []
for col in texas_newmexico_data.columns[0:26]:
    Q1 = np.percentile(texas_newmexico_data[col],25) #Q1 is at 25 - firs
t quartile
    Q3 = np.percentile(texas_newmexico_data[col],75) #Q3 is at 75 - thir
d quartile
    IQR = Q3 - Q1 #inter-quartile range
    multiplier = 1.5 * IQR #This multiplier works for finding outliers f
or our data, so we stick to it.
    # Determine a list of indices of outliers for feature col
    list_outliers = texas_newmexico_data[(texas_newmexico_data[col] < Q1
- multiplier) | (texas_newmexico_data[col] > Q3 + multiplier )].index
    indices.extend(list_outliers) # appending the found outlier indices
 to the list of outlier indices
indices = Counter(indices)
outliers = list( k for k, v in indices.items() if v > n )
```

In [9]:
```
#These are the indices of the rows which had atleast one outlier value
outliers
```

Out[9]:
```
[55, 56, 51, 52]
```

In [10]:
```
#Throwing the outliers - we discard the ones we found
texas_newmexico_data.drop(outliers, axis = 0)
texas_newmexico_data = texas_newmexico_data.drop(outliers, axis = 0).res
et_index(drop=True)
```

In [11]:
```
# texas_newmexico_data.info()
texas_newmexico_data['state'] = 'TX,NM'
# texas_newmexico_data['date'].unique()
texas_newmexico_data.to_csv("tx_nm_data2.csv")
```

In [12]:
```
texas_newmexico_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53 entries, 0 to 52
Data columns (total 23 columns):
date                        53 non-null int64
positive                    53 non-null float64
negative                    53 non-null float64
pending                     53 non-null float64
hospitalizedCurrently       53 non-null float64
hospitalizedCumulative      53 non-null float64
inIcuCurrently              53 non-null float64
inIcuCumulative             53 non-null float64
onVentilatorCurrently       53 non-null float64
onVentilatorCumulative      53 non-null float64
recovered                   53 non-null float64
death                       53 non-null float64
hospitalized                53 non-null float64
total                       53 non-null float64
totalTestResults            53 non-null float64
posNeg                      53 non-null float64
fips                        53 non-null int64
deathIncrease               53 non-null float64
hospitalizedIncrease        53 non-null float64
negativeIncrease            53 non-null float64
positiveIncrease            53 non-null float64
totalTestResultsIncrease    53 non-null float64
state                       53 non-null object
dtypes: float64(20), int64(2), object(1)
memory usage: 9.6+ KB
```

Conclusion

We applied Tukey's rule and found around 4 outlier values, these are mostly the ones where there was a sudden spike in the positive or negative cases. In few cases, these were even when the data reported a relatively low (0) number of people getting hospitalized. Now, this could have been since we imputed the null values of our dataset as 0 - but since no data was available for these rows, we could not replace it with mean/median etc. - as that would tamper our original data. We assumed on these days no cases happened or for that matter were reported.

Since these values fell either below 1.5 range of first quartile or they were above 1.5 range of third quartile, we threw them and proceeded with the rest of the data for our further analysis.

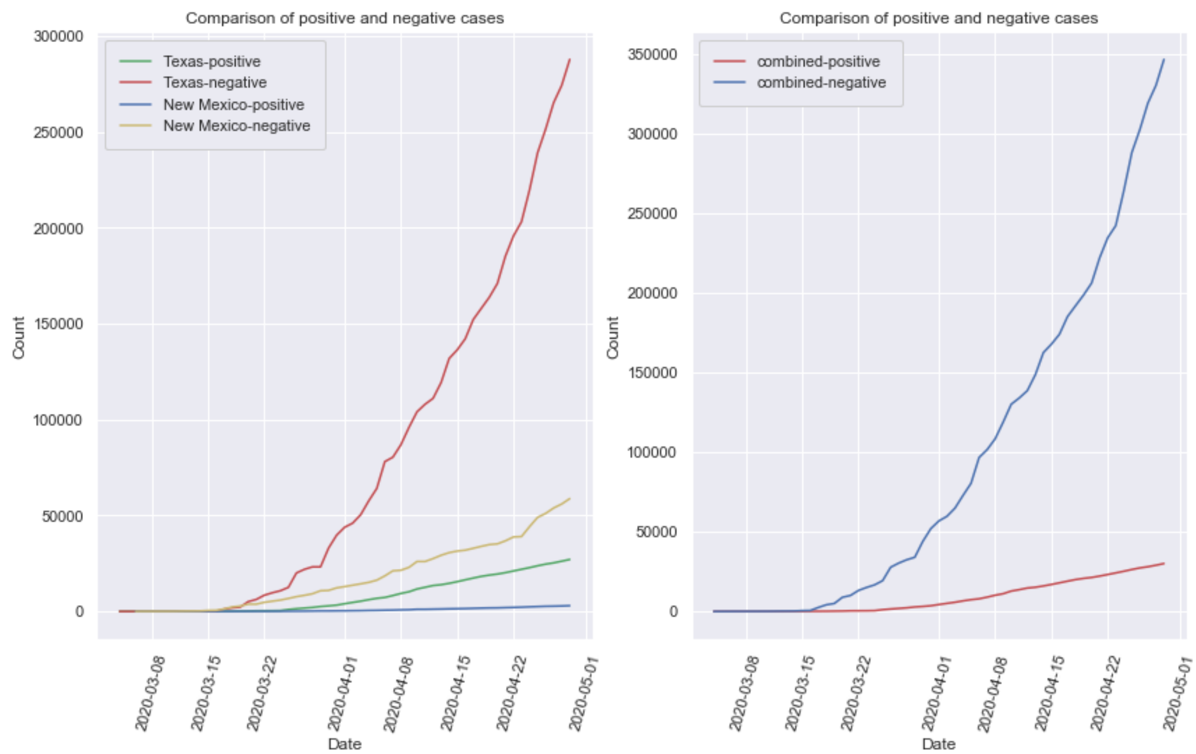
Visualizations

References:
[1] https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Texas (https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Texas)
[2]https://en.wikipedia.org/wiki/COVID-19_pandemic_in_New_Mexico (https://en.wikipedia.org/wiki/COVID-19_pandemic_in_New_Mexico)

In [13]:
```
#Original dataset with some preprocessing
states = ['TX','NM']
texas_newmexico_data = full_data[full_data.state.isin(states)]
texas_newmexico_data = texas_newmexico_data.fillna("0")
texas_newmexico_data = texas_newmexico_data.fillna(texas_newmexico_data.
mean())
texas_newmexico_data['death'] = texas_newmexico_data['death'].astype(int
)
texas_newmexico_data['date']=pd.to_datetime(texas_newmexico_data['date']
.astype(str), format='%Y%m%d')

#Separating the dataset based on the value of the two states-Texas and N
ew Mexico for comparison
data_NM=texas_newmexico_data.loc[texas_newmexico_data["state"] == "NM"]
data_TX=texas_newmexico_data.loc[texas_newmexico_data["state"] == "TX"]

#Dataset containing daily data considering Texas and New Mexico as a sin
gle region (and some preprocessing)
#Loading Covid data
comb_data=pd.read_csv("daily.csv")
states = ['TX','NM']
comb_data = full_data[full_data.state.isin(states)]
comb_data=comb_data.groupby(['date']).sum().reset_index()
comb_data = comb_data.fillna("0")
comb_data = comb_data.fillna(full_data.mean())
comb_data['death'] = comb_data['death'].astype(int)
comb_data.astype(int)
comb_data['date']=pd.to_datetime(comb_data['date'].astype(str), format=
'%Y%m%d')
```

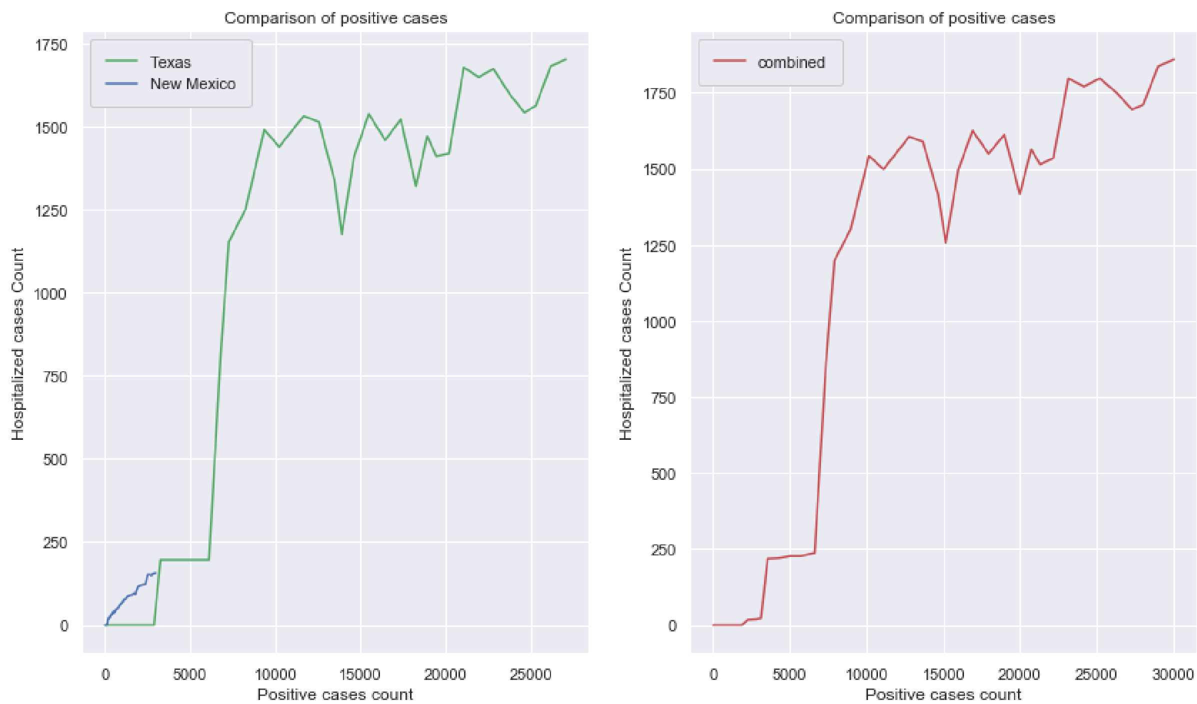
In [14]:
```
#Plot 1:Comparison of the number of postive and negative cases(daily) [S
tates separated and combined both]

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 8))

axes[0].plot(data_TX["date"],data_TX["positive"],'-g', label='Texas-posi
tive');
axes[0].plot(data_TX["date"],data_TX["negative"],'-r', label='Texas-nega
tive');
axes[0].plot(data_NM["date"],data_NM["positive"],'-b', label='New Mexico
-positive');
axes[0].plot(data_NM["date"],data_NM["negative"],'-y', label='New Mexico
-negative');

axes[0].set(title = "Comparison of positive and negative cases",xlabel =
"Date",ylabel = "Count");
axes[0].tick_params(axis='x', labelrotation=75)
axes[0].legend(fancybox=True, framealpha=1, borderpad=1)


axes[1].plot(comb_data["date"],comb_data["positive"],'-r', label='combin
ed-positive');
axes[1].plot(comb_data["date"],comb_data["negative"],'-b', label='combin
ed-negative');
axes[1].set(title = "Comparison of positive and negative cases",xlabel =
"Date",ylabel = "Count");
axes[1].legend(fancybox=True, framealpha=1, borderpad=1)
axes[1].tick_params(axis='x', labelrotation=75)
```

```
/Users/sakshigupta/anaconda3/lib/python3.7/site-packages/pandas/plottin
g/_converter.py:129: FutureWarning: Using an implicitly registered date
time converter for a matplotlib plotting method. The converter was regi
stered by pandas on import. Future versions of pandas will require you
to explicitly register matplotlib converters.

To register the converters:
        >>> from pandas.plotting import register_matplotlib_converters
        >>> register_matplotlib_converters()
  warnings.warn(msg, FutureWarning)
```



Idea:To compare the number of postive and negative cases(daily) [States separated and combined both]
As we can see from the above plot, the increase in negative cases has been really high for both the states, Also, the growth of positive cases is much more lesser than the growth of negative cases which summarizes for us that out of all tested individuals, the number of positive cases is quite less for both the states.
For an overview, we can refer to the olot on the right which clearly compares the growth rates of both negative and positive cases.

In [15]:
```
#Plot 2:Comparison of how many patients out of positive cases were hospi
talized [States separated and combined both]
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 8))

axes[0].plot(data_TX["positive"],data_TX["hospitalizedCurrently"],'-g',
label='Texas');
axes[0].plot(data_NM["positive"],data_NM["hospitalizedCurrently"],'-b',
label='New Mexico');
axes[0].legend(fancybox=True, framealpha=1, borderpad=1)
axes[0].set(title = "Comparison of positive cases",xlabel = "Positive ca
ses count",ylabel = "Hospitalized cases Count");

axes[1].plot(comb_data["positive"],comb_data["hospitalizedCurrently"],'-
r', label='combined');
axes[1].legend(fancybox=True, framealpha=1, borderpad=1)
axes[1].set(title = "Comparison of positive cases",xlabel = "Positive ca
ses count",ylabel = "Hospitalized cases Count");
```

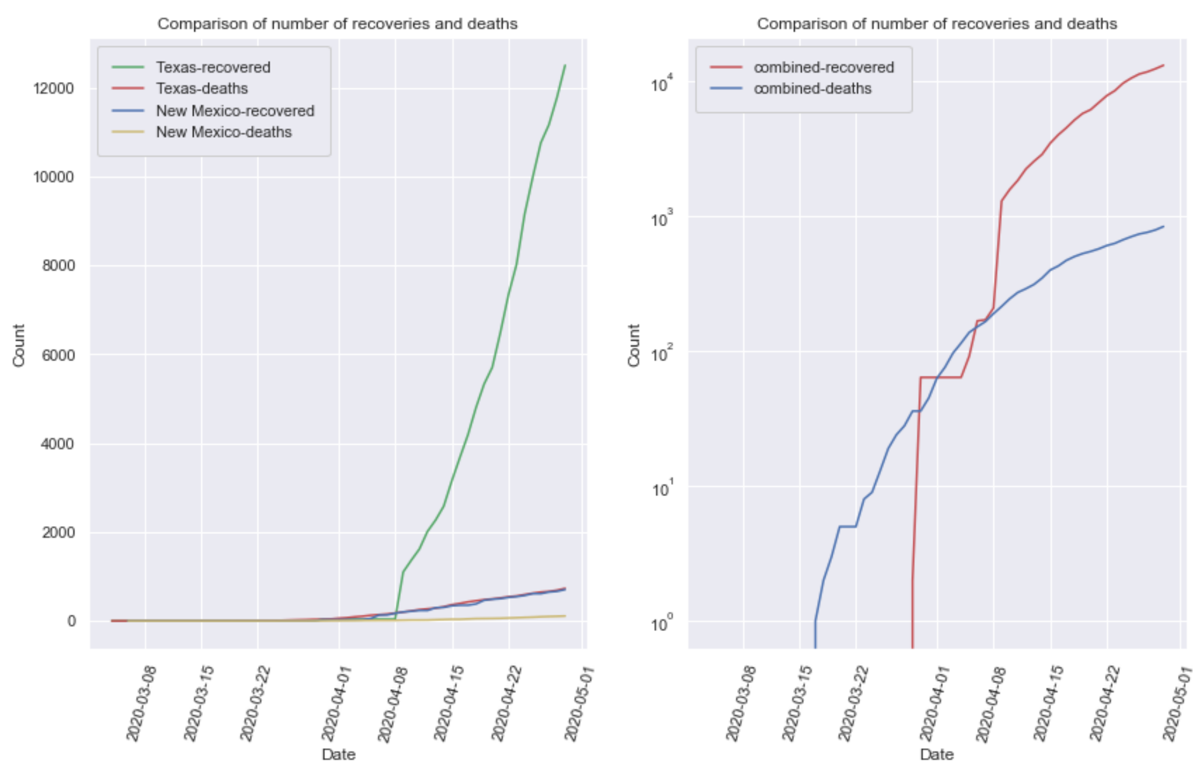


Idea: To compare how many patients out of positive cases were hospitalized [States separated and combined both]

From the curve of the above plot we can see that as the number of positive cases kept on increasing there was a simultaneous increase in patients who were hospitalized which also led to the high number of recoveries and low death rate for both the states.

The plot on the right summaries the combined information for us.

In [16]:
```
#Plot 3:Comparison of the number of recoveries and deaths[States separated and combined both]

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 8))
axes[1].set_yscale('log')
axes[0].plot(data_TX["date"],data_TX["recovered"],'-g', label='Texas-recovered');
axes[0].plot(data_TX["date"],data_TX["death"],'-r', label='Texas-deaths');

axes[0].tick_params(axis='x', labelrotation=75)

axes[0].plot(data_NM["date"],data_NM["recovered"],'-b', label='New Mexico-recovered');
axes[0].plot(data_NM["date"],data_NM["death"],'-y', label='New Mexico-deaths');
axes[0].set(title = "Comparison of number of recoveries and deaths",xlabel = "Date",ylabel = "Count");
axes[0].legend(fancybox=True, framealpha=1, borderpad=1)

axes[1].plot(comb_data["date"],comb_data["recovered"],'-r', label='combined-recovered');
axes[1].plot(comb_data["date"],comb_data["death"],'-b', label='combined-deaths');
axes[1].set(title = "Comparison of number of recoveries and deaths",xlabel = "Date",ylabel = "Count");
axes[1].legend(fancybox=True, framealpha=1, borderpad=1)
axes[1].tick_params(axis='x', labelrotation=75)
```

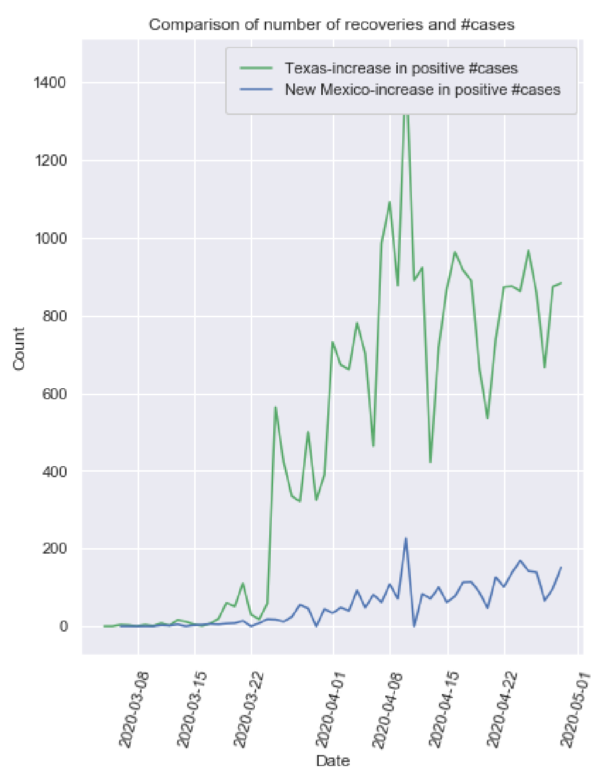


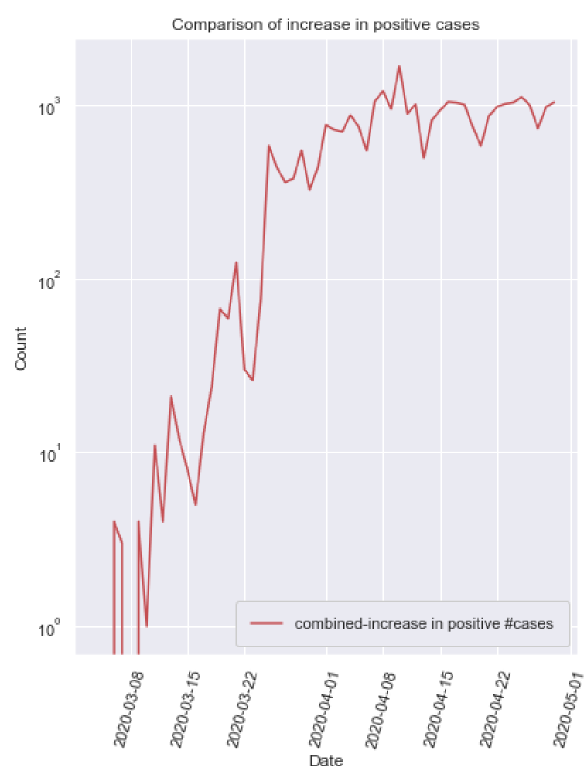
Idea:To compare the number of recoveries and deaths[States separated and combined both]

From the above graph we can see that the recovery rate for Texas has been really great while the death rate has had a really slow growth. For New Mexico, the recovery rate has been okay in comparison to the death rate which looks like a good sign.

The plot on the right is on a logarithmic scale and displays the combined stats for the combined region.

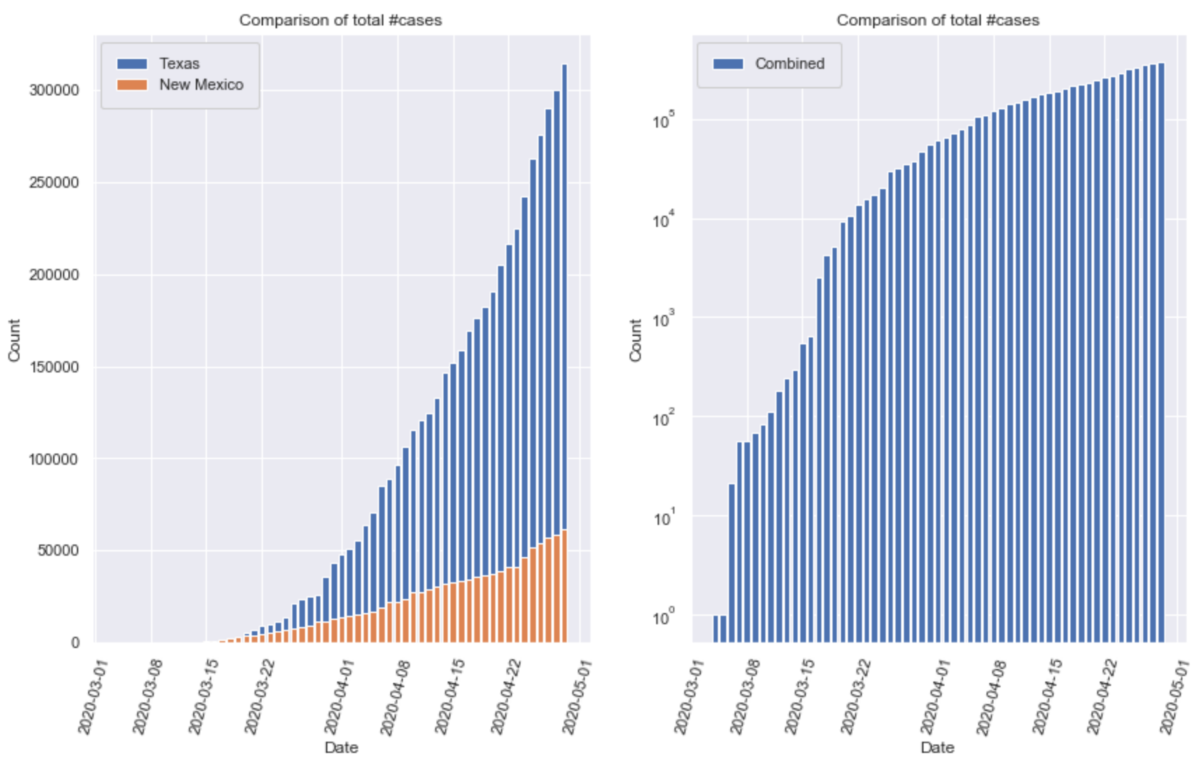
In [17]:
```
#Plot 4:Comparison of increase in number of positive cases[States separated and combined both]

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 8))

axes[0].plot(data_TX["date"],data_TX["positiveIncrease"],'-g', label='Texas-increase in positive #cases');
axes[0].tick_params(axis='x', labelrotation=75)

axes[0].plot(data_NM["date"],data_NM["positiveIncrease"],'-b', label='New Mexico-increase in positive #cases');
axes[0].set(title = "Comparison of number of recoveries and #cases",xlabel = "Date",ylabel = "Count");
axes[0].legend(fancybox=True, framealpha=1, borderpad=1)

axes[1].plot(comb_data["date"],comb_data["positiveIncrease"],'-r', label='combined-increase in positive #cases');
axes[1].set(title = "Comparison of increase in positive cases",xlabel = "Date",ylabel = "Count");
axes[1].legend(fancybox=True, framealpha=1, borderpad=1)
axes[1].tick_params(axis='x', labelrotation=75)
axes[1].set_yscale('log')
```





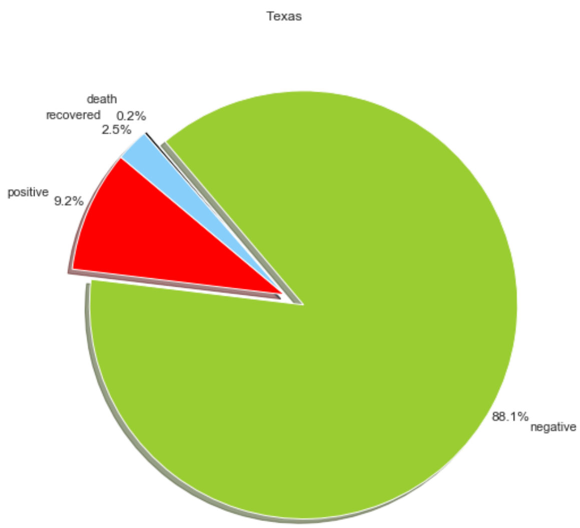
Idea:To compare increase in number of positive cases[States separated and combined both]
From the above line chart, we can clearly see that the count of positive cases was much more in Texas than in New Mexico and the number fluctuates quite a lot for both the states. We also noticed a sudden spike around April first week which relates to the news headline "Sixty residents and nine staff members at a San Antonio nursing home have been infected with COVID-19". Similarly, in the week around March 15 there is also a sudden spike relating to the news "Dallas County Judge Clay Jenkins announced five additional positive cases with one of the cases being the first instance of community spread in the North Texas area.". Similarly for New Mexico, we see a spike around April 4 as "51 new cases are reported with 23 in Bernalillo, 9 in San Juan, 6 in Santa Fe, 4 in Cibola, 3 in Torrance, 2 in Sandoval, and the first reported cases in Lincoln and Los Alamos counties and one new case in McKinley and Rio Arriba counties bringing the statewide total to 543."
Apart from that we can also see that the increase on the logarithmic scale keeps up with the general distribution of increase in covid positive numbers.

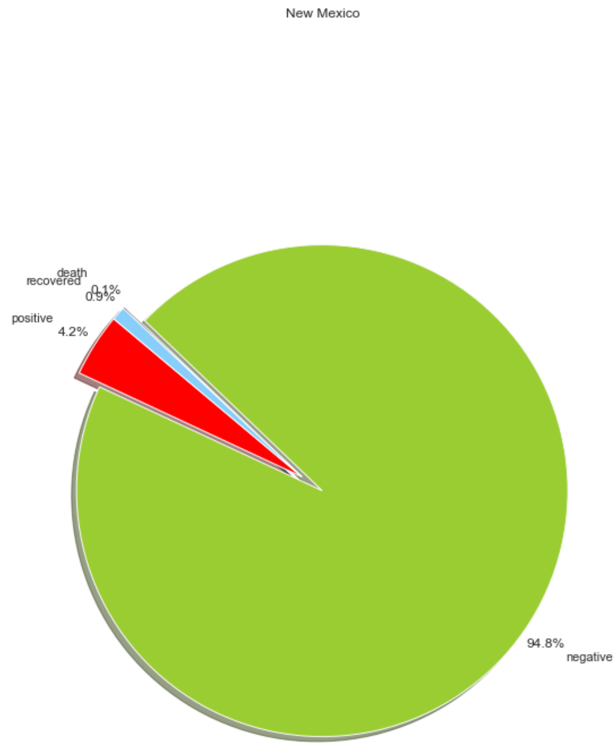
In [18]:
```
#Plot 5:Comparison of total number of cases[States separated and combine
d both]

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 8))

axes[0].bar(data_TX["date"],data_TX["total"],label='Texas');

axes[0].bar(data_NM["date"],data_NM["total"],label='New Mexico');
axes[0].tick_params(axis='x', labelrotation=75)
axes[0].set(title = "Comparison of total #cases",xlabel = "Date",ylabel
= "Count");
axes[0].legend(fancybox=True, framealpha=1, borderpad=1)

axes[1].bar(comb_data["date"],comb_data["total"], label='Combined');
axes[1].set(title = "Comparison of total #cases",xlabel = "Date",ylabel
= "Count");
axes[1].legend(fancybox=True, framealpha=1, borderpad=1)
axes[1].tick_params(axis='x', labelrotation=75)
axes[1].set_yscale('log')
```



Idea: To compare the total number of cases of the two states[States separated and combined both]
A histogram seemed like a good idea here to see the growth rate and as we can see the number of cases increased exponentially day by day. The plot on the right is on a logarithmic scale which confirms the belief that the growth in the number of corona virus cases increases exponentially.

In [19]:
```
#Plot 6:Comparison of final stats till the latest data available for Tex
as and New Mexico
data_TX_red=data_TX[['positive','negative','death','recovered']]
data_TX_red=data_TX_red.astype(int)
TX_sum=data_TX_red.sum(axis=0)

data_NM_red=data_NM[['positive','negative','death','recovered']]
data_NM_red=data_NM_red.astype(int)
NM_sum=data_NM_red.sum(axis=0)

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(15, 12))

labels = 'positive','negative','death','recovered'
colors = ['red', 'yellowgreen', 'black', 'lightskyblue']
explode = (0, 0.1, 0, 0)
axes[0].pie(TX_sum,labels=labels, colors=colors,explode = explode,autopc
t='%1.1f%%',shadow=True,startangle=140,pctdistance=1.1, labeldistance=1.
2)
axes[0].set(title='Texas')
axes[1].pie(NM_sum,labels=labels, colors=colors,explode = explode,autopc
t='%1.1f%%',shadow=True,startangle=140,pctdistance=1.1, labeldistance=1.
2)
axes[1].set(title='New Mexico')

#plt.title('Comparison of final stats till the latest data available for
Texas and New Mexico',loc='left')
plt.axis('equal')
plt.tight_layout()
plt.show()
```





Idea: To compare the final statistics of the two states till the latest data available.
A pie chart makes it easy to compare and we can see from the above plot that a very large percentage of tests carried out were negative for both the states and out of the percentage of positive cases the death percentage is extremely small which is a good thing.

Required Inferences

In [20]:
```
#Required cleanup for this inference
df = pd.read_csv('daily.csv')
#Read data from texas and new mexico
df = df[df["state"].isin(["TX", "NM"])]
df['date'] = pd.to_datetime(df['date'], format='%Y%m%d')
df = df.sort_values('date')
df.fillna(0, inplace=True)

#Extract the data for the two relevant weeks
week_one = df.iloc[len(df) - 14:len(df) - 7]
week_n = df.iloc[len(df) - 7:, :]
```

Required Inference 1

Time Series

In [21]:
```
# Functions to compute timeseries p: p value for AR(p); data: Data; num:
the number of days to predict
def AR(p, data, num =7):
    #Create Training data for the beta values
    y = []
    x = []
    #All but the last few days determined by the variable num
    for i in range(p, len(data)-num):
        y.append(data[i])
        temp = []
        for j in range(1, p + 1):
            temp.append(data[i-j])
        x.append(temp)

    real_values = data[len(data)-num:]
    x = np.array(x)
    y_orig = y
    y = np.array(y)
    #X is of shape(n, p)
    X = x.reshape(len(x), p)
    X = np.c_[ X, np.ones(len(X)) ]
    #Y is of shape (n, 1)
    Y = y.reshape(len(y), 1)
    #Computing the beta values. Similar to the linear regression.
    b = inv(X.T.dot(X)).dot(X.T).dot(Y) # The beta values
    #Predicting the next num values
    ans = []
    #Predict the values for num number of days
    for i in range(0, num):
        pred = 0
        for j in range(1, p+1):
            pred = pred + y_orig[len(y_orig)-j] * b[j-1]
        pred = pred + b[p]
        ans.append(pred[0])
        y_orig.append(pred)

    #Printing the real and the computed values. Can comment out
    print("Real Data", real_values)
    print("Predicted Values", ans)
    #Computing MAPE
    sums = 0
    for i in range(0, len(ans)):
        diff = (abs(real_values[i]-ans[i])/real_values[i]) * 100
        sums = sums + diff
    sums = sums/len(real_values)
    #Computing MSE
    MSE = np.sum(np.square(real_values - ans))/len(real_values)
    #Returning MAPE% and MSE
    return {"MAPE %": sums, "MSE": MSE}

#Function to predict data using EWMA; Arguments self explanatory
def EWMA(data, alpha, num=7):
    #Initialize and find the last predicted value
    y_last_predicted = data[0]
    for i in range(1, len(data)-num):
        y_last_predicted = y_last_predicted * (1 - alpha) + data[i] * al
```

```
pha
    ans = []
    #Use the num variable to get the real values for the final computati
on
    real_values = data[len(data)-num: ]
    #Compute the values
    for i in range(len(data)-num, len(data)):
        x = data[i] * alpha + y_last_predicted * (1 - alpha)
        ans.append(x)
    #Print the data
    print("Real Data", real_values)
    print("Predicted Values", ans)
    sums = 0
    #Calculate MAPE
    for i in range(0, len(ans)):
        diff = (abs(real_values[i]-ans[i])/real_values[i]) * 100
        sums = sums + diff
    sums = sums/len(real_values)
    #Calculate MSE
    MSE = np.sum(np.square(real_values - ans))/len(real_values)
    return {"MAPE %": sums, "MSE": MSE}
```

In [22]:

```
deaths = df["death"].to_numpy()
print(AR(3, deaths))
print(AR(5, deaths))
print(EWMA(deaths, 0.5))
print(EWMA(deaths, 0.8))
```

```
Real Data [ 93. 663.  99. 690. 104. 732. 110.]
Predicted Values [187.19813837614188, 472.25771404707444, 304.281131206
79166, 364.1020729601822, 341.70661919614975, 324.6912738917074, 334.90
966657535336]
{'MAPE %': 124.75925953891631, 'MSE': 66656.41484216126}
Real Data [ 93. 663.  99. 690. 104. 732. 110.]
Predicted Values [170.4130449195075, 540.3997417586421, 267.31936764654
37, 511.66522432930594, 323.1482932146489, 474.84111180918046, 375.5301
7258410955]
{'MAPE %': 112.11976197992259, 'MSE': 37974.463618569374}
Real Data [ 93. 663.  99. 690. 104. 732. 110.]
Predicted Values [275.16120950963773, 560.1612095096377, 278.1612095096
3773, 573.6612095096377, 280.66120950963773, 594.6612095096377, 283.661
20950963773]
{'MAPE %': 107.95964504695168, 'MSE': 24231.61689319507}
Real Data [ 93. 663.  99. 690. 104. 732. 110.]
Predicted Values [185.0366675195392, 641.0366675195392, 189.83666751953
922, 662.6366675195392, 193.83666751953922, 696.2366675195392, 198.6366
6751953923]
{'MAPE %': 52.83466990266051, 'MSE': 5022.755697476369}
```

Required Inference 2

Walds, Z and T test

In [23]:
```
#Sample mean calculation
def calc_sample_mean(data):
    n = len(data)
    sum = 0
    for i in data:
        sum += i
    return sum / n
#As described in the class
def calc_normal_mle(data):
    u_hat = calc_sample_mean(data)
    sigma_hat = 0
    for i in data:
        sigma_hat += (i - u_hat) * (i - u_hat)
    sigma_hat = sigma_hat / len(data)
    return round(u_hat, 3), round(sigma_hat, 3)

#One tailed test
def walds_test(X, u, H0):
    #MLE
    mu_x_hat, var_x_hat = calc_normal_mle(X)
    delta_hat = mu_x_hat - u
    se_hat = np.sqrt(var_x_hat/len(X))
    #Calculating W
    W = np.absolute(delta_hat/se_hat)
    if W > 1.96:
        print("walds_test: Rejecting the null hypothesis: ", H0, " With
 value W=",W)
    else:
        print("walds_test: Accepting the null hypothesis: ", H0, " With
 value W=",W)
#Two tailed test
def walds_two_tail_test(X, Y, H0):
    #Calculating Mean
    mu_x_hat = calc_sample_mean(X)
    mu_y_hat = calc_sample_mean(Y)
    delta_hat = mu_x_hat - mu_y_hat
    var_x = np.sum(np.square(X-mu_x_hat))/len(X)
    var_y = np.sum(np.square(Y-mu_y_hat))/len(Y)
    #SE hat
    se_hat = np.sqrt((var_x + var_y)/len(X))
    #Calculate W
    W = np.absolute(delta_hat/se_hat)
    if W > 1.96:
        print("walds_two_tail_test: Rejecting the null hypothesis: ", H0
, " With value W=",W)
    else:
        print("walds_two_tail_test: Accepting the null hypothesis: ", H0
, " With value W=",W)
#Z Test
def z_test(X, sigma, u, H0):
    x_bar = np.mean(X)
    n = len(X)
    #Calculate Z
    Z = (x_bar - u)/(sigma/np.sqrt(n))
    if Z > 1.96:
        print("z_test: Rejecting the null hypothesis: ", H0, " With valu
```

```
e Z=",Z)
    else:
        print("z_test: Accepting the null hypothesis: ", H0, " With valu
e Z=",Z)

def t_test(X, u, H0, alpha = 0.05):
    x_bar = np.mean(X)
    n = len(X) - 1
    sigma = np.std(X)
    #Calculate T
    T = np.abs((x_bar - u)/(sigma/np.sqrt(n)))
    val = stats.t.ppf(1-alpha, n-1)
    if T > val:
        print("t_test: Rejecting the null hypothesis: ", H0, " With valu
e T=",T)
    else:
        print("t_test: Accepting the null hypothesis: ", H0, " With valu
e T=",T)
#Two Sample tests
def t_test_two_tailed_paired(X, Y, H0):
    D = X-Y
    n = len(D)
    d_bar = np.mean(D)
    sigma = np.std(D)
    se = sigma/(np.sqrt(n))
    T = np.abs(d_bar/se)
    alpha = 0.05/2
    val = stats.t.ppf(1-alpha, n-1)
    if T > val:
        print("t_test_two_tailed_paired: Rejecting the null hypothesis:
 ", H0, " With value T=",T)
    else:
        print("t_test_two_tailed_paired: Accepting the null hypothesis:
 ", H0, " With value T=",T)

#Two sample unpaired test
def t_test_two_tailed_unpaired(X, Y, H0):
    D = X-Y
    n = len(D)
    s_x = np.std(X)
    s_y = np.std(Y)
    #Calculate pooled std
    pooled_std = np.sqrt((np.square(s_x) / n) + (np.square(s_y) / n))
    d_bar = np.mean(D)
    #Calculate T
    T = np.abs(d_bar/pooled_std)
    alpha = 0.05/2
    val = stats.t.ppf(1-alpha, n-1)
    if T > val:
        print("t_test_two_tailed_unpaired: Rejecting the null hypothesi
s: ", H0, " With value T=",T)
    else:
        print("t_test_two_tailed_unpaired: Accepting the null hypothesi
s: ", H0, " With value T=",T)
```

In [24]:
```
#@Warning the column values are imputed with 0 inplace
def inference_two(column, H0):
    df[column].fillna(0, inplace=True)
    mu = week_one[column].mean()
    c_array_wn = week_n[column].to_numpy()
    c_array_w1 = week_one[column].to_numpy()
    sigma = df[column].std()
    walds_test(c_array_wn, mu, H0)
    walds_two_tail_test(c_array_w1, c_array_wn, H0)
    z_test(c_array_wn, sigma, mu, H0)
    t_test(c_array_wn, mu, H0)
    t_test_two_tailed_paired(c_array_w1, c_array_wn, H0)
    t_test_two_tailed_unpaired(c_array_w1, c_array_wn, H0)
```

In [25]:
```
#On the Death column
inference_two("death", "Mean of Covid 19 deaths in second last and last
 week of dataset is the same")
```

```
walds_test: Accepting the null hypothesis:  Mean of Covid 19 deaths in
second last and last week of dataset is the same  With value W= 0.21445
329512670405
walds_two_tail_test: Accepting the null hypothesis:  Mean of Covid 19 d
eaths in second last and last week of dataset is the same  With value W
= 0.15997737074729546
z_test: Accepting the null hypothesis:  Mean of Covid 19 deaths in seco
nd last and last week of dataset is the same  With value Z= -0.33500965
643443403
t_test: Accepting the null hypothesis:  Mean of Covid 19 deaths in seco
nd last and last week of dataset is the same  With value T= 0.198543981
85384695
t_test_two_tailed_paired: Accepting the null hypothesis:  Mean of Covid
19 deaths in second last and last week of dataset is the same  With val
ue T= 0.13185244332785478
t_test_two_tailed_unpaired: Accepting the null hypothesis:  Mean of Cov
id 19 deaths in second last and last week of dataset is the same  With
value T= 0.15997737074729523
```

In [26]:
```
#Running on the total column
inference_two("total", "Total Covid 19 cases in second last and last wee
k of dataset is the same")
```

```
walds_test: Accepting the null hypothesis:  Total Covid 19 cases in sec
ond last and last week of dataset is the same  With value W= 0.02866186
6518321706
walds_two_tail_test: Accepting the null hypothesis:  Total Covid 19 cas
es in second last and last week of dataset is the same  With value W=
0.021860515114876786
z_test: Accepting the null hypothesis:  Total Covid 19 cases in second
last and last week of dataset is the same  With value Z= -0.04509816638
243091
t_test: Accepting the null hypothesis:  Total Covid 19 cases in second
last and last week of dataset is the same  With value T= 0.026535729228
428343
t_test_two_tailed_paired: Accepting the null hypothesis:  Total Covid 1
9 cases in second last and last week of dataset is the same  With value
T= 0.017990054069822224
t_test_two_tailed_unpaired: Accepting the null hypothesis:  Total Covid
19 cases in second last and last week of dataset is the same  With valu
e T= 0.021860515114876578
```

Conclusion

Applicability

Below we check the applicability of the tests.

Walds test

Since the walds test requires the estimator to be asymptotically normal this test is not applicable in this current data.

Z-Test

In Z test we know that either the number of datapoints needs to be very high or the points should follow a normal distribution. In our case none of the above is true hence Z test is not applicable as well.

T-Test

T-test is usually used when we have small number of datapoints. But the test still requires the datapoints to be in normal distribution. Hence if we can't prove the values are normally distributed this test won't be applicable as well.

Now Since the number of deaths per day or the number of total cases are not normally distributed we can say that the above tests are not applicable in the current state.

Required Inference 3

KS Test and P-Test

In [27]:
```
#Helper functions
#Calculating the MME based on class derivations.
def calculate_poission_mme(data):
    return  round(calc_sample_mean(data), 3)

def calc_geometric_mme(data):
    return round(1 / calc_sample_mean(data), 3)

def calc_square_sum(data):
    square_sum = 0
    for i in data:
        square_sum += i * i
    return square_sum

def calc_second_moment(data):
    return calc_square_sum(data) / len(data)

def calc_binomeal_mme(data):
    u_hat = calc_sample_mean(data)
    sub = 0
    for d in data:
        sub = sub + pow(d - u_hat, 2)
    sub = sub/np.sum(data)
    p = 1 - sub
    n = u_hat/p
    return n, p
```

In [28]:

```
#Ks Two Sample test
def ks_test_two_sample(X, Y, c, H0):
    if len(Y) < len(X):
        ks_test_two_sample(Y, X)
        return
    #X is always the smaller set or equal in this particular case.
    counter_x = Counter(X)
    counter_y = Counter(Y)
    #Get number of change points
    change_points = len(Y)
    #Calculate the increments
    inc = (1/change_points)
    sorted_y = sorted(Y)
    prev = 0
    max_diff = -sys.maxsize - 1
    #Run through all the unique values in x
    for val in sorted(counter_x.keys()):
        ch = 0
        #Find all the values smaller than val in Y and add to ch
        for v in sorted(counter_y.keys()):
            if v > val:
                break
            ch = ch + inc * counter_y[v]
        v1 = ch
        v2 = prev
        v3 = prev + inc * counter_x[val]
        prev = v3
        #Check and update max difference
        max_diffn = max(max_diff, max(abs(v1-v2), abs(v1-v3)))
        if max_diffn != max_diff:
            max_diff = max_diffn
    if max_diff > c:
         print("KS_test: Rejecting the null hypothesis: ", H0, " With va
lue D=",max_diff)
    else:
        print("KS_test: Accepting the null hypothesis: ", H0, " With val
ue D=",max_diff)

#The cdf data contains the values for the proposed distribution. (ex: bi
nomial)
def ks_test(data, cdf_data, c, H0):
    change_points = len(data)
    dictionary = Counter(data)
    unique_values = sorted(dictionary.keys())
    F_Y_x = []
    F_hat_X_x_mi = []
    F_hat_X_x_ma = []
    prev = 0
    #Calculate the increments
    inc = (1/change_points)
    max_diff = -sys.maxsize - 1
    for val in unique_values:
        v1 = cdf_data[val]#distribution.cdf(val, lambda_hat)
        v2 = prev
        #The number of times this value is seen is the dataset times the
increment
```

```
            prev = prev + inc * dictionary[val]
            v3 = prev
            F_Y_x.append(v1)
            F_hat_X_x_mi.append(v2)
            F_hat_X_x_ma.append(v3)
            #Check and update the max difference
            max_diff = max(max_diff, max(abs(v1-v2), abs(v1-v3)))
    if max_diff > c:
             print("KS_test: Rejecting the null hypothesis: ", H0, " With va
lue D=",max_diff)
    else:
            print("KS_test: Accepting the null hypothesis: ", H0, " With val
ue D=",max_diff)

def permutation_test(X, Y, p_count, p_threshold):
    x_bar = np.mean(X)
    y_bar = np.mean(Y)
    #Observed statistic
    t_obs = abs(x_bar - y_bar)
    C = np.concatenate((X, Y))
    count = 0
    #Run the test p_count times
    for i in range(p_count):
        p = np.random.permutation(C)
        mid = int(len(C) / 2)
        t_i = abs(np.mean(p[:mid]) - np.mean(p[mid:]))
        if t_i > t_obs:
            count += 1
    #Calculating the p value
    p_value = count / p_count
    if p_value <= p_threshold:
        print("P-Test: Rejecting Null Hypothesis, X and Y don't come fro
m same distribution with p value", p_value)
    else:
        print("P-Test: Accepting Null Hypothesis, X and Y come from same
distribution with p value", p_value)
```

In [29]:
```
###Deaths
#Poisson test 1 sample ks
lambda_hat = calculate_poission_mme(week_one["death"].to_numpy())
death2 = week_n["death"].to_numpy()
val_to_cdf_dic = {}
for d in death2:
    val_to_cdf_dic[d] = stats.poisson.cdf(d, lambda_hat)

ks_test(death2, val_to_cdf_dic, 0.05, "Deaths belong to poisson distribu
tion")

#Geometric Distribution
geometric_mean = calc_geometric_mme(week_one["death"].to_numpy())
val_to_cdf_dic = {}
for d in death2:
    val_to_cdf_dic[d] = stats.geom.cdf(d, geometric_mean)
ks_test(death2, val_to_cdf_dic, 0.05, "Deaths belong to Geometric distri
bution")

#Binomial Distribution
n, p = calc_binomeal_mme(week_one["death"].to_numpy())
val_to_cdf_dic = {}
for d in death2:
    val_to_cdf_dic[d] = stats.binom.cdf(d, n, p)
ks_test(death2, val_to_cdf_dic, 0.05, "Deaths belong to Binomial distrib
ution")

#KS Two sample test
ks_test_two_sample(week_n["death"].to_numpy(), week_one["death"].to_nump
y(), 0.05, "Second last and last week are from same distribution")

#P-Test
X, Y = week_one["death"], week_n["death"]
permutation_test(X.to_numpy(), Y.to_numpy(), 50000, 0.05)
```

```
KS_test: Rejecting the null hypothesis:  Deaths belong to poisson distr
ibution  With value D= 0.5714285714285714
KS_test: Rejecting the null hypothesis:  Deaths belong to Geometric dis
tribution  With value D= 0.2921476856603765
KS_test: Rejecting the null hypothesis:  Deaths belong to Binomial dist
ribution  With value D= 1.0
KS_test: Rejecting the null hypothesis:  Second last and last week are
from same distribution  With value D= 0.42857142857142855
P-Test: Accepting Null Hypothesis, X and Y come from same distribution
with p value 0.99946
```

In [30]:
```
#Total
lambda_hat = calculate_poission_mme(week_one["total"].to_numpy())
totals = week_n["total"].to_numpy()
val_to_cdf_dic = {}
for d in totals:
    val_to_cdf_dic[d] = stats.poisson.cdf(d, lambda_hat)

ks_test(totals, val_to_cdf_dic, 0.05, "total belong to poisson distribut
ion")

#Geometric Distribution
geometric_mean = calc_geometric_mme(week_one["total"].to_numpy())
val_to_cdf_dic = {}
for d in totals:
    val_to_cdf_dic[d] = stats.geom.cdf(d, geometric_mean)
ks_test(totals, val_to_cdf_dic, 0.05, "total belong to Geometric distrib
ution")

#Binomial Distribution
n, p = calc_binomeal_mme(week_one["total"].to_numpy())
val_to_cdf_dic = {}
for d in totals:
    val_to_cdf_dic[d] = stats.binom.cdf(d, n, p)
ks_test(totals, val_to_cdf_dic, 0.05, "total belong to Binomial distribu
tion")

#KS Two sample test
ks_test_two_sample(week_n["total"].to_numpy(), week_one["total"].to_nump
y(), 0.05, "Second last and last week are from same distribution")

#P-Test
X, Y = week_one["total"], week_n["total"]
permutation_test(X.to_numpy(), Y.to_numpy(), 50000, 0.05)
```

```
KS_test: Rejecting the null hypothesis:  total belong to poisson distri
bution  With value D= 0.5714285714285714
KS_test: Rejecting the null hypothesis:  total belong to Geometric dist
ribution  With value D= 0.9999999999999998
KS_test: Rejecting the null hypothesis:  total belong to Binomial distr
ibution  With value D= 1.0
KS_test: Rejecting the null hypothesis:  Second last and last week are
from same distribution  With value D= 0.42857142857142855
P-Test: Accepting Null Hypothesis, X and Y come from same distribution
with p value 0.99922
```

Required Inference 4

Correlation

Note:

We are using the financial stock data of TripAdvisor as our X dataset(which is an online travel site for booking tickets to travel around the world.) We sense our covid dataset and this dataset could be related with people travelling all around the globe - they might potentially be carrying and spreading the virus.

In this task, let us calculate the pearson correlation to find out if some specific columns between these two are correlated or not.

In [31]:

```
tx_nm_data =  pd.read_csv("tx_nm_data2.csv")
#Reading in data that has dates between march april for getting 1-1.5 mo
nth data.
tx_nm_data = tx_nm_data[tx_nm_data['date'].astype(str).str.contains("202
003|202004", na=False)]
# tx_nm_data['date'].unique()
#Reading in X dataset - stock data of TripAdvisor
trip_stock_data = pd.read_csv("TRIP.csv")
trip_stock_data.rename(columns={'Date':'date'}, inplace=True)
#We need to format time of datasets to match with the date time format o
f all.
df = pd.DataFrame({'year': tx_nm_data.date.astype(str).str.slice(0,4),
                   'month': tx_nm_data.date.astype(str).str.slice(4,6),
                   'day': tx_nm_data.date.astype(str).str.slice(6,8)})
tx_nm_data['date'] = pd.Series(pd.to_datetime(df))
tx_nm_data['date'] = pd.to_datetime(tx_nm_data['date'], utc = True)
# nm_data['date'] = pd.to_datetime(nm_data['date'], utc = True)
trip_stock_data['date'] = pd.to_datetime(trip_stock_data['date'], utc =
True)
#Merging data on date as common column
tx_nm_stock = pd.merge(tx_nm_data, trip_stock_data, on='date')
# nm_stock = pd.merge(nm_data,trip_stock_data,on='date')
#Keeping only the date part, time is 00:00:00 anyway
tx_nm_stock['date'] = tx_nm_stock['date'].dt.date
tx_nm_stock.to_csv("tx_nm_stock.csv")
```

From the X dataset, we are using the column "Stock Volatility" as its the most relevant column which gives us the stock price multiplied by the number of stocks bought/sold on a particular date which may or may not be impacted by our COVID data (we'll see that further!)

In [32]:
```
#Calculating pearson correlation between #deaths and stock volatility of tripadvisor
mean_x = tx_nm_stock['death'].mean()
mean_x
mean_y = tx_nm_stock['Volume'].mean()
mean_y

numerator = 0
temp = pd.DataFrame()

temp['x'] = tx_nm_stock['death'] - mean_x
temp['y'] = tx_nm_stock['Volume'] - mean_y

numerator = (temp['x'] * temp['y']).sum()
numerator

temp['xsq'] = temp['x'] * temp['x']
temp['ysq'] = temp['y'] * temp['y']

sumofxsquares = temp['xsq'].sum()
sumofysquares = temp['ysq'].sum()

denominator = np.sqrt(sumofxsquares) * np.sqrt(sumofysquares)
denominator

pearson_correlation_dv = np.divide(numerator,denominator)
pearson_correlation_dv
```

Out[32]:
```
-0.45852727730530113
```

In [33]:
```
#Calculating correlation between #positive cases and stock volatility of
trip advisor

#Calculating mean of both X and Y => Xbar, Ybar
mean_x = tx_nm_stock['positive'].mean()
# mean_x
mean_y = tx_nm_stock['Volume'].mean()
# mean_y

numerator = 0
temp = pd.DataFrame()

#Calculation of X-Xbar, Y - Ybar
temp['x'] = tx_nm_stock['positive'] - mean_x
temp['y'] = tx_nm_stock['Volume'] - mean_y

#Sum of products of (X-Xbar)(Y-Ybar)
numerator = (temp['x'] * temp['y']).sum()
# numerator

temp['xsq'] = temp['x'] * temp['x']
temp['ysq'] = temp['y'] * temp['y']

sumofxsquares = temp['xsq'].sum()
sumofysquares = temp['ysq'].sum()

denominator = np.sqrt(sumofxsquares) * np.sqrt(sumofysquares)
# denominator

pearson_correlation_cv = np.divide(numerator,denominator)
pearson_correlation_cv
```

Out[33]:
```
-0.4915368676458895
```

In [34]:
```
print("The pearson correlation coeffecient between the number of deaths
 and stock volatility is ",pearson_correlation_dv)
if pearson_correlation_dv > 0.5:
  print("Positive Linear Correlation")
elif pearson_correlation_dv < -0.5:
  print("Negative linear correlation")
elif pearson_correlation_dv <= 0.5:
  print("No linear correlation")
```

```
The pearson correlation coeffecient between the number of deaths and st
ock volatility is  -0.45852727730530113
No linear correlation
```

In [35]:
```
print("The pearson correlation coeffecient between the number of positiv
e cases and stock volatility is ",pearson_correlation_cv)
if pearson_correlation_cv > 0.5:
  print("Positive Linear Correlation")
elif pearson_correlation_cv < -0.5:
  print("Negative linear correlation")
elif pearson_correlation_cv <= 0.5:
  print("No linear correlation")
```

```
The pearson correlation coeffecient between the number of positive case
s and stock volatility is  -0.4915368676458895
No linear correlation
```

There is no linear correlation observed between the #deaths or stock volatility or even in #positive_cases or stock volatility

But we can surely see that the value is closer to a negative correlation than to a positive one.

Required Inference 5

Prior Posterior.

In [36]:
```
i=range(7,math.ceil(comb_data.shape[0]/7)*7,7)
week=np.split(comb_data,i)
```

In [37]:
```
week
```

Out[37]:

```
[         date  positive  negative  pending  hospitalizedCurrently  \
 0 2020-03-04       1.0       0.0      0.0                    0.0
 1 2020-03-05       1.0       0.0      0.0                    0.0
 2 2020-03-06       5.0      16.0      0.0                    0.0
 3 2020-03-07       8.0      48.0      0.0                    0.0
 4 2020-03-08       8.0      48.0      0.0                    0.0
 5 2020-03-09      12.0      57.0      0.0                    0.0
 6 2020-03-10      13.0      69.0      0.0                    0.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
 0                     0.0             0.0              0.0
 1                     0.0             0.0              0.0
 2                     0.0             0.0              0.0
 3                     0.0             0.0              0.0
 4                     0.0             0.0              0.0
 5                     0.0             0.0              0.0
 6                     0.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized  t
otal  \
 0                    0.0                     0.0  ...           0.0
1.0
 1                    0.0                     0.0  ...           0.0
1.0
 2                    0.0                     0.0  ...           0.0
21.0
 3                    0.0                     0.0  ...           0.0
56.0
 4                    0.0                     0.0  ...           0.0
56.0
 5                    0.0                     0.0  ...           0.0
69.0
 6                    0.0                     0.0  ...           0.0
82.0

    totalTestResults  posNeg  fips  deathIncrease  hospitalizedIncrease
\
 0               1.0     1.0    48            0.0                   0.0
 1               1.0     1.0    48            0.0                   0.0
 2              21.0    21.0    83            0.0                   0.0
 3              56.0    56.0    83            0.0                   0.0
 4              56.0    56.0    83            0.0                   0.0
 5              69.0    69.0    83            0.0                   0.0
 6              82.0    82.0    83            0.0                   0.0

    negativeIncrease  positiveIncrease  totalTestResultsIncrease
 0               0.0               0.0                       0.0
 1               0.0               0.0                       0.0
 2               0.0               4.0                       4.0
 3              32.0               3.0                      35.0
 4               0.0               0.0                       0.0
 5               9.0               4.0                      13.0
 6              12.0               1.0                      13.0

 [7 rows x 22 columns],
          date  positive  negative  pending  hospitalizedCurrently  \
 7  2020-03-11      24.0      87.0      0.0                    0.0
```

```
8  2020-03-12      28.0     155.0      0.0                    0.0
9  2020-03-13      49.0     190.0      0.0                    0.0
10 2020-03-14      61.0     237.0      0.0                    0.0
11 2020-03-15      69.0     482.0      0.0                    0.0
12 2020-03-16      74.0     566.0      0.0                    0.0
13 2020-03-17      87.0    2453.0      0.0                    0.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
7                      0.0             0.0              0.0
8                      0.0             0.0              0.0
9                      0.0             0.0              0.0
10                     0.0             0.0              0.0
11                     0.0             0.0              0.0
12                     0.0             0.0              0.0
13                     0.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
total  \
7                     0.0                     0.0  ...           0.0
111.0
8                     0.0                     0.0  ...           0.0
183.0
9                     0.0                     0.0  ...           0.0
239.0
10                    0.0                     0.0  ...           0.0
298.0
11                    0.0                     0.0  ...           0.0
551.0
12                    0.0                     0.0  ...           0.0
640.0
13                    0.0                     0.0  ...           0.0
2540.0

    totalTestResults  posNeg  fips  deathIncrease  hospitalizedIncreas
e  \
7              111.0   111.0    83            0.0                   0.
0
8              183.0   183.0    83            0.0                   0.
0
9              239.0   239.0    83            0.0                   0.
0
10             298.0   298.0    83            0.0                   0.
0
11             551.0   551.0    83            0.0                   0.
0
12             640.0   640.0    83            0.0                   0.
0
13            2540.0  2540.0    83            1.0                   0.
0

    negativeIncrease  positiveIncrease  totalTestResultsIncrease
7               18.0              11.0                      29.0
8               68.0               4.0                      72.0
9               35.0              21.0                      56.0
10              47.0              12.0                      59.0
11             245.0               8.0                     253.0
12              84.0               5.0                      89.0
```

```
 13           1887.0                13.0                        1900.0

 [7 rows x 22 columns],
          date  positive  negative  pending  hospitalizedCurrently  \
 14 2020-03-18     111.0    4150.0      0.0                    0.0
 15 2020-03-19     178.0    4974.0      0.0                    0.0
 16 2020-03-20     237.0    8854.0      0.0                    0.0
 17 2020-03-21     361.0    9989.0      0.0                    0.0
 18 2020-03-22     391.0   13144.0      0.0                    0.0
 19 2020-03-23     417.0   15024.0      0.0                    0.0
 20 2020-03-24     493.0   16647.0      0.0                    0.0

     hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
 14                     0.0             0.0              0.0
 15                     0.0             0.0              0.0
 16                     0.0             0.0              0.0
 17                     0.0             0.0              0.0
 18                     0.0             0.0              0.0
 19                     0.0             0.0              0.0
 20                     0.0             0.0              0.0

     onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
total  \
 14                    0.0                     0.0  ...           0.0
4261.0
 15                    0.0                     0.0  ...           0.0
5152.0
 16                    0.0                     0.0  ...           0.0
9091.0
 17                    0.0                     0.0  ...           0.0
10350.0
 18                    0.0                     0.0  ...           0.0
13535.0
 19                    0.0                     0.0  ...           0.0
15441.0
 20                    0.0                     0.0  ...           0.0
17140.0

     totalTestResults   posNeg  fips  deathIncrease  hospitalizedIncrea
se  \
 14           4261.0   4261.0    83            1.0
0.0
 15           5152.0   5152.0    83            1.0
0.0
 16           9091.0   9091.0    83            2.0
0.0
 17          10350.0  10350.0    83            0.0
0.0
 18          13535.0  13535.0    83            0.0
0.0
 19          15441.0  15441.0    83            3.0
0.0
 20          17140.0  17140.0    83            1.0
0.0

     negativeIncrease  positiveIncrease  totalTestResultsIncrease
 14           1697.0              24.0                    1721.0
```

```
 15           824.0              67.0                       891.0
 16          3880.0              59.0                      3939.0
 17          1135.0             124.0                      1259.0
 18          3155.0              30.0                      3185.0
 19          1880.0              26.0                      1906.0
 20          1623.0              76.0                      1699.0

 [7 rows x 22 columns],
          date  positive  negative  pending  hospitalizedCurrently  \
 21 2020-03-25    1074.0   19262.0      0.0                    0.0
 22 2020-03-26    1508.0   27709.0      0.0                    0.0
 23 2020-03-27    1867.0   30312.0      0.0                    0.0
 24 2020-03-28    2243.0   32404.0      0.0                   17.0
 25 2020-03-29    2789.0   33977.0      0.0                   19.0
 26 2020-03-30    3114.0   43945.0      0.0                   22.0
 27 2020-03-31    3547.0   51972.0      0.0                  218.0

     hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
 21                     0.0             0.0              0.0
 22                     0.0             0.0              0.0
 23                     0.0             0.0              0.0
 24                     0.0             0.0              0.0
 25                     0.0             0.0              0.0
 26                     0.0             0.0              0.0
 27                     0.0             0.0              0.0

     onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
total  \
 21                    0.0                     0.0  ...           0.0
20336.0
 22                    0.0                     0.0  ...           0.0
29217.0
 23                    0.0                     0.0  ...           0.0
32179.0
 24                    0.0                     0.0  ...           0.0
34647.0
 25                    0.0                     0.0  ...           0.0
36766.0
 26                    0.0                     0.0  ...           0.0
47059.0
 27                    0.0                     0.0  ...           0.0
55519.0

     totalTestResults   posNeg  fips  deathIncrease  hospitalizedIncrea
se  \
 21           20336.0  20336.0    83            4.0
0.0
 22           29217.0  29217.0    83            6.0
0.0
 23           32179.0  32179.0    83            5.0
0.0
 24           34647.0  34647.0    83            4.0
0.0
 25           36766.0  36766.0    83            8.0
0.0
 26           47059.0  47059.0    83            0.0
0.0
```

```
27           55519.0  55519.0    83            9.0
0.0

    negativeIncrease  positiveIncrease  totalTestResultsIncrease
21            2615.0             581.0                    3196.0
22            8447.0             434.0                    8881.0
23            2603.0             359.0                    2962.0
24            2092.0             376.0                    2468.0
25            1573.0             546.0                    2119.0
26            9968.0             325.0                   10293.0
27            8027.0             433.0                    8460.0

[7 rows x 22 columns],
         date  positive  negative  pending  hospitalizedCurrently  \
28 2020-04-01    4312.0   56785.0      0.0                  220.0
29 2020-04-02    5032.0   59658.0      0.0                  227.0
30 2020-04-03    5733.0   64809.0      0.0                  227.0
31 2020-04-04    6605.0   72778.0      0.0                  237.0
32 2020-04-05    7355.0   80411.0      0.0                  864.0
33 2020-04-06    7900.0   96593.0      0.0                 1198.0
34 2020-04-07    8948.0  101526.0      0.0                 1300.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
28                     0.0             0.0              0.0
29                     0.0             0.0              0.0
30                     0.0             0.0              0.0
31                     0.0             0.0              0.0
32                     0.0             0.0              0.0
33                     0.0             0.0              0.0
34                     0.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
\
28                    0.0                     0.0  ...           0.0
29                    0.0                     0.0  ...           0.0
30                    0.0                     0.0  ...           0.0
31                   18.0                     0.0  ...           0.0
32                   18.0                     0.0  ...           0.0
33                   18.0                     0.0  ...           0.0
34                   18.0                     0.0  ...           0.0

       total  totalTestResults    posNeg  fips  deathIncrease  \
28   61097.0           61097.0   61097.0    83           18.0
29   64690.0           64690.0   64690.0    83           13.0
30   70542.0           70542.0   70542.0    83           21.0
31   79383.0           79383.0   79383.0    83           18.0
32   87766.0           87766.0   87766.0    83           23.0
33  104493.0          104493.0  104493.0    83           14.0
34  110474.0          110474.0  110474.0    83           14.0

    hospitalizedIncrease  negativeIncrease  positiveIncrease  \
28                   0.0            4813.0             765.0
29                   0.0            2873.0             720.0
30                   0.0            5151.0             701.0
31                   0.0            7969.0             872.0
32                   0.0            7633.0             750.0
33                   0.0           16182.0             545.0
```

```
34                 0.0          4933.0           1048.0

    totalTestResultsIncrease
28                    5578.0
29                    3593.0
30                    5852.0
31                    8841.0
32                    8383.0
33                   16727.0
34                    5981.0

[7 rows x 22 columns],
         date  positive  negative  pending  hospitalizedCurrently  \
35 2020-04-08   10147.0  108356.0      0.0                 1542.0
36 2020-04-09   11095.0  118846.0      0.0                 1498.0
37 2020-04-10   12762.0  130054.0      0.0                 1605.0
38 2020-04-11   13652.0  133979.0      0.0                 1589.0
39 2020-04-12   14658.0  138567.0      0.0                 1416.0
40 2020-04-13   15151.0  148590.0      0.0                 1256.0
41 2020-04-14   15969.0  162468.0      0.0                 1496.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
35                     0.0             0.0              0.0
36                     0.0             0.0              0.0
37                     0.0             0.0              0.0
38                     0.0             0.0              0.0
39                     0.0             0.0              0.0
40                     0.0             0.0              0.0
41                   181.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
\
35                   18.0                     0.0  ...           0.0
36                   18.0                     0.0  ...           0.0
37                   18.0                     0.0  ...           0.0
38                   18.0                     0.0  ...           0.0
39                   18.0                     0.0  ...           0.0
40                    0.0                     0.0  ...           0.0
41                    0.0                     0.0  ...         181.0

       total  totalTestResults    posNeg  fips  deathIncrease  \
35  118503.0          118503.0  118503.0    83           24.0
36  129941.0          129941.0  129941.0    83           25.0
37  142816.0          142816.0  142816.0    83           30.0
38  147631.0          147631.0  147631.0    83           28.0
39  153225.0          153225.0  153225.0    83           18.0
40  163741.0          163741.0  163741.0    83           22.0
41  178437.0          178437.0  178437.0    83           36.0

    hospitalizedIncrease  negativeIncrease  positiveIncrease  \
35                   0.0            6830.0            1199.0
36                   0.0           10490.0             948.0
37                   0.0           11208.0            1667.0
38                   0.0            3925.0             890.0
39                   0.0            4588.0            1006.0
40                   0.0           10023.0             493.0
41                 181.0           13878.0             818.0
```

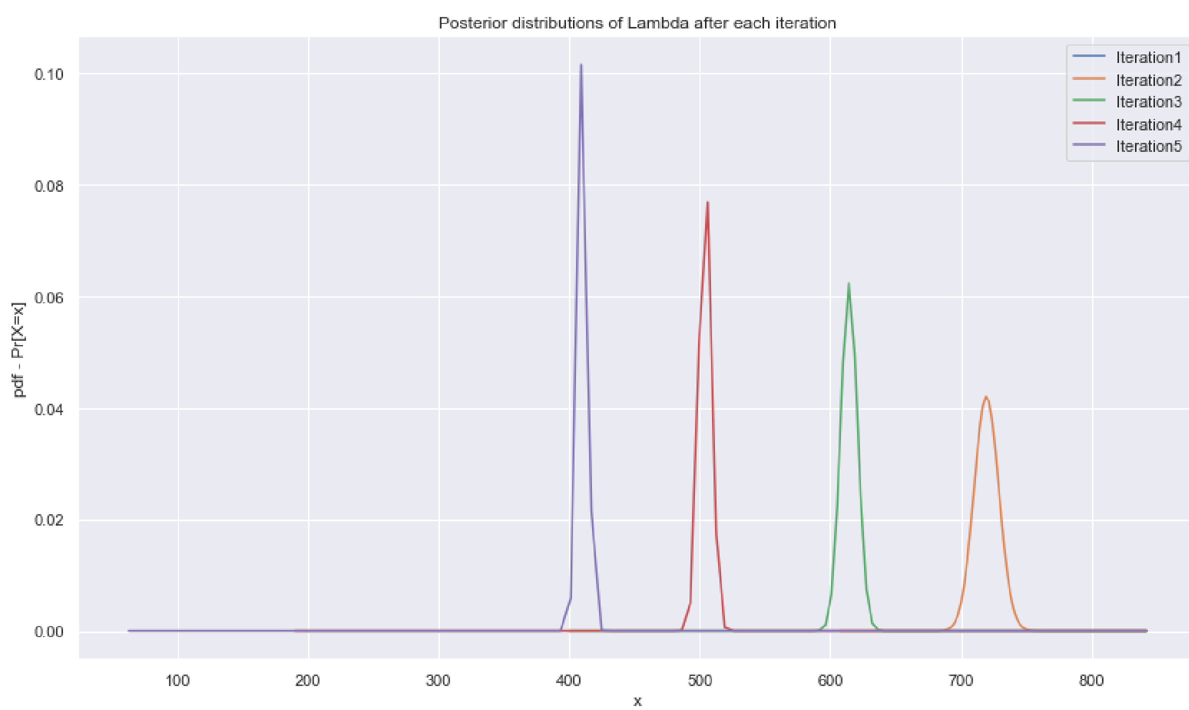
```
    totalTestResultsIncrease
35                    8029.0
36                   11438.0
37                   12875.0
38                    4815.0
39                    5594.0
40                   10516.0
41                   14696.0

[7 rows x 22 columns],
         date  positive  negative  pending  hospitalizedCurrently  \
42 2020-04-15   16899.0  167761.0      0.0                 1625.0
43 2020-04-16   17939.0  174002.0      0.0                 1549.0
44 2020-04-17   18968.0  185056.0      0.0                 1612.0
45 2020-04-18   19971.0  191881.0      0.0                 1417.0
46 2020-04-19   20721.0  198621.0      0.0                 1563.0
47 2020-04-20   21303.0  206133.0      0.0                 1514.0
48 2020-04-21   22167.0  221987.0      0.0                 1535.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
42                   181.0             0.0              0.0
43                   215.0             0.0              0.0
44                   230.0             0.0              0.0
45                   242.0             0.0              0.0
46                   258.0             0.0              0.0
47                   274.0             0.0              0.0
48                   291.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
\
42                    0.0                     0.0  ...         181.0
43                    0.0                     0.0  ...         215.0
44                    0.0                     0.0  ...         230.0
45                    0.0                     0.0  ...         242.0
46                    0.0                     0.0  ...         258.0
47                    0.0                     0.0  ...         274.0
48                    0.0                     0.0  ...         291.0

       total  totalTestResults    posNeg  fips  deathIncrease  \
42  184660.0          184660.0  184660.0    83           51.0
43  191941.0          191941.0  191941.0    83           29.0
44  204024.0          204024.0  204024.0    83           43.0
45  211852.0          211852.0  211852.0    83           32.0
46  219342.0          219342.0  219342.0    83           26.0
47  227436.0          227436.0  227436.0    83           20.0
48  244154.0          244154.0  244154.0    83           25.0

    hospitalizedIncrease  negativeIncrease  positiveIncrease  \
42                   0.0            5293.0             930.0
43                  34.0            6241.0            1040.0
44                  15.0           11054.0            1029.0
45                  12.0            6825.0            1003.0
46                  16.0            6740.0             750.0
47                  16.0            7512.0             582.0
48                  17.0           15854.0             864.0
```

```
    totalTestResultsIncrease
42                    6223.0
43                    7281.0
44                   12083.0
45                    7828.0
46                    7490.0
47                    8094.0
48                   16718.0

[7 rows x 22 columns],
         date  positive  negative  pending  hospitalizedCurrently  \
49 2020-04-22   23141.0  234519.0      0.0                 1797.0
50 2020-04-23   24154.0  242156.0      0.0                 1770.0
51 2020-04-24   25185.0  263925.0      0.0                 1797.0
52 2020-04-25   26294.0  288032.0      0.0                 1749.0
53 2020-04-26   27291.0  302465.0      0.0                 1694.0
54 2020-04-27   28023.0  319109.0      0.0                 1711.0
55 2020-04-28   28994.0  330193.0      0.0                 1837.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
49                   306.0             0.0              0.0
50                   331.0             0.0              0.0
51                   367.0             0.0              0.0
52                   412.0             0.0              0.0
53                   412.0             0.0              0.0
54                   412.0             0.0              0.0
55                   481.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
\
49                    0.0                     0.0  ...         306.0
50                    0.0                     0.0  ...         331.0
51                    0.0                     0.0  ...         367.0
52                    0.0                     0.0  ...         412.0
53                    0.0                     0.0  ...         412.0
54                    0.0                     0.0  ...         412.0
55                    0.0                     0.0  ...         481.0

       total  totalTestResults    posNeg  fips  deathIncrease  \
49  257660.0          257660.0  257660.0    83           33.0
50  266310.0          266310.0  266310.0    83           24.0
51  289110.0          289110.0  289110.0    83           39.0
52  314326.0          314326.0  314326.0    83           36.0
53  329756.0          329756.0  329756.0    83           34.0
54  347132.0          347132.0  347132.0    83           21.0
55  359187.0          359187.0  359187.0    83           32.0

    hospitalizedIncrease  negativeIncrease  positiveIncrease  \
49                  15.0           12532.0             974.0
50                  25.0            7637.0            1013.0
51                  36.0           21769.0            1031.0
52                  45.0           24107.0            1109.0
53                   0.0           14433.0             997.0
54                   0.0           16644.0             732.0
55                  69.0           11084.0             971.0

    totalTestResultsIncrease
```

```
49                   13506.0
50                    8650.0
51                   22800.0
52                   25216.0
53                   15430.0
54                   17376.0
55                   12055.0

[7 rows x 22 columns],
         date  positive  negative  pending  hospitalizedCurrently  \
56 2020-04-29   30028.0  346507.0      0.0                 1859.0

    hospitalizedCumulative  inIcuCurrently  inIcuCumulative  \
56                   509.0             0.0              0.0

    onVentilatorCurrently  onVentilatorCumulative  ...  hospitalized
\
56                    0.0                     0.0  ...         509.0

       total  totalTestResults    posNeg  fips  deathIncrease  \
56  376535.0          376535.0  376535.0    83           48.0

    hospitalizedIncrease  negativeIncrease  positiveIncrease  \
56                  28.0           16314.0            1034.0

    totalTestResultsIncrease
56                   17348.0

[1 rows x 22 columns]]
```

In [38]:
```
#After solving, we found the distribution to be gamma distribution with
 alpha = n*mean(X) and beta = n+(1/beta_prior)
lambda_m=week[len(week)-1].death.mean()
beta_pri=lambda_m

temp = []
plt.figure(figsize=(14,8))
for i in range(5):
    temp.extend(week[len(week)-1-i].death)
    max_temp = max(temp)
    min_temp = min(temp)
    mean_ = np.mean(temp)
    range_ = np.linspace(min_temp, max_temp, 100)
    alpha = (len(temp)*mean_)+1
    beta = len(temp)+(1/beta_pri)
    g_pdf = gamma.pdf(range_, a = alpha, scale = float(1)/beta)
    print("MAP " + str(i+1)+ " = " + str(range_[np.argmax(g_pdf)]))
    plt.plot(range_, g_pdf, label="Iteration"+str(i+1))


plt.legend()
plt.xlabel("x")
plt.ylabel("pdf - Pr[X=x]")
plt.title('Posterior distributions of Lambda after each iteration')
plt.show()
```

```
MAP 1 = 842.0
MAP 2 = 719.0909090909091
MAP 3 = 614.3030303030303
MAP 4 = 506.1212121212121
MAP 5 = 409.22222222222223
```



I have kept iteration values 5 as april data is getting divided into 5 parts and there is only one day (April29th) in the last part and to get the inference for the ntire month of April we kept the iteration values as 5. Also,I started my calculations from the last week to the first week of April.

Creative Inference

Creative Inference 1

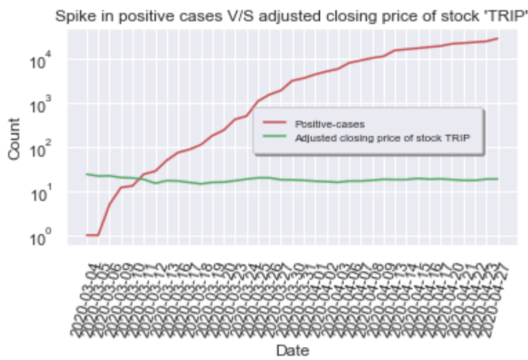
For the new creative inference - We choose the chi square test to show whether or not the X dataset and the COVID dataset had an impact on each other

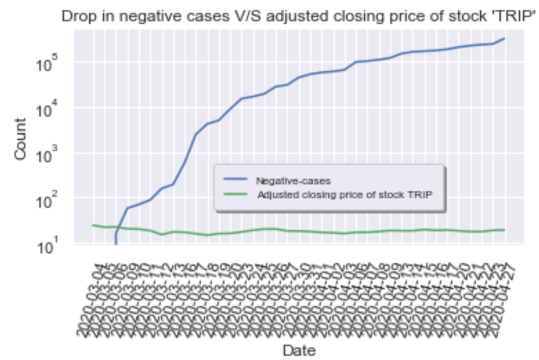
The hypothesis we present is listed down below.

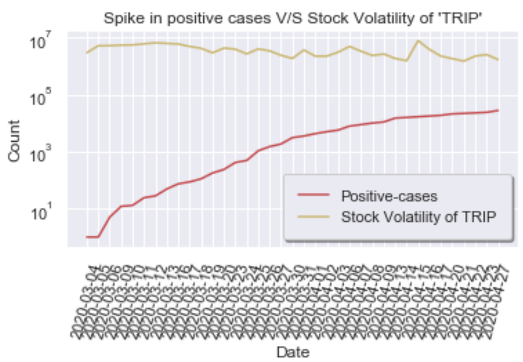
Before actually beginning with our chi-square test, we'd like to see the trends in the data by plotting a few graphs

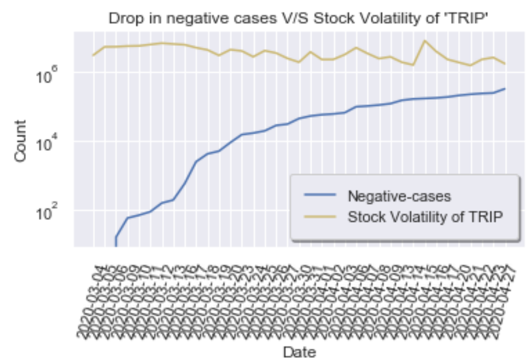
In [39]:
```
tx_nm_stock =  pd.read_csv("tx_nm_stock.csv")
trip_stock_data = pd.read_csv("TRIP.csv")
```

In [40]:
```
#We'd just like to see if there is any relation between the stock price
 and the spike in positive cases when plotted on a graph

fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15, 10))
fig.tight_layout(pad=10.0)
sns.set()
axes[0][0].set_yscale('log')
axes[1][0].set_yscale('log')
axes[0][1].set_yscale('log')
axes[1][1].set_yscale('log')
axes[0][0].plot(tx_nm_stock['date'],tx_nm_stock["positive"],'-r', label=
'Positive-cases');
axes[0][0].tick_params(axis='x', labelrotation=75);
axes[0][0].plot(tx_nm_stock["date"],tx_nm_stock["Adj Close"],'-g',label=
'Adjusted closing price of stock TRIP');
axes[0][0].set(title = "Spike in positive cases V/S adjusted closing pri
ce of stock 'TRIP'",xlabel = "Date",ylabel = "Count");
axes[0][0].legend(loc='best', shadow=True , prop={'size': 8}, bbox_to_an
chor = (0.4,0.4), fancybox=True, framealpha=1, borderpad=1);

axes[0][1].plot(tx_nm_stock['date'],tx_nm_stock["negative"],'-b', label=
'Negative-cases');
axes[0][1].tick_params(axis='x', labelrotation=75);
axes[0][1].plot(tx_nm_stock["date"],tx_nm_stock["Adj Close"],'-g',label=
'Adjusted closing price of stock TRIP');
axes[0][1].set(title = "Drop in negative cases V/S adjusted closing pric
e of stock 'TRIP'",xlabel = "Date",ylabel = "Count");
axes[0][1].legend(loc='best', shadow=True, prop={'size': 8}, bbox_to_anc
hor = (0.3,0.4), fancybox=True, framealpha=1, borderpad=1);

axes[1][0].plot(tx_nm_stock['date'],tx_nm_stock["positive"],'-r', label=
'Positive-cases');
axes[1][0].tick_params(axis='x', labelrotation=75);
axes[1][0].plot(tx_nm_stock["date"],tx_nm_stock["Volume"],'-y',label='St
ock Volatility of TRIP');
axes[1][0].set(title = "Spike in positive cases V/S Stock Volatility of
 'TRIP'",xlabel = "Date",ylabel = "Count");
axes[1][0].legend(loc='lower right', shadow=True, fancybox=True, frameal
pha=1, borderpad=1);

axes[1][1].plot(tx_nm_stock['date'],tx_nm_stock["negative"],'-b', label=
'Negative-cases');
axes[1][1].tick_params(axis='x', labelrotation=75);
axes[1][1].plot(tx_nm_stock["date"],tx_nm_stock["Volume"],'-y',label='St
ock Volatility of TRIP');
axes[1][1].set(title = "Drop in negative cases V/S Stock Volatility of
 'TRIP'",xlabel = "Date",ylabel = "Count");
axes[1][1].legend(loc='lower right', shadow=True, fancybox=True, frameal
pha=1, borderpad=1);
```









In [41]:
```python
#Basic functions we need ahead in our program of chi square calculation

#Returns the value rounded to 2 decimals
def get_round(val):
  return round(val, 2)

#Returns percentage difference
def get_percentage_difference(col):
  return col.pct_change(periods = 1)

#Returns difference between expected and observed
def get_difference_exp_obs(obs, exp):
  return((obs - exp) ** 2)/exp
```

Hypothesis

Tripadvisor is an American online travel company that offers online hotel reservations as well as bookings for transportation, lodging, travel experiences, and restaurants. With chi square test, we aim to show whether or not the increase in number of people booking tickets and travelling to/fro(can be identified by adjusted closing price of a stock) is related to the spike in number of positive cases of COVID.

Null Hypothesis:

H0: The change in the adjusted closing price of stock "TRIP" is independent of the rise in positive cases.

Alternate Hypothesis:

H1: The adjusted closing price of stock "TRIP" is related to the number of positive cases of COVID.

In [42]:
```
#We look at the independence/dependence between adj close price and the
 change in positive cases

adj_close_percent_diff = pd.Series(get_percentage_difference(tx_nm_stock
['Adj Close']))
cases_percent_diff = pd.Series(get_percentage_difference(tx_nm_stock['po
sitive']))

#print(adj_close_percent_diff)
tx_nm_stock['adjusted_closeprice_percent_change'] = adj_close_percent_di
ff
tx_nm_stock['positivecase_percent_change'] = cases_percent_diff

#To remove the headers we assign the rows from row1 instead of 0 to the
 df
tx_nm_stock = tx_nm_stock.iloc[1:]

#Calculating difference between the percentage differences
adjstockprice = pd.Series(get_percentage_difference(tx_nm_stock['adjuste
d_closeprice_percent_change']))
cases = pd.Series(get_percentage_difference(tx_nm_stock['positivecase_pe
rcent_change']))

tx_nm_stock['stockprice_slope'] = adjstockprice
tx_nm_stock['cases_slope'] = cases

#To remove the headers we assign the rows from row1 instead of 0 to the
 df
tx_nm_stock = tx_nm_stock.iloc[1:]
```

In [43]:
```
#Marking the confusion matrix - Positives and Negatives
tx_nm_stock['Lpositive'] = np.where(tx_nm_stock['cases_slope'] >= 0, 'Po
sitive', 'Negative')
tx_nm_stock['Lstock'] = np.where(tx_nm_stock['stockprice_slope'] >= 0,
'Positive', 'Negative')
tx_nm_stock.iloc[:,20:-1].head(1)
```

Out[43]:

	hospitalizedIncrease	negativeIncrease	positiveIncrease	totalTestResultsIncrease	state	Open
2	0.0	0.0	4.0	4.0	TX,NM	20.92

alpha = 0.05

We will calculate degree of freedom as:

degree_of_freedom = (2 - 1) * (2 - 1) = 1

We reject when p-value < alpha for a given degree of freedom. (in our case its 1)

Reference:-

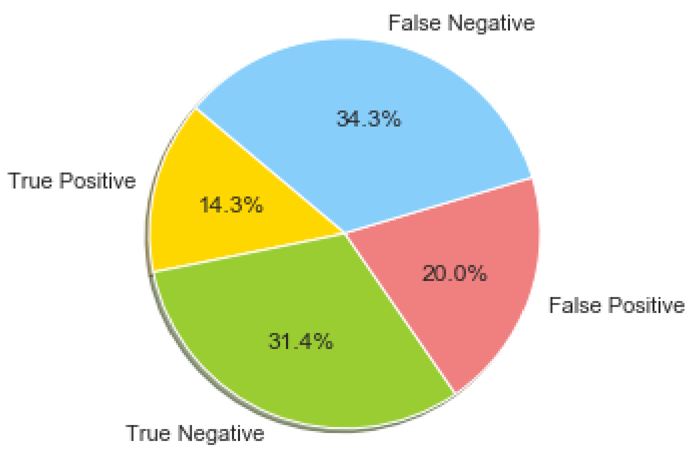
[http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html (http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html)](http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html)

Therefore if our p-value comes out to be less than alpha, then we reject the null hypothesis.

In [44]:
```
#Forming the table for True positives(TP), False Positives(FP) - Type1 e
rror, True Negatives(TN), False Negatives(FN) - Type2error
Q = pd.pivot_table(tx_nm_stock, index = ['Lpositive'], columns=['Lstock'
], aggfunc=len, fill_value=0)
Q = Q['Adj Close']
Q

true_positives = Q['Positive']['Positive']
true_negatives = Q['Negative']['Negative']
false_positives = Q['Positive']['Negative']
false_negatives = Q['Negative']['Positive']

#Calculating precision and recall
precision = true_positives / (true_positives + false_positives)
recall = true_positives / (true_positives + false_negatives)

print("The precision is ",precision)
print("The recall is ",recall)
```

```
The precision is  0.4166666666666667
The recall is  0.29411764705882354
```

In [45]:
```
#Visualizing the percentage of TP, TN, FP, FN
labels = ['True Positive','True Negative','False Positive','False Negati
ve']
sizes = [true_positives, true_negatives, false_positives, false_negative
s]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
plt.pie(sizes, labels = labels, colors = colors, autopct = '%1.1f%%', sh
adow = True, startangle = 140)
plt.axis('equal')
plt.show()
```



In [46]:
```
#Now we get the no. of days where TP, TN, FP, FN are observed as per our
conditions
stats_table = tx_nm_stock.groupby(['Lpositive','Lstock'])['date'].count
().reset_index()
print(stats_table)
stats_table.columns = ['Lpositive','Lstock','noOfDates']
print(stats_table)
```

```
  Lpositive    Lstock  date
0  Negative  Negative    11
1  Negative  Positive     7
2  Positive  Negative    12
3  Positive  Positive     5
  Lpositive    Lstock  noOfDates
0  Negative  Negative         11
1  Negative  Positive          7
2  Positive  Negative         12
3  Positive  Positive          5
```

In [47]:
```
#To get total number of days of positive spike and negative drop we do b
elow.

#We need the total sum to get the expected values
agg_count = stats_table['noOfDates'].sum()
no_of_days = agg_count

#Two variables in consideration
v1 = 'Lpositive'
v2 = 'Lstock'

#Checking the conditions for TP, FP, TN, FN
cond1 = (stats_table[v1] == 'Positive')
cond2 = (stats_table[v2] == 'Positive')
cond3 = (stats_table[v1] == 'Negative')
cond4 = (stats_table[v2] == 'Negative')

#Filtering on those conditions
temp_pos_table1 = stats_table[cond1]
temp_pos_table2 = stats_table[cond2]
temp_neg_table1 = stats_table[cond3]
temp_neg_table2 = stats_table[cond4]

#Getting the ratios
ratio_pos1 = temp_pos_table1.noOfDates.sum() / no_of_days
ratio_pos2 = temp_pos_table2.noOfDates.sum() / no_of_days

#Just rounding it off to nearest 2 decimals
total_cases = get_round(ratio_pos1)
total_stockprices = get_round(ratio_pos2)

#The observed values for TP, TN, FP, FN
obs_pos_pos = stats_table[cond1 & cond2].noOfDates.sum()
obs_pos_neg = stats_table[cond1 & cond4].noOfDates.sum()
obs_neg_pos = stats_table[cond3 & cond2].noOfDates.sum()
obs_neg_neg = stats_table[cond3 & cond4].noOfDates.sum()

#Alternate cases
comp_tot_cases = 1 - total_cases
comp_tot_stock = 1 - total_stockprices

#Calculation of expected values
exp_pos_pos = total_cases * total_stockprices * no_of_days
exp_pos_neg = total_cases * comp_tot_stock * no_of_days
exp_neg_pos = comp_tot_cases * total_stockprices * no_of_days
exp_neg_neg = comp_tot_cases * comp_tot_stock * no_of_days
```

In [48]:
```
#Calculating the sum of (obs - exp) ^ 2 / exp
term1 = get_difference_exp_obs(obs_pos_pos, exp_pos_pos)
term2 = get_difference_exp_obs(obs_pos_neg, exp_pos_neg)
term3 = get_difference_exp_obs(obs_neg_pos, exp_neg_pos)
term4 = get_difference_exp_obs(obs_neg_neg, exp_neg_neg)

results = term1 + term2 + term3 + term4
results
```

Out[48]:
```
0.3539939021190341
```

From http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html (http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pchisq.html)

From above link, we get p-value as 0.5519 for df = 1 and chi sq value of 0.3539939021190341

In [49]:
```
p_value = 0.5519 #after looking up on the link with chi sq value as 0.09
676343009676323 and df = 1
alpha = 0.05 #threshold
print("The p-value is",p_value)
if(p_value < alpha):
  print("The result is significant at alpha and we reject the null hypot
hesis.")
  print("Thus we can say that the value of stock 'TRIP' is related to th
e number of positive cases of COVID.")
else:
  print("The result is not significant at alpha and we fail to reject th
e null hypothesis.")
  print("Thus we can say that the change in the adjusted closing price o
f stock 'TRIP' is independent of the rise in positive cases.")
```

```
The p-value is 0.5519
The result is not significant at alpha and we fail to reject the null h
ypothesis.
Thus we can say that the change in the adjusted closing price of stock
'TRIP' is independent of the rise in positive cases.
```

Conclusion:

The above chi-square test shows that the change in value of the stock's closing price is independent of the rise in positive cases; now this could be because - maybe people who were tested positive in the states of Texas and new Mexico did not use TripAdvisor much to book flights/hotels but used some other means to travel. This result is plainly based on the data at hand.

The graphs plotted above do show the independent nature of both the factors considered.

Creative Inference 2

Hypothesis

H0: The onset of coronavirus has caused a distruption in the US China trade relations.

H1: The US China trade relations are statistically unaffected.

We can measure this by calculating the difference in the projected values for March 2020 for import and export. We specifically chose March because the US was affected most by the virus in late february and early-mid march(As shown in the Covid Dataset).

In [50]:
```
x_df = pd.read_excel("USCensusDatasetByCountry.xlsx")
china_census = x_df[x_df["CTYNAME"] == "China"]
```

In [51]:
```
#Calculate the differences in predictions for the year 2020
print("AR 3:", AR(3, china_census["IMAR"].to_numpy(), 1))
print("AR 5:", AR(5, china_census["IMAR"].to_numpy(), 1))
print("EWMA:", EWMA(china_census["IMAR"].to_numpy(), 0.5, 1))
```

```
Real Data [19805.426244]
Predicted Values [36271.74413134652]
AR 3: {'MAPE %': 83.14043678981635, 'MSE': 271139624.7671479}
Real Data [19805.426244]
Predicted Values [37289.11283075228]
AR 5: {'MAPE %': 88.27725478540972, 'MSE': 305679296.6637817}
Real Data [19805.426244]
Predicted Values [26659.79947013768]
EWMA: {'MAPE %': 34.6085620258448, 'MSE': 46982432.3231931}
```

In [52]:
```
#Exports March
print("AR 3:", AR(3, china_census["EMAR"].to_numpy(), 1))
print("AR 5:", AR(5, china_census["EMAR"].to_numpy(), 1))
print("EWMA:", EWMA(china_census["EMAR"].to_numpy(), 0.5, 1))
```

```
Real Data [7971.889801]
Predicted Values [10485.732904507116]
AR 3: {'MAPE %': 31.53384161421521, 'MSE': 6319407.14905029}
Real Data [7971.889801]
Predicted Values [9584.596179858818]
AR 5: {'MAPE %': 20.229913096095714, 'MSE': 2600821.86441192}
Real Data [7971.889801]
Predicted Values [9372.506953776516]
EWMA: {'MAPE %': 17.56944949992687, 'MSE': 1961728.408651793}
```

Conclusion

While the imports were infact affected by the virus significantly the exports stayed very close to the predicted values. But since most of the United States' largest exports to China consist of crops or raw materials according to https://www.chinabusinessreview.com/what-america-exports-to-china/ (https://www.chinabusinessreview.com/what-america-exports-to-china/) its safe to say that the chances of the exports going down were less to begin with.

Creative Inference 3- Regression

I wanted to understand the correlation between various columns of the covid and X dataset and so I combined them into one and plotted a heatmap to visualize the correlation values and decide the columns for my inference.

In [53]:
```
#Loading Covid data
covid=pd.read_csv("daily.csv")
states = ['TX','NM']
covid = full_data[full_data.state.isin(states)]
covid=covid.groupby(['date']).sum().reset_index()
covid = covid.fillna("0")
covid = covid.fillna(texas_newmexico_data.mean())
covid['death'] = covid['death'].astype(int)
covid.astype(int)
```

Out[53]:

	date	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	inIcuCurr
0	20200304	1	0	0	0	0	
1	20200305	1	0	0	0	0	
2	20200306	5	16	0	0	0	
3	20200307	8	48	0	0	0	
4	20200308	8	48	0	0	0	
5	20200309	12	57	0	0	0	
6	20200310	13	69	0	0	0	
7	20200311	24	87	0	0	0	
8	20200312	28	155	0	0	0	
9	20200313	49	190	0	0	0	
10	20200314	61	237	0	0	0	
11	20200315	69	482	0	0	0	
12	20200316	74	566	0	0	0	
13	20200317	87	2453	0	0	0	
14	20200318	111	4150	0	0	0	
15	20200319	178	4974	0	0	0	
16	20200320	237	8854	0	0	0	
17	20200321	361	9989	0	0	0	
18	20200322	391	13144	0	0	0	
19	20200323	417	15024	0	0	0	
20	20200324	493	16647	0	0	0	
21	20200325	1074	19262	0	0	0	
22	20200326	1508	27709	0	0	0	
23	20200327	1867	30312	0	0	0	
24	20200328	2243	32404	0	17	0	
25	20200329	2789	33977	0	19	0	
26	20200330	3114	43945	0	22	0	
27	20200331	3547	51972	0	218	0	
28	20200401	4312	56785	0	220	0	
29	20200402	5032	59658	0	227	0	
30	20200403	5733	64809	0	227	0	
31	20200404	6605	72778	0	237	0	
32	20200405	7355	80411	0	864	0	
33	20200406	7900	96593	0	1198	0	

	date	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	inIcuCurr
34	20200407	8948	101526	0	1300	0	
35	20200408	10147	108356	0	1542	0	
36	20200409	11095	118846	0	1498	0	
37	20200410	12762	130054	0	1605	0	
38	20200411	13652	133979	0	1589	0	
39	20200412	14658	138567	0	1416	0	
40	20200413	15151	148590	0	1256	0	
41	20200414	15969	162468	0	1496	181	
42	20200415	16899	167761	0	1625	181	
43	20200416	17939	174002	0	1549	215	
44	20200417	18968	185056	0	1612	230	
45	20200418	19971	191881	0	1417	242	
46	20200419	20721	198621	0	1563	258	
47	20200420	21303	206133	0	1514	274	
48	20200421	22167	221987	0	1535	291	
49	20200422	23141	234519	0	1797	306	
50	20200423	24154	242156	0	1770	331	
51	20200424	25185	263925	0	1797	367	
52	20200425	26294	288032	0	1749	412	
53	20200426	27291	302465	0	1694	412	
54	20200427	28023	319109	0	1711	412	
55	20200428	28994	330193	0	1837	481	
56	20200429	30028	346507	0	1859	509	

57 rows × 22 columns

In [54]:
```
#Merging the two datasets and filtering a little to match dimensions and
number of rows
covid=covid[['negative','positive','hospitalizedCurrently','recovered',
'death','total','negativeIncrease','positiveIncrease','deathIncrease','t
otalTestResultsIncrease']]
X=pd.read_csv('TRIP.csv')
X=X[['Open','High','Low','Close','Adj Close','Volume']]
merged=pd.concat([X, covid], axis=1)
merged=merged.head(42)
```

After this we first performed Multiple Linear Regression by considering various columns of the X dataset as the vector of independent variables and a single column of covid dataset as the dependent variable. After experimenting with various columns the best RMSE error achieved was 88.51390270429455 on the number of deaths column.
It is agreed that the error value is not so great but given the limited amount of training data, we believe it seems good enough and there is a clear relationship between the change in the stock values of TripAdvisor (which can be equated to number of flights and transactions which give us a measure of travel) with the number of deaths. This seems like as the number of people who travel increases, more people get infected and hence the number of deaths indirectly increases.

Error functions:

In [55]:
```
#Formula=(sum(abs(Yi-Y_hat))/Yi)*100/n
def mape(y_hat,test):
  error=0
  for i in range(len(test)):
    error=error+((abs(test[i]-y_hat[i])/test[i])*100)
  return error/len(test)

#Formula=sum((Yi-Y_hat)^2)/n
def mse(y_hat,test):
  error=0
  for i in range(len(test)):
    error=error+((test[i]-y_hat[i])*(test[i]-y_hat[i]))
  return error/len(test)

#Formula=sqrt(sum((Yi-Y_hat)^2)/n)
import math
def rmse(y_hat,test):
  error=0
  for i in range(len(test)):
    error=error+((test[i]-y_hat[i])*(test[i]-y_hat[i]))
  return math.sqrt(error/len(test))

def plot(X1,Y1,Y1_hat):
  plt.plot(X1, Y1, 'r', label='Original data')
  plt.plot(X1, Y1_hat, 'g', label='Fitted line')
  plt.legend()
  plt.show()
```

Multiple Linear Regression

In [56]:
```
#Selecting a vector of Independent variables
X.values.tolist()
#Selecting the dependent variable
test=merged[['death']].values.tolist()
#Idea: Understand the relationship between the various features related
 to stock values(flight transactions) and the number of deaths

#Implementing the formula: b_hat=(x.T*x)^-1*x.T*y
emp=np.ones((len(test),1))
x=np.matrix(X)
x=np.concatenate((emp,x),axis=1)
t=x.T
prod=np.matmul(t,x)
inv=np.linalg.pinv(prod)
res_x=np.matmul(inv,t)
y=np.matrix(test)

b_hat=np.matmul(res_x,y)

#Implementing the formula:y_hat=x*b_hat
y_hat=np.matmul(x,b_hat)
```

In [57]:
```
#Calculating error scores between the true value and the predicted value
print("MAPE=",mape(y_hat,test))
print("MSE=",mse(y_hat,test))
print("RMSE=",rmse(y_hat,test))
```

```
MAPE= [[inf]]
MSE= [[7834.71097259]]
RMSE= 88.51390270795827

/Users/sakshigupta/anaconda3/lib/python3.7/site-packages/ipykernel_laun
cher.py:5: RuntimeWarning: divide by zero encountered in true_divide
  """
```

In an attempt to decrease the error a little more, we tried our hand at Simple Linear Regression with Volume as the Independent feature and death as the dependent feature after lots of experimenting. This combination gives an RMSE of 9.3757042719415 which seems really good and supports our above inference and idea.

Note: MAPE comes out to be inf as there are lots of 0's in the data avaialble to us(true values) and division by zero leads to inf value.

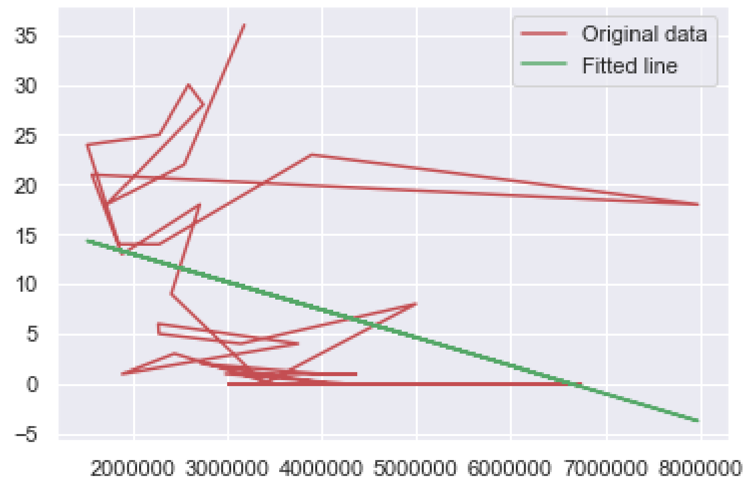
Simple Linear Regression

In [58]:
```
#Similar to Assignment 6
#Formulae
#b1_hat = (sum(X_i*Y_i) - n*Xbar*Ybar)/(sum(Xi*Xi) - n*Xbar*Xbar)
#b0_hat = Ybar - b1_hat*Xbar
#Y_i_hat = b0_hat + b1_hat*X_i
#Calculating the value of b1_hat
def calculate_b1hat(X,Y):
    b1_hat = (np.sum(X*Y) - (np.mean(X)*np.mean(Y)*len(X)))/(np.sum(X*X)
- (len(X)*np.mean(X)*np.mean(X)))
    return b1_hat
#Calculating the value of b0_hat
def calculate_b0hat(X,Y,b1_hat):
    b0_hat = np.mean(Y) - (b1_hat*np.mean(X))
    return b0_hat
#Calculating the value of Y_hat
def calculate_Yhat(b0_hat, b1_hat, X):
  Y_hat = b0_hat + (b1_hat*X)
  return Y_hat
```

In [59]:
```
#Choosing features for performing simple linear regression
#Independent feature is Volume from X dataset and dependent feature is d
eathIncrease from Covid dataset
#Intention is to analyze the relationship between the number of flights
 booked on TripAdvisor(value of Volume) and the increase in number of de
aths
X1=X['Volume'].values
Y1=merged['deathIncrease'].values
#Calling the above functions
b11_hat = calculate_b1hat(X1, Y1)
b01_hat = calculate_b0hat(X1, Y1, b11_hat)
Y1_hat = calculate_Yhat(b01_hat, b11_hat, X1)
```

In [60]:
```
#Calculating error scores between the true value and the predicted value
print("MAPE=",mape(Y1,Y1_hat))
print("MSE=",mse(Y1,Y1_hat))
print("RMSE=",rmse(Y1,Y1_hat))
plot(X1,Y1,Y1_hat)
```

```
MAPE= 68.05521515698895
MSE= 87.90383059490209
RMSE= 9.3757042719415
```



Since we computed the correlations before performing regression and there were some evident correlations, We think that regression works for this task and provides us with some interesting insights.