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

```
df = pd.read_csv('/cyber_crime - cyber_crime.csv')
```

```
df.head()
```

	S. No	Category	State/UT	2016	2017	2018	Percentage Share of State/UT (2018)	Mid-Year Projected Population (in Lakhs) (2018)+	Rate of Total Cyber Crimes (2018)++
0	1	State	Andhra Pradesh	616	931	1207	4.4	520.3	2.3
1	2	State	Arunachal Pradesh	4	1	7	0.0	14.9	0.5
2	3	State	Assam	696	1120	2022	7.4	340.4	5.9
3	4	State	Bihar	309	433	374	1.4	1183.3	0.3
4	5	State	Chhattisgarh	90	171	139	0.5	284.7	0.5

```
df.tail()
```

	S. No	Category	State/UT	2016	2017	2018	Percentage Share of State/UT (2018)	Mid-Year Projected Population (in Lakhs) (2018)+	Rat T C Cr (201
34	34	Union Territory	Delhi UT	98	162	189	0.7	195.6	
35	35	Union Territory	Lakshadweep	0	0	4	0.0	0.7	
36	36	Union Territory	Puducherry	2	5	14	0.1	14.8	
37	Union Territory	Union Territory	Total UT(s)	130	203	244	0.9	236.0	
38	Total (All	Total (All	Total (All	12317	21796	27248	100.0	13233.8	

```
df.shape
```

```
(39, 9)
```

```
df.columns
```

```
Index(['S. No', 'Category', 'State/UT', '2016', '2017', '2018',
       'Percentage Share of State/UT (2018)',
       'Mid-Year Projected Population (in Lakhs) (2018)+',
       'Rate of Total Cyber Crimes (2018)++'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 9 columns):
 #   Column                                            Non-Null Count  Dtype
---  ------                                            --------------  -----
 0   S. No                                             39 non-null     object
 1   Category                                          39 non-null     object
 2   State/UT                                          39 non-null     object
 3   2016                                              39 non-null     int64
 4   2017                                              39 non-null     int64
 5   2018                                              39 non-null     int64
 6   Percentage Share of State/UT (2018)               39 non-null     float64
 7   Mid-Year Projected Population (in Lakhs) (2018)+  39 non-null     float64
 8   Rate of Total Cyber Crimes (2018)++               39 non-null     float64
dtypes: float64(3), int64(3), object(3)
memory usage: 2.9+ KB
```

```
df.describe()
```

	2016	2017	2018	Percentage Share of State/UT (2018)	Mid-Year Projected Population (in Lakhs) (2018)+	Rate of Total Cyber Crimes (2018)++
count	39.000000	39.000000	39.000000	39.000000	39.000000	39.000000
mean	947.461538	1676.615385	2096.000000	7.689744	1017.987179	1.689744
std	2724.974532	4832.658115	6065.161416	22.257391	2885.991893	1.811193
min	0.000000	0.000000	0.000000	0.000000	0.700000	0.000000
25%	9.500000	11.500000	24.500000	0.100000	18.300000	0.500000
50%	102.000000	176.000000	239.000000	0.900000	284.000000	1.000000
75%	439.500000	772.000000	886.500000	3.250000	663.850000	2.200000
max	12317.000000	21796.000000	27248.000000	100.000000	13233.800000	8.900000

```
df.isnull().sum()
```

```
S. No                                               0
Category                                            0
State/UT                                            0
```

```
    2016                                               0
    2017                                               0
    2018                                               0
    Percentage Share of State/UT (2018)                0
    Mid-Year Projected Population (in Lakhs) (2018)+   0
    Rate of Total Cyber Crimes (2018)++                0
    dtype: int64
```

```
df.nunique()
```

```
    S. No                                              39
    Category                                            3
    State/UT                                           39
    2016                                               34
    2017                                               35
    2018                                               36
    Percentage Share of State/UT (2018)                23
    Mid-Year Projected Population (in Lakhs) (2018)+   38
    Rate of Total Cyber Crimes (2018)++                23
    dtype: int64
```

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

```
    array(['State', 'Union Territory', 'Total (All India)'], dtype=object)
```

```
df['Category'].value_counts()
```

```
    State                30
    Union Territory       8
    Total (All India)     1
    Name: Category, dtype: int64
```

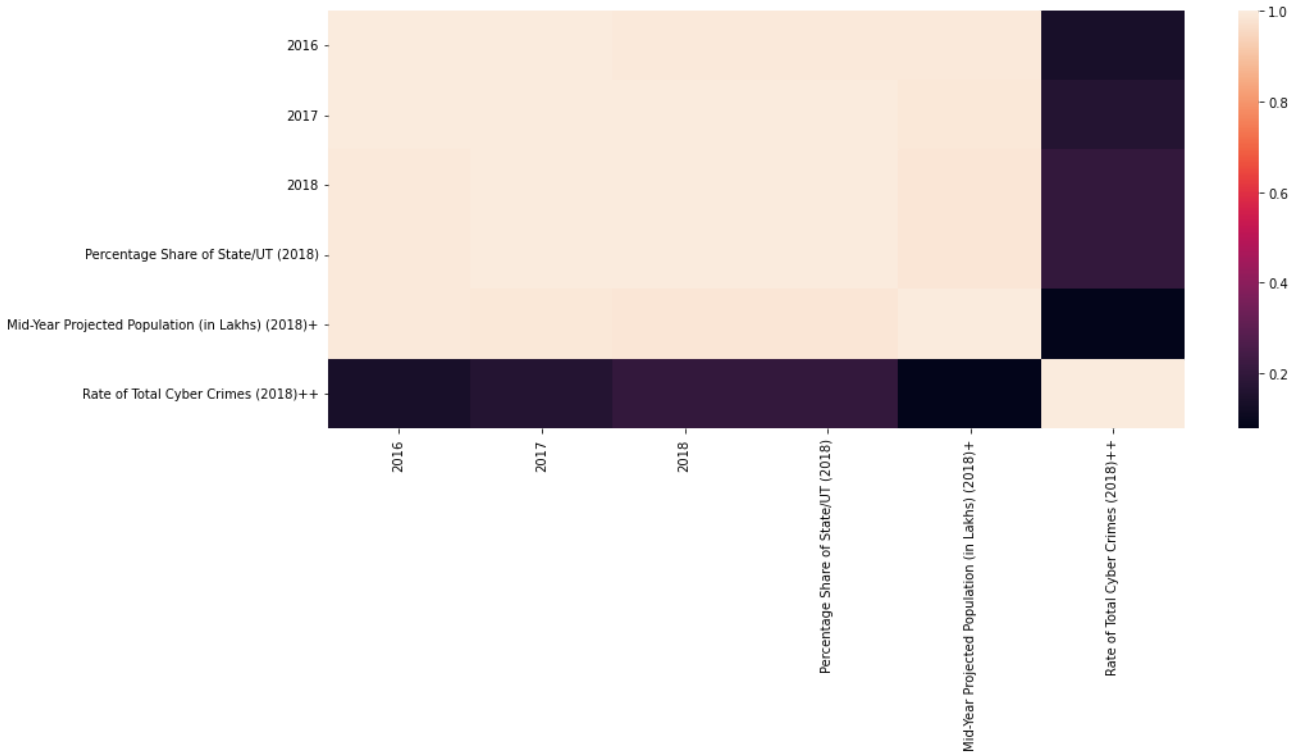
```
df.corr()
```

	2016	2017	2018	Percentage Share of State/UT (2018)	Mid-Year Projected Population (in Lakhs) (2018)+	Rate of Total Cyber Crimes (2018)++
2016	1.00000	0.998590	0.993830	0.993860	0.992970	0.136820
2017	0.99859	1.000000	0.998014	0.998030	0.991394	0.164416
2018	0.99383	0.998014	1.000000	0.999999	0.986735	0.200750
Percentage Share of State/UT (2018)	0.99386	0.998030	0.999999	1.000000	0.986789	0.200419
Mid-Year Projected Population (in Lakhs) (2018)+	0.99297	0.991394	0.986735	0.986789	1.000000	0.077051
Rate of Total Cyber Crimes (2018)++	0.13682	0.164416	0.200750	0.200419	0.077051	1.000000

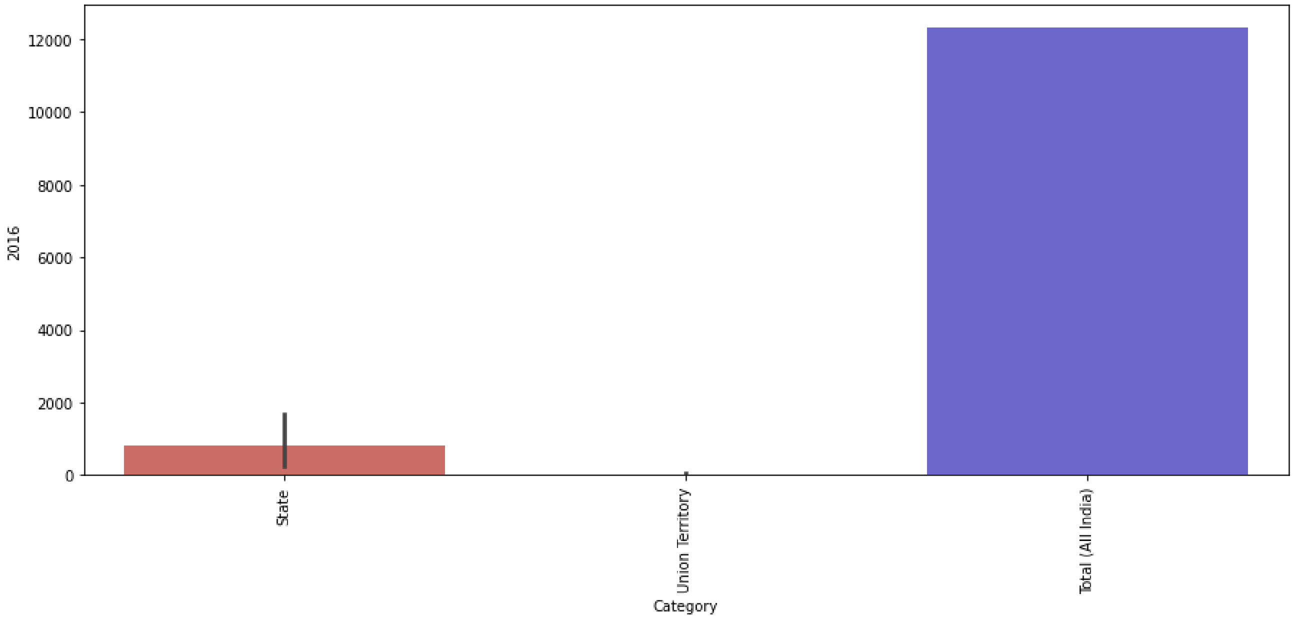
```
df.corr().style.background_gradient(cmap = 'coolwarm')
```

	2016	2017	2018	Percentage Share of State/UT (2018)	Mid-Year Projected Population (in Lakhs) (2018)+	Rate of Total Cyber Crimes (2018)++
2016	1.000000	0.998590	0.993830	0.993860	0.992970	0.136820
2017	0.998590	1.000000	0.998014	0.998030	0.991394	0.164416
2018	0.993830	0.998014	1.000000	0.999999	0.986735	0.200750
Percentage Share of State/UT (2018)	0.993860	0.998030	0.999999	1.000000	0.986789	0.200419
Mid-Year Projected Population (in Lakhs) (2018)+	0.992970	0.991394	0.986735	0.986789	1.000000	0.077051

```
plt.figure(figsize = (15,6))
sns.heatmap(df.corr())
plt.show()
```

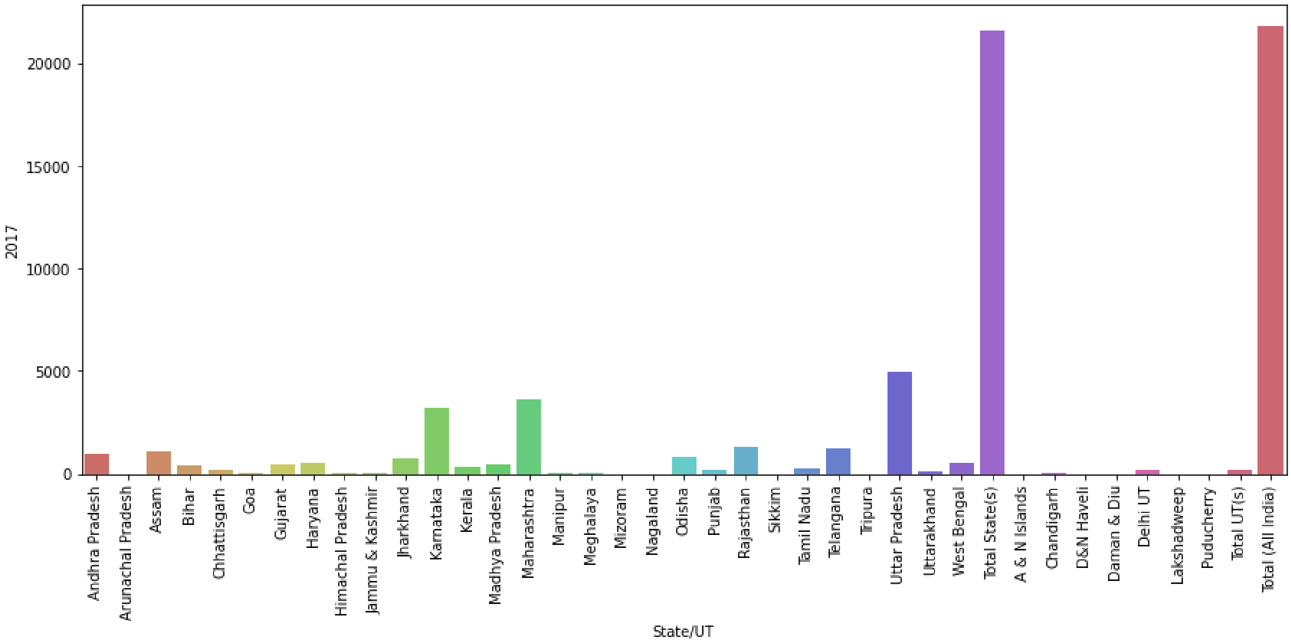


```
plt.figure(figsize = (15,6))
sns.barplot(x = 'Category',y = '2016',data = df,palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



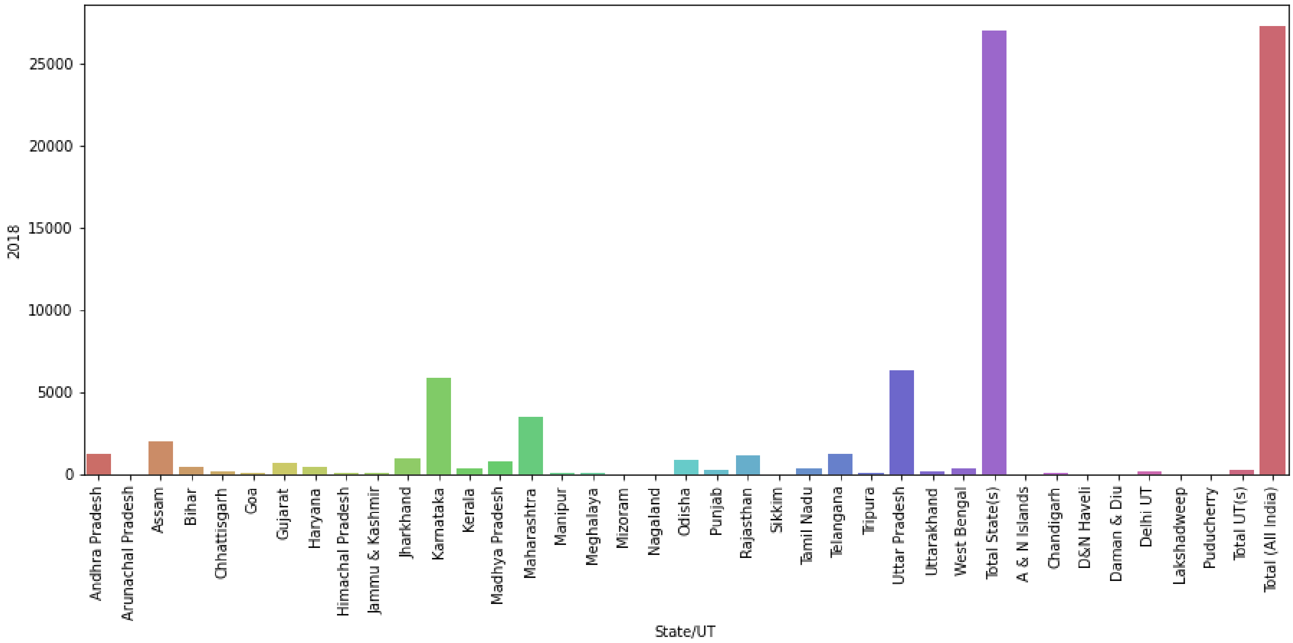
```
plt.figure(figsize =(15,6))
sns.barplot(x = 'State/UT',y = '2016',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```

```
plt.figure(figsize =(15,6))
sns.barplot(x = 'State/UT',y = '2017',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```

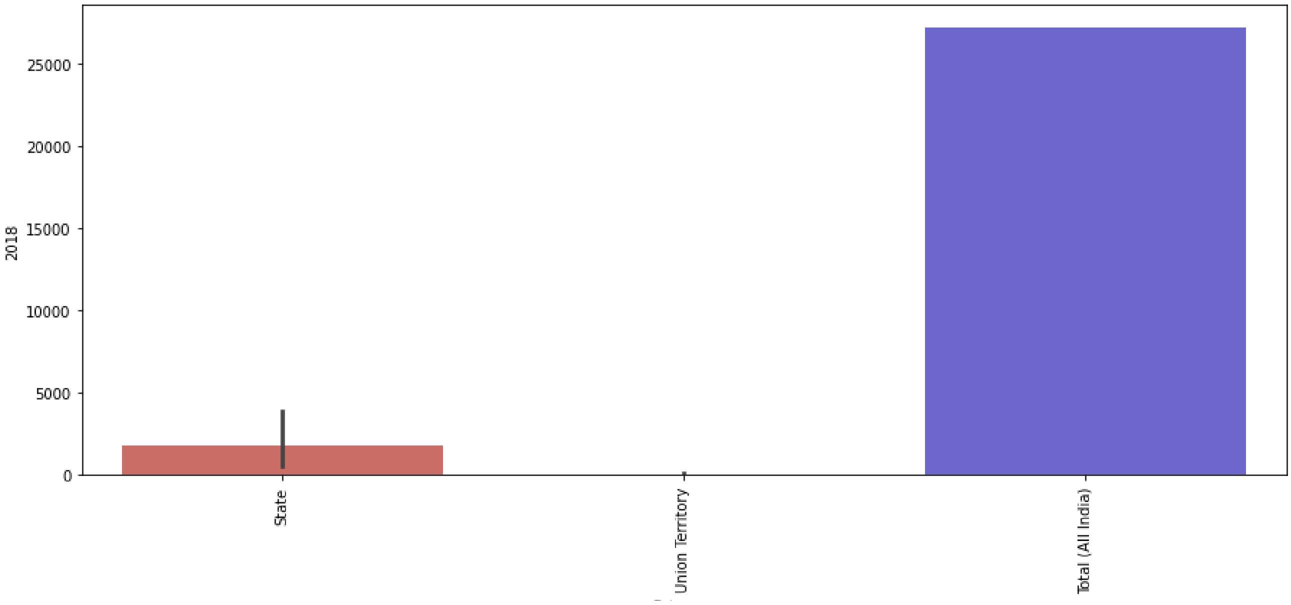


```
plt.figure(figsize =(15,6))
sns.barplot(x = 'Category',y = '2017',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```

```
plt.figure(figsize =(15,6))
sns.barplot(x = 'State/UT',y = '2018',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



```
plt.figure(figsize =(15,6))
sns.barplot(x = 'Category',y = '2018',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```


25000
20000
15000
10000
5000
0
2018
State
Union Territory
Total (All India)

```
df
```

0	1	State	Andhra Pradesh	616	931	1207	4.4	520.3
1	2	State	Arunachal Pradesh	4	1	7	0.0	14.9
2	3	State	Assam	696	1120	2022	7.4	340.4
3	4	State	Bihar	309	433	374	1.4	1183.3
4	5	State	Chhattisgarh	90	171	139	0.5	284.7
5	6	State	Goa	31	13	29	0.1	15.3
6	7	State	Gujarat	362	458	702	2.6	673.2
7	8	State	Haryana	401	504	418	1.5	284.0
8	9	State	Himachal Pradesh	31	56	69	0.3	72.7
9	10	State	Jammu & Kashmir	28	63	73	0.3	134.3
10	11	State	Jharkhand	259	720	930	3.4	370.5
11	12	State	Karnataka	1101	3174	5839	21.4	654.5
12	13	State	Kerala	283	320	340	1.2	350.0
13	14	State	Madhya Pradesh	258	490	740	2.7	814.7
14	15	State	Maharashtra	2380	3604	3511	12.9	1213.9
15	16	State	Manipur	11	74	29	0.1	30.8
16	17	State	Meghalaya	39	39	74	0.3	32.0
17	18	State	Mizoram	1	10	6	0.0	11.8
18	19	State	Nagaland	2	0	2	0.0	21.3
19	20	State	Odisha	317	824	843	3.1	435.5
20	21	State	Punjab	102	176	239	0.9	297.0
21	22	State	Rajasthan	941	1304	1104	4.1	765.9
22	23	State	Sikkim	1	1	1	0.0	6.6
23	24	State	Tamil Nadu	144	228	295	1.1	754.6
24	25	State	Telangana	593	1209	1205	4.4	370.3

```
df.columns
```

```
Index(['S. No', 'Category', 'State/UT', '2016', '2017', '2018',
       'Percentage Share of State/UT (2018)',
       'Mid-Year Projected Population (in Lakhs) (2018)+',
       'Rate of Total Cyber Crimes (2018)++'],
      dtype='object')
```

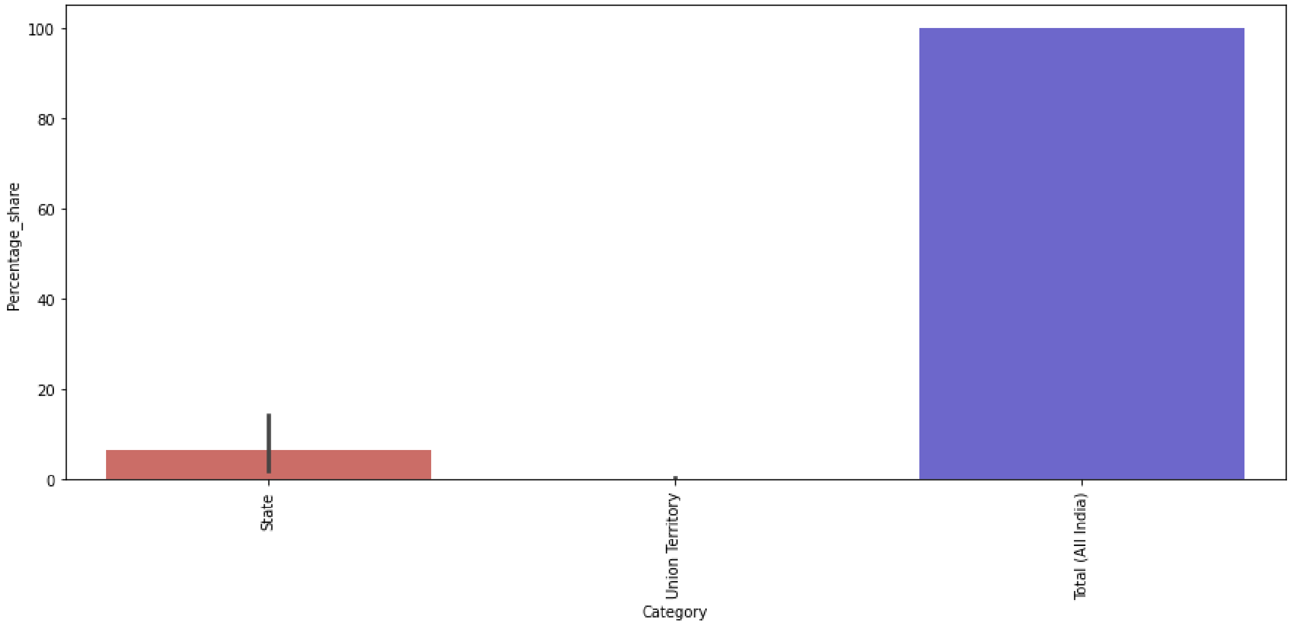
```
df.rename(columns = {'Percentage Share of State/UT (2018)':'Percentage_share',
       'Mid-Year Projected Population (in Lakhs) (2018)+':'Projected_Population',
       'Rate of Total Cyber Crimes (2018)++':'Rate_cyber_crime_2018'},inplace = True
```

```
df.columns
```

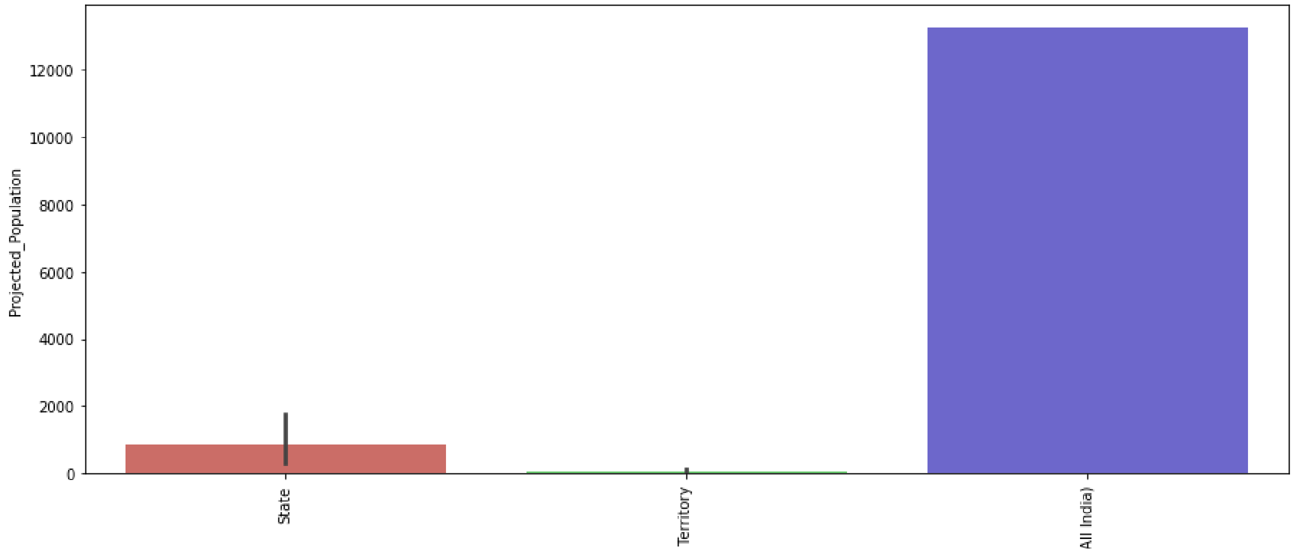
```
Index(['S. No', 'Category', 'State/UT', '2016', '2017', '2018',
       'Percentage_share', 'Projected_Population', 'Rate_cyber_crime_2018'],
      dtype='object')
```

Union

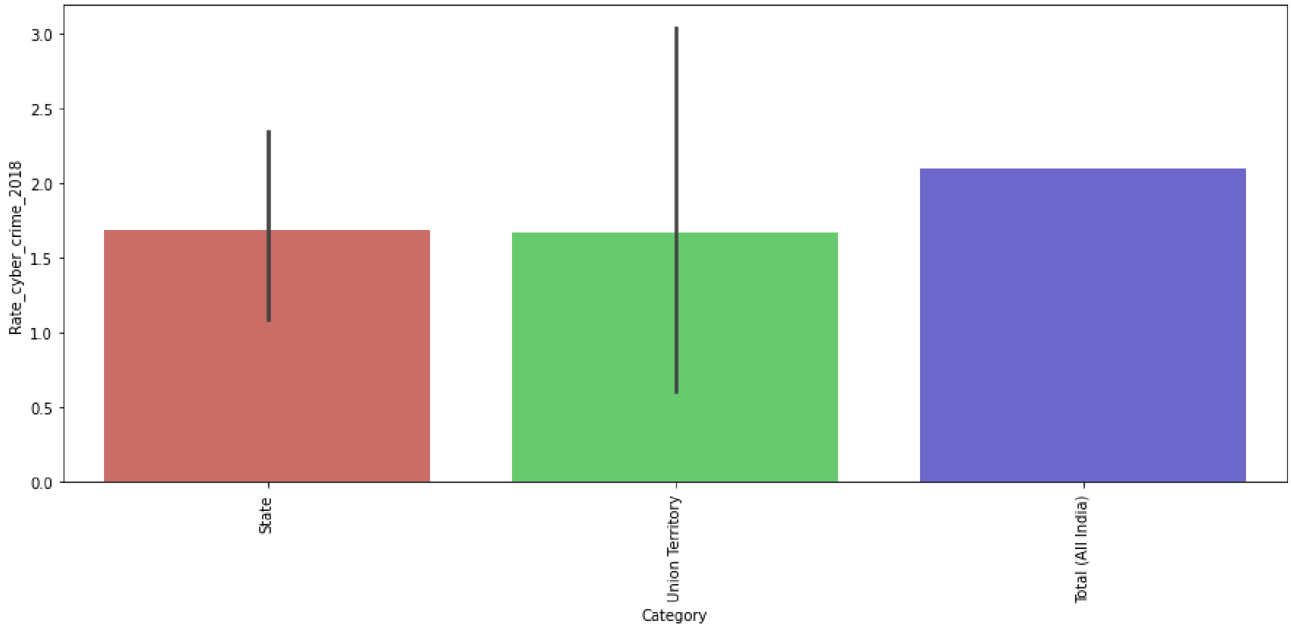
```
plt.figure(figsize =(15,6))
sns.barplot(x = 'Category',y = 'Percentage_share',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



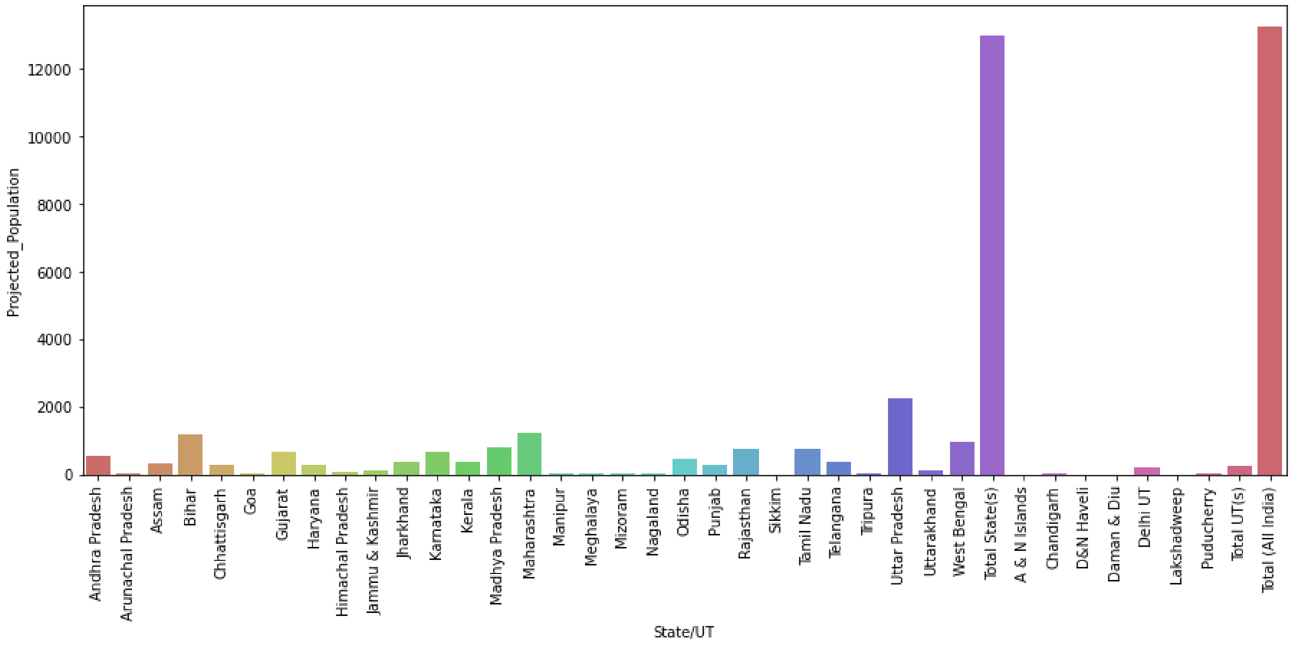
```
plt.figure(figsize =(15,6))
sns.barplot(x = 'Category',y = 'Projected_Population',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



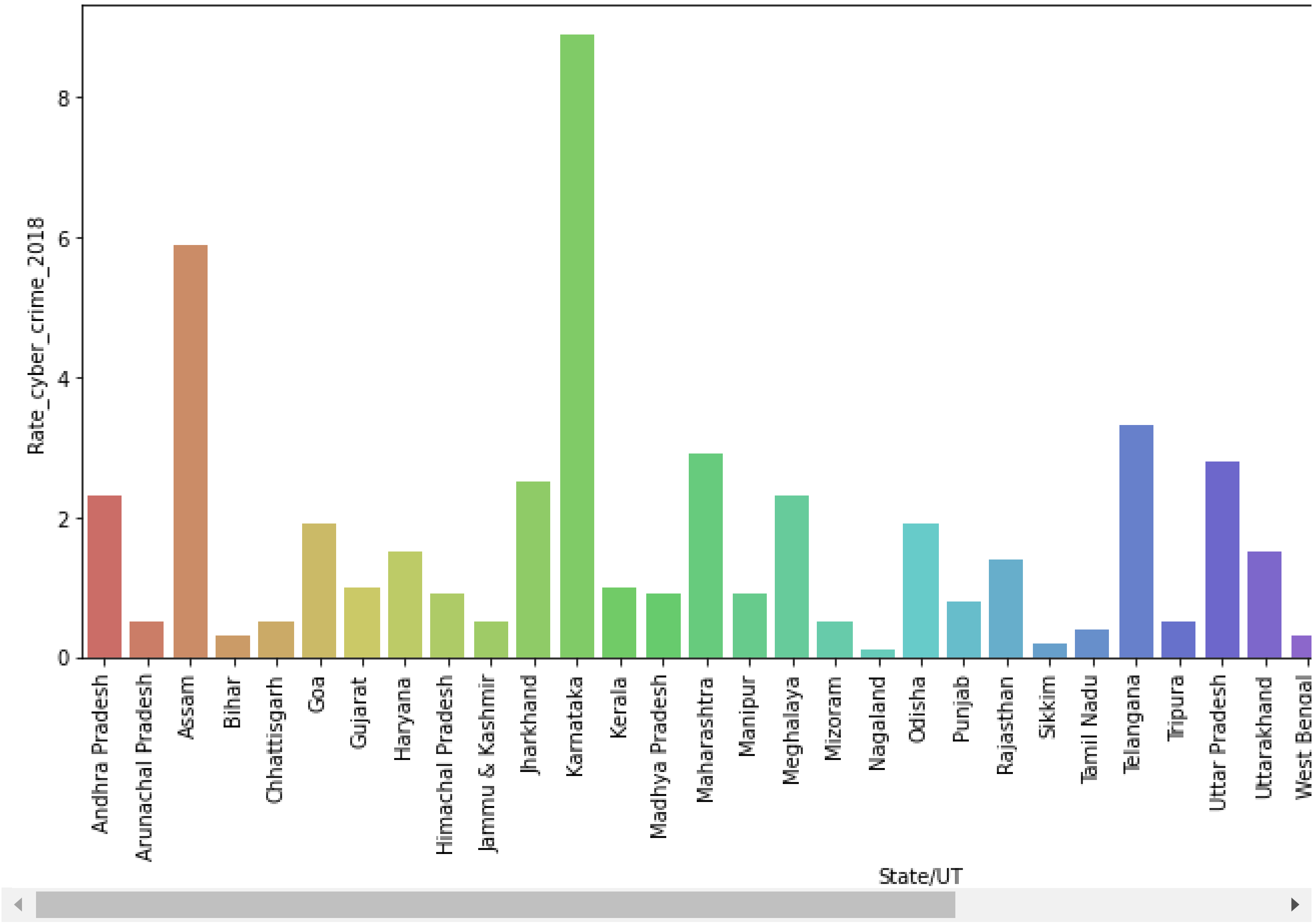
```
plt.figure(figsize =(15,6))
sns.barplot(x = 'Category',y = 'Rate_cyber_crime_2018',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



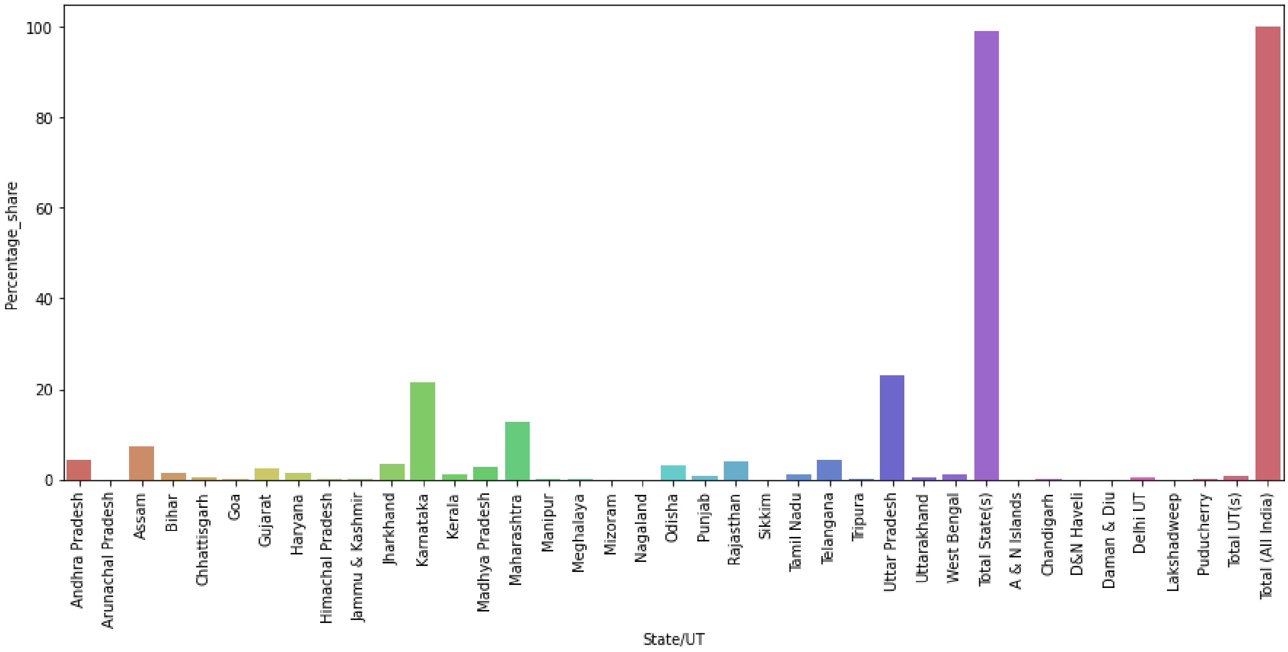
```
plt.figure(figsize =(15,6))
sns.barplot(x = 'State/UT',y ='Projected_Population',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



```
plt.figure(figsize =(15,6))
sns.barplot(x = 'State/UT',y ='Rate_cyber_crime_2018',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



```
plt.figure(figsize =(15,6))
sns.barplot(x = 'State/UT',y ='Percentage_share',data = df,palette='hls')
plt.xticks(rotation=90)
plt.show()
```



```
df['Total'] = df['2016'] + df['2017']+df["2018"]
```

```
df_new = df[['2016','2017','2018']]
```

```
df_new.head()
```

	2016	2017	2018
0	616	931	1207
1	4	1	7
2	696	1120	2022
3	309	433	374
4	90	171	139

```
df_new = df_new.transpose()
```

```
df_new.head()
```

	0	1	2	3	4	5	6	7	8	9	...	29	30	31	32	33	34
2016	616	4	696	309	90	31	362	401	31	28	...	12187	3	26	1	0	98
2017	931	1	1120	433	171	13	458	504	56	63	...	21593	3	32	1	0	162
2018	1207	7	2022	374	139	29	702	418	69	73		27004	7	30	0	0	189

```
df_new = df_new.reset_index()
```

```
df_new = df_new.rename(columns = {'index':'year'})
```

```
df_new.columns
```

```
Index(['year',      0,      1,      2,      3,      4,      5,      6,      7,
            8,      9,     10,     11,     12,     13,     14,     15,     16,
           17,     18,     19,     20,     21,     22,     23,     24,     25,
           26,     27,     28,     29,     30,     31,     32,     33,     34,
           35,     36,     37,     38],
      dtype='object')
```

```
cols = [  0,      1,      2,      3,      4,      5,      6,      7,
            8,      9,     10,     11,     12,     13,     14,     15,     16,
           17,     18,     19,     20,     21,     22,     23,     24,     25,
           26,     27,     28,     29,     30,     31,     32,     33,     34,
           35,     36,     37,     38]
```

```
df_new['total']= df_new[cols].sum(axis = 1)
```

```
plt.figure(figsize =(15,6))
sns.barplot(x = 'year',y ='total',data = df_new,palette='hls')
plt.xticks(rotation=90)
plt.show()
```

Double-click (or enter) to edit

```
df.columns
```

```
Index(['S. No', 'Category', 'State/UT', '2016', '2017', '2018',
       'Percentage_share', 'Projected_Population', 'Rate_cyber_crime_2018',
       'Total'],
      dtype='object')
```

```
X = df.drop(['S. No', 'Category', 'State/UT',
       'Percentage_share', 'Projected_Population', 'Rate_cyber_crime_2018',
       'Total'],axis = 1)
```

```
y = df['Total']
```

```
X.shape
```

```
(39, 3)
```

```
y.shape
```

```
(39,)
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2)
```

```
X_train.shape,y_train.shape,X_test.shape,y_test.shape
```

```
((31, 3), (31,), (8, 3), (8,))
```

```
from sklearn.tree import DecisionTreeRegressor
```

```
regressor = DecisionTreeRegressor(max_depth = 6)
```

```
regressor.fit(X_train,y_train)
```

```
DecisionTreeRegressor(max_depth=6)
```

```
print("Training_Accuracy:",regressor.score(X_train,y_train))
print("Testing_Accuracy:",regressor.score(X_test,y_test))
```

```
Training_Accuracy: 0.9999967254006523
Testing_Accuracy: 0.9252704801031749
```

```
from sklearn.ensemble import RandomForestRegressor
```

```
rf_regressor = RandomForestRegressor(n_estimators=100,random_state=0)
rf_regressor.fit(X_train,y_train)
```

```
RandomForestRegressor(random_state=0)
```

```
print("Training_Accuracy:",rf_regressor.score(X_train,y_train))
print("Testing_Accuracy:",rf_regressor.score(X_test,y_test))
```

```
Training_Accuracy: 0.980921722584463
Testing_Accuracy: 0.747963602167363
```