Author - AYUSH CHHOKER

DATA SCIENCE AND BUSINESS ANALYTICS INTERN

Task 8 : Timeline Analysis : Covid-19

Data set = "http://covidtracking.com/api/states/daily.csv (http://covidtracking.com/api/states/daily.csv)"

In [1]:

```
import numpy as np
import pandas as pd
import io
import requests
import matplotlib.pyplot as plt
```

In [6]:

```
url= "http://covidtracking.com/api/states/daily.csv"
s=requests.get(url).content
```

In [7]:

```
df = pd.read_csv(io.StringIO(s.decode('utf-8')))
```

Converts dates to a specific format

In [8]:

```
df['date'] =  pd.to_datetime(df['date'], format='%Y%m%d')
```

Drops unnecessary column(s)

In [9]:

```
df.drop(['dateChecked'],axis=1,inplace=True)
```

Converts the state data to string-type

In [10]:

```
df['state']=df['state'].apply(str)
```

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20780 entries, 0 to 20779
Data columns (total 55 columns):
 #   Column                        Non-Null Count  Dtype         
---  ------                        --------------  -----         
 0   date                          20780 non-null  datetime64[ns]
 1   state                         20780 non-null  object        
 2   positive                      20592 non-null  float64       
 3   probableCases                 9271 non-null   float64       
 4   negative                      13290 non-null  float64       
 5   pending                       2138 non-null   float64       
 6   totalTestResultsSource        20780 non-null  object        
 7   totalTestResults              20614 non-null  float64       
 8   hospitalizedCurrently         17339 non-null  float64       
 9   hospitalizedCumulative        12382 non-null  float64       
 10  inIcuCurrently                11636 non-null  float64       
 11  inIcuCumulative               3789 non-null   float64       
 12  onVentilatorCurrently         9126 non-null   float64       
 13  onVentilatorCumulative        1290 non-null   float64       
 14  recovered                     12003 non-null  float64       
 15  lastUpdateEt                  20164 non-null  object        
 16  dateModified                  20164 non-null  object        
 17  checkTimeEt                   20164 non-null  object        
 18  death                         19930 non-null  float64       
 19  hospitalized                  12382 non-null  float64       
 20  hospitalizedDischarged        3070 non-null   float64       
 21  totalTestsViral               14516 non-null  float64       
 22  positiveTestsViral            8958 non-null   float64       
 23  negativeTestsViral            5024 non-null   float64       
 24  positiveCasesViral            14246 non-null  float64       
 25  deathConfirmed                9469 non-null   float64       
 26  deathProbable                 7593 non-null   float64       
 27  totalTestEncountersViral      5231 non-null   float64       
 28  totalTestsPeopleViral         9181 non-null   float64       
 29  totalTestsAntibody            4789 non-null   float64       
 30  positiveTestsAntibody         3346 non-null   float64       
 31  negativeTestsAntibody         1458 non-null   float64       
 32  totalTestsPeopleAntibody      2200 non-null   float64       
 33  positiveTestsPeopleAntibody   1094 non-null   float64       
 34  negativeTestsPeopleAntibody   972 non-null    float64       
 35  totalTestsPeopleAntigen       999 non-null    float64       
 36  positiveTestsPeopleAntigen    633 non-null    float64       
 37  totalTestsAntigen             3421 non-null   float64       
 38  positiveTestsAntigen          2233 non-null   float64       
 39  fips                          20780 non-null  int64         
 40  positiveIncrease              20780 non-null  int64         
 41  negativeIncrease              20780 non-null  int64         
 42  total                         20780 non-null  int64         
 43  totalTestResultsIncrease      20780 non-null  int64         
 44  posNeg                        20780 non-null  int64         
 45  dataQualityGrade              0 non-null      float64       
 46  deathIncrease                 20780 non-null  int64         
 47  hospitalizedIncrease          20780 non-null  int64         
 48  hash                          20780 non-null  object        
 49  commercialScore               20780 non-null  int64         
 50  negativeRegularScore          20780 non-null  int64         
 51  negativeScore                 20780 non-null  int64         
 52  positiveScore                 20780 non-null  int64         
 53  score                         20780 non-null  int64         
 54  grade                         0 non-null      float64       
```

```
dtypes: datetime64[ns](1), float64(35), int64(13), object(6)
memory usage: 8.7+ MB
```

In [13]:

```
df.head()
```

Out[13]:

	date	state	positive	probableCases	negative	pending	totalTestResultsSource	totalTe
0	2021-03-07	AK	56886.0	NaN	NaN	NaN	totalTestsViral	
1	2021-03-07	AL	499819.0	107742.0	1931711.0	NaN	totalTestsPeopleViral	
2	2021-03-07	AR	324818.0	69092.0	2480716.0	NaN	totalTestsViral	
3	2021-03-07	AS	0.0	NaN	2140.0	NaN	totalTestsViral	
4	2021-03-07	AZ	826454.0	56519.0	3073010.0	NaN	totalTestsViral	

5 rows × 55 columns

Replacing the NaN by -1

In [14]:

```
df.fillna(value=-1, inplace=True)
df.head()
```

Out[14]:

	date	state	positive	probableCases	negative	pending	totalTestResultsSource	totalTe
0	2021-03-07	AK	56886.0	-1.0	-1.0	-1.0	totalTestsViral	
1	2021-03-07	AL	499819.0	107742.0	1931711.0	-1.0	totalTestsPeopleViral	
2	2021-03-07	AR	324818.0	69092.0	2480716.0	-1.0	totalTestsViral	
3	2021-03-07	AS	0.0	-1.0	2140.0	-1.0	totalTestsViral	
4	2021-03-07	AZ	826454.0	56519.0	3073010.0	-1.0	totalTestsViral	

5 rows × 55 columns

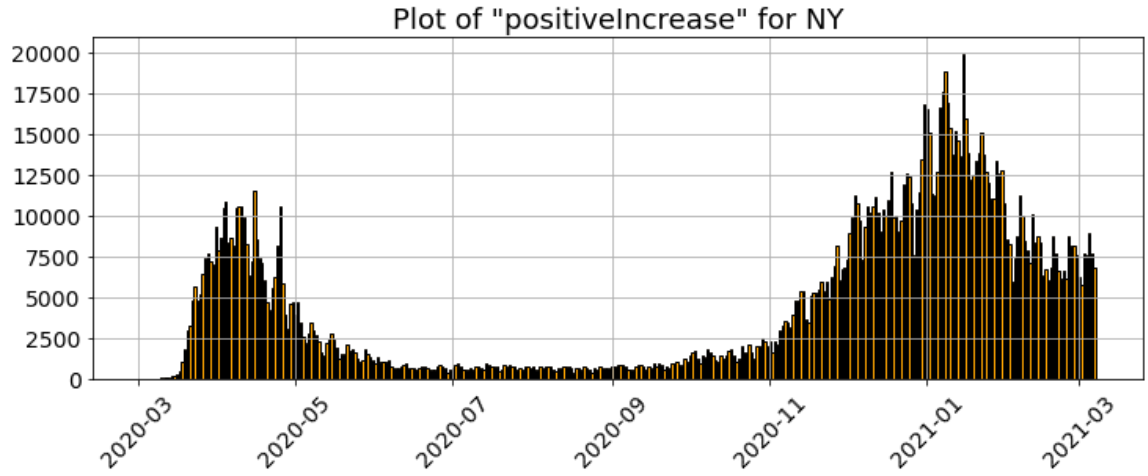
Function to plot a bar chart of the given variable/state

In [15]:

```
def plot_var(var='positiveIncrease',
             state='NY'):
    """
    Plots a bar chart of the given variable over the date range
    """
    assert type(var)==str, "Expected string as the variable name"
    assert type(state)==str, "Expected string as the state name"

    y = df[df['state']==state][var]
    x = df[df['state']==state]['date']
    plt.figure(figsize=(12,4))
    plt.title("Plot of \"{}\" for {}".format(var,state),fontsize=18)
    plt.bar(x=x,height=y,edgecolor='k',color='orange')
    plt.grid(True)
    plt.xticks(fontsize=14,rotation=45)
    plt.yticks(fontsize=14)
    plt.show()
```

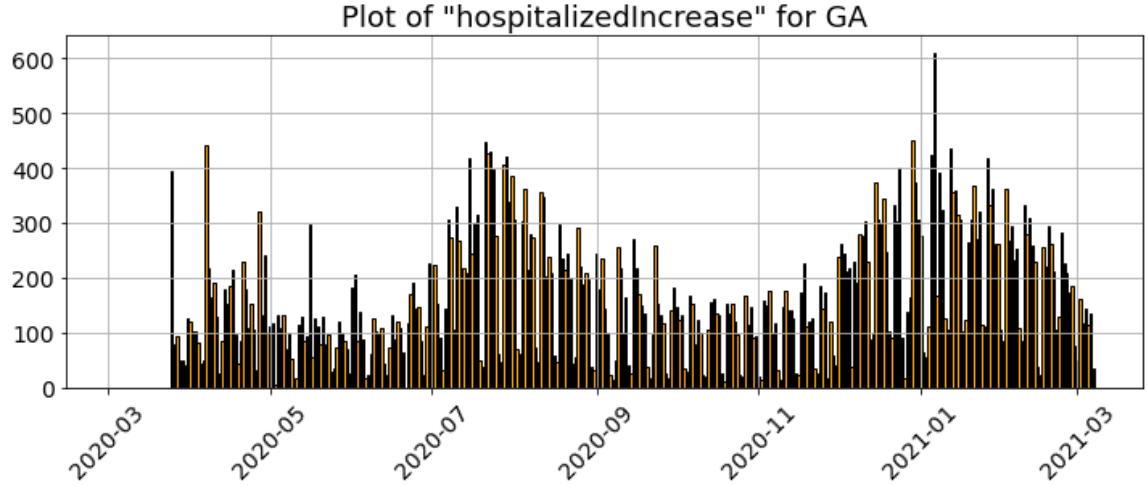
In [16]:

```
plot_var()
```



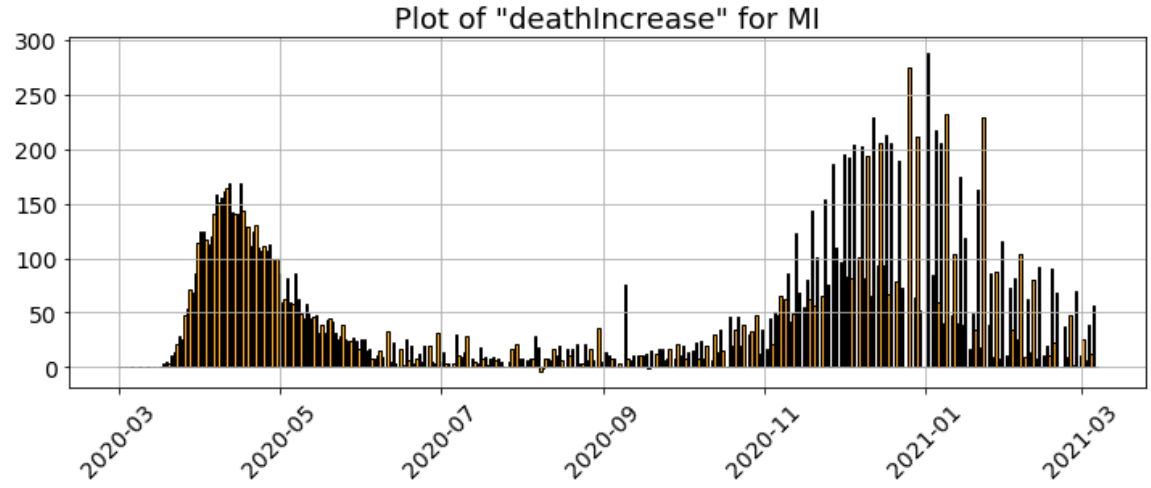
In [17]:

```
plot_var('hospitalizedIncrease','GA')
```



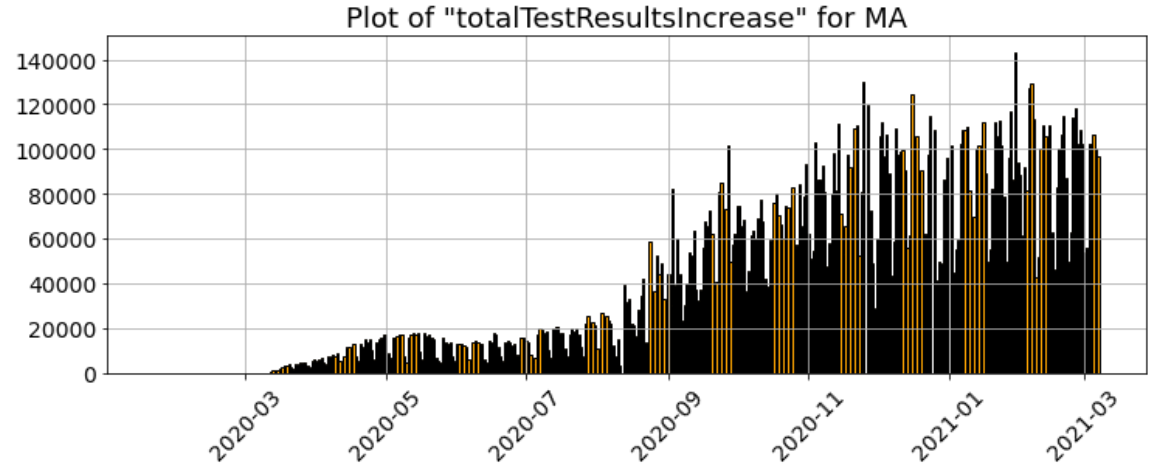
In [18]:

```
plot_var('deathIncrease','MI')
```



In [19]:

```
plot_var('totalTestResultsIncrease','MA')
```



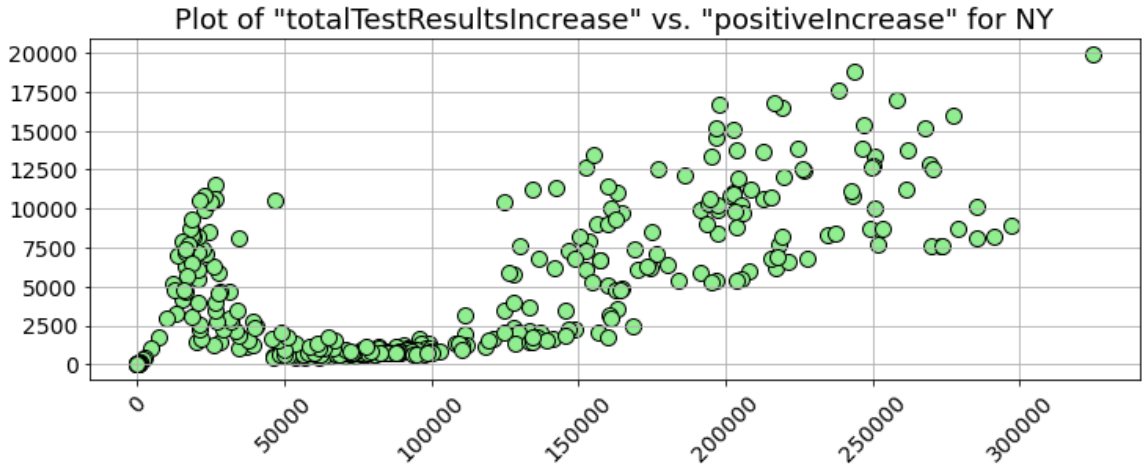
Function to create scatter plot of two variables for a given state

In [20]:

```
def plot_xy(varx='totalTestResultsIncrease',
            vary='positiveIncrease',
            state='NY'):
    """
    Plots a bar chart of the given variable over the date range
    """
    assert type(varx)==str, "Expected string as the variable x name"
    assert type(vary)==str, "Expected string as the variable y name"

    y = df[df['state']==state][vary]
    x = df[df['state']==state][varx]
    if (x.nunique()!=1) and (y.nunique()!=1):
        plt.figure(figsize=(12,4))
        plt.title("Plot of \"{}\" vs. \"{}\" for {}".format(varx,vary,state),fontsize=1
8)
        plt.scatter(x=x,y=y,edgecolor='k',color='lightgreen',s=100)
        plt.grid(True)
        plt.xticks(fontsize=14,rotation=45)
        plt.yticks(fontsize=14)
        plt.show()
    else:
        print("Some of the data unavailable for a scatter plot. Sorry!")
```

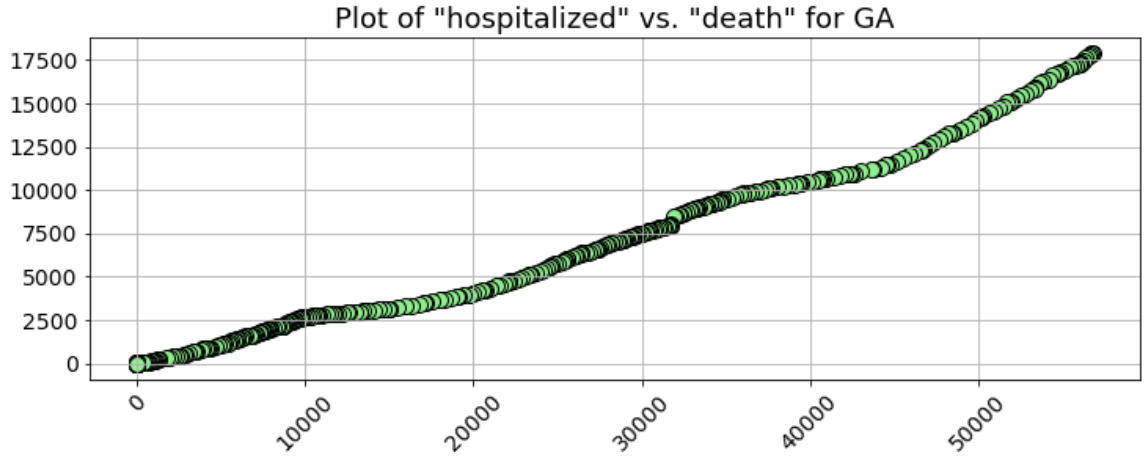
In [21]:

```
plot_xy(state='NY')
```



In [22]:

```
plot_xy('hospitalized','death','GA')
```



In [23]:

```
plot_xy('hospitalized','death','CA')
```

```
Some of the data unavailable for a scatter plot. Sorry!
```

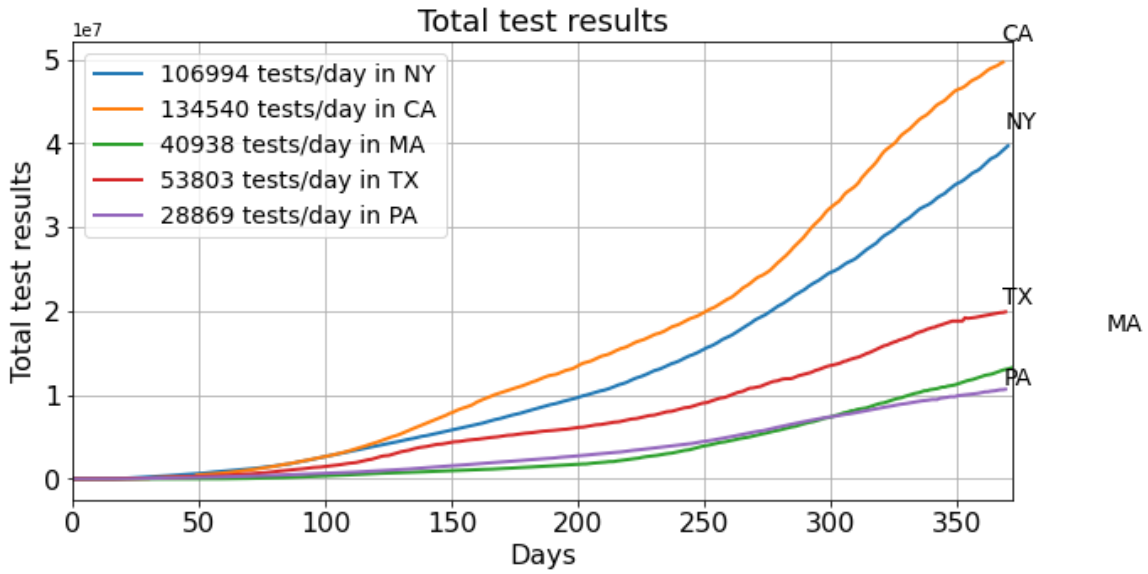
Testing tracker function

In [28]:

```
def plotTesting(lst_states=['NY','CA','MA','TX','PA']):
    """
    Plots the cumulative testing done by the given list of states
    """
    legends = []
    plt.figure(figsize=(10,5))
    plt.title("Total test results",fontsize=18)
    for s in lst_states:
        data = np.array(df[df['state']==s]['totalTestResults'])[-1::-1]
        slope = int((data[-1]-data[0])/len(data))
        plt.plot(data,linewidth=2)
        plt.text(x=len(data)-2,y=data[-1]*1.05,s=s,fontsize=14)
        legends.append(str(slope)+" tests/day in " + s)
    plt.legend(legends,fontsize=14)
    plt.grid(True)
    plt.xlim(0,len(data)+2)
    plt.xticks(fontsize=16)
    plt.yticks(fontsize=16)
    plt.xlabel("Days",fontsize=16)
    plt.ylabel("Total test results",fontsize=16)
    plt.show()
```

In [29]:

```
plotTesting()
```



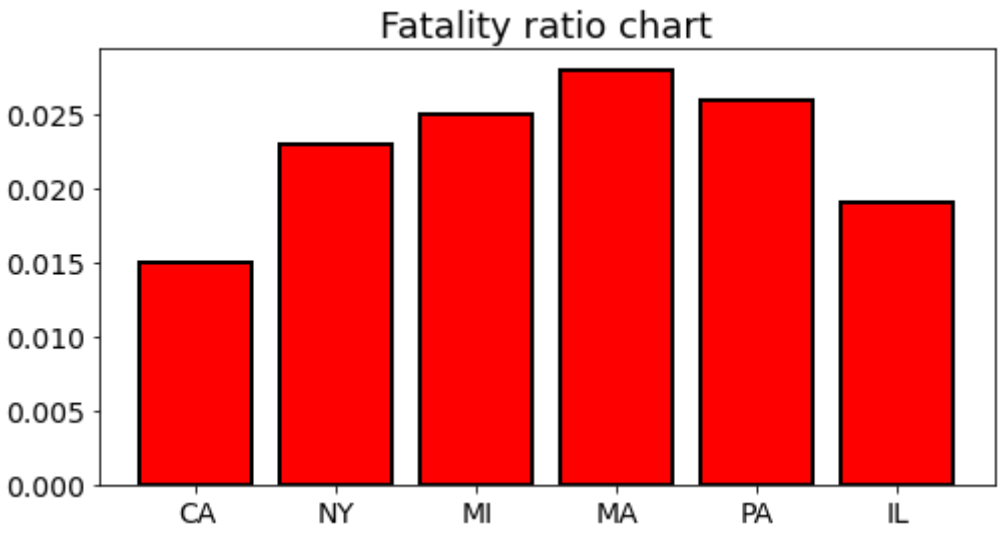
Fatality ratio chart

We will plot the chart with six states,

- California
- New York
- Michigan
- Massachusetts
- Pennsylvania
- Illinois

In [30]:

```
states = ['CA','NY','MI','MA','PA','IL']
fr,x = [],[]
for s in states:
    data = fatality_ratio(s)
    if data!=-1:
        fr.append(data)
        x.append(s)

plt.figure(figsize=(8,4))
plt.title("Fatality ratio chart",fontsize=18)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.bar(x=x,height=fr,color='red',
        edgecolor='k',linewidth=2)
plt.show()
```



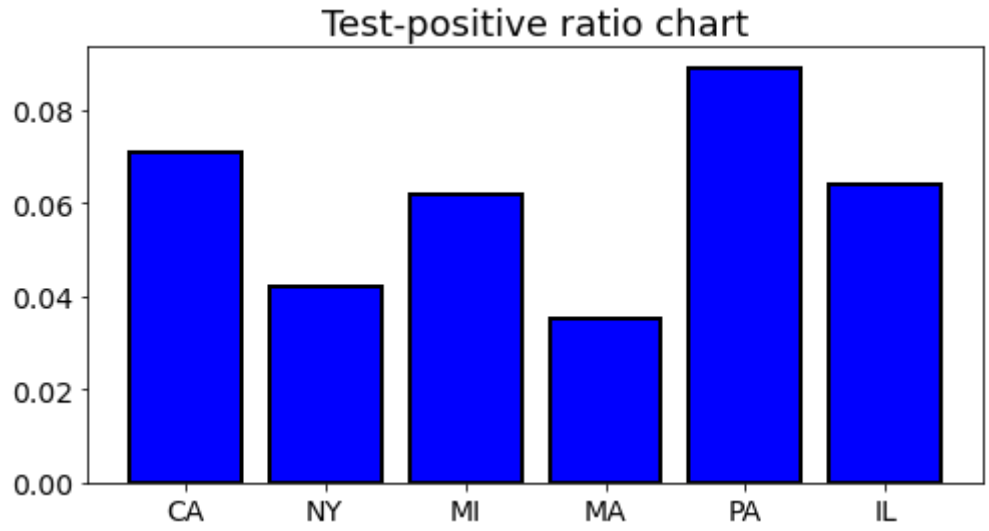
Test-positive ratio chart

In [40]:

```
def positiveTest_ratio(state='NY'):
    """
    Computes the test-positive ratio for the given state
    Test-positive ratio is the ratio of total positive cases to total number of tests
    """

    date = df.iloc[0]['date']
    try:
        p = float(df[(df['state']==state) & (df['date']==date)]['positive'])
        t = float(df[(df['state']==state) & (df['date']==date)]['totalTestResults'])
    except:
        print("Could not retrieve the necessary information")
        return -1

    if (p!=-1.0) and (t!=-1.0) and (t!=0):
        return round(p/t,3)
    else:
        return -1
```

In [41]:

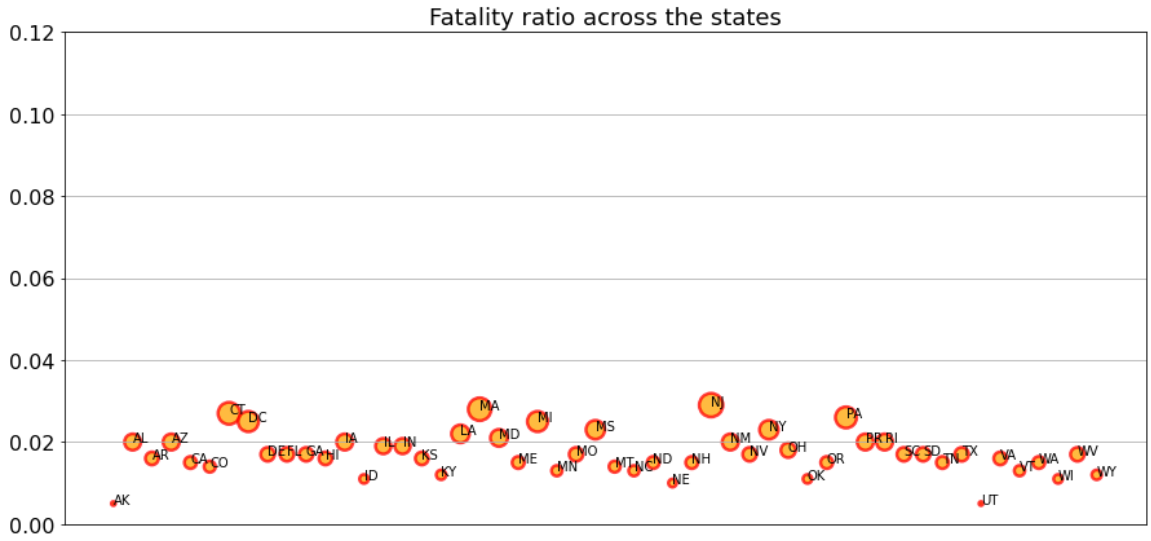
```
states = ['CA','NY','MI','MA','PA','IL']
tp,x = [],[]
for s in states:
    data = positiveTest_ratio(s)
    if data!=-1:
        tp.append(data)
        x.append(s)
plt.figure(figsize=(8,4))
plt.title("Test-positive ratio chart",fontsize=18)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.bar(x=x,height=tp,color='blue',
        edgecolor='k',linewidth=2)
plt.show()
```



Bubble charts...

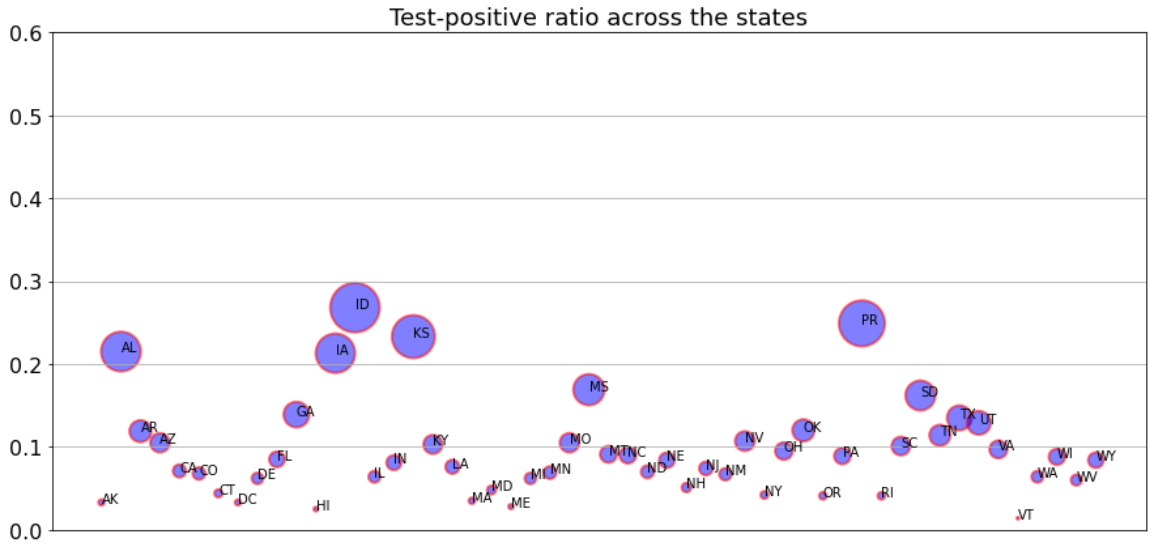
In [44]:

```
states = list(df['state'].unique())
for s in ['AS','GU','MP','PU','VI']:
    try:
        states.remove(s)
    except:
        pass

fr,x = [],[]
for s in states:
    data = fatality_ratio(s)
    if data!=-1:
        fr.append(data)
        x.append(s)
fr = np.array(fr)
plt.figure(figsize=(15,7))
plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom=False,      # ticks along the bottom edge are off
    top=False,         # ticks along the top edge are off
    labelbottom=False)
plt.title("Fatality ratio across the states",fontsize=18)
plt.scatter(x=x,y=fr,
            s=4e5*fr**2,
            color='orange',edgecolor='red',alpha=0.75,linewidth=2.5)
#plt.xticks(rotation=45,fontsize=12)
for i,s in enumerate(x):
    plt.annotate(s=s,xy=(x[i],fr[i]))
plt.ylim(0,0.12)
plt.yticks(fontsize=16)
plt.grid(True,axis='y')
plt.show()
```



In [45]:

```
states = list(df['state'].unique())
for s in ['AS','GU','MP','PU','VI']:
    try:
        states.remove(s)
    except:
        pass

tp,x = [],[]
for s in states:
    data = positiveTest_ratio(s)
    if data!=-1:
        tp.append(data)
        x.append(s)
tp = np.array(tp)
plt.figure(figsize=(15,7))
plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom=False,      # ticks along the bottom edge are off
    top=False,         # ticks along the top edge are off
    labelbottom=False)
plt.title("Test-positive ratio across the states",fontsize=18)
plt.scatter(x=x,y=tp,
            s=2e4*tp**2,
            color='blue',edgecolor='red',alpha=0.5,linewidth=2)
plt.xticks(rotation=90,fontsize=12)
for i,s in enumerate(x):
    plt.annotate(s=s,xy=(x[i],tp[i]))
plt.ylim(0,0.6)
plt.yticks(fontsize=16)
plt.grid(True,axis='y')
plt.show()
```



Plot for a few states

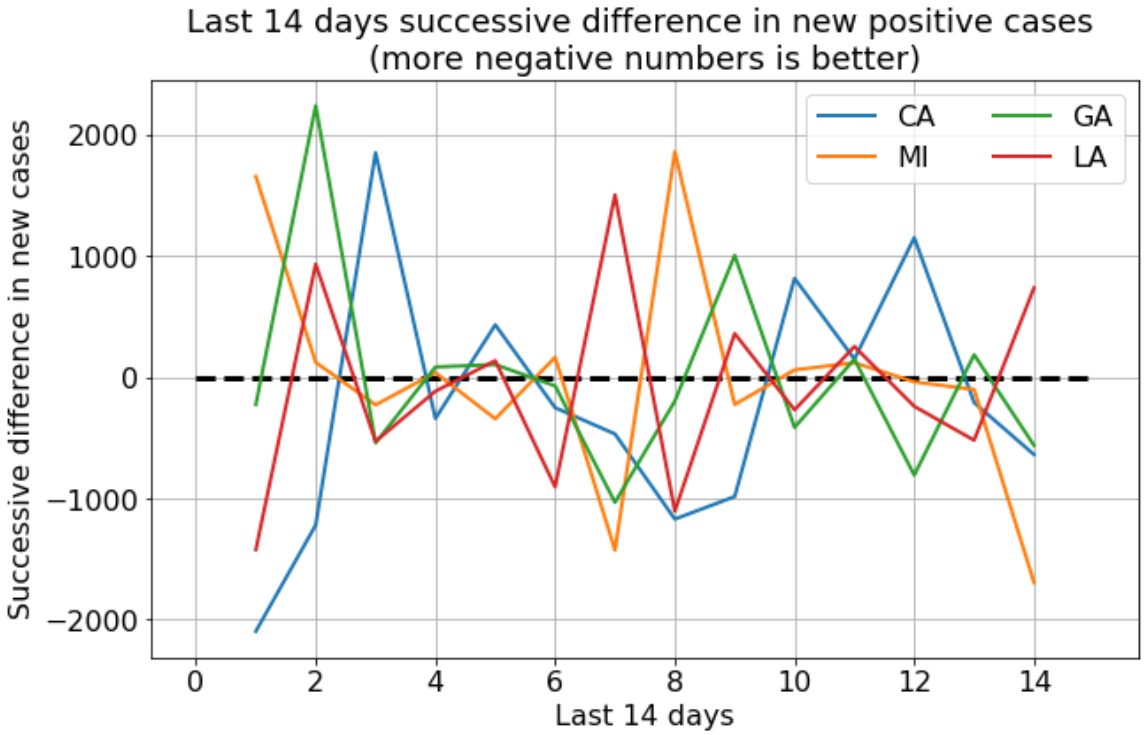
We will note that **no state, so far, has shown a consistent decrease of new cases for the last 14 days**!

In [47]:

```
def caseCountsdecrease(days=14,state='NY'):
    """
    Determines whether the given state has a decreasing case counts for given number of
days
    Arguments:
            days: Number of days to go back
            state: Name of the state (a string)
    Returns:
            A tuple containing the successive difference vector (of new cases) and
            the number of negative quantities in that vector. When all the quantities a
re negative,
            the state has shown consistent decrease in new cases for the given number o
f days.
    """
    positiveIncrease = np.array(df[df['state']==state]['positiveIncrease'][:days+1])[-1
::-1]
    diff = np.diff(positiveIncrease)
    countofNeg = np.sum(diff <= 0, axis=0)
    return (countofNeg, diff)
```

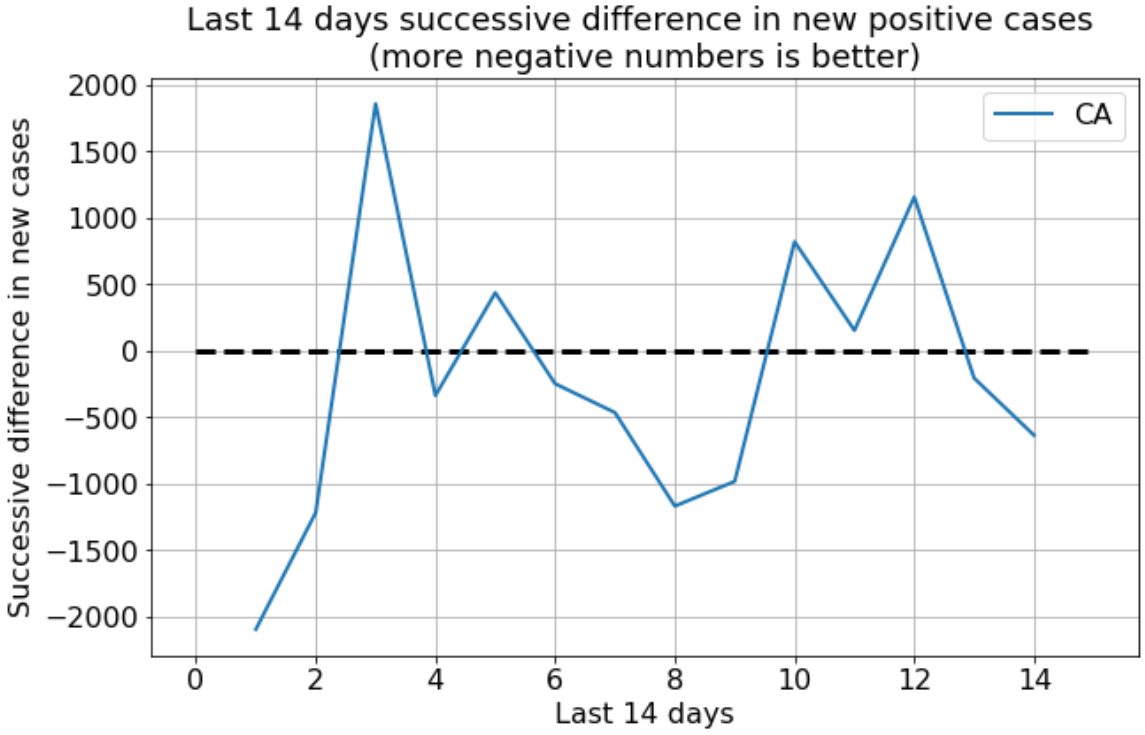
In [48]:

```
states = ['CA','MI','GA','LA']
cd = []
x = np.arange(1,15,1)
plt.figure(figsize=(10,6))
plt.title("Last 14 days successive difference in new positive cases \n(more negative nu
mbers is better)",
          fontsize=18)
for s in states:
    _,data = caseCountsdecrease(days=14,state=s)
    plt.plot(x,data,linewidth=2)
plt.legend(states,fontsize=16,ncol=2)
plt.grid(True)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xlabel("Last 14 days",fontsize=16)
plt.ylabel("Successive difference in new cases",fontsize=16)
plt.hlines(y=0,xmin=0,xmax=15,linestyles='--',lw=3)
plt.show()
```


Last 14 days successive difference in new positive cases
(more negative numbers is better)
Successive difference in new cases
2000
1000
0
−1000
−2000
0
2
4
6
8
10
12
14
Last 14 days
CA
GA
MI
LA

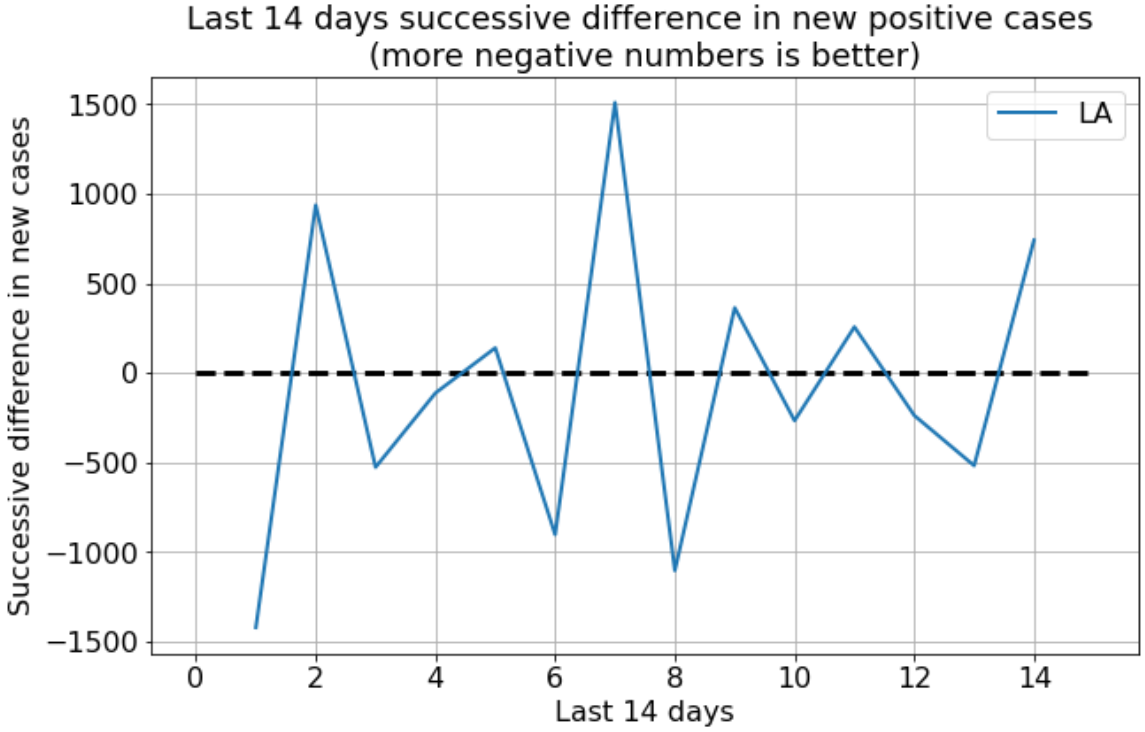
In [49]:

```
states = ['CA']
cd = []
x = np.arange(1,15,1)
plt.figure(figsize=(10,6))
plt.title("Last 14 days successive difference in new positive cases \n(more negative nu
mbers is better)",
          fontsize=18)
for s in states:
    _,data = caseCountsdecrease(days=14,state=s)
    plt.plot(x,data,linewidth=2)
plt.legend(states,fontsize=16,ncol=2)
plt.grid(True)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xlabel("Last 14 days",fontsize=16)
plt.ylabel("Successive difference in new cases",fontsize=16)
plt.hlines(y=0,xmin=0,xmax=15,linestyles='--',lw=3)
plt.show()
```



In [50]:

```
states = ['LA']
cd = []
x = np.arange(1,15,1)
plt.figure(figsize=(10,6))
plt.title("Last 14 days successive difference in new positive cases \n(more negative nu
mbers is better)",
          fontsize=18)
for s in states:
    _,data = caseCountsdecrease(days=14,state=s)
    plt.plot(x,data,linewidth=2)
plt.legend(states,fontsize=16,ncol=2)
plt.grid(True)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)
plt.xlabel("Last 14 days",fontsize=16)
plt.ylabel("Successive difference in new cases",fontsize=16)
plt.hlines(y=0,xmin=0,xmax=15,linestyles='--',lw=3)
plt.show()
```



In [ ]: