TOPIC: MASS SHOOTING IN UNITED STATES

IMPORTING LIBRARIES

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import metrics
import warnings
warnings.filterwarnings('ignore')
```

READING THE DATASET

In [3]:

```
data = pd.read_csv(r"D:\adypu\shootings_2022.csv")
```

In [4]:

```
data.head(5) #For reading the starting few rows
```

Out[4]:

	Date	State	Dead	Injured	Total	Description
0	03-03-2022	Nevada	1	6	7	A dispute between neighbors led to a shooting ...
1	02-03-2022	Maryland	1	3	4	Four men were shot in the Walbrook neighborhoo...
2	28-02-2022	California	5	0	5	A man fatally shot himself, his three children...
3	27-02-2022	California	0	4	4	An argument between two groups of people at a ...
4	27-02-2022	Louisiana	0	4	4	An argument at a business in the southeast par...

In [5]:

```
data.tail(5) #For reading the last few rows
```

Out[5]:

	Date	State	Dead	Injured	Total	Description
76	01-01-2022	Georgia	1	3	4	After officers were dispatched to respond to a...
77	01-01-2022	Wisconsin	1	3	4	A man was killed and three others wounded in a...
78	01-01-2022	Indiana	0	4	4	Four people were wounded at a New Years party ...
79	01-01-2022	Colorado	2	2	4	Two adults were killed and two wounded in an e...
80	01-01-2022	Missouri	0	4	4	Four adults were wounded in the early morning ...

SUMMARIZING THE DATA

In [6]:

```
data.shape # foe finding no. of rows and columns
```

Out[6]:

```
(81, 6)
```

In [7]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81 entries, 0 to 80
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Date         81 non-null     object
 1   State        81 non-null     object
 2   Dead         81 non-null     int64 
 3   Injured      81 non-null     int64 
 4   Total        81 non-null     int64 
 5   Description  81 non-null     object
dtypes: int64(3), object(3)
memory usage: 3.9+ KB
```

In [8]:

```
data.describe() # for statistical summary of data
```

Out[8]:

	Dead	Injured	Total
count	81.000000	81.000000	81.000000
mean	1.135802	3.802469	4.938272
std	1.339338	2.288120	2.063648
min	0.000000	0.000000	4.000000
25%	0.000000	3.000000	4.000000
50%	1.000000	4.000000	4.000000
75%	1.000000	4.000000	5.000000
max	6.000000	14.000000	16.000000

In [9]:

```
#Checking null values
data.isnull().sum()
```

Out[9]:

```
Date           0
State          0
Dead           0
Injured        0
Total          0
Description    0
dtype: int64
```

DATA VISUALIZATION

In [10]:

```
#finding the sum of victims state wise
total_victims_state = data.groupby('State').sum()
print('Total Victims = ' ,total_victims_state['Total'].sum())
total_victims_state.sort_values(by = 'Total', ascending = False)
```

```
Total Victims =  400
```

Out[10]:

	Dead	Injured	Total
State			
California	15	38	53
Texas	15	22	37
Georgia	9	25	34
Missouri	5	23	28

	Dead	Injured	Total
State			
Louisiana	2	20	22
Wisconsin	8	13	21
Nevada	2	19	21
Florida	3	15	18
Illinois	3	15	18
Oregon	2	14	16
Tennessee	5	8	13
North Carolina	2	10	12
Arizona	2	9	11
Alabama	1	10	11
Mississippi	2	7	9
Pennsylvania	1	8	9
South Carolina	1	8	9
New Mexico	1	7	8
Maryland	4	4	8
Colorado	4	4	8
Virginia	1	4	5
Washington D.C.	1	4	5
Minnesota	1	3	4
New York	0	4	4
Indiana	0	4	4
Arkansas	1	3	4
Washington	1	3	4
Nebraska	0	4	4

In [11]:

```
total_victims_state.sort_values(by = 'Total', ascending = False).head()
```

Out[11]:

	Dead	Injured	Total
State			
California	15	38	53
Texas	15	22	37
Georgia	9	25	34
Missouri	5	23	28
Louisiana	2	20	22

In [12]:

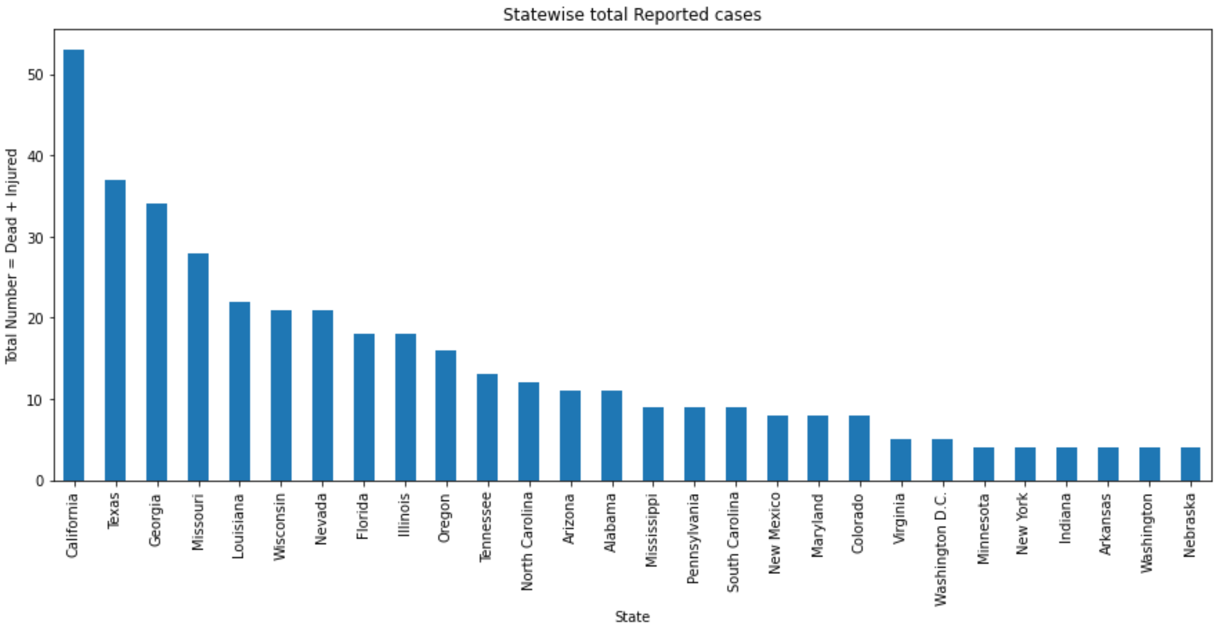
```
total_victims_state.sort_values(by = 'Total', ascending = False).tail()
```

Out[12]:

	Dead	Injured	Total
State			
New York	0	4	4
Indiana	0	4	4
Arkansas	1	3	4
Washington	1	3	4
Nebraska	0	4	4

In [13]:

```
# Plotting bar graph for Statewise total reported cases
plt.subplots(figsize = (15, 6) )
cr = total_victims_state['Total'].sort_values(ascending = False)
ax = cr.plot.bar()
ax.set_xlabel('State')  # labelling x-axis as State
ax.set_ylabel('Total Number = Dead + Injured') # labelling y-axis as total which rep
ax.set_title('Statewise total Reported cases') # giving title to our chart
plt.show()
```



In [14]:

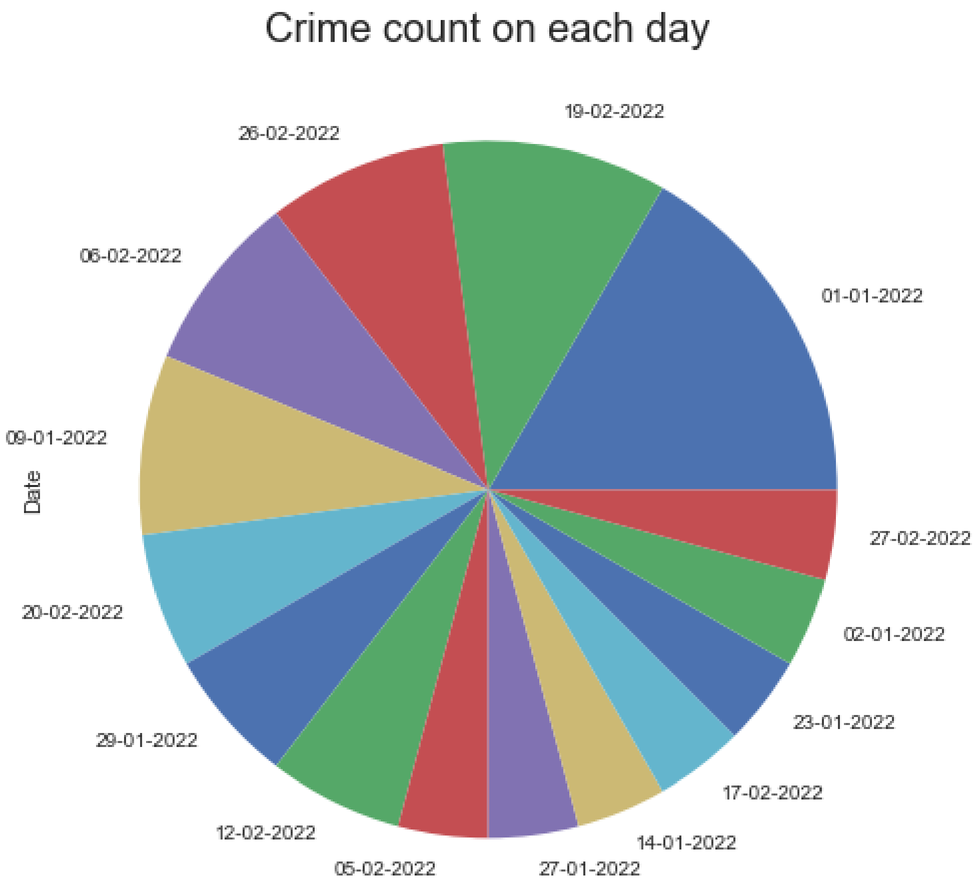
```
#finding out victims specifically for calafornia state
california_victims = data[data['State'] == 'California']
california_victims.head()
```

Out[14]:

	Date	State	Dead	Injured	Total	Description
2	28-02-2022	California	5	0	5	A man fatally shot himself, his three children...
3	27-02-2022	California	0	4	4	An argument between two groups of people at a ...
21	19-02-2022	California	0	7	7	A man shot seven people outside a bar in the e...
28	12-02-2022	California	0	4	4	Outside an Italian restaurant hosting a party ...
33	06-02-2022	California	2	2	4	A chance encounter between two rival gangs at ...

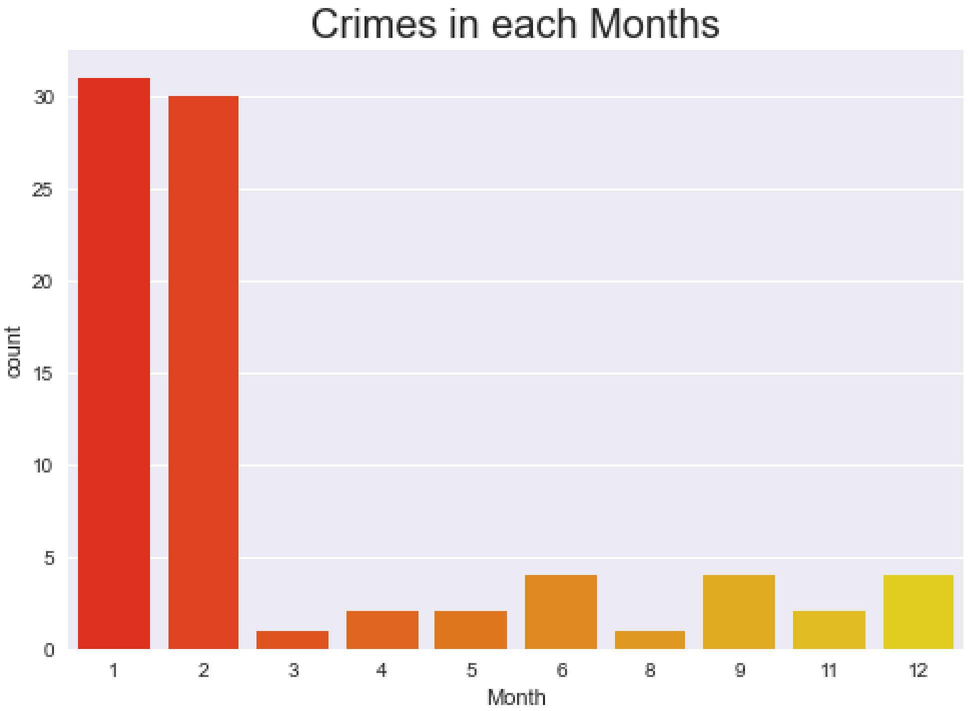
In [15]:

```
#plotiing pie plot for finding each day crime
plt.style.use('seaborn')
data['Date'].value_counts().head(15).plot.pie(figsize = (15, 8))
plt.title('Crime count on each day',fontsize = 20)
plt.xticks(rotation = 90)
plt.show()
```



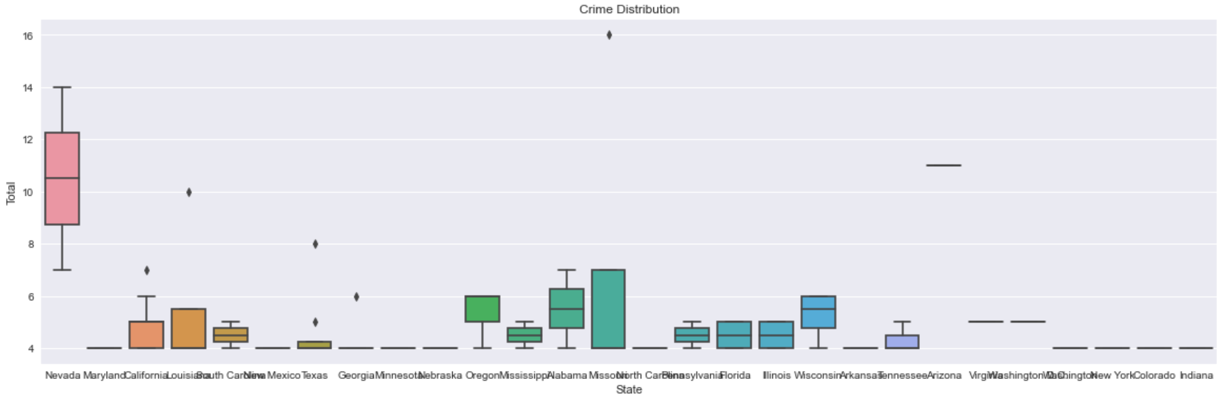
In [16]:

```
#finding the crimes of each month
data['Date'] = pd.to_datetime(data['Date'])
data['Month'] = data['Date'].dt.month
sns.countplot(data['Month'], palette = 'autumn',)
plt.title('Crimes in each Months', fontsize = 20)
plt.show()
```



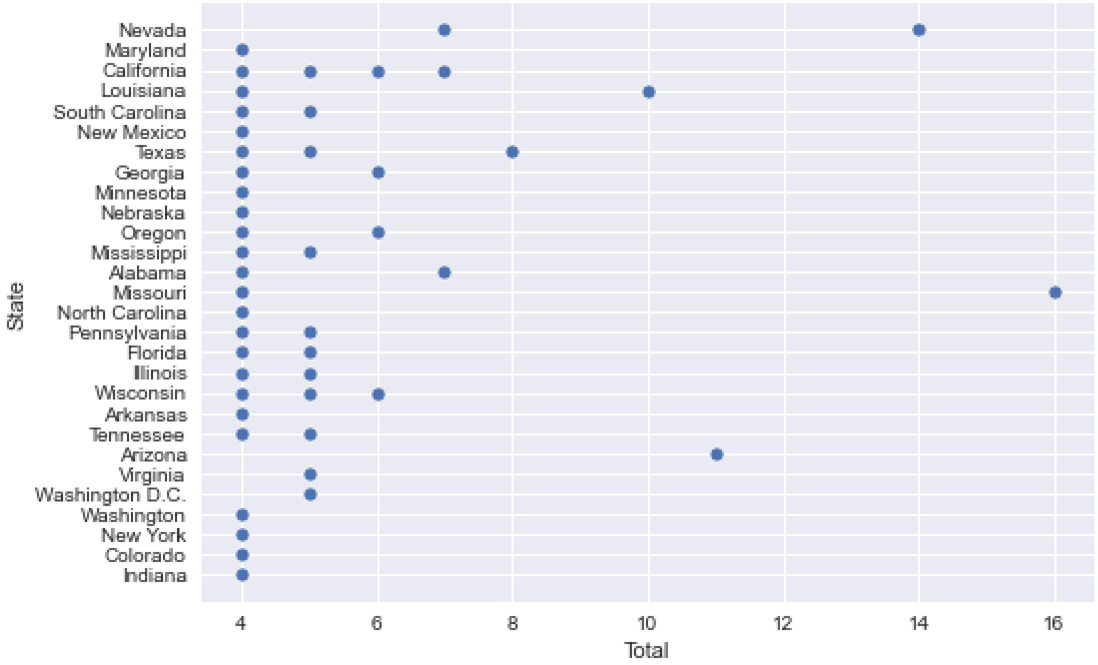
In [17]:
```
#box plot for crime in each state
plt.figure(figsize=(20,6))
ax = sns.boxplot(x='State',y='Total',data=data)
ax.set_title("Crime Distribution")
```

Out[17]:
```
Text(0.5, 1.0, 'Crime Distribution')
```



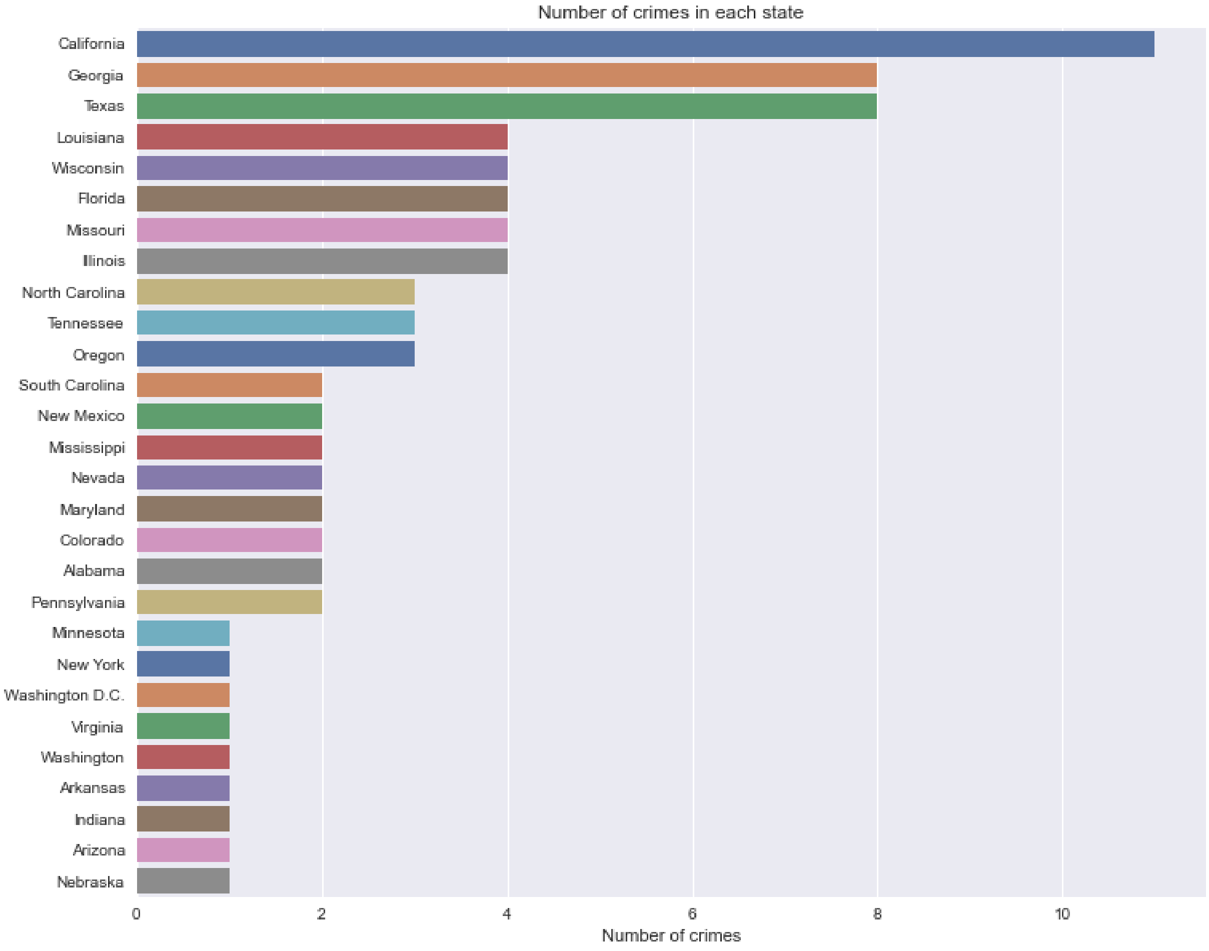
In [18]:
```
#SCATTERPLOT
sns.scatterplot(x="Total",y="State",data=data)
```

Out[18]:
```
<AxesSubplot:xlabel='Total', ylabel='State'>
```



In [25]:

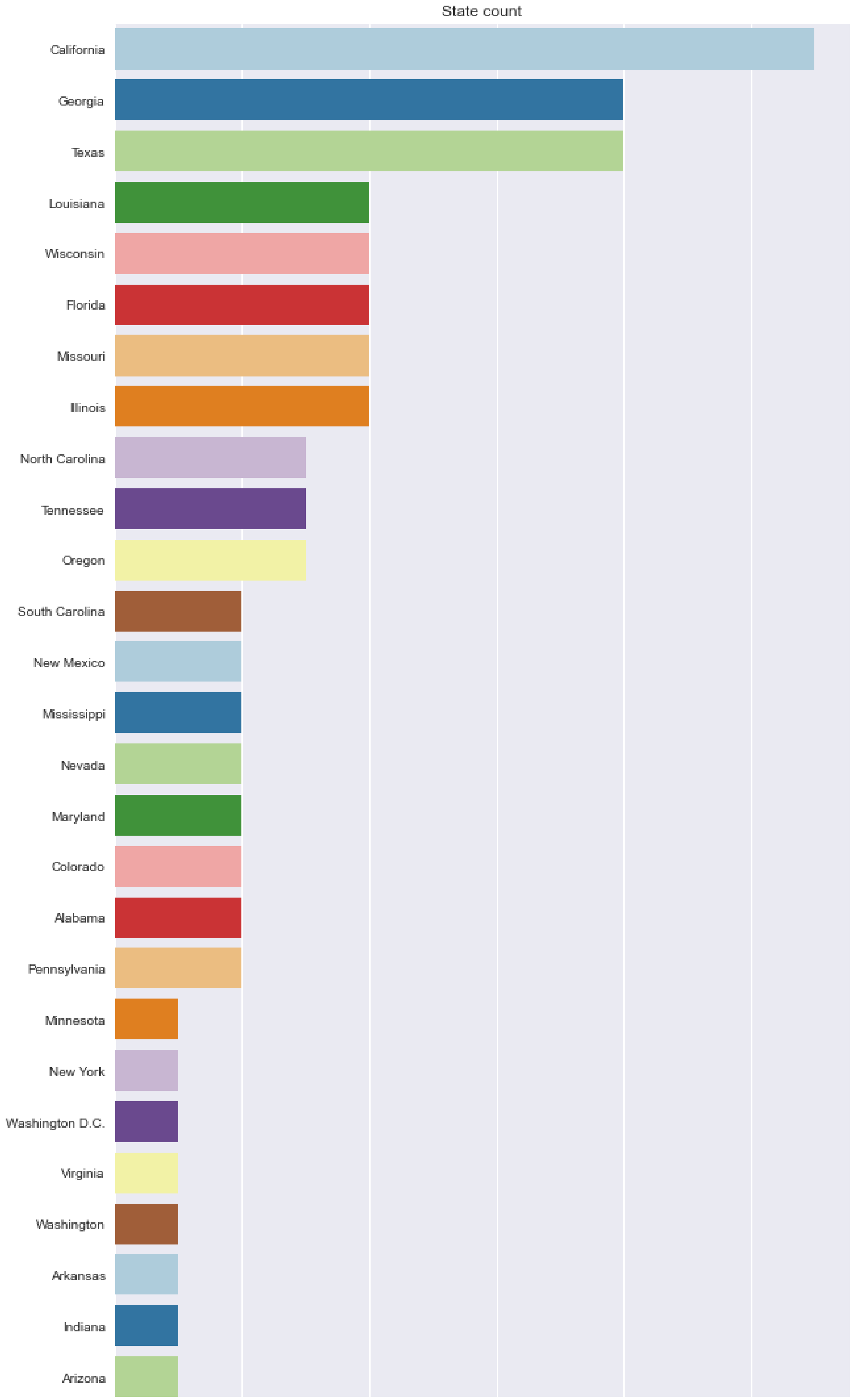
```
plt.figure(figsize = (12,10))
state = data['State'].value_counts()
sns.barplot(x = state, y= state.index,  palette= 'deep')
plt.title('Number of crimes in each state')
plt.xlabel('Number of crimes')
plt.show()
```



In [27]:

```
plt.figure(figsize = (10,20))
t = data['State'].value_counts()
sns.barplot(y = t.index ,x = t, palette = 'Paired')
```

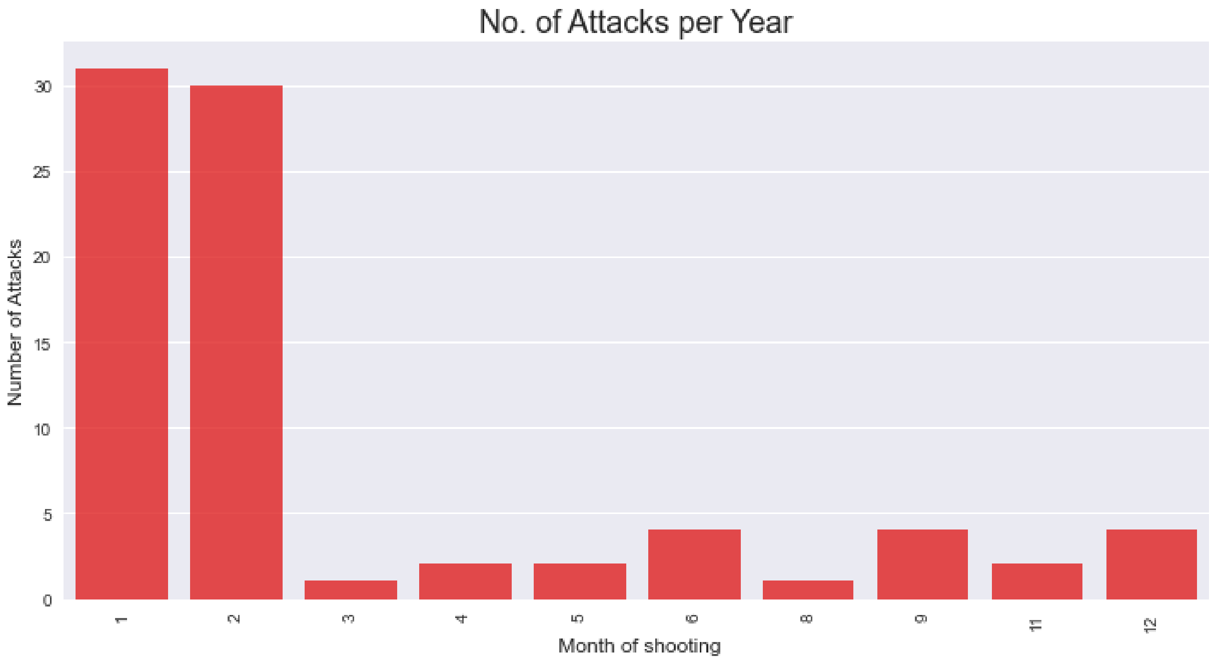
```
plt.title('State count')
plt.show()
```



Nebraska
0
2
4
6
8
10
State

In [37]:

```
# Total affectees over the time.
data['Year'] = data['Month']
cnt_srs = data['Year'].value_counts()
plt.figure(figsize=(12,6))
sns.barplot(cnt_srs.index, cnt_srs.values, alpha=0.8,color= 'red')
plt.xticks(rotation='vertical')
plt.xlabel(' Month of shooting', fontsize=12)
plt.ylabel('Number of Attacks', fontsize=12)
plt.title('No. of Attacks per Year', fontsize=18)
plt.show()
```



Conclusion:

1. California was the most affected state and New York,Indiana,Washington,Nebraska was the least
2. On 1st january most number of crimes took place.
3. In month of January and February most number of crimes took place

In [ ]: