Import the Required Libraries

In [11]:
```
# import 'Pandas'
import pandas as pd

# import 'Numpy'
import numpy as np

# import subpackage of Matplotlib
import matplotlib.pyplot as plt

# import 'Seaborn'
import seaborn as sns

# to suppress warnings
from warnings import filterwarnings
filterwarnings('ignore')

# display all columns of the dataframe
pd.options.display.max_columns = None

# display all rows of the dataframe
pd.options.display.max_rows = None

# to display the float values upto 6 decimal places
pd.options.display.float_format = '{:.0f}'.format
```

In [12]:
```
# pass width and height in inches to 'figure.figsize'
plt.rcParams['figure.figsize'] = [15,8]
```

Load the dataset:

In [13]:
```
# load the dataset using panda dataframe
crop_prod=pd.read_csv("Crop_Production_Data.csv")
```

In [14]:
```
crop_prod.head()
```

Out[14]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720	165

Information about dataset

- The Dataset is about the crop-production from year 1997 to 2015.
- The Dataset contains columns :
  - State_Name:The Name of the State
  - District_Name: The Name od District
  - Crop_Year: The Year of Production
  - Season: Season of the crop
  - Crop: Name of the crop
  - Area : Agricultural land
  - Production: The Production of the crop which is also our traget column

In [15]:
```
# Tells us how many rows and columns are there.
crop_prod.shape
```

Out[15]:
```
(246091, 7)
```

In [16]:
```
# Code to show the basic information about the data.
crop_prod.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     246053 non-null  object 
 1   District_Name  245871 non-null  object 
 2   Crop_Year      245992 non-null  float64
 3   Season         245999 non-null  object 
 4   Crop           246006 non-null  object 
 5   Area           245580 non-null  float64
 6   Production     242340 non-null  float64
dtypes: float64(3), object(4)
memory usage: 13.1+ MB
```

In [ ]:

In [17]:
```
#Show the density of the data
crop_prod.describe().T
```

Out[17]:

	count	mean	std	min	25%	50%	75%	max
Crop_Year	245992	2006	5	1997	2002	2006	2010	2015
Area	245580	12017	50571	0	80	583	4400	8580100
Production	242340	582554	17066552	0	88	729	7025	1250800000

In [18]:
```
#Code to show the basic central tendency of the data.
crop_prod.mean()
```

Out[18]:
```
Crop_Year      2006
Area          12017
Production   582554
dtype: float64
```

In [19]:
```
crop_prod.median()
```

Out[19]:
```
Crop_Year    2006
Area          583
Production    729
dtype: float64
```

In [20]:
```
#Show the Datatypes of the data
crop_prod.dtypes
```

Out[20]:
```
State_Name        object
District_Name     object
Crop_Year        float64
Season            object
Crop              object
Area             float64
Production       float64
dtype: object
```

Data Overview

- The dataset contains 246091 rows and 7 columns
- The size od dataset is arround 1722637 bit and memory usage is arroud 13.1+MB.
- There are 4 Categorical and 3 Numerical columns.
- Area and Productions columns contains outlier by looking at the diffrence between mean and median (50%)

In [ ]:

Null-value Handling:

In [21]:
```
# Show the percentage and count of missing values.
null_count=crop_prod.isnull().sum()
null_count
```

Out[21]:
```
State_Name         38
District_Name     220
Crop_Year          99
Season             92
Crop               85
Area              511
Production       3751
dtype: int64
```

In [22]:
```
count_perc=((crop_prod.isnull().sum())/len(crop_prod))*100
count_perc
```

Out[22]:
```
State_Name      0
District_Name   0
Crop_Year       0
Season          0
Crop            0
Area            0
Production      2
dtype: float64
```

In [23]:
```
null_df=pd.concat([null_count,count_perc], axis=1, keys=["Total","Percentage"]
null_df
```

Out[23]:

	Total	Percentage
State_Name	38	0
District_Name	220	0
Crop_Year	99	0
Season	92	0
Crop	85	0
Area	511	0
Production	3751	2

In [24]:
```
# Treat the missing value based on requirement.(Don't Remove the nul values)
```

- We will treat the null values of the columns one by one:

State_Name

In [25]:
```
crop_prod.State_Name.value_counts()
```

Out[25]:
```
Uttar Pradesh                  33306
Madhya Pradesh                 22943
Karnataka                      21122
Bihar                          18872
Assam                          14628
Odisha                         13575
Tamil Nadu                     13547
Maharashtra                    12628
Rajasthan                      12514
Chhattisgarh                   10709
Andhra Pradesh                  9613
West Bengal                     9607
Gujarat                         8436
Haryana                         5875
Telangana                       5649
Uttarakhand                     4892
Kerala                          4261
Nagaland                        3906
Punjab                          3173
Meghalaya                       2867
Arunachal Pradesh               2546
Himachal Pradesh                2494
Jammu and Kashmir               1634
Tripura                         1412
Manipur                         1267
Jharkhand                       1266
Mizoram                          957
Puducherry                       876
Sikkim                           714
Dadra and Nagar Haveli           263
Goa                              208
Andaman and Nicobar Islands      203
Chandigarh                        90
Name: State_Name, dtype: int64
```

In [26]:
```
crop_prod[crop_prod.State_Name.isnull()]
```

Out[26]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
3181	NaN	GUNTUR	2014	Kharif	Arhar/Tur	NaN	13805
3207	NaN	GUNTUR	2014	Rabi	Arhar/Tur	NaN	301
3232	NaN	KADAPA	1997	Kharif	Arhar/Tur	NaN	1400
3258	NaN	KADAPA	1998	Kharif	Arhar/Tur	NaN	4400
3319	NaN	KADAPA	2000	Kharif	Arhar/Tur	NaN	8369
3341	NaN	KADAPA	2000	Rabi	Arhar/Tur	NaN	93
3368	NaN	KADAPA	2001	Kharif	Arhar/Tur	NaN	7139
3389	NaN	KADAPA	2001	Rabi	Arhar/Tur	NaN	40
3416	NaN	KADAPA	2002	Kharif	NaN	NaN	2605
3434	NaN	KADAPA	2002	Rabi	NaN	NaN	6
3472	NaN	KADAPA	2003	Kharif	NaN	NaN	8997
3490	NaN	KADAPA	2003	Rabi	NaN	NaN	23
3526	NaN	KADAPA	2004	Kharif	NaN	NaN	4809
3544	NaN	KADAPA	2004	Rabi	NaN	NaN	11
3570	NaN	KADAPA	2005	Kharif	NaN	NaN	4377
30822	NaN	NaN	NaN	NaN	Arhar/Tur	403	965
30843	NaN	NaN	NaN	NaN	Arhar/Tur	345	689
30864	NaN	NaN	NaN	NaN	Arhar/Tur	1017	1665
30894	NaN	NaN	NaN	NaN	Arhar/Tur	959	1106
30922	NaN	NaN	NaN	NaN	Arhar/Tur	872	901
30953	NaN	NaN	NaN	NaN	Arhar/Tur	907	1248
30983	NaN	NaN	NaN	NaN	Arhar/Tur	881	537
31011	NaN	NaN	NaN	NaN	Arhar/Tur	371	467
31046	NaN	NaN	NaN	NaN	Arhar/Tur	419	194
31078	NaN	NaN	NaN	NaN	Arhar/Tur	273	321
31114	NaN	NaN	NaN	NaN	Arhar/Tur	331	276
31146	NaN	NaN	NaN	NaN	Arhar/Tur	356	265
31167	NaN	NaN	NaN	NaN	Arhar/Tur	191	187
231669	NaN	NaN	NaN	NaN	NaN	7	4
231710	NaN	NaN	NaN	NaN	NaN	1	1
231733	NaN	NaN	NaN	NaN	NaN	5	4
231759	NaN	NaN	NaN	NaN	NaN	5	4
245743	NaN	NaN	NaN	NaN	NaN	1253	1020
245779	NaN	NaN	NaN	NaN	NaN	1234	1024
245819	NaN	NaN	NaN	NaN	NaN	415	308

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
245861	NaN	NaN	NaN	NaN	NaN	144	63
245902	NaN	NaN	NaN	NaN	NaN	107	73
245938	NaN	NaN	NaN	NaN	NaN	105	95

- We will fill the null value of State with most occured value i.e "Uttar Pradesh"

In [ ]:

In [27]:
```
crop_prod.State_Name.isnull().sum()
```

Out[27]:
```
38
```

In [28]:
```
crop_prod["State_Name"].fillna("Uttar Pradesh", inplace=True)
```

In [29]:
```
crop_prod.State_Name.isnull().sum()
```

Out[29]:
```
0
```

In [30]:
```
crop_prod.isnull().sum()
```

Out[30]:
```
State_Name          0
District_Name     220
Crop_Year          99
Season             92
Crop               85
Area              511
Production       3751
dtype: int64
```

District Name

In [31]:
```
crop_prod.District_Name.value_counts().sort_values(ascending=False)
```

Out[31]:
```
BIJAPUR                945
TUMKUR                 936
BELGAUM                925
HASSAN                 895
BELLARY                887
DAVANGERE              886
AURANGABAD             878
HAVERI                 870
CHAMARAJANAGAR         844
CHITRADURGA            840
GULBARGA               833
MYSORE                 832
KURNOOL                827
DHARWAD                825
SHIMOGA                825
KADAPA                 824
CHIKMAGALUR            820
BILASPUR               815
BANGALORE RURAL        794
```

In [32]:
```
crop_prod[crop_prod["District_Name"].isnull()]
```

Out[32]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
5488	Andhra Pradesh	NaN	2014	Rabi	Arhar/Tur	NaN	212
5520	Andhra Pradesh	NaN	1997	Kharif	Arhar/Tur	NaN	2200
5549	Andhra Pradesh	NaN	1998	Kharif	Arhar/Tur	NaN	29900
5564	Andhra Pradesh	NaN	1998	Rabi	Arhar/Tur	NaN	500
5612	Andhra Pradesh	NaN	2000	Kharif	Arhar/Tur	NaN	60198
5635	Andhra Pradesh	NaN	2000	Rabi	Arhar/Tur	NaN	265
5663	Andhra Pradesh	NaN	2001	Kharif	Arhar/Tur	NaN	29139
5682	Andhra Pradesh	NaN	2001	Rabi	Arhar/Tur	NaN	321
5710	Andhra Pradesh	NaN	2002	Kharif	Arhar/Tur	NaN	12271
5729	Andhra Pradesh	NaN	2002	Rabi	Arhar/Tur	NaN	146
5769	Andhra Pradesh	NaN	2003	Kharif	Arhar/Tur	NaN	27125

- We will fill the District Name according to the State Name.
- For each state will first find the maximum occured state and will fill accordingly.

In [33]:
```
#List of State where District is null

state_name=list(crop_prod["State_Name"][crop_prod["District_Name"].isnull()].u
state_name
```

Out[33]:
```
['Andhra Pradesh', 'Assam', 'Bihar', 'Uttar Pradesh']
```

In [34]:
```
for i in state_name:
    crop_prod["District_Name"].fillna(crop_prod[crop_prod["State_Name"]==i]["D
```

In [ ]:
```
```

Crop_Year

In [35]:
```
crop_prod.Crop_Year.isnull().sum()
```

Out[35]: 99

In [36]:
```
crop_prod.Crop_Year.value_counts().sort_values(ascending=False).keys()[0]
```

Out[36]: 2003.0

In [37]:
```
#crop_prod[crop_prod.isnull()]
```

In [38]:
```
crop_prod[crop_prod["Crop_Year"].isnull()]
```

Out[38]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
615	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	37000	17000
635	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	34067	13150
671	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	20269	5919
688	Andhra Pradesh	ANANTAPUR	NaN	Rabi	Arhar/Tur	316	92
711	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	66013	11156
751	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	51728	4293
771	Andhra Pradesh	ANANTAPUR	NaN	Rabi	Arhar/Tur	17	1
799	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	56586	8035
857	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	49932	9287
927	Andhra Pradesh	ANANTAPUR	NaN	Kharif	Arhar/Tur	37562	4845
991	Andhra Pradesh	CHITTOOR	NaN	Kharif	Arhar/Tur	6100	900

In [39]:
```
#To fill the value of Crop_Year we will use maximum occured value i.e 2003.

crop_prod["Crop_Year"].fillna(crop_prod.Crop_Year.value_counts().sort_values(a
```

In [40]:
```
crop_prod["Crop_Year"].isnull().sum()
```

Out[40]: 0

In [ ]:

Season

In [41]:
```
crop_prod["Season"].isnull().sum()
```

Out[41]:
```
92
```

In [42]:
```
crop_prod["Season"].value_counts()
```

Out[42]:
```
Kharif         95873
Rabi           66973
Whole Year     57305
Summer         14841
Winter          6058
Autumn          4949
Name: Season, dtype: int64
```

In [43]:
```
crop_prod["Season"].value_counts().sort_values(ascending=False).keys()[0]
```

Out[43]:
```
'Kharif     '
```

In [ ]:

- we will fill the null values with the maximumn occured value i.e 'Kharif' as Season is a Categorical Column.

In [44]:
```
crop_prod["Season"].fillna(crop_prod["Season"].value_counts().sort_values(asce
```

In [45]:
```
crop_prod["Season"].isnull().sum()
```

Out[45]:
```
0
```

In [ ]:

In [46]:
```
crop_prod.isnull().sum()
```

Out[46]:
```
State_Name          0
District_Name       0
Crop_Year           0
Season              0
Crop               85
Area              511
Production       3751
dtype: int64
```

Crop

In [47]:
```
crop_prod.Crop.value_counts()
```

Out[47]:
```
Rice                    15104
Maize                   13947
Moong(Green Gram)       10318
Urad                     9850
Sesamum                  9046
Groundnut                8834
Sugarcane                7921
Wheat                    7899
Rapeseed &Mustard        7592
Arhar/Tur                7493
Gram                     7361
Jowar                    7065
Onion                    7012
Potato                   6931
Dry chillies             6489
Sunflower                5571
Bajra                    5427
Small millets            4652
Peas & beans (Pulses)    4524
```

In [ ]:
```

```

In [48]:
```
crop_prod[crop_prod["Crop"].isnull()]
```

Out[48]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
3416	Uttar Pradesh	KADAPA	2002	Kharif	NaN	NaN	2605
3434	Uttar Pradesh	KADAPA	2002	Rabi	NaN	NaN	6
3472	Uttar Pradesh	KADAPA	2003	Kharif	NaN	NaN	8997
3490	Uttar Pradesh	KADAPA	2003	Rabi	NaN	NaN	23
3526	Uttar Pradesh	KADAPA	2004	Kharif	NaN	NaN	4809
3544	Uttar Pradesh	KADAPA	2004	Rabi	NaN	NaN	11
3570	Uttar Pradesh	KADAPA	2005	Kharif	NaN	NaN	4377
3612	Andhra Pradesh	KADAPA	2006	Kharif	NaN	NaN	1527
3628	Andhra Pradesh	KADAPA	2006	Rabi	NaN	NaN	1
3654	Andhra Pradesh	KADAPA	2007	Kharif	NaN	NaN	4000
3673	Andhra Pradesh	KADAPA	2008	Kharif	NaN	NaN	839

- We will fill the value of the crops with help of the seasons column.

In [49]:
```
seasons_c=list(crop_prod["Season"][crop_prod["Crop"].isnull()].unique())
seasons_c
```

Out[49]:
```
['Kharif     ', 'Rabi       ']
```

In [50]:
```
for i in seasons_c:
    crop_prod["Crop"].fillna(crop_prod[crop_prod["Season"]==i]["Crop"].value_c
```

In [51]:
```
crop_prod.isnull().sum()
```

Out[51]:
```
State_Name          0
District_Name       0
Crop_Year           0
Season              0
Crop                0
Area              511
Production       3751
dtype: int64
```

In [ ]:

In [ ]:

Area

In [52]:
```
crop_prod.Area.mean()
```

Out[52]:
```
12016.5708777995
```

In [53]:
```
crop_prod[crop_prod.Area.isnull()]
```

Out[53]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
1412	Andhra Pradesh	CHITTOOR	2003	Kharif	Arhar/Tur	NaN	1000
1433	Andhra Pradesh	CHITTOOR	2003	Kharif	Arhar/Tur	NaN	2554
1473	Andhra Pradesh	CHITTOOR	2009	Kharif	Arhar/Tur	NaN	1377
1490	Andhra Pradesh	CHITTOOR	2009	Rabi	Arhar/Tur	NaN	56
1514	Andhra Pradesh	CHITTOOR	2010	Kharif	Arhar/Tur	NaN	1182
1531	Andhra Pradesh	CHITTOOR	2010	Rabi	Arhar/Tur	NaN	1
1555	Andhra Pradesh	CHITTOOR	2011	Kharif	Arhar/Tur	NaN	1203
1572	Andhra Pradesh	CHITTOOR	2011	Rabi	Arhar/Tur	NaN	2
1599	Andhra Pradesh	CHITTOOR	2012	Kharif	Arhar/Tur	NaN	1268
1632	Andhra Pradesh	CHITTOOR	2012	Rabi	Arhar/Tur	NaN	2
1657	Andhra Pradesh	CHITTOOR	2013	Kharif	Arhar/Tur	NaN	1923

In [54]:
```
crop_prod.groupby(by="District_Name").median()
```

Out[54]:

District_Name	Crop_Year	Area	Production
24 PARAGANAS NORTH	2005	1454	3296
24 PARAGANAS SOUTH	2005	1188	1872
ADILABAD	2004	2042	1453
AGAR MALWA	2013	733	608
AGRA	2006	251	403
AHMADABAD	2004	2000	2500
AHMEDNAGAR	2005	5700	3400
AIZAWL	2002	230	285
AJMER	2003	1634	764
AKOLA	2005	2300	2102

In [55]:
```
crop_prod.groupby(by="District_Name").mean()
```

Out[55]:

District_Name	Crop_Year	Area	Production
24 PARAGANAS NORTH	2005	27126	351491
24 PARAGANAS SOUTH	2005	21596	265246
ADILABAD	2005	14856	22974
AGAR MALWA	2013	9859	9964
AGRA	2006	15147	72097
AHMADABAD	2004	21699	35552
AHMEDNAGAR	2005	39399	234733
AIZAWL	2003	1712	3310
AJMER	2004	15965	10875
AKOLA	2005	27854	31656

In [56]:
```
crop_prod["Area"].fillna(crop_prod.groupby("District_Name")["Area"].transform(
```

In [57]:
```
crop_prod["Area"].isnull().sum()
```

Out[57]:
```
0
```

In [ ]:
```

```

Production

In [58]:
```
crop_prod["Production"].isnull().sum()
```

Out[58]:
```
3751
```

In [ ]:
```

```

- We will fill the production column null value with average production of each crop.

In [59]:
```
crop_prod.groupby(by="Crop").mean()
```

Out[59]:

Crop	Crop_Year	Area	Production
Apple	2002	2	0
Arcanut (Processed)	2002	7206	9642
Arecanut	2006	3812	13229
Arhar/Tur	2005	8150	5304
Ash Gourd	2003	37	0
Atcanut (Raw)	2002	7206	46362
Bajra	2005	26007	24109
Banana	2006	1636	46643
Barley	2005	2479	5369
Bean	2014	232	312

In [60]:
```
crop_prod["Production"].fillna(crop_prod.groupby("Crop")["Production"].transfo
```

In [61]:
```
crop_prod["Production"].isnull().sum()
```

Out[61]:
```
0
```

In [ ]:
```

```

In [62]:
```
crop_prod.isnull().sum()
```

Out[62]:
```
State_Name       0
District_Name    0
Crop_Year        0
Season           0
Crop             0
Area             0
Production       0
dtype: int64
```

- hence we have removed all the null values from the dataset.

Detect and handle the outlier

In [66]:
```
# Detect the outlier in the data.
crop_prod.describe().T
```

Out[66]:

	count	mean	std	min	25%	50%	75%	max
Crop_Year	246091	2006	5	1997	2002	2006	2010	2015
Area	246091	12020	50520	0	81	589	4443	8580100
Production	246091	581405	16950146	0	91	772	7100	1250800000

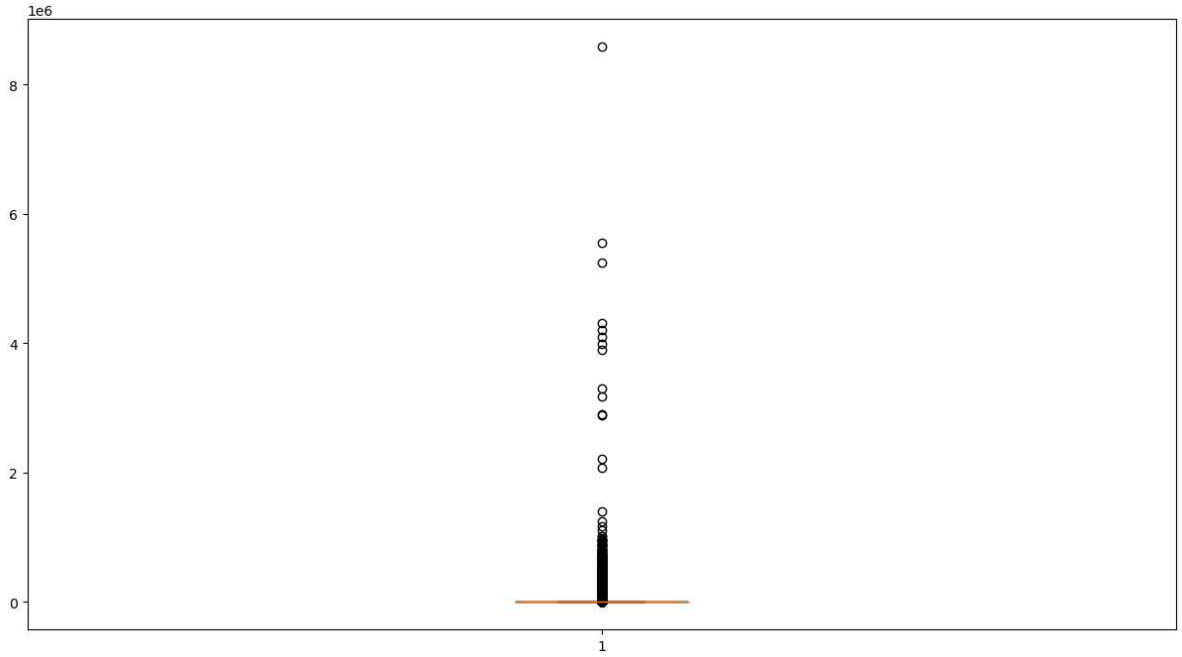
- by looking at the diffrence between the mean and median of the data we can conclude that the data has outlier.
- We will remove the outlier one by one

- We will first detect and seperate the outlier.
- We will create a new dataframe with minimum outlier

In [ ]:
```

```

Area

In [67]:
```
plot=plt.boxplot(crop_prod["Area"])
```



In [68]:
```
wiskers=[i.get_ydata() for i in plot["caps"]]
wiskers
```

Out[68]:
```
[array([0.04, 0.04]), array([10986., 10986.])]
```

- The lower wiskers is equal to 0.04 and upper wiskers is equal to 10986
- Hence the data above 10986 and below 0.04 are outliers in columns "area".

In [69]:
```
crop_new1=crop_prod[(crop_prod['Area']<10986)&(crop_prod["Area"]>0.04)]
```

In [70]:
```
crop_new1.head()
```

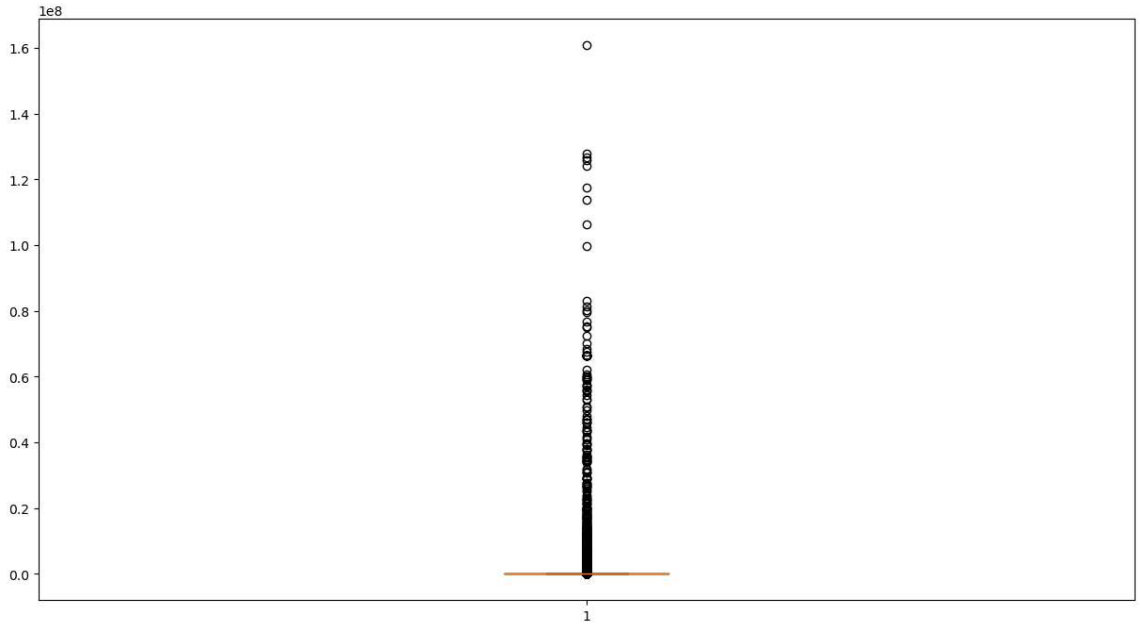
Out[70]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720	165

Production

- We check the outlier of the this column from new data set i.e crop_new1

In [71]:
```
plot1=plt.boxplot(crop_new1["Production"])
```



In [72]:
```
wiskers2=[i.get_ydata() for i in plot1['caps']]
wiskers2
```

Out[72]:
```
[array([0., 0.]), array([5543., 5543.])]
```

In [73]:
```
crop_new_2=crop_new1[(crop_new1["Production"]>0) & (crop_new1["Production"]<55
```

In [74]:
```
crop_new_2.head()
```

Out[74]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720	165

In [ ]:
```

```

In [75]:
```
crop_new_2.shape
```

Out[75]:
```
(172804, 7)
```

- The dataframe 'crop_new_2' is the new dataframe after removing the oultier.

- Further we will perform the analysis on this new dataframe.

In [ ]:

Investigation of wrong datatype

In [77]:
```
# Check and visualize all data present in dataframe and investigate the wrong
crop_new_2.dtypes
```

Out[77]:
```
State_Name        object
District_Name     object
Crop_Year        float64
Season            object
Crop              object
Area             float64
Production       float64
dtype: object
```

In [78]:
```
crop_new_2.Crop_Year.nunique()
```

Out[78]: 19

- The column Crop_Year is a categorical columns as it has only 19 unique values
- Hence we will change the datatype into "object"

In [ ]:

In [79]:
```
# Perform the Datatype Casting, If needed.
crop_new_2.Crop_Year=crop_new_2.Crop_Year.astype("object")
```

In [80]:
```
crop_new_2.Crop_Year.dtype
```

Out[80]: dtype('O')

In [ ]:

Check for the Duplicated in the dataset

In [84]:
```
# Show the duplicates here
crop_new_2.duplicated().sum()
```

Out[84]: 1

In [85]:
```
# Remove the duplicates.
crop_new_2.drop_duplicates(inplace=True)
```

In [86]:
```
crop_new_2.duplicated().sum()
```

Out[86]:
```
0
```

In [ ]:
```

```

Identifying the irrelevent columns in dataset:

In [87]:
```
#First let divide the categorical and numerical columns.
# Then we will check for the irrelevent columns
num_crop=crop_new_2.select_dtypes(np.number)
cat_crop=crop_new_2.select_dtypes(np.object)
```

In [88]:
```
num_crop.head()
```

Out[88]:

	Area	Production
0	1254	2000
1	2	1
2	102	321
3	176	641
4	720	165

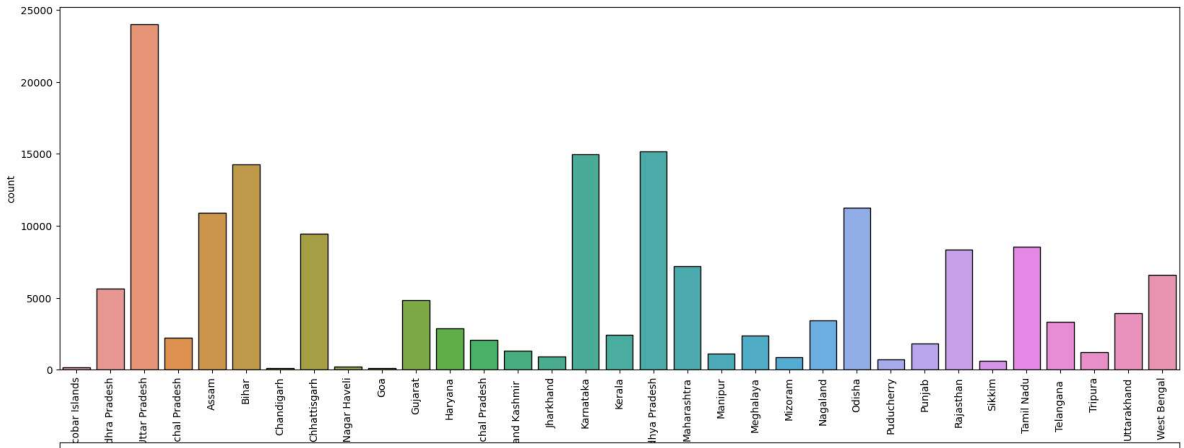
In [89]:
```
cat_crop.head()
```

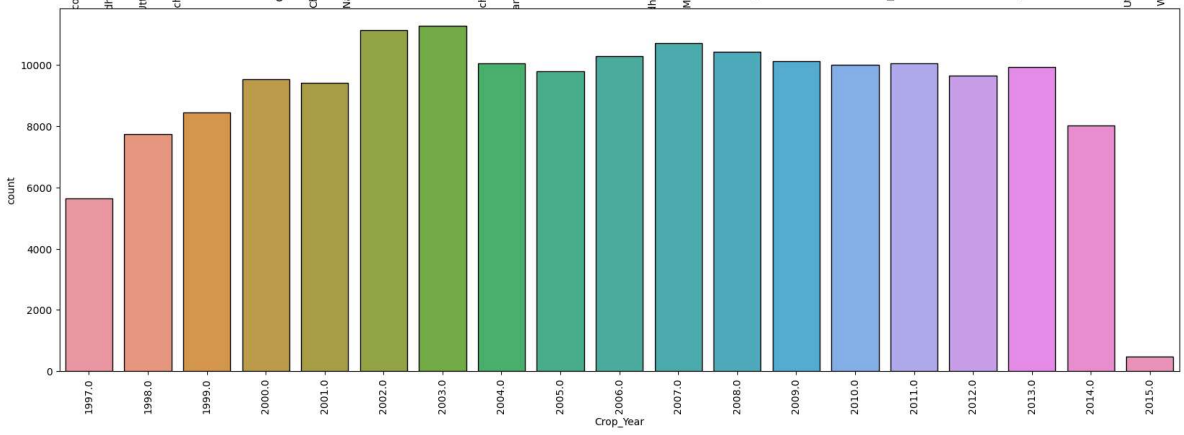
Out[89]:

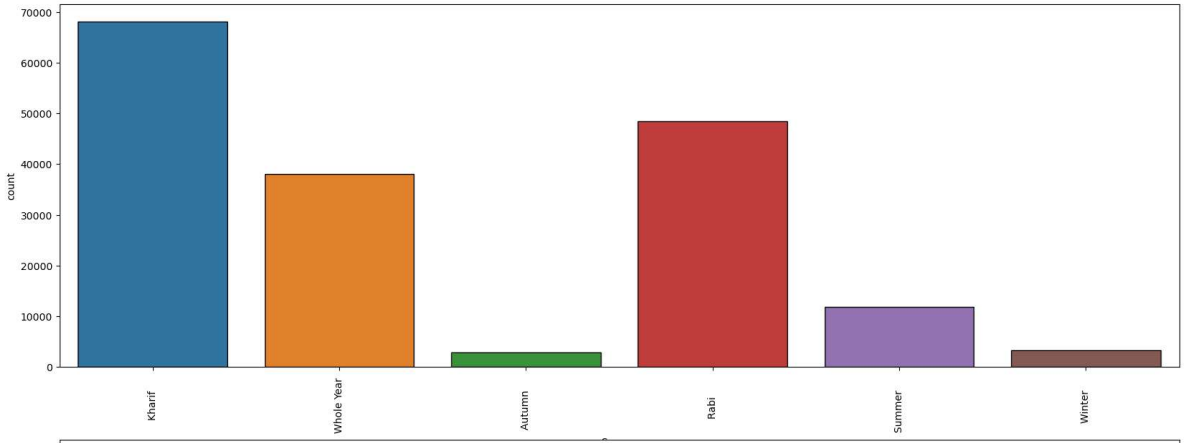
	State_Name	District_Name	Crop_Year	Season	Crop
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut

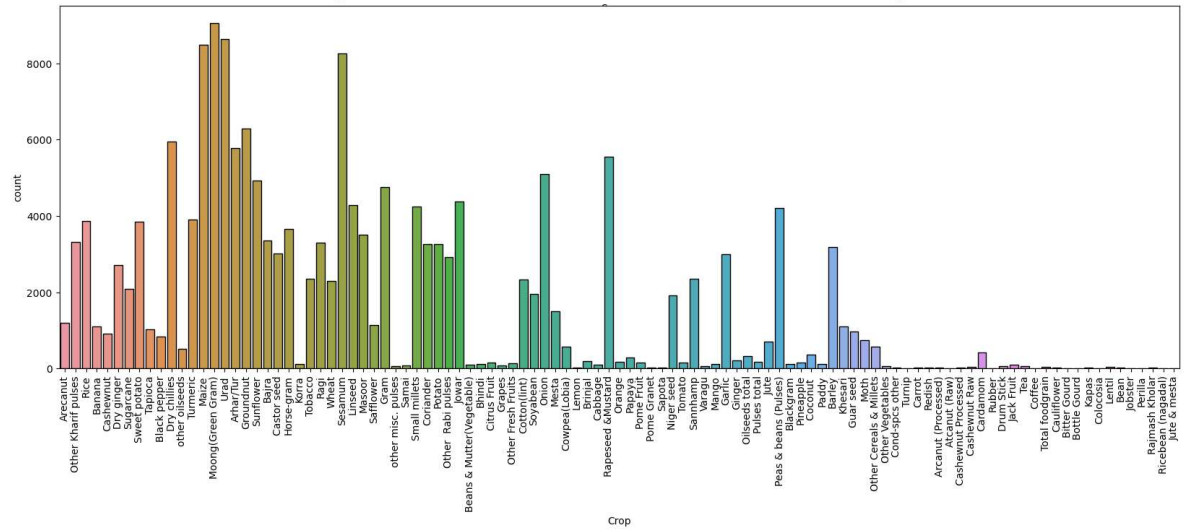
In [90]:
```
# Visualize all columns at a time (Numerical columns and Categorical Columns).

```

In [91]:
```
# Categorical Columns



plt.figure(figsize=(20,30))
num=1
for i in cat_crop.columns:

    if i != "District_Name":

        plt.subplot(4,1,num)

        sns.countplot(data=cat_crop, x=i, edgecolor='black')
        plt.xticks(rotation=90)
        num=num+1
```

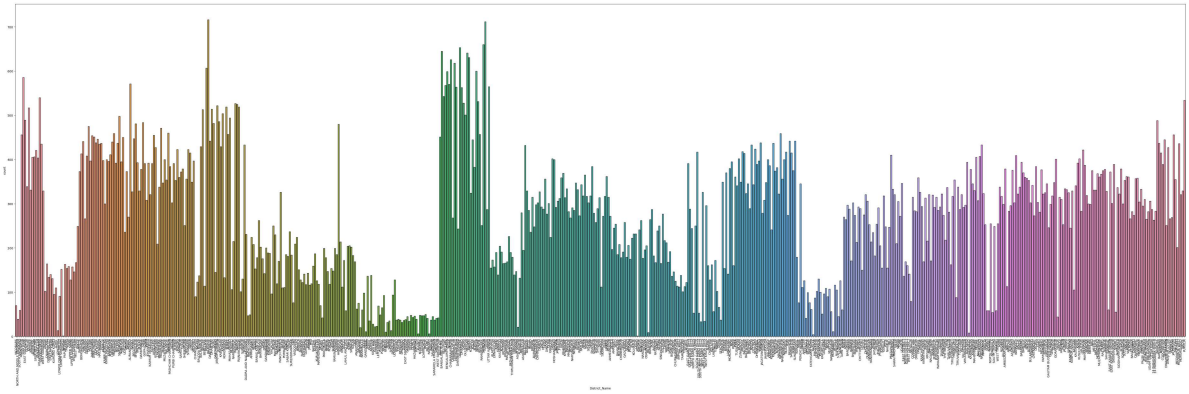

count
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Dadra and Nagar Haveli
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttarakhand
West Bengal


count
Crop_Year


count
Kharif
Whole Year
Autumn
Rabi
Summer
Winter


count
Crop

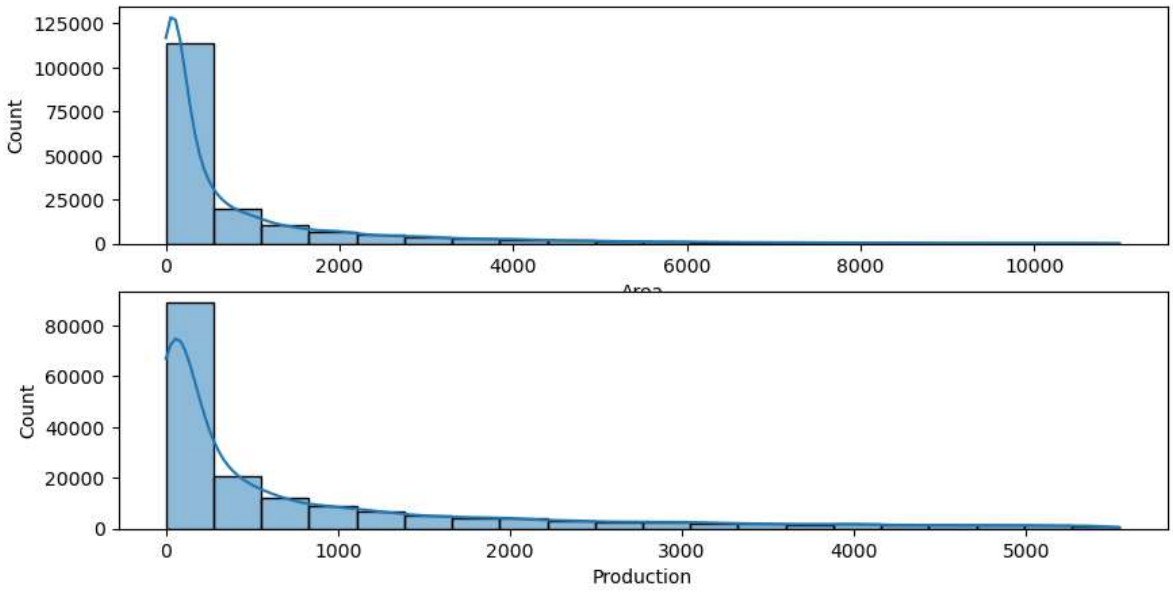
In [92]:
```
plt.figure(figsize=(70,20))

sns.countplot(data=cat_crop, x="District_Name", edgecolor="black")
plt.xticks(rotation=90)
plt.show()
```



- All the Categorical columns looks fine.

In [93]:
```
plt.figure(figsize=(10,5))
plt.subplot(2,1,1)
sns.histplot(data=num_crop, x="Area", kde=True, bins=20)
plt.subplot(2,1,2)
sns.histplot(data=num_crop, x="Production", kde=True, bins=20)
```

Out[93]: <AxesSubplot:xlabel='Production', ylabel='Count'>



- The numerical column-data is highly skeweed even after removing the outlier
- But this two columns are important for our analysis.

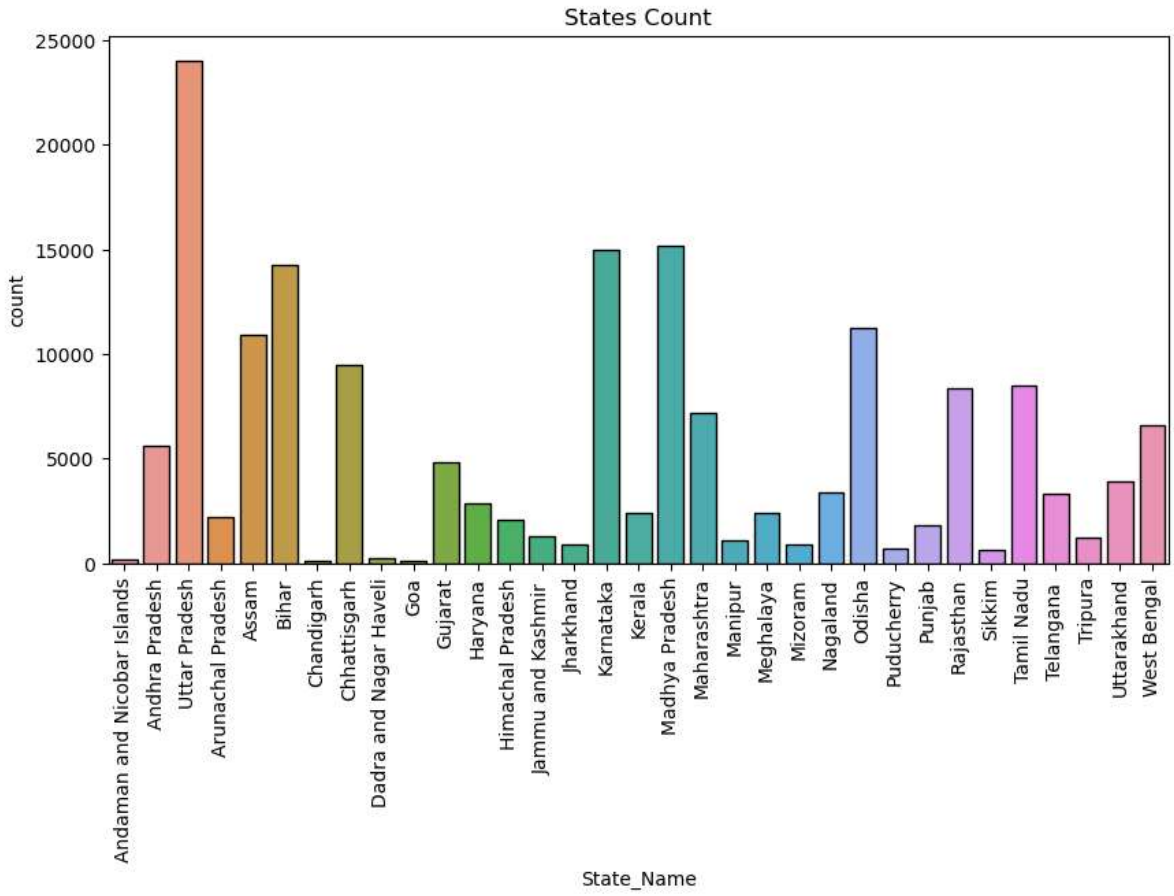
- All the categorical and numerical columns are important for our analysis.
- Hence we will not remove the colums from the dataset

Univarate Analysis.

In [95]:
```
crop_new_2.State_Name.value_counts()
```

Out[95]:
```
Uttar Pradesh                23996
Madhya Pradesh               15170
Karnataka                    14969
Bihar                        14238
Odisha                       11270
Assam                        10895
Chhattisgarh                  9442
Tamil Nadu                    8522
Rajasthan                     8328
Maharashtra                   7179
West Bengal                   6565
Andhra Pradesh                5626
Gujarat                       4830
Uttarakhand                   3919
Nagaland                      3402
Telangana                     3330
Haryana                       2882
Kerala                        2435
Meghalaya                     2385
Arunachal Pradesh             2206
Himachal Pradesh              2067
Punjab                        1798
Jammu and Kashmir             1296
Tripura                       1240
Manipur                       1107
Jharkhand                      921
Mizoram                        883
Puducherry                     711
Sikkim                         606
Dadra and Nagar Haveli         231
Andaman and Nicobar Islands    168
Goa                             96
Chandigarh                      90
Name: State_Name, dtype: int64
```

In [96]:
```
plt.figure(figsize=(10,5))

sns.countplot(data=crop_new_2, x="State_Name", edgecolor="black")

plt.title("States Count")
plt.xticks(rotation=90)
plt.show()
```



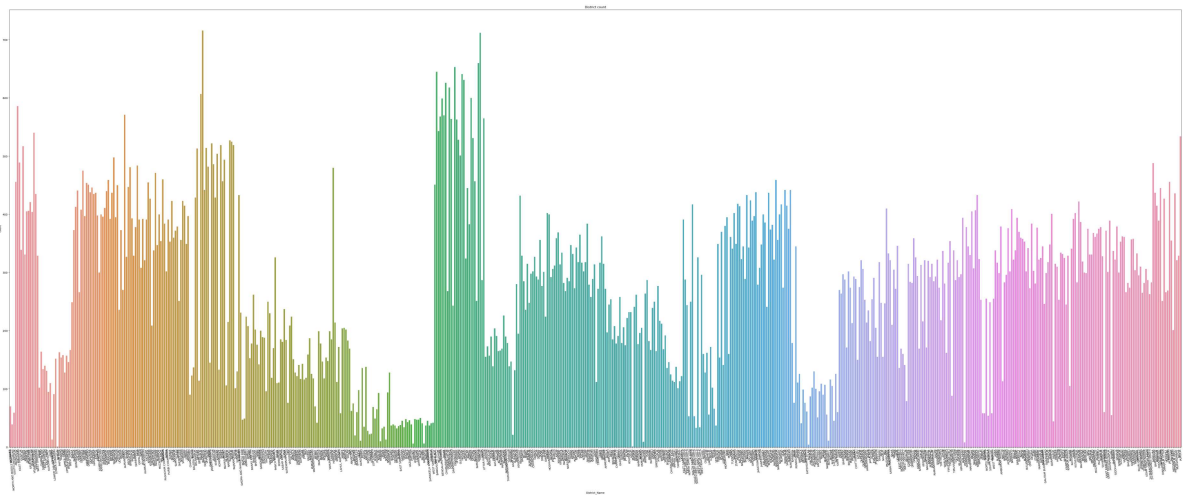
- Uttar Pradesh and Madhay Pradedh as highest count.
- The count for Goa and Chandigarh is lowest.

District_Name

In [97]:
```
crop_new_2.District_Name.value_counts()
```

Out[97]:
```
BILASPUR                716
TUMKUR                  712
SHIMOGA                 660
DAVANGERE               653
BANGALORE RURAL         645
HASSAN                  641
HAVERI                  631
CHAMARAJANAGAR          626
CHIKMAGALUR             618
BIJAPUR                 607
MANDYA                  600
BENGALURU URBAN         599
CHITTOOR                586
AURANGABAD              571
BIDAR                   570
BELLARY                 568
UTTAR KANNAD            565
CHITRADURGA             564
DHARWAD                 563
```

In [98]:
```
plt.figure(figsize=(80,30))


sns.countplot(data=crop_new_2, x="District_Name")
plt.title("District count", fontdict={"size":12, "color":"black"})
plt.xticks(rotation=95)
plt.show()
```



- The above graph does not gives any clear picture.
- Hence to get a clear picture we will look at top 10 and bottom 10 district count.

In [99]:
```
count_dict=crop_new_2.District_Name.value_counts()

count_dict
```

Out[99]:
```
BILASPUR                716
TUMKUR                  712
SHIMOGA                 660
DAVANGERE               653
BANGALORE RURAL         645
HASSAN                  641
HAVERI                  631
CHAMARAJANAGAR          626
CHIKMAGALUR             618
BIJAPUR                 607
MANDYA                  600
BENGALURU URBAN         599
CHITTOOR                586
AURANGABAD              571
BIDAR                   570
BELLARY                 568
UTTAR KANNAD            565
CHITRADURGA             564
DHARWAD                 563
```

In [100]:
```
top10=count_dict[:11]
top10
```

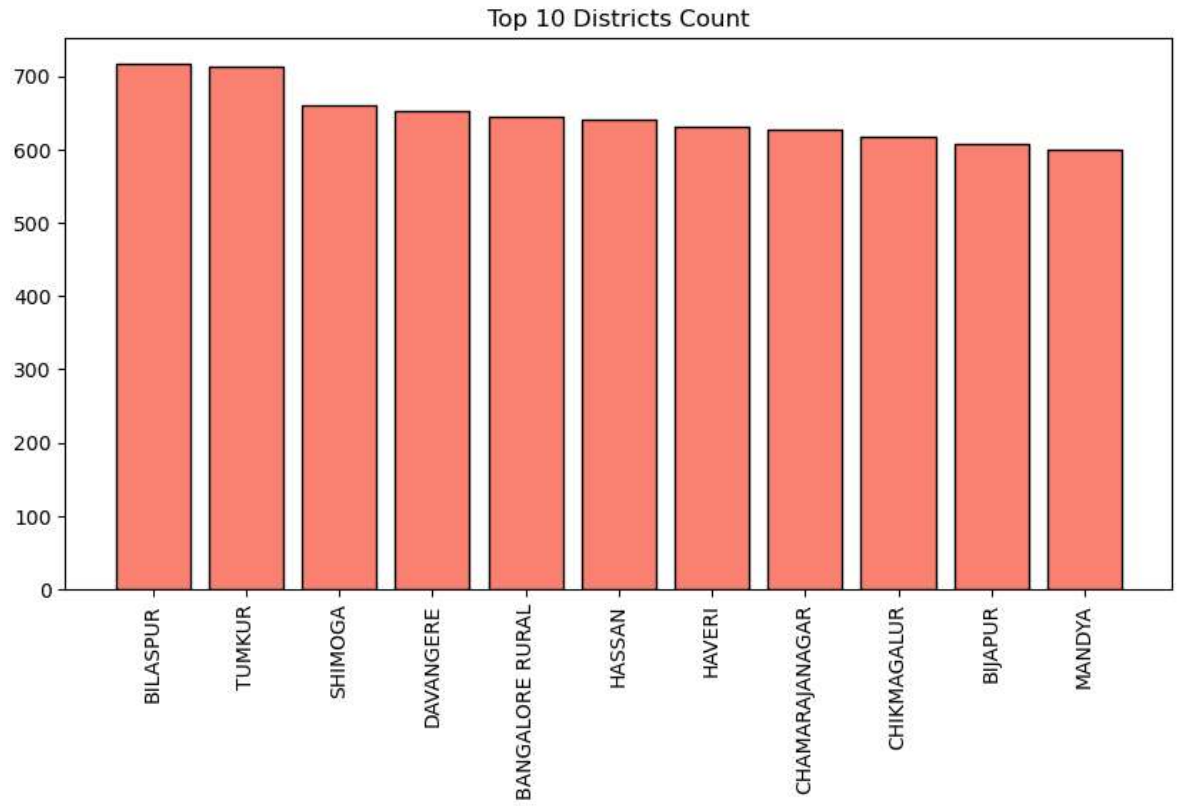
Out[100]:
```
BILASPUR           716
TUMKUR             712
SHIMOGA            660
DAVANGERE          653
BANGALORE RURAL    645
HASSAN             641
HAVERI             631
CHAMARAJANAGAR     626
CHIKMAGALUR        618
BIJAPUR            607
MANDYA             600
Name: District_Name, dtype: int64
```

In [101]:
```
bottom10=count_dict[-11:-1]
bottom10
```

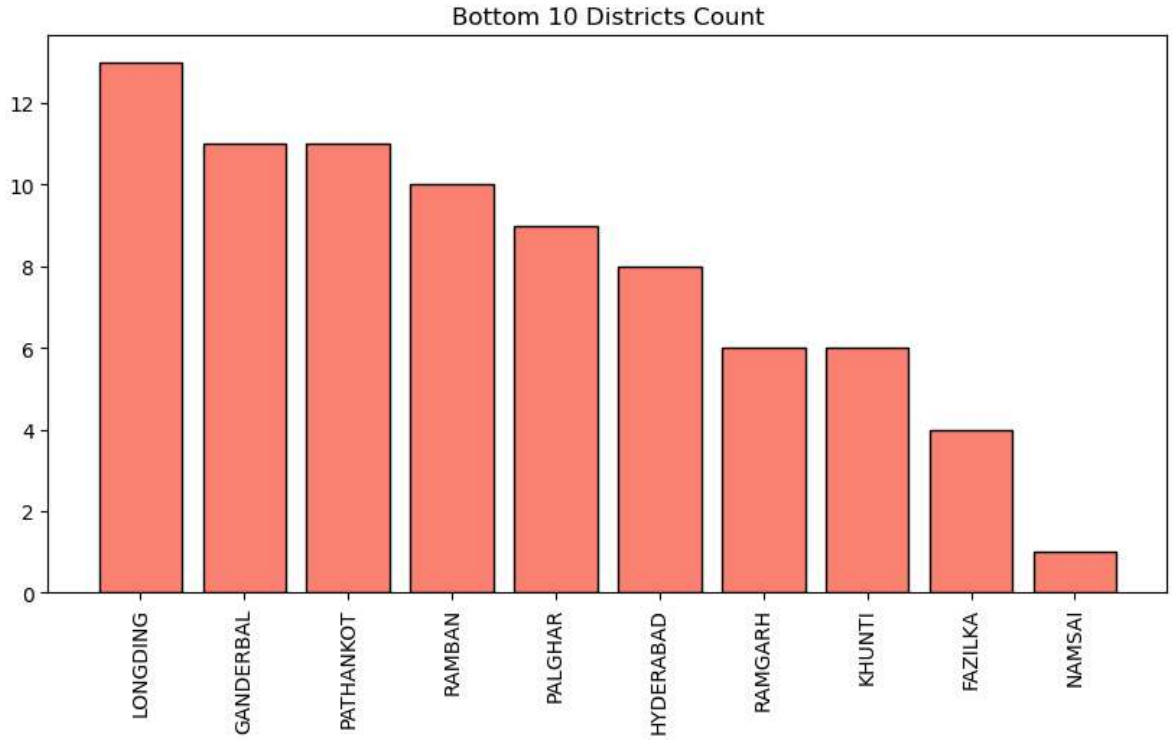
Out[101]:
```
LONGDING     13
GANDERBAL    11
PATHANKOT    11
RAMBAN       10
PALGHAR       9
HYDERABAD     8
RAMGARH       6
KHUNTI        6
FAZILKA       4
NAMSAI        1
Name: District_Name, dtype: int64
```

In [ ]:

In [102]:
```
#For Top10 district:
plt.figure(figsize=(10,5))
plt.bar(top10.keys(), top10.values, color="salmon", edgecolor="black")

plt.title("Top 10 Districts Count", fontdict={"size":12})
plt.xticks(rotation=90)
plt.show()
```



In [103]:
```
#For Bottom10 district:
plt.figure(figsize=(10,5))
plt.bar(bottom10.keys(), bottom10.values, color="salmon", edgecolor="black")

plt.title("Bottom 10 Districts Count", fontdict={"size":12})
plt.xticks(rotation=90)
plt.show()
```



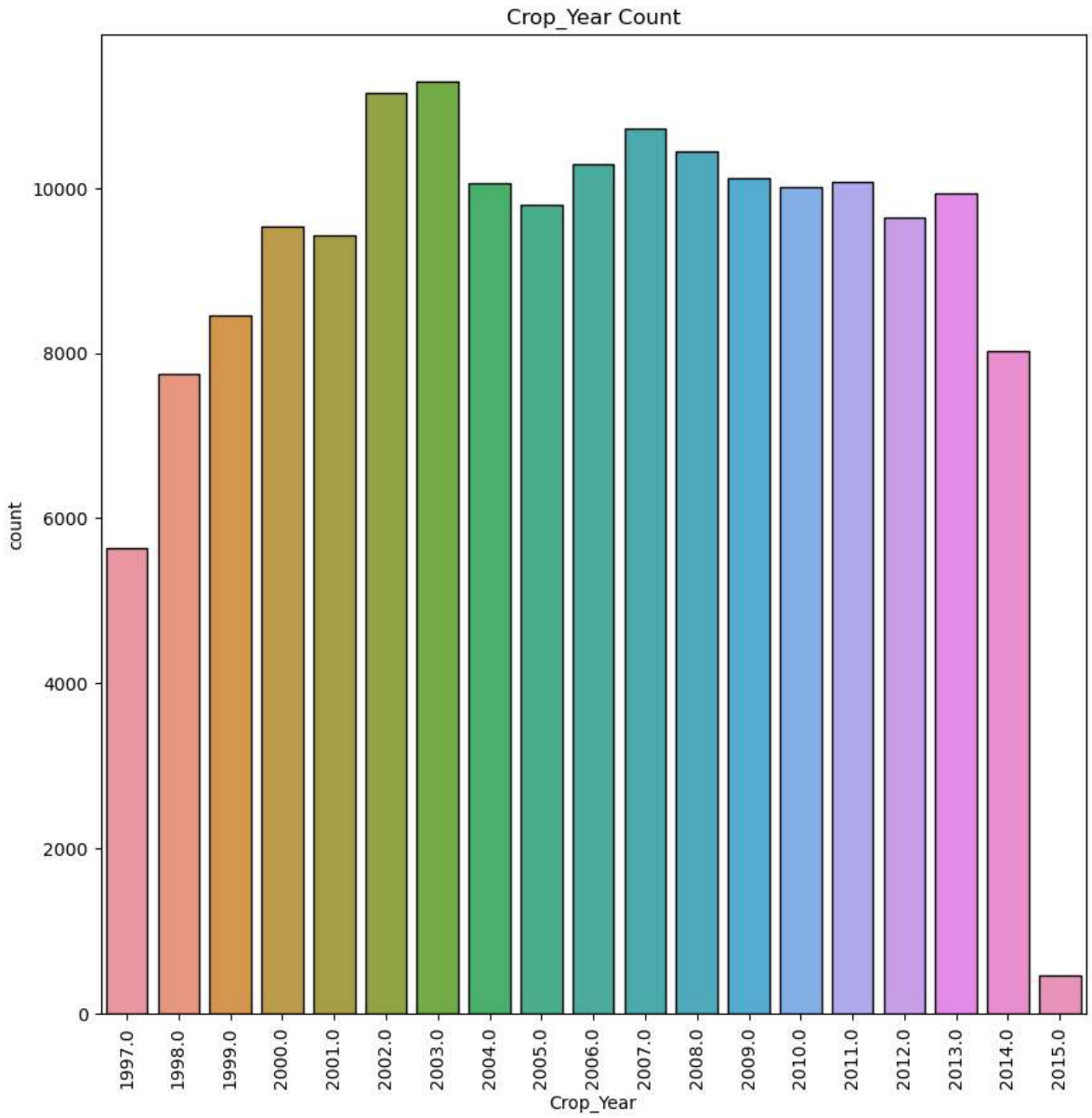
- The District Bilaspur has maximum count, i.e it has maximum crop production.
- Namsai has lowest count, i.e it has lowest crop production.

Crop_Year

In [104]:
```
crop_new_2.Crop_Year.value_counts()
```

Out[104]:
```
2003    11295
2002    11146
2007    10721
2008    10441
2006    10288
2009    10119
2011    10066
2004    10051
2010    10004
2013     9939
2005     9796
2012     9647
2000     9534
2001     9420
1999     8455
2014     8030
1998     7748
1997     5636
2015      467
Name: Crop_Year, dtype: int64
```

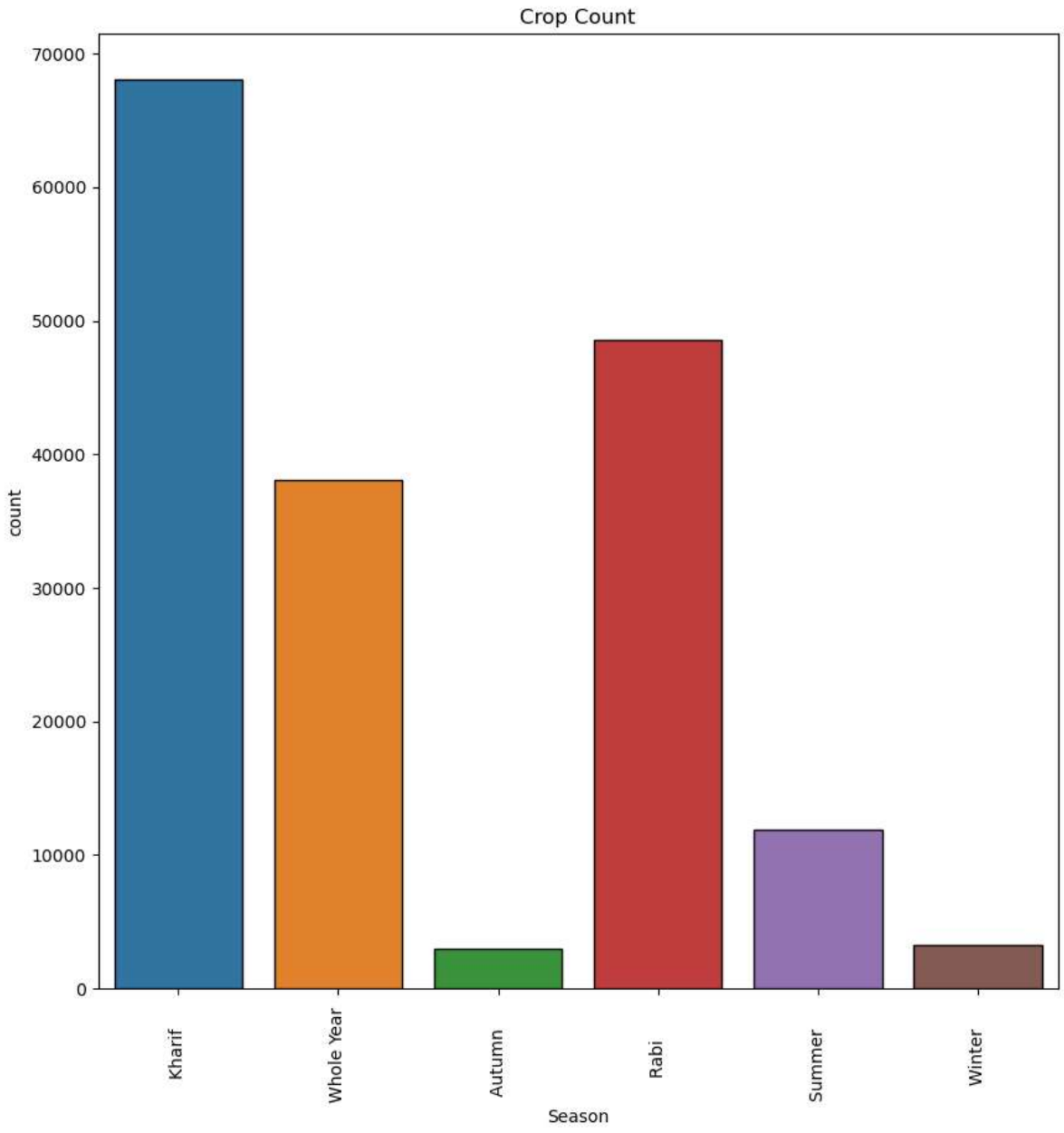
In [105]:
```
plt.figure(figsize=(10,10))

sns.countplot(data=crop_new_2, x="Crop_Year", edgecolor="black")
plt.title("Crop_Year Count")

plt.xticks(rotation=90)

plt.show()
```



- The Year 2003 and 2004 has maximum count
- The Year 2015 have lowest count.

Season

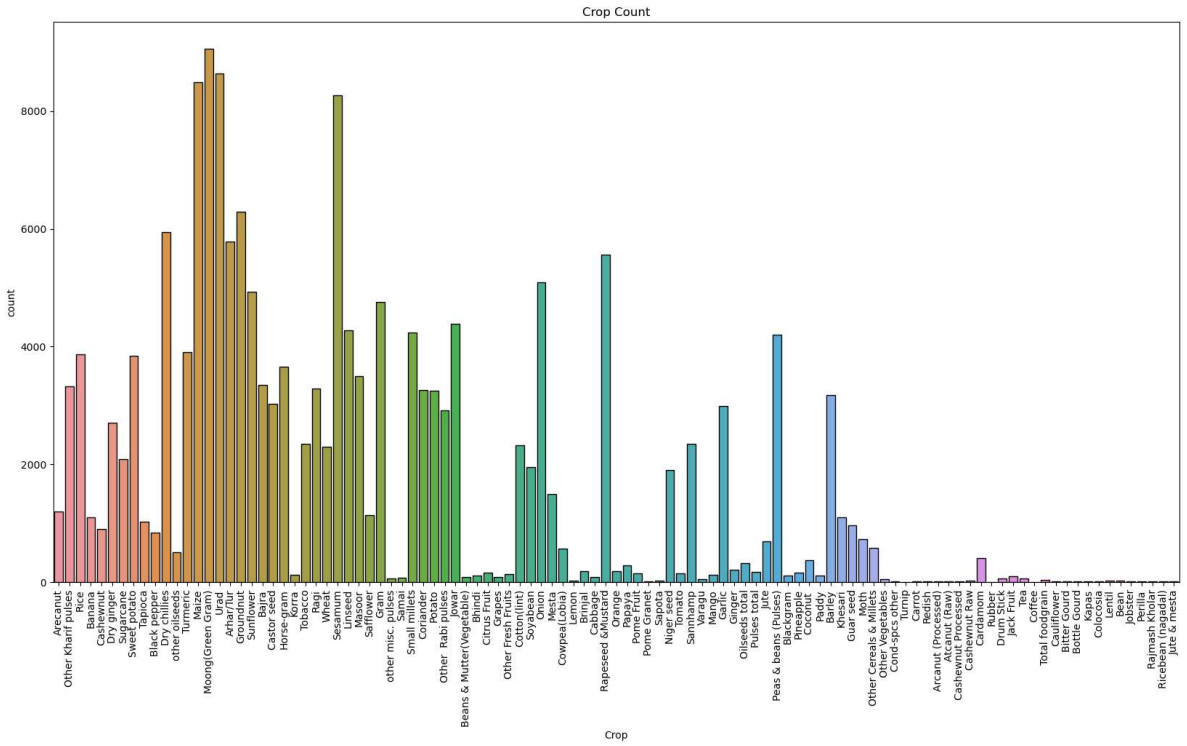
In [106]:
```
plt.figure(figsize=(10,10))

sns.countplot(data=crop_new_2, x="Season", edgecolor="black")
plt.title("Crop Count")

plt.xticks(rotation=90)

plt.show()
```



- The Kharif Season have maximum count.
- The Autumn Season has minimum count.

Crop

In [107]:
```
plt.figure(figsize=(20,10))

sns.countplot(data=crop_new_2, x="Crop", edgecolor="black")
plt.title("Crop Count")

plt.xticks(rotation=90)

plt.show()
```



- Here also to get clear picture we will find top 10 and bottom 10 crop count

In [ ]:
```
```

In [108]:
```
top10_crop=crop_new_2.Crop.value_counts()[0:11]

bottom_crop=crop_new_2.Crop.value_counts()[-11:-1]
```

In [109]:
```
top10_crop
```

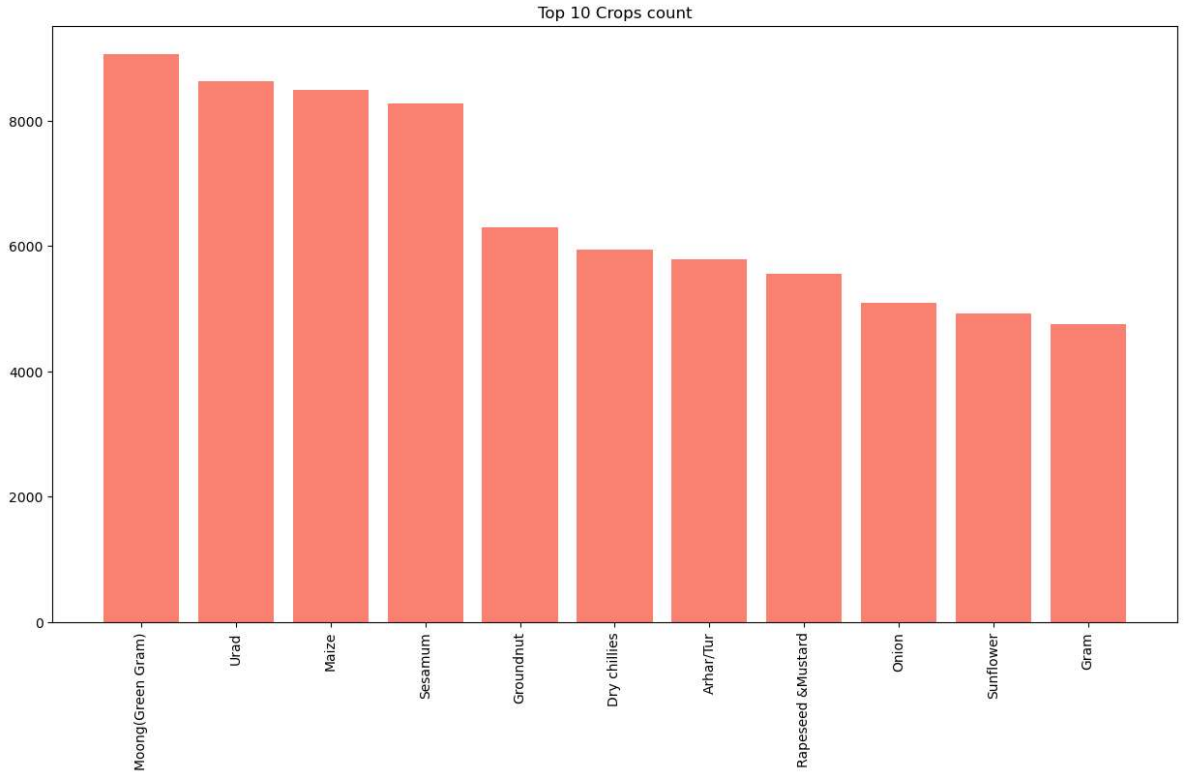
Out[109]:
```
Moong(Green Gram)    9062
Urad                 8636
Maize                8496
Sesamum              8271
Groundnut            6295
Dry chillies         5946
Arhar/Tur            5782
Rapeseed &Mustard    5557
Onion                5095
Sunflower            4929
Gram                 4760
Name: Crop, dtype: int64
```

In [110]:
```
bottom_crop
```

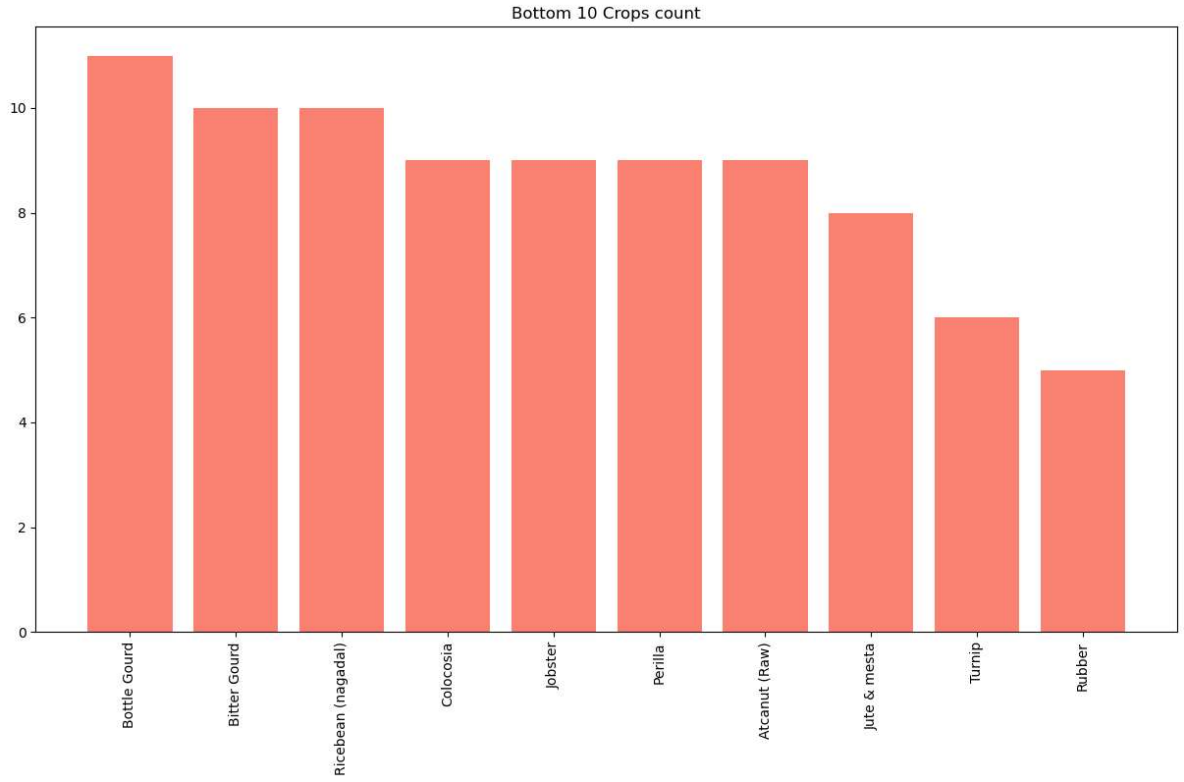
Out[110]:
```
Bottle Gourd          11
Bitter Gourd          10
Ricebean (nagadal)    10
Colocosia              9
Jobster                9
Perilla                9
Atcanut (Raw)          9
Jute & mesta           8
Turnip                 6
Rubber                 5
Name: Crop, dtype: int64
```

In [ ]:

In [111]:
```
#Top 10 Crops:
plt.bar(top10_crop.keys(), top10_crop.values, color="salmon")
plt.title("Top 10 Crops count", fontdict={"size":12})
plt.xticks(rotation=90)
plt.show()
```



In [112]:
```
#Top 10 Crops:
plt.bar(bottom_crop.keys(), bottom_crop.values, color="salmon")
plt.title("Bottom 10 Crops count", fontdict={"size":12})
plt.xticks(rotation=90)
plt.show()
```



- Moong(Green Grams) has maximum count.
- Rubber has lowest count.

In [ ]:
```
```

In [113]:
```
crop_new_2.head()
```

Out[113]:

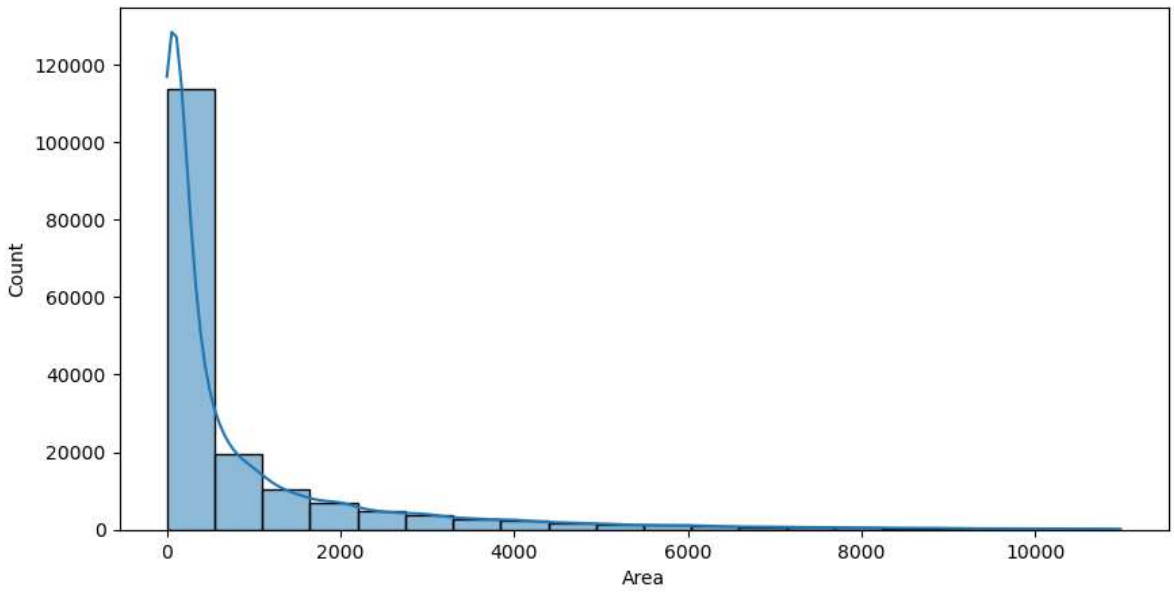
	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720	165

In [ ]:

Area

In [114]:
```
plt.figure(figsize=(10,5))

sns.histplot(data=crop_new_2, x="Area", kde=True, bins=20)
```

Out[114]: <AxesSubplot:xlabel='Area', ylabel='Count'>



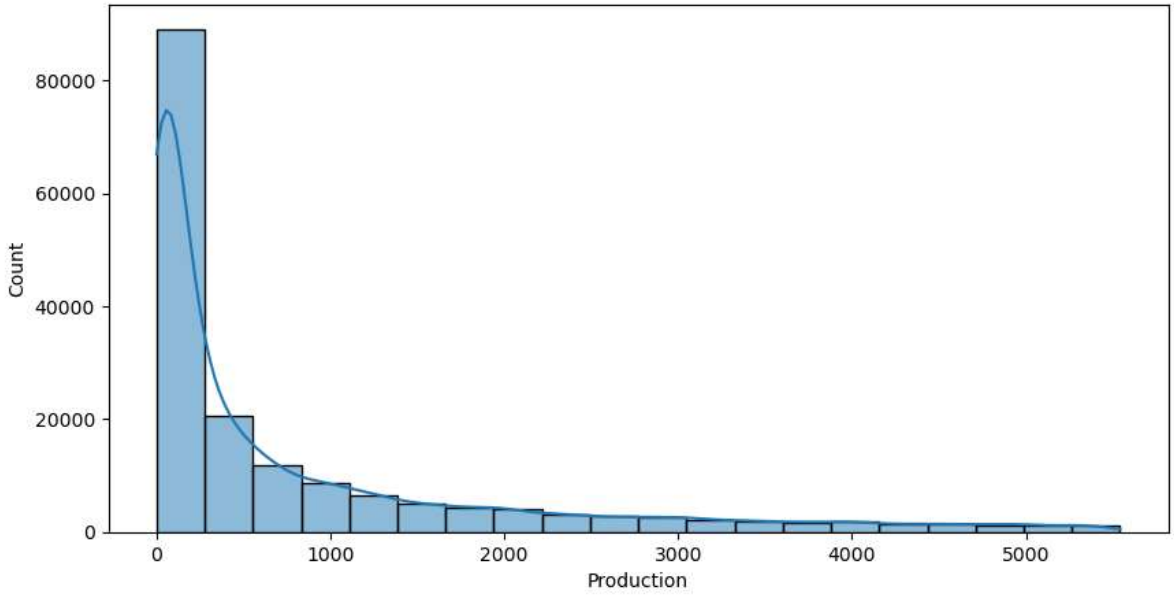
- The data is highly skweed .
- This is maybe beacuse that every state has varying number of agricultural land.

In [ ]:

Production

In [115]:
```
plt.figure(figsize=(10,5))

sns.histplot(data=crop_new_2, x="Production", kde=True, bins=20)
```

Out[115]:
```
<AxesSubplot:xlabel='Production', ylabel='Count'>
```



- The data is highly skweed .
- This is maybe beacuse that every state has varying number of agricultural land.
- Every State produce diffrent crops in abundance.

In [ ]:
```
```

Bivariate Analysis

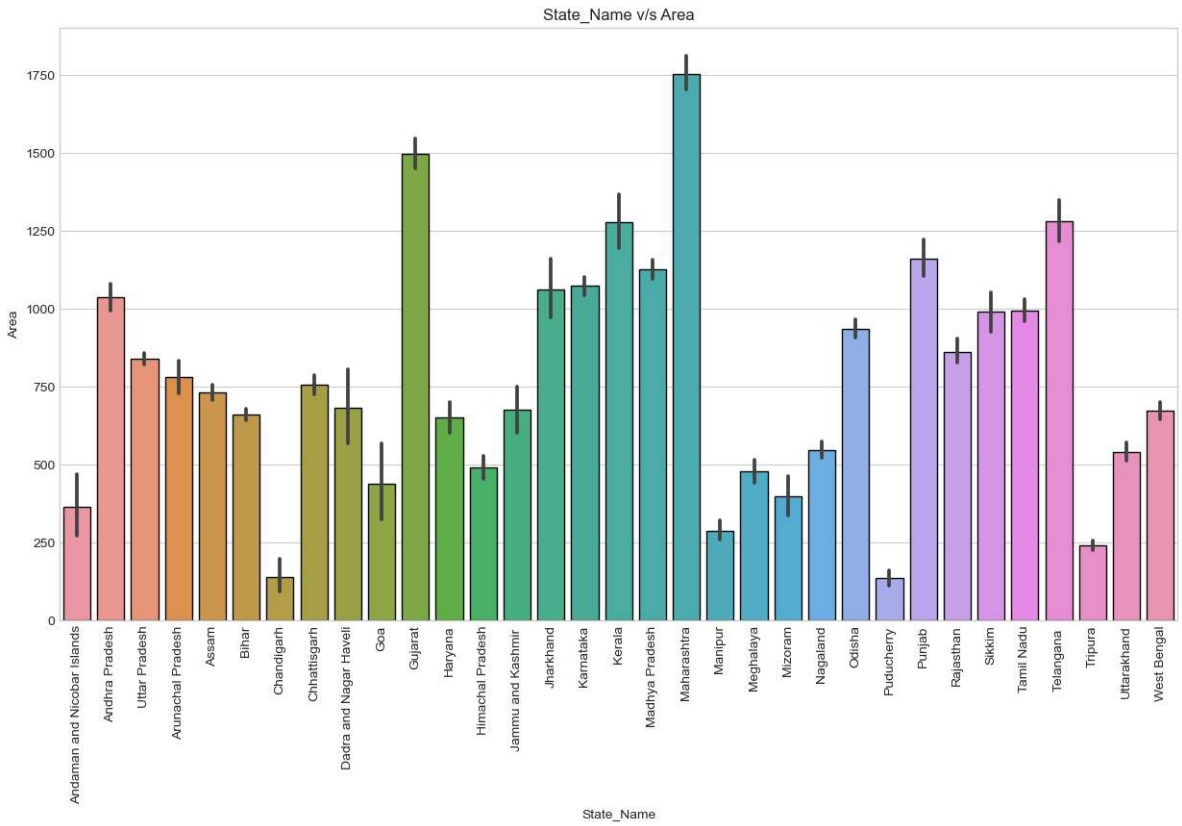
In [116]:
```
crop_new_2.head()
```

Out[116]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720	165

States_Name v/s Area

In [117]:
```
plt.style.use("seaborn-whitegrid")
sns.barplot(data=crop_new_2, x='State_Name', y='Area', edgecolor="black")
plt.title("State_Name v/s Area")
plt.xticks(rotation= 90 )
plt.show()
```



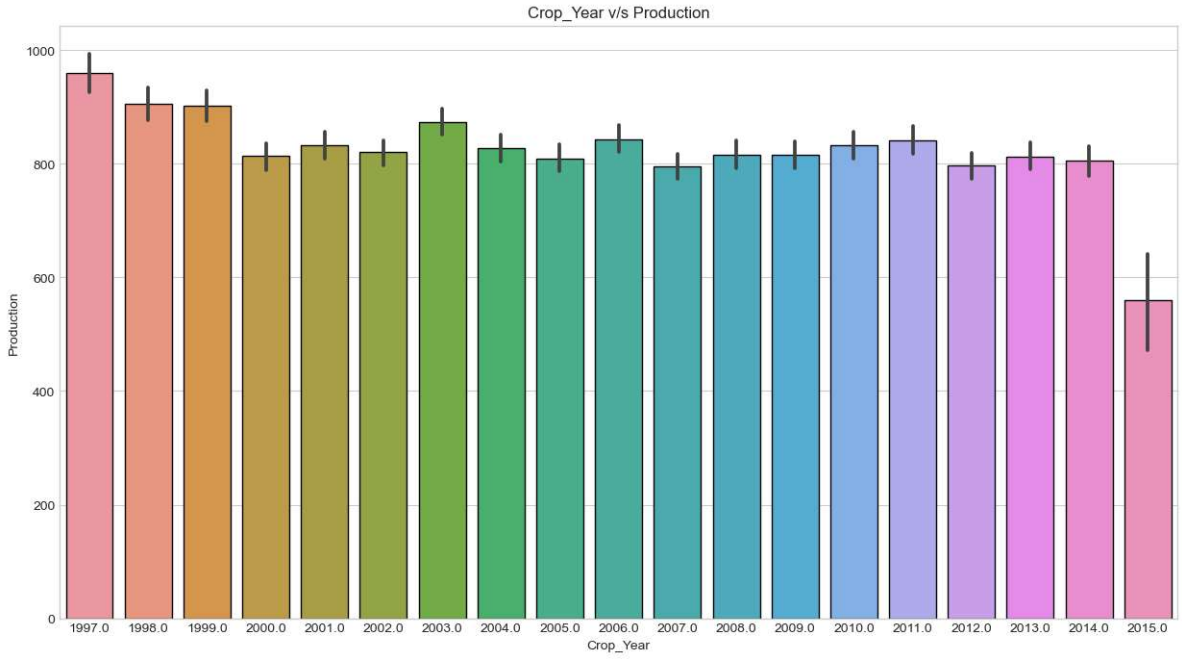
- The States Punjab, Sikkim, Gujarat has maximumn number of Agricutural land among all the states
- Chandigarh has lowest number of agricultural land.

In [ ]:
```
```

Crop_Year v/s Production

In [118]:
```
sns.barplot(data=crop_new_2, x="Crop_Year", y="Production", edgecolor="black")
plt.title("Crop_Year v/s Production")
```

Out[118]:
```
Text(0.5, 1.0, 'Crop_Year v/s Production')
```

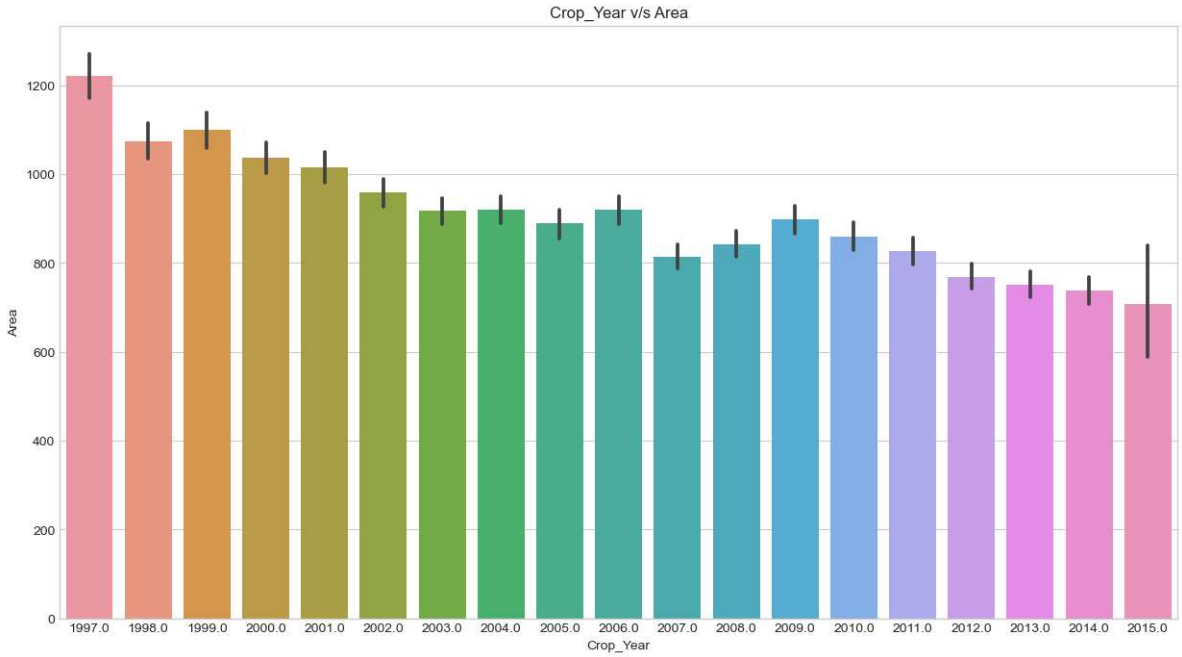


- Since 1997 there isn't any much diffrence in Crop Production.
- Except 2015 has lowest crop production.

Crop_Year v/s Area

In [119]:
```
sns.barplot(data=crop_new_2, x="Crop_Year", y="Area")
plt.title("Crop_Year v/s Area")
```

Out[119]: Text(0.5, 1.0, 'Crop_Year v/s Area')

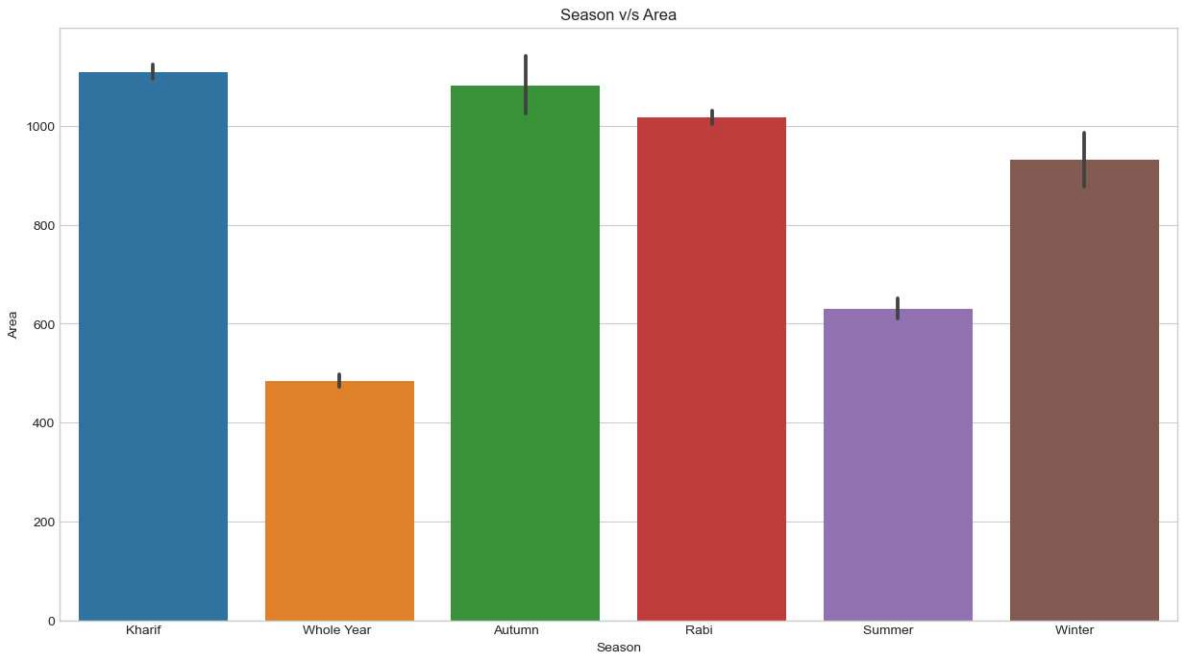


- From 1997 to 2015 there is a decrease in agricultral area.

Season v/s Area

In [120]:
```
sns.barplot(data=crop_new_2, x="Season", y="Area")
plt.title("Season v/s Area")
```

Out[120]: Text(0.5, 1.0, 'Season v/s Area')



In [ ]:

In [ ]:

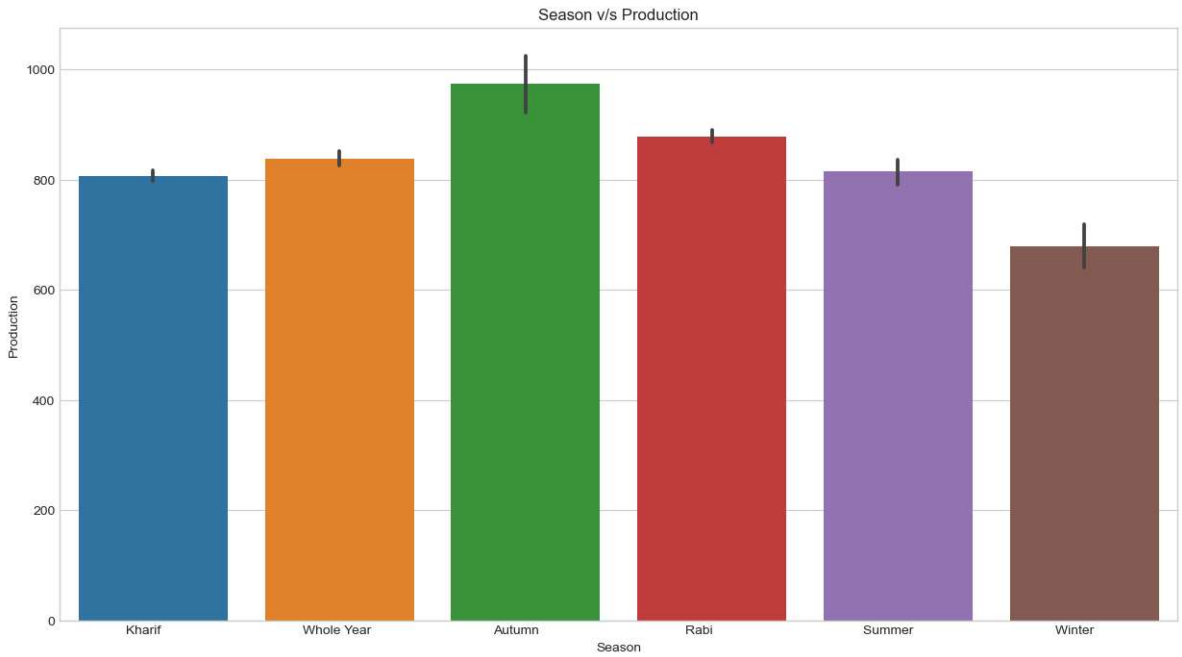
- The agricultural area is Maximum in "Autumn"

In [ ]:

Season v/s Production

In [121]:
```
sns.barplot(data=crop_new_2, x="Season", y="Production")
plt.title("Season v/s Production")
```

Out[121]:
```
Text(0.5, 1.0, 'Season v/s Production')
```



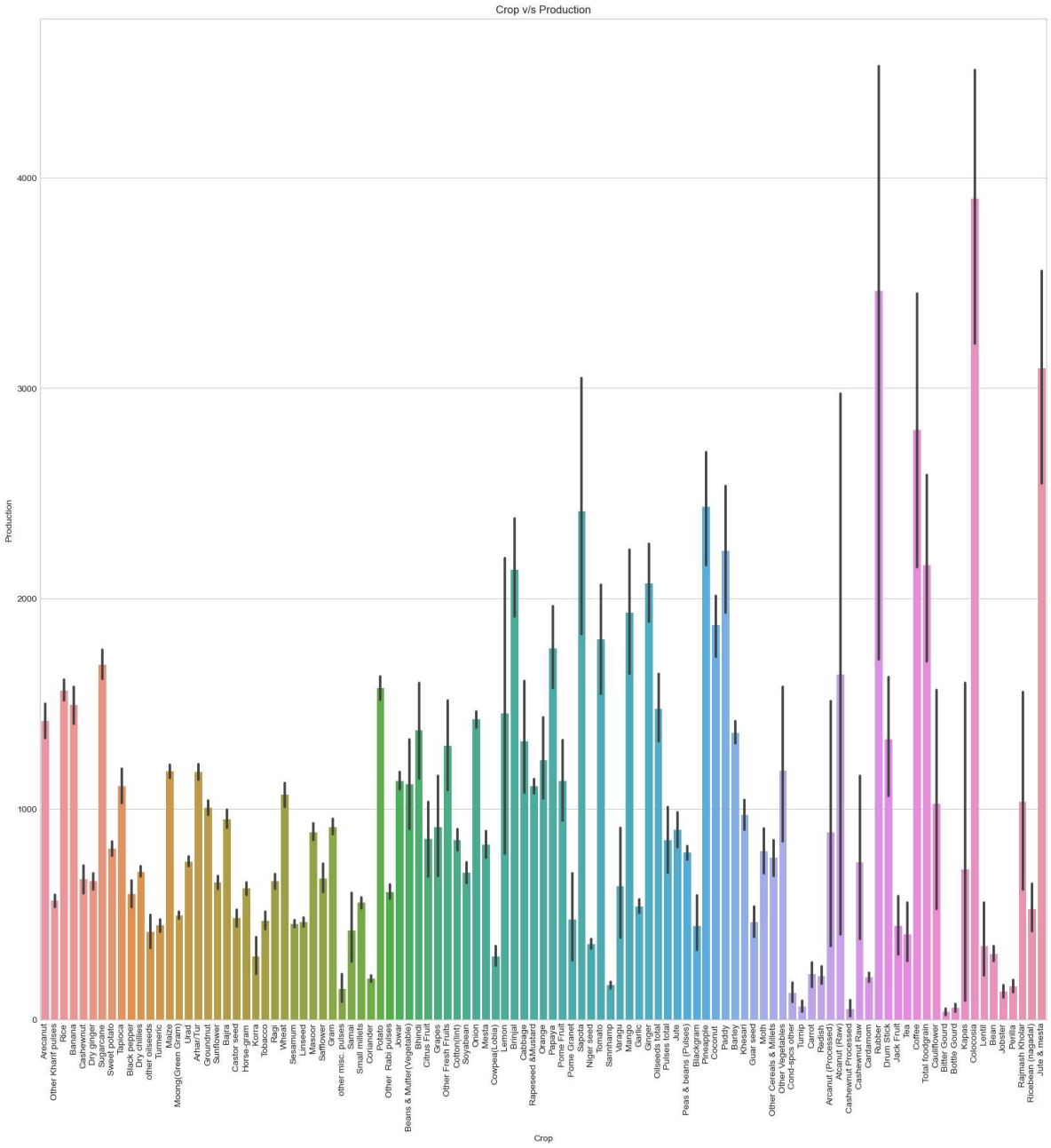
- The Production area is Maximum in "Autumn"

In [ ]:

Crop v/s Production

In [ ]:

In [122]:
```
plt.figure(figsize=(20,20))
sns.barplot(data=crop_new_2, x="Crop", y="Production")
plt.title("Crop v/s Production")
plt.xticks(rotation=90)
plt.show()
```

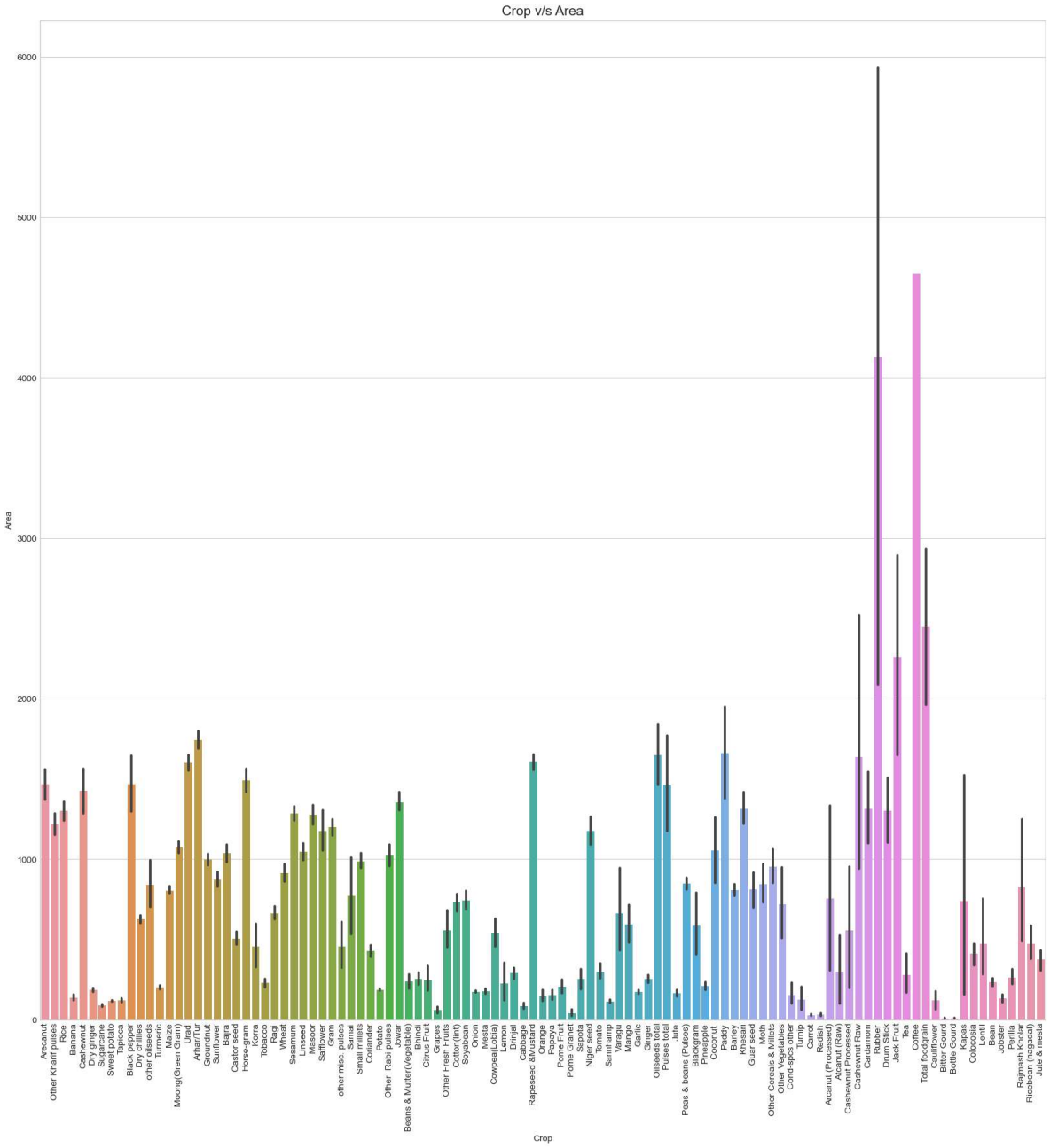


- The production of colcosisia is done in maximum.

In [ ]:
```

```

Crop v/s Area

In [123]:
```
plt.figure(figsize=(20,20))
sns.barplot(data=crop_new_2, x="Crop", y="Area")
plt.title("Crop v/s Area", fontdict={"size":15})
plt.xticks(rotation=90)
plt.show()
```



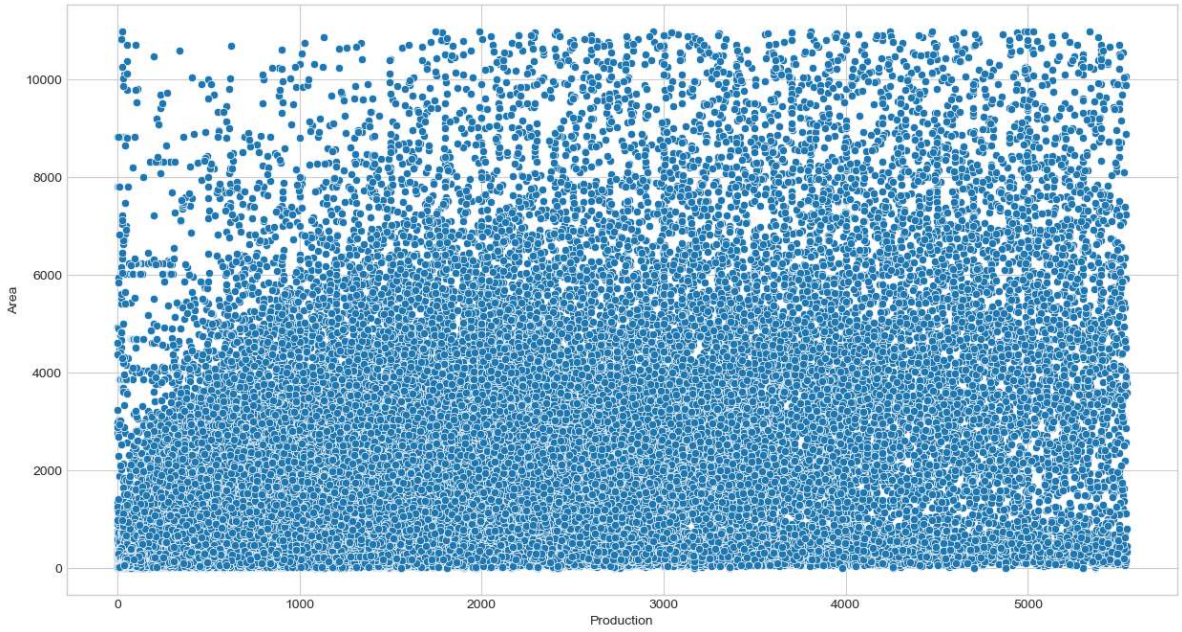
- Area for total food-grains is maximum.

In [ ]:
```

```

Production v/s Area

In [124]:
```
sns.scatterplot(data=crop_new_2, x="Production", y="Area")
```

Out[124]:
```
<AxesSubplot:xlabel='Production', ylabel='Area'>
```

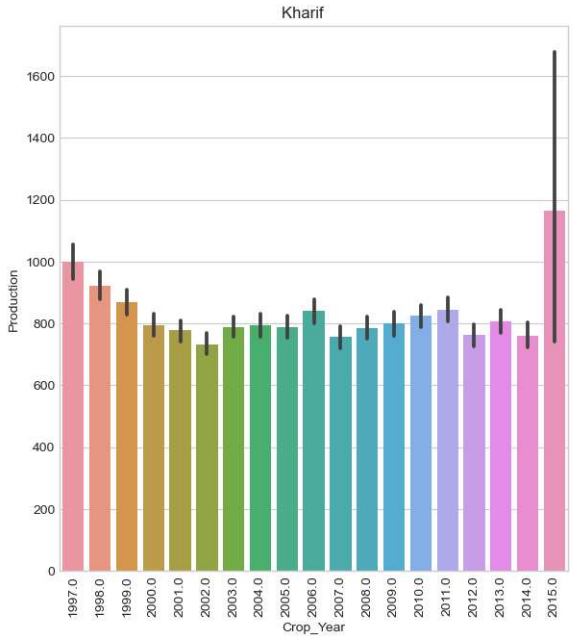


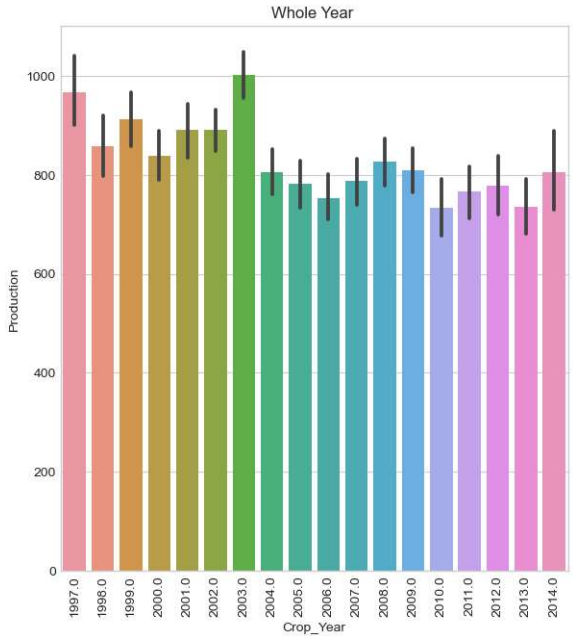
- Production is corretated with agricultural area

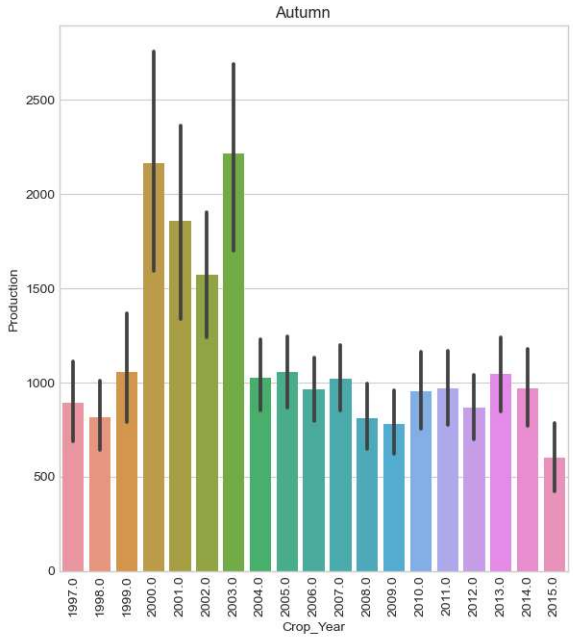
In [ ]:
```

```

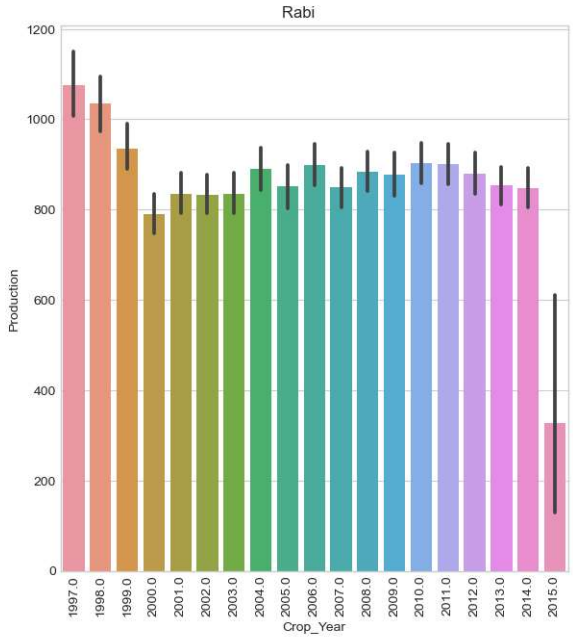
Season-vise Yearly Production

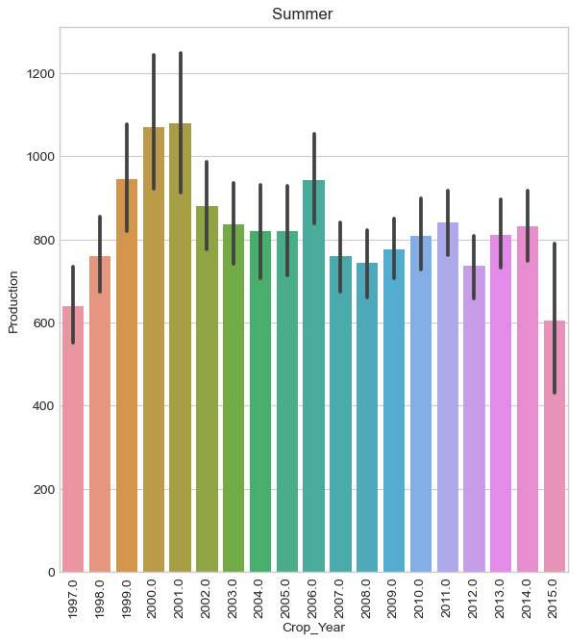
In [125]:
```
num=1
plt.figure(figsize=(15,25))
for i in crop_new_2.Season.unique():
    plt.subplot(3,2,num)
    sns.barplot(data=crop_new_2[crop_new_2["Season"]==i], x="Crop_Year", y="Pr
    plt.xticks(rotation=90)
    plt.title(i)
    num+=1
```

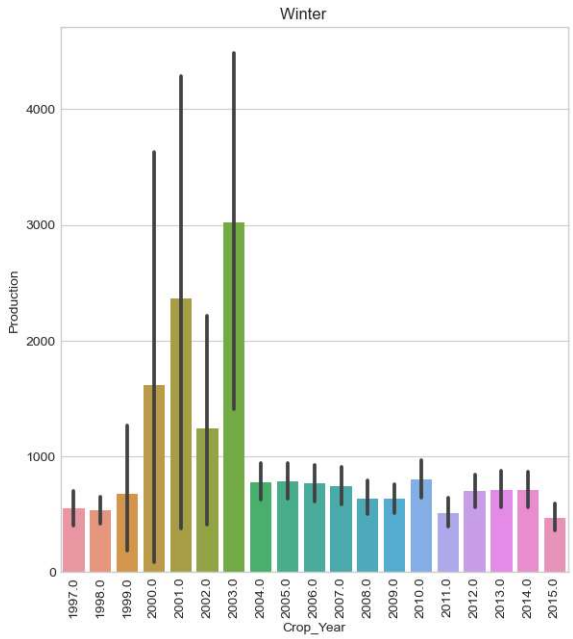

Kharif
Production
Crop_Year


Whole Year
Production
Crop_Year


Autumn
Production
Crop_Year


Rabi
Production
Crop_Year


Summer
Production
Crop_Year

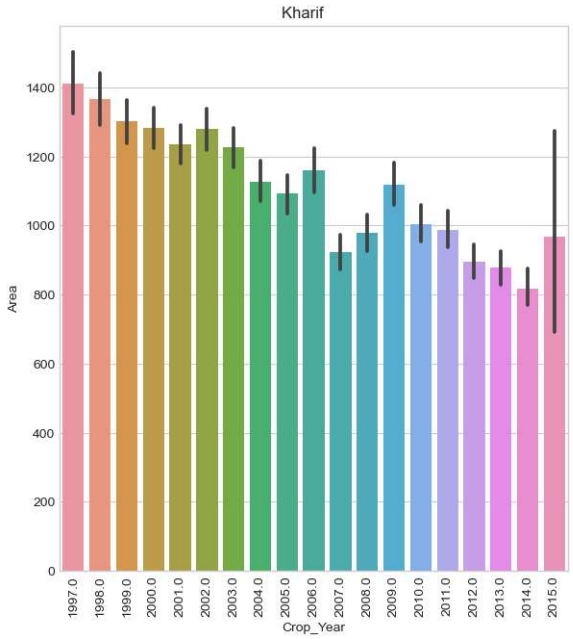

Winter
Production
Crop_Year

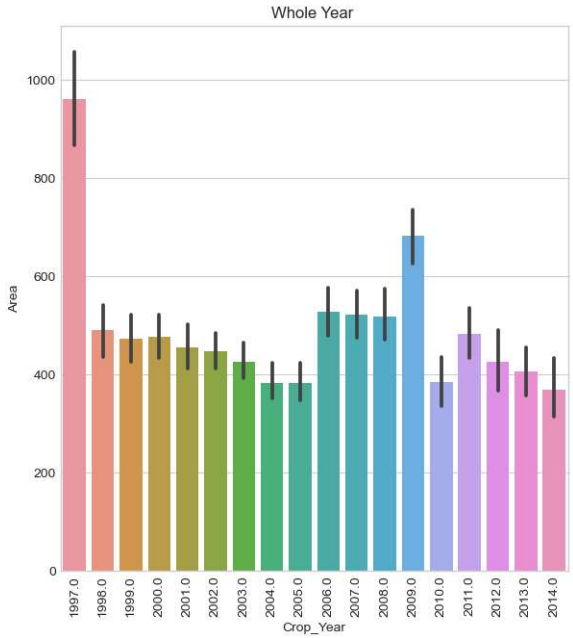
- During Kharif season the in 2015 there was maximum production
- Winter crops have very low production exacey of year 2001 and 2003 and similar trend is shown by Auttum.
- Whole year crop does not have vast diffrence.

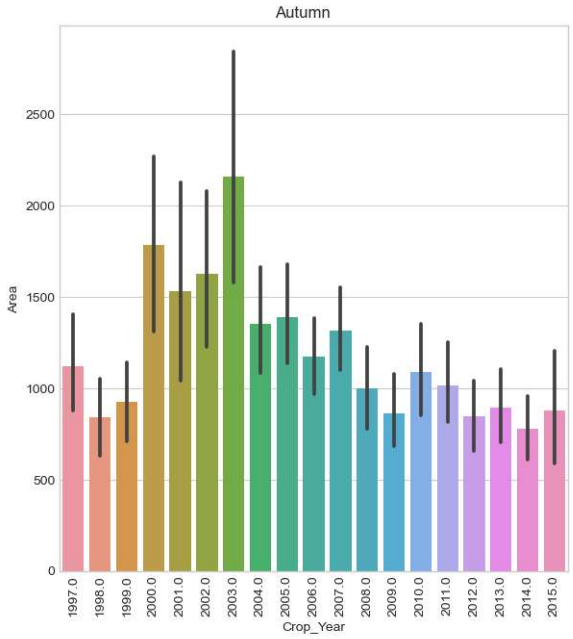
In [ ]:

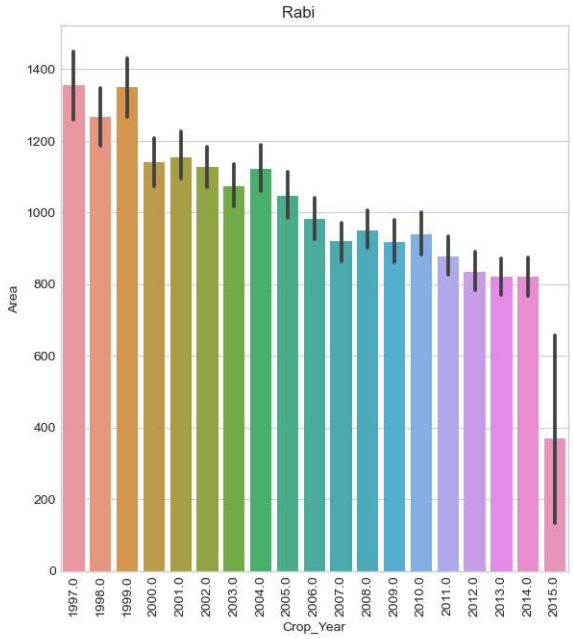
Season-Wise v/s Area

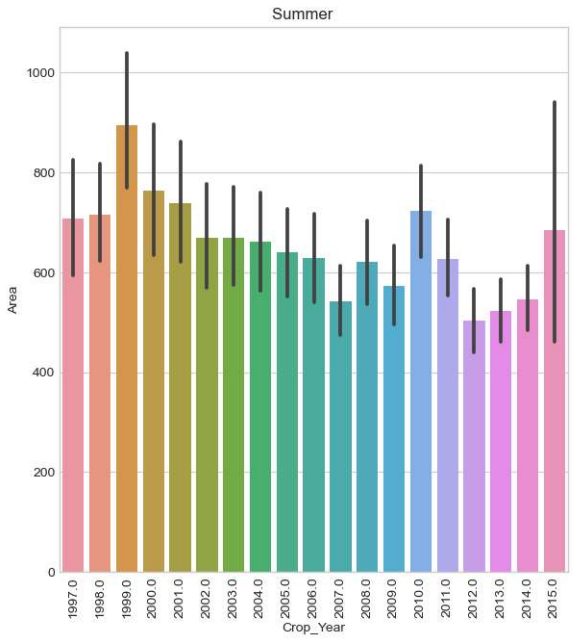
In [126]:
```
num=1
plt.figure(figsize=(15,25))
for i in crop_new_2.Season.unique():
    plt.subplot(3,2,num)
    sns.barplot(data=crop_new_2[crop_new_2["Season"]==i], x="Crop_Year", y="Ar
    plt.xticks(rotation=90)
    plt.title(i)
    num+=1
```

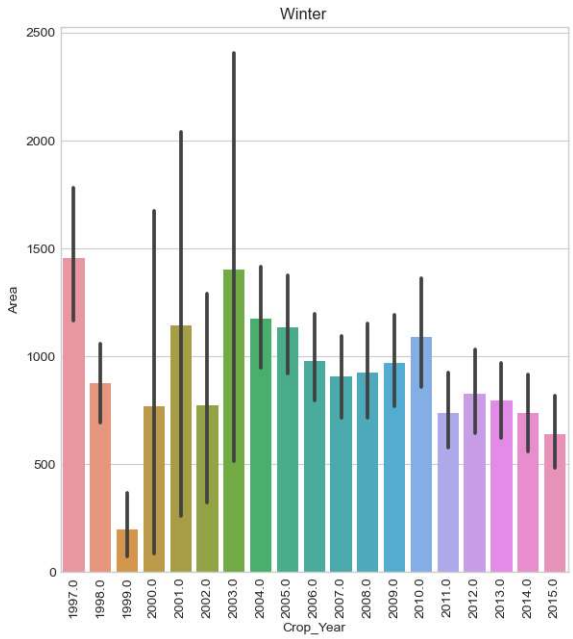

Kharif
Area
Crop_Year


Whole Year
Area
Crop_Year


Autumn
Area
Crop_Year


Rabi
Area
Crop_Year


Summer
Area
Crop_Year


Winter
Area
Crop_Year

- Rabi crops agricural land is decreasing by the years
- Similar but slow tred is seen in Autumn Season and Kharif Season.

In [ ]:

Summary

- The Dataset given was about the Production of Crops from the year 1997 to 2015
- The Traget variable was the "Production" colums.
- Univate Analysis:
  - The agricultural area is Maximum in "Autumn"
  - The States Punjab, Sikkim, Gujarat has maximumn number of Agricutural land among all the states
  - Chandigarh has lowest number of agricultural land.
  - The District Bilaspur has maximum count, i.e it has maximum crop production.
  - Namsai has lowest count, i.e it has lowest crop production.
  - Moong(Green Grams) has maximum count.
  - Rubber has lowest count.
  - The data of Production and Arae highly skweed .
  - This is maybe beacuse that every state has varying number of agricultural land.
  - Every State produce diffrent crops in abundance.
  - The Procudtion is maximum in Autumn Season.

- Bivariate Analysis
  - During Kharif season the in 2015 there was maximum production
  - Winter crops have very low production exacey of year 2001 and 2003 and similar trend is shown by Auttum.
  - Whole year crop does not have vast diffrence.
  - Rabi crops agricural land is decreasing by the years
  - Similar but slow tred is seen in Autumn Season and Kharif Season.
  - Production is corretated with agricultural area.
- The Production and Quality of Land for the agricultural is afftected by the year
- Hence we need to take necessary measure to ensure that the production increase by the year

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: