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

In [2]:
```
df = pd.read_csv(r"C:\Users\admin\Downloads\crop.csv")
df
```

Out[2]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units
0	Andaman and Nicobar Islands	NICOBARS	Arecanut	2001-02	Kharif	1254.0	Hectare	2061.0	Tonnes
1	Andaman and Nicobar Islands	NICOBARS	Arecanut	2002-03	Whole Year	1258.0	Hectare	2083.0	Tonnes
2	Andaman and Nicobar Islands	NICOBARS	Arecanut	2003-04	Whole Year	1261.0	Hectare	1525.0	Tonnes
3	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Arecanut	2001-02	Kharif	3100.0	Hectare	5239.0	Tonnes
4	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2002-03	Whole Year	3105.0	Hectare	5267.0	Tonnes
...	...	...	...	...	...	...	...	...	...
344203	Manipur	IMPHAL WEST	NaN	2019-20	Rabi	NaN	Hectare	NaN	Tonnes
344204	Manipur	SENAPATI	NaN	2019-20	Rabi	NaN	Hectare	NaN	Tonnes
344205	Manipur	TAMENGLONG	NaN	2019-20	Rabi	NaN	Hectare	NaN	Tonnes
344206	Manipur	THOUBAL	NaN	2019-20	Rabi	NaN	Hectare	NaN	Tonnes
344207	Manipur	UKHRUL	NaN	2019-20	Rabi	NaN	Hectare	NaN	Tonnes

344208 rows × 10 columns

Data Exploration

In [3]:
```
df.isnull().sum()
```

Out[3]:

```
State                0
District             0
Crop               109
Year                 0
Season               0
Area               109
Area Units           0
Production        5021
Production Units     0
Yield              109
dtype: int64
```

In [4]:

```
# Droping Empty Values
data = df.dropna()
print(data.shape)
test = df[~df["Production"].notna()].drop("Production",axis=1)
print(test.shape)
```

```
(339187, 10)
(5021, 9)
```

In [5]:

```
data
```

Out[5]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units
0	Andaman and Nicobar Islands	NICOBARS	Arecanut	2001-02	Kharif	1254.0	Hectare	2061.0	Tonnes
1	Andaman and Nicobar Islands	NICOBARS	Arecanut	2002-03	Whole Year	1258.0	Hectare	2083.0	Tonnes
2	Andaman and Nicobar Islands	NICOBARS	Arecanut	2003-04	Whole Year	1261.0	Hectare	1525.0	Tonnes
3	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Arecanut	2001-02	Kharif	3100.0	Hectare	5239.0	Tonnes
4	Andaman and Nicobar Islands	SOUTH ANDAMANS	Arecanut	2002-03	Whole Year	3105.0	Hectare	5267.0	Tonnes
...	...	...	...	...	...	...	...	...	...
344094	West Bengal	PURBA BARDHAMAN	Wheat	2000-01	Rabi	6310.0	Hectare	15280.0	Tonnes
344095	West Bengal	PURULIA	Wheat	1997-98	Rabi	1895.0	Hectare	2760.0	Tonnes
344096	West Bengal	PURULIA	Wheat	1998-99	Rabi	3736.0	Hectare	5530.0	Tonnes
344097	West Bengal	PURULIA	Wheat	1999-00	Rabi	2752.0	Hectare	6928.0	Tonnes

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units
344098	West Bengal	PURULIA	Wheat	2000-01	Rabi	2979.0	Hectare	7430.0	Tonnes

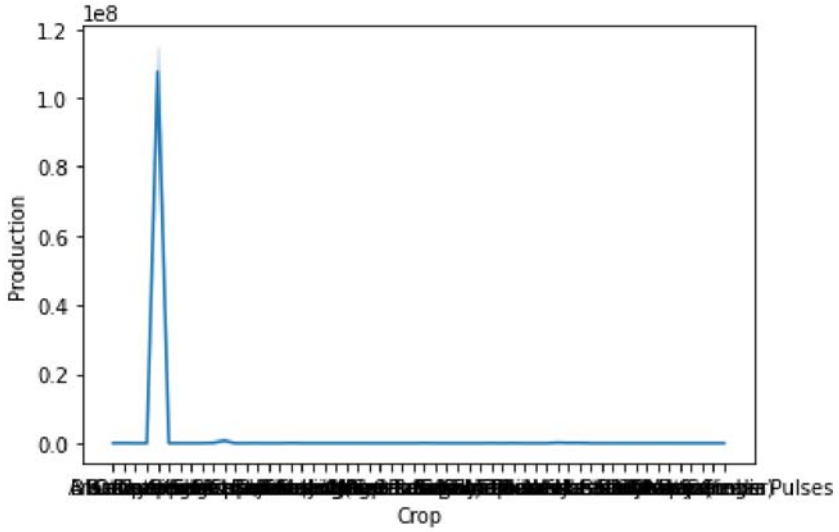
339187 rows × 10 columns

Data Visulization

In [6]:
```
sns.lineplot(data["Crop"], data["Production"])
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[6]: <AxesSubplot:xlabel='Crop', ylabel='Production'>

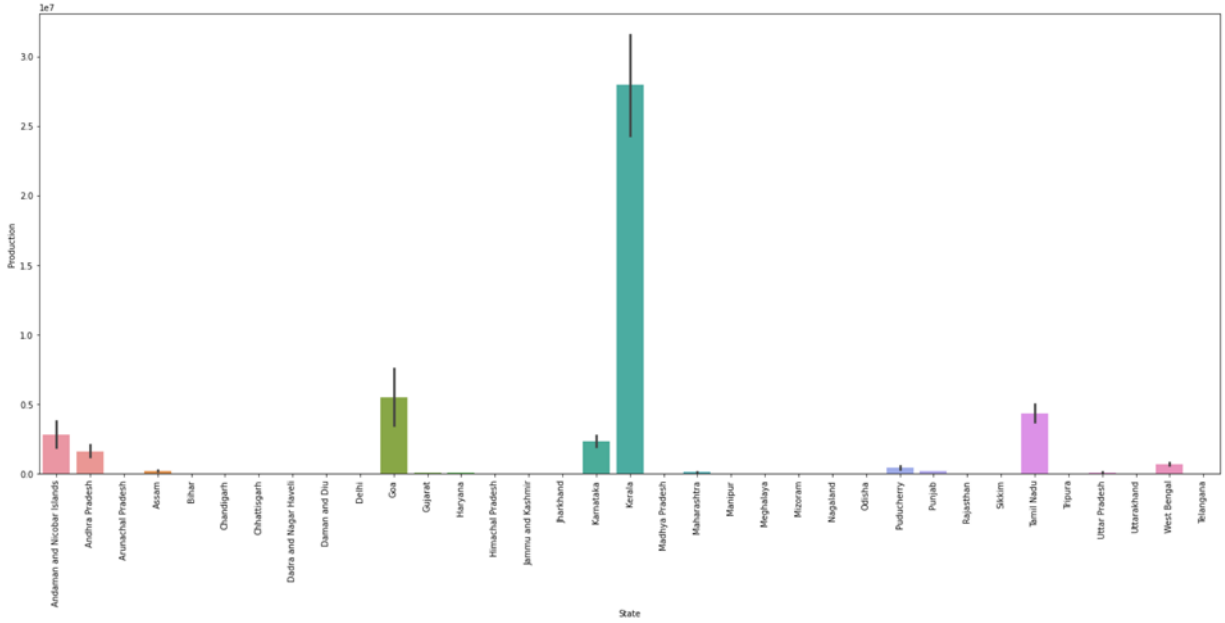


In [7]:
```
plt.figure(figsize = (25, 10))
sns.barplot(data["State"], data["Production"])
plt.xticks(rotation = 90)
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[7]:
```
(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]),
 [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, 'Daman and Diu'),
  Text(9, 0, 'Delhi'),
  Text(10, 0, 'Goa'),
```

```
Text(11, 0, 'Gujarat'),
Text(12, 0, 'Haryana'),
Text(13, 0, 'Himachal Pradesh'),
Text(14, 0, 'Jammu and Kashmir'),
Text(15, 0, 'Jharkhand'),
Text(16, 0, 'Karnataka'),
Text(17, 0, 'Kerala'),
Text(18, 0, 'Madhya Pradesh'),
Text(19, 0, 'Maharashtra'),
Text(20, 0, 'Manipur'),
Text(21, 0, 'Meghalaya'),
Text(22, 0, 'Mizoram'),
Text(23, 0, 'Nagaland'),
Text(24, 0, 'Odisha'),
Text(25, 0, 'Puducherry'),
Text(26, 0, 'Punjab'),
Text(27, 0, 'Rajasthan'),
Text(28, 0, 'Sikkim'),
Text(29, 0, 'Tamil Nadu'),
Text(30, 0, 'Tripura'),
Text(31, 0, 'Uttar Pradesh'),
Text(32, 0, 'Uttarakhand'),
Text(33, 0, 'West Bengal'),
Text(34, 0, 'Telangana')])
```

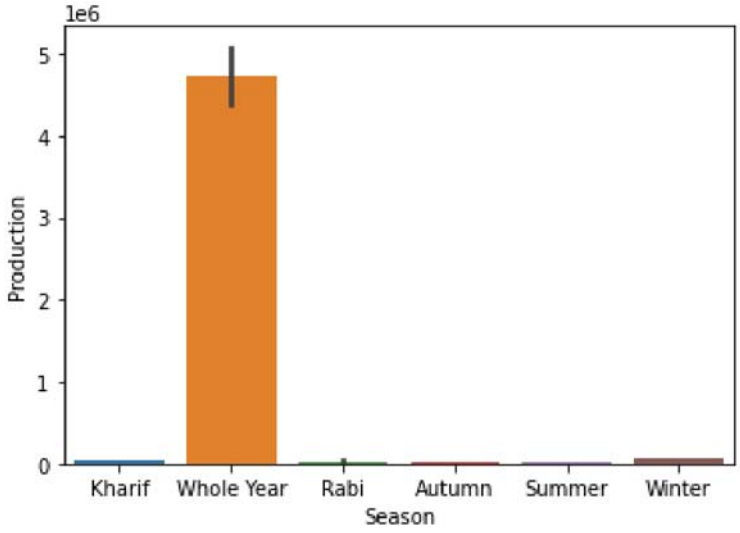


In [8]:

```
sns.barplot(data["Season"], data["Production"])
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[8]: <AxesSubplot:xlabel='Season', ylabel='Production'>



In [9]:
```
data.groupby("Season", axis=0).agg({"Production" : np.sum})
```

Out[9]:

Season	Production
Autumn	8.464143e+07
Kharif	5.612134e+09
Rabi	3.149021e+09
Summer	2.437104e+08
Whole Year	3.165385e+11
Winter	5.877503e+08

In [10]:
```
data["Crop"].value_counts()[:5]
```

Out[10]:
```
Rice                 21529
Maize                20284
Moong(Green Gram)    14758
Urad                 14320
Sesamum              12704
Name: Crop, dtype: int64
```

In [11]:
```
top_crop_pro = data.groupby("Crop")["Production"].sum().reset_index().sort_values(by
top_crop_pro[:5]
```

Out[11]:

	Crop	Production
9	Coconut	3.100040e+11
47	Sugarcane	7.224526e+09
41	Rice	2.227134e+09
54	Wheat	2.006287e+09
38	Potato	6.321391e+08

Exploring top 3 crops

1. Rice

In [12]:
```
rice_df = data[data["Crop"] == "Rice"]
print(rice_df.shape)
rice_df[:3]
```

```
(21529, 10)
```

Out[12]:

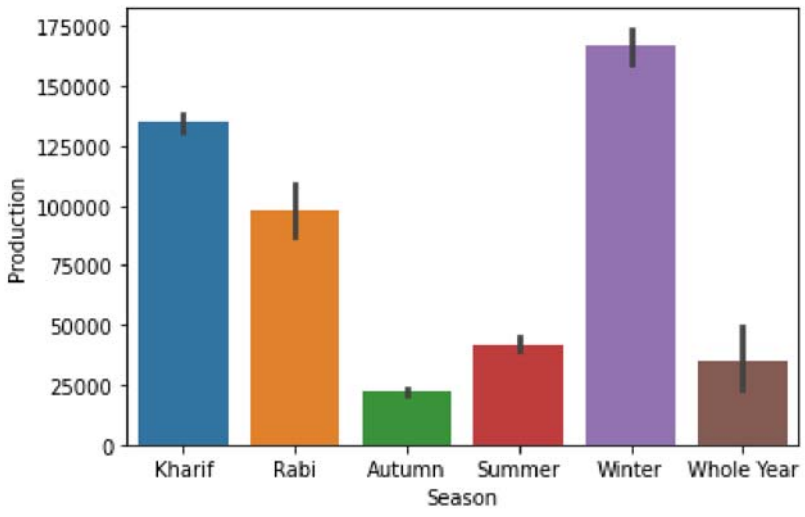
	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yield
41	Andaman and Nicobar Islands	NICOBARS	Rice	2001-02	Kharif	83.0	Hectare	300.00	Tonnes	3.614458
42	Andaman and Nicobar Islands	NICOBARS	Rice	2002-03	Kharif	189.2	Hectare	510.84	Tonnes	2.700000
43	Andaman and Nicobar Islands	NICOBARS	Rice	2003-04	Kharif	52.0	Hectare	90.17	Tonnes	1.734038

In [13]:
```
sns.barplot("Season", "Production", data=rice_df)
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[13]:

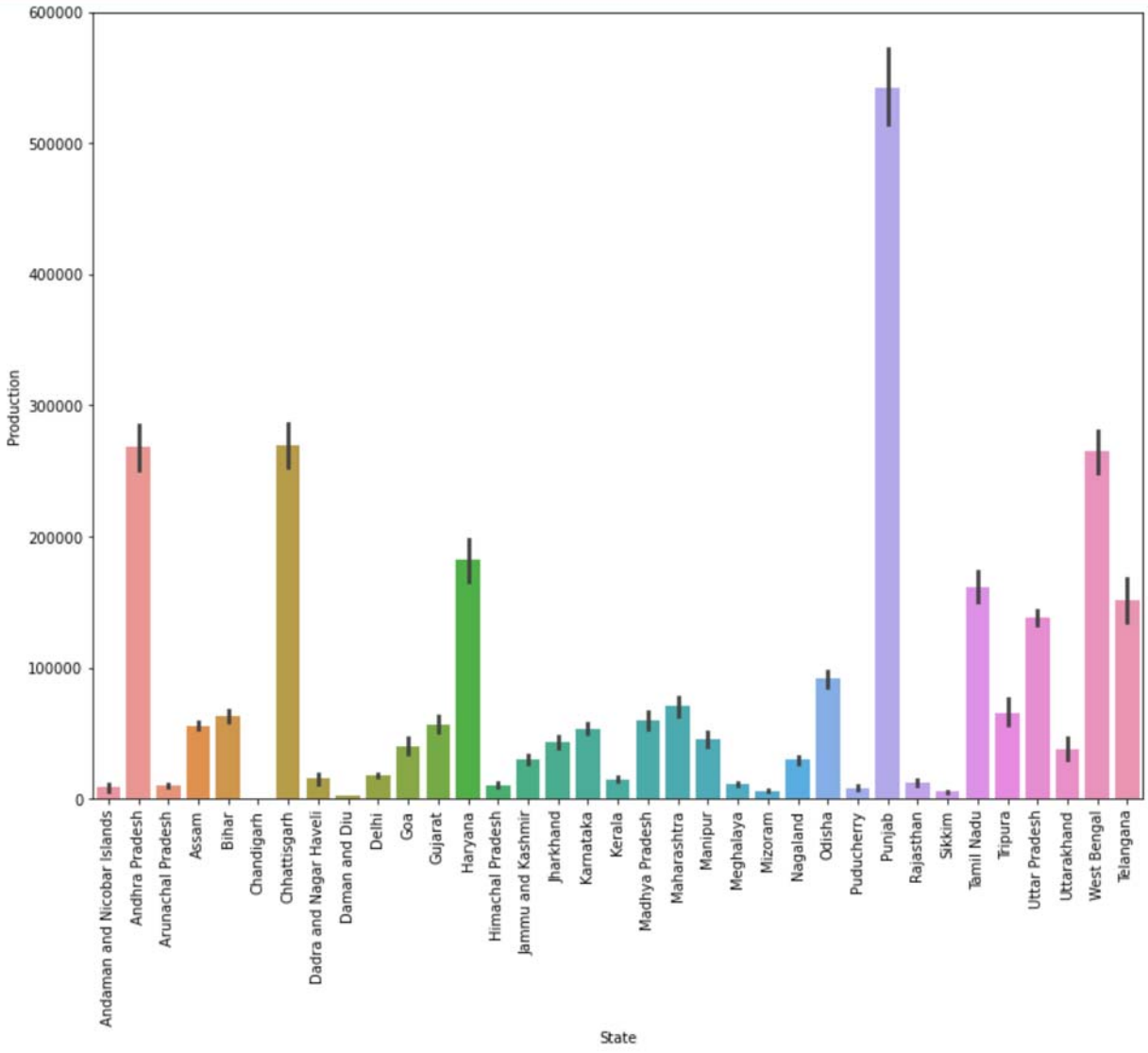
```
<AxesSubplot:xlabel='Season', ylabel='Production'>
```



In [14]:
```
plt.figure(figsize = (13, 10))
sns.barplot("State", "Production", data = rice_df)
plt.xticks(rotation = 90)
plt.show()
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
```

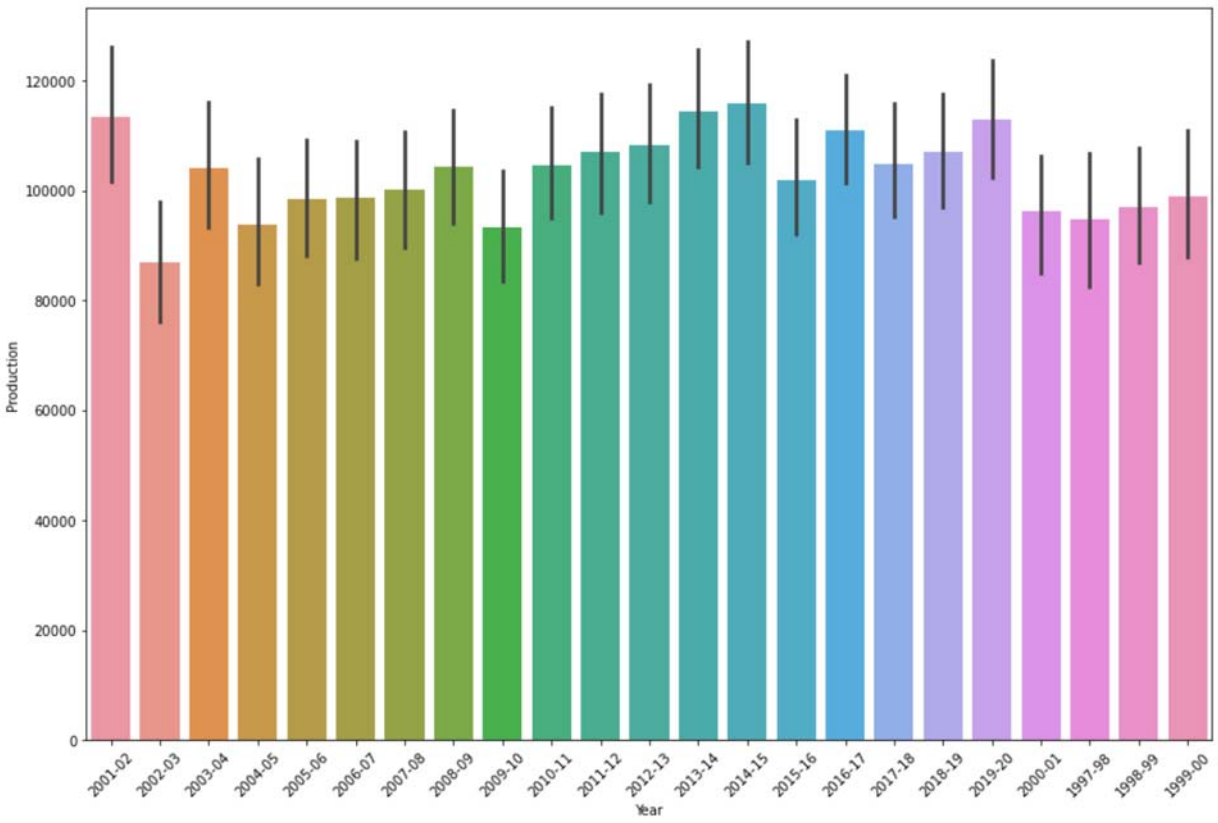
```
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [15]:

```
plt.figure(figsize = (15, 10))
sns.barplot("Year", "Production", data = rice_df)
plt.xticks(rotation = 45)
plt.show()
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



2. Coconut

In [16]:
```
coc_df = data[data["Crop"] == "Coconut"]
print(coc_df.shape)
coc_df[:3]
```

```
(2891, 10)
```

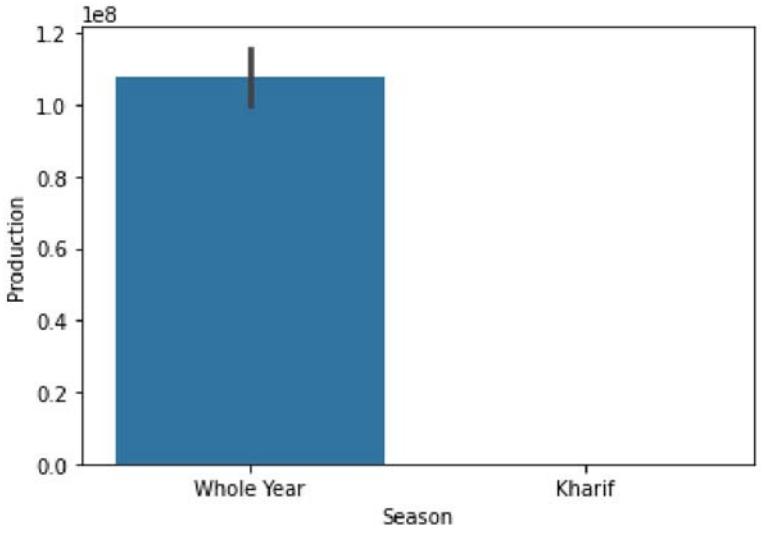
Out[16]:

	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	
20	Andaman and Nicobar Islands	NICOBARS	Coconut	2001-02	Whole Year	18190.00	Hectare	64430000.0	Nuts	3542.0
21	Andaman and Nicobar Islands	NICOBARS	Coconut	2002-03	Whole Year	18240.00	Hectare	67490000.0	Nuts	3700.1
22	Andaman and Nicobar Islands	NICOBARS	Coconut	2003-04	Whole Year	18284.74	Hectare	68580000.0	Nuts	3750.6

In [17]:
```
sns.barplot("Season", "Production", data = coc_df)
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

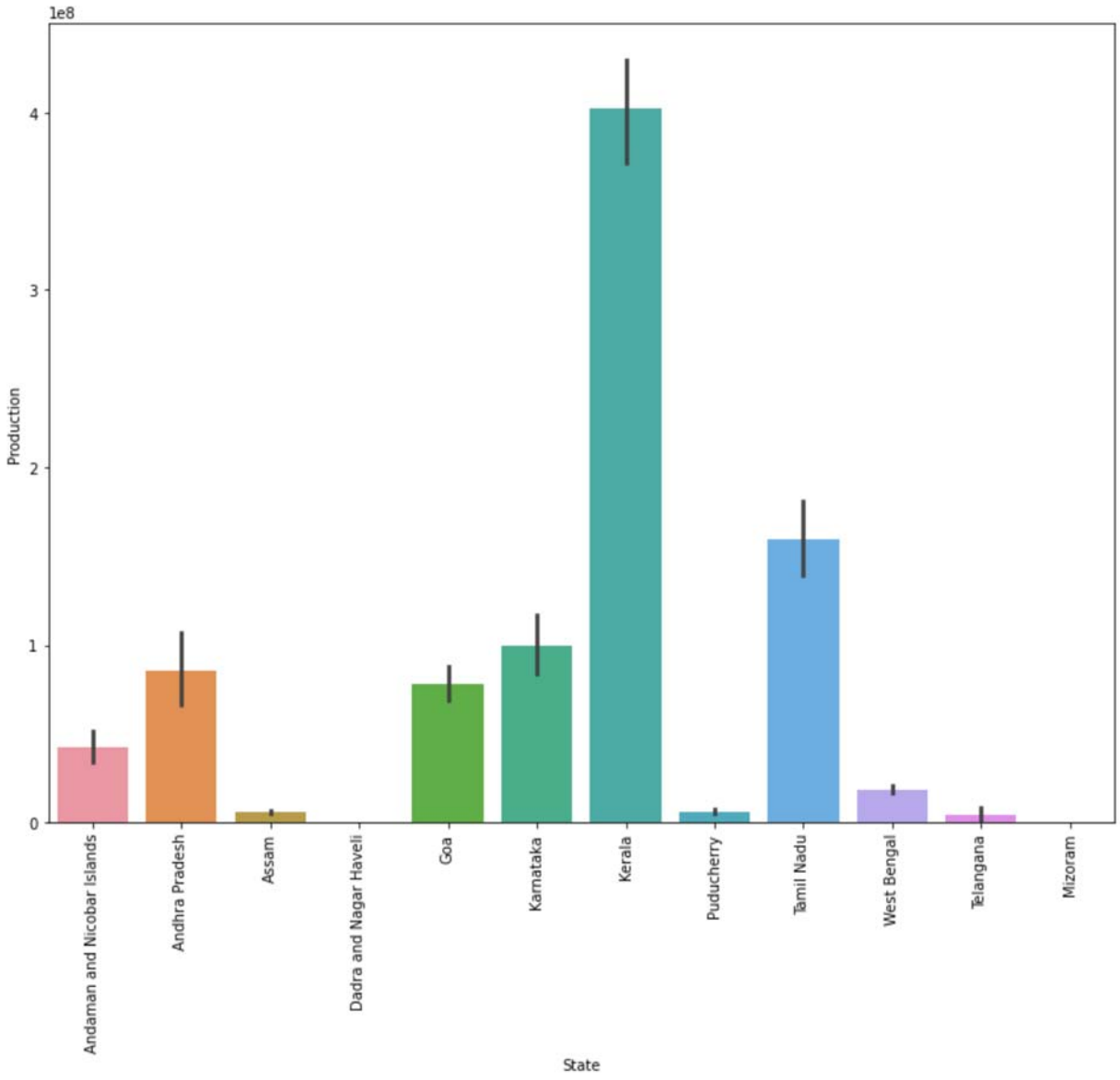
Out[17]: <AxesSubplot:xlabel='Season', ylabel='Production'>



In [18]:

```
plt.figure(figsize = (13, 10))
sns.barplot("State", "Production", data = coc_df)
plt.xticks(rotation = 90)
plt.show()
```

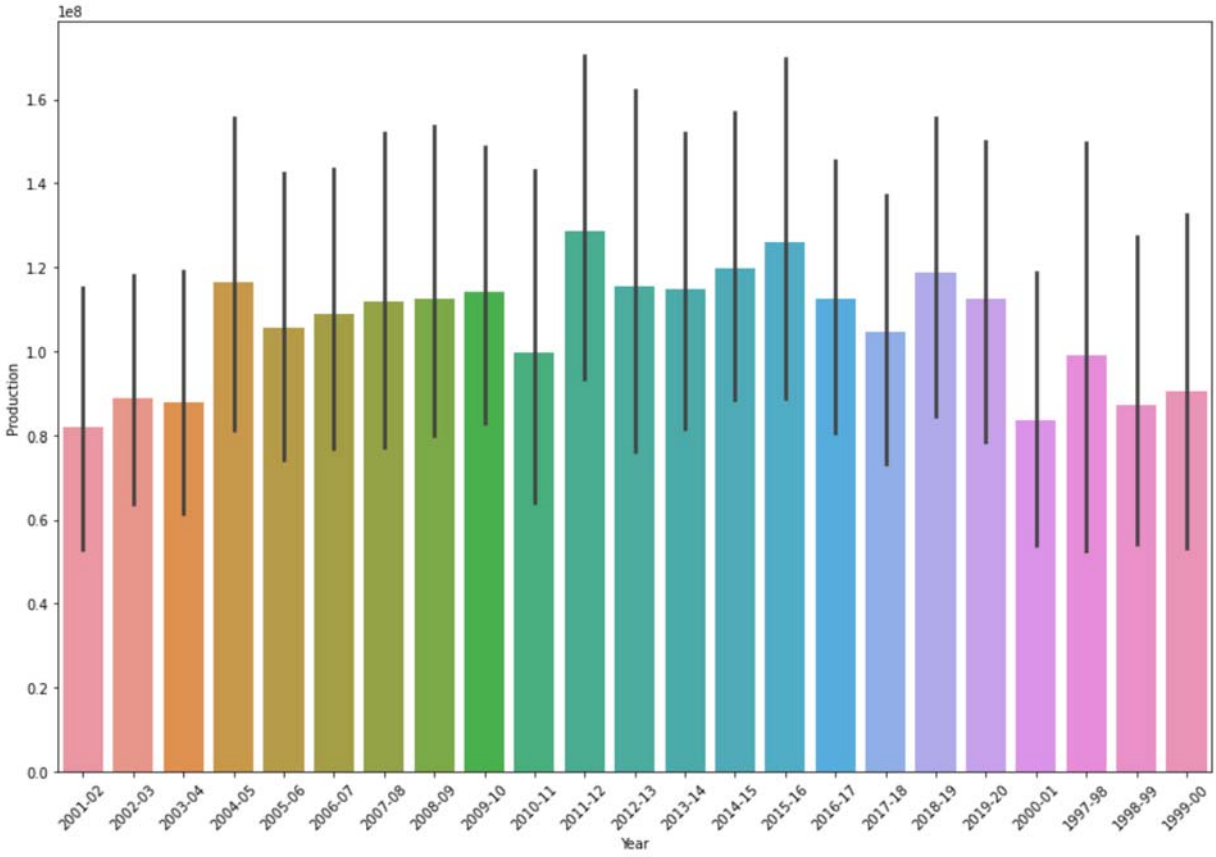
```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [19]:

```
plt.figure(figsize = (15, 10))
sns.barplot("Year", "Production", data = coc_df)
plt.xticks(rotation = 45)
plt.show()
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



3. Sugarcane

In [20]:

```
sug_df = data[data["Crop"] == "Sugarcane"]
print(sug_df.shape)
sug_df[:3]
```

```
(10800, 10)
```

Out[20]:

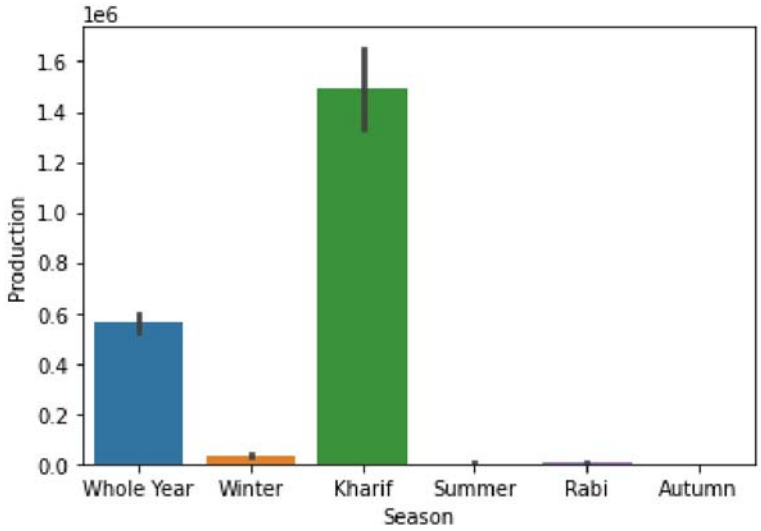
	State	District	Crop	Year	Season	Area	Area Units	Production	Production Units	Yiel
47	Andaman and Nicobar Islands	NICOBARS	Sugarcane	2001-02	Whole Year	1.0	Hectare	1.0	Tonnes	1.0000
48	Andaman and Nicobar Islands	NICOBARS	Sugarcane	2002-03	Whole Year	5.0	Hectare	40.0	Tonnes	8.0000
49	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	Sugarcane	2001-02	Whole Year	81.0	Hectare	2379.0	Tonnes	29.3703

In [21]:

```
sns.barplot("Season", "Production", data = sug_df)
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

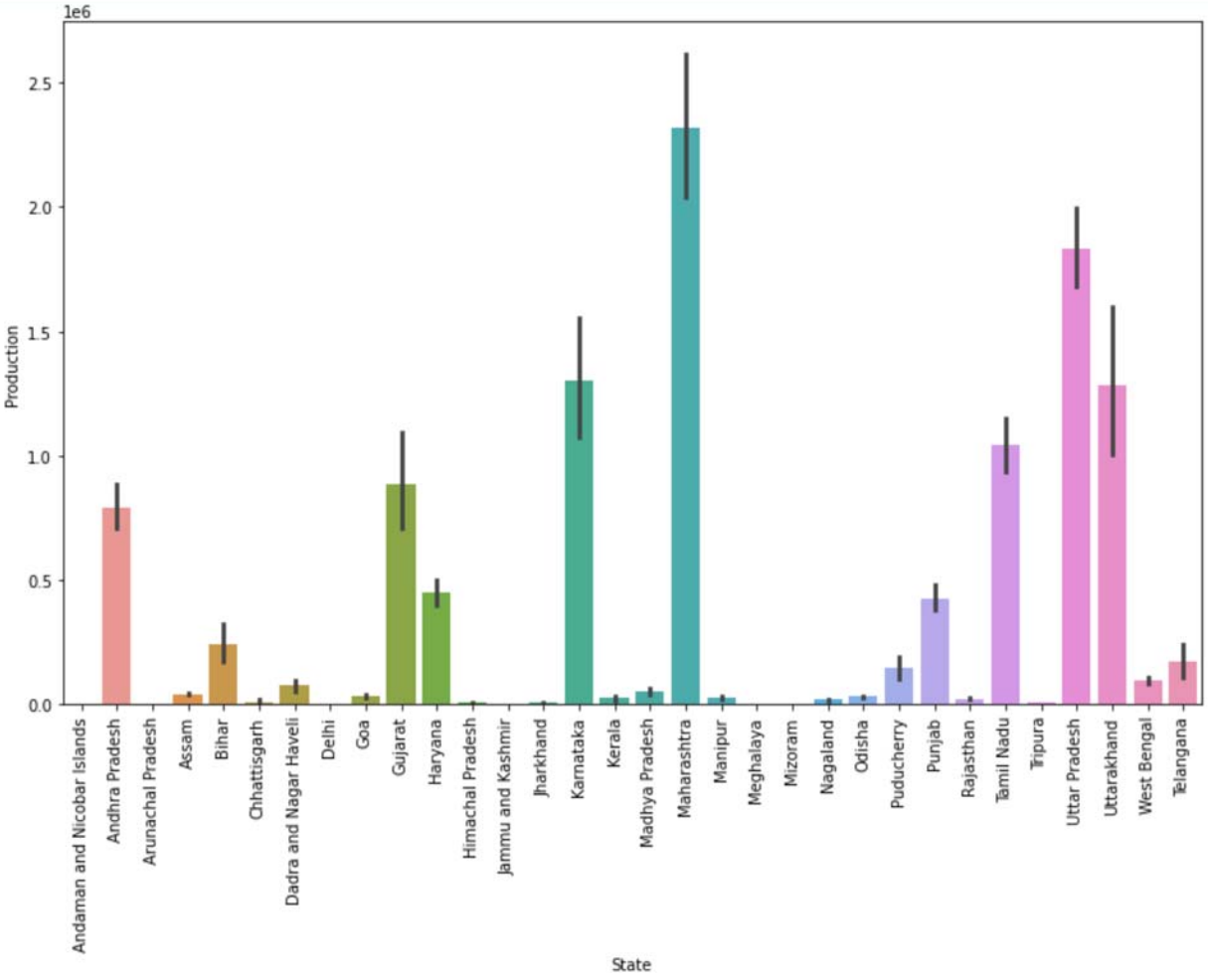
Out[21]: <AxesSubplot:xlabel='Season', ylabel='Production'>



In [22]:

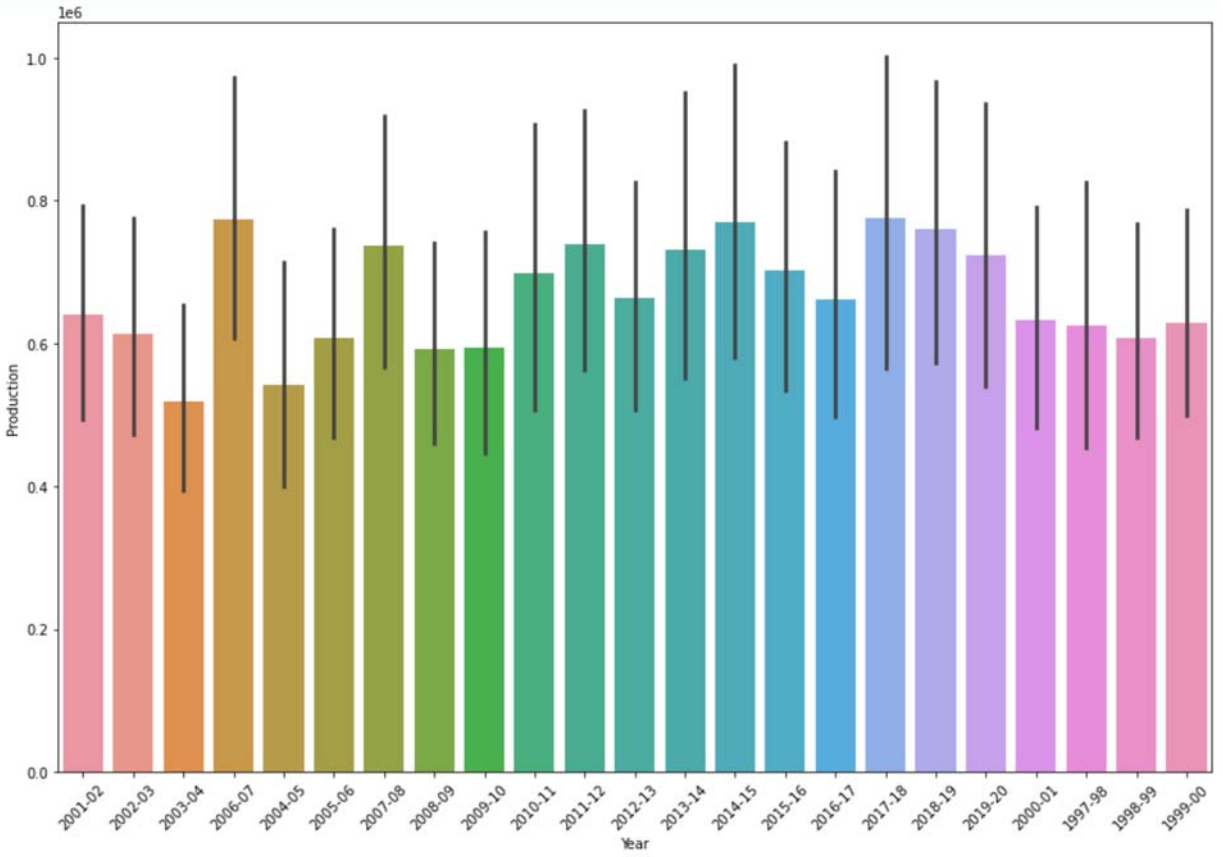
```
plt.figure(figsize = (13, 8))
sns.barplot("State", "Production", data = sug_df)
plt.xticks(rotation = 90)
plt.show()
```

```
C:\Users\admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [23]:
```
plt.figure(figsize = (15, 10))
sns.barplot("Year", "Production", data = sug_df)
plt.xticks(rotation = 45)
plt.show()
```

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



Feature Selection

In [24]:
```
data1 = data.drop(["District", "Year"], axis = 1)
```

In [25]:
```
data_dum = pd.get_dummies(data1)
data_dum[:5]
```

Out[25]:

	Area	Production	Yield	State_Andaman and Nicobar Islands	State_Andhra Pradesh	State_Arunachal Pradesh	State_Assam	State_
0	1254.0	2061.0	1.643541	1	0	0	0	
1	1258.0	2083.0	1.655803	1	0	0	0	
2	1261.0	1525.0	1.209358	1	0	0	0	
3	3100.0	5239.0	1.690000	1	0	0	0	
4	3105.0	5267.0	1.696296	1	0	0	0	

5 rows × 104 columns

Test Train Split

In [26]:
```
x = data_dum.drop("Production",axis=1)
y = data_dum["Production"]
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.33, random_s
print("x_train:", x_train.shape)
print("x_test:", x_test.shape)
print("y_train:", y_train.shape)
print("y_test:", y_test.shape)
```

```
x_train: (227255, 103)
x_test: (111932, 103)
y_train: (227255,)
y_test: (111932,)
```

In [27]:
```
x_train[:5]
```

Out[27]:

	Area	Yield	State_Andaman and Nicobar Islands	State_Andhra Pradesh	State_Arunachal Pradesh	State_Assam	State_Bihar	S
102882	79.0	0.379747	0	0	0	0	0	
67513	200.0	0.500000	0	0	0	0	0	
259793	43.0	3.023256	0	0	0	0	0	
205129	1.0	6.000000	0	0	0	0	0	
57580	102.0	0.862745	0	0	0	0	0	

5 rows × 103 columns

Model_1: Linear Regression

In [28]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train, y_train)
```

Out[28]:
```
LinearRegression()
```

In [29]:
```
preds = model.predict(x_test)
```

In [30]:
```
from sklearn.metrics import mean_squared_error, r2_score
mean_squared_error(y_test, preds)
```

Out[30]:
```
377805097713273.25
```

In [31]:
```
r2_score(y_test, preds)
```

Out[31]: 0.22435650490293835

Model-2: Decision Tree

In [32]:
```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 42)
regressor.fit(x_train, y_train)
```

Out[32]: DecisionTreeRegressor(random_state=42)

In [33]:
```
preds = regressor.predict(x_test)
mean_squared_error(y_test,preds)
```

Out[33]: 9901607626148.123

In [34]:
```
r2_score(y_test,preds)
```

Out[34]: 0.9796717471714638

In [ ]: