

```
In [1]: 1 # importing necessary libraries
        2
        3 import numpy as np
        4 import pandas as pd
        5 import matplotlib.pyplot as plt
        6 import seaborn as sns
```

```
In [2]: 1 # loading the dataset
        2
        3 crop_data=pd.read_csv("crop_production.csv")
        4 crop_data
```

```
Out[2]:
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0
...	...	...	...	...	...	...	...
246086	West Bengal	PURULIA	2014	Summer	Rice	306.0	801.0
246087	West Bengal	PURULIA	2014	Summer	Sesamum	627.0	463.0
246088	West Bengal	PURULIA	2014	Whole Year	Sugarcane	324.0	16250.0
246089	West Bengal	PURULIA	2014	Winter	Rice	279151.0	597899.0
246090	West Bengal	PURULIA	2014	Winter	Sesamum	175.0	88.0

246091 rows × 7 columns

## Exploratory Data Analysis and Data Pre-processing

```
In [3]: 1 crop_data.shape
        2
        3 #rows X columns
```

```
Out[3]: (246091, 7)
```

```
In [4]: 1 # dataset columns
        2 crop_data.columns
```

```
Out[4]: Index(['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
               'Production'],
              dtype='object')
```

```
In [5]: 1 # statistical inference of the dataset
        2
        3 crop_data.describe()
```

```
Out[5]:
```

	Crop_Year	Area	Production
<b>count</b>	246091.000000	2.460910e+05	2.423610e+05
<b>mean</b>	2005.643018	1.200282e+04	5.825034e+05
<b>std</b>	4.952164	5.052340e+04	1.706581e+07
<b>min</b>	1997.000000	4.000000e-02	0.000000e+00
<b>25%</b>	2002.000000	8.000000e+01	8.800000e+01
<b>50%</b>	2006.000000	5.820000e+02	7.290000e+02
<b>75%</b>	2010.000000	4.392000e+03	7.023000e+03
<b>max</b>	2015.000000	8.580100e+06	1.250800e+09

```
In [6]: 1 # viewing unique crops available in the dataset
        2
        3 print(crop_data['Crop'].unique())
```

```
['Arecanut' 'Other Kharif pulses' 'Rice' 'Banana' 'Cashewnut' 'Coconut'
'Dry ginger' 'Sugarcane' 'Sweet potato' 'Tapioca' 'Black pepper'
'Dry chillies' 'other oilseeds' 'Turmeric' 'Maize' 'Moong(Green Gram)'
'Urad' 'Arhar/Tur' 'Groundnut' 'Sunflower' 'Bajra' 'Castor seed'
'Cotton(lint)' 'Horse-gram' 'Jowar' 'Korra' 'Ragi' 'Tobacco' 'Gram'
'Wheat' 'Masoor' 'Sesamum' 'Linseed' 'Safflower' 'Onion'
'other misc. pulses' 'Samai' 'Small millets' 'Coriander' 'Potato'
'Other Rabi pulses' 'Soyabean' 'Beans & Mutter(Vegetable)' 'Bhindi'
'Brinjal' 'Citrus Fruit' 'Cucumber' 'Grapes' 'Mango' 'Orange'
'other fibres' 'Other Fresh Fruits' 'Other Vegetables' 'Papaya'
'Pome Fruit' 'Tomato' 'Rapeseed &Mustard' 'Mesta' 'Cowpea(Lobia)' 'Lemon'
'Pome Granet' 'Sapota' 'Cabbage' 'Peas (vegetable)' 'Niger seed'
'Bottle Gourd' 'Sannhamp' 'Varagu' 'Garlic' 'Ginger' 'Oilseeds total'
'Pulses total' 'Jute' 'Peas & beans (Pulses)' 'Blackgram' 'Paddy'
'Pineapple' 'Barley' 'Khesari' 'Guar seed' 'Moth'
'Other Cereals & Millets' 'Cond-spcs other' 'Turnip' 'Carrot' 'Redish'
'Arcanut (Processed)' 'Atcanut (Raw)' 'Cashewnut Processed'
'Cashewnut Raw' 'Cardamom' 'Rubber' 'Bitter Gourd' 'Drum Stick'
'Jack Fruit' 'Snak Guard' 'Pump Kin' 'Tea' 'Coffee' 'Cauliflower'
'Other Citrus Fruit' 'Water Melon' 'Total foodgrain' 'Kapas' 'Colocosia'
'Lentil' 'Bean' 'Jobster' 'Perilla' 'Rajmash Kholar' 'Ricebean (nagadal)'
'Ash Gourd' 'Beet Root' 'Lab-Lab' 'Ribed Guard' 'Yam' 'Apple' 'Peach'
'Pear' 'Plums' 'Litchi' 'Ber' 'Other Dry Fruit' 'Jute & mesta']
```

In [7]: 1 crop\_data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246091 entries, 0 to 246090
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   State_Name      246091 non-null object
1   District_Name   246091 non-null object
2   Crop_Year       246091 non-null int64
3   Season         246091 non-null object
4   Crop            246091 non-null object
5   Area           246091 non-null float64
6   Production      242361 non-null float64
dtypes: float64(2), int64(1), object(4)
memory usage: 13.1+ MB
```

In [8]: 1 *# Checking missing values of the dataset in each column*  
2 crop\_data.isnull().sum()

```
Out[8]: State_Name      0
District_Name    0
Crop_Year        0
Season           0
Crop             0
Area             0
Production       3730
dtype: int64
```

```
In [9]: 1 # Dropping missing values
        2 crop_data = crop_data.dropna()
        3 crop_data
```

```
Out[9]:
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0
...	...	...	...	...	...	...	...
246086	West Bengal	PURULIA	2014	Summer	Rice	306.0	801.0
246087	West Bengal	PURULIA	2014	Summer	Sesamum	627.0	463.0
246088	West Bengal	PURULIA	2014	Whole Year	Sugarcane	324.0	16250.0
246089	West Bengal	PURULIA	2014	Winter	Rice	279151.0	597899.0
246090	West Bengal	PURULIA	2014	Winter	Sesamum	175.0	88.0

242361 rows × 7 columns

```
In [10]: 1 #checking
        2 crop_data.isnull().values.any()
        3
```

Out[10]: False

```
In [11]: 1 # Displaying State Names present in the dataset
        2 crop_data.State_Name.unique()
```

```
Out[11]: array(['Andaman and Nicobar Islands', 'Andhra Pradesh',
                'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
                'Chhattisgarh', 'Dadra and Nagar Haveli', 'Goa', 'Gujarat',
                'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir ', 'Jharkhand',
                'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
                'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Puducherry',
                'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Telangana ',
                'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West Bengal'],
                dtype=object)
```

```
In [12]: 1 # Adding a new column Yield which indicates Production per unit Area.
2
3 crop_data['Yield'] = (crop_data['Production'] / crop_data['Area'])
4 crop_data.head(10)
```

<ipython-input-12-21ef19bb9e83>:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
crop_data['Yield'] = (crop_data['Production'] / crop_data['Area'])
```

Out[12]:

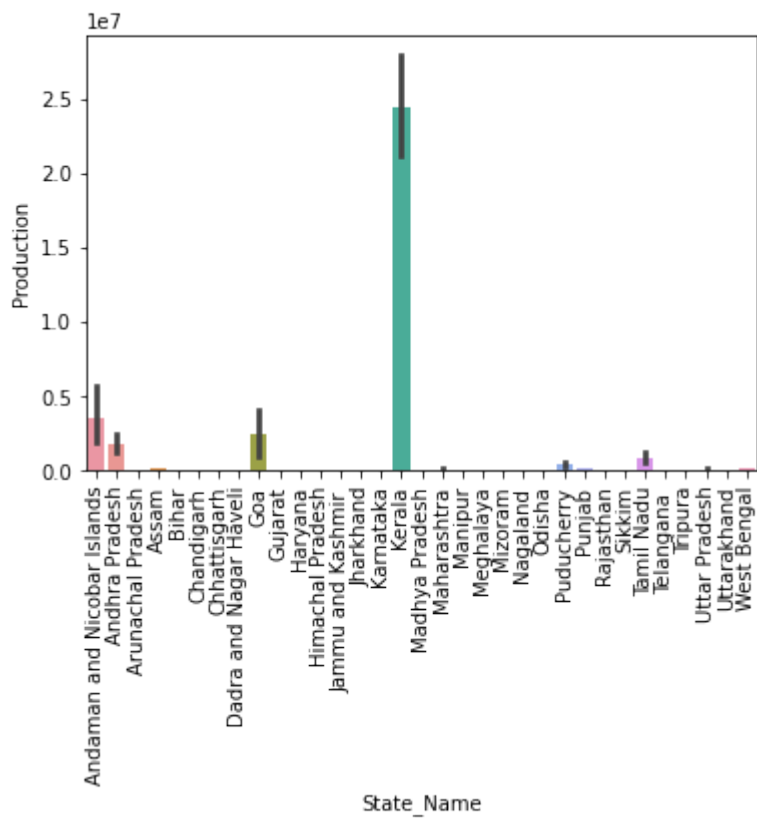
	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0	1.5948
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0	0.5000
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0	3.1470
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0	3.6420
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0	0.2291
5	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Coconut	18168.0	65100000.0	3583.2232
6	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Dry ginger	36.0	100.0	2.7777
7	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1.0	2.0	2.0000
8	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sweet potato	5.0	15.0	3.0000
9	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Tapioca	40.0	169.0	4.2250

Visualizing the dataset :

In [13]:

```
1
2
3 sns.barplot(crop_data["State_Name"], crop_data["Production"])
4 plt.xticks(rotation = 90)
```

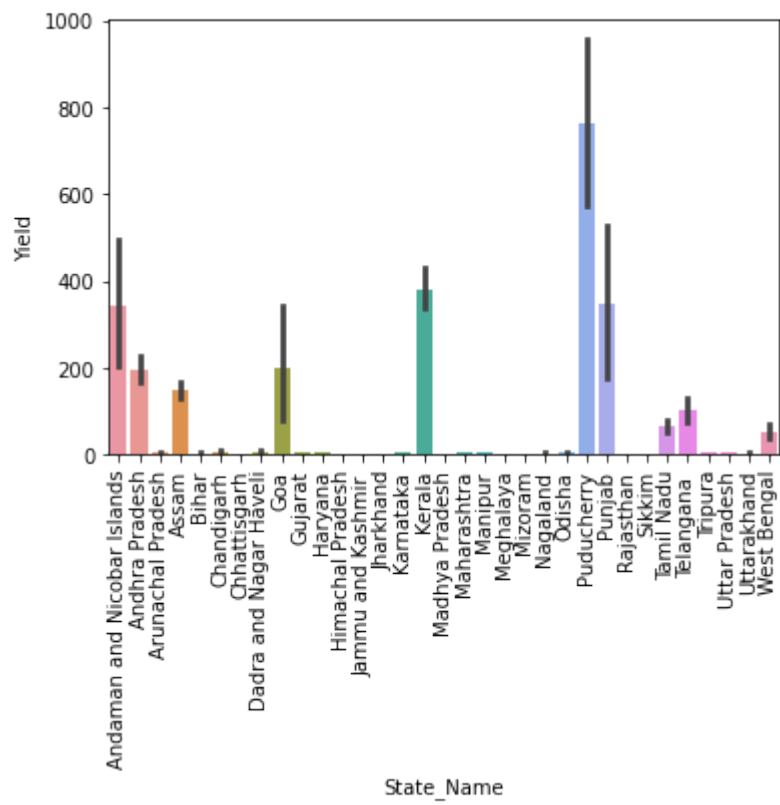
Out[13]: (array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]),  
[Text(0, 0, 'Andaman and Nicobar Islands'),  
Text(1, 0, 'Andhra Pradesh'),  
Text(2, 0, 'Arunachal Pradesh'),  
Text(3, 0, 'Assam'),  
Text(4, 0, 'Bihar'),  
Text(5, 0, 'Chandigarh'),  
Text(6, 0, 'Chhattisgarh'),  
Text(7, 0, 'Dadra and Nagar Haveli'),  
Text(8, 0, 'Goa'),  
Text(9, 0, 'Gujarat'),  
Text(10, 0, 'Haryana'),  
Text(11, 0, 'Himachal Pradesh'),  
Text(12, 0, 'Jammu and Kashmir '),  
Text(13, 0, 'Jharkhand'),  
Text(14, 0, 'Karnataka'),  
Text(15, 0, 'Kerala'),  
Text(16, 0, 'Madhya Pradesh'),  
Text(17, 0, 'Maharashtra'),  
Text(18, 0, 'Manipur'),  
Text(19, 0, 'Meghalaya'),  
Text(20, 0, 'Mizoram'),  
Text(21, 0, 'Nagaland'),  
Text(22, 0, 'Odisha'),  
Text(23, 0, 'Puducherry'),  
Text(24, 0, 'Punjab'),  
Text(25, 0, 'Rajasthan'),  
Text(26, 0, 'Sikkim'),  
Text(27, 0, 'Tamil Nadu'),  
Text(28, 0, 'Telangana '),  
Text(29, 0, 'Tripura'),  
Text(30, 0, 'Uttar Pradesh'),  
Text(31, 0, 'Uttarakhand'),  
Text(32, 0, 'West Bengal')])



```
In [14]: 1 sns.barplot(crop_data["State_Name"], crop_data["Yield"])
        2 plt.xticks(rotation = 90)
```

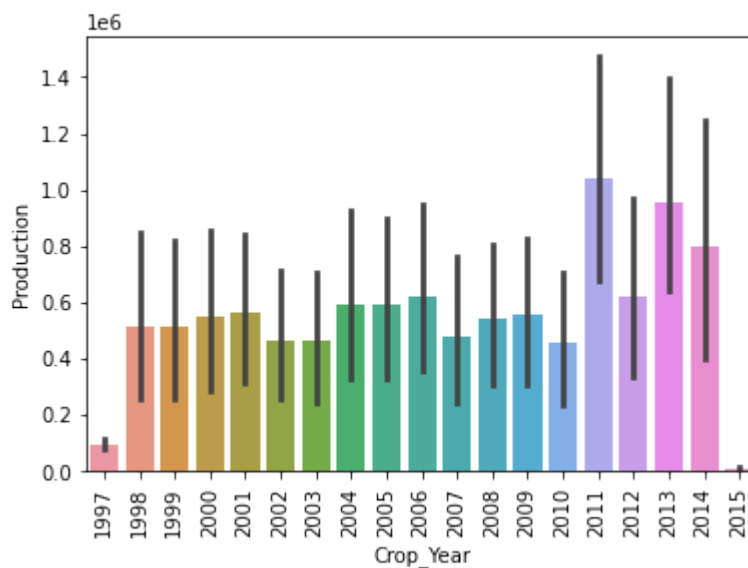
```
Out[14]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
 Text(1, 0, 'Andhra Pradesh'),
 Text(2, 0, 'Arunachal Pradesh'),
 Text(3, 0, 'Assam'),
 Text(4, 0, 'Bihar'),
 Text(5, 0, 'Chandigarh'),
 Text(6, 0, 'Chhattisgarh'),
 Text(7, 0, 'Dadra and Nagar Haveli'),
 Text(8, 0, 'Goa'),
 Text(9, 0, 'Gujarat'),
 Text(10, 0, 'Haryana'),
 Text(11, 0, 'Himachal Pradesh'),
 Text(12, 0, 'Jammu and Kashmir '),
 Text(13, 0, 'Jharkhand'),
 Text(14, 0, 'Karnataka'),
 Text(15, 0, 'Kerala'),
 Text(16, 0, 'Madhya Pradesh'),
 Text(17, 0, 'Maharashtra'),
 Text(18, 0, 'Manipur'),
 Text(19, 0, 'Meghalaya'),
 Text(20, 0, 'Mizoram'),
 Text(21, 0, 'Nagaland'),
 Text(22, 0, 'Odisha'),
 Text(23, 0, 'Puducherry'),
 Text(24, 0, 'Punjab'),
 Text(25, 0, 'Rajasthan'),
 Text(26, 0, 'Sikkim'),
 Text(27, 0, 'Tamil Nadu'),
 Text(28, 0, 'Telangana '),
 Text(29, 0, 'Tripura'),
 Text(30, 0, 'Uttar Pradesh'),
 Text(31, 0, 'Uttarakhand'),
 Text(32, 0, 'West Bengal')])
```





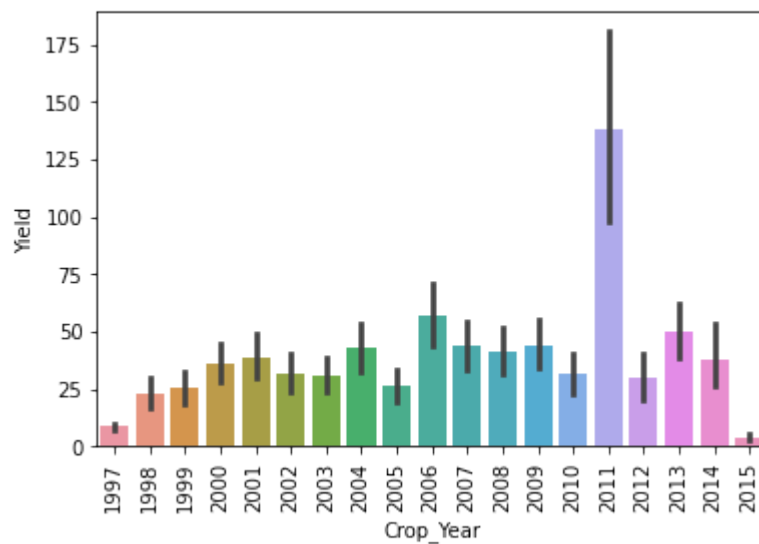
```
In [15]: 1 sns.barplot(crop_data["Crop_Year"], crop_data["Production"])
        2 plt.xticks(rotation = 90)
```

```
Out[15]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18]),
 [Text(0, 0, '1997'),
  Text(1, 0, '1998'),
  Text(2, 0, '1999'),
  Text(3, 0, '2000'),
  Text(4, 0, '2001'),
  Text(5, 0, '2002'),
  Text(6, 0, '2003'),
  Text(7, 0, '2004'),
  Text(8, 0, '2005'),
  Text(9, 0, '2006'),
  Text(10, 0, '2007'),
  Text(11, 0, '2008'),
  Text(12, 0, '2009'),
  Text(13, 0, '2010'),
  Text(14, 0, '2011'),
  Text(15, 0, '2012'),
  Text(16, 0, '2013'),
  Text(17, 0, '2014'),
  Text(18, 0, '2015')])
```



```
In [16]: 1 sns.barplot(crop_data["Crop_Year"], crop_data["Yield"])
        2 plt.xticks(rotation = 90)
```

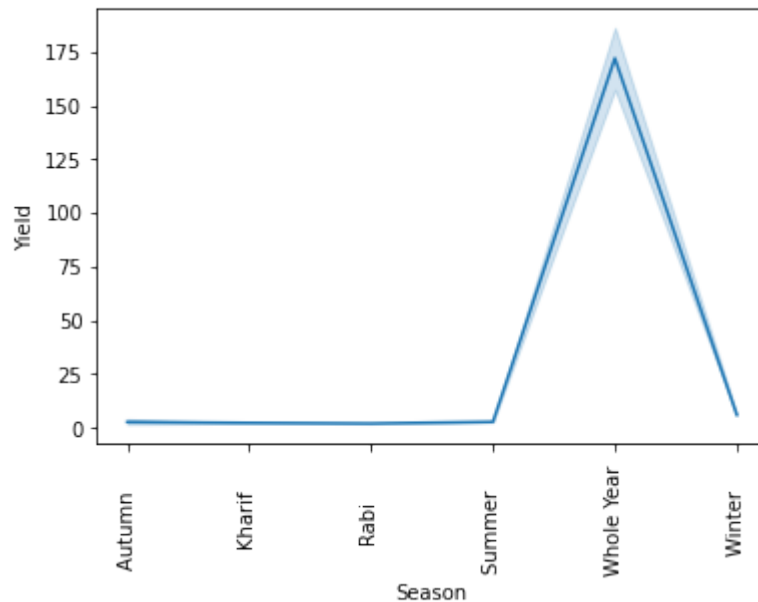
```
Out[16]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18]),
 [Text(0, 0, '1997'),
  Text(1, 0, '1998'),
  Text(2, 0, '1999'),
  Text(3, 0, '2000'),
  Text(4, 0, '2001'),
  Text(5, 0, '2002'),
  Text(6, 0, '2003'),
  Text(7, 0, '2004'),
  Text(8, 0, '2005'),
  Text(9, 0, '2006'),
  Text(10, 0, '2007'),
  Text(11, 0, '2008'),
  Text(12, 0, '2009'),
  Text(13, 0, '2010'),
  Text(14, 0, '2011'),
  Text(15, 0, '2012'),
  Text(16, 0, '2013'),
  Text(17, 0, '2014'),
  Text(18, 0, '2015')])
```



```
In [17]: 1 sns.lineplot(crop_data["Season"], crop_data["Yield"])
          2 plt.xticks(rotation = 90)
```

```
1 sns.lineplot(crop_data["Season"], crop_data["Yield"])
2 plt.xticks(rotation = 90)
```

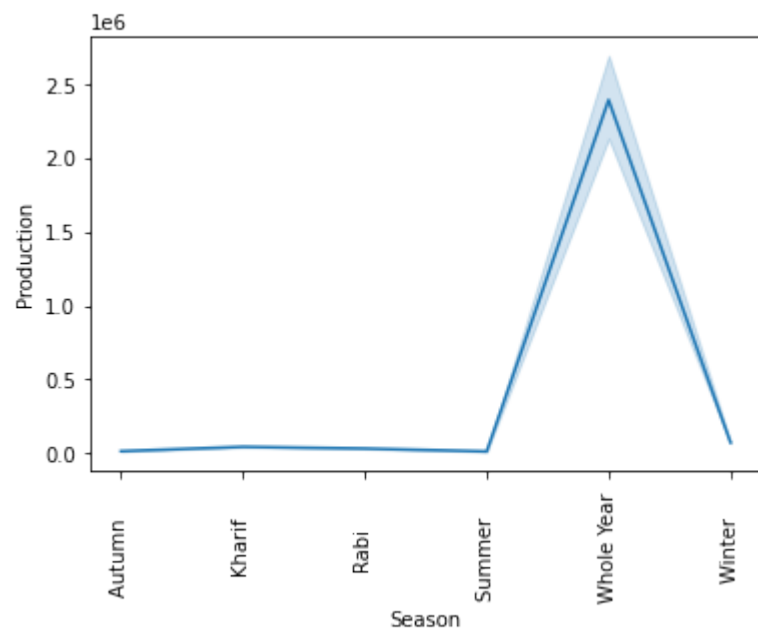
```
Out[17]: ([0, 1, 2, 3, 4, 5],
          [Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, '')])
```



```
In [18]: 1 sns.lineplot(crop_data["Season"], crop_data["Production"])
          2 plt.xticks(rotation = 90)
```

```
1 sns.lineplot(crop_data["Season"], crop_data["Production"])
2 plt.xticks(rotation = 90)
```

```
Out[18]: ([0, 1, 2, 3, 4, 5],
          [Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, ''),
           Text(0, 0, '')])
```

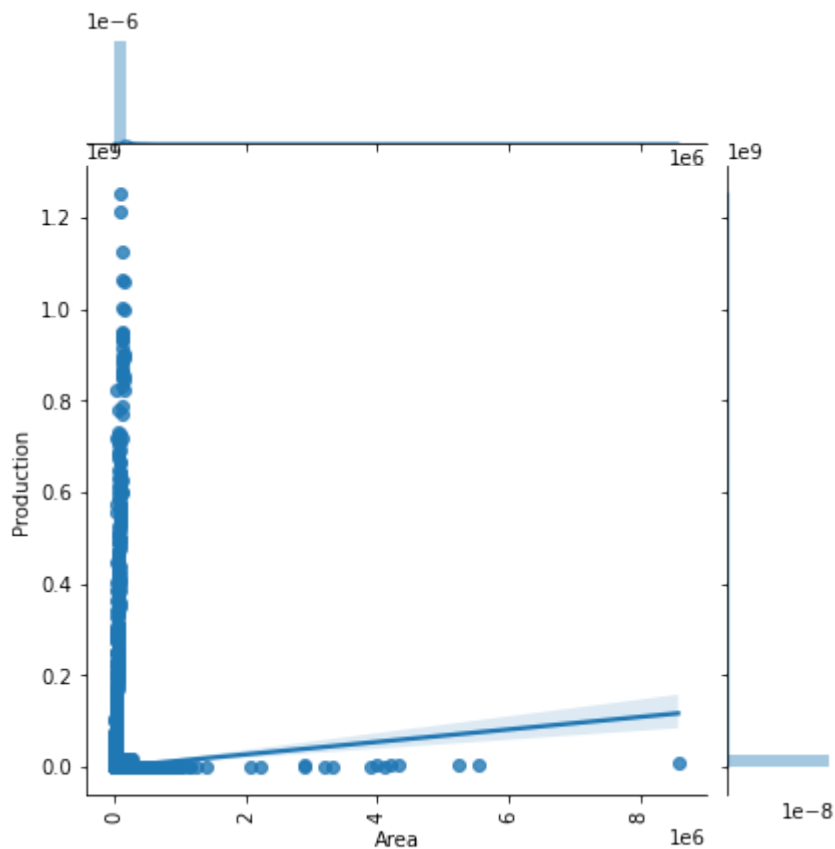


```
In [19]: 1 sns.barplot(crop_data["Crop"], crop_data["Yield"])
        2 plt.xticks(rotation = 60)
```

```
Out[19]: (array([ 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,
                  39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
                  52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
                  65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
                  78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90,
                  91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103,
                  104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
                  117, 118, 119, 120, 121, 122, 123]),
          [Text(0, 0, 'Arecanut'),
           Text(1, 0, 'Other Kharif pulses'),
           Text(2, 0, 'Rice'),
           Text(3, 0, 'Banana'),
           Text(4, 0, 'Cashewnut'),
           Text(5, 0, 'Coconut '),
           Text(6, 0, 'Dry ginger'),
           Text(7, 0, 'Sugarcane'),
           Text(8, 0, 'Sweet potato'),
           Text(9, 0, 'Mango'),
           Text(10, 0, 'Guava'),
           Text(11, 0, 'Jackfruit'),
           Text(12, 0, 'Pineapple'),
           Text(13, 0, 'Watermelon'),
           Text(14, 0, 'Custard apple'),
           Text(15, 0, 'Mango'),
           Text(16, 0, 'Guava'),
           Text(17, 0, 'Jackfruit'),
           Text(18, 0, 'Pineapple'),
           Text(19, 0, 'Watermelon'),
           Text(20, 0, 'Custard apple'),
           Text(21, 0, 'Mango'),
           Text(22, 0, 'Guava'),
           Text(23, 0, 'Jackfruit'),
           Text(24, 0, 'Pineapple'),
           Text(25, 0, 'Watermelon'),
           Text(26, 0, 'Custard apple'),
           Text(27, 0, 'Mango'),
           Text(28, 0, 'Guava'),
           Text(29, 0, 'Jackfruit'),
           Text(30, 0, 'Pineapple'),
           Text(31, 0, 'Watermelon'),
           Text(32, 0, 'Custard apple'),
           Text(33, 0, 'Mango'),
           Text(34, 0, 'Guava'),
           Text(35, 0, 'Jackfruit'),
           Text(36, 0, 'Pineapple'),
           Text(37, 0, 'Watermelon'),
           Text(38, 0, 'Custard apple'),
           Text(39, 0, 'Mango'),
           Text(40, 0, 'Guava'),
           Text(41, 0, 'Jackfruit'),
           Text(42, 0, 'Pineapple'),
           Text(43, 0, 'Watermelon'),
           Text(44, 0, 'Custard apple'),
           Text(45, 0, 'Mango'),
           Text(46, 0, 'Guava'),
           Text(47, 0, 'Jackfruit'),
           Text(48, 0, 'Pineapple'),
           Text(49, 0, 'Watermelon'),
           Text(50, 0, 'Custard apple'),
           Text(51, 0, 'Mango'),
           Text(52, 0, 'Guava'),
           Text(53, 0, 'Jackfruit'),
           Text(54, 0, 'Pineapple'),
           Text(55, 0, 'Watermelon'),
           Text(56, 0, 'Custard apple'),
           Text(57, 0, 'Mango'),
           Text(58, 0, 'Guava'),
           Text(59, 0, 'Jackfruit'),
           Text(60, 0, 'Pineapple'),
           Text(61, 0, 'Watermelon'),
           Text(62, 0, 'Custard apple'),
           Text(63, 0, 'Mango'),
           Text(64, 0, 'Guava'),
           Text(65, 0, 'Jackfruit'),
           Text(66, 0, 'Pineapple'),
           Text(67, 0, 'Watermelon'),
           Text(68, 0, 'Custard apple'),
           Text(69, 0, 'Mango'),
           Text(70, 0, 'Guava'),
           Text(71, 0, 'Jackfruit'),
           Text(72, 0, 'Pineapple'),
           Text(73, 0, 'Watermelon'),
           Text(74, 0, 'Custard apple'),
           Text(75, 0, 'Mango'),
           Text(76, 0, 'Guava'),
           Text(77, 0, 'Jackfruit'),
           Text(78, 0, 'Pineapple'),
           Text(79, 0, 'Watermelon'),
           Text(80, 0, 'Custard apple'),
           Text(81, 0, 'Mango'),
           Text(82, 0, 'Guava'),
           Text(83, 0, 'Jackfruit'),
           Text(84, 0, 'Pineapple'),
           Text(85, 0, 'Watermelon'),
           Text(86, 0, 'Custard apple'),
           Text(87, 0, 'Mango'),
           Text(88, 0, 'Guava'),
           Text(89, 0, 'Jackfruit'),
           Text(90, 0, 'Pineapple'),
           Text(91, 0, 'Watermelon'),
           Text(92, 0, 'Custard apple'),
           Text(93, 0, 'Mango'),
           Text(94, 0, 'Guava'),
           Text(95, 0, 'Jackfruit'),
           Text(96, 0, 'Pineapple'),
           Text(97, 0, 'Watermelon'),
           Text(98, 0, 'Custard apple'),
           Text(99, 0, 'Mango'),
           Text(100, 0, 'Guava'),
           Text(101, 0, 'Jackfruit'),
           Text(102, 0, 'Pineapple'),
           Text(103, 0, 'Watermelon'),
           Text(104, 0, 'Custard apple'),
           Text(105, 0, 'Mango'),
           Text(106, 0, 'Guava'),
           Text(107, 0, 'Jackfruit'),
           Text(108, 0, 'Pineapple'),
           Text(109, 0, 'Watermelon'),
           Text(110, 0, 'Custard apple'),
           Text(111, 0, 'Mango'),
           Text(112, 0, 'Guava'),
           Text(113, 0, 'Jackfruit'),
           Text(114, 0, 'Pineapple'),
           Text(115, 0, 'Watermelon'),
           Text(116, 0, 'Custard apple'),
           Text(117, 0, 'Mango'),
           Text(118, 0, 'Guava'),
           Text(119, 0, 'Jackfruit'),
           Text(120, 0, 'Pineapple'),
           Text(121, 0, 'Watermelon'),
           Text(122, 0, 'Custard apple'),
           Text(123, 0, 'Mango')])
```

```
In [20]: 1 sns.jointplot("Area", "Production", data=crop_data, kind="reg")
        2 plt.xticks(rotation=90)
```

```
Out[20]: (array([-2000000.,          0.,  2000000.,  4000000.,  6000000.,  8000000.,
        10000000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.6000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, '')])
```



Conclusions Obtained from the dataset:

1. Kerela is the largest producer of crop in India.
2. Production per unit area (Yield ) of Puducherry is maximum.
3. In the year 2011, crop yield was maximum.

Analyzing each type of Crop


## 1. RICE

```
In [21]: 1 rice_data = crop_data[crop_data["Crop"]=="Rice"]
         2 rice_data
```

Out[21]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.00	321.00	3.14706
12	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Rice	83.00	300.00	3.61447
18	Andaman and Nicobar Islands	NICOBARS	2002	Kharif	Rice	189.20	510.84	2.70000
27	Andaman and Nicobar Islands	NICOBARS	2003	Kharif	Rice	52.00	90.17	1.73404
36	Andaman and Nicobar Islands	NICOBARS	2004	Kharif	Rice	52.94	72.57	1.37079
...	...	...	...	...	...	...	...	...
246049	West Bengal	PURULIA	2013	Summer	Rice	516.00	1274.00	2.46899
246052	West Bengal	PURULIA	2013	Winter	Rice	302274.00	730136.00	2.41541
246058	West Bengal	PURULIA	2014	Autumn	Rice	264.00	721.00	2.73106
246086	West Bengal	PURULIA	2014	Summer	Rice	306.00	801.00	2.61765
246089	West Bengal	PURULIA	2014	Winter	Rice	279151.00	597899.00	2.14184

15082 rows × 8 columns

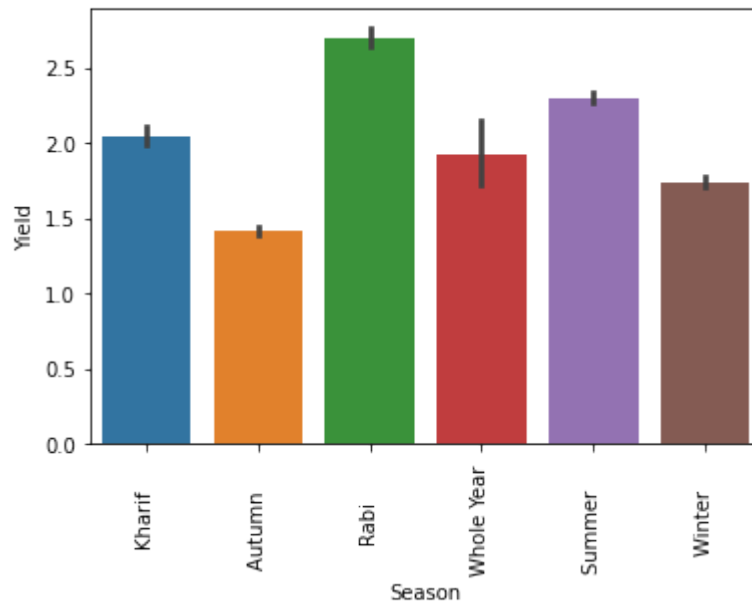


```
In [22]: 1 rice_data.shape
```

Out[22]: (15082, 8)

```
In [23]: 1 sns.barplot("Season", "Yield", data = rice_data)
        2 plt.xticks(rotation = 90)
```

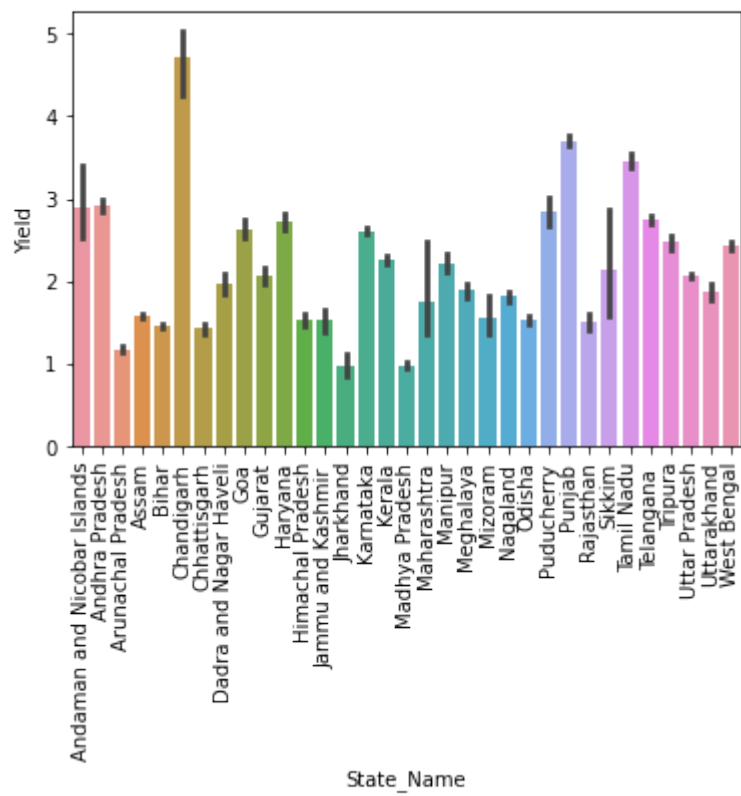
```
Out[23]: (array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Kharif'),
  Text(1, 0, 'Autumn'),
  Text(2, 0, 'Rabi'),
  Text(3, 0, 'Whole Year'),
  Text(4, 0, 'Summer'),
  Text(5, 0, 'Winter')])
```





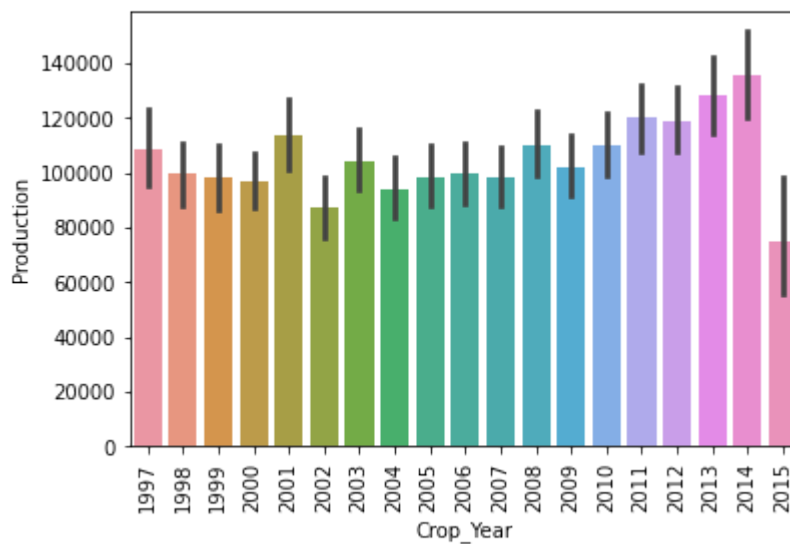
```
In [24]: 1 sns.barplot("State_Name", "Yield", data=rice_data)
        2 plt.xticks(rotation=90)
```

```
Out[24]: (array([ 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]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
 Text(1, 0, 'Andhra Pradesh'),
 Text(2, 0, 'Arunachal Pradesh'),
 Text(3, 0, 'Assam'),
 Text(4, 0, 'Bihar'),
 Text(5, 0, 'Chandigarh'),
 Text(6, 0, 'Chhattisgarh'),
 Text(7, 0, 'Dadra and Nagar Haveli'),
 Text(8, 0, 'Goa'),
 Text(9, 0, 'Gujarat'),
 Text(10, 0, 'Haryana'),
 Text(11, 0, 'Himachal Pradesh'),
 Text(12, 0, 'Jammu and Kashmir '),
 Text(13, 0, 'Jharkhand'),
 Text(14, 0, 'Karnataka'),
 Text(15, 0, 'Kerala'),
 Text(16, 0, 'Madhya Pradesh'),
 Text(17, 0, 'Maharashtra'),
 Text(18, 0, 'Manipur'),
 Text(19, 0, 'Meghalaya'),
 Text(20, 0, 'Mizoram'),
 Text(21, 0, 'Nagaland'),
 Text(22, 0, 'Odisha'),
 Text(23, 0, 'Puducherry'),
 Text(24, 0, 'Punjab'),
 Text(25, 0, 'Rajasthan'),
 Text(26, 0, 'Sikkim'),
 Text(27, 0, 'Tamil Nadu'),
 Text(28, 0, 'Telangana '),
 Text(29, 0, 'Tripura'),
 Text(30, 0, 'Uttar Pradesh'),
 Text(31, 0, 'Uttarakhand'),
 Text(32, 0, 'West Bengal')])
```



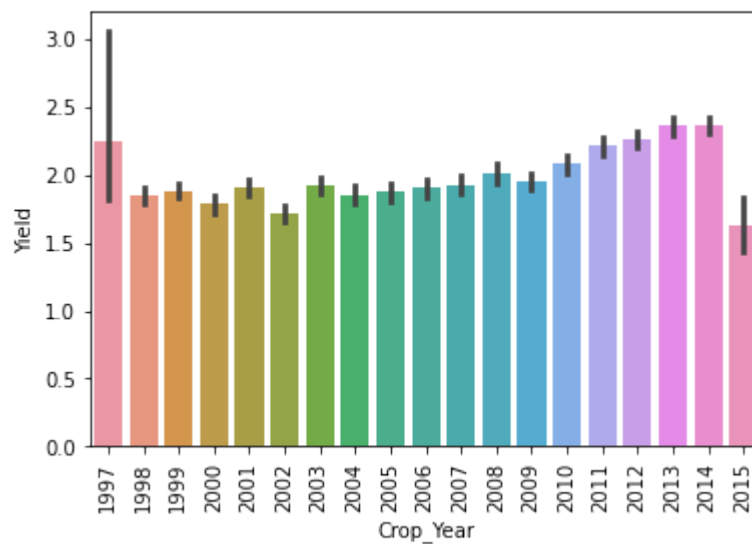
```
In [25]: 1 sns.barplot("Crop_Year", "Production", data=rice_data)
        2 plt.xticks(rotation=90)
```

```
Out[25]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18]),
 [Text(0, 0, '1997'),
 Text(1, 0, '1998'),
 Text(2, 0, '1999'),
 Text(3, 0, '2000'),
 Text(4, 0, '2001'),
 Text(5, 0, '2002'),
 Text(6, 0, '2003'),
 Text(7, 0, '2004'),
 Text(8, 0, '2005'),
 Text(9, 0, '2006'),
 Text(10, 0, '2007'),
 Text(11, 0, '2008'),
 Text(12, 0, '2009'),
 Text(13, 0, '2010'),
 Text(14, 0, '2011'),
 Text(15, 0, '2012'),
 Text(16, 0, '2013'),
 Text(17, 0, '2014'),
 Text(18, 0, '2015')])
```



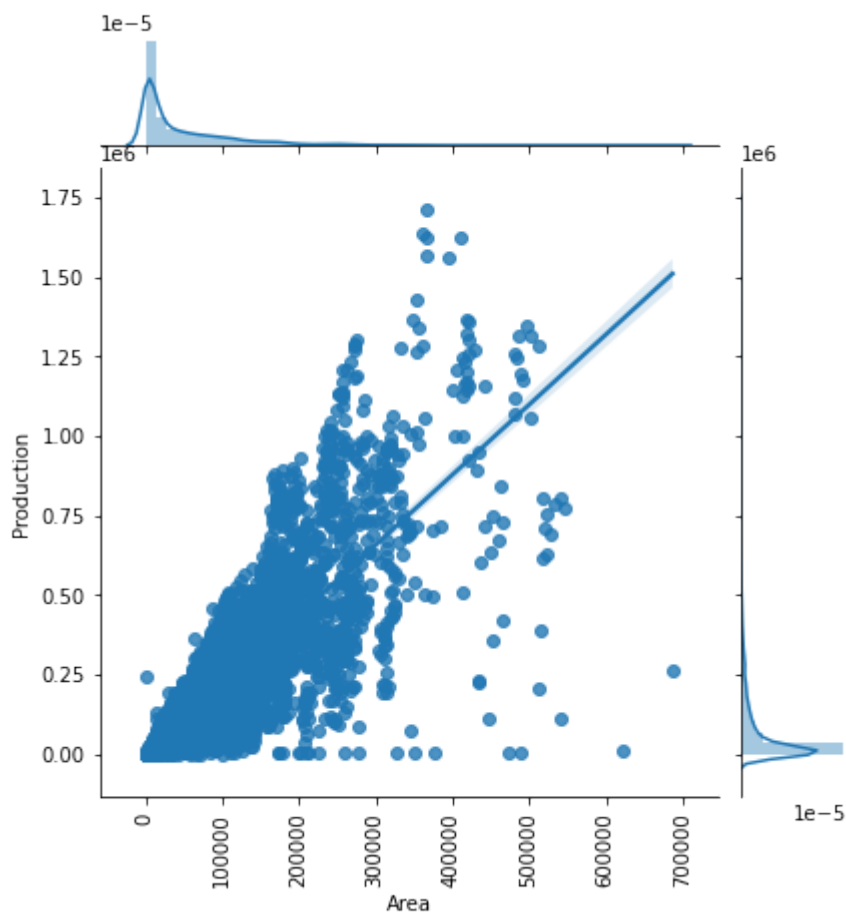
```
In [26]: 1 sns.barplot("Crop_Year", "Yield", data=rice_data)
        2 plt.xticks(rotation=90)
```

```
Out[26]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18]),
 [Text(0, 0, '1997'),
  Text(1, 0, '1998'),
  Text(2, 0, '1999'),
  Text(3, 0, '2000'),
  Text(4, 0, '2001'),
  Text(5, 0, '2002'),
  Text(6, 0, '2003'),
  Text(7, 0, '2004'),
  Text(8, 0, '2005'),
  Text(9, 0, '2006'),
  Text(10, 0, '2007'),
  Text(11, 0, '2008'),
  Text(12, 0, '2009'),
  Text(13, 0, '2010'),
  Text(14, 0, '2011'),
  Text(15, 0, '2012'),
  Text(16, 0, '2013'),
  Text(17, 0, '2014'),
  Text(18, 0, '2015')])
```



```
In [27]: 1 sns.jointplot("Area", "Production", data=rice_data, kind="reg")
2 plt.xticks(rotation=90)
```

```
Out[27]: (array([-100000.,      0., 100000., 200000., 300000., 400000.,
      500000., 600000., 700000., 800000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.6000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[0]])
```



Observations :

1. Rice yield is maximum in Rabi season.
2. Rice yield is maximum in Chandigarh.
3. Rice yield has been growing a little from the year 2009 to 2014.

## WHEAT

```
In [28]: 1 wheat_data = crop_data[crop_data["Crop"]=="Wheat"]
         2 wheat_data
```

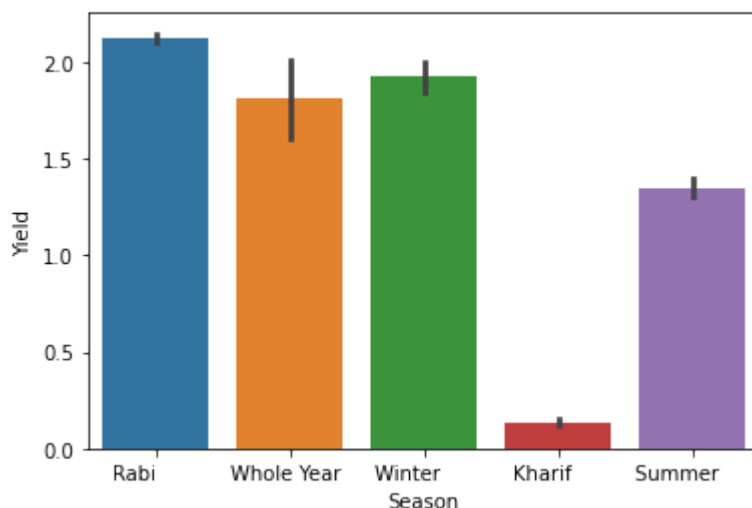
```
Out[28]:
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
230	Andhra Pradesh	ANANTAPUR	1997	Rabi	Wheat	300.0	200.0	0.666667
255	Andhra Pradesh	ANANTAPUR	1998	Rabi	Wheat	400.0	200.0	0.500000
284	Andhra Pradesh	ANANTAPUR	1999	Rabi	Wheat	439.0	294.0	0.669704
326	Andhra Pradesh	ANANTAPUR	2000	Rabi	Wheat	520.0	297.0	0.571154
372	Andhra Pradesh	ANANTAPUR	2001	Rabi	Wheat	307.0	213.0	0.693811
...	...	...	...	...	...	...	...	...
245949	West Bengal	PURULIA	2010	Rabi	Wheat	2013.0	5152.0	2.559364
245980	West Bengal	PURULIA	2011	Rabi	Wheat	1880.0	4206.0	2.237234
246012	West Bengal	PURULIA	2012	Rabi	Wheat	1648.0	3310.0	2.008495
246047	West Bengal	PURULIA	2013	Rabi	Wheat	1187.0	2675.0	2.253580
246084	West Bengal	PURULIA	2014	Rabi	Wheat	1622.0	3663.0	2.258323

7878 rows × 8 columns

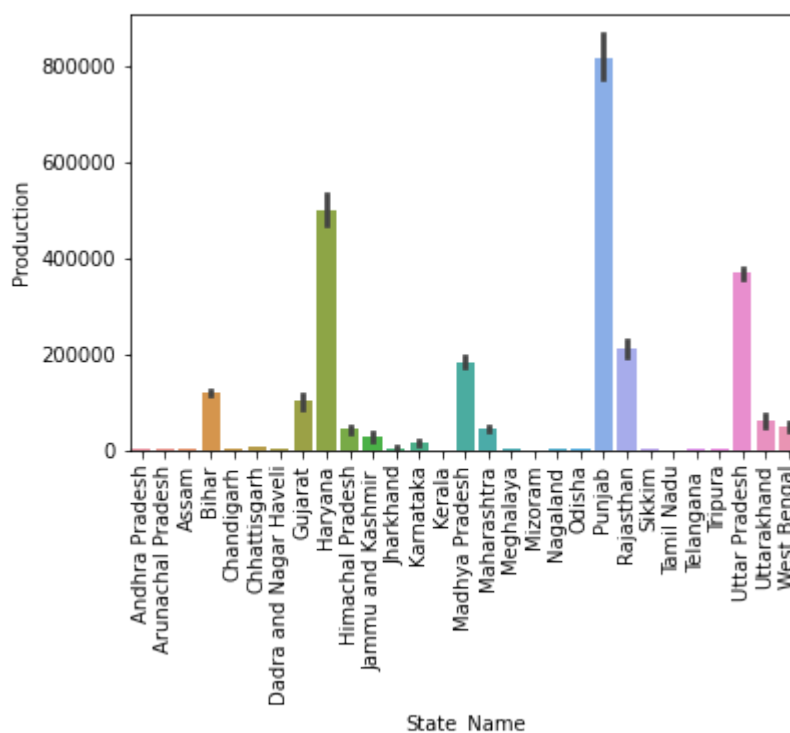
```
In [29]: 1 sns.barplot("Season", "Yield", data = wheat_data)
         2
```

```
Out[29]: <AxesSubplot:xlabel='Season', ylabel='Yield'>
```



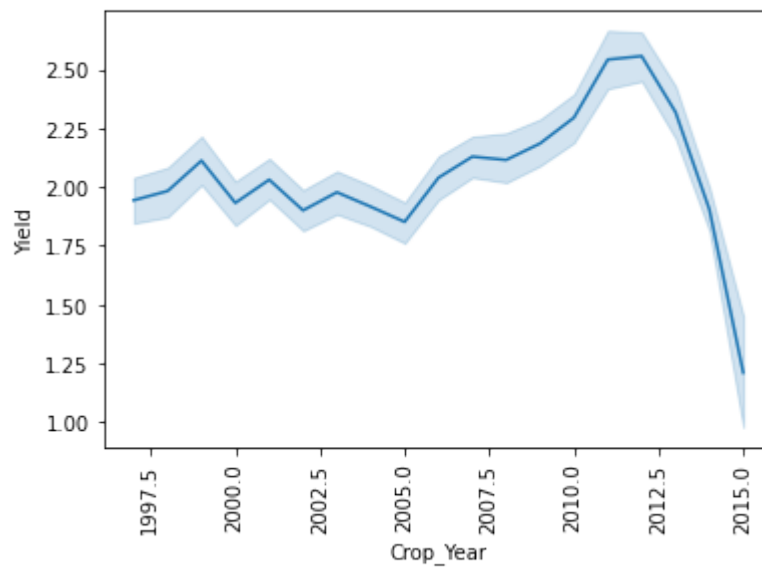
```
In [30]: 1 sns.barplot("State_Name", "Production", data = wheat_data)
2 plt.xticks(rotation = 90)
```

```
Out[30]: (array([ 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]),
 [Text(0, 0, 'Andhra Pradesh'),
  Text(1, 0, 'Arunachal Pradesh'),
  Text(2, 0, 'Assam'),
  Text(3, 0, 'Bihar'),
  Text(4, 0, 'Chandigarh'),
  Text(5, 0, 'Chhattisgarh'),
  Text(6, 0, 'Dadra and Nagar Haveli'),
  Text(7, 0, 'Gujarat'),
  Text(8, 0, 'Haryana'),
  Text(9, 0, 'Himachal Pradesh'),
  Text(10, 0, 'Jammu and Kashmir '),
  Text(11, 0, 'Jharkhand'),
  Text(12, 0, 'Karnataka'),
  Text(13, 0, 'Kerala'),
  Text(14, 0, 'Madhya Pradesh'),
  Text(15, 0, 'Maharashtra'),
  Text(16, 0, 'Meghalaya'),
  Text(17, 0, 'Mizoram'),
  Text(18, 0, 'Nagaland'),
  Text(19, 0, 'Odisha'),
  Text(20, 0, 'Punjab'),
  Text(21, 0, 'Rajasthan'),
  Text(22, 0, 'Sikkim'),
  Text(23, 0, 'Tamil Nadu'),
  Text(24, 0, 'Telangana '),
  Text(25, 0, 'Tripura'),
  Text(26, 0, 'Uttar Pradesh'),
  Text(27, 0, 'Uttarakhand'),
  Text(28, 0, 'West Bengal')])
```



```
In [31]: 1 sns.lineplot("Crop_Year", "Yield", data=wheat_data)
        2 plt.xticks(rotation=90)
```

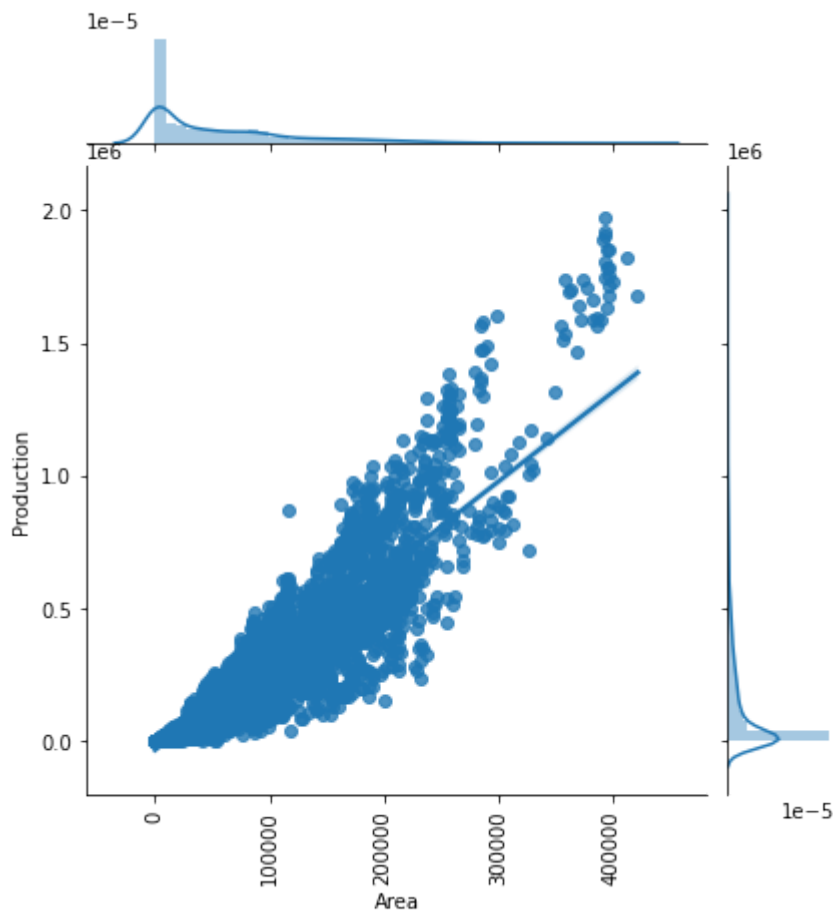
```
Out[31]: (array([1995. , 1997.5, 2000. , 2002.5, 2005. , 2007.5, 2010. , 2012.5,
                2015. , 2017.5]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[0]])
```





```
In [32]: 1 sns.jointplot("Area", "Production", data=wheat_data, kind="reg")
        2 plt.xticks(rotation=90)
```

```
Out[32]: (array([-100000.,      0.,  100000.,  200000.,  300000.,  400000.,
                  500000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.6000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, '')])
```



Conclusions obtained :

1. Rice yield is maximum in Rabi season.
2. Rice yield is maximum in Chandigarh.
3. Rice yield has been growing a little from the year 2009 to 2014.

### 3. COCONUT

```
In [33]: 1 coconut_data = crop_data[crop_data["Crop"]=="Urad"]
          2 coconut_data
```

```
Out[33]:
```

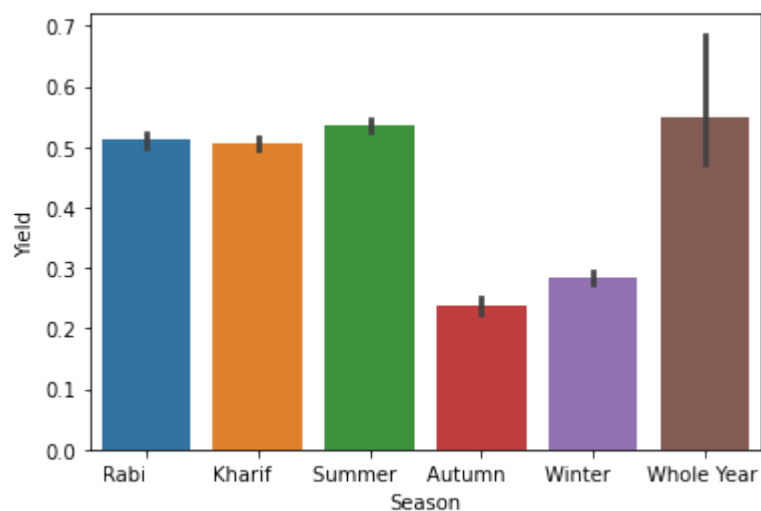
	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
75	Andaman and Nicobar Islands	NICOBARS	2010	Rabi	Urad	1.5	1.16	0.773333
125	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	2010	Rabi	Urad	1059.5	458.79	0.433025
199	Andaman and Nicobar Islands	SOUTH ANDAMANS	2010	Rabi	Urad	34.0	15.05	0.442647
270	Andhra Pradesh	ANANTAPUR	1999	Kharif	Urad	4.0	2.00	0.500000
309	Andhra Pradesh	ANANTAPUR	2000	Kharif	Urad	135.0	8.00	0.059259
...	...	...	...	...	...	...	...	...
246000	West Bengal	PURULIA	2012	Kharif	Urad	9258.0	2910.00	0.314323
246011	West Bengal	PURULIA	2012	Rabi	Urad	259.0	133.00	0.513514
246033	West Bengal	PURULIA	2013	Kharif	Urad	12986.0	2877.00	0.221546
246070	West Bengal	PURULIA	2014	Kharif	Urad	11493.0	3287.00	0.286000
246083	West Bengal	PURULIA	2014	Rabi	Urad	220.0	113.00	0.513636

9710 rows × 8 columns



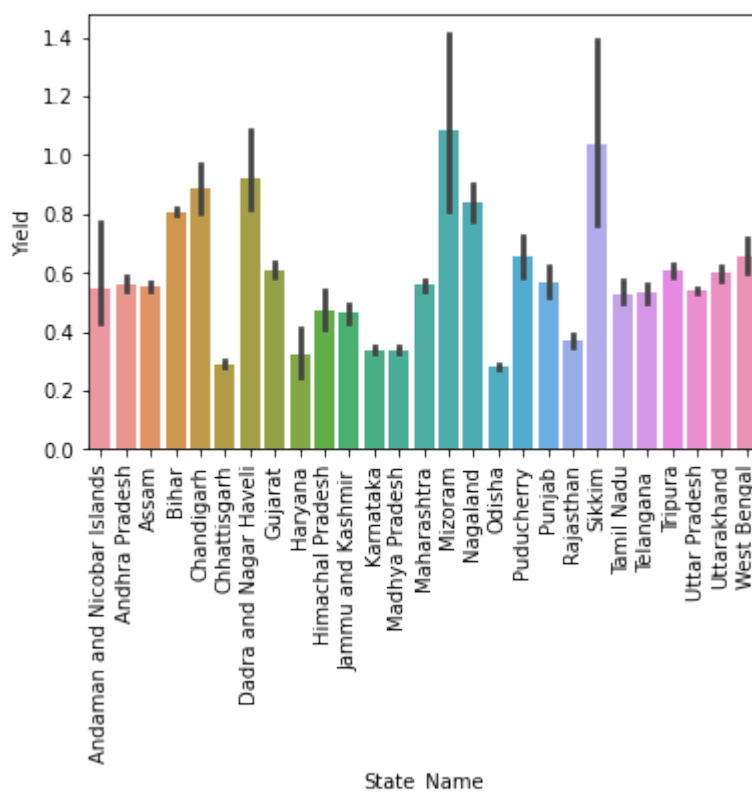
```
In [34]: 1 sns.barplot("Season", "Yield", data = coconut_data)
```

```
Out[34]: <AxesSubplot:xlabel='Season', ylabel='Yield'>
```



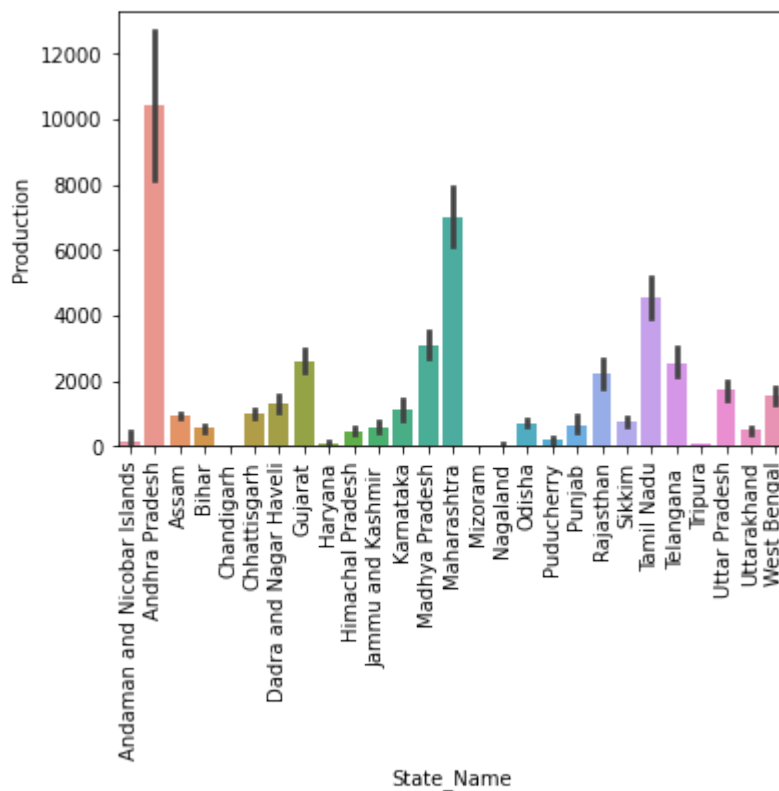
```
In [35]: 1 sns.barplot("State_Name", "Yield", data = coconut_data)
2 plt.xticks(rotation = 90)
```

```
Out[35]: (array([ 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]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
  Text(1, 0, 'Andhra Pradesh'),
  Text(2, 0, 'Assam'),
  Text(3, 0, 'Bihar'),
  Text(4, 0, 'Chandigarh'),
  Text(5, 0, 'Chhattisgarh'),
  Text(6, 0, 'Dadra and Nagar Haveli'),
  Text(7, 0, 'Gujarat'),
  Text(8, 0, 'Haryana'),
  Text(9, 0, 'Himachal Pradesh'),
  Text(10, 0, 'Jammu and Kashmir '),
  Text(11, 0, 'Karnataka'),
  Text(12, 0, 'Madhya Pradesh'),
  Text(13, 0, 'Maharashtra'),
  Text(14, 0, 'Mizoram'),
  Text(15, 0, 'Nagaland'),
  Text(16, 0, 'Odisha'),
  Text(17, 0, 'Puducherry'),
  Text(18, 0, 'Punjab'),
  Text(19, 0, 'Rajasthan'),
  Text(20, 0, 'Sikkim'),
  Text(21, 0, 'Tamil Nadu'),
  Text(22, 0, 'Telangana '),
  Text(23, 0, 'Tripura'),
  Text(24, 0, 'Uttar Pradesh'),
  Text(25, 0, 'Uttarakhand'),
  Text(26, 0, 'West Bengal')])
```



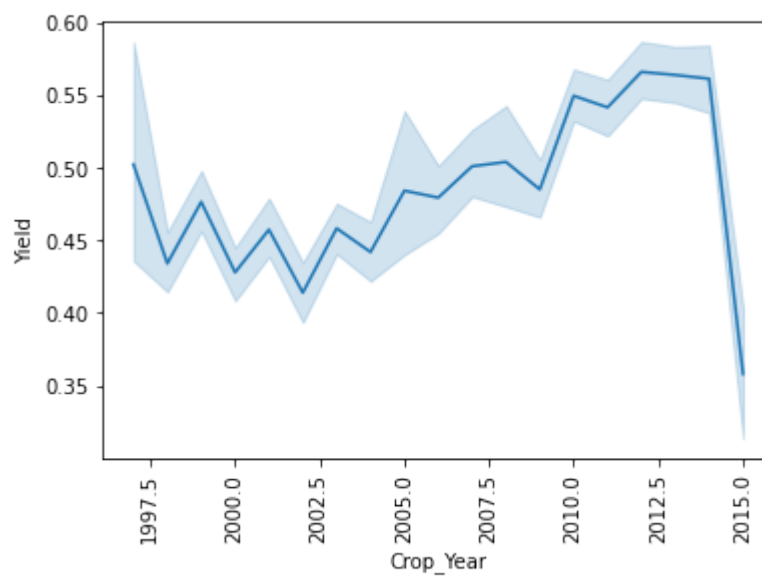
```
In [36]: 1 sns.barplot("State_Name", "Production", data = coconut_data)
2 plt.xticks(rotation = 90)
```

```
Out[36]: (array([ 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]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
  Text(1, 0, 'Andhra Pradesh'),
  Text(2, 0, 'Assam'),
  Text(3, 0, 'Bihar'),
  Text(4, 0, 'Chandigarh'),
  Text(5, 0, 'Chhattisgarh'),
  Text(6, 0, 'Dadra and Nagar Haveli'),
  Text(7, 0, 'Gujarat'),
  Text(8, 0, 'Haryana'),
  Text(9, 0, 'Himachal Pradesh'),
  Text(10, 0, 'Jammu and Kashmir '),
  Text(11, 0, 'Karnataka'),
  Text(12, 0, 'Madhya Pradesh'),
  Text(13, 0, 'Maharashtra'),
  Text(14, 0, 'Mizoram'),
  Text(15, 0, 'Nagaland'),
  Text(16, 0, 'Odisha'),
  Text(17, 0, 'Puducherry'),
  Text(18, 0, 'Punjab'),
  Text(19, 0, 'Rajasthan'),
  Text(20, 0, 'Sikkim'),
  Text(21, 0, 'Tamil Nadu'),
  Text(22, 0, 'Telangana '),
  Text(23, 0, 'Tripura'),
  Text(24, 0, 'Uttar Pradesh'),
  Text(25, 0, 'Uttarakhand'),
  Text(26, 0, 'West Bengal')])
```



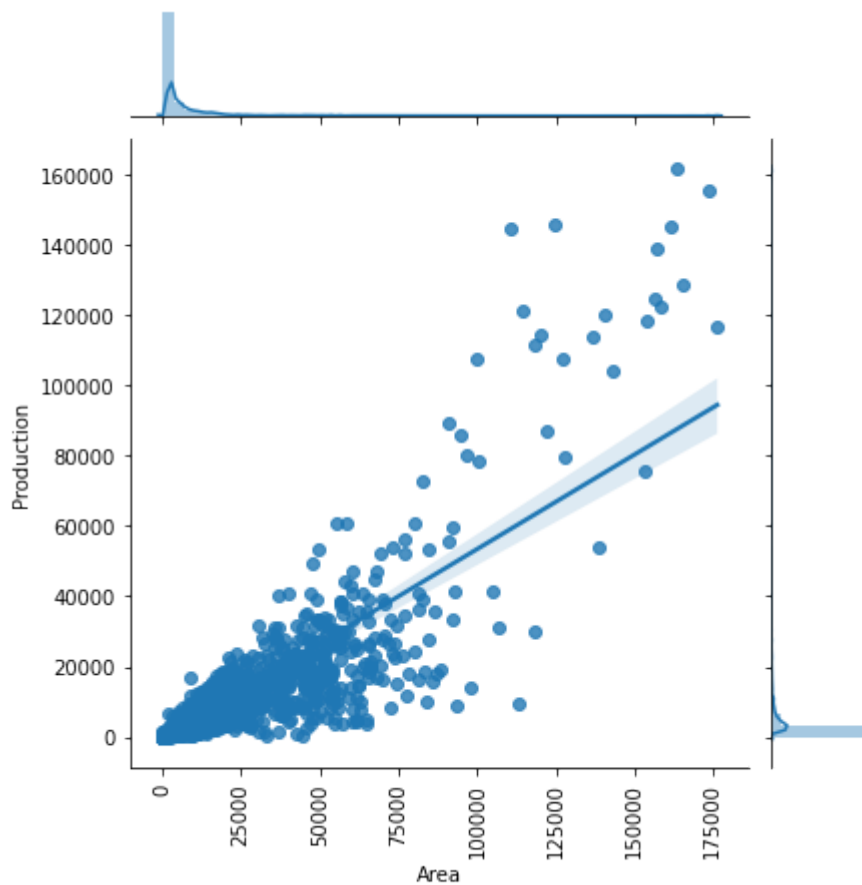
```
In [37]: 1 sns.lineplot("Crop_Year", "Yield", data=coconut_data)
        2 plt.xticks(rotation=90)
```

```
Out[37]: (array([1995. , 1997.5, 2000. , 2002.5, 2005. , 2007.5, 2010. , 2012.5,
                2015. , 2017.5]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[11]])
```



```
In [38]: 1 sns.jointplot("Area", "Production", data=coconut_data, kind="reg")
        2 plt.xticks(rotation=90)
```

```
Out[38]: (array([-25000.,      0.,  25000.,  50000.,  75000., 100000., 125000.,
        150000., 175000., 200000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.6000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[0]])
```



Observations obtained :

1. Andhra Pradesh is the largest producing cocnut states.
2. Production per unit area is higher in Mizoram ans Sikkim.
3. Coconut yield is decreasing in the year 2012 to 2015.

#### 4. POTATO

```
In [39]: 1 potato_data = crop_data[crop_data["Crop"]=="Potato"]
        2 potato_data
```

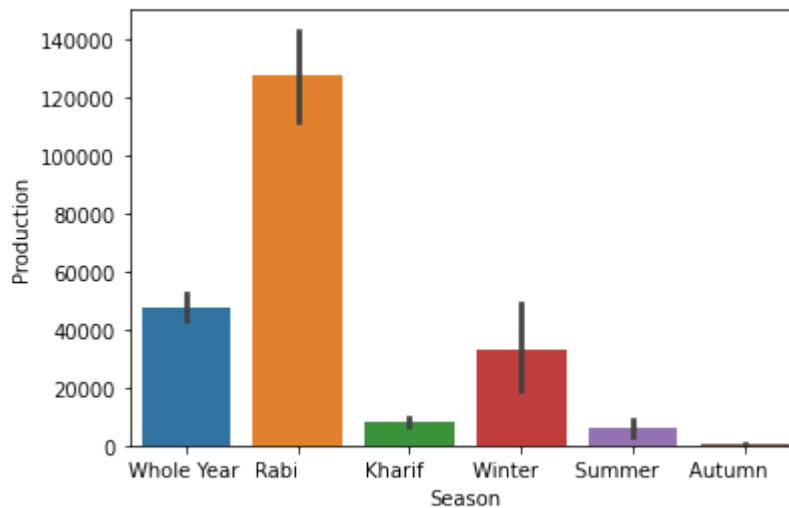
```
Out[39]:
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
331	Andhra Pradesh	ANANTAPUR	2000	Whole Year	Potato	4.0	34.0	8.500000
433	Andhra Pradesh	ANANTAPUR	2002	Whole Year	Potato	2.0	17.0	8.500000
530	Andhra Pradesh	ANANTAPUR	2004	Whole Year	Potato	2.0	20.0	10.000000
745	Andhra Pradesh	ANANTAPUR	2010	Whole Year	Potato	21.0	236.0	11.238095
792	Andhra Pradesh	ANANTAPUR	2011	Whole Year	Potato	18.0	181.0	10.055556
...	...	...	...	...	...	...	...	...
245945	West Bengal	PURULIA	2010	Rabi	Potato	977.0	15920.0	16.294780
245976	West Bengal	PURULIA	2011	Rabi	Potato	1077.0	17412.0	16.167131
246008	West Bengal	PURULIA	2012	Rabi	Potato	913.0	17203.0	18.842278
246043	West Bengal	PURULIA	2013	Rabi	Potato	1726.0	43703.0	25.320394
246080	West Bengal	PURULIA	2014	Rabi	Potato	477.0	9995.0	20.953878

6914 rows × 8 columns

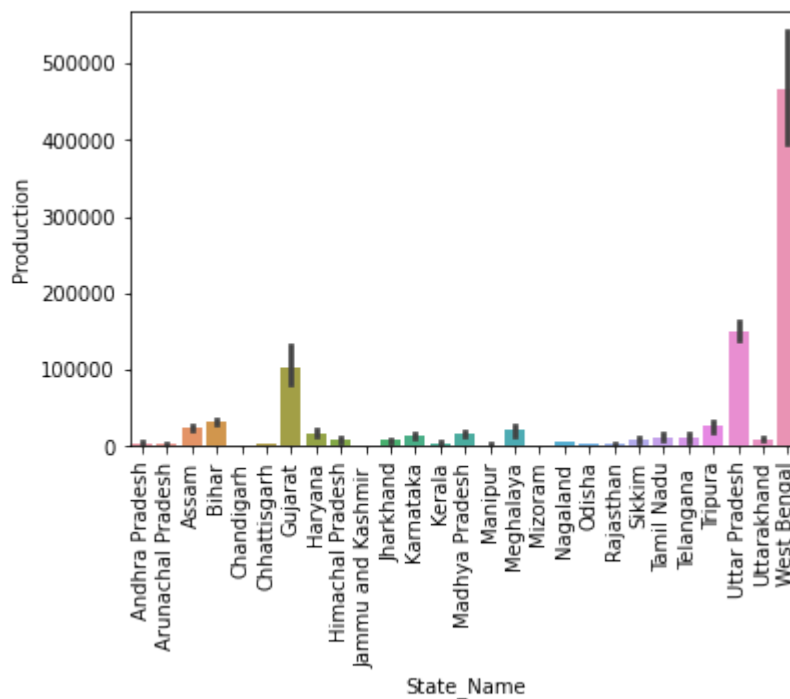
```
In [40]: 1 sns.barplot("Season", "Production", data = potato_data)
```

```
Out[40]: <AxesSubplot:xlabel='Season', ylabel='Production'>
```



```
In [41]: 1 sns.barplot("State_Name", "Production", data = potato_data)
2 plt.xticks(rotation = 90)
```

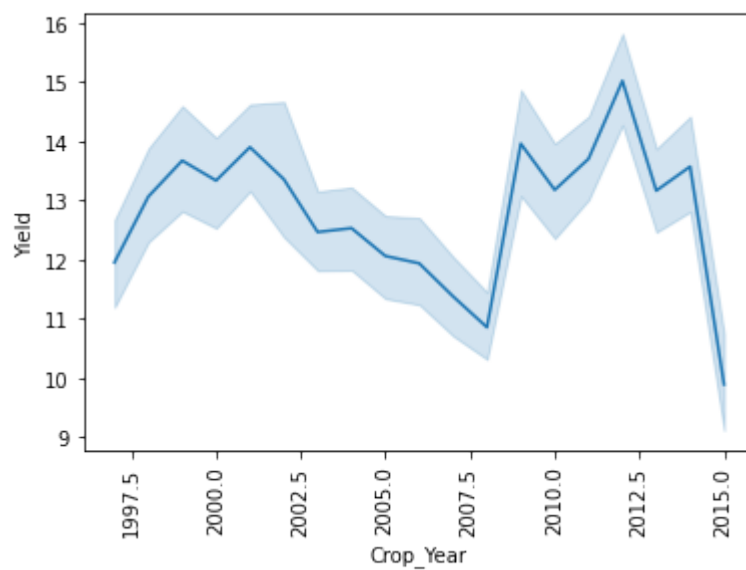
```
Out[41]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26])),
[Text(0, 0, 'Andhra Pradesh'),
Text(1, 0, 'Arunachal Pradesh'),
Text(2, 0, 'Assam'),
Text(3, 0, 'Bihar'),
Text(4, 0, 'Chandigarh'),
Text(5, 0, 'Chhattisgarh'),
Text(6, 0, 'Gujarat'),
Text(7, 0, 'Haryana'),
Text(8, 0, 'Himachal Pradesh'),
Text(9, 0, 'Jammu and Kashmir '),
Text(10, 0, 'Jharkhand'),
Text(11, 0, 'Karnataka'),
Text(12, 0, 'Kerala'),
Text(13, 0, 'Madhya Pradesh'),
Text(14, 0, 'Manipur'),
Text(15, 0, 'Meghalaya'),
Text(16, 0, 'Mizoram'),
Text(17, 0, 'Nagaland'),
Text(18, 0, 'Odisha'),
Text(19, 0, 'Rajasthan'),
Text(20, 0, 'Sikkim'),
Text(21, 0, 'Tamil Nadu'),
Text(22, 0, 'Telangana '),
Text(23, 0, 'Tripura'),
Text(24, 0, 'Uttar Pradesh'),
Text(25, 0, 'Uttarakhand'),
Text(26, 0, 'West Bengal')])
```





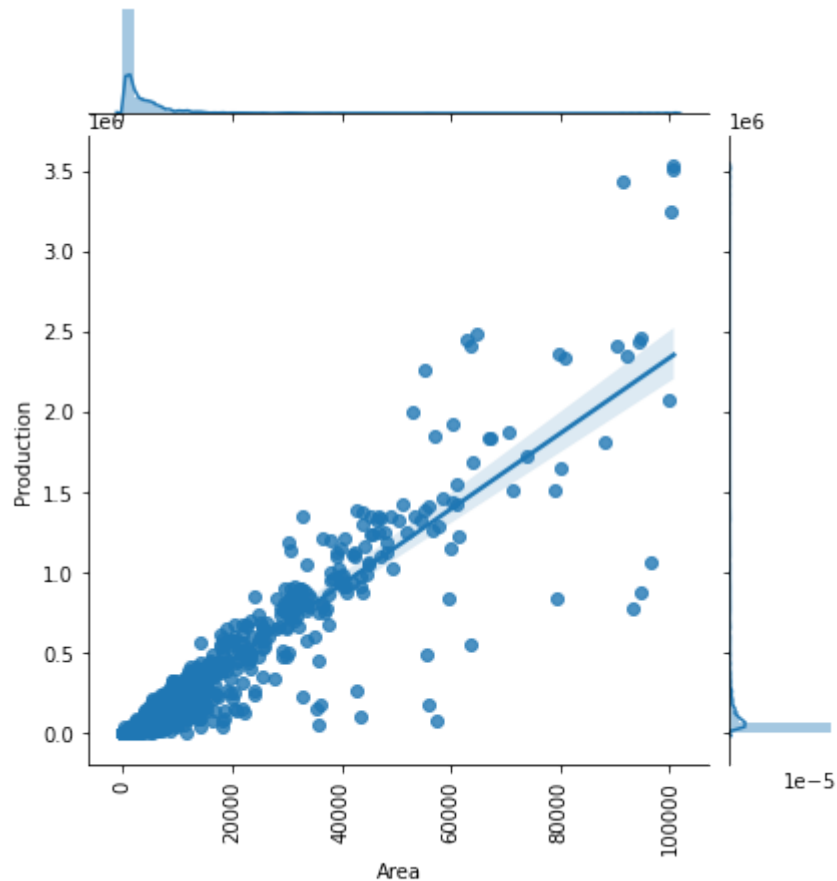
```
In [42]: 1 sns.lineplot("Crop_Year", "Yield", data=potato_data)
        2 plt.xticks(rotation=90)
```

```
Out[42]: (array([1995. , 1997.5, 2000. , 2002.5, 2005. , 2007.5, 2010. , 2012.5,
                2015. , 2017.5]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[12]])
```



```
In [43]: 1 sns.jointplot("Area", "Production", data=potato_data, kind="reg")
        2 plt.xticks(rotation=90)
```

```
Out[43]: (array([-20000.,      0.,  20000.,  40000.,  60000.,  80000., 100000.,
        120000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.6000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



Conclusions obtained :

1. Potato is a Rabi crop.
2. West Bengal is the largest producer of potatoes.

## 5. ONION

```
In [44]: 1 onion_data = crop_data[crop_data["Crop"]=="Onion"]
        2 onion_data
```

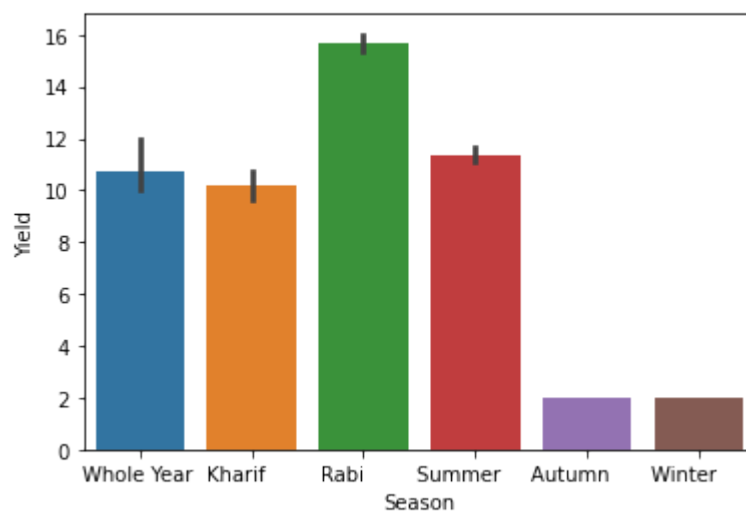
```
Out[44]:
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
286	Andhra Pradesh	ANANTAPUR	1999	Whole Year	Onion	1770.0	32364.0	18.284746
300	Andhra Pradesh	ANANTAPUR	2000	Kharif	Onion	1103.0	15470.0	14.025385
320	Andhra Pradesh	ANANTAPUR	2000	Rabi	Onion	482.0	11514.0	23.887967
348	Andhra Pradesh	ANANTAPUR	2001	Kharif	Onion	1165.0	19232.0	16.508155
365	Andhra Pradesh	ANANTAPUR	2001	Rabi	Onion	380.0	8465.0	22.276316
...	...	...	...	...	...	...	...	...
236375	Uttarakhand	UTTAR KASHI	2010	Whole Year	Onion	47.0	306.0	6.510638
236398	Uttarakhand	UTTAR KASHI	2011	Whole Year	Onion	8.0	52.0	6.500000
236420	Uttarakhand	UTTAR KASHI	2012	Whole Year	Onion	23.0	150.0	6.521739
236443	Uttarakhand	UTTAR KASHI	2013	Whole Year	Onion	32.0	212.0	6.625000
236473	Uttarakhand	UTTAR KASHI	2014	Whole Year	Onion	26.0	172.0	6.615385

6984 rows × 8 columns

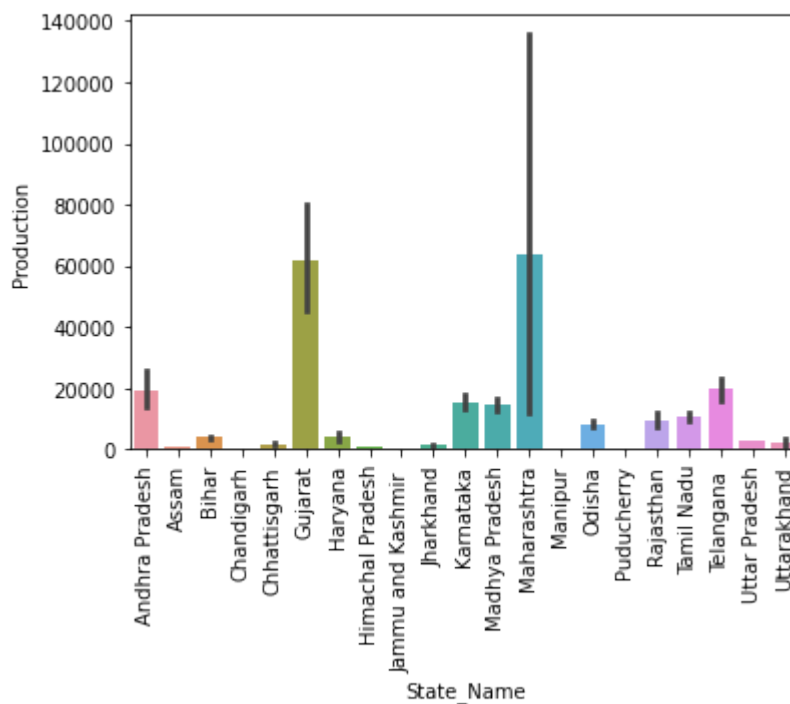
```
In [45]: 1 sns.barplot("Season","Yield",data = onion_data)
```

```
Out[45]: <AxesSubplot:xlabel='Season', ylabel='Yield'>
```



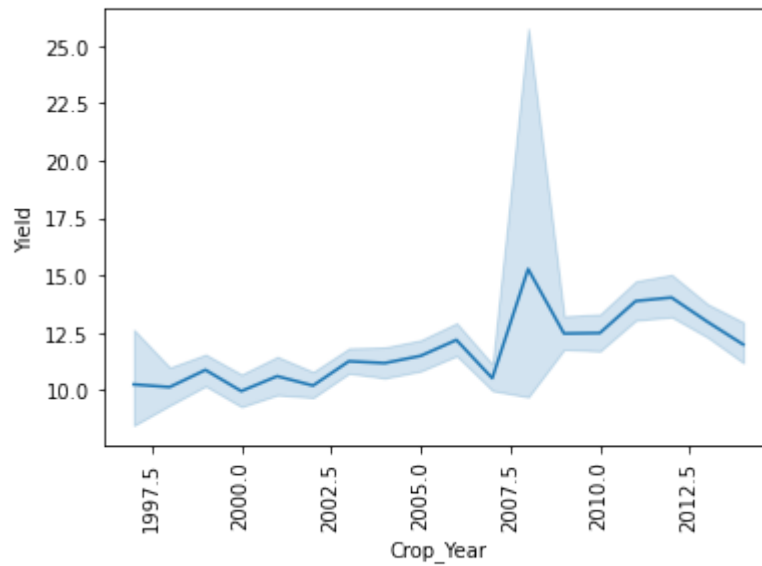
```
In [46]: 1 sns.barplot("State_Name", "Production", data = onion_data)
        2 plt.xticks(rotation = 90)
```

```
Out[46]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20]),
 [Text(0, 0, 'Andhra Pradesh'),
  Text(1, 0, 'Assam'),
  Text(2, 0, 'Bihar'),
  Text(3, 0, 'Chandigarh'),
  Text(4, 0, 'Chhattisgarh'),
  Text(5, 0, 'Gujarat'),
  Text(6, 0, 'Haryana'),
  Text(7, 0, 'Himachal Pradesh'),
  Text(8, 0, 'Jammu and Kashmir '),
  Text(9, 0, 'Jharkhand'),
  Text(10, 0, 'Karnataka'),
  Text(11, 0, 'Madhya Pradesh'),
  Text(12, 0, 'Maharashtra'),
  Text(13, 0, 'Manipur'),
  Text(14, 0, 'Odisha'),
  Text(15, 0, 'Puducherry'),
  Text(16, 0, 'Rajasthan'),
  Text(17, 0, 'Tamil Nadu'),
  Text(18, 0, 'Telangana '),
  Text(19, 0, 'Uttar Pradesh'),
  Text(20, 0, 'Uttarakhand')])
```



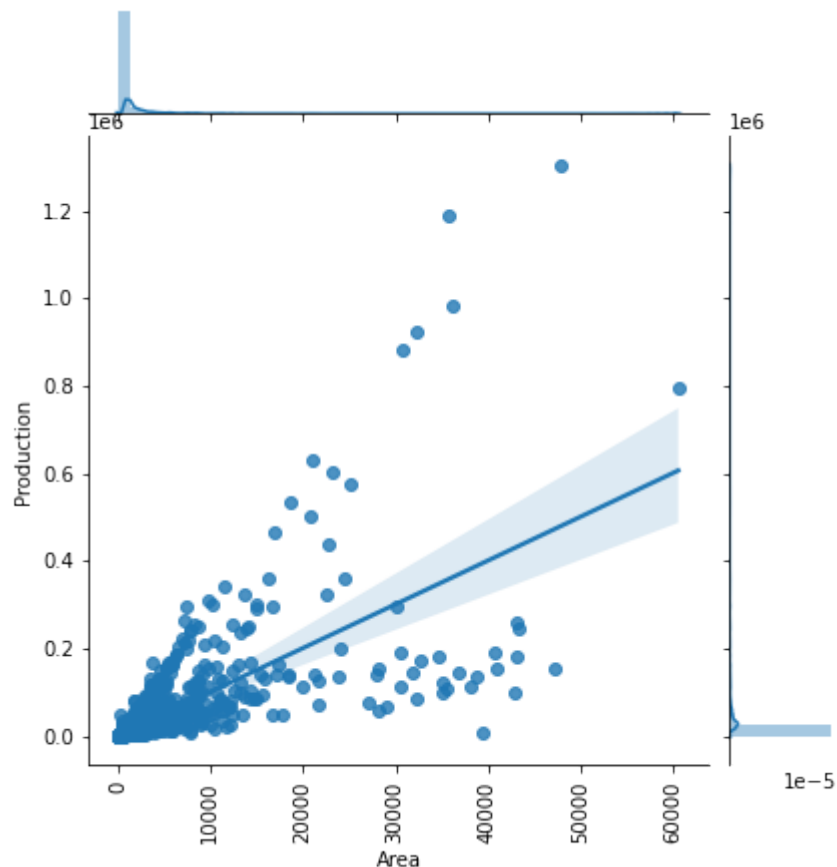
```
In [47]: 1 sns.lineplot("Crop_Year", "Yield", data=onion_data)
        2 plt.xticks(rotation=90)
```

```
Out[47]: (array([1995. , 1997.5, 2000. , 2002.5, 2005. , 2007.5, 2010. , 2012.5,
                2015. ]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[11]])
```



```
In [48]: 1 sns.jointplot("Area", "Production", data=onion_data, kind="reg")
2 plt.xticks(rotation=90)
```

```
Out[48]: (array([-10000.,    0., 10000., 20000., 30000., 40000., 50000.,
        60000., 70000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.6000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



Observations :

1. Onion is a Rabi crop.
2. Gujrat and Maharastra are the major onion-producing states.

## 6. MAIZE

```
In [49]: 1 maize_data = crop_data[crop_data["Crop"]=="Maize"]
         2 maize_data
```

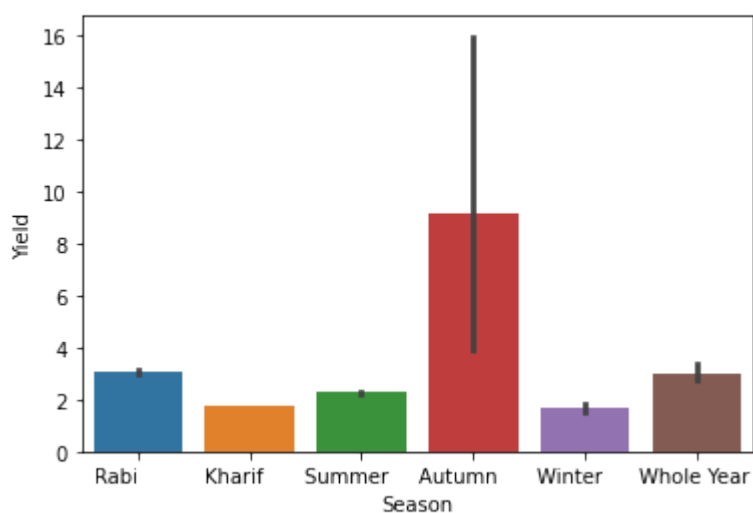
```
Out[49]:
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
71	Andaman and Nicobar Islands	NICOBARS	2010	Rabi	Maize	3.84	18.22	4.744792
120	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	2010	Rabi	Maize	86.70	96.40	1.111880
194	Andaman and Nicobar Islands	SOUTH ANDAMANS	2010	Rabi	Maize	73.00	253.00	3.465753
212	Andhra Pradesh	ANANTAPUR	1997	Kharif	Maize	2800.00	4900.00	1.750000
226	Andhra Pradesh	ANANTAPUR	1997	Rabi	Maize	600.00	2400.00	4.000000
...	...	...	...	...	...	...	...	...
246013	West Bengal	PURULIA	2012	Summer	Maize	290.00	476.00	1.641379
246020	West Bengal	PURULIA	2013	Autumn	Maize	6189.00	10665.00	1.723219
246048	West Bengal	PURULIA	2013	Summer	Maize	325.00	522.00	1.606154
246056	West Bengal	PURULIA	2014	Autumn	Maize	6317.00	13337.00	2.111287
246085	West Bengal	PURULIA	2014	Summer	Maize	325.00	2039.00	6.273846

13787 rows × 8 columns

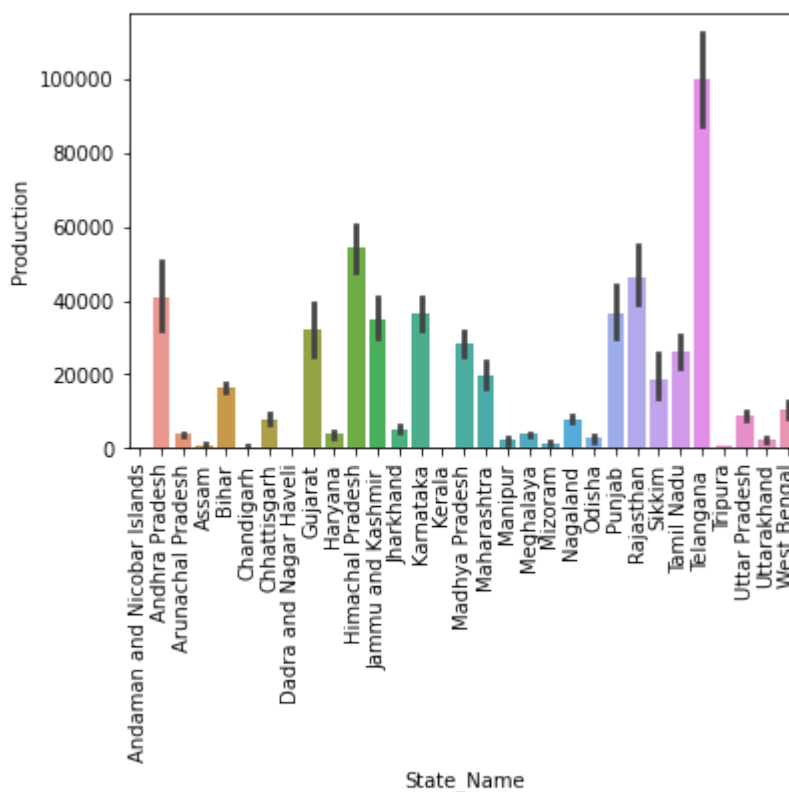
```
In [50]: 1 sns.barplot("Season", "Yield", data = maize_data)
```

```
Out[50]: <AxesSubplot:xlabel='Season', ylabel='Yield'>
```



```
In [51]: 1 sns.barplot("State_Name", "Production", data = maize_data)
2 plt.xticks(rotation = 90)
```

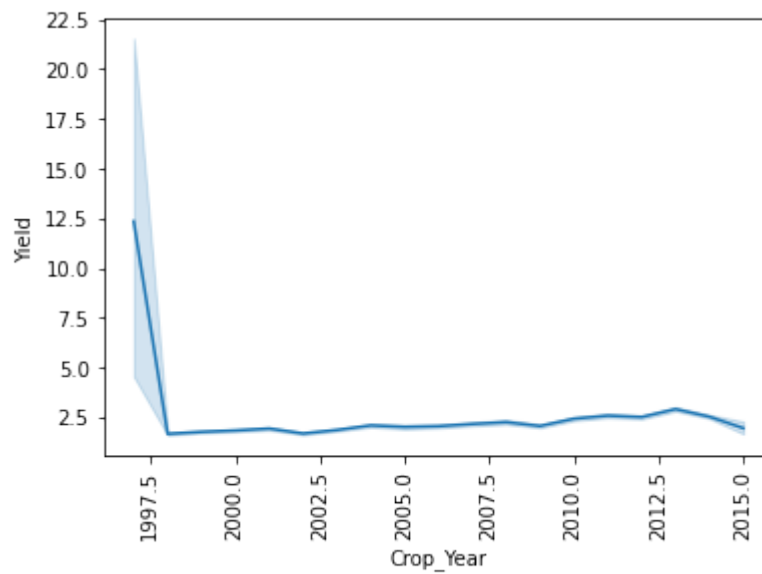
```
Out[51]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]),
 [Text(0, 0, 'Andaman and Nicobar Islands'),
 Text(1, 0, 'Andhra Pradesh'),
 Text(2, 0, 'Arunachal Pradesh'),
 Text(3, 0, 'Assam'),
 Text(4, 0, 'Bihar'),
 Text(5, 0, 'Chandigarh'),
 Text(6, 0, 'Chhattisgarh'),
 Text(7, 0, 'Dadra and Nagar Haveli'),
 Text(8, 0, 'Gujarat'),
 Text(9, 0, 'Haryana'),
 Text(10, 0, 'Himachal Pradesh'),
 Text(11, 0, 'Jammu and Kashmir '),
 Text(12, 0, 'Jharkhand'),
 Text(13, 0, 'Karnataka'),
 Text(14, 0, 'Kerala'),
 Text(15, 0, 'Madhya Pradesh'),
 Text(16, 0, 'Maharashtra'),
 Text(17, 0, 'Manipur'),
 Text(18, 0, 'Meghalaya'),
 Text(19, 0, 'Mizoram'),
 Text(20, 0, 'Nagaland'),
 Text(21, 0, 'Odisha'),
 Text(22, 0, 'Punjab'),
 Text(23, 0, 'Rajasthan'),
 Text(24, 0, 'Sikkim'),
 Text(25, 0, 'Tamil Nadu'),
 Text(26, 0, 'Telangana '),
 Text(27, 0, 'Tripura'),
 Text(28, 0, 'Uttar Pradesh'),
 Text(29, 0, 'Uttarakhand'),
 Text(30, 0, 'West Bengal')])
```





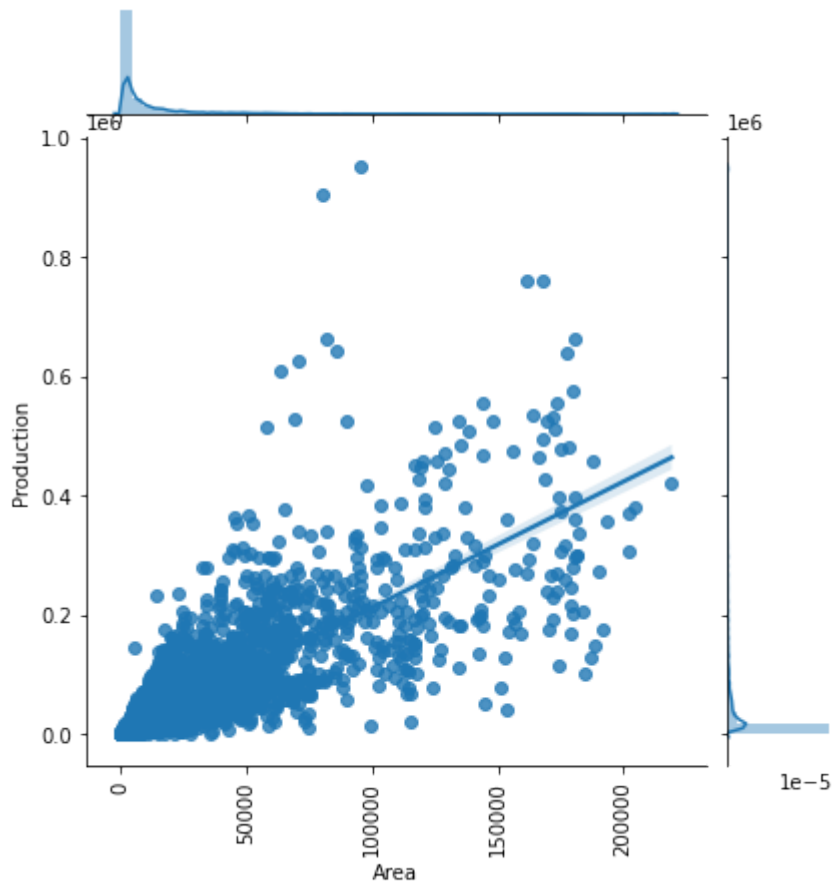
```
In [52]: 1 sns.lineplot("Crop_Year", "Yield", data=maize_data)
         2 plt.xticks(rotation=90)
```

```
Out[52]: (array([1995. , 1997.5, 2000. , 2002.5, 2005. , 2007.5, 2010. , 2012.5,
                2015. , 2017.5]),
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')[11]])
```



```
In [53]: 1 sns.jointplot("Area", "Production", data=maize_data, kind="reg")
2 plt.xticks(rotation=90)
```

```
Out[53]: (array([-50000.,    0.,  50000., 100000., 150000., 200000., 250000.]),
 [Text(0.0, 0, '0.0'),
  Text(0.2, 0, '0.2'),
  Text(0.4, 0, '0.4'),
  Text(0.60000000000000001, 0, '0.6'),
  Text(0.8, 0, '0.8'),
  Text(1.0, 0, '1.0'),
  Text(0, 0, '')])
```



Observations Obtained -

1. Maize is produced in the autumn season
2. Telangana is the major maize-producing states.
3. There was a sudden decline in maize production from the year 2000.