In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
```

In [5]:
```
data = pd.read_csv('Suicides with reason.csv')
data
```

Out[5]:

	State	Year	Type_code	Type	Gender	Age_group	Total
0	A & N ISLANDS	2001	Causes	Cancer	Male	15-29	0
1	A & N ISLANDS	2001	Causes	Divorce	Male	60+	0
2	A & N ISLANDS	2001	Causes	Dowry Dispute	Female	60+	0
3	A & N ISLANDS	2001	Causes	Ideological Causes/Hero Worshipping	Female	60+	0
4	A & N ISLANDS	2001	Causes	Illness (Aids/STD)	Female	0-14	0
...	...	...	...	...	...	...	...
237356	WEST BENGAL	2012	Professional_Profile	Professional Activity	Male	60+	0
237357	WEST BENGAL	2012	Professional_Profile	Self-employed (Business activity)	Male	0-14	0
237358	WEST BENGAL	2012	Professional_Profile	Service (Government)	Male	15-29	0
237359	WEST BENGAL	2012	Professional_Profile	Service (Government)	Male	60+	0
237360	WEST BENGAL	2012	Social_Status	Never Married	Male	0-100+	2658

237361 rows × 7 columns

In [6]:
```
data.head()
```

Out[6]:

	State	Year	Type_code	Type	Gender	Age_group	Total
0	A & N ISLANDS	2001	Causes	Cancer	Male	15-29	0
1	A & N ISLANDS	2001	Causes	Divorce	Male	60+	0
2	A & N ISLANDS	2001	Causes	Dowry Dispute	Female	60+	0
3	A & N ISLANDS	2001	Causes	Ideological Causes/Hero Worshipping	Female	60+	0
4	A & N ISLANDS	2001	Causes	Illness (Aids/STD)	Female	0-14	0

In [9]:
```
data.tail()
```

Out[9]:

	State	Year	Type_code	Type	Gender	Age_group	Total
237356	WEST BENGAL	2012	Professional_Profile	Professional Activity	Male	60+	0
237357	WEST BENGAL	2012	Professional_Profile	Self-employed (Business activity)	Male	0-14	0
237358	WEST BENGAL	2012	Professional_Profile	Service (Government)	Male	15-29	0
237359	WEST BENGAL	2012	Professional_Profile	Service (Government)	Male	60+	0
237360	WEST BENGAL	2012	Social_Status	Never Married	Male	0-100+	2658

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 237361 entries, 0 to 237360
Data columns (total 7 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   State       237361 non-null  object
 1   Year        237361 non-null  int64 
 2   Type_code   237361 non-null  object
 3   Type        237361 non-null  object
 4   Gender      237361 non-null  object
 5   Age_group   237361 non-null  object
 6   Total       237361 non-null  int64 
dtypes: int64(2), object(5)
memory usage: 12.7+ MB
```

In [13]:
```
data.describe()
```

Out[13]:

	Year	Total
count	237361.000000	237361.000000
mean	2006.500508	50.396548
std	3.452195	707.467748
min	2001.000000	0.000000
25%	2004.000000	0.000000
50%	2007.000000	0.000000
75%	2010.000000	6.000000
max	2012.000000	63343.000000

In [14]:
```
data.shape
```

Out[14]:
```
(237361, 7)
```

In [15]:
```
data.columns
```

Out[15]:
```
Index(['State', 'Year', 'Type_code', 'Type', 'Gender', 'Age_group', 'Total'],
dtype='object')
```

In [16]:
```
data.nunique()
```

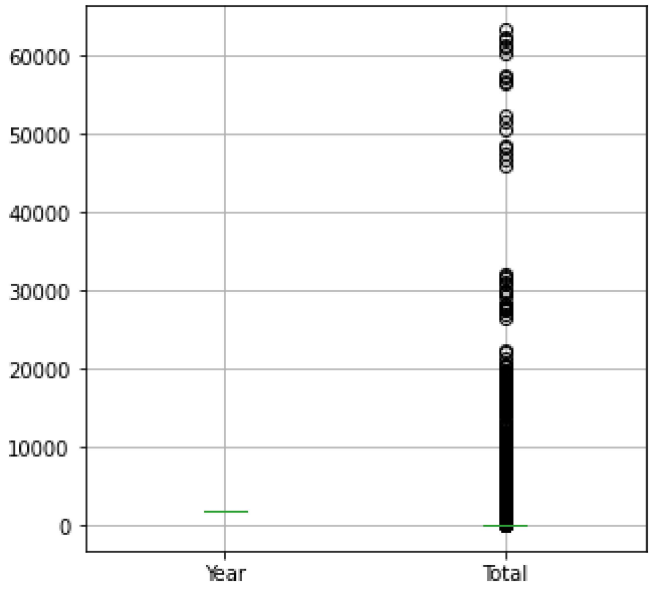
Out[16]:
```
State          38
Year           12
Type_code       5
Type           69
Gender          2
Age_group       6
Total        2111
dtype: int64
```

In [17]:
```
# Cleaning our dataset
```

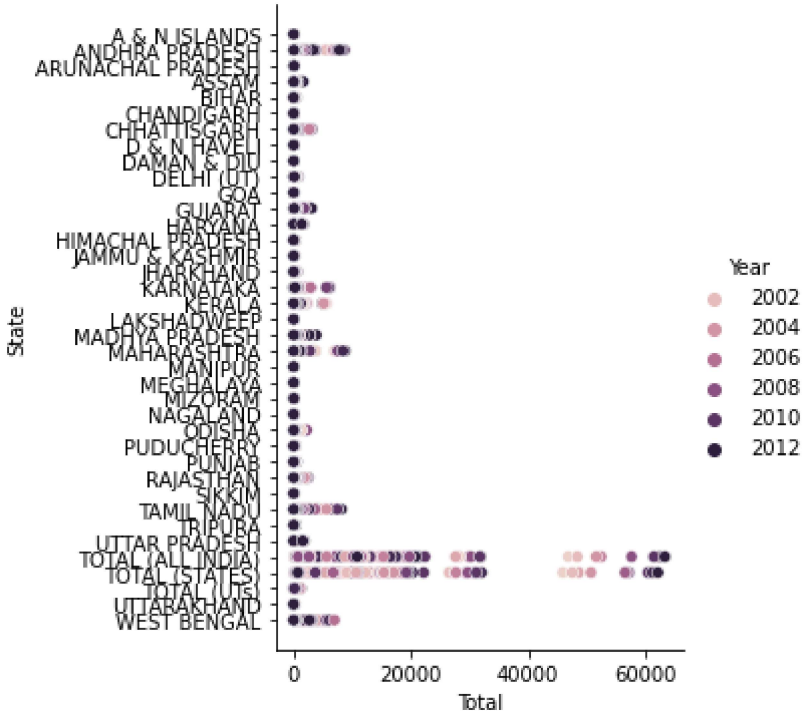
In [19]:
```
data.isnull().sum()
```

Out[19]:
```
State        0
Year         0
Type_code    0
Type         0
Gender       0
Age_group    0
Total        0
dtype: int64
```

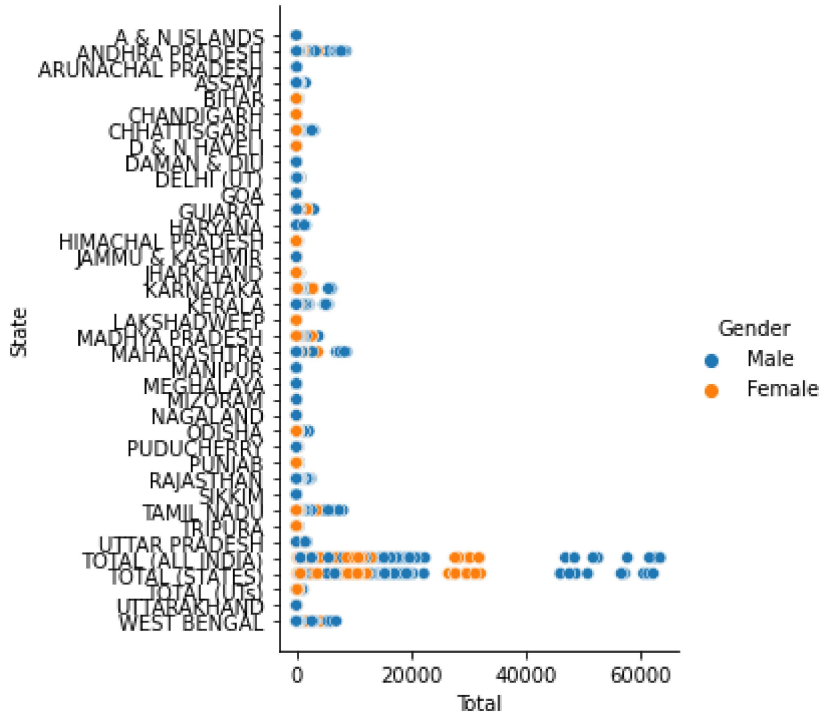
In [22]:
```
data.boxplot(figsize=(5,5))
plt.show()
```



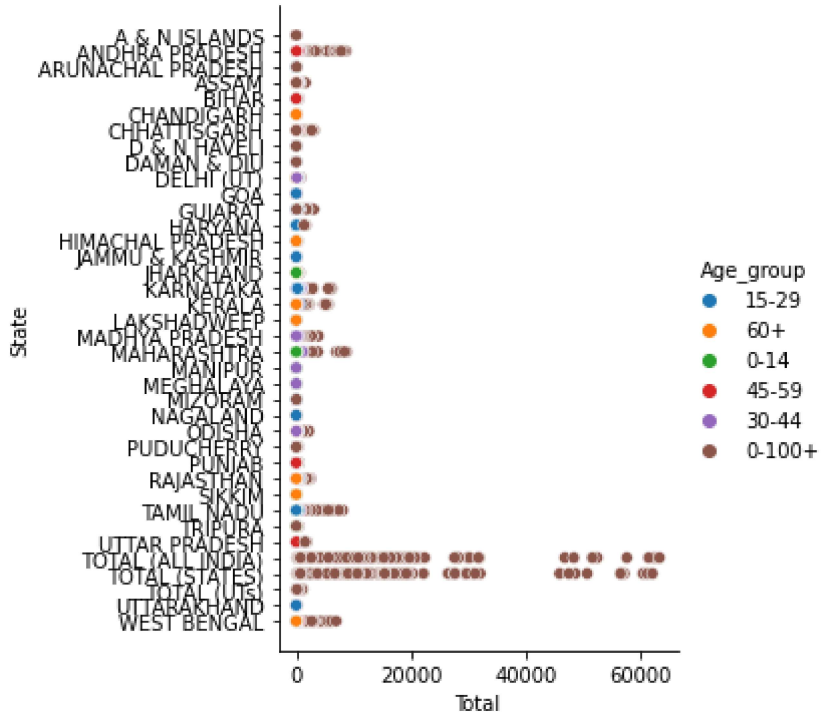
In [26]:
```
import seaborn as sns
sns.relplot(x="Total",y="State",hue="Year",data=data)
plt.show()
```



In [25]:
```
sns.relplot(x="Total",y="State",hue="Gender",data=data)
plt.show()
```



In [27]:
```
sns.relplot(x="Total",y="State",hue="Age_group",data=data)
plt.show()
```



In [29]:
```
index = data['Gender']=='Female'
data[index].groupby('Year').Total.sum()
```

Out[29]:
```
Year
2001    356253
2002    341512
2003    355586
2004    346672
2005    342231
2006    370826
2007    271682
2008    395534
2009    367126
2010    400571
2011    383042
2012    359515
Name: Total, dtype: int64
```

In [35]:
```
newdata = data.groupby(['State','Age_group','Gender','Year',]).Total.sum().dro
p(labels='TOTAL (ALL INDIA)').drop(labels='TOTAL (STATES)').drop(labels='TOTAL
(UTs)')
newdata
```

Out[35]:
```
State          Age_group  Gender  Year
A & N ISLANDS  0-100+     Female  2001     100
                                  2002     106
                                  2003      86
                                  2004      82
                                  2005     106
                                           ...
WEST BENGAL    60+        Male    2008    1104
                                  2009    1389
                                  2010    1674
                                  2011    2193
                                  2012     665
Name: Total, Length: 5040, dtype: int64
```

In [36]:
```
newdata.to_csv('stateyear.csv')
```

In [37]:
```
reloaddata = pd.read_csv('stateyear.csv')
reloaddata
```

Out[37]:

	State	Age_group	Gender	Year	Total
0	A & N ISLANDS	0-100+	Female	2001	100
1	A & N ISLANDS	0-100+	Female	2002	106
2	A & N ISLANDS	0-100+	Female	2003	86
3	A & N ISLANDS	0-100+	Female	2004	82
4	A & N ISLANDS	0-100+	Female	2005	106
...	...	...	...	...	...
5035	WEST BENGAL	60+	Male	2008	1104
5036	WEST BENGAL	60+	Male	2009	1389
5037	WEST BENGAL	60+	Male	2010	1674
5038	WEST BENGAL	60+	Male	2011	2193
5039	WEST BENGAL	60+	Male	2012	665

5040 rows × 5 columns

In [38]:
```
newdata2 = reloaddata.groupby(['Age_group','State','Gender','Year']).Total.sum
().drop(labels='0-100+')
newdata2
```

Out[38]:
```
Age_group  State          Gender  Year
0-14       A & N ISLANDS  Female  2001       6
                                  2002       6
                                  2003      12
                                  2004       3
                                  2005       9
                                             ...
60+        WEST BENGAL    Male    2008    1104
                                  2009    1389
                                  2010    1674
                                  2011    2193
                                  2012     665
Name: Total, Length: 4200, dtype: int64
```

In [39]:
```
newdata2.to_csv('yearcount.csv')
```

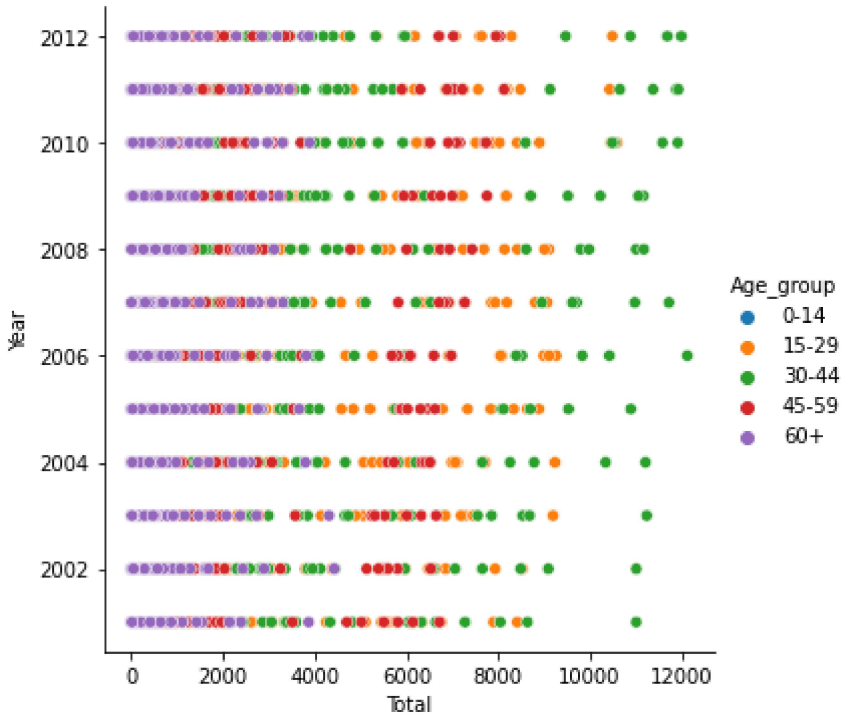
In [40]:
```
reloaddata2 = pd.read_csv('yearcount.csv')
reloaddata2
```

Out[40]:

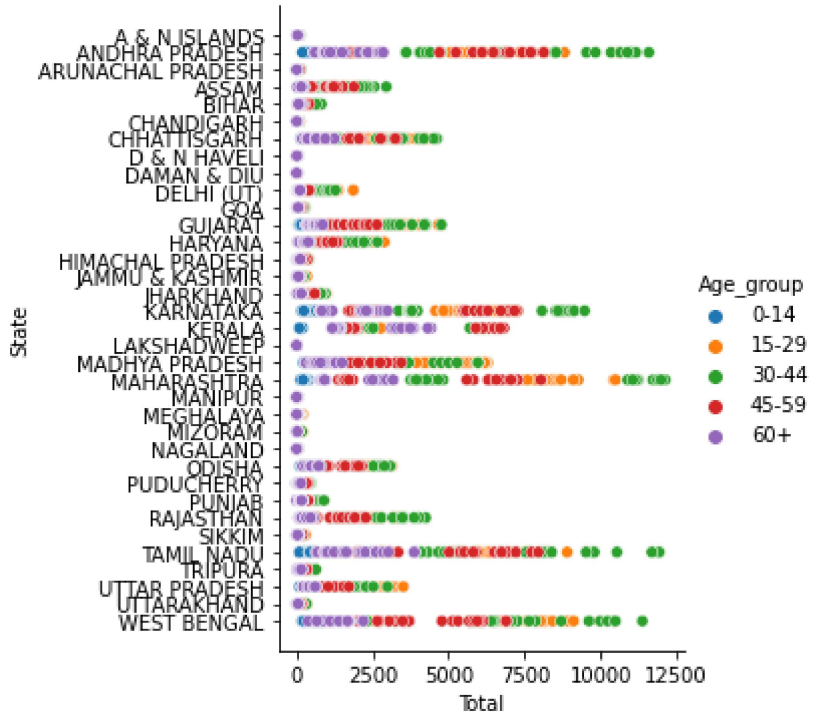
	Age_group	State	Gender	Year	Total
0	0-14	A & N ISLANDS	Female	2001	6
1	0-14	A & N ISLANDS	Female	2002	6
2	0-14	A & N ISLANDS	Female	2003	12
3	0-14	A & N ISLANDS	Female	2004	3
4	0-14	A & N ISLANDS	Female	2005	9
...	...	...	...	...	...
4195	60+	WEST BENGAL	Male	2008	1104
4196	60+	WEST BENGAL	Male	2009	1389
4197	60+	WEST BENGAL	Male	2010	1674
4198	60+	WEST BENGAL	Male	2011	2193
4199	60+	WEST BENGAL	Male	2012	665

4200 rows × 5 columns

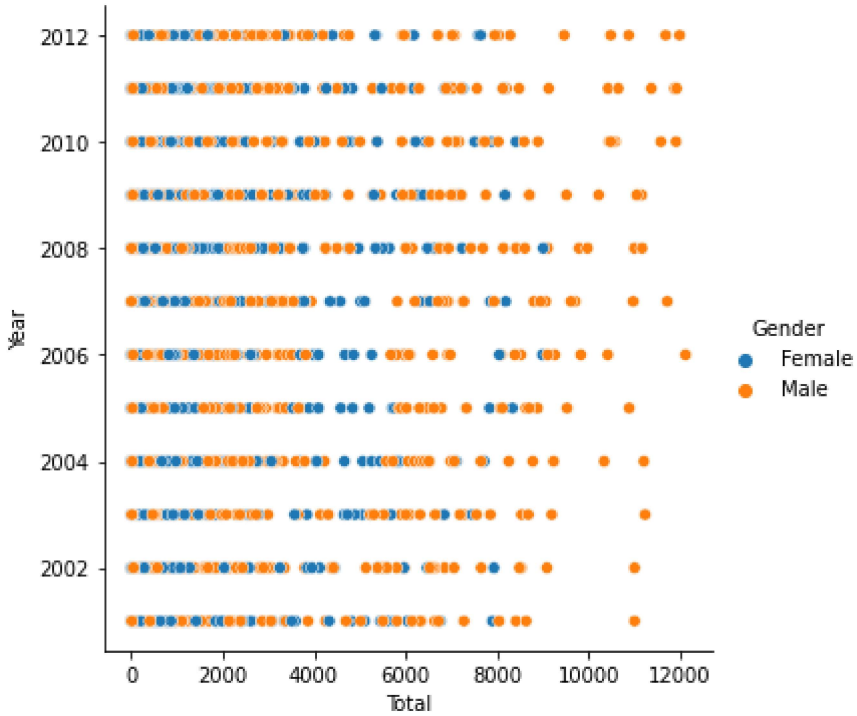
In [42]:
```
sns.relplot(x="Total",y="Year",hue="Age_group",data=reloaddata2)
plt.show()
```



In [44]:
```
sns.relplot(x="Total",y="State",hue="Age_group",data=reloaddata2)
plt.show()
```

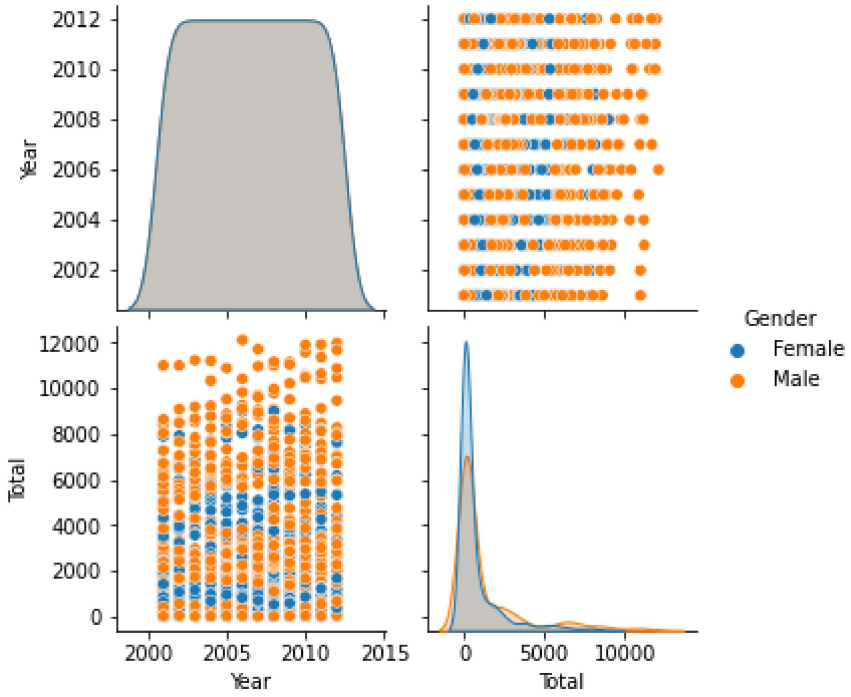


In [45]:
```
sns.relplot(x="Total",y="Year",hue="Gender",data=reloaddata2)
plt.show()
```



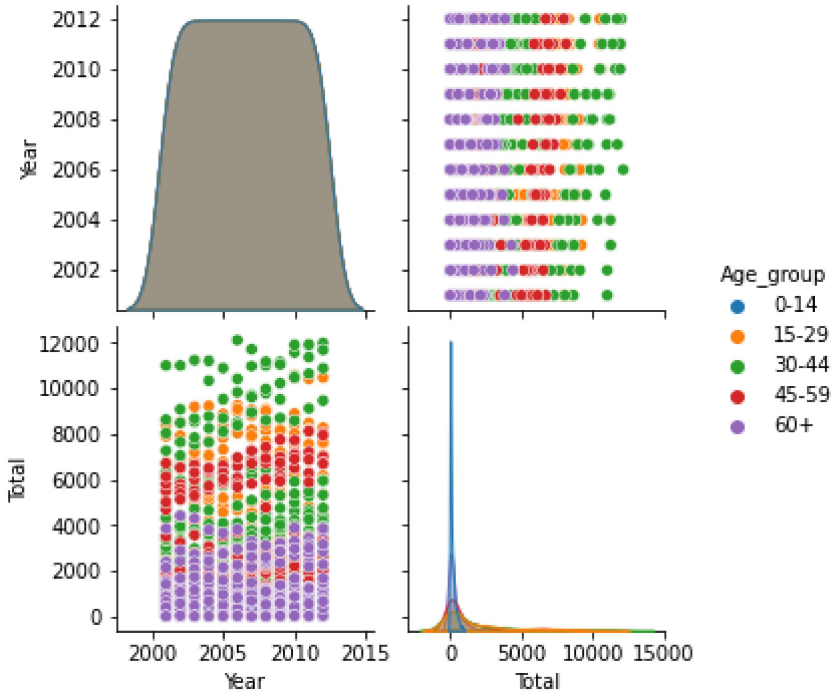
In [47]:
```
sns.pairplot(reloaddata2,hue='Gender')
```

Out[47]: <seaborn.axisgrid.PairGrid at 0x7f6378420610>



In [49]:
```
sns.pairplot(reloaddata2,hue='Age_group')
```

Out[49]: <seaborn.axisgrid.PairGrid at 0x7f6369c26510>



In [50]:
```
data1 = reloaddata2.groupby('Year').Total.sum()
data1
```

Out[50]:
```
Year
2001    325428
2002    331146
2003    332516
2004    340955
2005    341717
2006    354319
2007    367845
2008    374980
2009    381127
2010    403728
2011    405989
2012    376398
Name: Total, dtype: int64
```

In [51]:
```
x1 = data1.index.values.reshape(-1,1)
x1
```

Out[51]:
```
array([[2001],
       [2002],
       [2003],
       [2004],
       [2005],
       [2006],
       [2007],
       [2008],
       [2009],
       [2010],
       [2011],
       [2012]])
```

In [52]:
```
y1 = data1.values.reshape(-1,1)
y1
```

Out[52]:
```
array([[325428],
       [331146],
       [332516],
       [340955],
       [341717],
       [354319],
       [367845],
       [374980],
       [381127],
       [403728],
       [405989],
       [376398]])
```

In [55]:
```
from sklearn.linear_model import LinearRegression
reg_all_india = LinearRegression()
reg_all_india.fit(x1,y1)
```

Out[55]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fals
e)
```

In [56]:
```
reg_all_india.coef_
```

Out[56]:
```
array([[7157.04895105]])
```

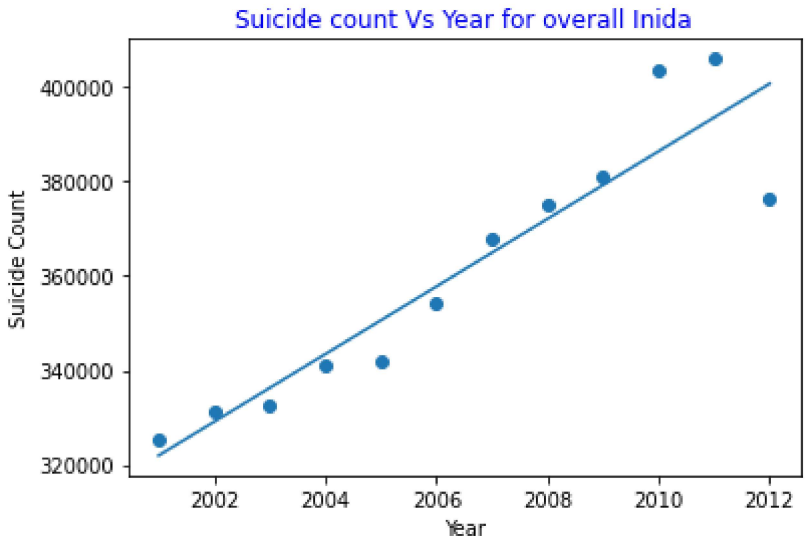
In [59]:
```
reg_all_india.intercept_
```

Out[59]:
```
array([-13999273.05361305])
```

In [60]:
```
result_all_india = reg_all_india.predict(x1)
result_all_india
```

Out[60]:
```
array([[321981.8974359 ],
       [329138.94638695],
       [336295.99533799],
       [343453.04428904],
       [350610.09324009],
       [357767.14219114],
       [364924.19114219],
       [372081.24009324],
       [379238.28904429],
       [386395.33799534],
       [393552.38694639],
       [400709.43589743]])
```

In [83]:
```
import matplotlib.pyplot as plt
plt.scatter(x1,y1)
plt.xlabel('Year')
plt.ylabel('Suicide Count')
plt.title(label="Suicide count Vs Year for overall Inida",
          fontsize=12,
          color="blue")
plt.plot(x1,result_all_india)
plt.show()
```



In [65]:
```
reg_all_india.score(x1,y1) * 100
```

Out[65]:
```
85.98369710042914
```

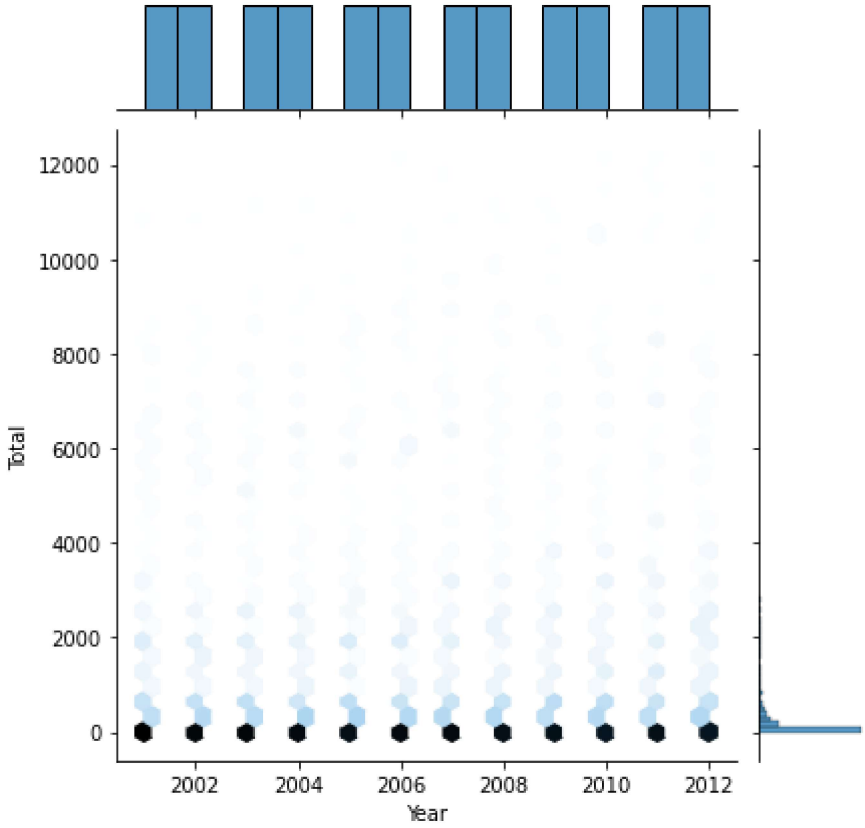
In [66]:
```
import joblib
joblib.dump(reg_all_india, 'All_India_model1.pkl', compress=9)
```

Out[66]:
```
['All_India_model1.pkl']
```

In [68]:
```
reg_all_india.predict([[2022]])
```

Out[68]:
```
array([[472279.92540792]])
```

In [69]:
```
import seaborn
import matplotlib.pyplot as plt
seaborn.jointplot(x=reloaddata2['Year'], y=reloaddata2['Total'], kind='hex')
# to show
plt.show()
```



In [71]:
```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
poly=PolynomialFeatures(degree=5)
x1 = data1.index.values.reshape(-1,1)
xtrans=poly.fit_transform(x1)
pd.DataFrame(xtrans)
```

Out[71]:

	0	1	2	3	4	5
0	1.0	2001.0	4004001.0	8.012006e+09	1.603202e+13	3.208008e+16
1	1.0	2002.0	4008004.0	8.024024e+09	1.606410e+13	3.216032e+16
2	1.0	2003.0	4012009.0	8.036054e+09	1.609622e+13	3.224072e+16
3	1.0	2004.0	4016016.0	8.048096e+09	1.612838e+13	3.232128e+16
4	1.0	2005.0	4020025.0	8.060150e+09	1.616060e+13	3.240201e+16
5	1.0	2006.0	4024036.0	8.072216e+09	1.619287e+13	3.248289e+16
6	1.0	2007.0	4028049.0	8.084294e+09	1.622518e+13	3.256393e+16
7	1.0	2008.0	4032064.0	8.096385e+09	1.625754e+13	3.264514e+16
8	1.0	2009.0	4036081.0	8.108487e+09	1.628995e+13	3.272651e+16
9	1.0	2010.0	4040100.0	8.120601e+09	1.632241e+13	3.280804e+16
10	1.0	2011.0	4044121.0	8.132727e+09	1.635491e+13	3.288973e+16
11	1.0	2012.0	4048144.0	8.144866e+09	1.638747e+13	3.297159e+16

In [72]:
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(xtrans,y1)
```

Out[72]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fals
e)
```

In [73]:
```
y1
```

Out[73]:
```
array([[325428],
       [331146],
       [332516],
       [340955],
       [341717],
       [354319],
       [367845],
       [374980],
       [381127],
       [403728],
       [405989],
       [376398]])
```

In [74]:
```
reg.coef_
```

Out[74]:
```
array([[ 0.00000000e+00, -8.58249983e-02, -2.28546526e-01,
        -2.29288963e+02,  1.71428267e-01, -3.41781393e-05]])
```

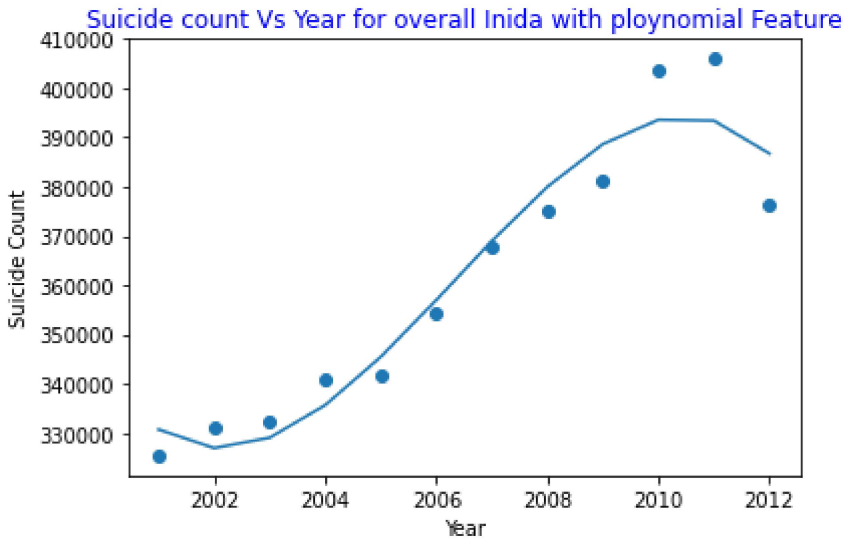
In [77]:
```
reg.intercept_
```

Out[77]:
```
array([1.85161148e+11])
```

In [78]:
```
Yp=reg.predict(xtrans)
Yp
```

Out[78]:
```
array([[330708.23236084],
       [326935.43939209],
       [329040.76800537],
       [335664.05340576],
       [345441.03045654],
       [357003.32781982],
       [368978.46624756],
       [379989.85321045],
       [388656.77923584],
       [393594.41204834],
       [393413.7958374 ],
       [386721.84490967]])
```

In [84]:
```
import matplotlib.pyplot as plt
plt.scatter(x1,y1)
plt.xlabel('Year')
plt.ylabel('Suicide Count')
plt.title(label="Suicide count Vs Year for overall Inida with ploynomial Featu
re",
          fontsize=12,
          color="blue")
plt.plot(x1,Yp)
plt.show()
```



In [87]:
```
reg.score(xtrans,y1) * 100
```

Out[87]:
```
93.45880902419331
```

In [86]:
```
import joblib
joblib.dump(reg, 'All_India_model2.pkl', compress=9)
```

Out[86]:
```
['All_India_model2.pkl']
```

In [98]:
```
x1 = data1.index.values.reshape(-1,1)
y1 = data1.values.reshape(-1,1)
X_train, X_test, y_train, y_test = train_test_split(
x1, y1, test_size = 0.3, random_state = 100)
```

In [99]:
```
from sklearn.linear_model import LinearRegression
reg_all_india_split = LinearRegression()
reg_all_india_split.fit(X_train,y_train)
```

Out[99]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fals
e)
```

In [100]:
```
reg_all_india_split.coef_
```

Out[100]:
```
array([[6679.54721274]])
```

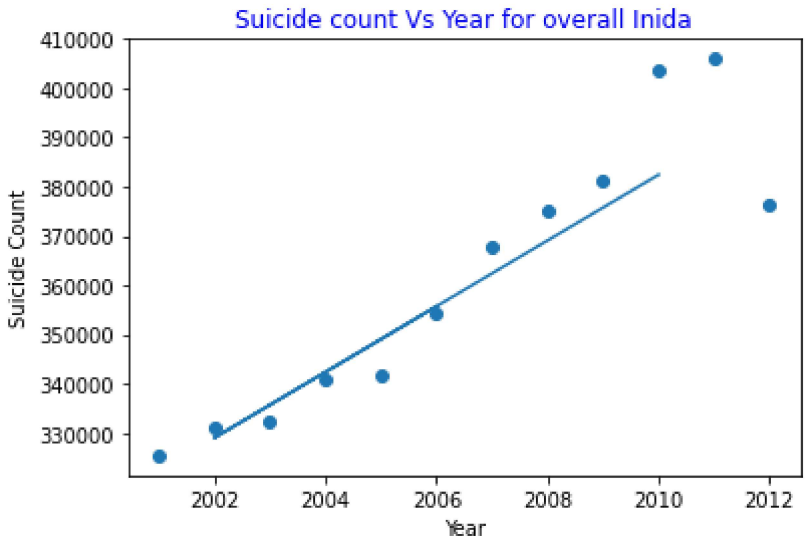
In [101]:
```
reg_all_india_split.intercept_
```

Out[101]:
```
array([-13043457.67576792])
```

In [102]:
```
predict2=reg_all_india_split.predict(X_test)
predict2
```

Out[102]:
```
array([[382432.221843  ],
       [362393.58020478],
       [328995.84414107],
       [355714.03299204]])
```

In [104]:
```
import matplotlib.pyplot as plt
plt.scatter(x1,y1)
plt.xlabel('Year')
plt.ylabel('Suicide Count')
plt.title(label="Suicide count Vs Year for overall Inida",
          fontsize=12,
          color="blue")
plt.plot(X_test,predict2)
plt.show()
```



In [105]:
```
reg_all_india_split.score(X_test,y_test) * 100
```

Out[105]:
```
82.29179991656218
```

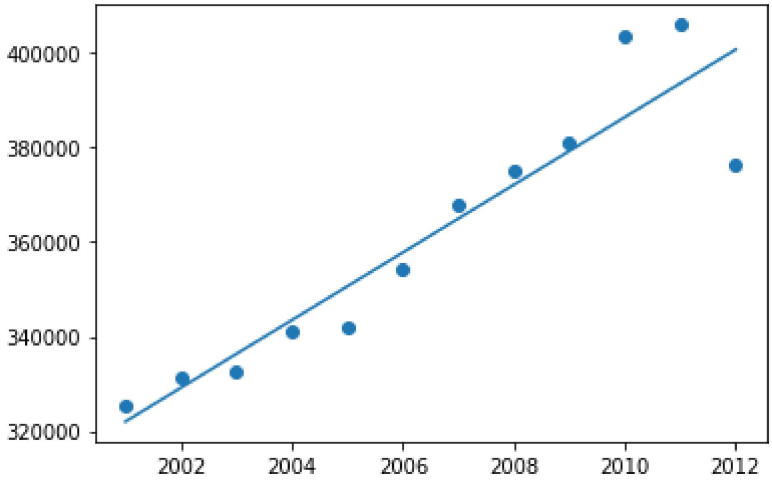
In [107]:
```
import joblib
joblib.dump(reg_all_india_split, 'All_India_model3.pkl', compress=9)
```

Out[107]:
```
['All_India_model3.pkl']
```

In [110]:
```
data1 = reloaddata2.groupby('Year').Total.sum()
print('Data Set :')
print(data1)
x1 = data1.index.values.reshape(-1,1)
print('x coordinated :')
print(x1)

y1 = data1.values.reshape(-1,1)
print('y coordinated :')
print(y1)

from sklearn.linear_model import LinearRegression
reg_all_india = LinearRegression()
reg_all_india.fit(x1,y1)
print('Coefficent Value :')
print(reg_all_india.coef_)
print('Intercept Value :')
print(reg_all_india.intercept_)

result_all_india = reg_all_india.predict(x1)

print('Predicted Value :')
print(result_all_india)
plt.scatter(x1,y1)
print('Plotting graph :')
plt.plot(x1,result_all_india)
plt.show()
print('Model Score % :')
print(reg_all_india.score(x1,y1) * 100)
print('Predicted value for 2022 :')
print(reg_all_india.predict([[2022]]))
```

```
Data Set :
Year
2001    325428
2002    331146
2003    332516
2004    340955
2005    341717
2006    354319
2007    367845
2008    374980
2009    381127
2010    403728
2011    405989
2012    376398
Name: Total, dtype: int64
x coordinated :
[[2001]
 [2002]
 [2003]
 [2004]
 [2005]
 [2006]
 [2007]
 [2008]
 [2009]
 [2010]
 [2011]
 [2012]]
y coordinated :
[[325428]
 [331146]
 [332516]
 [340955]
 [341717]
 [354319]
 [367845]
 [374980]
 [381127]
 [403728]
 [405989]
 [376398]]
Coefficent Value :
[[7157.04895105]]
Intercept Value :
[-13999273.05361305]
Predicted Value :
[[321981.8974359 ]
 [329138.94638695]
 [336295.99533799]
 [343453.04428904]
 [350610.09324009]
 [357767.14219114]
 [364924.19114219]
 [372081.24009324]
 [379238.28904429]
 [386395.33799534]
 [393552.38694639]
```

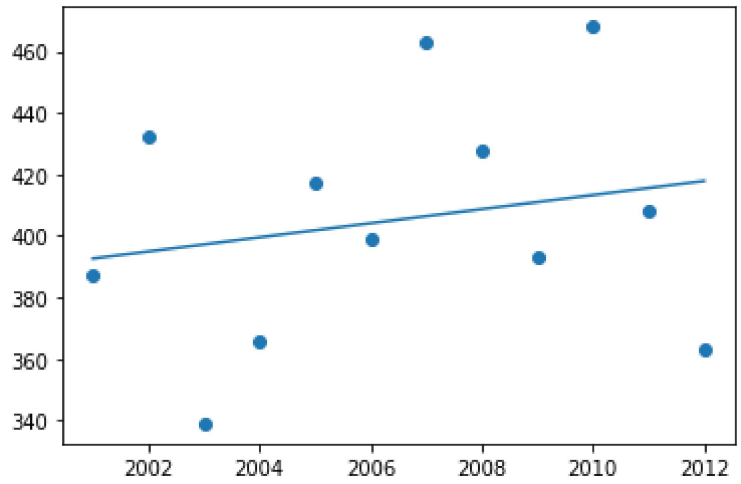
```
 [400709.43589743]]
Plotting graph :
```




```
Model Score % :
85.98369710042914
Predicted value for 2022 :
[[472279.92540792]]
```

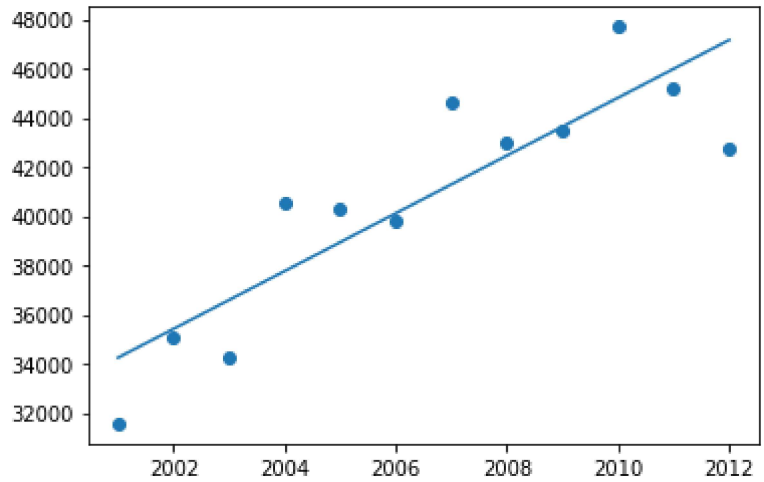
In [113]:
```
statelist = ['A & N ISLANDS' ,'ANDHRA PRADESH' ,'ARUNACHAL PRADESH' ,'ASSAM' ,
'BIHAR' ,'CHANDIGARH' ,'CHHATTISGARH' ,'D & N HAVELI' ,'DAMAN & DIU' ,
             'DELHI (UT)' ,'GOA' ,'GUJARAT' ,'HARYANA' ,'HIMACHAL PRADESH' ,'J
AMMU & KASHMIR' ,'JHARKHAND' ,'KARNATAKA' ,'KERALA' ,'LAKSHADWEEP' ,
             'MADHYA PRADESH' ,'MAHARASHTRA' ,'MANIPUR' ,'MEGHALAYA' ,'MIZORA
M' ,'NAGALAND' ,'ODISHA' ,'PUDUCHERRY' ,'PUNJAB' ,'RAJASTHAN' ,'SIKKIM' ,
             'TAMIL NADU' ,'TRIPURA' ,'UTTAR PRADESH' ,'UTTARAKHAND', 'WEST BE
NGAL' ]
for i in statelist:
  index = reloaddata2['State']==i
  data2 = reloaddata2[index].groupby('Year').Total.sum()
  #print('Data Set :')
  #print(data2)
  print("\n\n")
  print(i)
  x2 = data2.index.values.reshape(-1,1)
  #print('x coordinated :')
  #print(x2)

  y2 = data2.values.reshape(-1,1)
  #print('y coordinated :')
  #print(y2)

  from sklearn.linear_model import LinearRegression
  reg_gender = LinearRegression()
  reg_gender.fit(x2,y2)
  #print('Coefficent Value :')
  #print(reg_gender.coef_)
  #print('Intercept Value :')
  #print(reg_gender.intercept_)

  result_gender = reg_gender.predict(x2)

  #print('Predicted Value :')
  #print(result_gender)
  plt.scatter(x2,y2)
  print('Plotting graph :')
  plt.plot(x2,result_gender)
  plt.show()
  print('Model Score % :')
  print(reg_gender.score(x2,y2) * 100)
  print('Predicted value for 2022 :')
  print(reg_gender.predict([[2022]]))
  name = i + "_model.pkl"
  import joblib
  joblib.dump(reg_gender, name, compress=9)
```

```
A & N ISLANDS
Plotting graph :
```



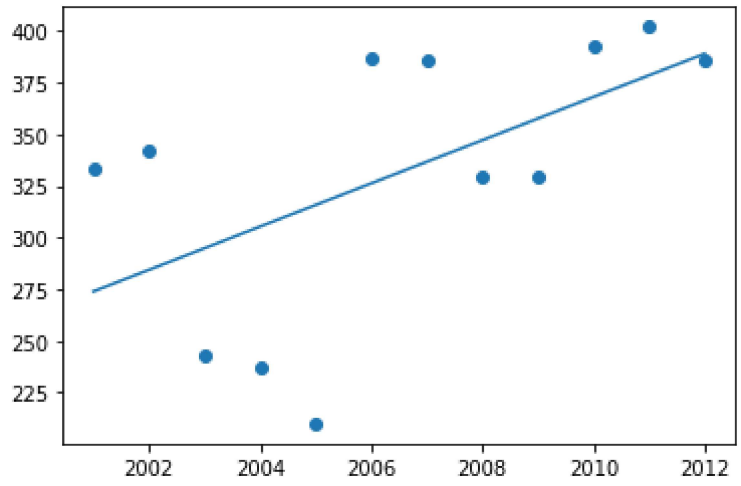

```
Model Score % :
4.441216267485482
Predicted value for 2022 :
[[440.74825175]]
```

```
ANDHRA PRADESH
Plotting graph :
```



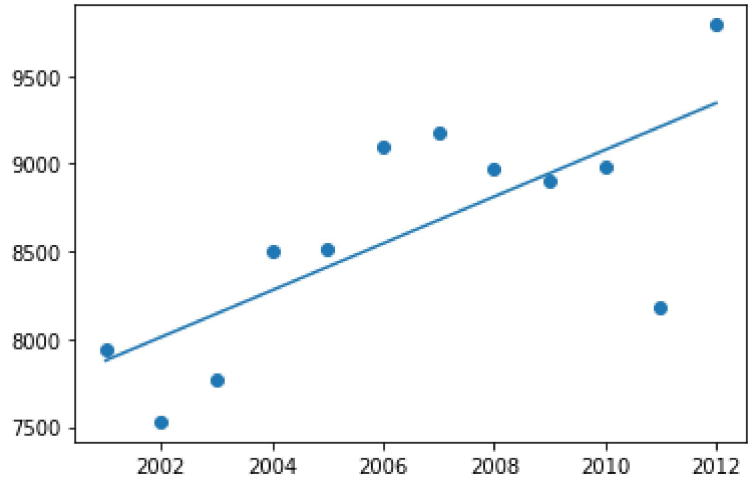

```
Model Score % :
75.76227778000121
Predicted value for 2022 :
[[58902.75641026]]
```

```
ARUNACHAL PRADESH
Plotting graph :
```



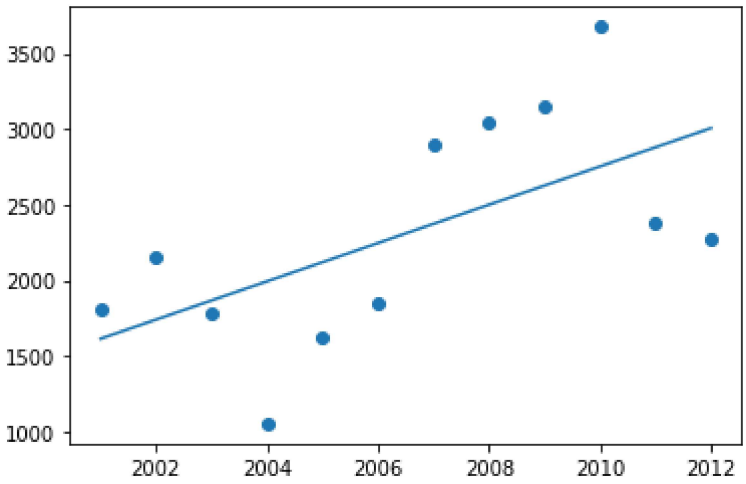

```
Model Score % :
31.602935866140548
Predicted value for 2022 :
[[493.40792541]]
```

```
ASSAM
Plotting graph :
```



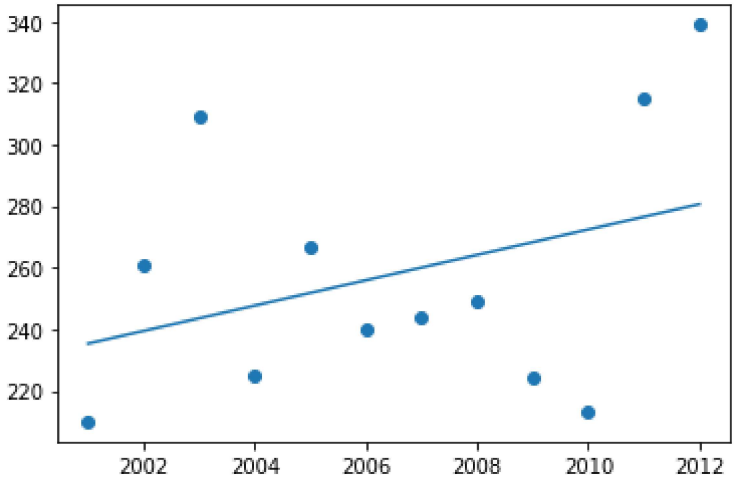

```
Model Score % :
52.61493224139879
Predicted value for 2022 :
[[10677.87762238]]
```

```
BIHAR
Plotting graph :
```



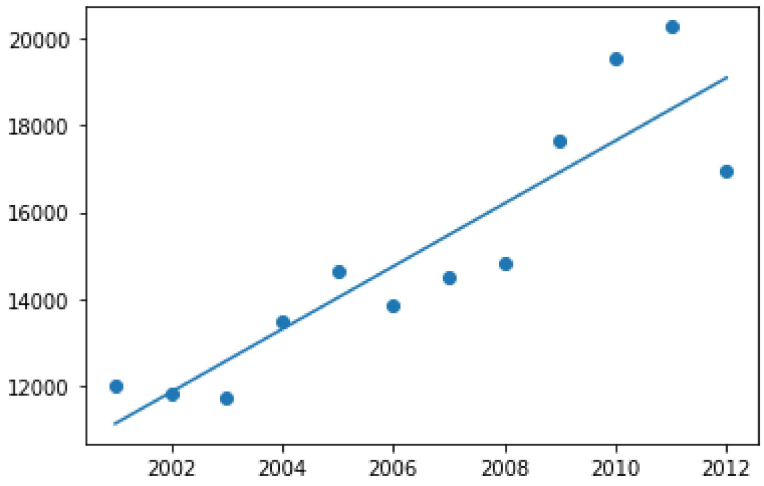

```
Model Score % :
36.54468322321801
Predicted value for 2022 :
[[4272.54662005]]
```

```
CHANDIGARH
Plotting graph :
```



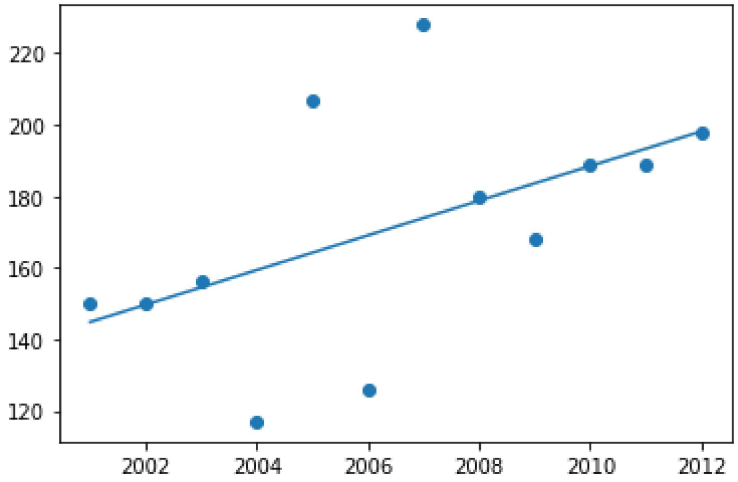

```
Model Score % :
12.329848439829782
Predicted value for 2022 :
[[321.84265734]]
```

```
CHHATTISGARH
Plotting graph :
```



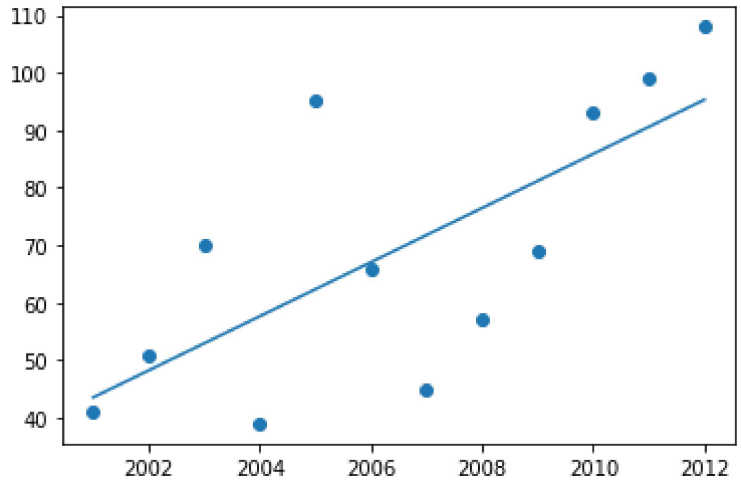

```
Model Score % :
80.96138805166831
Predicted value for 2022 :
[[26316.04778555]]
```

```
D & N HAVELI
Plotting graph :
```



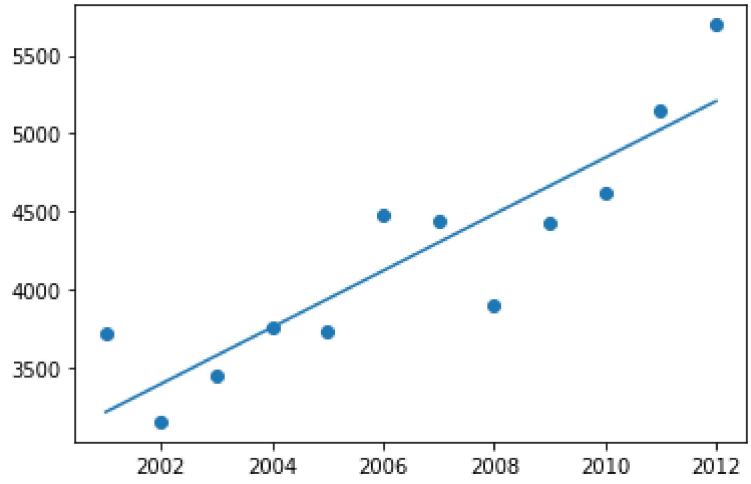

```
Model Score % :
27.854230864929143
Predicted value for 2022 :
[[246.61538462]]
```

```
DAMAN & DIU
Plotting graph :
```



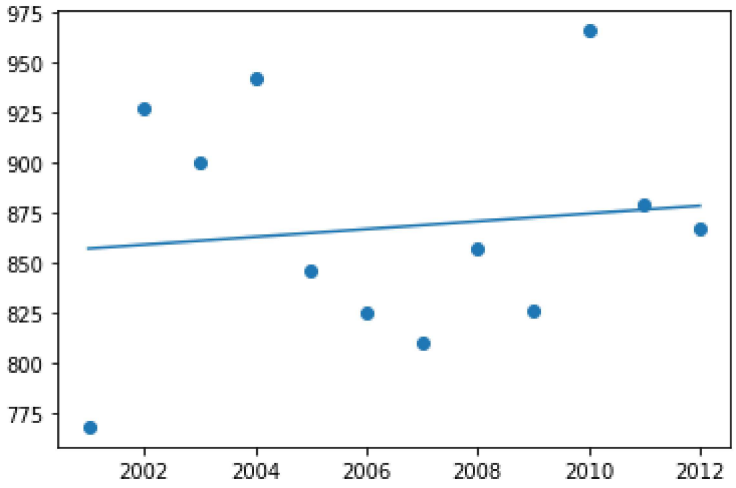

```
Model Score % :
49.347359052587464
Predicted value for 2022 :
[[142.31002331]]
```

```
DELHI (UT)
Plotting graph :
```



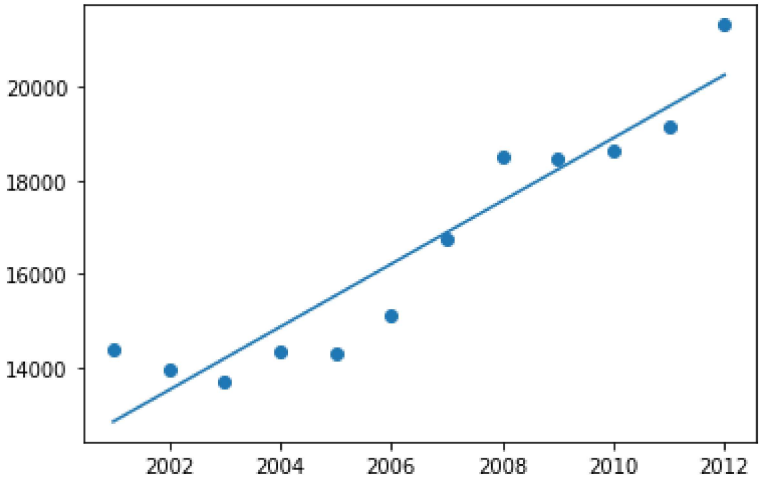

```
Model Score % :
79.55944183348153
Predicted value for 2022 :
[[7012.81351981]]
```

```
GOA
Plotting graph :
```



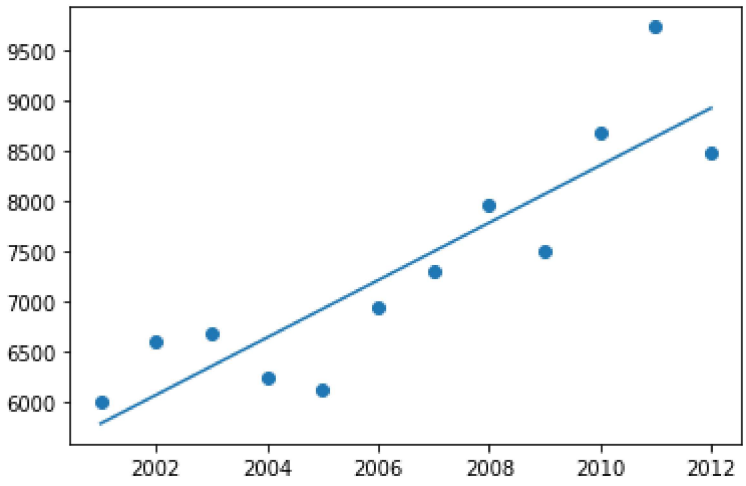

```
Model Score % :
1.4545959025788593
Predicted value for 2022 :
[[897.93706294]]
```

```
GUJARAT
Plotting graph :
```



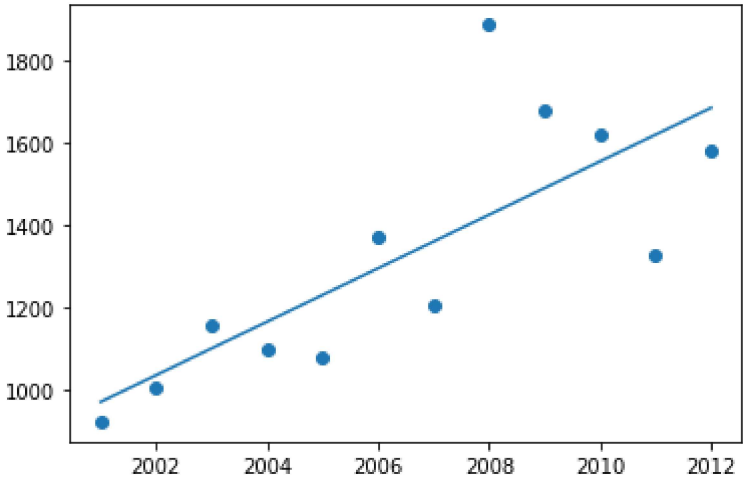

```
Model Score % :
88.81388407022936
Predicted value for 2022 :
[[26997.56293706]]
```

```
HARYANA
Plotting graph :
```



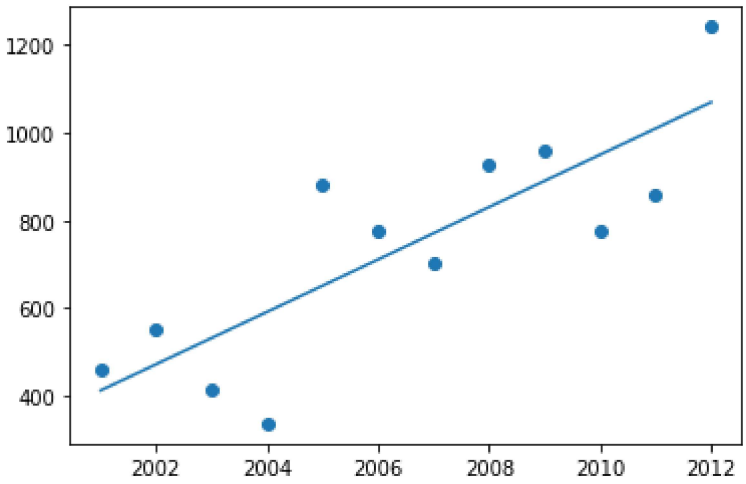

```
Model Score % :
78.49797023108526
Predicted value for 2022 :
[[11773.72377622]]
```

```
HIMACHAL PRADESH
Plotting graph :
```



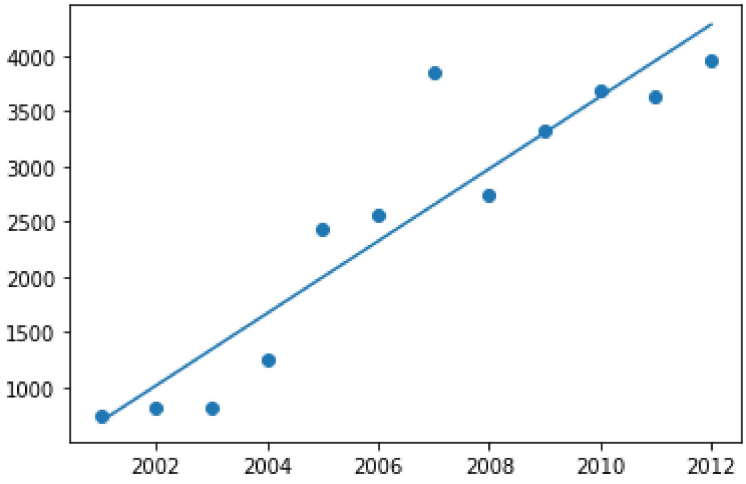

```
Model Score % :
59.36884444457051
Predicted value for 2022 :
[[2335.90909091]]
```

```
JAMMU & KASHMIR
Plotting graph :
```



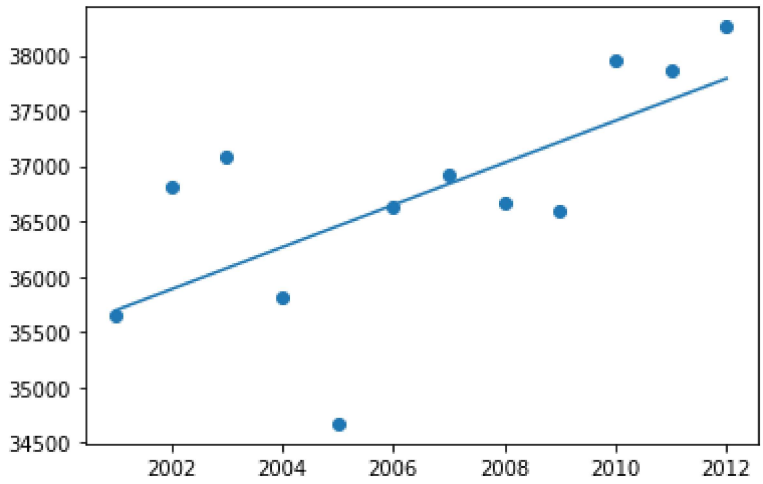

```
Model Score % :
67.43426759913784
Predicted value for 2022 :
[[1667.002331]]
```

```
JHARKHAND
Plotting graph :
```



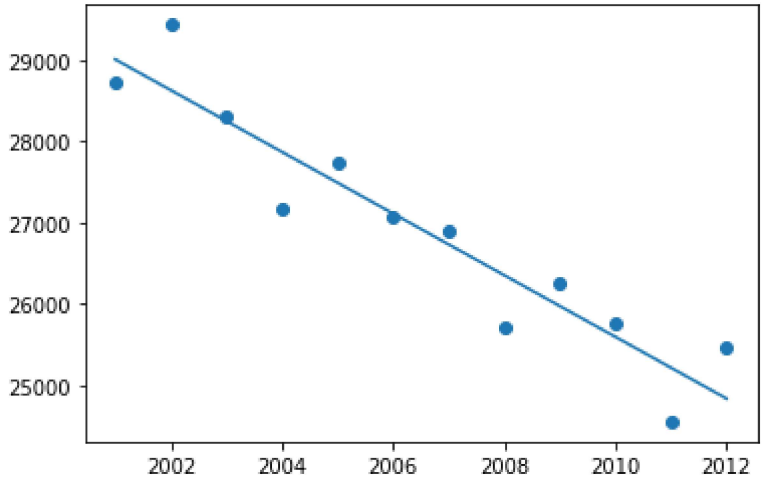

```
Model Score % :
86.15123095702076
Predicted value for 2022 :
[[7546.12937063]]
```

```
KARNATAKA
Plotting graph :
```



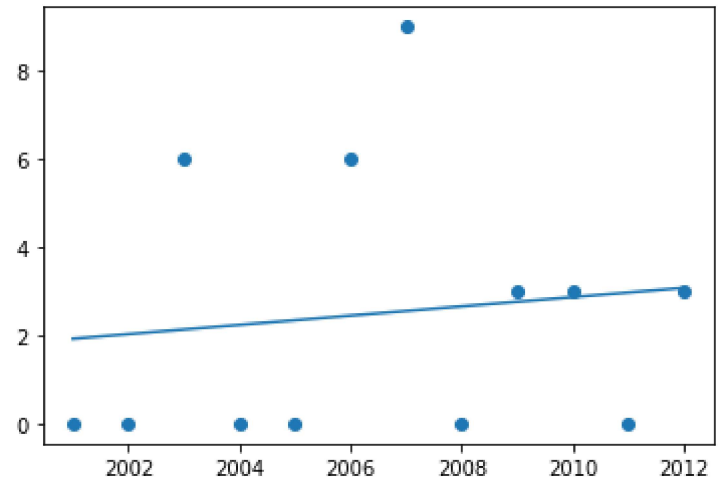

```
Model Score % :
44.819817462771944
Predicted value for 2022 :
[[39694.97552448]]
```

```
KERALA
Plotting graph :
```



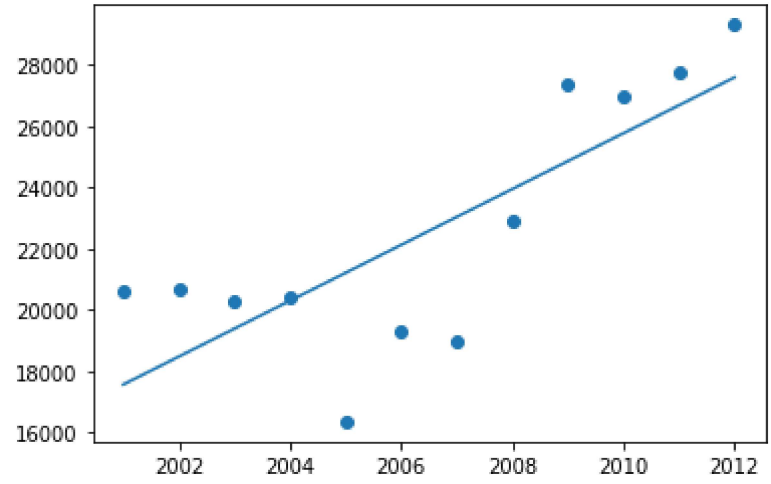

```
Model Score % :
88.41115139478747
Predicted value for 2022 :
[[21060.25641026]]
```

```
LAKSHADWEEP
Plotting graph :
```



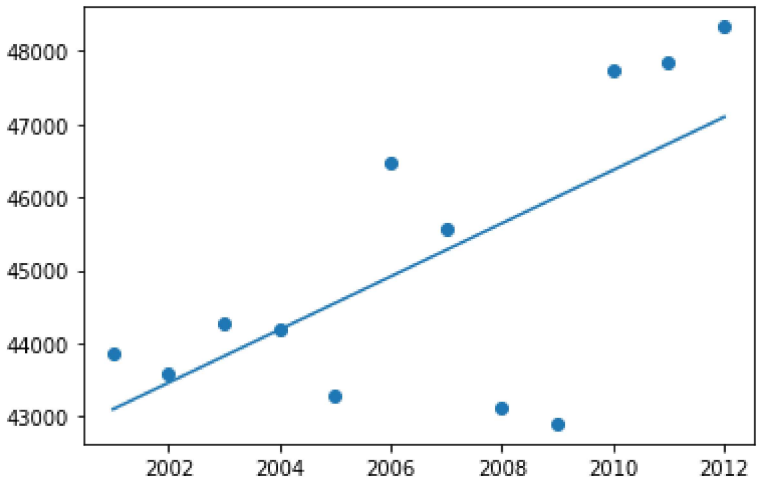

```
Model Score % :
1.4985014985015255
Predicted value for 2022 :
[[4.12587413]]
```

```
MADHYA PRADESH
Plotting graph :
```



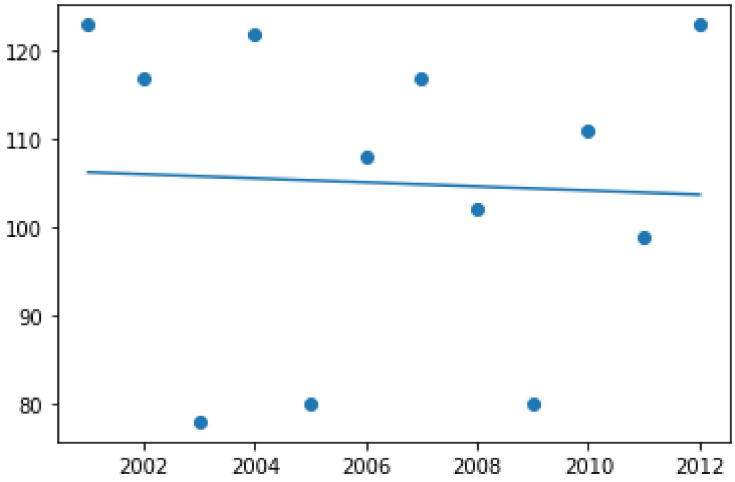

```
Model Score % :
61.09632758243136
Predicted value for 2022 :
[[36724.94405594]]
```

```
MAHARASHTRA
Plotting graph :
```



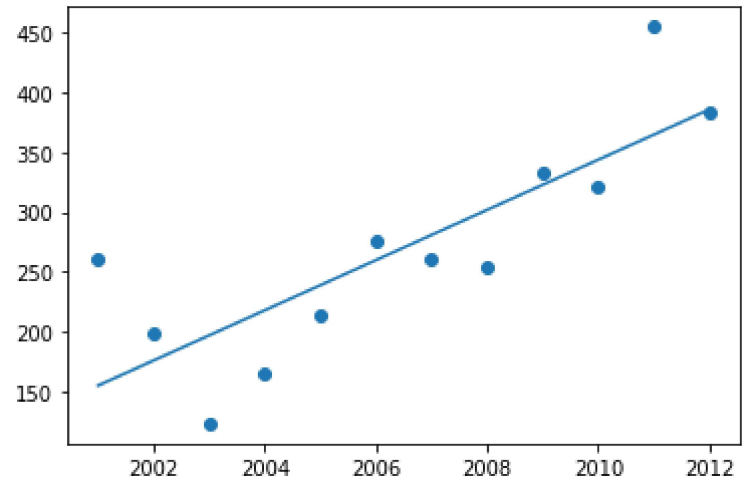

```
Model Score % :
42.488419681846715
Predicted value for 2022 :
[[50735.29370629]]
```

```
MANIPUR
Plotting graph :
```



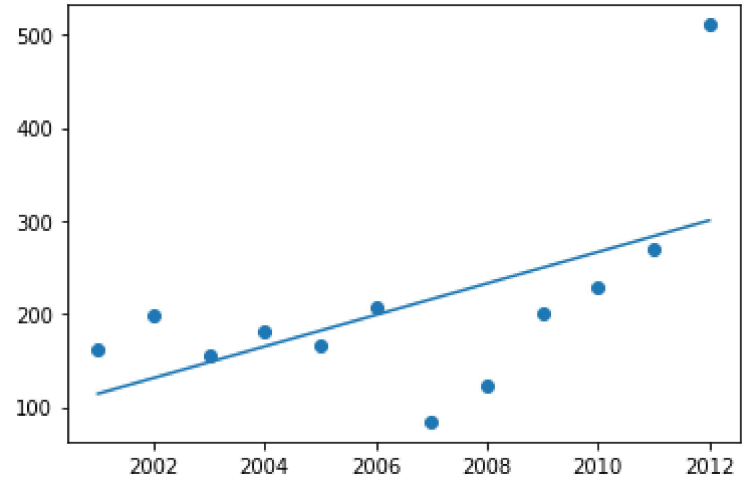

```
Model Score % :
0.23118957545187335
Predicted value for 2022 :
[[101.42307692]]
```

```
MEGHALAYA
Plotting graph :
```



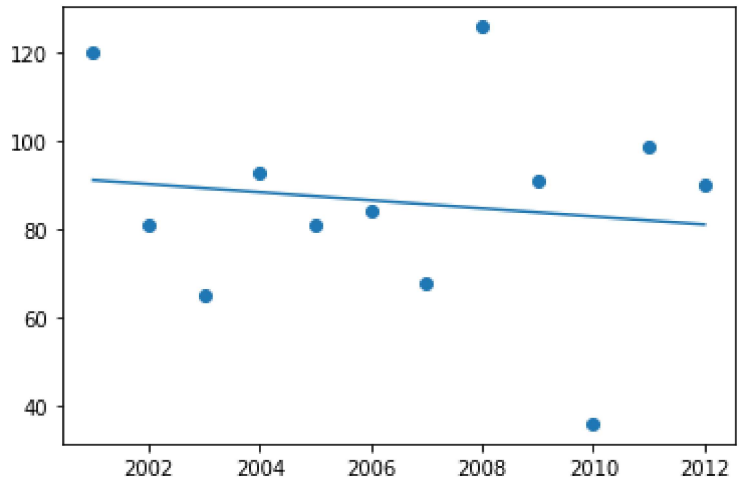

```
Model Score % :
66.06283163611239
Predicted value for 2022 :
[[595.47902098]]
```

```
MIZORAM
Plotting graph :
```



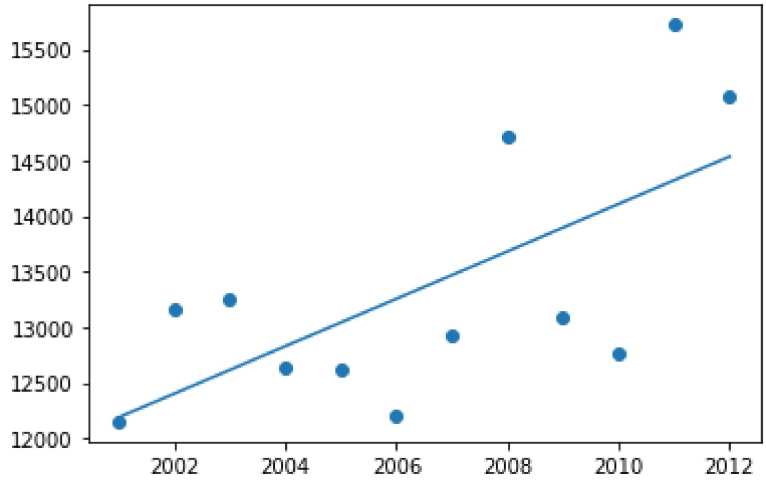

```
Model Score % :
32.519767403582875
Predicted value for 2022 :
[[469.7995338]]
```

```
NAGALAND
Plotting graph :
```



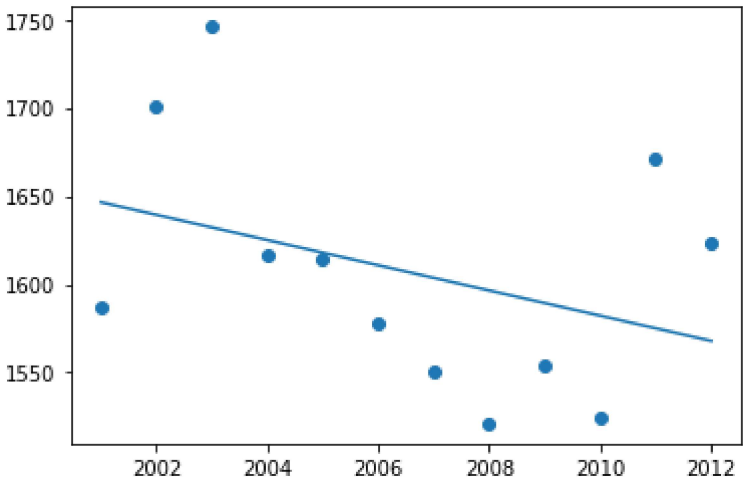

```
Model Score % :
1.89474753445098
Predicted value for 2022 :
[[71.96736597]]
```

```
ODISHA
Plotting graph :
```



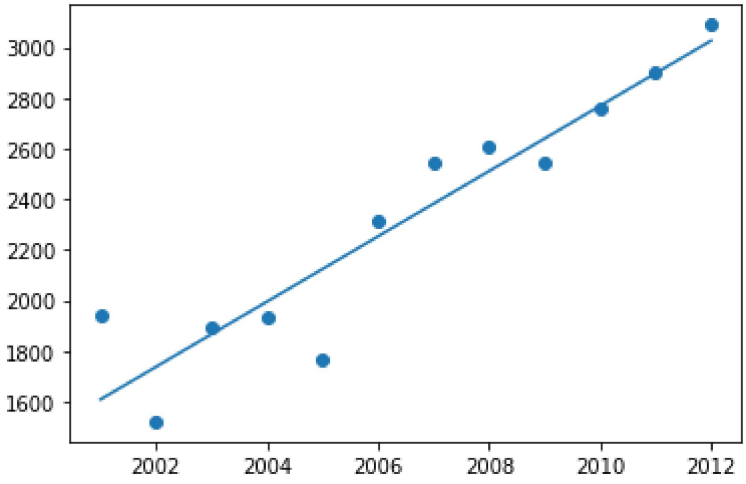

```
Model Score % :
43.700182063293745
Predicted value for 2022 :
[[16670.15384615]]
```

```
PUDUCHERRY
Plotting graph :
```



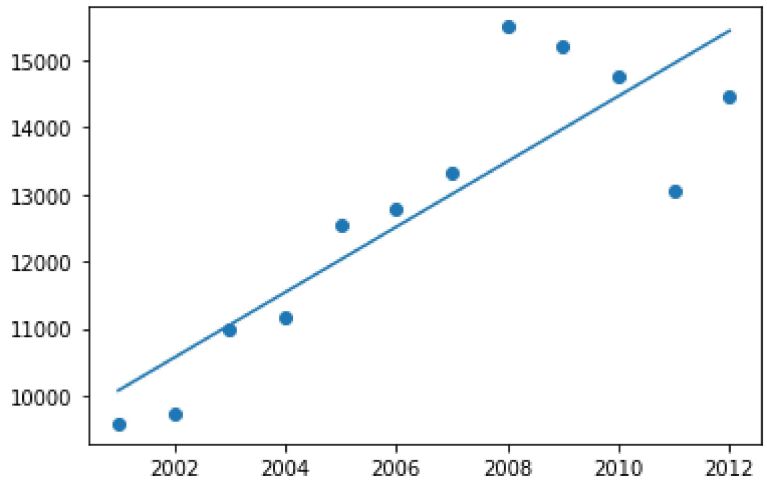

```
Model Score % :
13.498667365814065
Predicted value for 2022 :
[[1496.39044289]]
```

```
PUNJAB
Plotting graph :
```



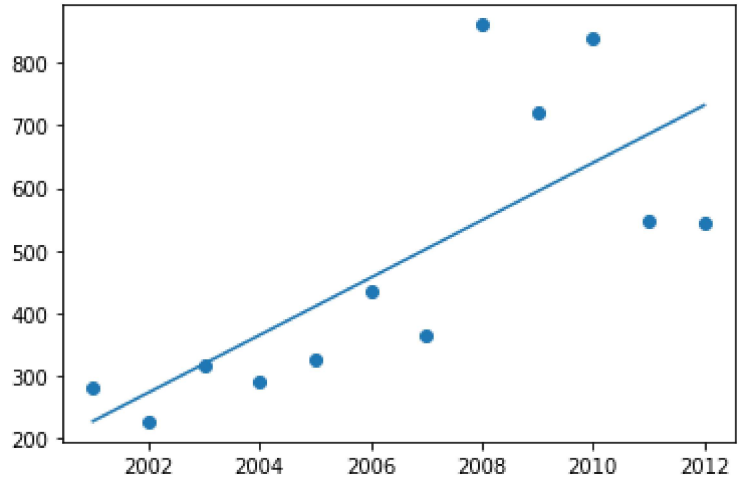

```
Model Score % :
87.28715258972285
Predicted value for 2022 :
[[4314.72377622]]
```

```
RAJASTHAN
Plotting graph :
```



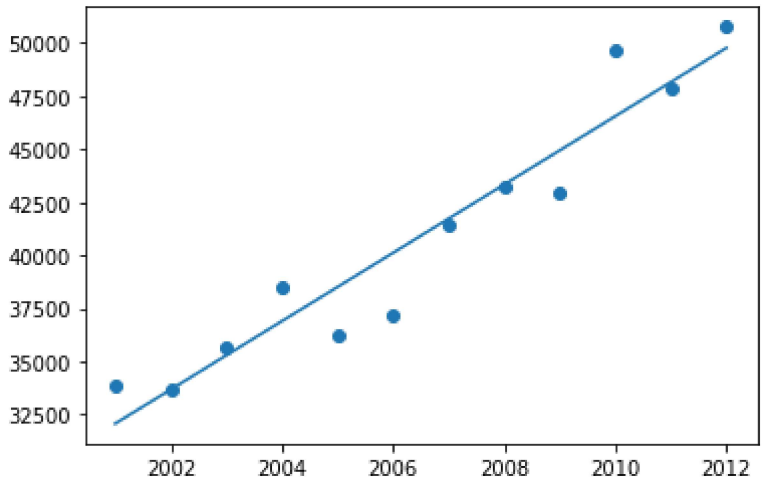

```
Model Score % :
74.3941990699204
Predicted value for 2022 :
[[20306.9009324]]
```

```
SIKKIM
Plotting graph :
```



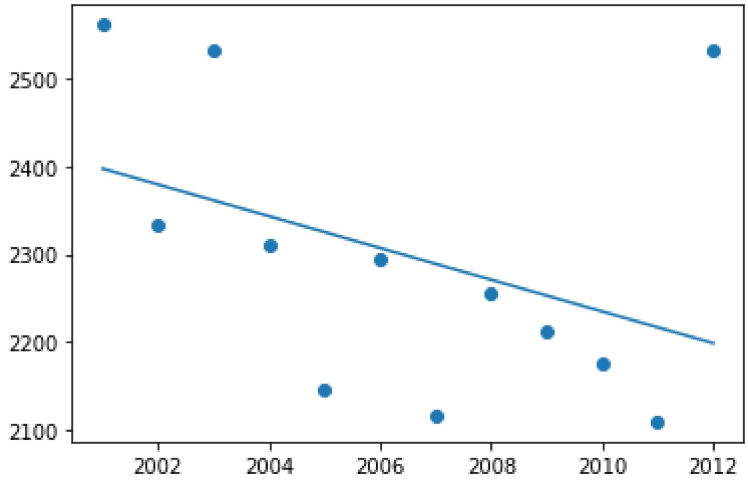

```
Model Score % :
55.08628130600081
Predicted value for 2022 :
[[1190.77389277]]
```

```
TAMIL NADU
Plotting graph :
```



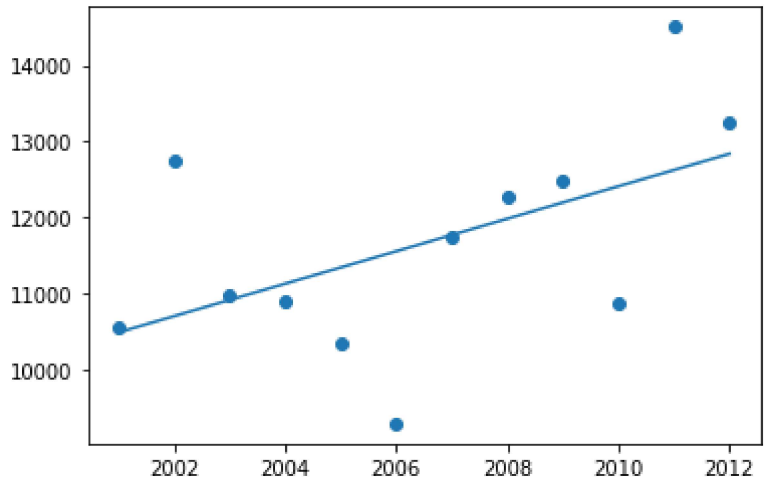

```
Model Score % :
91.41143924306596
Predicted value for 2022 :
[[65876.16200466]]
```

```
TRIPURA
Plotting graph :
```



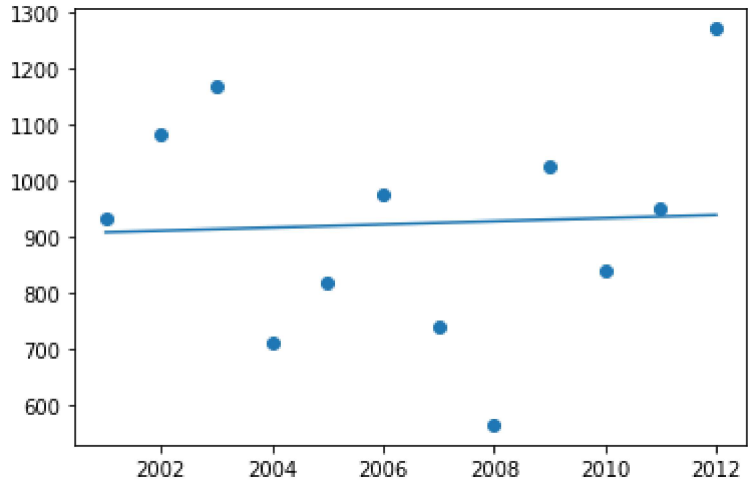

```
Model Score % :
15.75911833673127
Predicted value for 2022 :
[[2017.51165501]]
```

```
UTTAR PRADESH
Plotting graph :
```



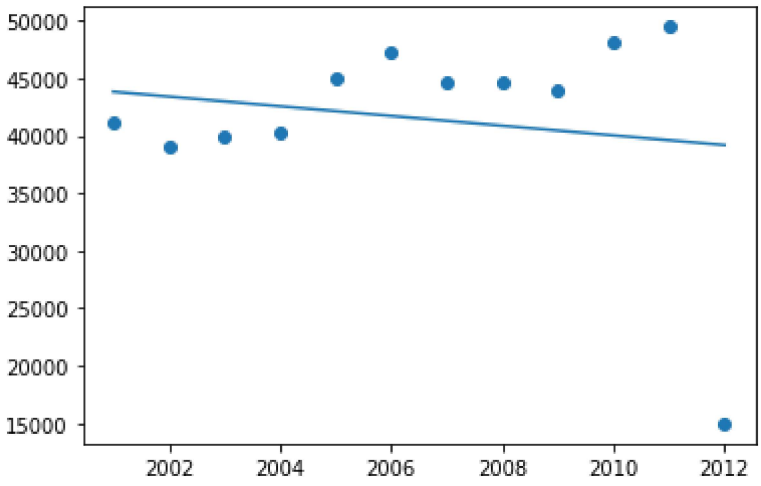

```
Model Score % :
28.186859048328227
Predicted value for 2022 :
[[14971.18531469]]
```

```
UTTARAKHAND
Plotting graph :
```




```
Model Score % :
0.25323717425927983
Predicted value for 2022 :
[[967.90675991]]
```

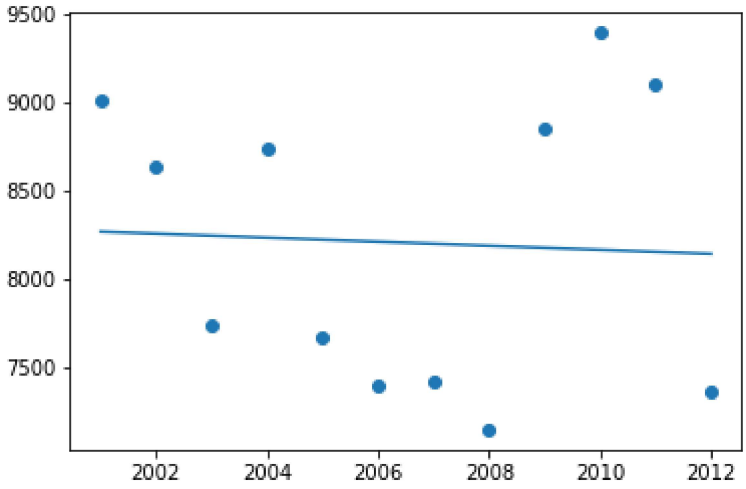
```
WEST BENGAL
Plotting graph :
```




```
Model Score % :
2.8464343515718915
Predicted value for 2022 :
[[34964.72144522]]
```

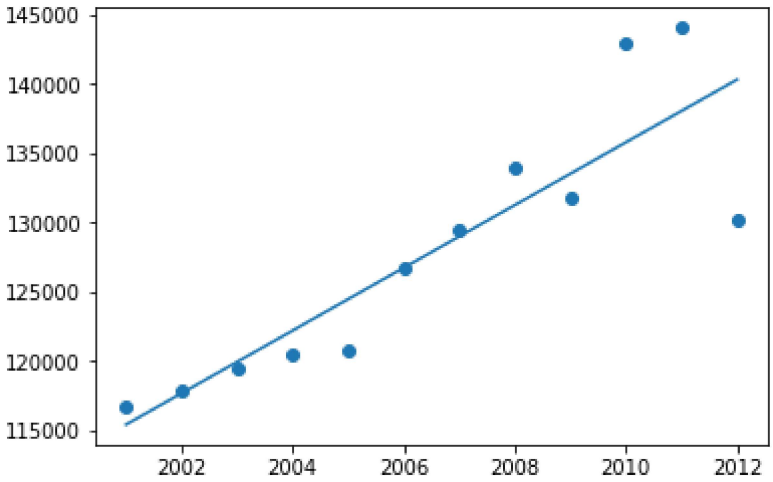
In [115]:
```
Agelist = ['0-14', '15-29','30-44','45-59','60+']
for i in Agelist:
  index = reloaddata2['Age_group']==i
  data2 = reloaddata2[index].groupby('Year').Total.sum()
  #print('Data Set :')
  #print(data2)
  print("\n\n")
  print(i)
  x2 = data2.index.values.reshape(-1,1)
  #print('x coordinated :')
  #print(x2)

  y2 = data2.values.reshape(-1,1)
  #print('y coordinated :')
  #print(y2)

  from sklearn.linear_model import LinearRegression
  reg_gender = LinearRegression()
  reg_gender.fit(x2,y2)
  #print('Coefficent Value :')
  #print(reg_gender.coef_)
  #print('Intercept Value :')
  #print(reg_gender.intercept_)

  result_gender = reg_gender.predict(x2)

  #print('Predicted Value :')
  #print(result_gender)
  plt.scatter(x2,y2)
  print('Plotting graph :')
  plt.plot(x2,result_gender)
  plt.show()
  print('Model Score % :')
  print(reg_gender.score(x2,y2) * 100)
  print('Predicted value for 2022 :')
  print(reg_gender.predict([[2022]]))
  name = i + "_Agegroup_model.pkl"
  import joblib
  joblib.dump(reg_gender, name, compress=9)
```

```
0-14
Plotting graph :
```



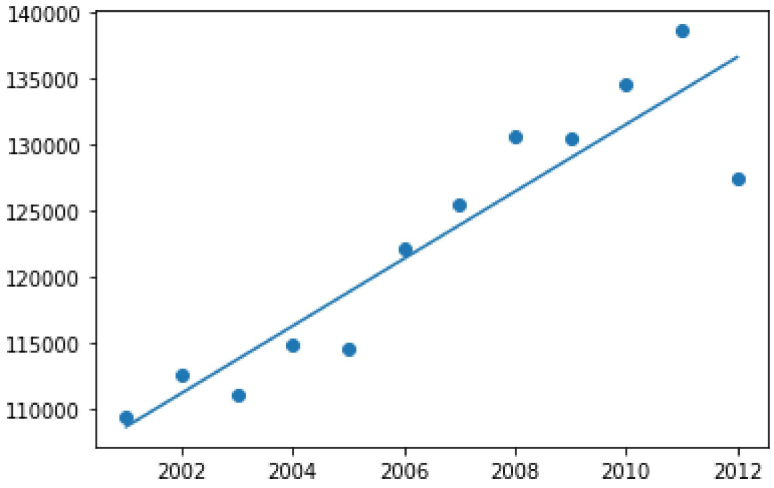

```
Model Score % :
0.25212925855540025
Predicted value for 2022 :
[[8024.04662005]]
```

```
15-29
Plotting graph :
```



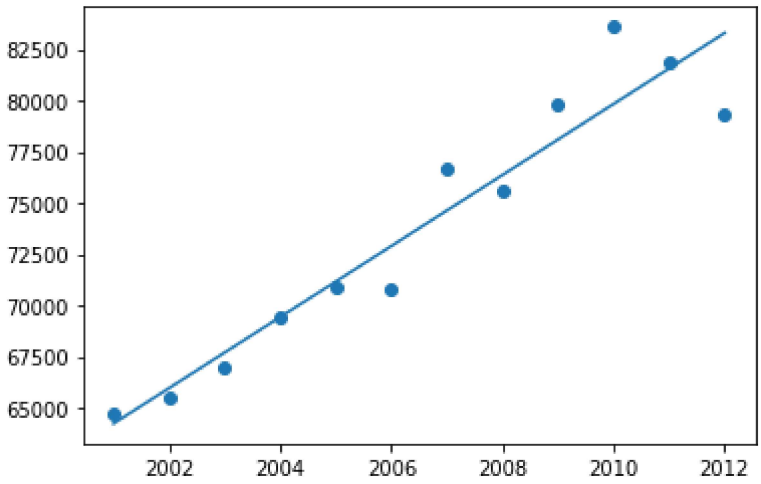

```
Model Score % :
77.08015243774774
Predicted value for 2022 :
[[162933.34848485]]
```

```
30-44
Plotting graph :
```



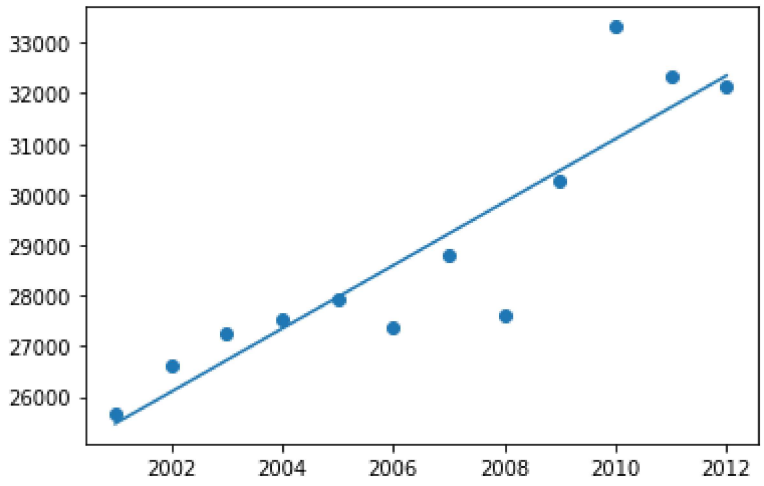

```
Model Score % :
84.6556778485366
Predicted value for 2022 :
[[162066.71328671]]
```

```
45-59
Plotting graph :
```




```
Model Score % :
90.79576741313826
Predicted value for 2022 :
[[100655.8951049]]
```

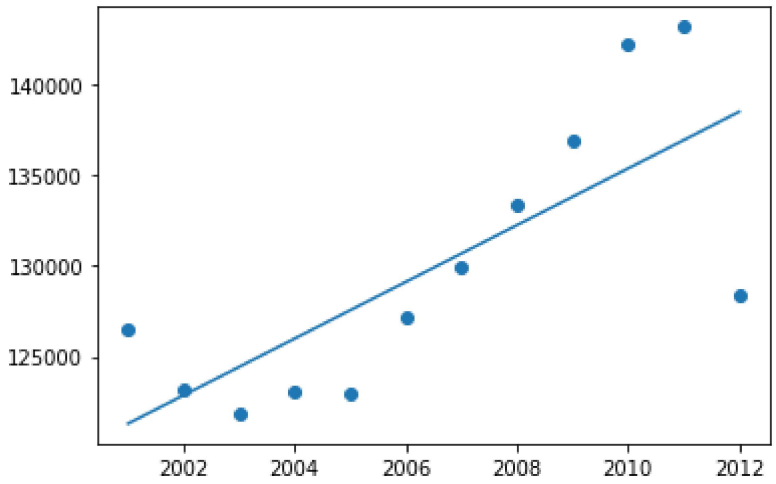
```
60+
Plotting graph :
```




```
Model Score % :
81.51927035714259
Predicted value for 2022 :
[[38599.92191142]]
```

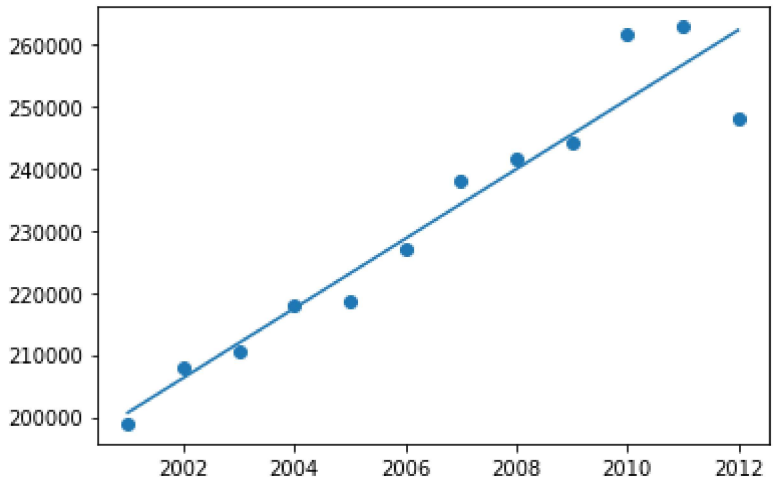
In [117]:
```
Genderlist = ['Female','Male']
for i in Genderlist:
  index = reloaddata2['Gender']==i
  data2 = reloaddata2[index].groupby('Year').Total.sum()
  #print('Data Set :')
  #print(data2)
  print("\n\n")
  print(i)
  x2 = data2.index.values.reshape(-1,1)
  #print('x coordinated :')
  #print(x2)

  y2 = data2.values.reshape(-1,1)
  #print('y coordinated :')
  #print(y2)

  from sklearn.linear_model import LinearRegression
  reg_gender = LinearRegression()
  reg_gender.fit(x2,y2)
  #print('Coefficent Value :')
  #print(reg_gender.coef_)
  #print('Intercept Value :')
  #print(reg_gender.intercept_)

  result_gender = reg_gender.predict(x2)

  #print('Predicted Value :')
  #print(result_gender)
  plt.scatter(x2,y2)
  print('Plotting graph :')
  plt.plot(x2,result_gender)
  plt.show()
  print('Model Score % :')
  print(reg_gender.score(x2,y2) * 100)
  print('Predicted value for 2022 :')
  print(reg_gender.predict([[2022]]))
  name = i + "_Gender_model.pkl"
  import joblib
  joblib.dump(reg_gender, name, compress=9)
```

```
Female
Plotting graph :
```




```
Model Score % :
56.79785455382249
Predicted value for 2022 :
[[154153.9020979]]
```

```
Male
Plotting graph :
```




```
Model Score % :
91.83616431569187
Predicted value for 2022 :
[[318126.02331002]]
```