Exploratory Data Analysis on Crop Production in India

I will be analyzing the Agriculture Crop Production dataset and try to answer some interesting questions. I have downloaded the dataset from Kaggle datasets.The libraries for data analysis and visualization that I have used in this project are Numpy, Pandas, Matplotlib and Seaborn.

Data Preparation and cleaning

- Load the file using Pandas
- Look at some information about the data
- Fix any missing values

Required Library

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

executed in 8.32s, finished 17:30:52 2024-07-02

Load Dataset

In [2]:
```
df = pd.read_csv('C:\\Users\\KALI LINNUX\\Desktop\\Data Analysis\\crop
```

executed in 230ms, finished 17:30:55 2024-07-02

Top 5 Row

In [3]:
```
df.head()
```

executed in 19ms, finished 17:30:56 2024-07-02

Out[3]:

	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

Last 5 Row

In [4]:
```
df.tail()
```

executed in 14ms, finished 17:30:57 2024-07-02

Out[4]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
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

Random 5 Rows

In [5]:
```
1 df.sample(5)
```
executed in 24ms, finished 17:30:58 2024-07-02

Out[5]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
165252	Rajasthan	ALWAR	2009	Whole Year	Sweet potato	20.0	NaN
111733	Madhya Pradesh	JABALPUR	2002	Whole Year	Sannhamp	39.0	89.0
127359	Maharashtra	BHANDARA	1998	Kharif	Soyabean	11900.0	20000.0
72190	Himachal Pradesh	KANGRA	2001	Whole Year	Garlic	49.0	65.0
114961	Madhya Pradesh	MANDSAUR	2009	Whole Year	Garlic	6519.0	27568.0

Shape of dataset

In [6]:
```
1 df.shape
```
executed in 5ms, finished 17:30:59 2024-07-02

Out[6]:
```
(246091, 7)
```

In [2]:
```
1 print(f'The Given dataset contain Rows is : {246091}')
2 print(f'The Given dataset contain Columns is : {7}')
3
```
executed in 6ms, finished 06:54:49 2024-07-05

```
The Given dataset contain Rows is : 246091
The Given dataset contain Columns is : 7
```

Total size of dataset

In [7]:
```
1 df.size
```
executed in 5ms, finished 17:31:00 2024-07-02

Out[7]:
```
1722637
```

In [3]:
```
1 print(f'The given dataset size is : {1722637}')
```
executed in 7ms, finished 06:55:54 2024-07-05

```
The given dataset size is : 1722637
```

Basic Information about dataset

In [8]:
```
1 df.info()
```

executed in 50ms, finished 17:31:00 2024-07-02

```
<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
```

Statical information about dataset

In [9]:
```
1 df.describe()
```

executed in 41ms, finished 17:31:01 2024-07-02

Out[9]:

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

Checking for Missing Data

In [10]:
```
1 df.isna().sum()
```

executed in 34ms, finished 17:31:03 2024-07-02

Out[10]:

```
State_Name          0
District_Name       0
Crop_Year           0
Season              0
Crop                0
Area                0
Production       3730
dtype: int64
```

In [11]:
```
3730/ 246091
```
executed in 6ms, finished 17:31:06 2024-07-02

Out[11]:
```
0.015156994770227274
```

In [12]:
```
# Dropping the sample having missing data
df.dropna(subset=['Production'], axis=0, inplace=True)
```
executed in 24ms, finished 17:31:07 2024-07-02

In [13]:
```
df.describe()
```
executed in 36ms, finished 17:31:08 2024-07-02

Out[13]:

	Crop_Year	Area	Production
count	242361.000000	2.423610e+05	2.423610e+05
mean	2005.625773	1.216741e+04	5.825034e+05
std	4.958285	5.085744e+04	1.706581e+07
min	1997.000000	1.000000e-01	0.000000e+00
25%	2002.000000	8.700000e+01	8.800000e+01
50%	2006.000000	6.030000e+02	7.290000e+02
75%	2010.000000	4.545000e+03	7.023000e+03
max	2015.000000	8.580100e+06	1.250800e+09

In [14]:
```
df.shape
```
executed in 4ms, finished 17:31:09 2024-07-02

Out[14]:
```
(242361, 7)
```

Exploratory Analysis and Visualization

Univarate Analysis

STATE

In [15]:
```
# STATES
states = df.State_Name.str.strip().unique()
states
```
executed in 84ms, finished 17:31:12 2024-07-02

Out[15]:
```
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 [16]:
```
1 len(states)
```

executed in 5ms, finished 17:31:13 2024-07-02

Out[16]: 33

In [17]:
```
1 df.State_Name.value_counts()
```

executed in 19ms, finished 17:31:15 2024-07-02

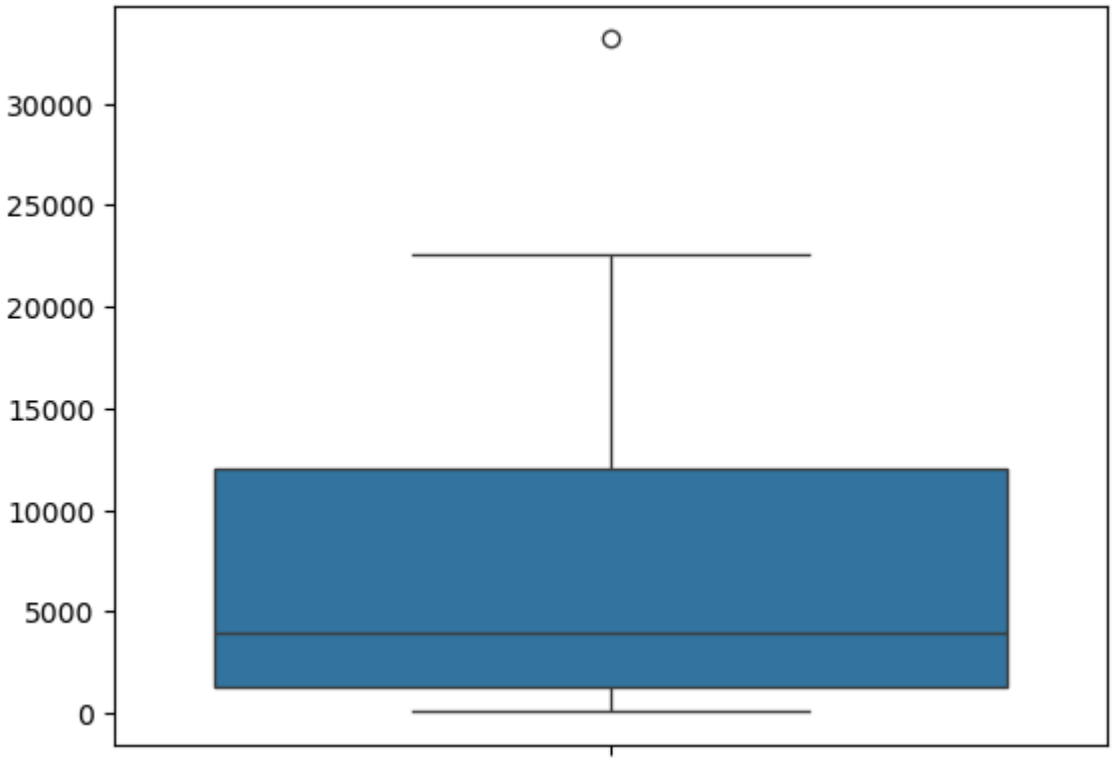
Out[17]:
```
State_Name
Uttar Pradesh                  33189
Madhya Pradesh                 22604
Karnataka                      21079
Bihar                          18874
Assam                          14622
Odisha                         13524
Tamil Nadu                     13266
Maharashtra                    12496
Rajasthan                      12066
Chhattisgarh                   10368
West Bengal                     9597
Andhra Pradesh                  9561
Gujarat                         8365
Telangana                       5591
Uttarakhand                     4825
Haryana                         4540
Kerala                          4003
Nagaland                        3904
Punjab                          3143
Meghalaya                       2867
Arunachal Pradesh               2545
Himachal Pradesh                2456
Jammu and Kashmir               1632
Tripura                         1412
Manipur                         1266
Jharkhand                       1266
Mizoram                          954
Puducherry                       872
Sikkim                           714
Dadra and Nagar Haveli           263
Goa                              207
Andaman and Nicobar Islands      201
Chandigarh                        89
Name: count, dtype: int64
```

- We have data from 33 states (including union territories).
- We have more data from top agriculture rich states like Uttar Pradesh, Madhya Pradesh and Karnataka than other states.

In [18]:
```
1 sns.boxplot(df.State_Name.value_counts().values)
```

executed in 340ms, finished 17:31:17 2024-07-02

Out[18]: <Axes: >



District

In [19]:

```
(len(df.District_Name.unique()),
df.District_Name.unique())
```

executed in 35ms, finished 17:31:19 2024-07-02

Out[19]: (646,
 array(['NICOBARS', 'NORTH AND MIDDLE ANDAMAN', 'SOUTH ANDAMANS',
       'ANANTAPUR', 'CHITTOOR', 'EAST GODAVARI', 'GUNTUR', 'KADAPA',
       'KRISHNA', 'KURNOOL', 'PRAKASAM', 'SPSR NELLORE', 'SRIKAKULAM',
       'VISAKHAPATANAM', 'VIZIANAGARAM', 'WEST GODAVARI', 'ANJAW',
       'CHANGLANG', 'DIBANG VALLEY', 'EAST KAMENG', 'EAST SIANG',
       'KURUNG KUMEY', 'LOHIT', 'LONGDING', 'LOWER DIBANG VALLEY',
       'LOWER SUBANSIRI', 'NAMSAI', 'PAPUM PARE', 'TAWANG', 'TIRAP',
       'UPPER SIANG', 'UPPER SUBANSIRI', 'WEST KAMENG', 'WEST SIANG',
       'BAKSA', 'BARPETA', 'BONGAIGAON', 'CACHAR', 'CHIRANG', 'DARRANG',
       'DHEMAJI', 'DHUBRI', 'DIBRUGARH', 'DIMA HASAO', 'GOALPARA',
       'GOLAGHAT', 'HAILAKANDI', 'JORHAT', 'KAMRUP', 'KAMRUP METRO',
       'KARBI ANGLONG', 'KARIMGANJ', 'KOKRAJHAR', 'LAKHIMPUR', 'MARIGAO
N',
       'NAGAON', 'NALBARI', 'SIVASAGAR', 'SONITPUR', 'TINSUKIA',
       'UDALGURI', 'ARARIA', 'ARWAL', 'AURANGABAD', 'BANKA', 'BEGUSARAI',
       'BHAGALPUR', 'BHOJPUR', 'BUXAR', 'DARBHANGA', 'GAYA', 'GOPALGANJ',
       'JAMUI', 'JEHANABAD', 'KAIMUR (BHABUA)', 'KATIHAR', 'KHAGARIA',
       'KISHANGANJ', 'LAKHISARAI', 'MADHEPURA', 'MADHUBANI', 'MUNGER',
       'MUZAFFARPUR', 'NALANDA', 'NAWADA', 'PASHCHIM CHAMPARAN', 'PATNA',
       'PURBI CHAMPARAN', 'PURNIA', 'ROHTAS', 'SAHARSA', 'SAMASTIPUR',
       'SARAN', 'SHEIKHPURA', 'SHEOHAR', 'SITAMARHI', 'SIWAN', 'SUPAUL',
       'VAISHALI', 'CHANDIGARH', 'BALOD', 'BALODA BAZAR', 'BALRAMPUR',
       'BASTAR', 'BEMETARA', 'BIJAPUR', 'BILASPUR', 'DANTEWADA',
       'DHAMTARI', 'DURG', 'GARIYABAND', 'JANJGIR-CHAMPA', 'JASHPUR',
       'KABIRDHAM', 'KANKER', 'KONDAGAON', 'KORBA', 'KOREA', 'MAHASAMUN
D',
       'MUNGELI', 'NARAYANPUR', 'RAIGARH', 'RAIPUR', 'RAJNANDGAON',
       'SUKMA', 'SURAJPUR', 'SURGUJA', 'DADRA AND NAGAR HAVELI',
       'NORTH GOA', 'SOUTH GOA', 'AHMADABAD', 'AMRELI', 'ANAND',
       'BANAS KANTHA', 'BHARUCH', 'BHAVNAGAR', 'DANG', 'DOHAD',
       'GANDHINAGAR', 'JAMNAGAR', 'JUNAGADH', 'KACHCHH', 'KHEDA',
       'MAHESANA', 'NARMADA', 'NAVSARI', 'PANCH MAHALS', 'PATAN',
       'PORBANDAR', 'RAJKOT', 'SABAR KANTHA', 'SURAT', 'SURENDRANAGAR',
       'TAPI', 'VADODARA', 'VALSAD', 'AMBALA', 'BHIWANI', 'FARIDABAD',
       'FATEHABAD', 'GURGAON', 'HISAR', 'JHAJJAR', 'JIND', 'KAITHAL',
       'KARNAL', 'KURUKSHETRA', 'MAHENDRAGARH', 'MEWAT', 'PALWAL',
       'PANCHKULA', 'PANIPAT', 'REWARI', 'ROHTAK', 'SIRSA', 'SONIPAT',
       'YAMUNANAGAR', 'CHAMBA', 'HAMIRPUR', 'KANGRA', 'KINNAUR', 'KULLU',
       'LAHUL AND SPITI', 'MANDI', 'SHIMLA', 'SIRMAUR', 'SOLAN', 'UNA',
       'ANANTNAG', 'BADGAM', 'BANDIPORA', 'BARAMULLA', 'DODA',
       'GANDERBAL', 'JAMMU', 'KARGIL', 'KATHUA', 'KISHTWAR', 'KULGAM',
       'KUPWARA', 'LEH LADAKH', 'POONCH', 'PULWAMA', 'RAJAURI', 'RAMBAN',
       'REASI', 'SAMBA', 'SHOPIAN', 'SRINAGAR', 'UDHAMPUR', 'BOKARO',
       'CHATRA', 'DEOGHAR', 'DHANBAD', 'DUMKA', 'EAST SINGHBUM', 'GARHW
A',
       'GIRIDIH', 'GODDA', 'GUMLA', 'HAZARIBAGH', 'JAMTARA', 'KHUNTI',
       'KODERMA', 'LATEHAR', 'LOHARDAGA', 'PAKUR', 'PALAMU', 'RAMGARH',
       'RANCHI', 'SAHEBGANJ', 'SARAIKELA KHARSAWAN', 'SIMDEGA',
       'WEST SINGHBHUM', 'BAGALKOT', 'BANGALORE RURAL', 'BELGAUM',
       'BELLARY', 'BENGALURU URBAN', 'BIDAR', 'CHAMARAJANAGAR',
       'CHIKBALLAPUR', 'CHIKMAGALUR', 'CHITRADURGA', 'DAKSHIN KANNAD',
       'DAVANGERE', 'DHARWAD', 'GADAG', 'GULBARGA', 'HASSAN', 'HAVERI',
       'KODAGU', 'KOLAR', 'KOPPAL', 'MANDYA', 'MYSORE', 'RAICHUR',
       'RAMANAGARA', 'SHIMOGA', 'TUMKUR', 'UDUPI', 'UTTAR KANNAD',
       'YADGIR', 'ALAPPUZHA', 'ERNAKULAM', 'IDUKKI', 'KANNUR',
       'KASARAGOD', 'KOLLAM', 'KOTTAYAM', 'KOZHIKODE', 'MALAPPURAM',
       'PALAKKAD', 'PATHANAMTHITTA', 'THIRUVANANTHAPURAM', 'THRISSUR',
       'WAYANAD', 'AGAR MALWA', 'ALIRAJPUR', 'ANUPPUR', 'ASHOKNAGAR',
       'BALAGHAT', 'BARWANI', 'BETUL', 'BHIND', 'BHOPAL', 'BURHANPUR',
       'CHHATARPUR', 'CHHINDWARA', 'DAMOH', 'DATIA', 'DEWAS', 'DHAR',

```
        'DINDORI', 'GUNA', 'GWALIOR', 'HARDA', 'HOSHANGABAD', 'INDORE',
        'JABALPUR', 'JHABUA', 'KATNI', 'KHANDWA', 'KHARGONE', 'MANDLA',
        'MANDSAUR', 'MORENA', 'NARSINGHPUR', 'NEEMUCH', 'PANNA', 'RAISEN',
        'RAJGARH', 'RATLAM', 'REWA', 'SAGAR', 'SATNA', 'SEHORE', 'SEONI',
        'SHAHDOL', 'SHAJAPUR', 'SHEOPUR', 'SHIVPURI', 'SIDHI', 'SINGRAUL
I',
        'TIKAMGARH', 'UJJAIN', 'UMARIA', 'VIDISHA', 'AHMEDNAGAR', 'AKOLA',
        'AMRAVATI', 'BEED', 'BHANDARA', 'BULDHANA', 'CHANDRAPUR', 'DHULE',
        'GADCHIROLI', 'GONDIA', 'HINGOLI', 'JALGAON', 'JALNA', 'KOLHAPUR',
        'LATUR', 'MUMBAI', 'NAGPUR', 'NANDED', 'NANDURBAR', 'NASHIK',
        'OSMANABAD', 'PALGHAR', 'PARBHANI', 'PUNE', 'RAIGAD', 'RATNAGIRI',
        'SANGLI', 'SATARA', 'SINDHUDURG', 'SOLAPUR', 'THANE', 'WARDHA',
        'WASHIM', 'YAVATMAL', 'BISHNUPUR', 'CHANDEL', 'CHURACHANDPUR',
        'IMPHAL EAST', 'IMPHAL WEST', 'SENAPATI', 'TAMENGLONG', 'THOUBAL',
        'UKHRUL', 'EAST GARO HILLS', 'EAST JAINTIA HILLS',
        'EAST KHASI HILLS', 'NORTH GARO HILLS', 'RI BHOI',
        'SOUTH GARO HILLS', 'SOUTH WEST GARO HILLS',
        'SOUTH WEST KHASI HILLS', 'WEST GARO HILLS', 'WEST JAINTIA HILLS',
        'WEST KHASI HILLS', 'AIZAWL', 'CHAMPHAI', 'KOLASIB', 'LAWNGTLAI',
        'LUNGLEI', 'MAMIT', 'SAIHA', 'SERCHHIP', 'DIMAPUR', 'KIPHIRE',
        'KOHIMA', 'LONGLENG', 'MOKOKCHUNG', 'MON', 'PEREN', 'PHEK',
        'TUENSANG', 'WOKHA', 'ZUNHEBOTO', 'ANUGUL', 'BALANGIR',
        'BALESHWAR', 'BARGARH', 'BHADRAK', 'BOUDH', 'CUTTACK', 'DEOGARH',
        'DHENKANAL', 'GAJAPATI', 'GANJAM', 'JAGATSINGHAPUR', 'JAJAPUR',
        'JHARSUGUDA', 'KALAHANDI', 'KANDHAMAL', 'KENDRAPARA', 'KENDUJHAR',
        'KHORDHA', 'KORAPUT', 'MALKANGIRI', 'MAYURBHANJ', 'NABARANGPUR',
        'NAYAGARH', 'NUAPADA', 'PURI', 'RAYAGADA', 'SAMBALPUR', 'SONEPUR',
        'SUNDARGARH', 'KARAIKAL', 'MAHE', 'PONDICHERRY', 'YANAM',
        'AMRITSAR', 'BARNALA', 'BATHINDA', 'FARIDKOT', 'FATEHGARH SAHIB',
        'FAZILKA', 'FIROZEPUR', 'GURDASPUR', 'HOSHIARPUR', 'JALANDHAR',
        'KAPURTHALA', 'LUDHIANA', 'MANSA', 'MOGA', 'MUKTSAR', 'NAWANSHAH
R',
        'PATHANKOT', 'PATIALA', 'RUPNAGAR', 'S.A.S NAGAR', 'SANGRUR',
        'TARN TARAN', 'AJMER', 'ALWAR', 'BANSWARA', 'BARAN', 'BARMER',
        'BHARATPUR', 'BHILWARA', 'BIKANER', 'BUNDI', 'CHITTORGARH',
        'CHURU', 'DAUSA', 'DHOLPUR', 'DUNGARPUR', 'GANGANAGAR',
        'HANUMANGARH', 'JAIPUR', 'JAISALMER', 'JALORE', 'JHALAWAR',
        'JHUNJHUNU', 'JODHPUR', 'KARAULI', 'KOTA', 'NAGAUR', 'PALI',
        'PRATAPGARH', 'RAJSAMAND', 'SAWAI MADHOPUR', 'SIKAR', 'SIROHI',
        'TONK', 'UDAIPUR', 'EAST DISTRICT', 'NORTH DISTRICT',
        'SOUTH DISTRICT', 'WEST DISTRICT', 'ARIYALUR', 'COIMBATORE',
        'CUDDALORE', 'DHARMAPURI', 'DINDIGUL', 'ERODE', 'KANCHIPURAM',
        'KANNIYAKUMARI', 'KARUR', 'KRISHNAGIRI', 'MADURAI', 'NAGAPATTINA
M',
        'NAMAKKAL', 'PERAMBALUR', 'PUDUKKOTTAI', 'RAMANATHAPURAM', 'SALE
M',
        'SIVAGANGA', 'THANJAVUR', 'THE NILGIRIS', 'THENI', 'THIRUVALLUR',
        'THIRUVARUR', 'TIRUCHIRAPPALLI', 'TIRUNELVELI', 'TIRUPPUR',
        'TIRUVANNAMALAI', 'TUTICORIN', 'VELLORE', 'VILLUPURAM',
        'VIRUDHUNAGAR', 'ADILABAD', 'HYDERABAD', 'KARIMNAGAR', 'KHAMMAM',
        'MAHBUBNAGAR', 'MEDAK', 'NALGONDA', 'NIZAMABAD', 'RANGAREDDI',
        'WARANGAL', 'DHALAI', 'GOMATI', 'KHOWAI', 'NORTH TRIPURA',
        'SEPAHIJALA', 'SOUTH TRIPURA', 'UNAKOTI', 'WEST TRIPURA', 'AGRA',
        'ALIGARH', 'ALLAHABAD', 'AMBEDKAR NAGAR', 'AMETHI', 'AMROHA',
        'AURAIYA', 'AZAMGARH', 'BAGHPAT', 'BAHRAICH', 'BALLIA', 'BANDA',
        'BARABANKI', 'BAREILLY', 'BASTI', 'BIJNOR', 'BUDAUN',
        'BULANDSHAHR', 'CHANDAULI', 'CHITRAKOOT', 'DEORIA', 'ETAH',
        'ETAWAH', 'FAIZABAD', 'FARRUKHABAD', 'FATEHPUR', 'FIROZABAD',
        'GAUTAM BUDDHA NAGAR', 'GHAZIABAD', 'GHAZIPUR', 'GONDA',
        'GORAKHPUR', 'HAPUR', 'HARDOI', 'HATHRAS', 'JALAUN', 'JAUNPUR',
        'JHANSI', 'KANNAUJ', 'KANPUR DEHAT', 'KANPUR NAGAR', 'KASGANJ',
```

```
        'KAUSHAMBI', 'KHERI', 'KUSHI NAGAR', 'LALITPUR', 'LUCKNOW',
        'MAHARAJGANJ', 'MAHOBA', 'MAINPURI', 'MATHURA', 'MAU', 'MEERUT',
        'MIRZAPUR', 'MORADABAD', 'MUZAFFARNAGAR', 'PILIBHIT', 'RAE BAREL
I',
        'RAMPUR', 'SAHARANPUR', 'SAMBHAL', 'SANT KABEER NAGAR',
        'SANT RAVIDAS NAGAR', 'SHAHJAHANPUR', 'SHAMLI', 'SHRAVASTI',
        'SIDDHARTH NAGAR', 'SITAPUR', 'SONBHADRA', 'SULTANPUR', 'UNNAO',
        'VARANASI', 'ALMORA', 'BAGESHWAR', 'CHAMOLI', 'CHAMPAWAT',
        'DEHRADUN', 'HARIDWAR', 'NAINITAL', 'PAURI GARHWAL', 'PITHORAGAR
H',
        'RUDRA PRAYAG', 'TEHRI GARHWAL', 'UDAM SINGH NAGAR', 'UTTAR KASH
I',
        '24 PARAGANAS NORTH', '24 PARAGANAS SOUTH', 'BANKURA', 'BARDHAMA
N',
        'BIRBHUM', 'COOCHBEHAR', 'DARJEELING', 'DINAJPUR DAKSHIN',
        'DINAJPUR UTTAR', 'HOOGHLY', 'HOWRAH', 'JALPAIGURI', 'MALDAH',
        'MEDINIPUR EAST', 'MEDINIPUR WEST', 'MURSHIDABAD', 'NADIA',
        'PURULIA'], dtype=object))
```

In [20]:

```
1 df.District_Name.value_counts()
```

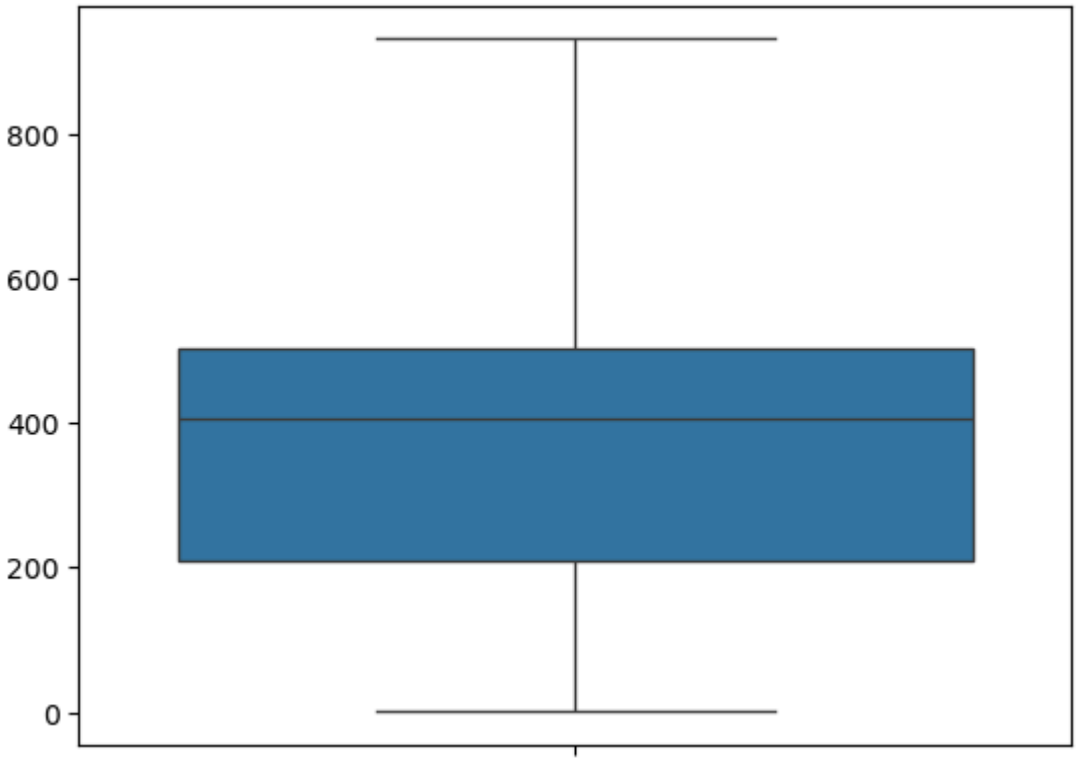
executed in 18ms, finished 17:31:22 2024-07-02

Out[20]:

```
District_Name
TUMKUR       931
BELGAUM      924
BIJAPUR      905
HASSAN       895
BELLARY      887
             ...
HYDERABAD      8
KHUNTI         6
RAMGARH        6
NAMSAI         1
MUMBAI         1
Name: count, Length: 646, dtype: int64
```

In [21]:
```
1 sns.boxplot(df.District_Name.value_counts().values)
```
executed in 248ms, finished 17:31:23 2024-07-02

Out[21]: <Axes: >



Crop Year

In [22]:
```
1 # CROP YEARS
2
3 print(df.Crop_Year.nunique())
4 print(df.Crop_Year.min())
5 print(df.Crop_Year.max())
```
executed in 9ms, finished 17:31:28 2024-07-02

```
19
1997
2015
```

In [23]:
```
1 df.Crop_Year.unique()
```
executed in 7ms, finished 17:31:29 2024-07-02

Out[23]:
```
array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2010, 1997, 1998, 1999,
       2007, 2008, 2009, 2011, 2012, 2013, 2014, 2015], dtype=int64)
```

In [24]:
```
1 df.Crop_Year.value_counts()
```
executed in 10ms, finished 17:31:30 2024-07-02

Out[24]:
```
Crop_Year
2003    17139
2002    16536
2007    14269
2008    14230
2006    13976
2004    13858
2010    13793
2011    13791
2009    13767
2000    13553
2005    13519
2013    13475
2001    13293
2012    13184
1999    12441
1998    11262
2014    10815
1997     8899
2015      561
Name: count, dtype: int64
```

- We have data of 19 years from 1997 to 2015.
- The years having more data are - 2003, 2002, 2007, 2008 and 2006

Season

In [25]:
```
1 # Season
2 print(df.Season.nunique())
3 df.Season = df.Season.str.strip()
```
executed in 76ms, finished 17:31:32 2024-07-02

```
6
```

In [26]:
```
1 Seasons = df.Season.unique()
2 Seasons
```
executed in 17ms, finished 17:31:33 2024-07-02

Out[26]:
```
array(['Kharif', 'Whole Year', 'Autumn', 'Rabi', 'Summer', 'Winter'],
      dtype=object)
```

In [27]:
```
1 df.Season.max()
```
executed in 20ms, finished 17:31:34 2024-07-02

Out[27]:
```
'Winter'
```

In [28]:
```
1 df.Season.value_counts()
```
executed in 32ms, finished 17:31:35 2024-07-02

Out[28]:
```
Season
Kharif        94283
Rabi          66160
Whole Year    56127
Summer        14811
Winter         6050
Autumn         4930
Name: count, dtype: int64
```

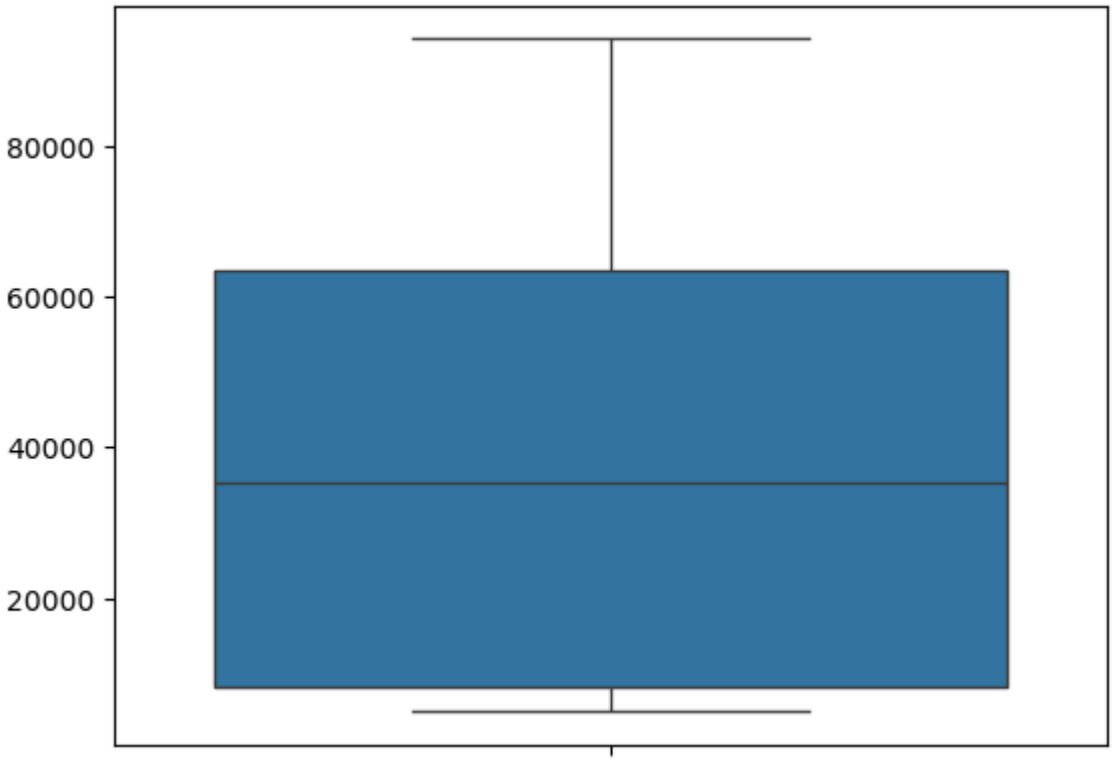
- Dataset talks of six different seasons - Kharif, Annual, Autumn, Rabi, Summer and Winter.

- More crops yielding in Winter.

- Frequency wise, we have more data points from Kharif, Rabi and Annual crop types.

In [29]:
```
1 sns.boxplot(df.Season.value_counts().values)
```
executed in 134ms, finished 17:31:37 2024-07-02

Out[29]:
```
<Axes: >
```



Crop

In [30]:
```
1 print(df.Crop.nunique())
```
executed in 20ms, finished 17:31:39 2024-07-02

```
124
```

In [31]:

```
Crop = df.Crop.unique()
Crop
```

executed in 20ms, finished 17:31:41 2024-07-02

Out[31]:

```
array(['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', 'Mesta', 'Cowpea(Lobia)',
       'Lemon', 'Pome Granet', 'Sapota', 'Cabbage', 'Rapeseed &Mustard',
       'Peas  (vegetable)', 'Niger seed', 'Bottle Gourd', 'Varagu',
       'Garlic', 'Ginger', 'Oilseeds total', 'Pulses total', 'Jute',
       'Peas & beans (Pulses)', 'Blackgram', 'Paddy', 'Pineapple',
       'Barley', 'Sannhamp', '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', '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',
       'Pump Kin', 'Apple', 'Peach', 'Pear', 'Plums', 'Litchi', 'Ber',
       'Other Dry Fruit', 'Jute & mesta'], dtype=object)
```

In [32]:
```
1 df.Crop.value_counts().head(25)
```

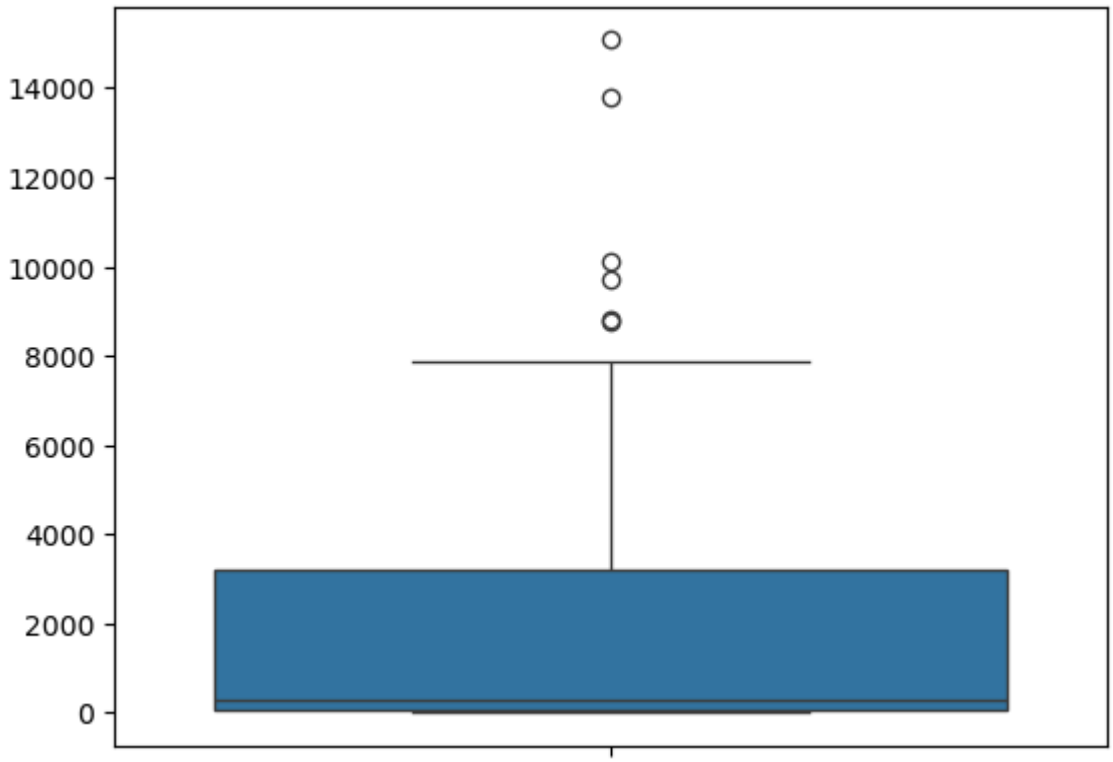
executed in 21ms, finished 17:31:42 2024-07-02

Out[32]:
```
Crop
Rice                    15082
Maize                   13787
Moong(Green Gram)       10106
Urad                     9710
Sesamum                  8821
Groundnut                8770
Wheat                    7878
Sugarcane                7827
Rapeseed &Mustard        7533
Arhar/Tur                7476
Gram                     7227
Jowar                    6990
Onion                    6984
Potato                   6914
Dry chillies             6421
Sunflower                5483
Bajra                    5379
Small millets            4593
Peas & beans (Pulses)    4447
Cotton(lint)             4382
Linseed                  4351
Turmeric                 4168
Masoor                   4152
Sweet potato             4122
Barley                   4116
Name: count, dtype: int64
```

In [33]:
```
1 sns.boxplot(df.Crop.value_counts().values)
```

executed in 157ms, finished 17:31:44 2024-07-02

Out[33]:
```
<Axes: >
```



Area

In [34]:
```
# Area
print(df.Area.max())
print(df.Area.min())
```

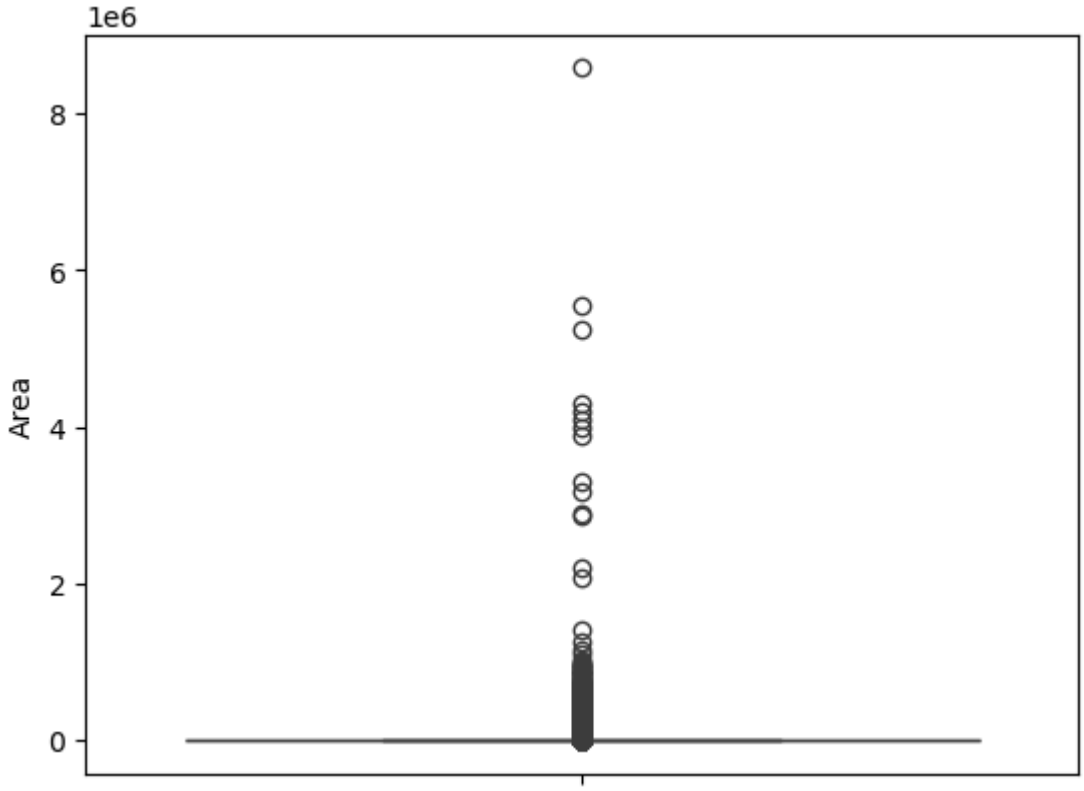
executed in 8ms, finished 17:31:46 2024-07-02

```
8580100.0
0.1
```

In [35]:
```
sns.boxplot(df.Area)
```

executed in 685ms, finished 17:31:47 2024-07-02

Out[35]: <Axes: ylabel='Area'>



Production

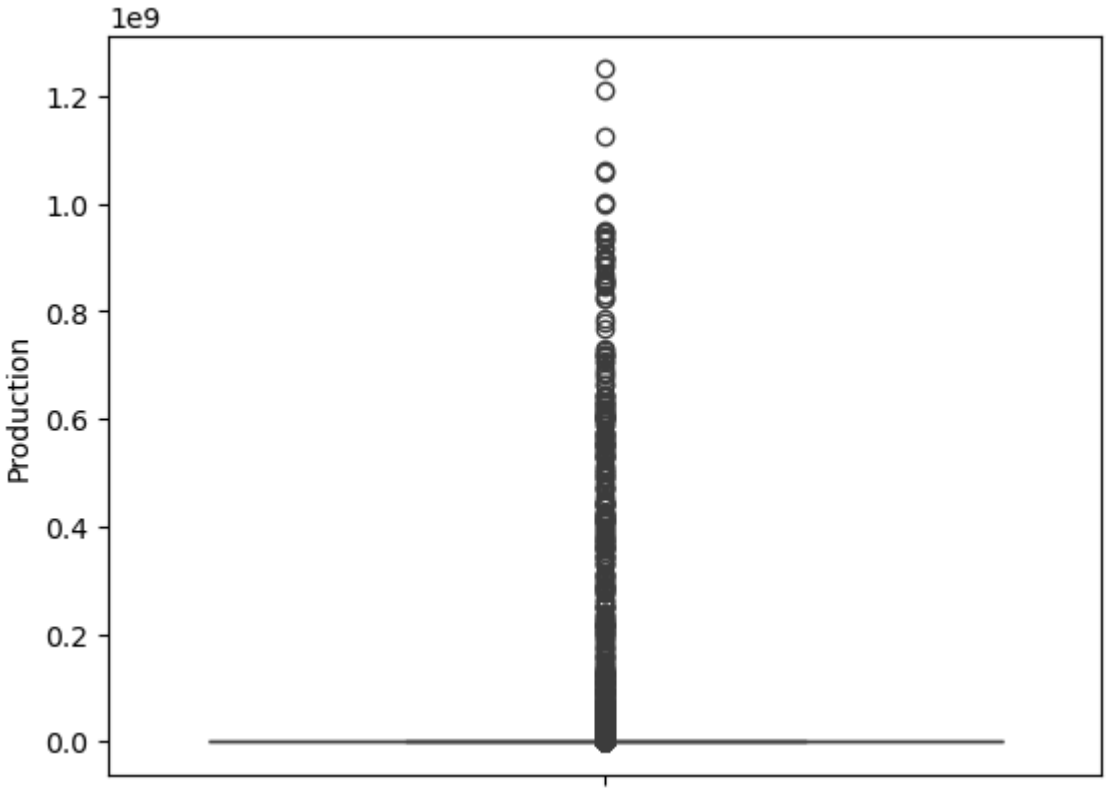
In [36]:
```
df.Production.describe()
```

executed in 18ms, finished 17:31:48 2024-07-02

Out[36]:
```
count    2.423610e+05
mean     5.825034e+05
std      1.706581e+07
min      0.000000e+00
25%      8.800000e+01
50%      7.290000e+02
75%      7.023000e+03
max      1.250800e+09
Name: Production, dtype: float64
```

In [37]:
```
1 sns.boxplot(df.Production)
```
executed in 712ms, finished 17:31:50 2024-07-02

Out[37]: <Axes: ylabel='Production'>



Create two columns Total Production and Productivity.

Total Production

In [75]:
```
1 df['Total Production'] = df['Production'] * df['Area']
```
executed in 6ms, finished 17:35:31 2024-07-02

In [76]:
```
1 df['Total Production']
```
executed in 8ms, finished 17:35:48 2024-07-02

Out[76]:
```
0         2.508000e+06
1         2.000000e+00
2         3.274200e+04
3         1.128160e+05
4         1.188000e+05
              ...
246086    2.451060e+05
246087    2.903010e+05
246088    5.265000e+06
246089    1.669041e+11
246090    1.540000e+04
Name: Total Production, Length: 242361, dtype: float64
```

Remove Outliers

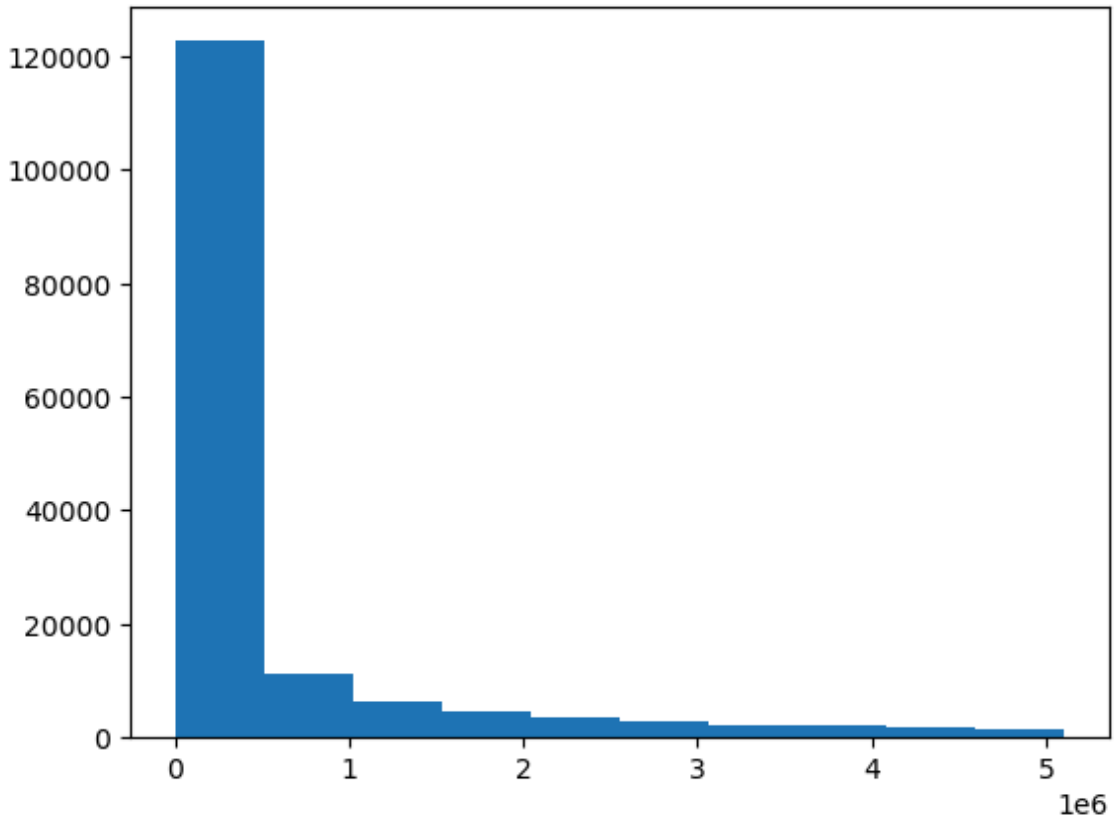
In [93]:
```
Q1= df["Total Production"].quantile(0.40)
Q3 =df["Total Production"].quantile (0.60)
IQR = Q3 - Q1
df =df[(df["Total Production"] >= Q1-1.5*IQR) & (df ["Total Production'
```

executed in 39ms, finished 17:53:07 2024-07-02

In [95]:
```
plt.hist(df['Total Production'])
```

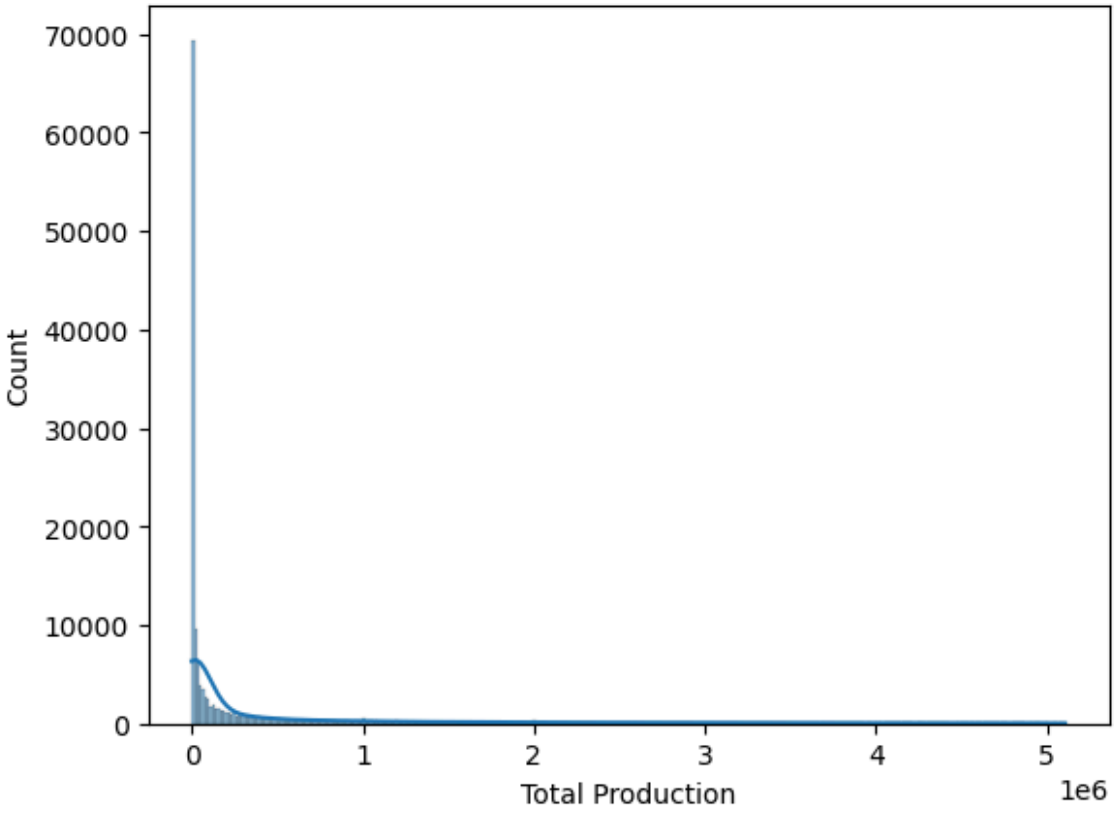
executed in 156ms, finished 17:53:38 2024-07-02

Out[95]:
```
(array([122699.,  11178.,   6477.,   4636.,   3380.,   2853.,   2178.,
         2007.,   1683.,   1531.]),
 array([      0. ,  510571.6, 1021143.2, 1531714.8, 2042286.4, 2552858. ,
        3063429.6, 3574001.2, 4084572.8, 4595144.4, 5105716. ]),
 <BarContainer object of 10 artists>)
```



In [96]:
```
1 sns.histplot(df['Total Production'], kde = True)
```
executed in 1.81s, finished 17:53:49 2024-07-02

Out[96]:
```
<Axes: xlabel='Total Production', ylabel='Count'>
```



Productivity

In [99]:
```
1 df['Productivity'] = df['Production'] / df['Area']
```
executed in 6ms, finished 17:54:22 2024-07-02

In [100]:
```
1 df['Productivity']
```
executed in 7ms, finished 17:54:30 2024-07-02

Out[100]:
```
1          0.500000
4          0.229167
11         0.500000
13         0.267038
16         1.000000
             ...
246081     0.800000
246082     0.685185
246083     0.513636
246087     0.738437
246090     0.502857
Name: Productivity, Length: 103594, dtype: float64
```

In [101]:
```
Q1 =df ["Productivity"].quantile (0.40)
Q3 =df["Productivity"].quantile (0.60)

IQR= Q3-Q1

df=df[(df["Productivity"] >= Q1 -1.5*IQR) & (df ["Productivity"] <= Q3
```

executed in 20ms, finished 17:54:32 2024-07-02

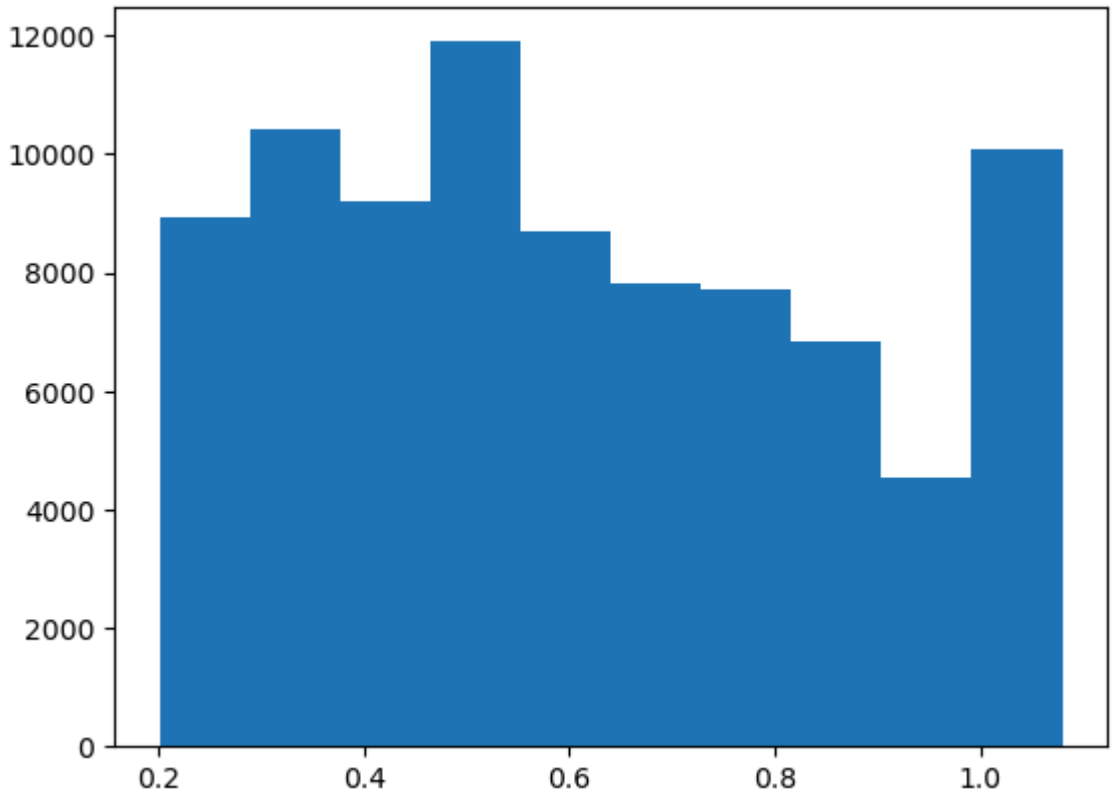
In [102]:
```
df['Productivity']
```

executed in 9ms, finished 17:54:34 2024-07-02

Out[102]:
```
1         0.500000
4         0.229167
11        0.500000
13        0.267038
16        1.000000
             ...
246081    0.800000
246082    0.685185
246083    0.513636
246087    0.738437
246090    0.502857
Name: Productivity, Length: 86161, dtype: float64
```

In [103]:
```
plt.hist(df['Productivity'])
plt.show()
```

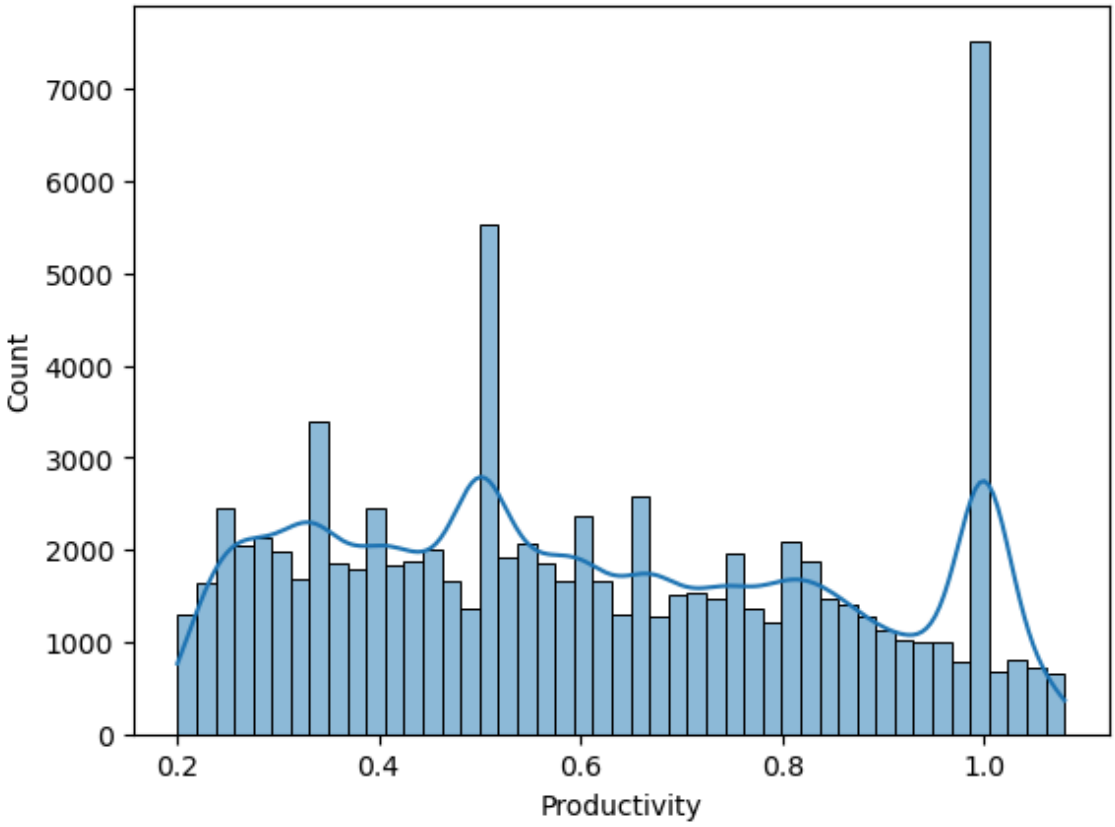
executed in 132ms, finished 17:54:38 2024-07-02



In [104]:
```
1 sns.histplot(df['Productivity'], kde = True)
```

executed in 583ms, finished 17:54:49 2024-07-02

Out[104]: <Axes: xlabel='Productivity', ylabel='Count'>



Area

In [105]:
```
1 df.Area
```

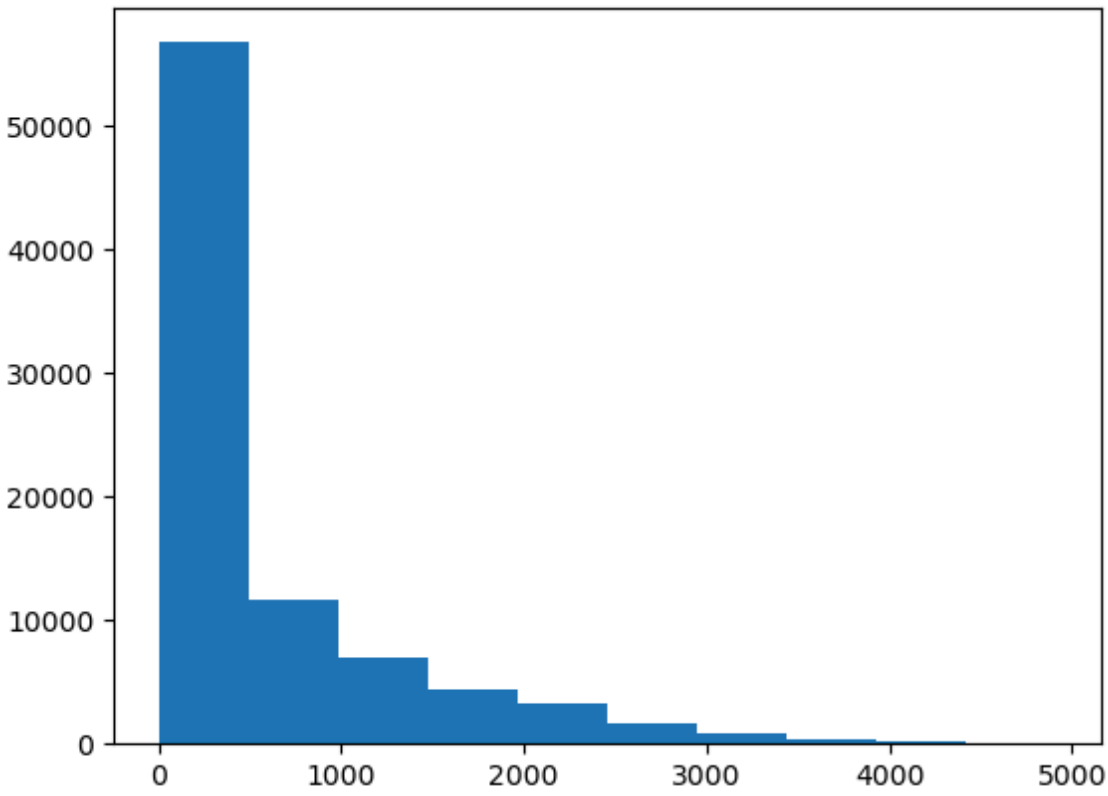
executed in 8ms, finished 17:55:10 2024-07-02

Out[105]:
```
1           2.0
4         720.0
11          2.0
13        719.0
16          1.0
             ...
246081    1885.0
246082      54.0
246083     220.0
246087     627.0
246090     175.0
Name: Area, Length: 86161, dtype: float64
```

In [106]:
```
1 plt.hist(df['Area'])
```

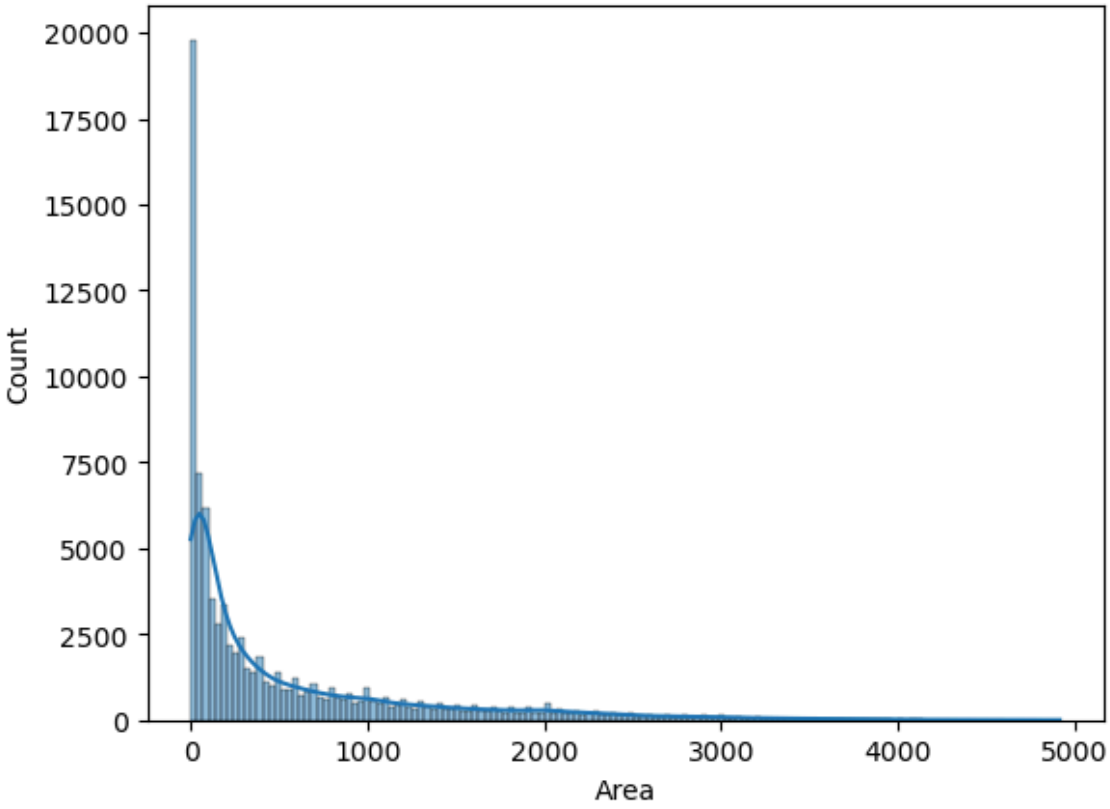
executed in 146ms, finished 17:55:14 2024-07-02

Out[106]:
```
(array([5.6739e+04, 1.1625e+04, 6.9610e+03, 4.3580e+03, 3.3170e+03,
        1.6310e+03, 9.2200e+02, 4.0800e+02, 1.5400e+02, 4.6000e+01]),
 array([1.00000e-01, 4.91490e+02, 9.82880e+02, 1.47427e+03, 1.96566e+03,
        2.45705e+03, 2.94844e+03, 3.43983e+03, 3.93122e+03, 4.42261e+03,
        4.91400e+03]),
 <BarContainer object of 10 artists>)
```



In [107]:
```
1 sns.histplot(df['Area'], kde = True)
```
executed in 876ms, finished 17:55:21 2024-07-02

Out[107]: <Axes: xlabel='Area', ylabel='Count'>



Production

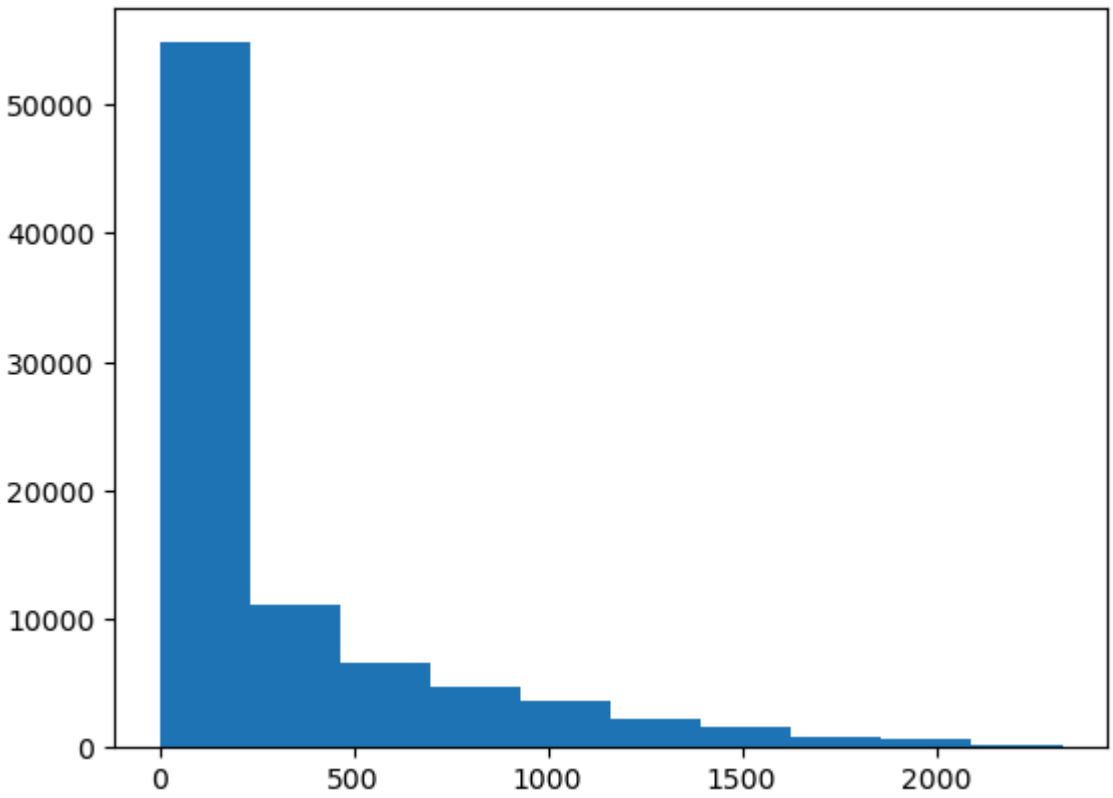
In [119]:
```
1 df.Production
```
executed in 6ms, finished 18:00:20 2024-07-02

Out[119]:
```
1           1.0
4         165.0
11          1.0
13        192.0
16          1.0
            ...
246081    1508.0
246082      37.0
246083     113.0
246087     463.0
246090      88.0
Name: Production, Length: 86161, dtype: float64
```

In [120]:
```
1 plt.hist(df['Production'])
```

executed in 147ms, finished 18:00:33 2024-07-02

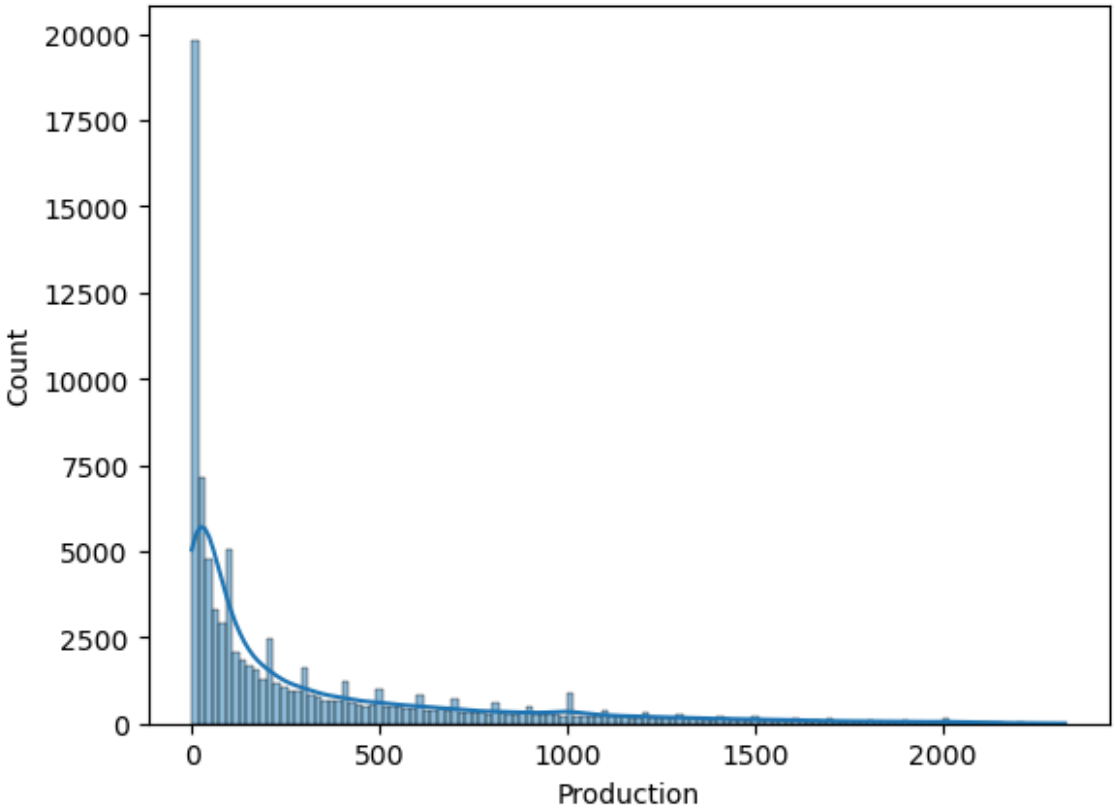
Out[120]:
```
(array([54829., 11050.,  6568.,  4659.,  3628.,  2204.,  1597.,   863.,
         602.,   161.]),
 array([8.000000e-02, 2.325720e+02, 4.650640e+02, 6.975560e+02,
        9.300480e+02, 1.162540e+03, 1.395032e+03, 1.627524e+03,
        1.860016e+03, 2.092508e+03, 2.325000e+03]),
 <BarContainer object of 10 artists>)
```



In [121]:
```
1 sns.histplot(df['Production'], kde=True)
```

executed in 836ms, finished 18:00:59 2024-07-02

Out[121]: <Axes: xlabel='Production', ylabel='Count'>

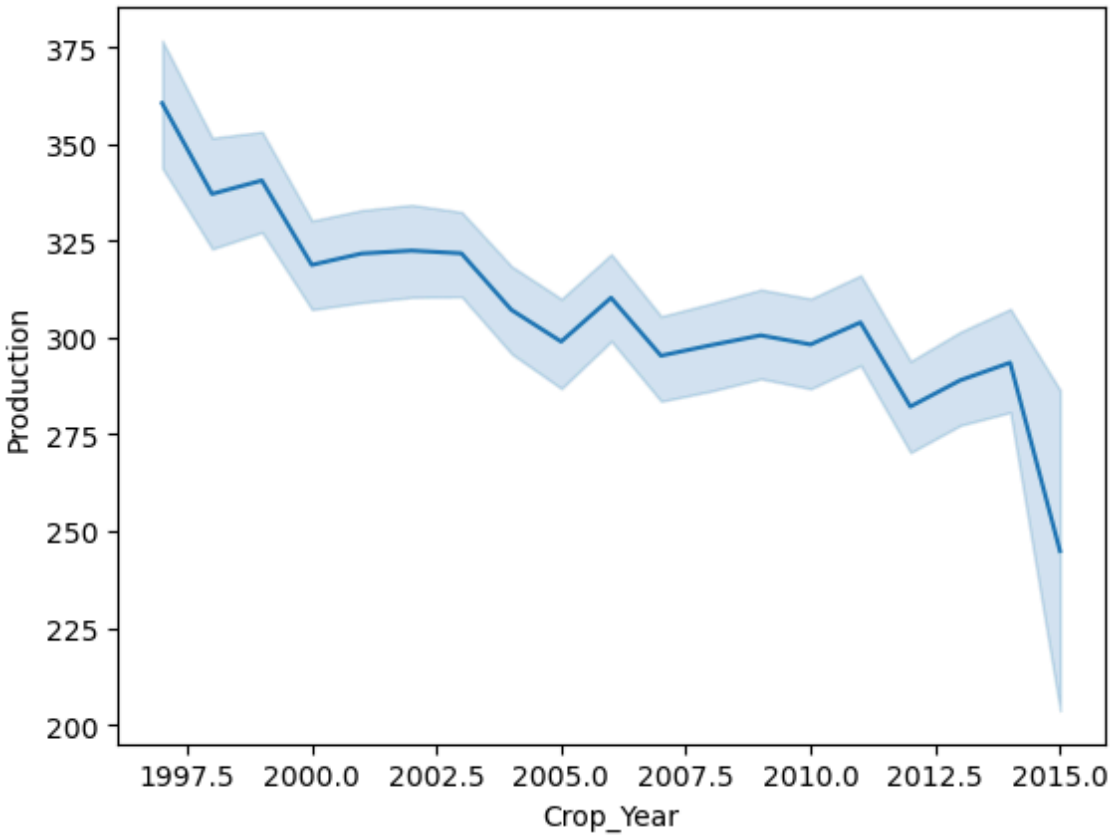


Plot crop production over the years.

In [108]:
```
1 sns.lineplot(x='Crop_Year', y='Production', data = df)
2 plt.show()
```

executed in 1.44s, finished 17:55:28 2024-07-02

Out[108]: <Axes: xlabel='Crop_Year', ylabel='Production'>

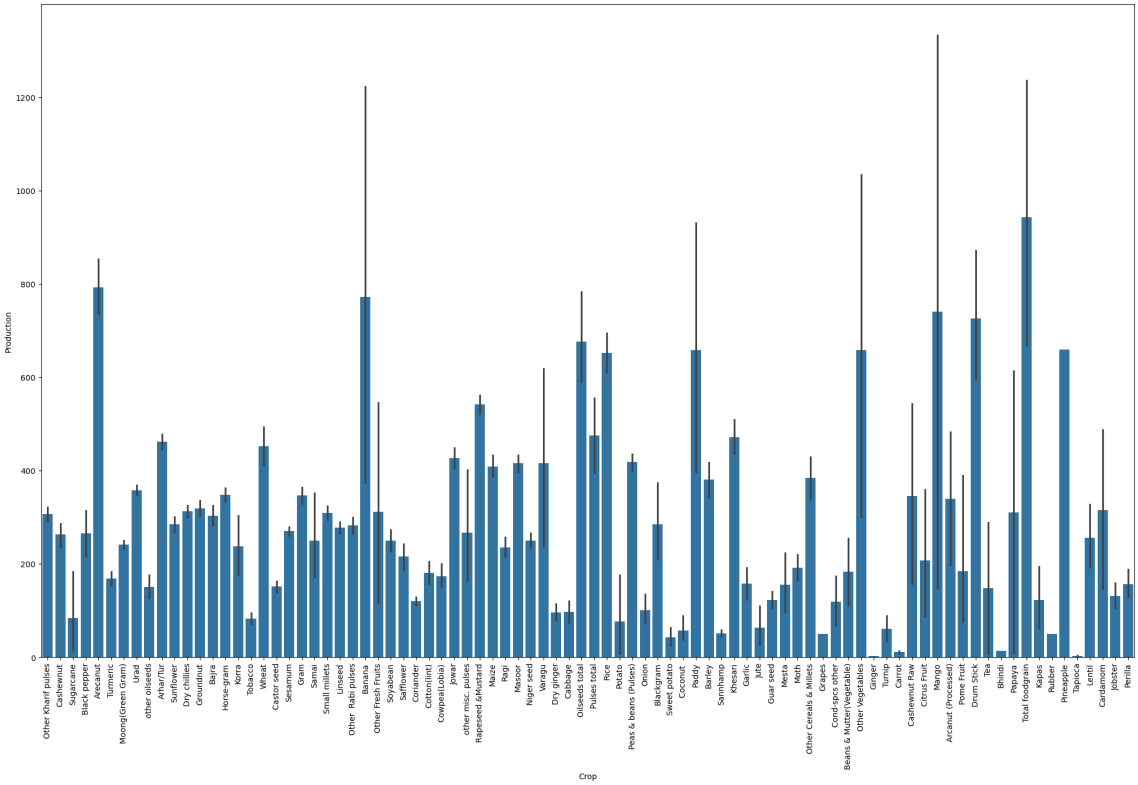


Plot crop production by crop

In [117]:

```
plt.figure(figsize=(25,15))
sns.barplot(x='Crop', y='Production', data = df)
plt.xticks(rotation = 90)
plt.show()
```

executed in 3.04s, finished 17:57:28 2024-07-02

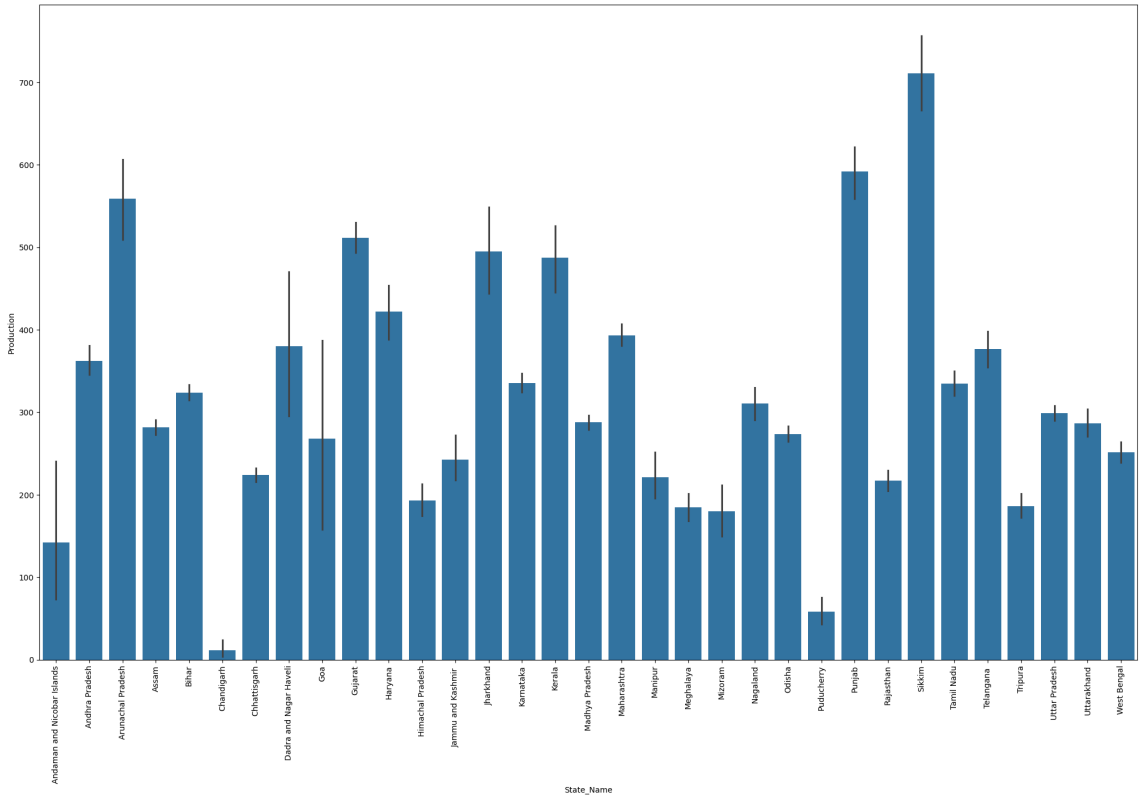


Plot crop production by state

In [116]:

```
1 plt.figure(figsize=(25,15))
2 sns.barplot(x='State_Name', y='Production', data = df)
3 plt.xticks(rotation = 90)
4 plt.show()
```

executed in 1.81s, finished 17:57:11 2024-07-02



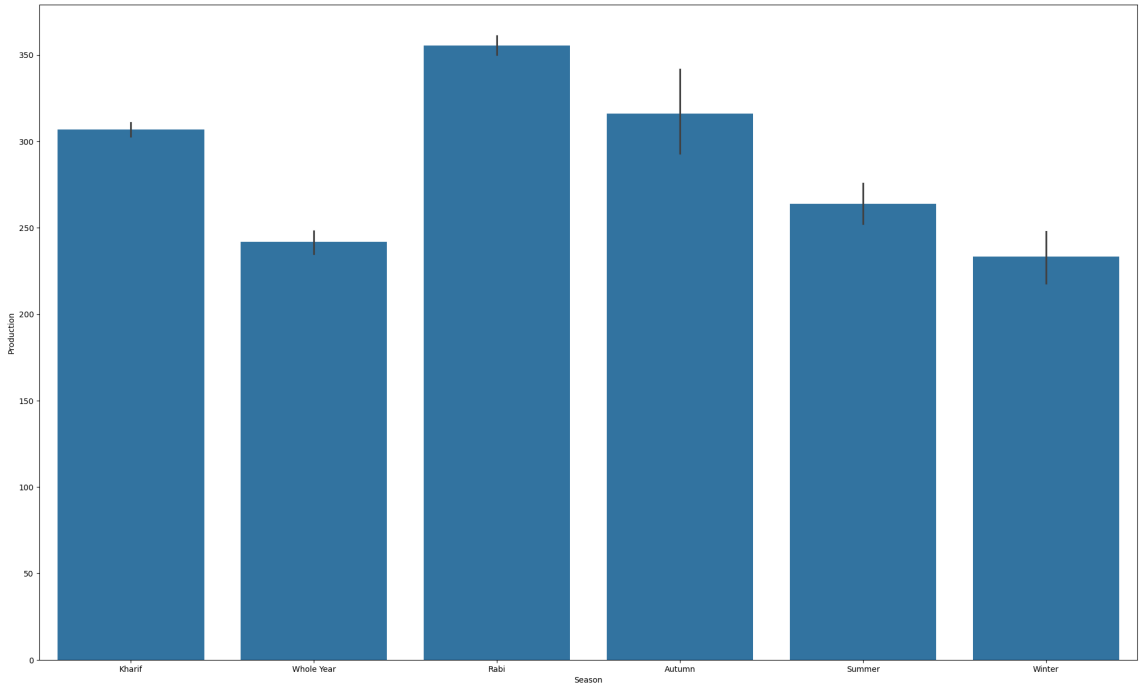
Plot crop production by season

In [118]:

```
plt.figure(figsize=(25,15))
sns.barplot(x='Season', y='Production', data = df)
plt.show()

```

executed in 1.21s, finished 17:59:51 2024-07-02



In [38]:

```
crop = df['Crop']
```

executed in 3ms, finished 17:32:08 2024-07-02

In [39]:

```
def crop_category(crop):
    for i in ['Rice','Maize','Wheat','Barley','Varagu','Other Cereals &
        if crop==i:
            return 'Cereal'
    for i in ['Moong','Urad','Arhar/Tur','Peas & beans','Masoor',
              'Other Kharif pulses','other misc. pulses','Ricebean (nag
              'Rajmash Kholar','Lentil','Samai','Blackgram','Korra','Co
              'Other  Rabi pulses','Other Kharif pulses','Peas & beans
        if crop==i:
            return 'Pulses'
    for i in ['Peach','Apple','Litchi','Pear','Plums','Ber','Sapota','L
              'Other Citrus Fruit','Water Melon','Jack Fruit','Grapes'
              'Pome Fruit','Citrus Fruit','Other Fresh Fruits','Mango'
        if crop==i:
            return 'Fruits'
    for i in ['Bean','Lab-Lab','Moth','Guar seed','Soyabean','Horse-gra
        if crop==i:
            return 'Beans'
    for i in ['Turnip','Peas','Beet Root','Carrot','Yam','Ribed Guard',
              'Bitter Gourd','Cucumber','Drum Stick','Cauliflower','Bea
              'Bhindi','Tomato','Brinjal','Khesari','Sweet potato','Pot
              if crop==i:
                return 'Vegetables'
    for i in ['Perilla','Ginger','Cardamom','Black pepper','Dry ginger'
        if crop==i:
            return 'spices'
    for i in ['other fibres','Kapas','Jute & mesta','Jute','Mesta','Cot
        if crop==i:
            return 'fibres'
    for i in ['Arcanut (Processed)','Atcanut (Raw)','Cashewnut Processe
        if crop==i:
            return 'Nuts'
    for i in ['other oilseeds','Safflower','Niger seed','Castor seed',
        if crop==i:
            return 'Oilseeds'
    for i in ['Tobacco','Coffee','Tea','Sugarcane','Rubber']:
        if crop==i:
            return 'Commercial'

df['crop_category']=df['Crop'].apply(crop_category)
```

executed in 452ms, finished 17:32:09 2024-07-02

In [40]:

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

executed in 18ms, finished 17:32:10 2024-07-02

Out[40]:

```
crop_category
Cereal        63283
Pulses        40898
Oilseeds      33801
Vegetables    23154
spices        21638
Nuts          11472
Commercial    10561
fibres         9785
Beans          9115
Fruits         6153
Name: count, dtype: int64
```

- A new variable 'crop_category' is created.

- Cereals, Pulses and Oilseeds are top producing categories.

In [41]:
```
# NEW VARIABLE - PRODUCTION PER UNIT AREA
df['Production_per_unit_area'] = df['Production'] / df['Area']
```

executed in 6ms, finished 17:32:11 2024-07-02

In [42]:
```
df.Production_per_unit_area
```

executed in 10ms, finished 17:32:12 2024-07-02

Out[42]:
```
0          1.594896
1          0.500000
2          3.147059
3          3.642045
4          0.229167
              ...
246086     2.617647
246087     0.738437
246088    50.154321
246089     2.141848
246090     0.502857
Name: Production_per_unit_area, Length: 242361, dtype: float64
```

- A new new variable 'prod_per_unit_area' for Production per unit area is created.

Visualising Data

1. State wise Production

In [43]:
```
prod = df.groupby(by = df.State_Name)['Production'].sum().reset_index()
prod
```

executed in 28ms, finished 17:32:15 2024-07-02

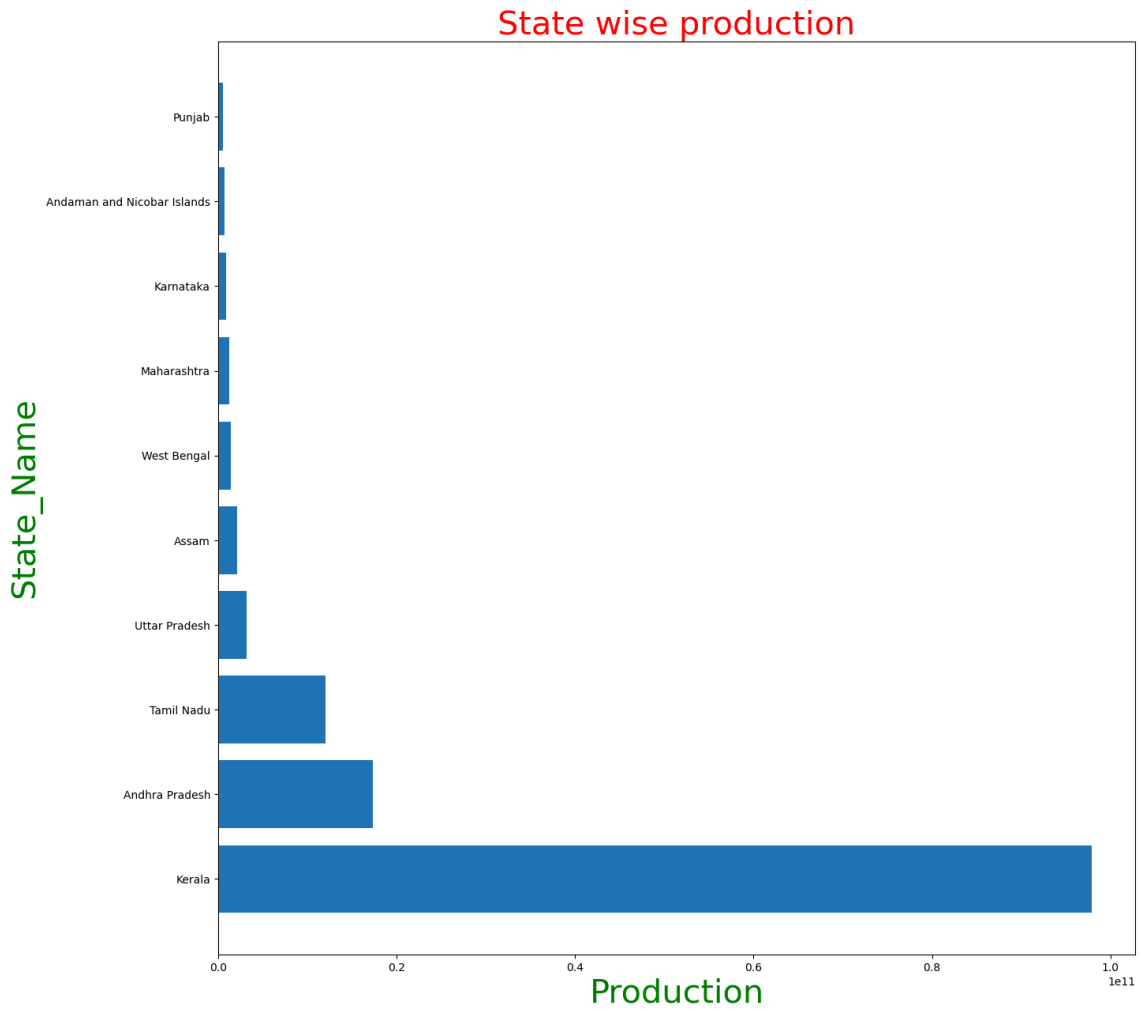
Out[43]:

	State_Name	Production
15	Kerala	9.788005e+10
1	Andhra Pradesh	1.732459e+10
27	Tamil Nadu	1.207644e+10
30	Uttar Pradesh	3.234493e+09
3	Assam	2.111752e+09
32	West Bengal	1.397904e+09
17	Maharashtra	1.263641e+09
14	Karnataka	8.634298e+08
0	Andaman and Nicobar Islands	7.182232e+08
24	Punjab	5.863850e+08
9	Gujarat	5.242913e+08
8	Goa	5.057558e+08
16	Madhya Pradesh	4.488407e+08
23	Puducherry	3.847245e+08
10	Haryana	3.812739e+08
4	Bihar	3.664836e+08
28	Telangana	3.351479e+08
25	Rajasthan	2.813203e+08
22	Odisha	1.609041e+08
31	Uttarakhand	1.321774e+08
6	Chhattisgarh	1.009519e+08
11	Himachal Pradesh	1.780517e+07
12	Jammu and Kashmir	1.329102e+07
21	Nagaland	1.276595e+07
29	Tripura	1.252292e+07
19	Meghalaya	1.211250e+07
13	Jharkhand	1.077774e+07
2	Arunachal Pradesh	6.823913e+06
18	Manipur	5.230917e+06
26	Sikkim	2.435735e+06
7	Dadra and Nagar Haveli	1.847871e+06
20	Mizoram	1.661540e+06
5	Chandigarh	6.395650e+04

In [44]:

```
plt.figure(figsize=(15,15))
x = prod['Production'].head(10)
y = prod['State_Name'].head(10)
plt.barh( y,x)
plt.title('State wise production', fontsize= 30 , color = 'r')
plt.xlabel('Production', fontsize = 30 , color = 'g')
plt.ylabel('State_Name', fontsize = 30, color = 'g')
plt.show()
```

executed in 288ms, finished 17:32:17 2024-07-02



In [45]:

```
df.Production.describe()
```

executed in 19ms, finished 17:32:18 2024-07-02

Out[45]:

```
count    2.423610e+05
mean     5.825034e+05
std      1.706581e+07
min      0.000000e+00
25%      8.800000e+01
50%      7.290000e+02
75%      7.023000e+03
max      1.250800e+09
Name: Production, dtype: float64
```

2. Crop wise Production

In [46]:
```
# TOP 10 CROPS BY PRODUCTION
crop1 = df.groupby(by='Crop')['Production'].sum().reset_index().sort_va
crop1
```

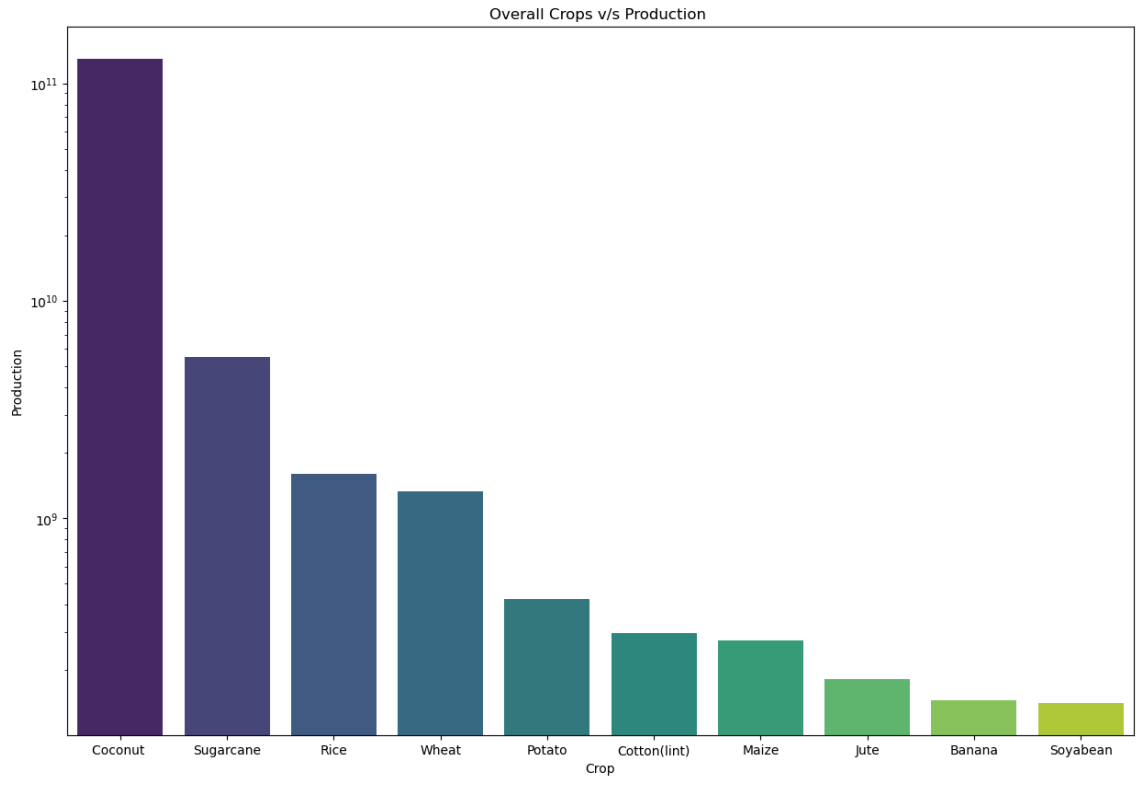
executed in 28ms, finished 17:32:20 2024-07-02

Out[46]:

	Crop	Production
28	Coconut	1.299816e+11
106	Sugarcane	5.535682e+09
95	Rice	1.605470e+09
119	Wheat	1.332826e+09
87	Potato	4.248263e+08
33	Cotton(lint)	2.970000e+08
59	Maize	2.733418e+08
49	Jute	1.815582e+08
7	Banana	1.461327e+08
105	Soyabean	1.418372e+08

In [47]:
```
plt.figure(figsize=(15,10))
sns.barplot(x='Crop', y='Production', data = crop1,palette='viridis')
plt.yscale('log')
plt.title('Overall Crops v/s Production ')
plt.show()
```

executed in 662ms, finished 17:32:20 2024-07-02



- The crops with maximum production are: Coconut, Sugarcane, Rice

3. Year wise production

In [48]:
```
year_wise_prod = df.groupby(by='Crop_Year')['Production'].sum().reset_i
year_wise_prod
```

executed in 16ms, finished 17:32:20 2024-07-02

Out[48]:

	Crop_Year	Production
14	2011	1.430890e+10
16	2013	1.290359e+10
9	2006	8.681913e+09
17	2014	8.664541e+09
7	2004	8.189462e+09
15	2012	8.171055e+09
8	2005	8.043757e+09
6	2003	7.917974e+09
11	2008	7.717018e+09
5	2002	7.696955e+09
12	2009	7.660494e+09
4	2001	7.465541e+09
3	2000	7.449709e+09
10	2007	6.879442e+09
2	1999	6.434666e+09
13	2010	6.307609e+09
1	1998	5.825321e+09
0	1997	8.512329e+08
18	2015	6.935065e+06

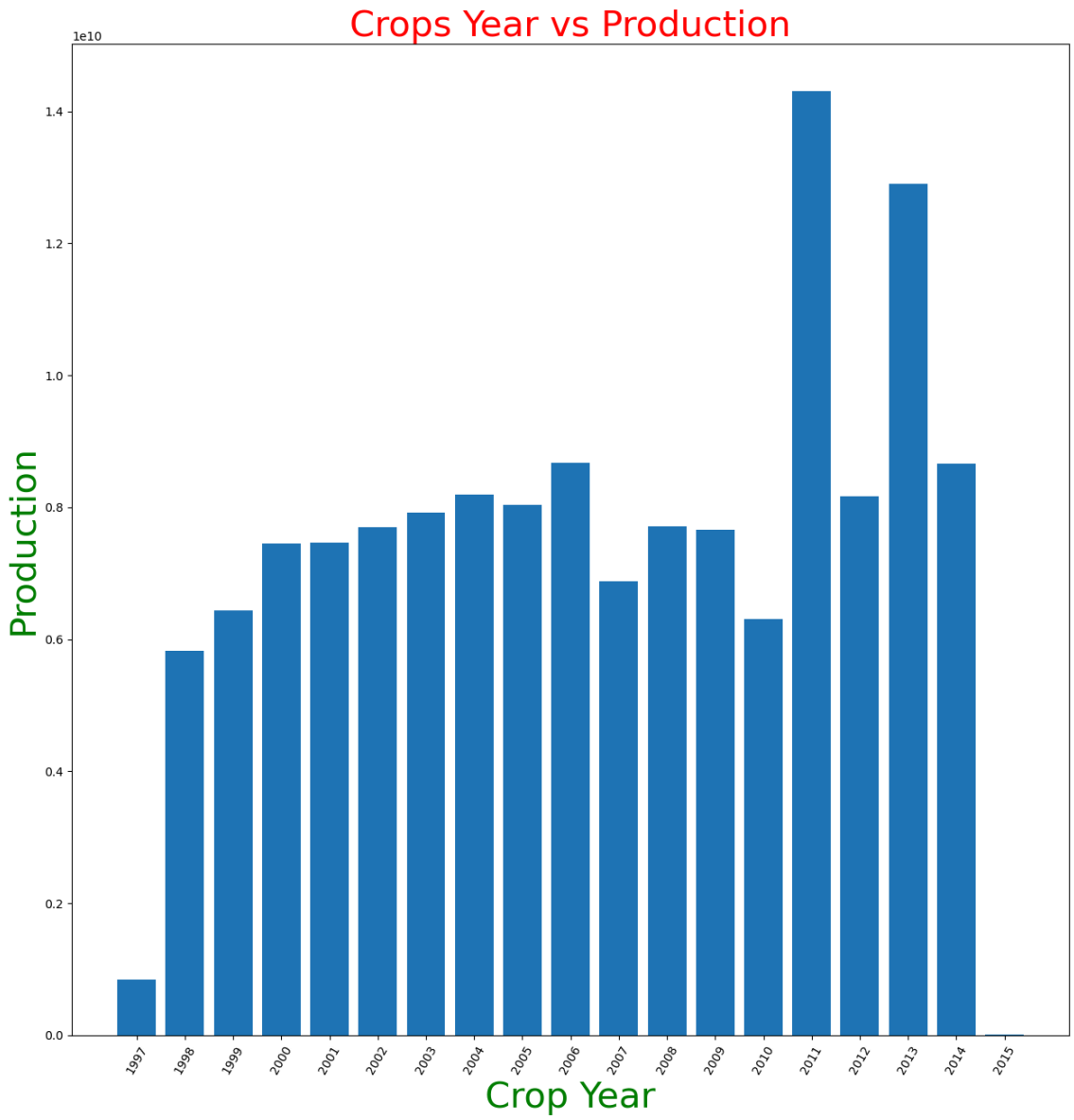
In [49]:

```
plt.figure(figsize=(15,15))
plt.bar(year_wise_prod.Crop_Year, year_wise_prod.Production, width=0.8)

plt.xticks(year_wise_prod.Crop_Year, rotation = 60)
plt.xlabel("Crop Year", fontsize = 30, color = 'g')
plt.ylabel("Production", fontsize = 30, color = 'g')
plt.title('Crops Year vs Production', fontsize = 30, color = 'r')
```

executed in 332ms, finished 17:32:21 2024-07-02

Out[49]: Text(0.5, 1.0, 'Crops Year vs Production')



- The production was maximum in years - 2011 and 2013

4. Season wise Production

In [50]:
```
season_wise_prod = df.groupby(by='Season')['Production'].sum().reset_ir
season_wise_prod
```

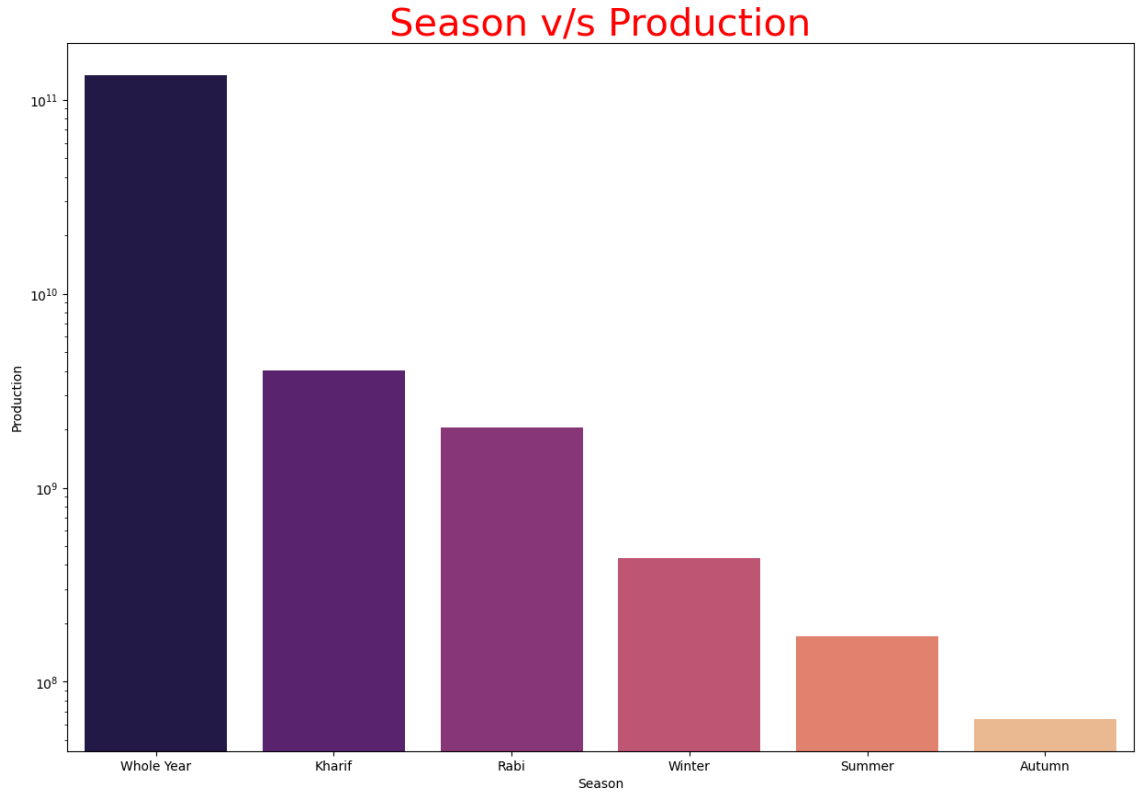
executed in 30ms, finished 17:32:21 2024-07-02

Out[50]:

	Season	Production
4	Whole Year	1.344248e+11
1	Kharif	4.029970e+09
2	Rabi	2.051688e+09
5	Winter	4.345498e+08
3	Summer	1.706579e+08
0	Autumn	6.441377e+07

In [51]:
```
plt.figure(figsize=(15,10))
sns.barplot(x='Season', y='Production',data = season_wise_prod, palette
plt.yscale('log')
plt.title('Season v/s Production', fontsize = 30,color = 'r')
plt.show()
```

executed in 477ms, finished 17:32:22 2024-07-02



- Top crop categories which shows high production values are Whole Year(Annual growing plants), Kharif and Rabi crops.

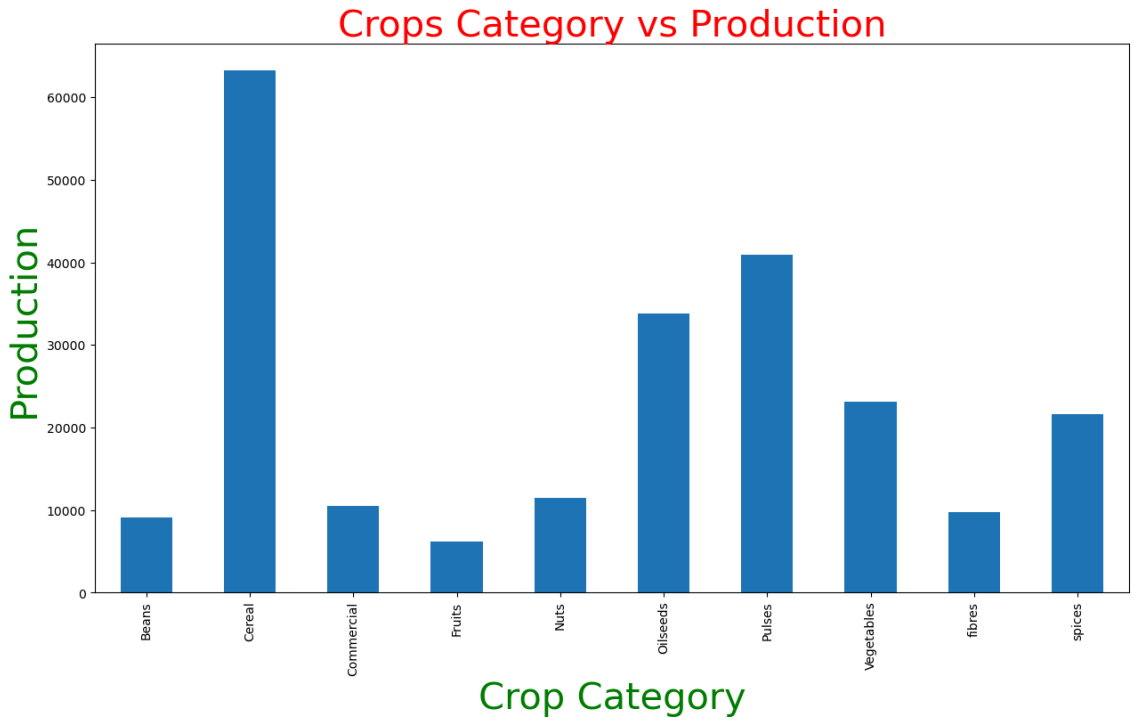
- These crop are generally dependent on monsoons.

5. Category wise production

In [52]:

```
plt.figure(figsize=(15,8))
plt.tick_params(labelsize=10)
df.groupby("crop_category")["Production"].agg("count").plot.bar()
plt.xlabel("Crop Category",  fontsize = 30, color = 'g')
plt.ylabel("Production", fontsize = 30, color = 'g')
plt.title('Crops Category vs Production', fontsize = 30, color = 'r')
plt.show()
```

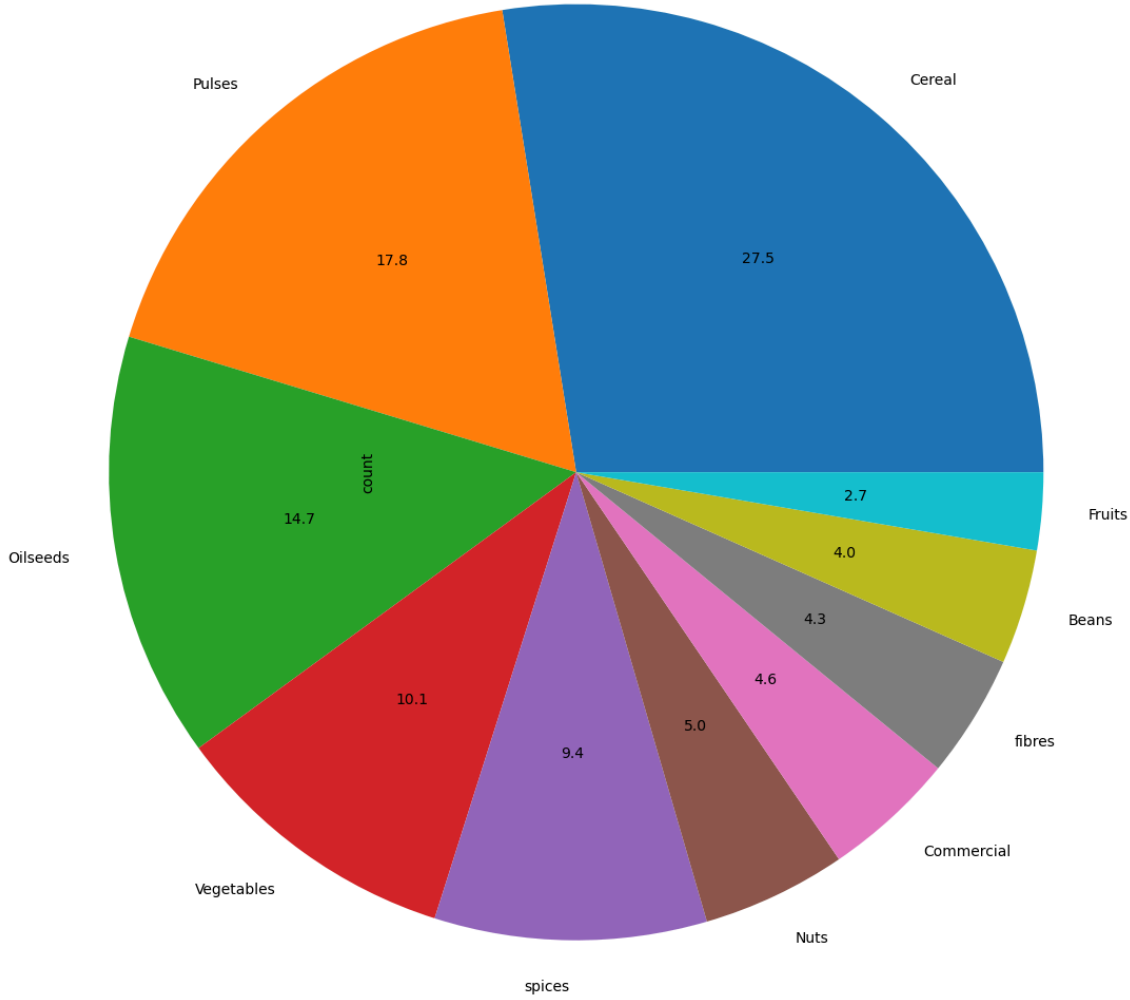
executed in 330ms, finished 17:32:23 2024-07-02



6. Different proportion of crop categories for india

In [53]:
```
df1=df["crop_category"].value_counts()
df1.plot(radius=3,kind="pie",autopct="%1.1f",pctdistance=0.6)
plt.tick_params(labelsize=10)
```

executed in 232ms, finished 17:32:24 2024-07-02



Top producing crop categories are

- Cereals - 27.5%

- Pulses - 17.8%

- Oilseeds - 14.7%

State wise crop production with different categories of crops

In [54]:

```
state_wise = pd.crosstab(df['State_Name'],df['crop_category'])
state_wise
```

executed in 74ms, finished 17:32:26 2024-07-02

Out[54]:

crop_category State_Name	Beans	Cereal	Commercial	Fruits	Nuts	Oilseeds	Pulses	Vegetables	fibre
Andaman and Nicobar Islands	0	20	15	16	37	11	9	20	
Andhra Pradesh	386	2264	474	502	674	1101	1336	1046	33
Arunachal Pradesh	26	1021	168	0	26	343	67	257	
Assam	0	2952	854	920	400	2097	2234	1781	128
Bihar	280	6108	756	226	130	2504	3731	1775	92
Chandigarh	0	39	0	0	0	7	14	26	
Chhattisgarh	646	1805	316	264	261	1496	2087	1143	53
Dadra and Nagar Haveli	0	116	12	9	9	30	64	0	1
Goa	0	62	22	16	47	0	32	0	
Gujarat	403	2466	372	157	683	1029	1521	473	32
Haryana	108	1427	259	52	126	543	860	463	25
Himachal Pradesh	179	726	67	0	54	236	530	214	3
Jammu and Kashmir	12	562	42	24	7	233	307	196	4
Jharkhand	0	575	16	0	0	124	304	247	
Karnataka	1096	5295	615	598	1470	3135	2776	1763	60
Kerala	3	819	236	437	536	168	13	636	1
Madhya Pradesh	962	5115	826	659	768	3281	3993	2738	92
Maharashtra	477	4009	458	83	868	3189	2326	56	46
Manipur	31	151	40	228	4	49	160	347	1
Meghalaya	113	606	182	162	143	329	314	399	17
Mizoram	42	230	123	0	15	143	213	96	6
Nagaland	211	1054	160	0	144	718	873	302	19
Odisha	629	3871	607	0	1156	2335	1760	909	28
Puducherry	0	198	30	73	98	51	101	84	3
Punjab	104	1123	216	0	75	496	728	0	18
Rajasthan	871	2634	518	257	444	1713	2174	1048	67
Sikkim	72	391	0	8	0	91	136	8	
Tamil Nadu	479	2680	623	992	1076	1235	1466	1827	55
Telangana	259	1365	250	201	338	774	882	416	19
Tripura	0	240	80	0	119	144	469	20	22
Uttar Pradesh	1112	9719	1741	269	958	4028	6549	3734	72
Uttarakhand	360	1423	127	0	76	626	1236	511	

crop_category State_Name	Beans	Cereal	Commercial	Fruits	Nuts	Oilseeds	Pulses	Vegetables	fibre
West Bengal	254	2217	356	0	730	1542	1633	619	71

Some Questions and Answers

1. Which Crop is seen in high frequency and when and where is it grown in India?

In [55]:
```
1 df.Crop.value_counts().head(10)
```
executed in 20ms, finished 17:32:27 2024-07-02

Out[55]:
```
Crop
Rice                  15082
Maize                 13787
Moong(Green Gram)     10106
Urad                   9710
Sesamum                8821
Groundnut              8770
Wheat                  7878
Sugarcane              7827
Rapeseed &Mustard      7533
Arhar/Tur              7476
Name: count, dtype: int64
```

In [56]:

```
rice_df = df[df['Crop'] == 'Rice']
rice_df.head(10)
```

executed in 37ms, finished 17:32:27 2024-07-02

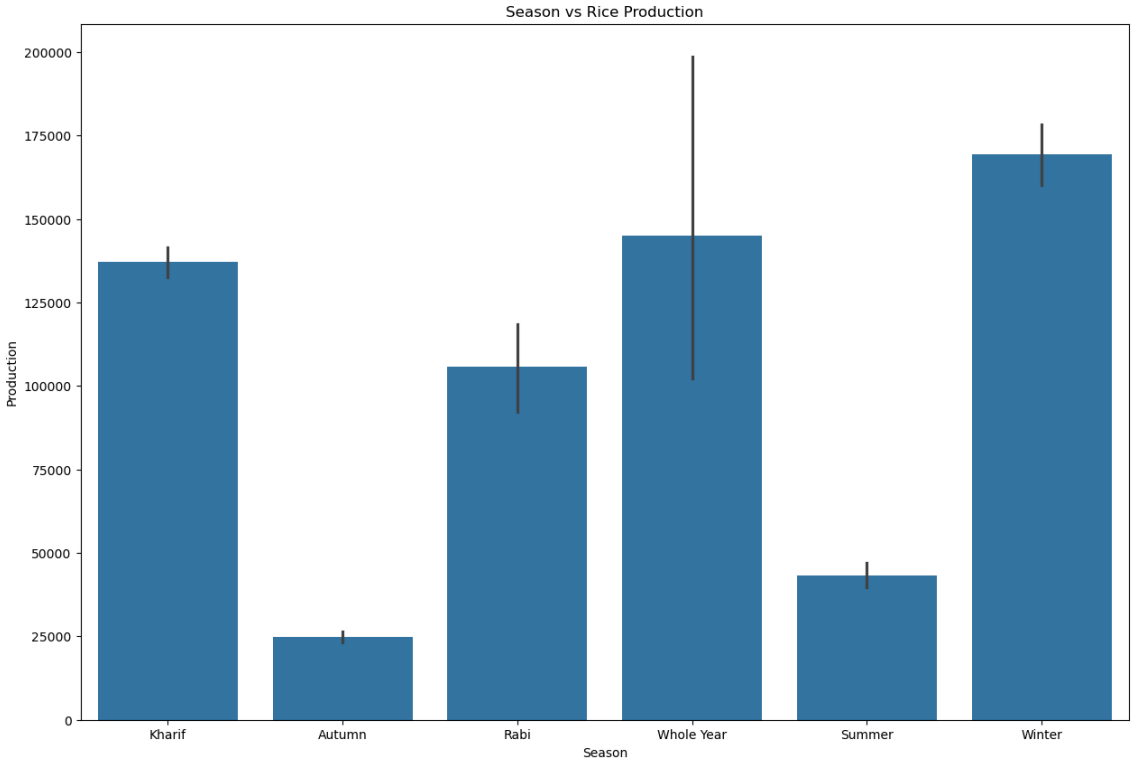
Out[56]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	crop_categ
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.00	321.00	Cer
12	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Rice	83.00	300.00	Cer
18	Andaman and Nicobar Islands	NICOBARS	2002	Kharif	Rice	189.20	510.84	Cer
27	Andaman and Nicobar Islands	NICOBARS	2003	Kharif	Rice	52.00	90.17	Cer
36	Andaman and Nicobar Islands	NICOBARS	2004	Kharif	Rice	52.94	72.57	Cer
45	Andaman and Nicobar Islands	NICOBARS	2005	Kharif	Rice	2.09	12.06	Cer
64	Andaman and Nicobar Islands	NICOBARS	2010	Autumn	Rice	3.50	10.00	Cer
81	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	2000	Kharif	Rice	10779.00	31863.00	Cer
92	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	2001	Kharif	Rice	9718.00	27033.00	Cer
98	Andaman and Nicobar Islands	NORTH AND MIDDLE ANDAMAN	2006	Kharif	Rice	6854.30	18995.62	Cer

In [57]:

```
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(x="Season",y="Production",data=rice_df);
plt.title('Season vs Rice Production')
plt.show()
```

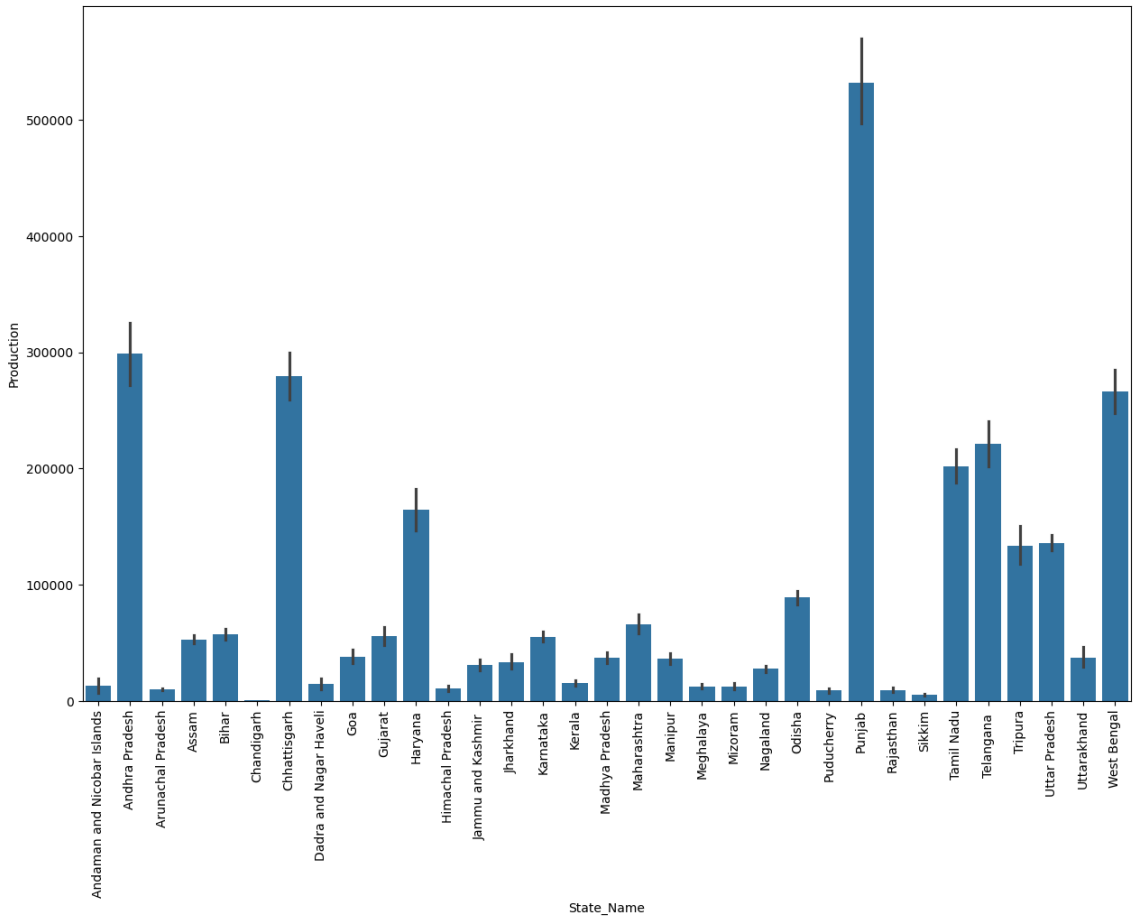
executed in 605ms, finished 17:32:28 2024-07-02



In [58]:

```
plt.figure(figsize=(15,10))
sns.barplot(x="State_Name",y="Production",data=rice_df)
plt.xticks(rotation=90)
plt.show()
```

executed in 1.21s, finished 17:32:30 2024-07-02



In [59]:

```
rice_prod_dis = rice_df.groupby("District_Name")["Production"].sum().re
sum_rice = rice_prod_dis["Production"].sum()
rice_prod_dis["Percent_of_Production"] = rice_prod_dis["Production"].ma
rice_prod_dis.head(10)
```

executed in 15ms, finished 17:32:30 2024-07-02

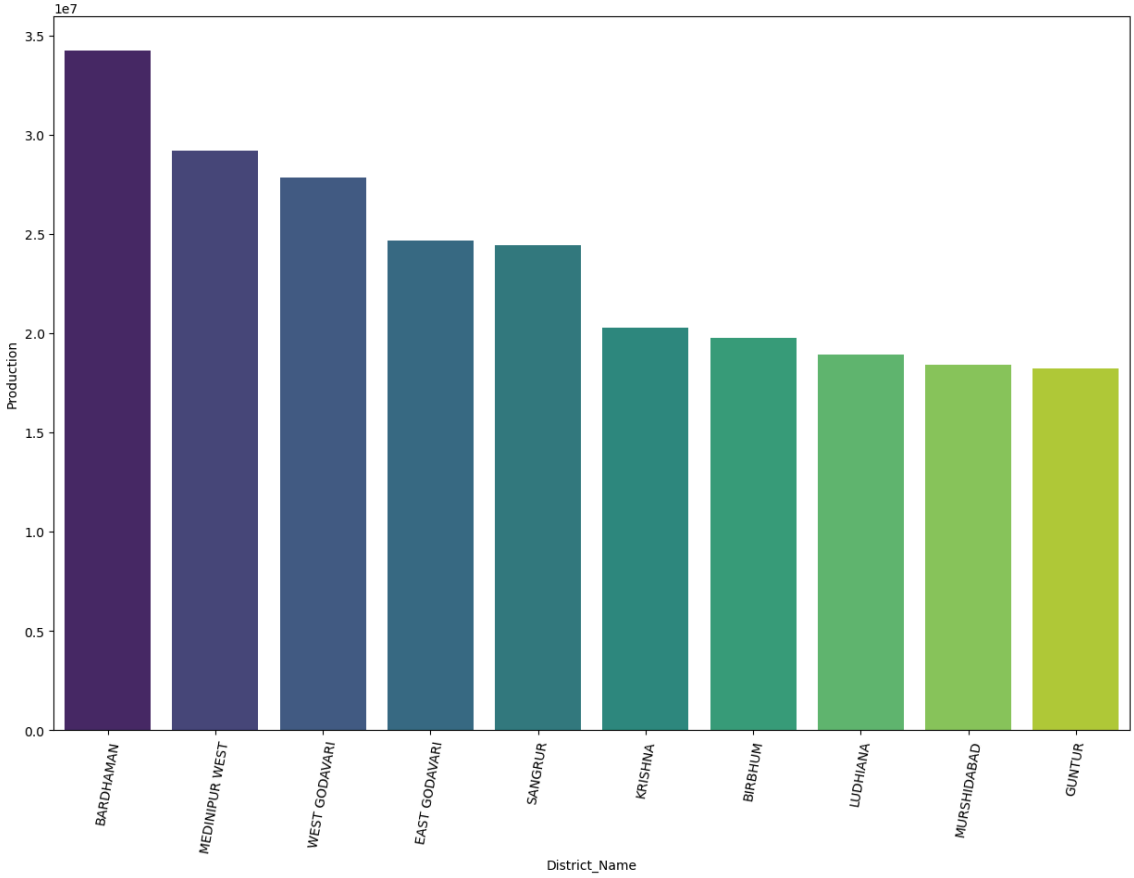
Out[59]:

	District_Name	Production	Percent_of_Production
58	BARDHAMAN	34239976.0	2.132707
374	MEDINIPUR WEST	29192719.0	1.818328
612	WEST GODAVARI	27845309.0	1.734402
169	EAST GODAVARI	24690929.0	1.537925
494	SANGRUR	24448000.0	1.522794
325	KRISHNA	20280606.0	1.263219
90	BIRBHUM	19753571.0	1.230391
347	LUDHIANA	18950000.0	1.180339
386	MURSHIDABAD	18403217.0	1.146282
214	GUNTUR	18245831.0	1.136479

In [60]:

```
plt.figure(figsize=(15,10))
sns.barplot(x="District_Name",y="Production",data=rice_prod_dis.head(10
plt.xticks(rotation=80)
plt.show()
```

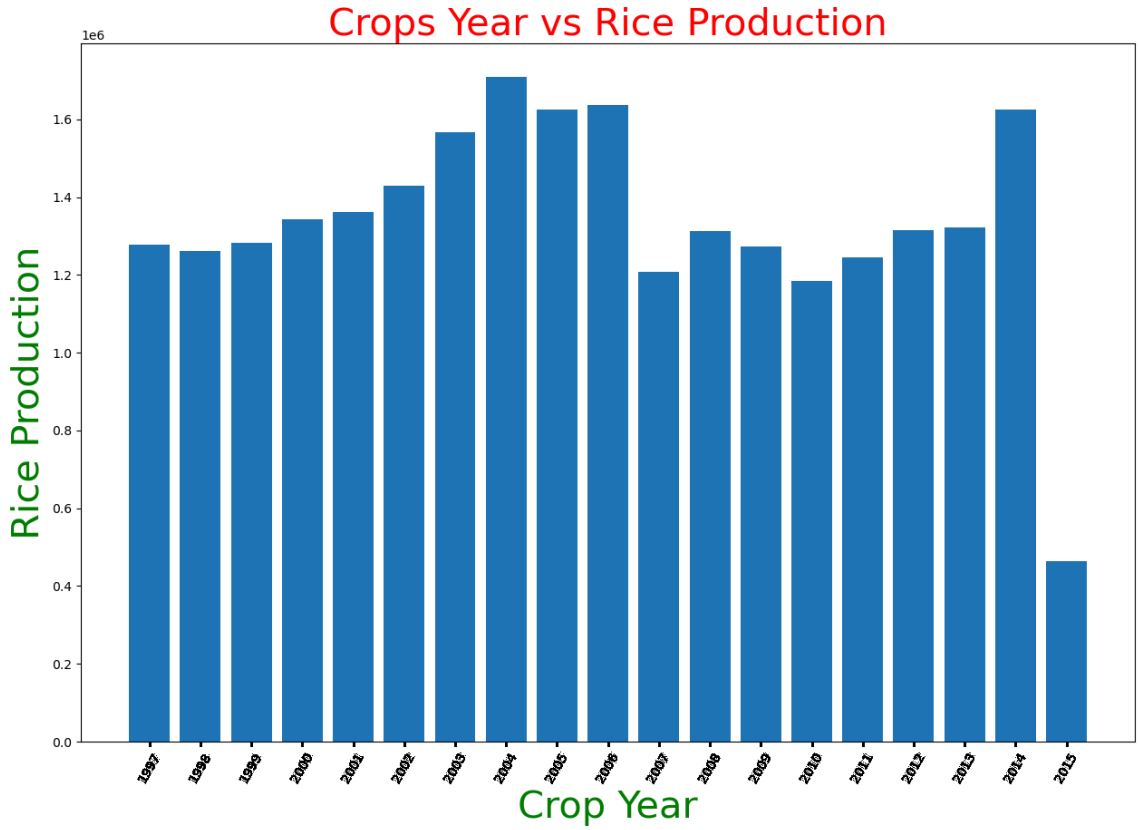
executed in 258ms, finished 17:32:30 2024-07-02



In [61]:

```
plt.figure(figsize= (15,10))
x= rice_df['Crop_Year']
y= rice_df["Production"]
plt.xticks(rice_df['Crop_Year'], rotation = 60)
plt.xlabel("Crop Year",  fontsize = 30, color = 'g')
plt.ylabel("Rice Production",  fontsize = 30, color = 'g')
plt.title('Crops Year vs Rice Production',  fontsize = 30, color = 'r')
plt.bar(x,y)
plt.show()
```

executed in 1m 14.5s, finished 17:33:44 2024-07-02



Answers

- Rice is seen to have more frequency.

- Rice is grown majorly in Winter.

- State wise Punjab dominates in rice production

- District wise its BARDHAMAN(2.13%), MEDINIPUR WEST(1.8%) and WEST GODAVARI(1.73%) which contributes to total rice production.

- Year wise 2004 is the year when production reached the peak production.

2. Which states ranks high in area wise crop production in India? Substantiate with facts and Figures.

In [62]:
```
df_area = df.groupby('State_Name')['Area'].sum().reset_index().sort_val
df_area.head(10)
```

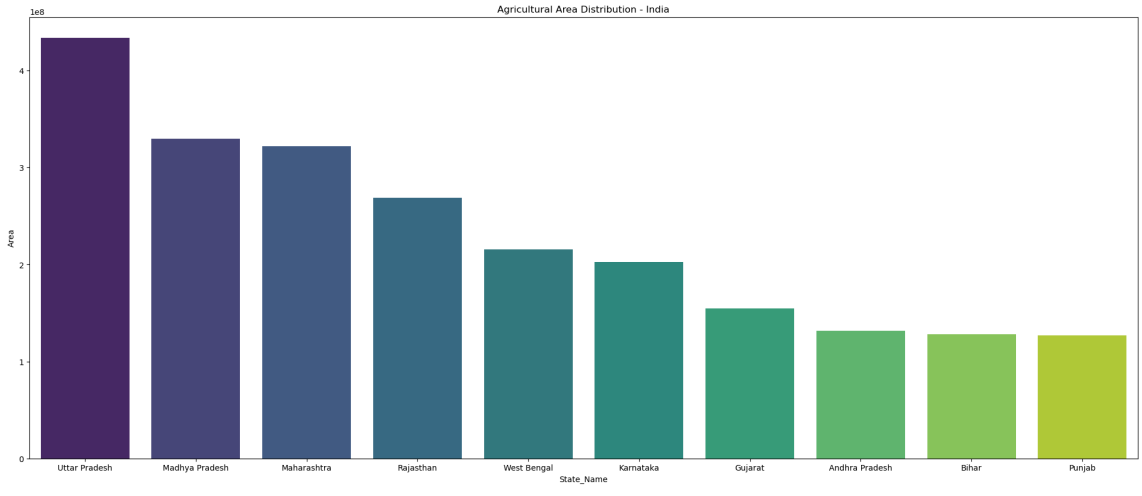
executed in 55ms, finished 17:33:44 2024-07-02

Out[62]:

	State_Name	Area
30	Uttar Pradesh	4.336223e+08
16	Madhya Pradesh	3.297913e+08
17	Maharashtra	3.221860e+08
25	Rajasthan	2.687882e+08
32	West Bengal	2.154030e+08
14	Karnataka	2.029086e+08
9	Gujarat	1.549261e+08
1	Andhra Pradesh	1.315073e+08
4	Bihar	1.282695e+08
24	Punjab	1.267152e+08

In [63]:
```
fig, ax = plt.subplots(figsize=(25,10))
sns.barplot(x='State_Name', y='Area',data= df_area.head(10) ,errwidth=0
plt.title('Agricultural Area Distribution - India');
```

executed in 384ms, finished 17:33:45 2024-07-02



- Top cultivating states based on the Cultivation area are: Uttar Pradesh, Madhya Pradesh and Maharashtra.

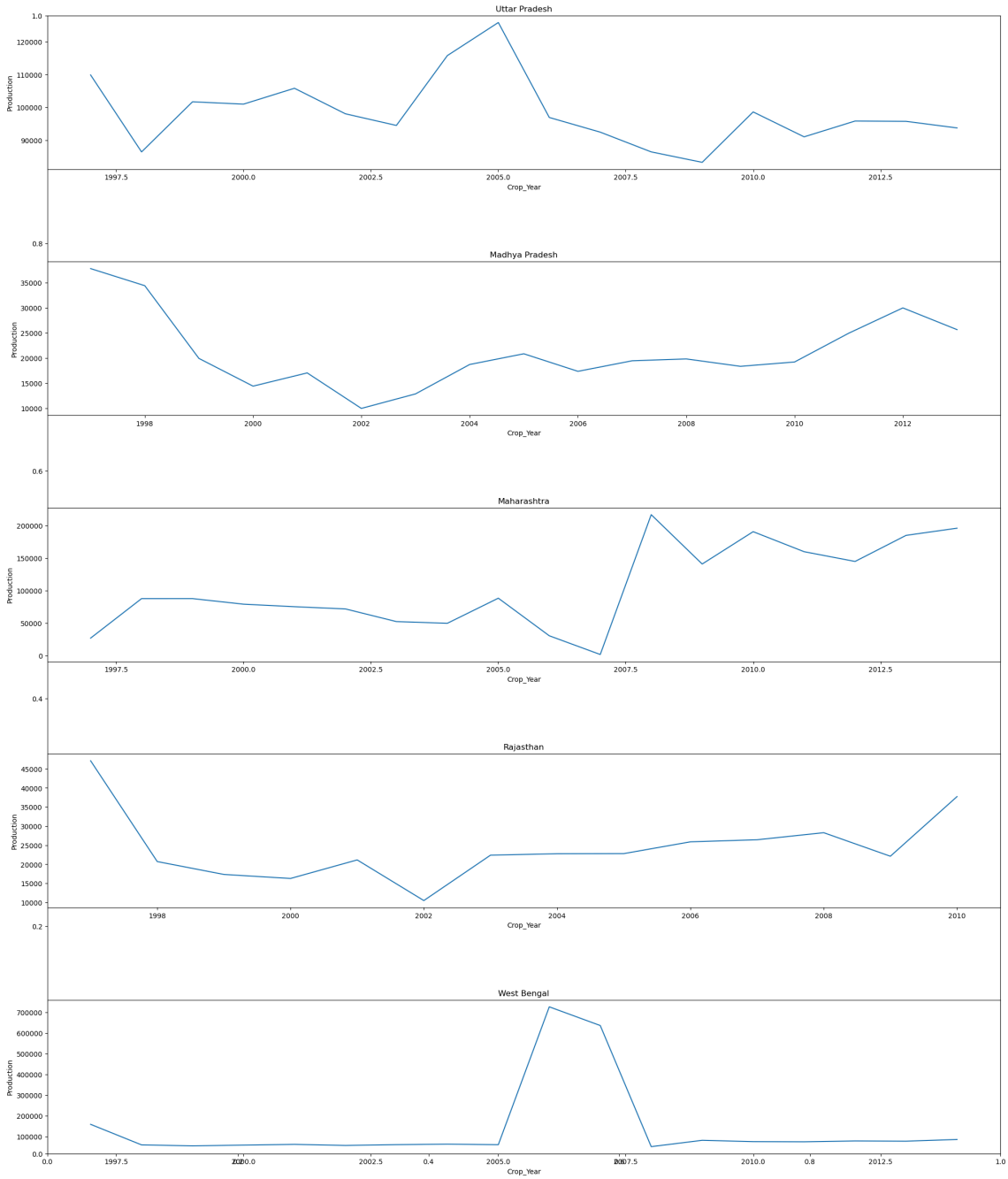
In [64]:

```
df_area_5 = df_area.head(5)

fig, ax = plt.subplots(figsize=(25,30), sharey='col')
count = 1

for state in df_area_5.State_Name.unique():
    plt.subplot(len(df_area_5.State_Name.unique()),1,count)
    sns.lineplot(x=df[df.State_Name==state]['Crop_Year'],y=df[df.State_
    plt.subplots_adjust(hspace=0.6)
    plt.title(state)
    count+=1;
```

executed in 1.48s, finished 17:33:51 2024-07-02



- Uttar Pradesh - High Production was seen in 2005 and after that it's been blueucing gradually.

- Madhya Pradesh - 1998 showed a high production and then there was gradual blueuction but it picked up and 2012 also showed a peak in Production.

- Maharashtra - Production went down drastically in 2006 and again the levels went up and hit a high peak after 2007.

- Rajasthan - Production hit a all time low in the year 2002 and then picked up by 2010.

- West Bengal - Production hit a peak around 2006 but it has hit a low after 2007 and never recoveblue back

3. Find the most efficient state (in terms of most production per unit area). Also find the most efficient state for some of the crop categories.

In [65]:
```
efficiency = df.groupby('State_Name')['Production_per_unit_area'].mean(
efficiency.head(10)
```

executed in 28ms, finished 17:33:55 2024-07-02

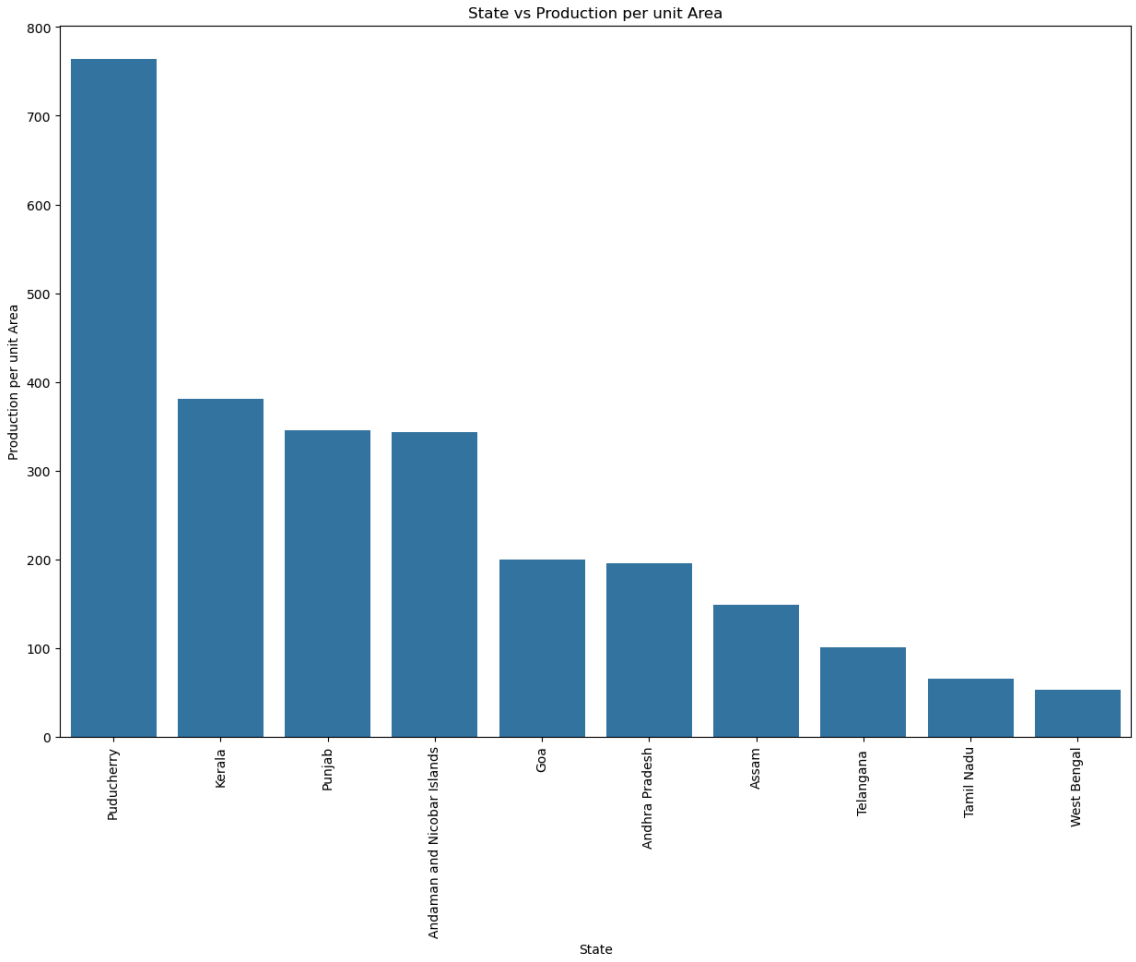
Out[65]:

	State_Name	Production_per_unit_area
23	Puducherry	763.596415
15	Kerala	381.272231
24	Punjab	345.754577
0	Andaman and Nicobar Islands	343.553142
8	Goa	199.160564
1	Andhra Pradesh	195.138587
3	Assam	148.630468
28	Telangana	101.211017
27	Tamil Nadu	65.287593
32	West Bengal	53.334813

In [66]:

```
fig, ax = plt.subplots(figsize=(15,10))
a = sns.barplot(x='State_Name', y='Production_per_unit_area',data = eff
plt.title('State vs Production per unit Area');
plt.xlabel("State")
plt.ylabel("Production per unit Area")
a.set_xticklabels(
    labels=efficiency.State_Name.head(10), rotation=90)
plt.show()
```

executed in 274ms, finished 17:33:56 2024-07-02



- Most efficieent states in terms of production per unit area are - Puducherry, Kerala and Punjab.

In [67]:

```
df.crop_category.unique()
```

executed in 11ms, finished 17:33:58 2024-07-02

Out[67]:

```
array(['Nuts', 'Pulses', 'Cereal', 'Fruits', None, 'spices', 'Commercial',
       'Vegetables', 'Oilseeds', 'fibres', 'Beans'], dtype=object)
```

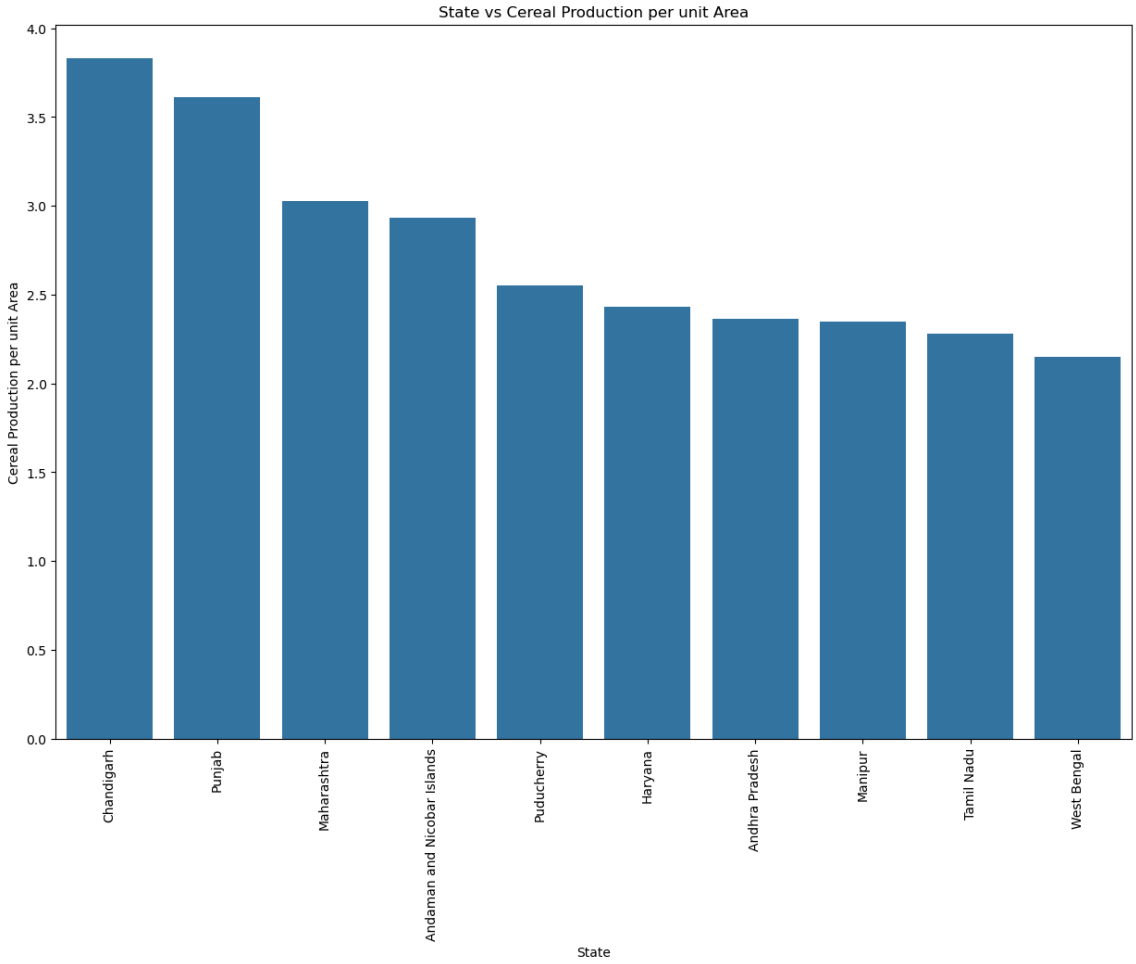
In [68]:

```
cereal_df = df[df["crop_category"]=="Cereal"]

efficiency_cereal = cereal_df.groupby('State_Name')['Production_per_uni
efficiency_cereal.head(10)

fig, ax = plt.subplots(figsize=(15,10))
b = sns.barplot(x='State_Name', y='Production_per_unit_area',data = eff
plt.title('State vs Cereal Production per unit Area');
plt.xlabel("State")
plt.ylabel("Cereal Production per unit Area")
b.set_xticklabels(
    labels=efficiency_cereal.State_Name.head(10), rotation=90)
plt.show()
```

executed in 318ms, finished 17:34:00 2024-07-02



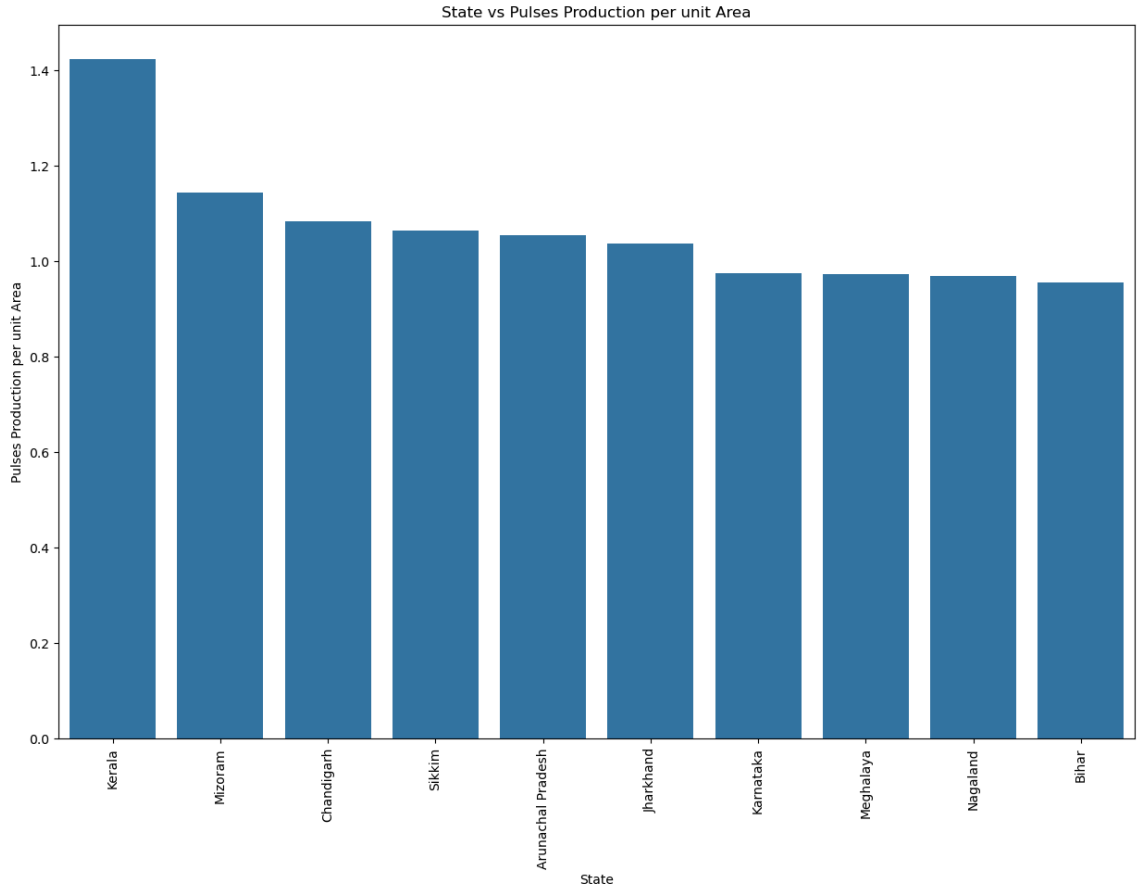
In [69]:

```
pulses_df = df[df["crop_category"]=="Pulses"]

efficiency_pulses = pulses_df.groupby('State_Name')['Production_per_uni
efficiency_pulses.head(10)

fig, ax = plt.subplots(figsize=(15,10))
b = sns.barplot(x='State_Name', y='Production_per_unit_area',data = eff
plt.title('State vs Pulses Production per unit Area');
plt.xlabel("State")
plt.ylabel("Pulses Production per unit Area")
b.set_xticklabels(
    labels=efficiency_pulses.State_Name.head(10), rotation=90)
plt.show()
```

executed in 269ms, finished 17:34:01 2024-07-02



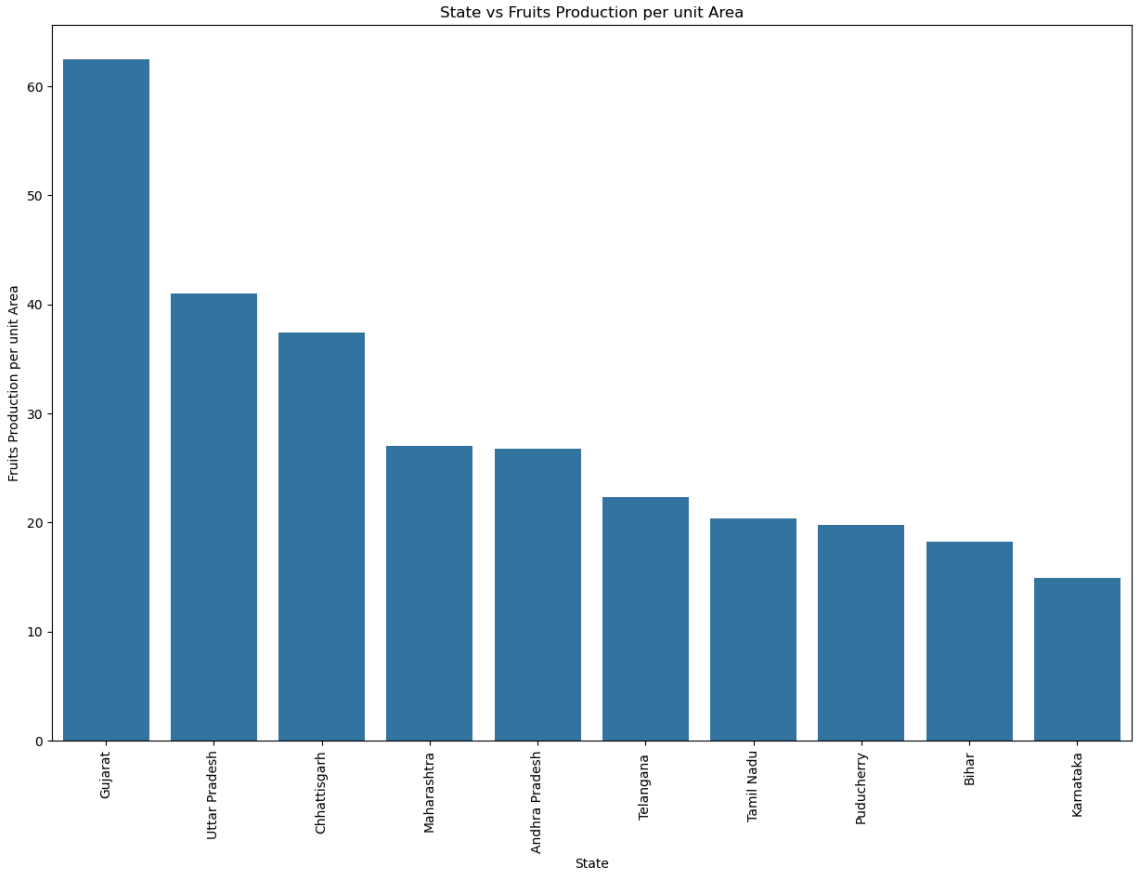
In [70]:

```
fruits_df = df[df["crop_category"]=="Fruits"]

efficiency_fruits = fruits_df.groupby('State_Name')['Production_per_uni
efficiency_fruits.head(10)

fig, ax = plt.subplots(figsize=(15,10))
b = sns.barplot(x='State_Name', y='Production_per_unit_area',data = eff
plt.title('State vs Fruits Production per unit Area');
plt.xlabel("State")
plt.ylabel("Fruits Production per unit Area")
b.set_xticklabels(
    labels=efficiency_fruits.State_Name.head(10), rotation=90)
plt.show()
```

executed in 275ms, finished 17:34:02 2024-07-02



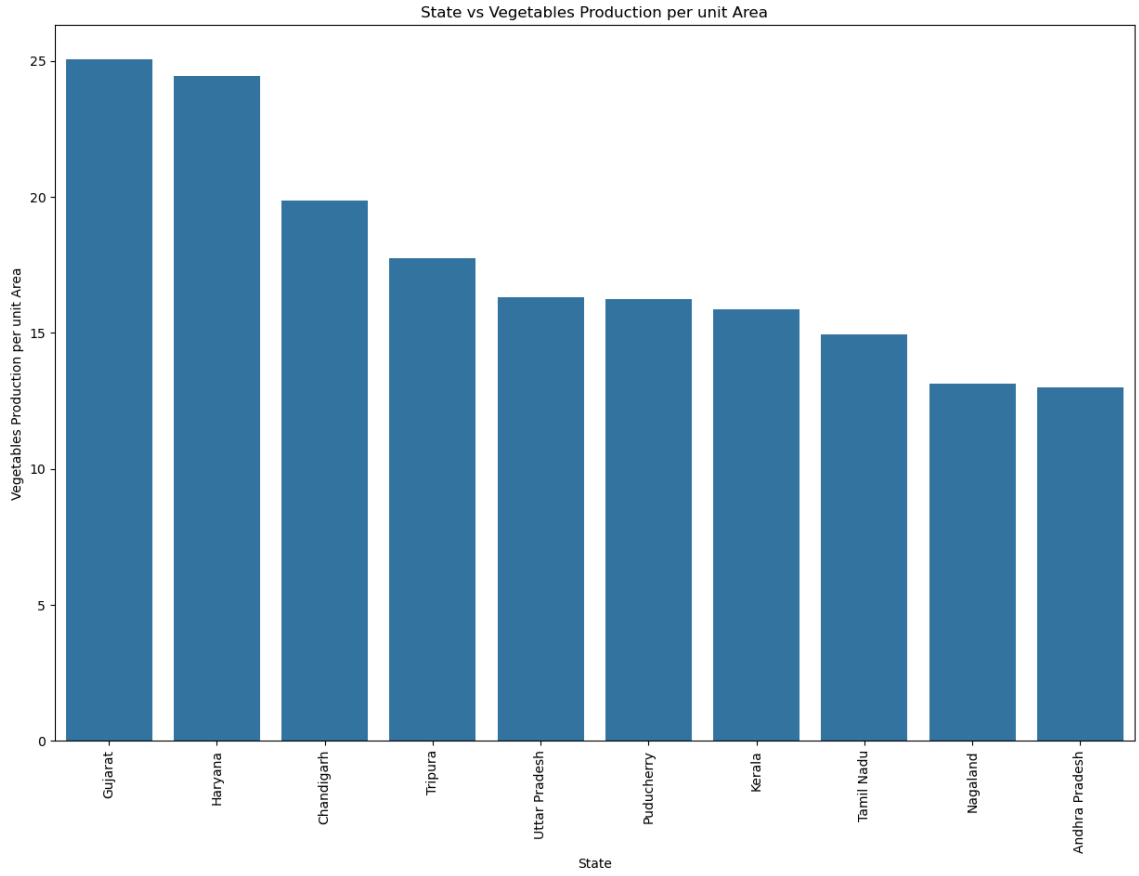
In [71]:

```
veg_df = df[df["crop_category"]=="Vegetables"]

efficiency_veg = veg_df.groupby('State_Name')['Production_per_unit_area
efficiency_veg.head(10)

fig, ax = plt.subplots(figsize=(15,10))
b = sns.barplot(x='State_Name', y='Production_per_unit_area',data = eff
plt.title('State vs Vegetables Production per unit Area');
plt.xlabel("State")
plt.ylabel("Vegetables Production per unit Area")
b.set_xticklabels(
    labels=efficiency_veg.State_Name.head(10), rotation=90)
plt.show()
```

executed in 275ms, finished 17:34:04 2024-07-02



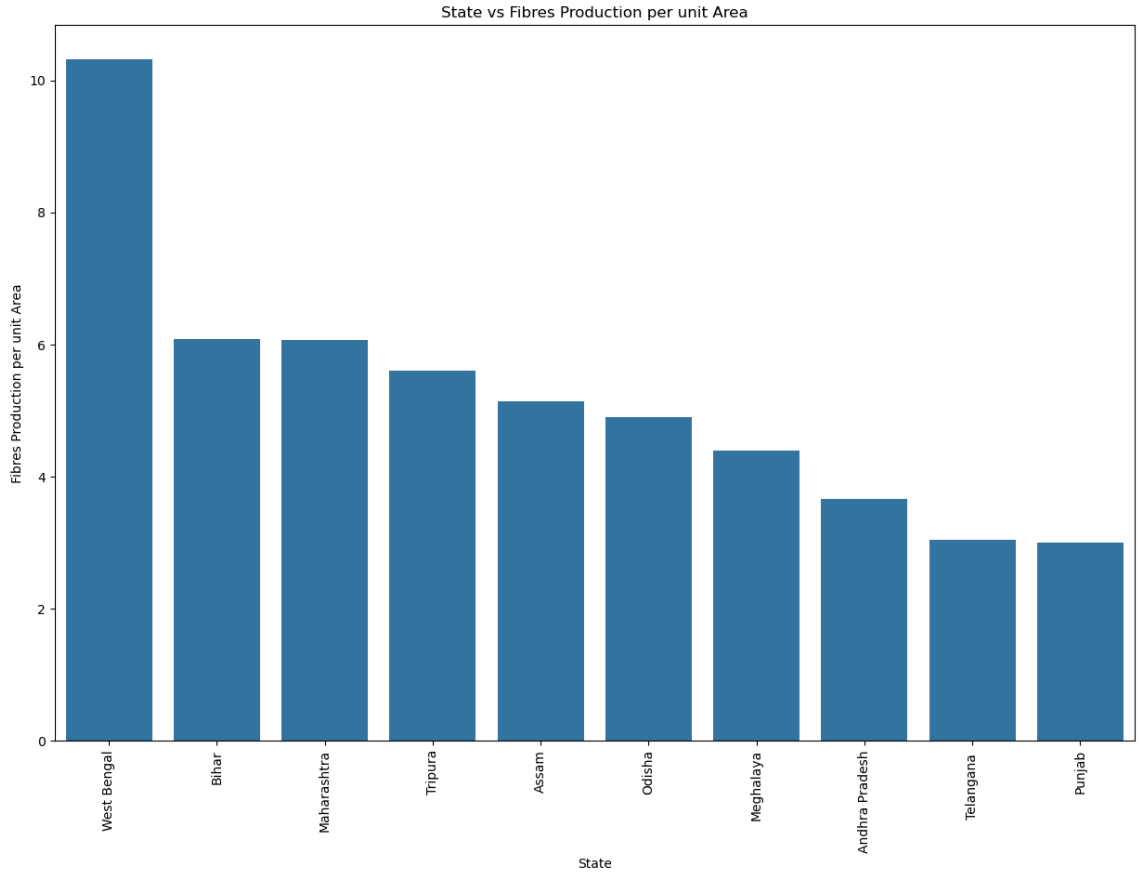
In [72]:

```
fibres_df = df[df["crop_category"]=="fibres"]

efficiency_fibres = fibres_df.groupby('State_Name')['Production_per_uni
efficiency_fibres.head(10)

fig, ax = plt.subplots(figsize=(15,10))
b = sns.barplot(x='State_Name', y='Production_per_unit_area',data = eff
plt.title('State vs Fibres Production per unit Area');
plt.xlabel("State")
plt.ylabel("Fibres Production per unit Area")
b.set_xticklabels(
    labels=efficiency_fibres.State_Name.head(10), rotation=90)
plt.show()
```

executed in 255ms, finished 17:34:07 2024-07-02



Most efficieent states in terms of production per unit area for various categories of crops are-

- Cereals - Chandigarh

- Pulses - Kerala

- Fruits - Gujrat

- Vegetables - Gujrat

- Fibres - West Bengal

Inferences and Conclusion

We started with 246091 samples with 7 columns. Production Variable had 3730 (about 1.52% of total sample size) missing values which was dropped and working dataset has 242361 sample size. A

Univarate-Analysis

- State_Name - 33 Names including Union territories. Top states contributing to dataset are Uttar Pardesh, Madhya Pradesh and Karnataka.
- Crop_Year - Dataset represents data for 19 years from 1997 to 2015 and maximum data from 2003, 2002 & 2007.
- Season - We see six seasons with maximum data from Kharif, Rabi and Whole year.
- Crop - We data for 124 different crops with maximum data from Rice, Maize and Moong(Green Gram).
- Area: Huge margin area used for production from 1 to 8580100 unit area. Distribution is highly right skewed due to lot of outliers.
- Production value ranges from 0 to 1.25e+09 and Distribution is highly right skewed due to lot of outliers.

New Variables created

- crop_category - 124 crops were divided into Cereal, Pulses, oilseeds, Vegetables, spices, Nuts, Commercial, fibers, Beans, Fruits. Dateset shows top categories are Cereal, Pulses and oilseeds.
- prod_per_unit_area - This variable was created as by dividing production with the area.

Visualisation of Data

1. State wise Production
2. Crop wise Production
3. Year wise Production
4. Season wise Production
5. Crop Category wise Production
6. Different Proportion of crop Categories

Questions and Answers

1. Which Crop is seen in high frequency and when and where is it grown in India?

    - Rice is seen to have more frequency.
    - Rice is grown majorly in Winter.
    - State wise Punjab dominates in rice production
    - District wise its BARDHAMAN(2.13%), MEDINIPUR WEST(1.8%) and WEST GODAVARI(1.73%) which contributes to total rice production.
    - Year wise 2004 is the year when production reached the peak production.
2. Which states ranks high in area wise crop production in India? Substantiate with facts and Figures.

    - Top cultivating states based on the Cultivation area are: Uttar Pradesh, Madhya Pradesh, Maharashtra, Rajasthan and West Bengal.
    - Year wise trend of these states:
        - Uttar Pradesh - High Production was seen in 2005 and after that it’s been reducing gradually.

    - Madhya Pradesh - 1998 showed a high production and then there was gradual reduction but it picked up and 2012 also showed a peak in Production.
    - Maharashtra - Production went down drastically in 2006 and again the levels went up and hit a high peak after 2007.
    - Rajasthan - Production hit a all time low in the year 2002 and then picked up by 2010.
    - West Bengal - Production hit a peak around 2006 but it has hit a low after 2007 and never recovered back.
3. Find the most efficient state (in terms of most production per unit area). Also find the most efficient state for some of the crop categories.

    - Most efficieent states in terms of production per unit area are - Puducherry, Kerala and Punjab.
    - Most efficieent states in terms of production per unit area for various categories of crops are:
        - Cereals - Chandigarh
        - Pulses - Kerala
        - Fruits - Gujrat
        - Vegetables - Gujrat