

```
In [1]: import os
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

```
In [2]: os.getcwd()
```

```
Out[2]: 'C:\\Users\\Mahesh'
```

```
In [3]: os.chdir("C:\\Users\\Mahesh\\OneDrive\\Desktop\\DATA SET")
```

```
In [4]: os.getcwd()
```

```
Out[4]: 'C:\\Users\\Mahesh\\OneDrive\\Desktop\\DATA SET'
```

```
In [5]: os.chdir("C:\\Users\\Mahesh\\OneDrive\\Desktop\\DATA SET")
```

```
In [6]: d=pd.read_csv("C:\\Users\\Mahesh\\OneDrive\\Desktop\\DATA SET\\rape.csv")
d
```

Out[6]:

	STATE/UT	CRIME HEAD	2001	2002	2003	2004	2005	2006	2007	2008
0	ANDHRA PRADESH	RAPE	1150	1340	1237	1443	1415	1360	1436	1500
1	ARUNACHAL PRADESH	RAPE	51	61	35	56	38	40	57	60
2	ASSAM	RAPE	928	1019	1188	1233	1406	1290	1477	1490
3	BIHAR	RAPE	1400	1304	1120	1157	1455	1451	1816	1490
4	CHHATTISGARH	RAPE	1134	1214	1020	1144	1107	1211	1146	1190
...
451	DELHI	TOTAL CRIMES AGAINST WOMEN	3124	3080	4789	5196	5853	6207	5648	3190
452	LAKSHADWEEP	TOTAL CRIMES AGAINST WOMEN	0	3	3	1	0	1	2	3
453	PUDUCHERRY	TOTAL CRIMES AGAINST WOMEN	223	246	221	173	191	260	337	340
454	TOTAL (UTs)	TOTAL CRIMES AGAINST WOMEN	3621	3588	5289	5792	6434	6922	6435	3190
455	TOTAL (ALL- INDIA)	TOTAL CRIMES AGAINST WOMEN	243589	247981	239200	261322	266955	279823	310808	323400

456 rows × 14 columns



In [7]: `d.head()`

Out[7]:

	STATE/UT	CRIME HEAD	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
0	ANDHRA PRADESH	RAPE	1150	1340	1237	1443	1415	1360	1436	1531	1487	1761	1758
1	ARUNACHAL PRADESH	RAPE	51	61	35	56	38	40	57	37	60	49	47
2	ASSAM	RAPE	928	1019	1188	1233	1406	1290	1477	1445	1644	1629	1470
3	BIHAR	RAPE	1400	1304	1120	1157	1455	1451	1816	1464	1086	892	1185
4	CHHATTISGARH	RAPE	1134	1214	1020	1144	1107	1211	1146	1108	1128	1198	1257

In [8]: `d.isnull().sum()`

Out[8]:

STATE/UT	0
CRIME HEAD	0
2001	0
2002	0
2003	0
2004	0
2005	0
2006	0
2007	0
2008	0
2009	0
2010	0
2011	0
2012	0

dtype: int64

In [9]: `d.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 456 entries, 0 to 455
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
0   STATE/UT        456 non-null    object
1   CRIME HEAD      456 non-null    object
2   2001            456 non-null    int64
3   2002            456 non-null    int64
4   2003            456 non-null    int64
5   2004            456 non-null    int64
6   2005            456 non-null    int64
7   2006            456 non-null    int64
8   2007            456 non-null    int64
9   2008            456 non-null    int64
10  2009            456 non-null    int64
11  2010            456 non-null    int64
12  2011            456 non-null    int64
13  2012            456 non-null    int64
dtypes: int64(12), object(2)
memory usage: 50.0+ KB
```

In [10]: `years_title=[str(i) for i in range(2001,2013)]`
`years_title`

Out[10]: `['2001',
'2002',
'2003',
'2004',
'2005',
'2006',
'2007',
'2008',
'2009',
'2010',
'2011',
'2012']`

In [11]: `STATES_OF_INDIA=d['STATE/UT'].unique()
STATES_OF_INDIA=STATES_OF_INDIA[:-4]
STATES_OF_INDIA`

Out[11]: `array(['ANDHRA PRADESH', 'ARUNACHAL PRADESH', 'ASSAM', 'BIHAR',
'CHHATTISGARH', 'GOA', 'GUJARAT', 'HARYANA', 'HIMACHAL PRADESH',
'JAMMU & KASHMIR', 'JHARKHAND', 'KARNATAKA', 'KERALA',
'MADHYA PRADESH', 'MAHARASHTRA', 'MANIPUR', 'MEGHALAYA', 'MIZORAM',
'NAGALAND', 'ODISHA', 'PUNJAB', 'RAJASTHAN', 'SIKKIM',
'TAMIL NADU', 'TRIPURA', 'UTTAR PRADESH', 'UTTARAKHAND',
'WEST BENGAL', 'TOTAL (STATES)', 'A & N ISLANDS', 'CHANDIGARH',
'D & N HAVELI', 'DAMAN & DIU', 'DELHI'], dtype=object)`

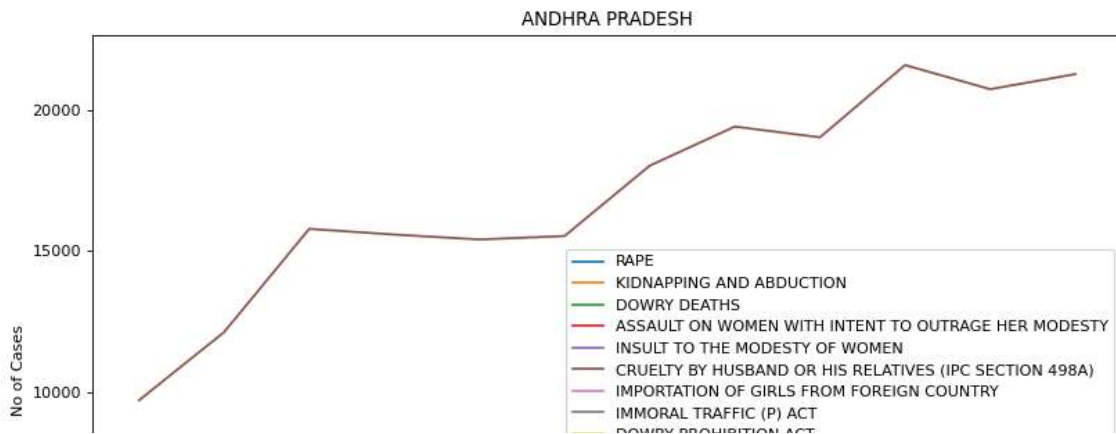
```
In [12]: TYPES_OF_CASES=d['CRIME HEAD'].unique()
TYPES_OF_CASES=TYPES_OF_CASES[:-1]
TYPES_OF_CASES
```

```
Out[12]: array(['RAPE', 'KIDNAPPING AND ABDUCTION', 'DOWRY DEATHS',
'ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY',
'INSULT TO THE MODESTY OF WOMEN',
'CRUELTY BY HUSBAND OR HIS RELATIVES (IPC SECTION 498A)',
'IMPORTATION OF GIRLS FROM FOREIGN COUNTRY',
'IMMORAL TRAFFIC (P) ACT', 'DOWRY PROHIBITION ACT',
'INDECENT REPRESENTATION OF WOMEN (P) ACT',
'COMMISSION OF SATI (P) ACT'], dtype=object)
```

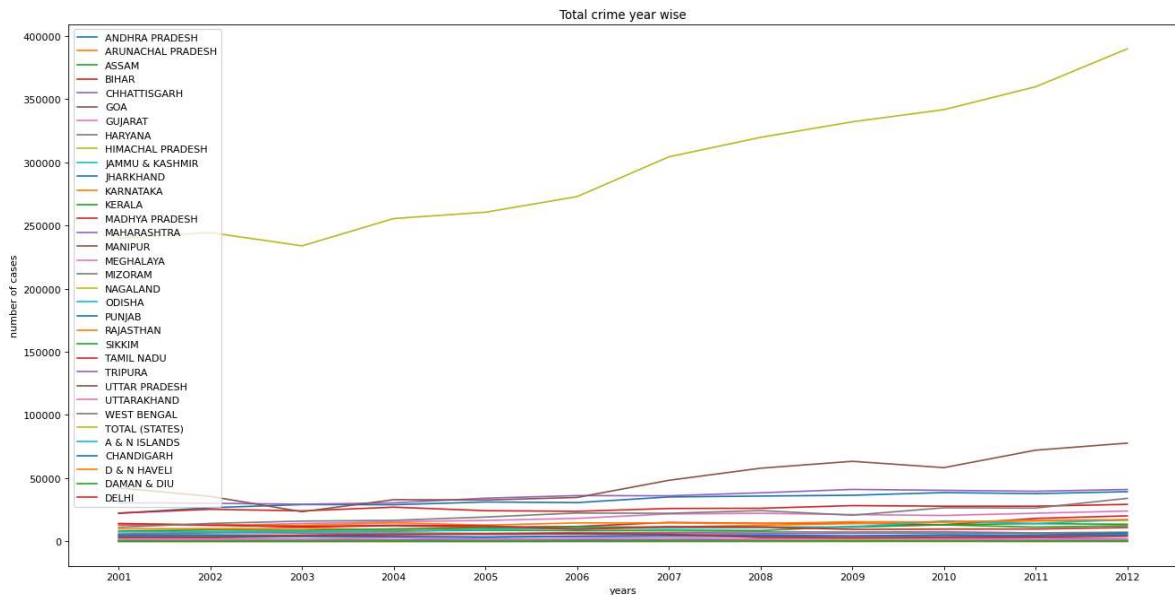
```
In [13]: for state in STATES_OF_INDIA:
fig=plt.figure(figsize=(12,8),dpi=80,facecolor='w',edgecolor='k')
plt.title(state)
plt.xlabel('Years')
plt.ylabel('No of Cases')
for case in TYPES_OF_CASES:
temp_df=d[(d['STATE/UT']==state)&(d['CRIME HEAD']==case)]
N_cases=[temp_df[c].values[0] for c in years_title]
plt.plot(years_title,N_cases)
plt.legend(TYPES_OF_CASES)
```

C:\Users\Mahesh\AppData\Local\Temp\ipykernel_19336\600561387.py:2: RuntimeWarning: More than 20 figures have been opened. Figures created through the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicitly closed and may consume too much memory. (To control this warning, see the rcParam `figure.max_open_warning`).

```
fig=plt.figure(figsize=(12,8),dpi=80,facecolor='w',edgecolor='k')
```



```
In [14]: fig=plt.figure(figsize=(20,10),dpi=80,facecolor='w',edgecolor='k')
plt.title('Total crime year wise')
plt.xlabel('years')
plt.ylabel('number of cases')
for state in STATES_OF_INDIA:
    temp_df=d[(d['STATE/UT']==state)&(d['CRIME HEAD']=='TOTAL CRIMES AGAINST W
    N_cases=[temp_df[c].values[0] for c in years_title]
    plt.plot(years_title,N_cases)
plt.legend(STATES_OF_INDIA)
```



```
In [15]: print('Dataset:')
for col_name in d.columns:
    if (d[col_name].dtypes)=='object':
        unique_cat=len(d[col_name].unique())
        print("Feature {col_name} has {unique_cat}").format(col_name=col_name, unique_cat=unique_cat)
```

Dataset:

Feature STATE/UT has 38

Feature CRIME HEAD has 12

In [16]:

d

Out[16]:

	STATE/UT	CRIME HEAD	2001	2002	2003	2004	2005	2006	2007	2008
0	ANDHRA PRADESH	RAPE	1150	1340	1237	1443	1415	1360	1436	1500
1	ARUNACHAL PRADESH	RAPE	51	61	35	56	38	40	57	60
2	ASSAM	RAPE	928	1019	1188	1233	1406	1290	1477	1490
3	BIHAR	RAPE	1400	1304	1120	1157	1455	1451	1816	1490
4	CHHATTISGARH	RAPE	1134	1214	1020	1144	1107	1211	1146	1190
...
451	DELHI	TOTAL CRIMES AGAINST WOMEN	3124	3080	4789	5196	5853	6207	5648	3190
452	LAKSHADWEEP	TOTAL CRIMES AGAINST WOMEN	0	3	3	1	0	1	2	0
453	PUDUCHERRY	TOTAL CRIMES AGAINST WOMEN	223	246	221	173	191	260	337	310
454	TOTAL (UTs)	TOTAL CRIMES AGAINST WOMEN	3621	3588	5289	5792	6434	6922	6435	3190
455	TOTAL (ALL-INDIA)	TOTAL CRIMES AGAINST WOMEN	243589	247981	239200	261322	266955	279823	310808	323500

456 rows × 14 columns

```
In [18]: from sklearn import preprocessing
lab=preprocessing.LabelEncoder()
d['CRIME HEAD']=lab.fit_transform(d['CRIME HEAD'])
d
```

```
Out[18]:
```

	STATE/UT	CRIME HEAD	2001	2002	2003	2004	2005	2006	2007	2008
0	ANDHRA PRADESH	10	1150	1340	1237	1443	1415	1360	1436	1531
1	ARUNACHAL PRADESH	10	51	61	35	56	38	40	57	37
2	ASSAM	10	928	1019	1188	1233	1406	1290	1477	1445
3	BIHAR	10	1400	1304	1120	1157	1455	1451	1816	1464
4	CHHATTISGARH	10	1134	1214	1020	1144	1107	1211	1146	1108
...
451	DELHI	11	3124	3080	4789	5196	5853	6207	5648	3115
452	LAKSHADWEEP	11	0	3	3	1	0	1	2	2
453	PUDUCHERRY	11	223	246	221	173	191	260	337	191
454	TOTAL (UTs)	11	3621	3588	5289	5792	6434	6922	6435	3724
455	TOTAL (ALL- INDIA)	11	243589	247981	239200	261322	266955	279823	310808	323510

456 rows × 14 columns



```
In [19]: from sklearn.cluster import KMeans
kmeans=KMeans(n_clusters=9)
kmeans.fit(d.iloc[:,1:])
```

```
Out[19]: KMeans(n_clusters=9)
```



```
In [21]: kmeans.cluster_centers_
```

```
Out[21]: array([[6.73684211e+00, 9.45142105e+03, 9.53452632e+03, 9.55673684e+03,
 9.86721053e+03, 1.01694737e+04, 1.04397895e+04, 1.12898421e+04,
 1.16063158e+04, 1.16904737e+04, 1.20100526e+04, 1.27533158e+04,
 1.42036842e+04],
 [1.10000000e+01, 2.41778500e+05, 2.46187000e+05, 2.36555500e+05,
 2.58426000e+05, 2.63738000e+05, 2.76362000e+05, 3.07590500e+05,
 3.21648000e+05, 3.33711500e+05, 3.43533000e+05, 3.61820500e+05,
 3.92121000e+05],
 [5.43163539e+00, 1.78694370e+02, 1.93235925e+02, 1.85713137e+02,
 2.00646113e+02, 2.17603217e+02, 2.23085791e+02, 2.31351206e+02,
 2.16801609e+02, 2.17672922e+02, 2.29128686e+02, 2.32343164e+02,
 2.51718499e+02],
 [2.00000000e+00, 1.08837000e+05, 1.12495000e+05, 1.09259500e+05,
 1.24357000e+05, 1.26188000e+05, 1.35427500e+05, 1.54949000e+05,
 1.64423500e+05, 1.74021000e+05, 1.79944500e+05, 1.80242500e+05,
 1.97194500e+05],
 [1.10000000e+01, 2.64090000e+04, 2.83665000e+04, 2.92185000e+04,
 2.97040000e+04, 3.26460000e+04, 3.34285000e+04, 3.55805000e+04,
 3.71105000e+04, 3.87800000e+04, 3.94735000e+04, 3.87045000e+04,
 4.01680000e+04],
 [3.66666667e+00, 4.19986667e+04, 4.03643333e+04, 3.58513333e+04,
 3.97473333e+04, 3.94793333e+04, 4.13096667e+04, 4.76406667e+04,
 5.15140000e+04, 5.27316667e+04, 5.24770000e+04, 5.95680000e+04,
 6.34590000e+04],
 [4.93023256e+00, 2.83067442e+03, 2.87346512e+03, 2.94300000e+03,
 3.10574419e+03, 3.10460465e+03, 3.21341860e+03, 3.44416279e+03,
 3.35686047e+03, 3.46253488e+03, 3.55058140e+03, 3.73739535e+03,
 4.04137209e+03],
 [7.00000000e+00, 1.76028889e+04, 1.80794444e+04, 1.80136667e+04,
 1.94847778e+04, 1.97697778e+04, 2.09772222e+04, 2.29368889e+04,
 2.36123333e+04, 2.40187778e+04, 2.49384444e+04, 2.54742222e+04,
 2.72938889e+04],
 [6.66666667e+00, 1.83713333e+04, 1.77296667e+04, 1.26860000e+04,
 1.74960000e+04, 1.74453333e+04, 1.86780000e+04, 2.34513333e+04,
 2.73036667e+04, 3.08090000e+04, 3.26413333e+04, 4.12066667e+04,
 4.45736667e+04]])
```

```
In [22]: labels=kmeans.labels_  
labels
```

```
Out[22]: array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 6, 2, 6, 7, 2, 2, 2, 2, 2, 2, 2, 7, 2, 2, 6, 6, 2, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 6,  
                8, 2, 2, 2, 2, 2, 2, 2, 2, 8, 2, 2, 2, 6, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 6, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 2, 2, 7, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 7, 6, 2, 2, 2, 6, 2, 2, 2, 2, 6, 2, 6, 6, 0, 6, 2, 2, 2,  
                2, 6, 2, 6, 2, 6, 2, 6, 2, 6, 5, 2, 2, 2, 2, 2, 2, 2, 2, 5, 6, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 6, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 7, 2, 6, 6, 6, 2, 0, 6,  
                2, 2, 2, 6, 6, 0, 7, 2, 2, 2, 2, 6, 6, 0, 2, 6, 2, 8, 2, 0, 3, 2,  
                2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 6, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 6,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
                4, 2, 0, 0, 6, 2, 7, 6, 2, 6, 6, 0, 0, 7, 4, 2, 2, 2, 2, 0, 6, 0,  
                2, 0, 2, 5, 6, 7, 1, 2, 2, 2, 2, 6, 2, 2, 6, 1])
```

```
In [24]: import numpy as np
         unique, counts = np.unique(kmeans.labels_, return_counts=True)
         dict_data = dict(zip(unique, counts))
         dict_data
```

```
Out[24]: {0: 19, 1: 2, 2: 373, 3: 2, 4: 2, 5: 3, 6: 43, 7: 9, 8: 3}
```

```
In [25]: d['cluster']=kmeans.labels_  
d
```

Out[25]:

	STATE/UT	CRIME HEAD	2001	2002	2003	2004	2005	2006	2007	2008
0	ANDHRA PRADESH	10	1150	1340	1237	1443	1415	1360	1436	1531
1	ARUNACHAL PRADESH	10	51	61	35	56	38	40	57	37
2	ASSAM	10	928	1019	1188	1233	1406	1290	1477	1445
3	BIHAR	10	1400	1304	1120	1157	1455	1451	1816	1464
4	CHHATTISGARH	10	1134	1214	1020	1144	1107	1211	1146	1108
...
451	DELHI	11	3124	3080	4789	5196	5853	6207	5648	3115
452	LAKSHADWEEP	11	0	3	3	1	0	1	2	2
453	PUDUCHERRY	11	223	246	221	173	191	260	337	191
454	TOTAL (UTs)	11	3621	3588	5289	5792	6434	6922	6435	3724
455	TOTAL (ALL- INDIA)	11	243589	247981	239200	261322	266955	279823	310808	323510

456 rows × 15 columns



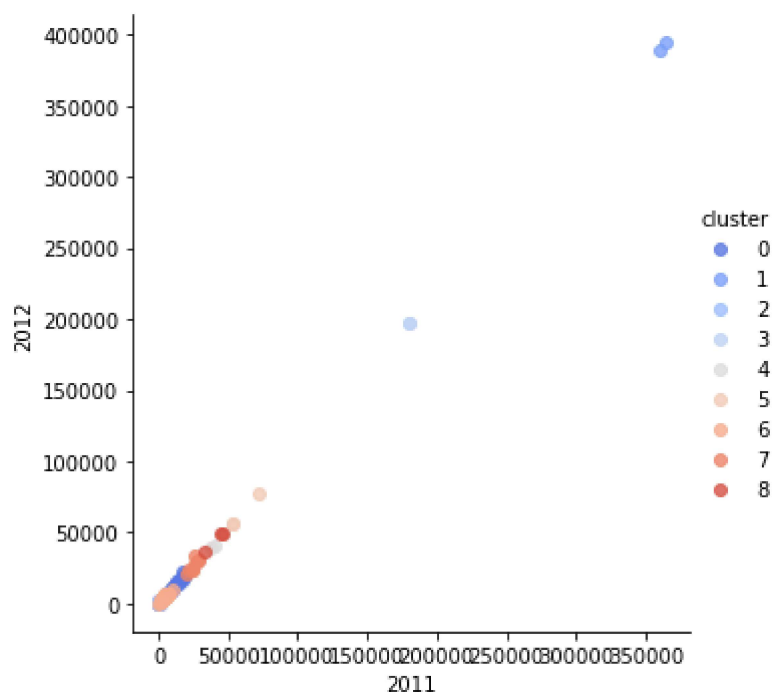
```
In [27]: import seaborn as sns
sns.lmplot('2011', '2012', data=d, hue='cluster', palette='coolwarm', size=5, aspect=1)
```

C:\Users\Mahesh\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(
C:\Users\Mahesh\anaconda3\lib\site-packages\seaborn\regression.py:581: UserWarning: The `size` parameter has been renamed to `height`; please update your code.

warnings.warn(msg, UserWarning)

Out[27]: <seaborn.axisgrid.FacetGrid at 0x1efbd7b1fa0>



```
In [28]: kmeans.inertia_
```

Out[28]: 7376543830.351573

```
In [29]: kmeans.score
```

Out[29]: <bound method KMeans.score of KMeans(n_clusters=9)>

In [30]:

d

Out[30]:

	STATE/UT	CRIME HEAD	2001	2002	2003	2004	2005	2006	2007	2008
0	ANDHRA PRADESH	10	1150	1340	1237	1443	1415	1360	1436	1531
1	ARUNACHAL PRADESH	10	51	61	35	56	38	40	57	37
2	ASSAM	10	928	1019	1188	1233	1406	1290	1477	1445
3	BIHAR	10	1400	1304	1120	1157	1455	1451	1816	1464
4	CHHATTISGARH	10	1134	1214	1020	1144	1107	1211	1146	1108
...
451	DELHI	11	3124	3080	4789	5196	5853	6207	5648	3115
452	LAKSHADWEEP	11	0	3	3	1	0	1	2	2
453	PUDUCHERRY	11	223	246	221	173	191	260	337	191
454	TOTAL (UTs)	11	3621	3588	5289	5792	6434	6922	6435	3724
455	TOTAL (ALL- INDIA)	11	243589	247981	239200	261322	266955	279823	310808	323510

456 rows × 15 columns

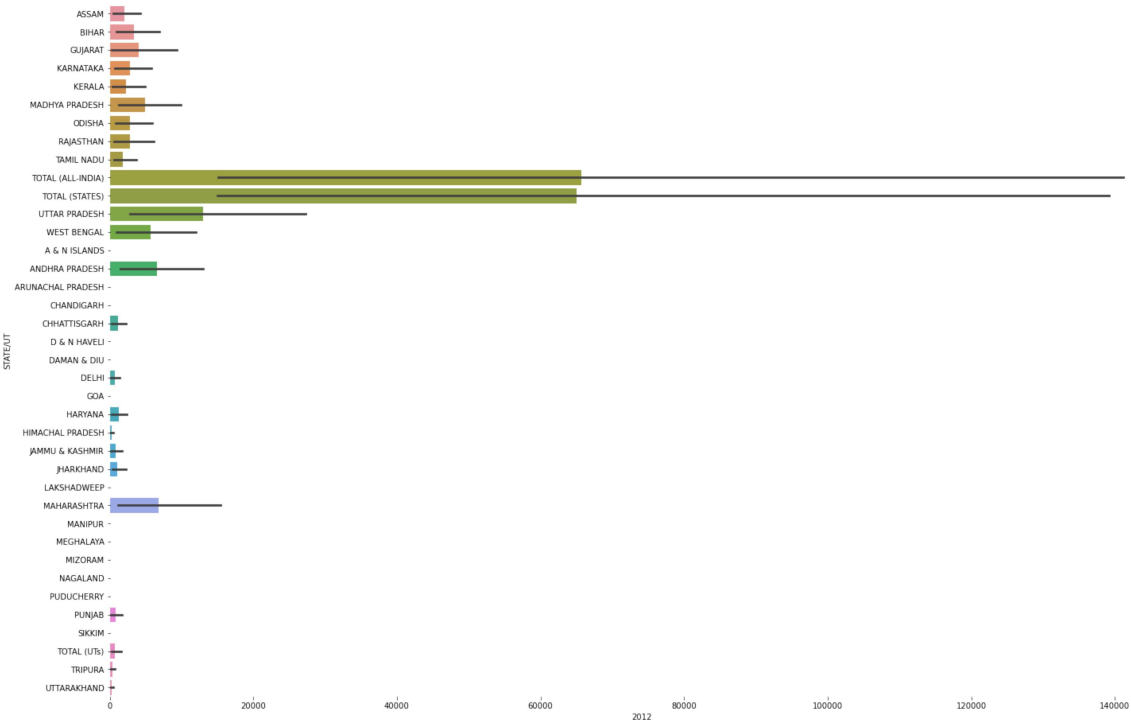


In [31]:

```
cust=[[10,1150,1340,1237,1443,1415,1360,1436,1531,1487,1761,1758,1664]]
kmeans.predict(cust)[0]
```

Out[31]: 2

```
In [32]: f,ax=plt.subplots(figsize=(24,16))
stats=d.sort_values(["cluster","STATE/UT"],ascending=True)
sns.set_color_codes("pastel")
sns.barplot(y="STATE/UT",x="2012",data=stats)
sns.despine(left=True,bottom=True)
```



```
In [37]: X=d.iloc[:,1:14]
y=d.iloc[:,d.columns=='cluster']
print(X.head())
print(y.head())
```

	CRIME	HEAD	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	\
0		10	1150	1340	1237	1443	1415	1360	1436	1531	1487	1761	
1		10	51	61	35	56	38	40	57	37	60	49	
2		10	928	1019	1188	1233	1406	1290	1477	1445	1644	1629	
3		10	1400	1304	1120	1157	1455	1451	1816	1464	1086	892	
4		10	1134	1214	1020	1144	1107	1211	1146	1108	1128	1198	
	2011	2012											
0	1758	1664											
1	47	47											
2	1470	1626											
3	1185	1327											
4	1257	1214											
	cluster												
0		2											
1		2											
2		2											
3		2											
4		2											


```
In [46]: from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
lg.fit(xtrain,ytrain)
pred2=lg.predict(xtest)
acc2=accuracy_score(pred2,ytest)
print(acc2)
clf2=classification_report(pred2,ytest)
print(clf2)
```

C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\utils\validation.py:63:
DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

return f(*args, **kwargs)

C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\linear_model_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html> (<https://scikit-learn.org/stable/modules/preprocessing.html>)

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
0.43478260869565216
          precision    recall  f1-score   support

     0       0.33      0.06      0.10        17
     1       0.00      0.00      0.00         0
     2       0.47      1.00      0.64        38
     3       0.00      0.00      0.00         8
     4       0.00      0.00      0.00         3
     5       0.00      0.00      0.00         9
     6       0.33      0.12      0.18         8
     7       0.00      0.00      0.00         3
     8       0.00      0.00      0.00         6

 accuracy          0.43        92
 macro avg       0.13      0.13      0.10        92
 weighted avg    0.28      0.43      0.30        92
```



```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.  
py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and b  
eing set to 0.0 in labels with no predicted samples. Use `zero_division` par  
ameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.  
py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and bein  
g set to 0.0 in labels with no true samples. Use `zero_division` parameter t  
o control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.  
py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and b  
eing set to 0.0 in labels with no predicted samples. Use `zero_division` par  
ameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.  
py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and bein  
g set to 0.0 in labels with no true samples. Use `zero_division` parameter t  
o control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.  
py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and b  
eing set to 0.0 in labels with no predicted samples. Use `zero_division` par  
ameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.  
py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and bein  
g set to 0.0 in labels with no true samples. Use `zero_division` parameter t  
o control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))
```

```
In [49]: from sklearn import svm
sv=svm.LinearSVC()
sv.fit(xtrain,ytrain)
pred3=sv.predict(xtest)
acc3=accuracy_score(pred3,ytest)
print(acc3)
clf3=classification_report(pred3,ytest)
print(clf3)
```

0.6956521739130435

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	0.00	0.00	0.00	0
2	0.78	1.00	0.88	63
4	1.00	0.05	0.10	19
5	0.00	0.00	0.00	2
6	0.00	0.00	0.00	0
7	0.00	0.00	0.00	0
8	0.00	0.00	0.00	8
accuracy			0.70	92
macro avg	0.22	0.13	0.12	92
weighted avg	0.74	0.70	0.62	92

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\utils\validation.py:63:
DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
```

```
return f(*args, **kwargs)
```

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\svm\_base.py:985: Conver-  
genceWarning: Liblinear failed to converge, increase the number of iteration  
s.
```

```
warnings.warn("Liblinear failed to converge, increase "
```

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

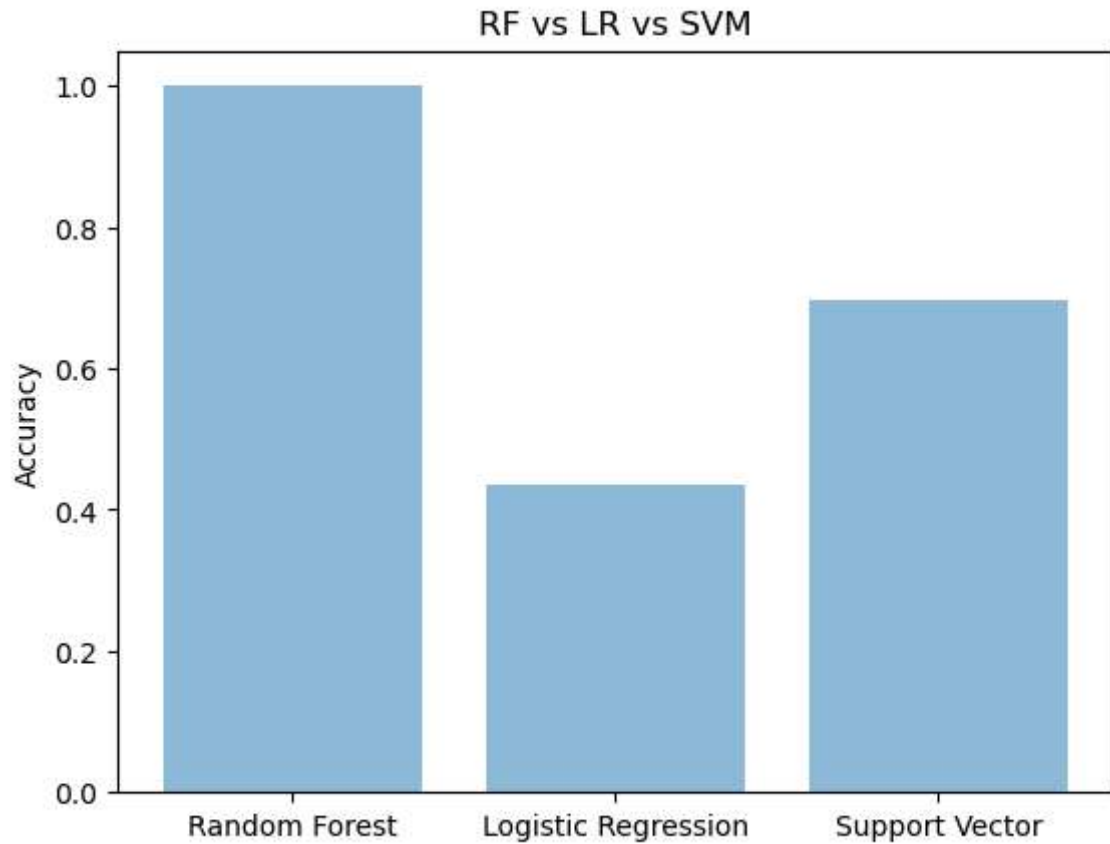
```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
warn prf(average, modifier, msg start, len(result))
```

```
C:\Users\Mahesh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1248: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, msg_start, len(result))
```

```
In [50]: from matplotlib import pyplot as plt;plt.rcParams()
objects=('Random Forest','Logistic Regression','Support Vector')
y_pos=np.arange(len(objects))
performance=[acc1,acc2,acc3]
plt.bar(y_pos,performance,align='center',alpha=0.5)
plt.xticks(y_pos,objects)
plt.ylabel('Accuracy')
plt.title('RF vs LR vs SVM')
plt.show()
```



```
In [ ]:
```