Import libraries

In [29]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:

```
df = pd.read_csv('state_crime.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	State	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary	Data.
0	Alabama	1960	3266740	1035.4	355.9	
1	Alabama	1961	3302000	985.5	339.3	
2	Alabama	1962	3358000	1067.0	349.1	
3	Alabama	1963	3347000	1150.9	376.9	
4	Alabama	1964	3407000	1358.7	466.6	

5 rows × 21 columns

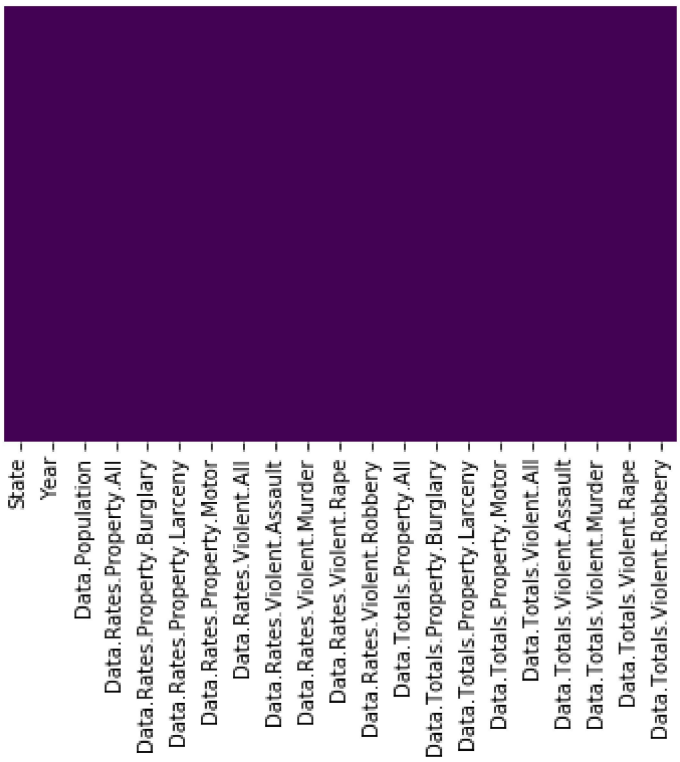
Missing Values

In [4]:

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')

print("{} missing values".format(df.isnull().values.sum()))
```

```
0 missing values
```



EDA

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2751 entries, 0 to 2750
Data columns (total 21 columns):
State                           2751 non-null object
Year                            2751 non-null int64
Data.Population                 2751 non-null int64
Data.Rates.Property.All         2751 non-null float64
Data.Rates.Property.Burglary    2751 non-null float64
Data.Rates.Property.Larceny     2751 non-null float64
Data.Rates.Property.Motor       2751 non-null float64
Data.Rates.Violent.All          2751 non-null float64
Data.Rates.Violent.Assault      2751 non-null float64
Data.Rates.Violent.Murder       2751 non-null float64
Data.Rates.Violent.Rape         2751 non-null float64
Data.Rates.Violent.Robbery      2751 non-null float64
Data.Totals.Property.All        2751 non-null int64
Data.Totals.Property.Burglary   2751 non-null int64
Data.Totals.Property.Larceny    2751 non-null int64
Data.Totals.Property.Motor      2751 non-null int64
Data.Totals.Violent.All         2751 non-null int64
Data.Totals.Violent.Assault     2751 non-null int64
Data.Totals.Violent.Murder      2751 non-null int64
Data.Totals.Violent.Rape        2751 non-null int64
Data.Totals.Violent.Robbery     2751 non-null int64
dtypes: float64(9), int64(11), object(1)
memory usage: 451.4+ KB
```

In [11]:

```
df.describe().T
```

Out[11]:

	count	mean	std	min	25%	
Year	2751.0	1.986044e+03	1.527932e+01	1960.0	1973.00	1
Data.Population	2751.0	9.349570e+06	3.368126e+07	226167.0	1208000.00	3282
Data.Rates.Property.All	2751.0	3.686539e+03	1.427900e+03	573.1	2613.40	3
Data.Rates.Property.Burglary	2751.0	9.312073e+02	4.424760e+02	182.6	592.10	
Data.Rates.Property.Larceny	2751.0	2.395550e+03	9.145975e+02	293.3	1745.35	2
Data.Rates.Property.Motor	2751.0	3.597821e+02	2.275578e+02	48.3	195.10	
Data.Rates.Violent.All	2751.0	4.003703e+02	2.989882e+02	9.5	204.95	
Data.Rates.Violent.Assault	2751.0	2.362985e+02	1.641829e+02	3.6	116.65	
Data.Rates.Violent.Murder	2751.0	6.691821e+00	6.127160e+00	0.2	3.20	
Data.Rates.Violent.Rape	2751.0	2.832156e+01	1.560217e+01	0.8	17.10	
Data.Rates.Violent.Robbery	2751.0	1.290566e+02	1.496025e+02	1.9	41.80	
Data.Totals.Property.All	2751.0	3.633906e+05	1.351616e+06	3147.0	39845.50	108
Data.Totals.Property.Burglary	2751.0	9.402040e+04	3.510220e+05	751.0	9850.00	28
Data.Totals.Property.Larceny	2751.0	2.293203e+05	8.565842e+05	1489.0	25898.00	70
Data.Totals.Property.Motor	2751.0	4.005010e+04	1.519735e+05	334.0	3199.00	9
Data.Totals.Violent.All	2751.0	4.571977e+04	1.768145e+05	37.0	3077.50	10
Data.Totals.Violent.Assault	2751.0	2.613832e+04	1.027906e+05	14.0	1879.00	6
Data.Totals.Violent.Murder	2751.0	6.708313e+02	2.470164e+03	1.0	47.00	
Data.Totals.Violent.Rape	2751.0	2.785149e+03	1.069350e+04	6.0	292.00	
Data.Totals.Violent.Robbery	2751.0	1.612547e+04	6.223056e+04	8.0	788.50	3

In [17]:

```
df['State'].unique()
```

Out[17]:

```
array(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Delaware', 'District of Columbia',
       'Florida', 'Georgia', 'Hawaii', 'Idaho', 'Illinois', 'Indiana',
       'Iowa', 'Kansas', 'Kentucky', 'Louisiana', 'Maine', 'Maryland',
       'Massachusetts', 'Michigan', 'Minnesota', 'Mississippi',
       'Missouri', 'Montana', 'Nebraska', 'Nevada', 'New Hampshire',
       'New Jersey', 'New Mexico', 'New York', 'North Carolina',
       'North Dakota', 'Ohio', 'Oklahoma', 'Oregon', 'Pennsylvania',
       'Rhode Island', 'South Carolina', 'South Dakota', 'Tennessee',
       'Texas', 'Utah', 'Vermont', 'Virginia', 'Washington',
       'West Virginia', 'Wisconsin', 'Wyoming', 'United States'],
      dtype=object)
```

In [30]:

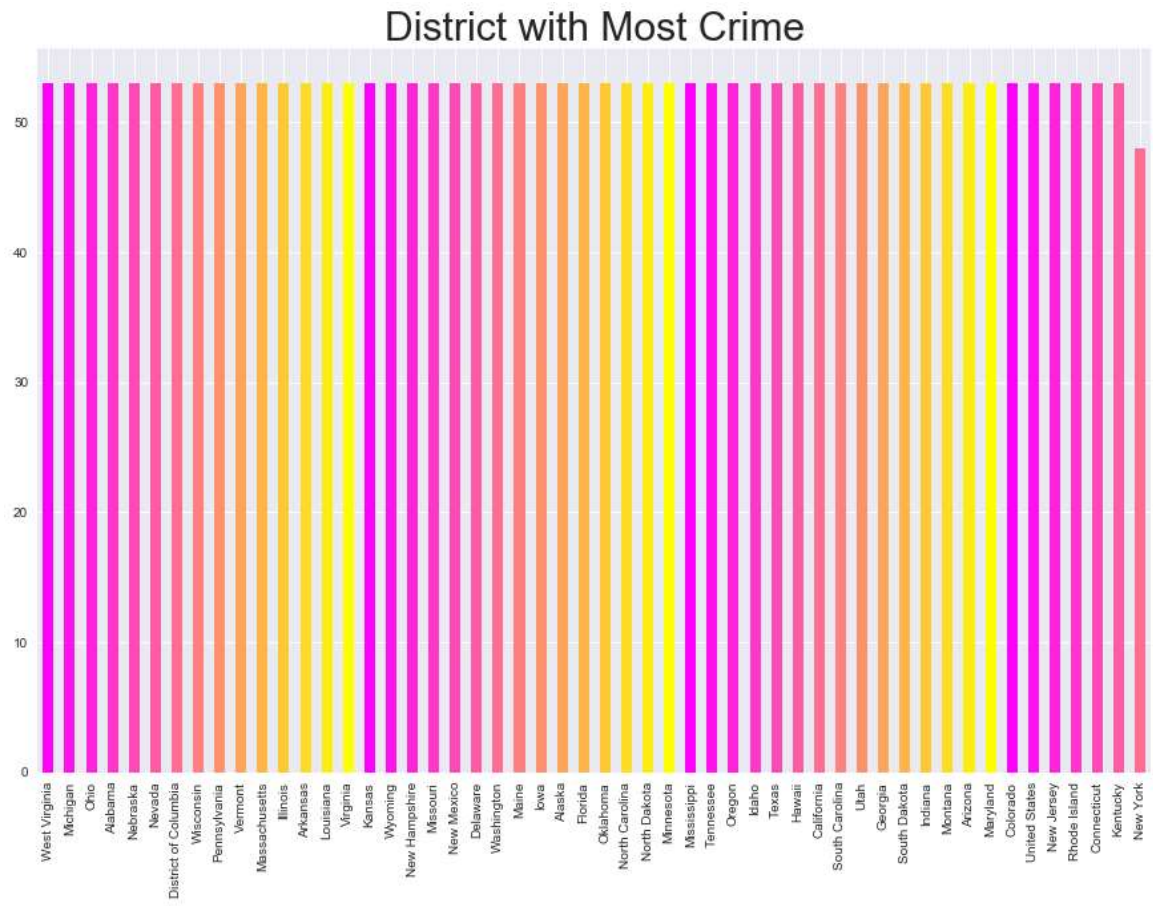
```
# Regions with count of crimes

plt.rcParams['figure.figsize'] = (20, 9)
plt.style.use('seaborn')

color = plt.cm.spring(np.linspace(0, 1, 15))
df['State'].value_counts().plot.bar(color = color, figsize = (15, 10))

plt.title('District with Most Crime',fontsize = 30)

plt.xticks(rotation = 90)
plt.show()
```



Modelling

In [31]:

```
from sklearn.cluster import KMeans
```

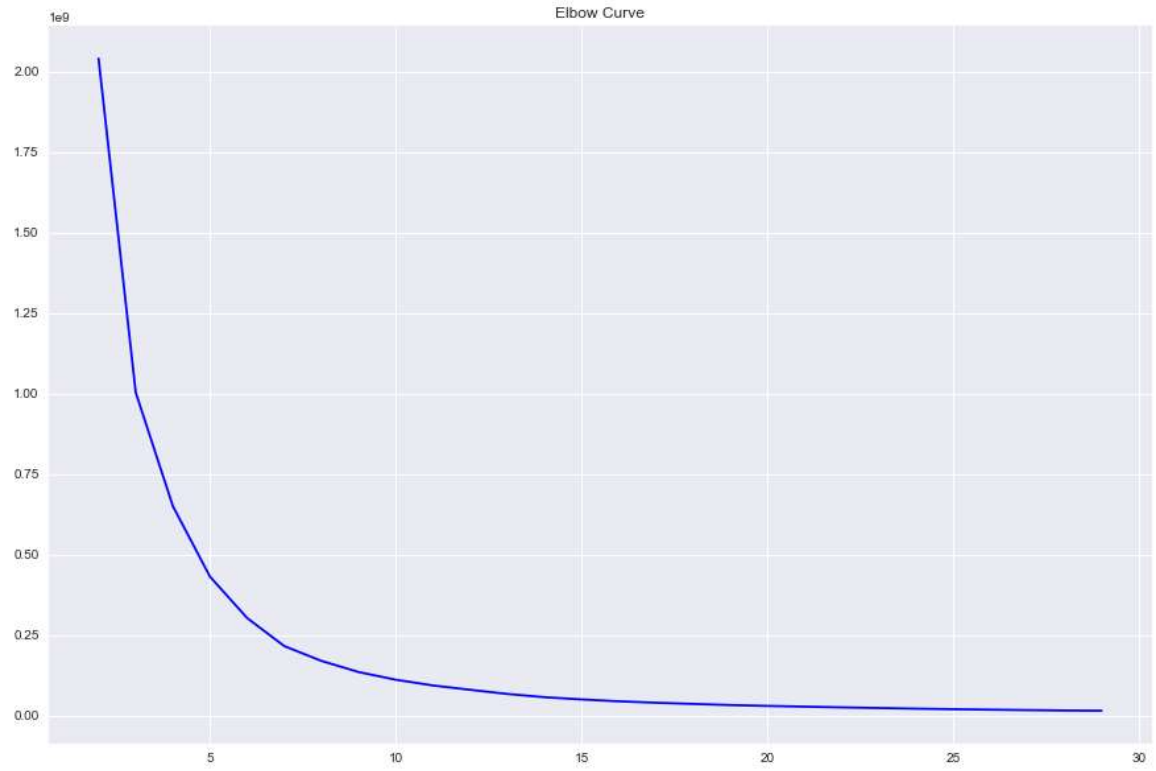
In [32]:

```
trial = df[['Data.Rates.Property.All', 'Year']]
data = np.asarray([np.asarray(trial['Data.Rates.Property.All']), np.asarray(trial['Yea
r'])]).T
```

In [33]:

```
X = data
distortions = []
for k in range(2,30):
    k_means = KMeans(n_clusters = k)
    k_means.fit(X)
    distortions.append(k_means.inertia_)

fig = plt.figure(figsize=(15,10))
plt.plot(range(2,30), distortions, 'bx-')
plt.title("Elbow Curve")
plt.show()
```



In [34]:

```
kmeans = KMeans(n_clusters = 6)
```

In [36]:

```
df.index = df.iloc[:,0]
df.index
```

Out[36]:

```
Index(['Alabama', 'Alabama', 'Alabama', 'Alabama', 'Alabama', 'Alabama',
       'Alabama', 'Alabama', 'Alabama', 'Alabama',
       ...
       'United States', 'United States', 'United States', 'United States',
       'United States', 'United States', 'United States', 'United States',
       'United States', 'United States'],
      dtype='object', name='State', length=2751)
```

In [37]:

```
df.head()
```

Out[37]:

State	State	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary
Alabama	Alabama	1960	3266740	1035.4	355.9
Alabama	Alabama	1961	3302000	985.5	339.3
Alabama	Alabama	1962	3358000	1067.0	349.1
Alabama	Alabama	1963	3347000	1150.9	376.9
Alabama	Alabama	1964	3407000	1358.7	466.6

5 rows × 21 columns

In [40]:

```
df = df.drop(['State'],axis=1)
```

In [41]:

```
df.head()
```

Out[41]:

State	Year	Data.Population	Data.Rates.Property.All	Data.Rates.Property.Burglary	Data.Ra
Alabama	1960	3266740	1035.4	355.9	
Alabama	1961	3302000	985.5	339.3	
Alabama	1962	3358000	1067.0	349.1	
Alabama	1963	3347000	1150.9	376.9	
Alabama	1964	3407000	1358.7	466.6	

In [48]:

```
k_fit = kmeans.fit(df)
```

In [49]:

```
sets = k_fit.labels_
```

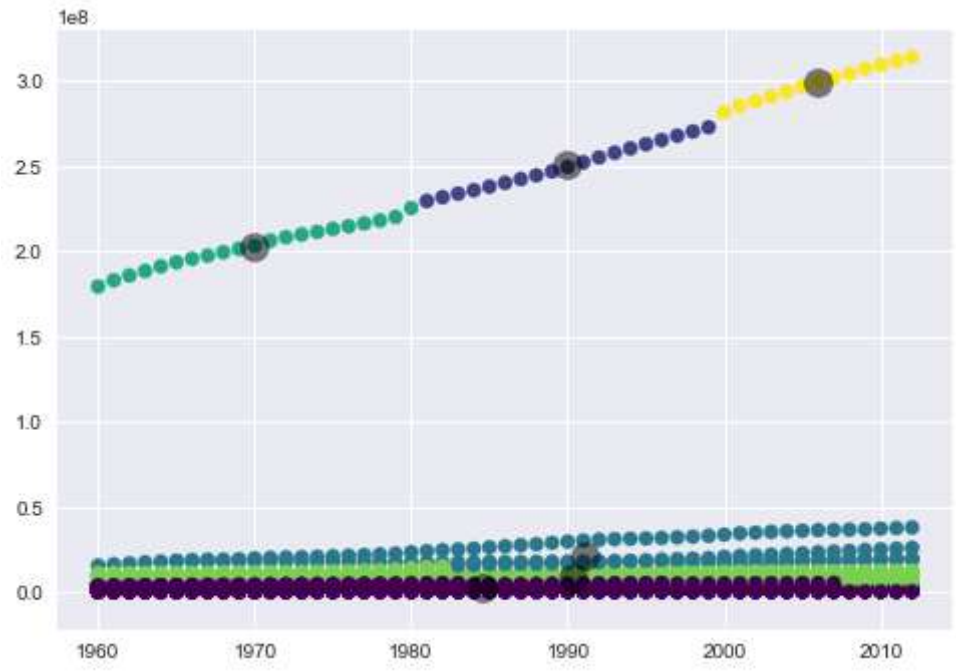
In [51]:

```
# Let's visualize the data we reduced to 2 sets.

plt.scatter(df.iloc[:,0], df.iloc[:,1], c = sets, s = 50, cmap = "viridis")

centers = k_fit.cluster_centers_

# We want to create 2 centers and show them on the visual.
plt.scatter(centers[:,0], centers[:,1], c = "black", s = 200, alpha = 0.5);
```



In [52]:

```
# Let us import 3D visualization. Otherwise it is necessary to download

from mpl_toolkits.mplot3d import Axes3D


# Let's create our sets again, this time it will be 3 dimensional variable

kmeans = KMeans(n_clusters = 3)
k_fit = kmeans.fit(df)
sets = k_fit.labels_
centers = kmeans.cluster_centers_
```

In [53]:

```
plt.rcParams['figure.figsize'] = (16, 9)
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(df.iloc[:, 0], df.iloc[:, 1], df.iloc[:, 2]);
```