Libraries

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

plt.rcParams['figure.figsize'] = [15,8]
```

File Loading

```
crop_prod = pd.read_csv('crop_production.csv')
crop_prod.head()
```

```
                    State_Name District_Name  Crop_Year       Season

0  Andaman and Nicobar Islands      NICOBARS       2000
Kharif       \
1  Andaman and Nicobar Islands      NICOBARS       2000  Kharif

2  Andaman and Nicobar Islands      NICOBARS       2000  Kharif

3  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year

4  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year


                  Crop  Area  Production
0             Arecanut  1254        2000
1  Other Kharif pulses     2           1
2                 Rice   102         321
```

```
3               Banana   176         641
4            Cashewnut   720         165
```

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 target column

```
crop_prod.shape
```

```
(246091, 7)
```

```
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     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
```

```
crop_prod.describe().T
```

```
             count   mean      std  min  25%  50%  75%         max
Crop_Year  246091   2006        5 1997 2002 2006 2010       2015
Area       246091  12003    50523    0   80  582 4392    8580100
Production 242361 582503 17065813    0   88  729 7023 1250800000
```

```
crop_prod.dtypes
```

```
State_Name         object
District_Name      object
Crop_Year           int64
Season             object
Crop               object
Area              float64
Production        float64
dtype: object
```

```
crop_prod['Crop_Year'] = crop_prod['Crop_Year'].astype('float64')
```

```
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      246091 non-null  object 
 1   District_Name   246091 non-null  object 
 2   Crop_Year       246091 non-null  float64
 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(3), object(4)
memory usage: 13.1+ MB
```

Null-value Handling

```
null_count = crop_prod.isnull().sum()
null_count
```

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

```
count_perc = ((crop_prod.isnull().sum())/len(crop_prod))*100
count_perc
```

```
State_Name      0
District_Name   0
```

```
Crop_Year       0
Season          0
Crop            0
Area            0
Production      2
dtype: float64
```

```
# Treating the missing value without removing null values
```

Production

```
crop_prod.groupby('Crop')['Production'].mean()
```

```
Crop
Apple                                0
Arcanut (Processed)               9642
Arecanut                         13229
Arhar/Tur                         5261
Ash Gourd                            0
Atcanut (Raw)                    46362
Bajra                            24109
Banana                           46643
Barley                            5369
Bean                               312
Beans & Mutter(Vegetable)         1266
Beet Root                            0
Ber                                  0
Bhindi                            1344
Bitter Gourd                         4
Black pepper                      1974
Blackgram                          444
Bottle Gourd                         7
Brinjal                           3736
Cabbage                           1527
Cardamom                           488
Carrot                             145
Cashewnut                         2426
Cashewnut Processed                387
Cashewnut Raw                     2813
Castor seed                       4878
Cauliflower                        151
Citrus Fruit                      2922
Coconut                       66384897
Coffee                           21669
Colocosia                         4531
Cond-spcs other                    126
Coriander                         1133
Cotton(lint)                     67777
```

Cowpea(Lobia)	442
Cucumber	0
Drum Stick	689
Dry chillies	2758
Dry ginger	3363
Garlic	2725
Ginger	3410
Gram	13756
Grapes	13578
Groundnut	12742
Guar seed	10144
Horse-gram	1089
Jack Fruit	620
Jobster	131
Jowar	16395
Jute	128948
Jute & mesta	5616
Kapas	712
Khesari	4257
Korra	982
Lab-Lab	0
Lemon	13872
Lentil	347
Linseed	583
Litchi	0
Maize	19826
Mango	28442
Masoor	3168
Mesta	7002
Moong(Green Gram)	1811
Moth	4706
Niger seed	670
Oilseeds total	102975
Onion	10374
Orange	8630
Other Rabi pulses	1558
Other Cereals & Millets	1951
Other Citrus Fruit	0
Other Dry Fruit	0
Other Fresh Fruits	974
Other Kharif pulses	1226
Other Vegetables	2528
Paddy	66185
Papaya	8121
Peach	0
Pear	0
Peas (vegetable)	0
Peas & beans (Pulses)	1968
Perilla	157

```
Pineapple                       8614
Plums                              0
Pome Fruit                      1621
Pome Granet                     1005
Potato                         61444
Pulses total                   58930
Pump Kin                           0
Ragi                            8535
Rajmash Kholar                  1033
Rapeseed &Mustard              12063
Redish                            65
Ribed Guard                        0
Rice                          106449
Ricebean (nagadal)              523
Rubber                         40528
Safflower                       2162
Samai                           1669
Sannhamp                         173
Sapota                          7190
Sesamum                         1248
Small millets                   1226
Snak Guard                         0
Soyabean                       44589
Sugarcane                     707255
Sunflower                       2419
Sweet potato                    1904
Tapioca                        66292
Tea                             2193
Tobacco                         3980
Tomato                         12840
Total foodgrain               230164
Turmeric                        2391
Turnip                            45
Urad                            2308
Varagu                           857
Water Melon                        0
Wheat                         169183
Yam                                0
other fibres                       0
other misc. pulses               139
other oilseeds                  8030
Name: Production, dtype: float64
```

- We well fill Production Column null value with avg Production of each crop.

```
crop_prod['Production'] =
crop_prod['Production'].fillna(crop_prod.groupby('Crop')
['Production'].transform('mean'))

crop_prod.Production.isnull().sum()
```

```
0
```

```
crop_prod.isna().sum()
```

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

- Hence all the Null Values removed

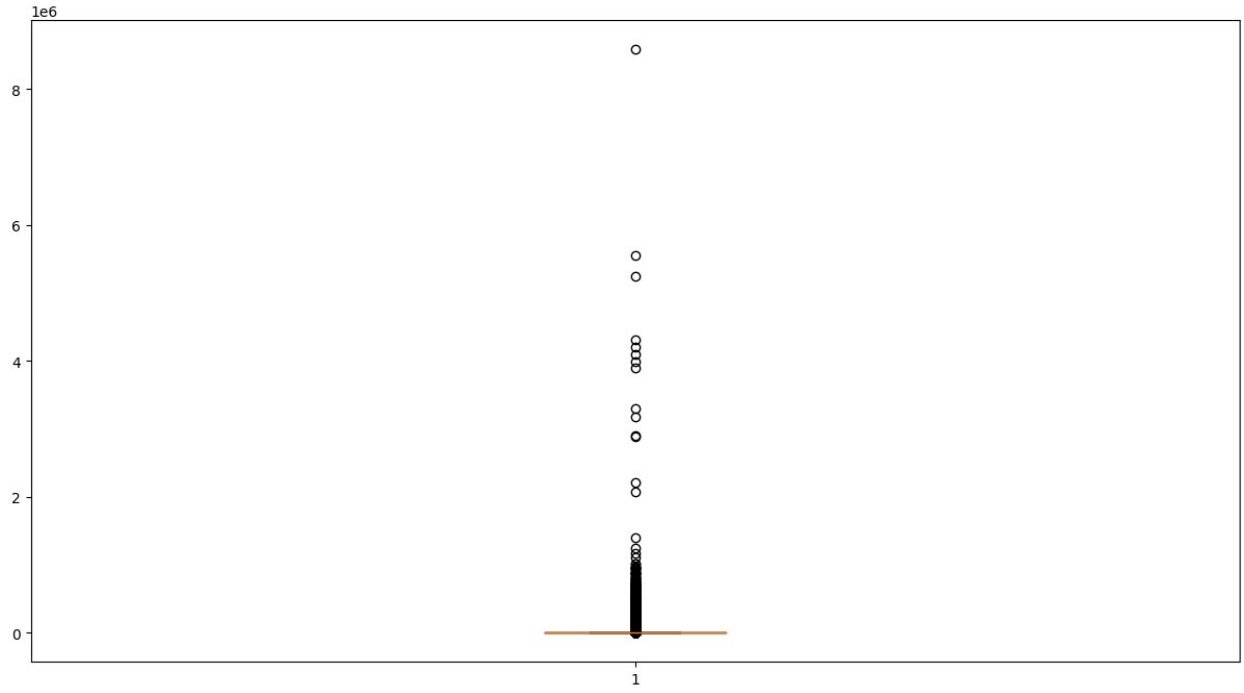
Detect and Handle the Outlier

```
crop_prod.describe().T
```

```
             count   mean       std  min  25%  50%  75%         max
Crop_Year   246091   2006         5 1997 2002 2006 2010        2015
Area        246091  12003     50523    0   80  582 4392     8580100
Production 246091 581403 16950146    0   91  771 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

```
plot = plt.boxplot(crop_prod['Area'])
```




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

```
[array([0.04, 0.04]), array([10860., 10860.])]
```

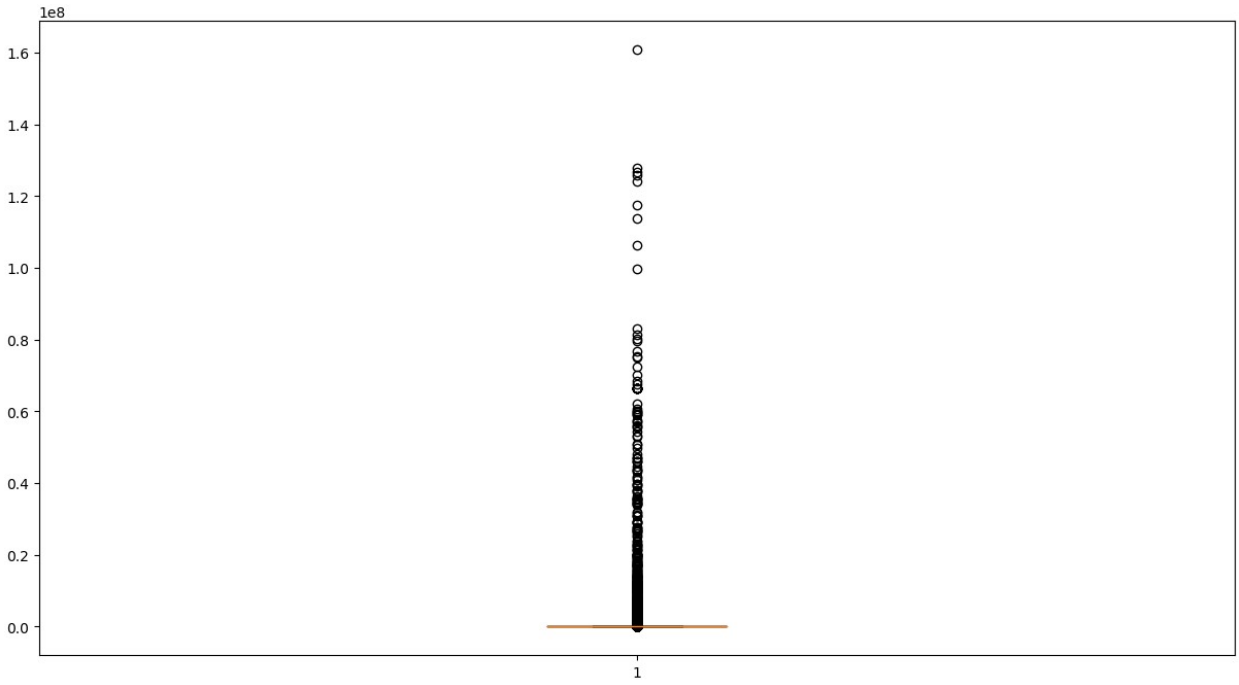
• The lower wiskers is equal to 0.04 and upper wiskers is equal to 10986

• Hence the data above 10986 and below 0.04 are outliers in columns "area".

```
crop_new1 =
crop_prod[(crop_prod['Area']<10986)&(crop_prod["Area"]>8.84)]
crop_new1.head()
```

```
                    State_Name District_Name  Crop_Year       Season

0  Andaman and Nicobar Islands      NICOBARS       2000
Kharif       \
2  Andaman and Nicobar Islands      NICOBARS       2000  Kharif

3  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year

4  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year

6  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year


          Crop  Area  Production
0    Arecanut  1254        2000
2        Rice   102         321
```

```
3      Banana   176         641
4   Cashewnut   720         165
6  Dry ginger    36         100

plot1 = plt.boxplot(crop_new1['Production'])
```




```
wiskers2 = [i.get_ydata() for i in plot1["caps"]]
wiskers2

[array([0., 0.]), array([6404., 6404.])]

crop_new_2 = crop_new1[(crop_new1['Production'] > 0) &
(crop_new1['Production'] < 6404)]
crop_new_2.head()

                    State_Name District_Name  Crop_Year       Season

0  Andaman and Nicobar Islands      NICOBARS       2000
Kharif       \
2  Andaman and Nicobar Islands      NICOBARS       2000  Kharif

3  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year

4  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year

6  Andaman and Nicobar Islands      NICOBARS       2000  Whole Year


         Crop  Area  Production
```

```
0    Arecanut  1254        2000
2        Rice   102         321
3      Banana   176         641
4   Cashewnut   720         165
6  Dry ginger    36         100
```

```
crop_new_2.shape
```

```
(157265, 7)
```

- The dataframe 'crop_new_2' is the new after removing the outlier.
- Further we will perform the analysis on this new dataframe.

Investigation of wrong datatype.

```
# Check and visualize all data present in dataframe and investigate
the wrong.
crop_new_2.dtypes
```

```
State_Name        object
District_Name     object
Crop_Year        float64
Season            object
Crop              object
Area             float64
Production       float64
dtype: object
```

```
crop_new_2.Crop_Year.nunique()
```

```
19
```

- The Column Crop_year is a categorical Columns as it has only 19 unique values.
- Hence we will change the datatype into 'object'

```
# Perform the DataType Casting, If needed.

crop_new_2.Crop_Year = crop_new_2.Crop_Year.astype('object')

crop_new_2.Crop_Year.dtype
```

```
dtype('O')
```

Check for the Duplicated in the dataset

```
# Show the duplicates here
crop_new_2.duplicated().sum()
```

```
0
```

Identifying the irrelevent columns in dataset:

```
# First let divide the categorical and numrical 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(object)

num_crop.head()
```

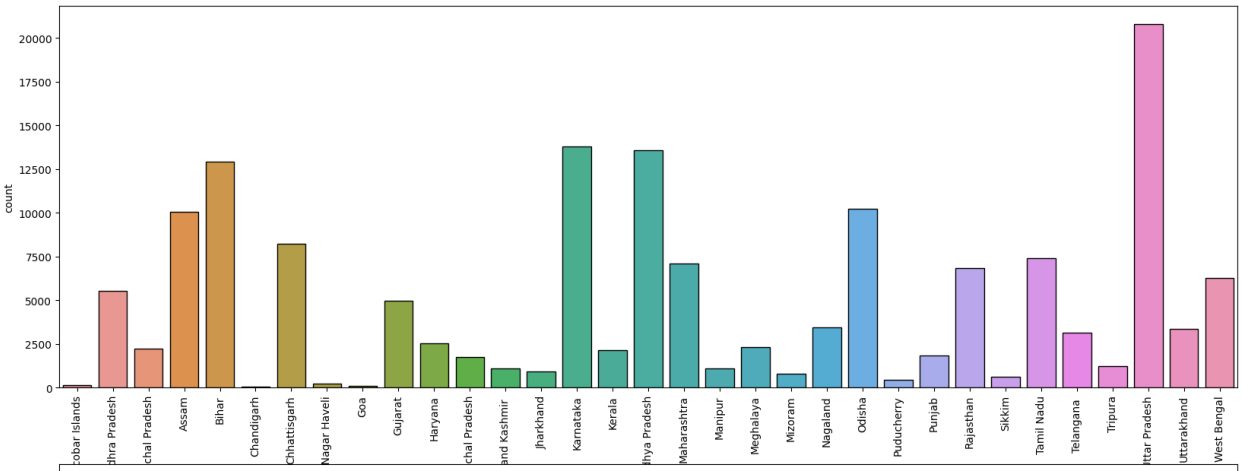
```
   Area  Production
0  1254        2000
2   102         321
3   176         641
4   720         165
6    36         100
```

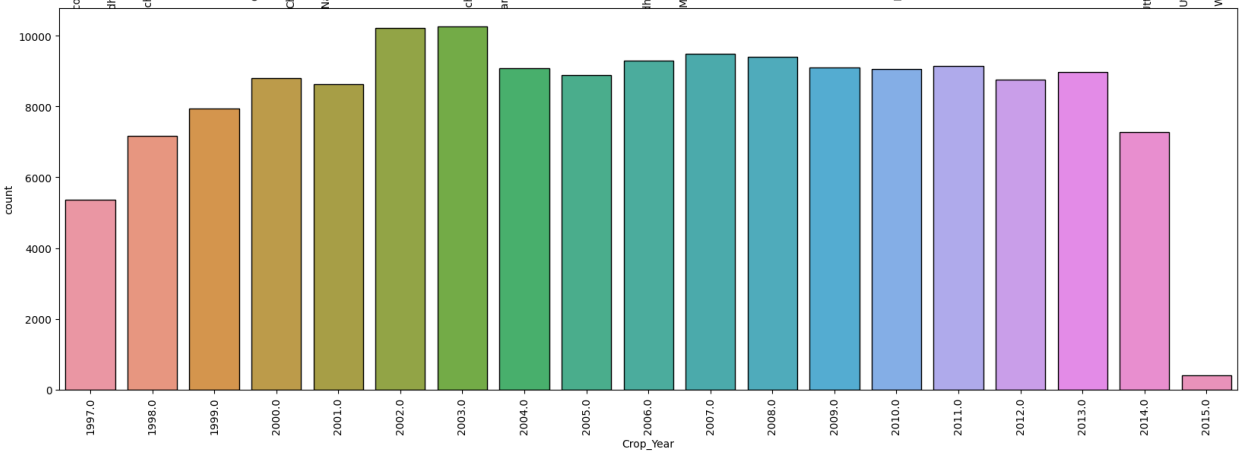
```
cat_crop.head()
```

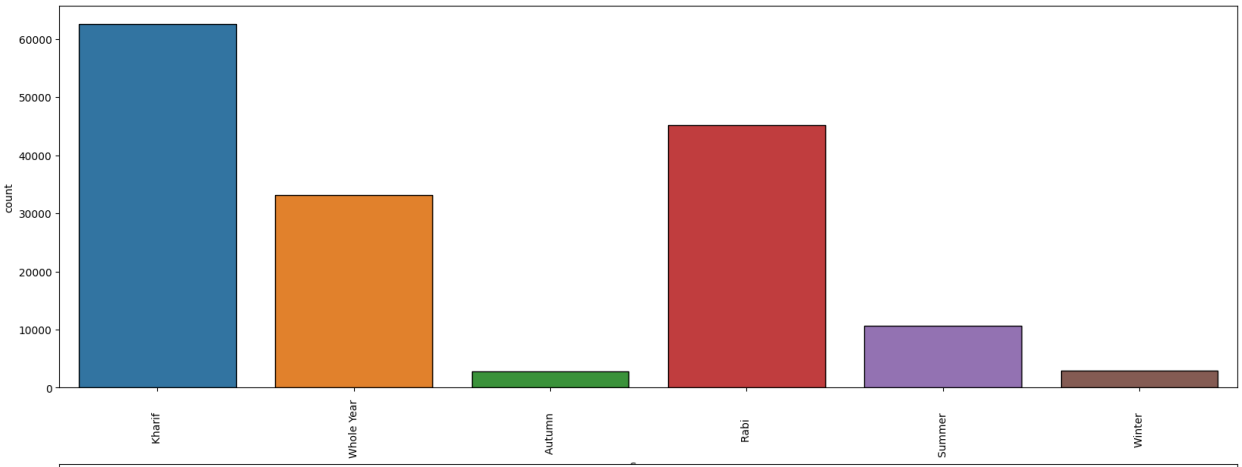
```
                    State_Name District_Name Crop_Year       Season  \
0  Andaman and Nicobar Islands      NICOBARS      2000  Kharif
2  Andaman and Nicobar Islands      NICOBARS      2000  Kharif
3  Andaman and Nicobar Islands      NICOBARS      2000  Whole Year
4  Andaman and Nicobar Islands      NICOBARS      2000  Whole Year
6  Andaman and Nicobar Islands      NICOBARS      2000  Whole Year

         Crop
0    Arecanut
2        Rice
3      Banana
4   Cashewnut
6  Dry ginger
```

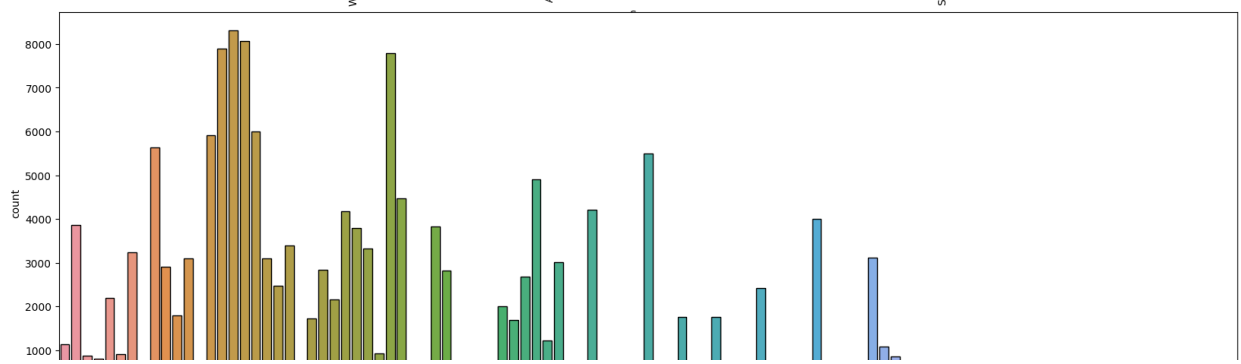
```
# Visualize all coulumns at a time ( Numerical and Categorical
Columns)

# 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
```

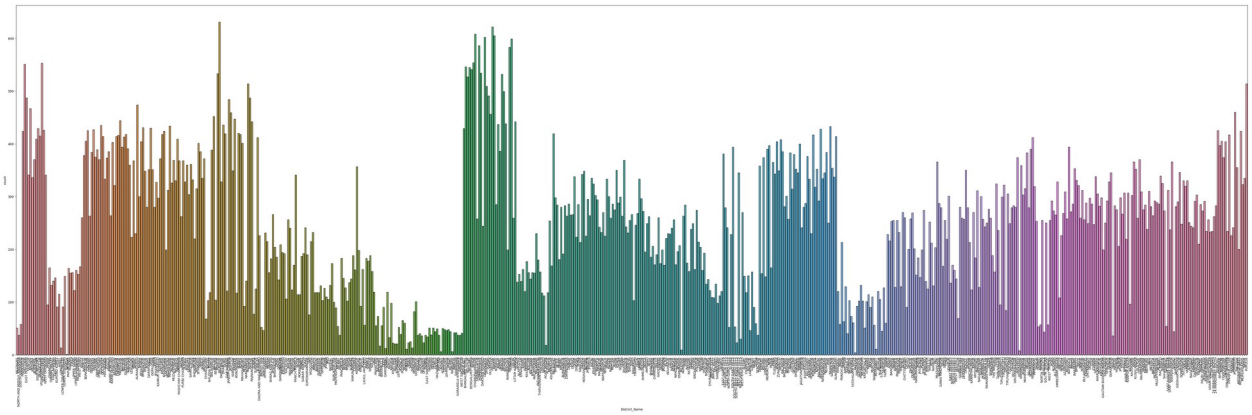

count
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Dadra and Nagar Haveli
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan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


count
1997.0
1998.0
1999.0
2000.0
2001.0
2002.0
2003.0
2004.0
2005.0
2006.0
2007.0
2008.0
2009.0
2010.0
2011.0
2012.0
2013.0
2014.0
2015.0
Crop_Year


count
Kharif
Whole Year
Autumn
Rabi
Summer
Winter


count

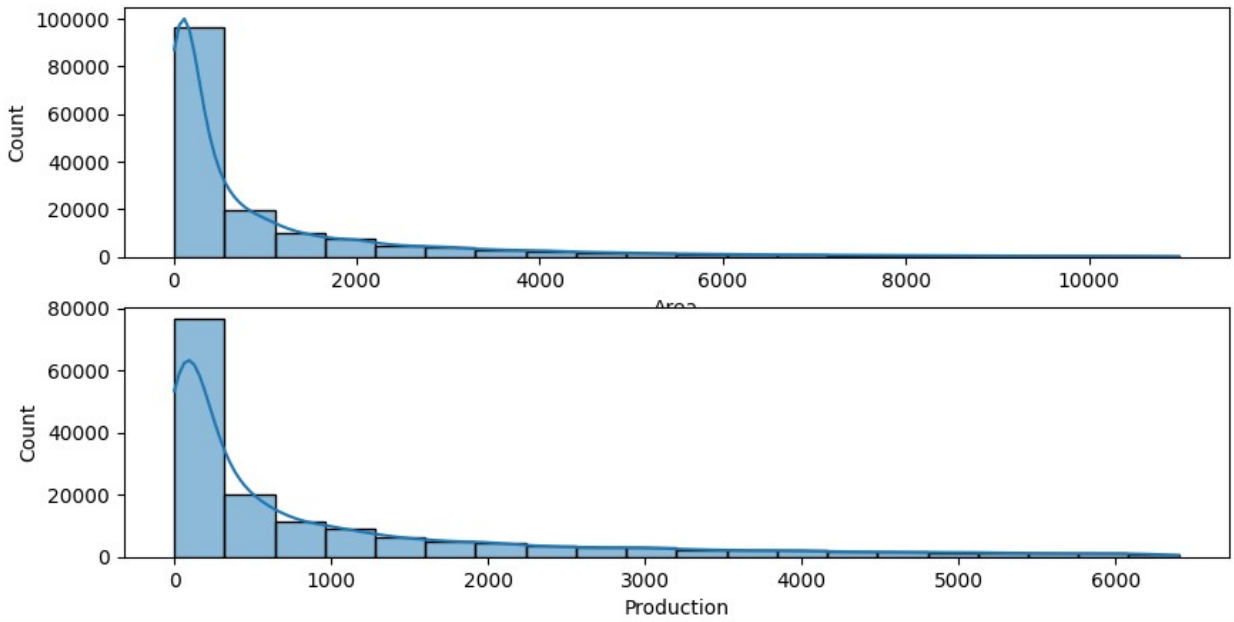
```
plt.figure(figsize = (70,20))

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



- All the Categorical columns look fine.

```
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)
```

```
<Axes: 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.

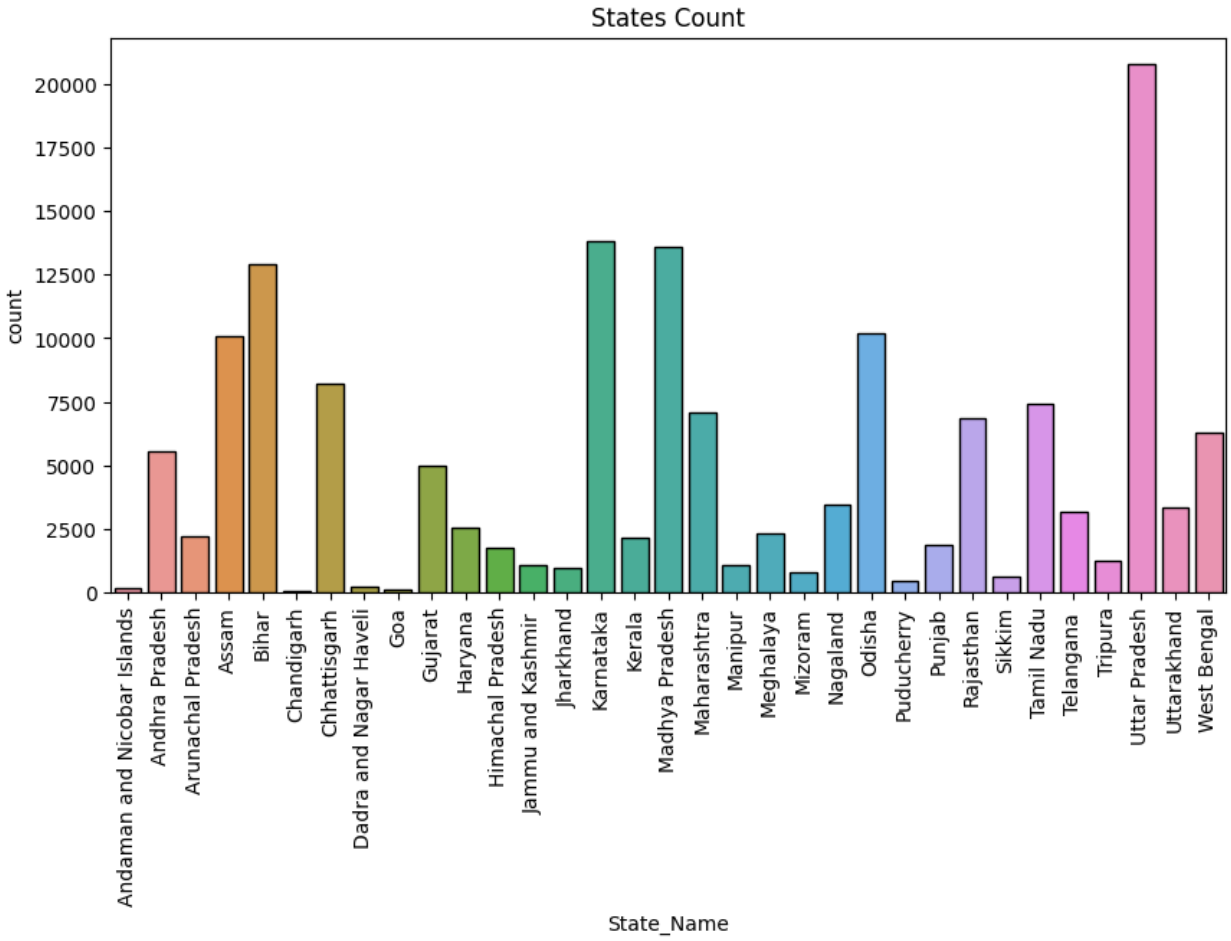
Univarate Analysis.

```
crop_new_2.State_Name.value_counts()
```

```
State_Name
Uttar Pradesh                  20801
Karnataka                      13814
Madhya Pradesh                 13595
Bihar                          12938
Odisha                         10211
Assam                          10077
Chhattisgarh                    8234
Tamil Nadu                      7409
Maharashtra                     7101
Rajasthan                       6836
West Bengal                     6279
Andhra Pradesh                  5549
Gujarat                         4973
Nagaland                        3447
Uttarakhand                     3346
Telangana                       3142
Haryana                         2540
Meghalaya                       2296
Arunachal Pradesh               2215
Kerala                          2158
Punjab                          1828
Himachal Pradesh                1754
Tripura                         1223
Jammu and Kashmir               1087
Manipur                         1080
Jharkhand                        924
Mizoram                          806
Sikkim                           610
Puducherry                       454
Dadra and Nagar Haveli           226
Andaman and Nicobar Islands      146
Goa                               98
Chandigarh                        68
Name: count, dtype: int64
```

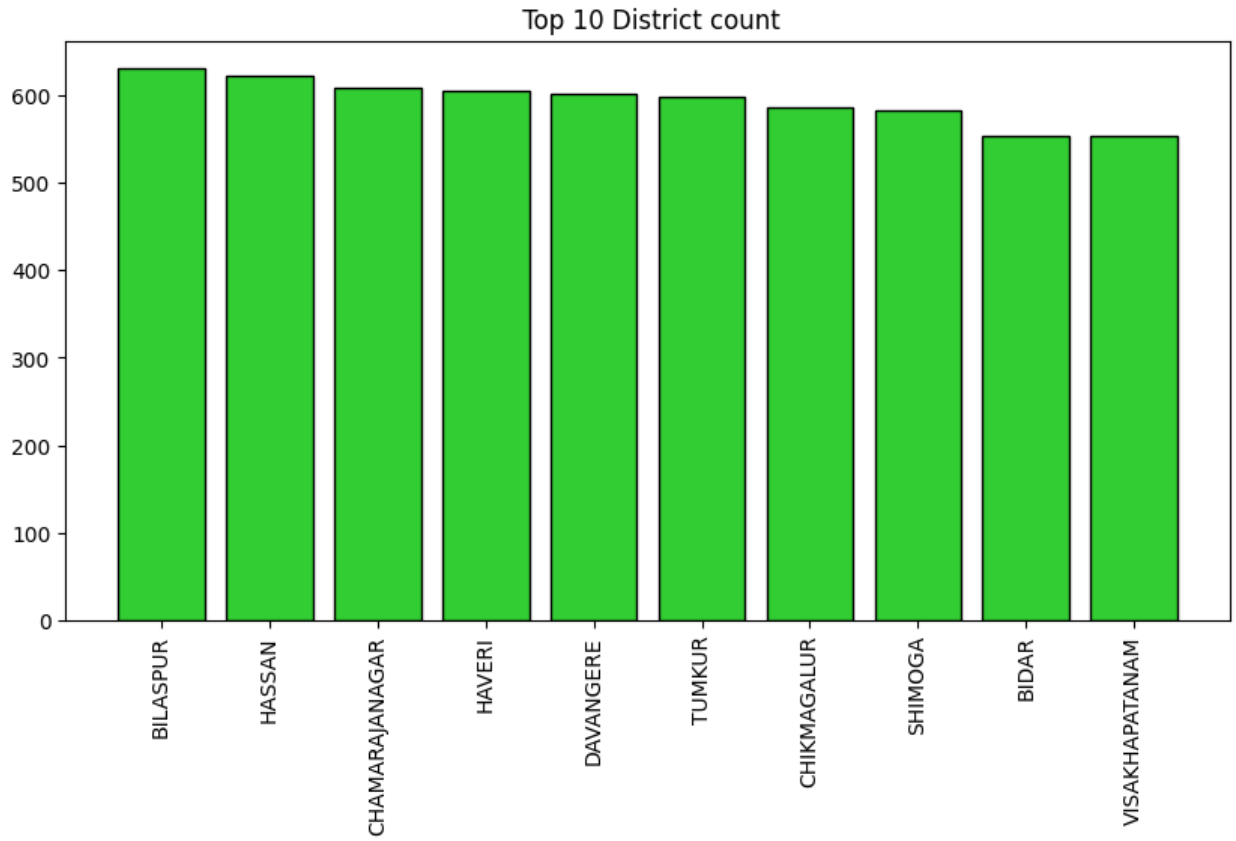
```
plt.figure(figsize=(10,5))

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

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

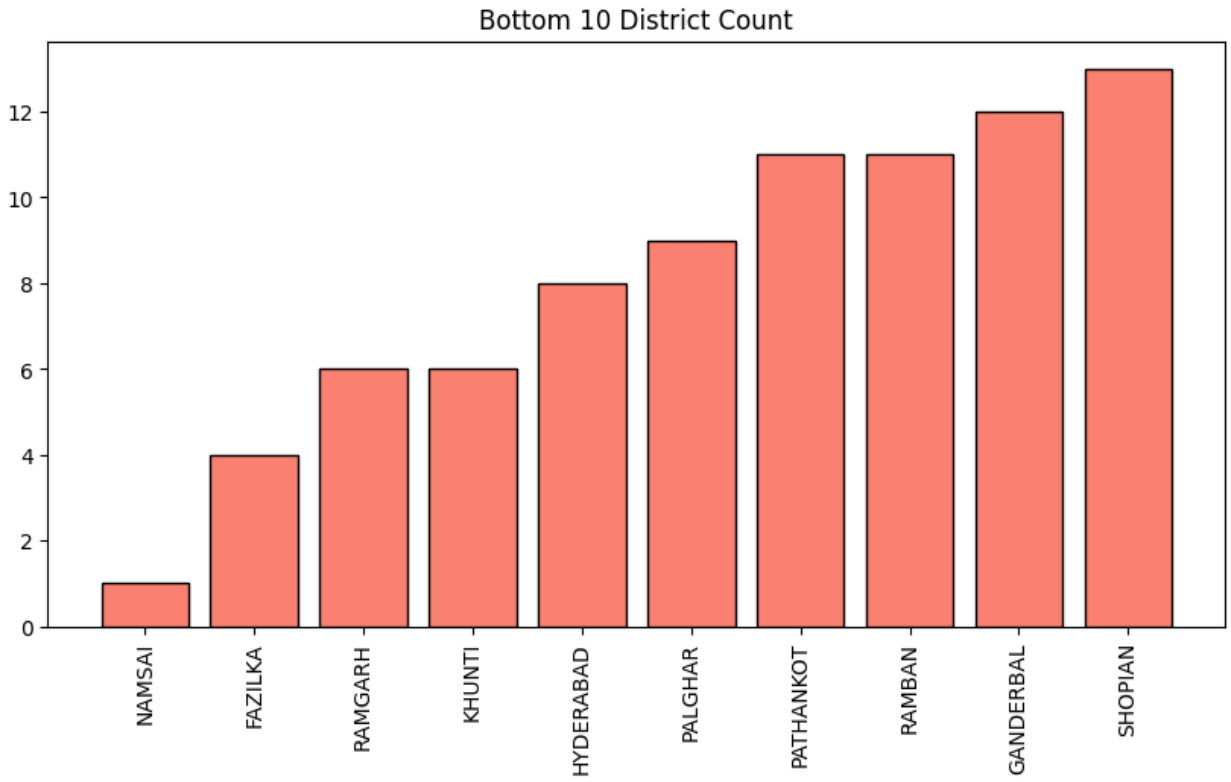


- Uttar Pradesh and Karnataka as the highest count.
- Goa and Chandigarh are the lowest

```
top10 = crop_new_2.District_Name.value_counts(ascending =
False).head(10)

plt.figure(figsize=(10,5))

plt.bar(top10.keys(),top10.values,color="limegreen",edgecolor='black')
plt.title(" Top 10 District count",fontdict={'size':12})
plt.xticks(rotation=90)
plt.show()
```




```
bottom10 = crop_new_2.District_Name.value_counts(ascending =
True).head(10)

plt.figure(figsize=(10,5))

plt.bar(bottom10.keys(),bottom10.values, color = 'salmon', edgecolor =
'black')
plt.title('Bottom 10 District Count', fontdict = {'size': 12 })
plt.xticks(rotation = 90)
plt.show()
```



- Bilaspur has maximum crop production.
- Namsai has the lowest crop production.

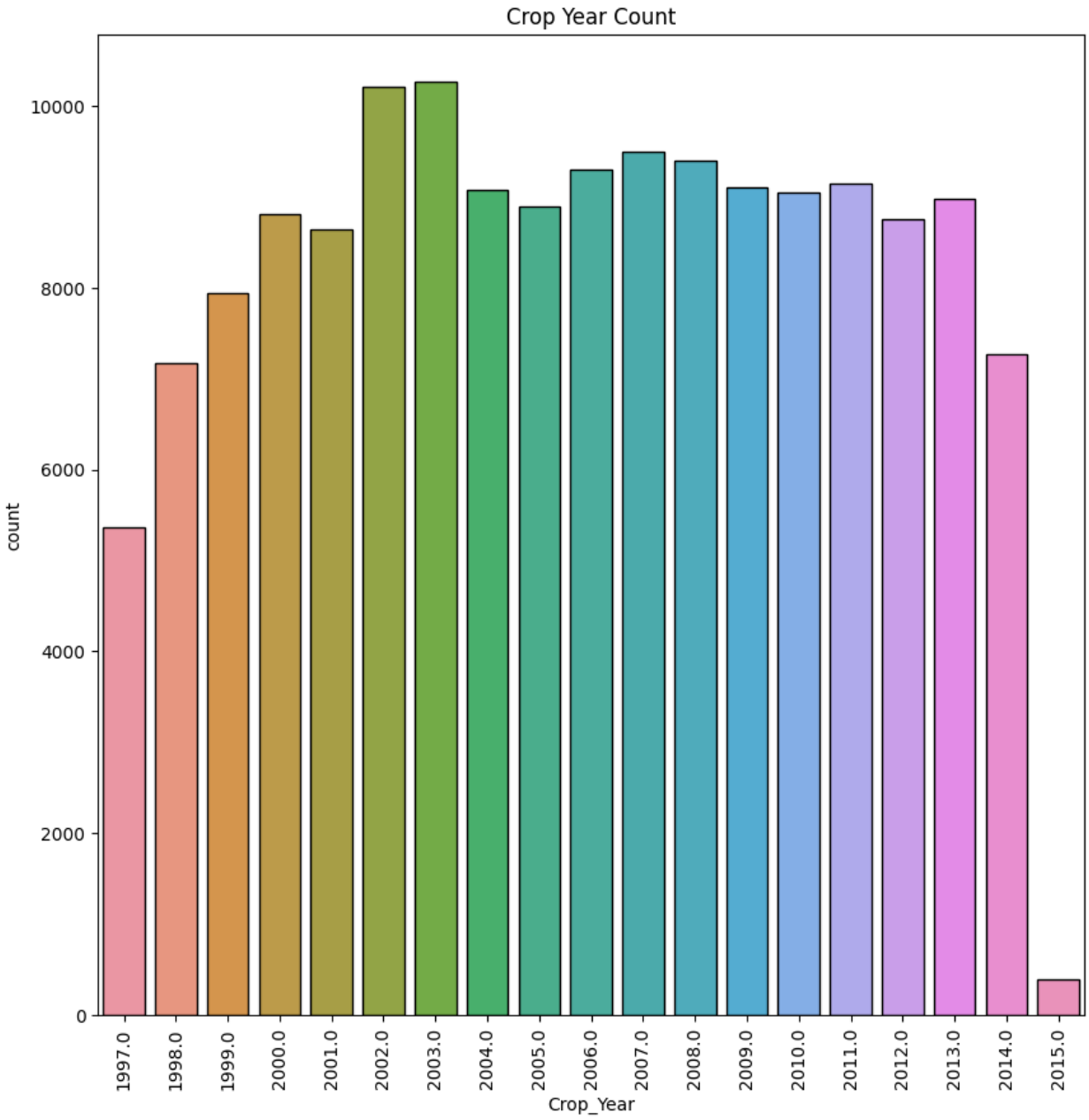
Crop_Year

```
crop_new_2.Crop_Year.value_counts()
```

```
Crop_Year
2003.0    10269
2002.0    10210
2007.0     9490
2008.0     9393
2006.0     9295
2011.0     9147
2009.0     9107
2004.0     9073
2010.0     9050
2013.0     8981
2005.0     8896
2000.0     8810
2012.0     8757
2001.0     8637
1999.0     7944
2014.0     7274
```

```
1998.0     7173
1997.0     5359
2015.0      400
Name: count, dtype: int64
```

```
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()
```

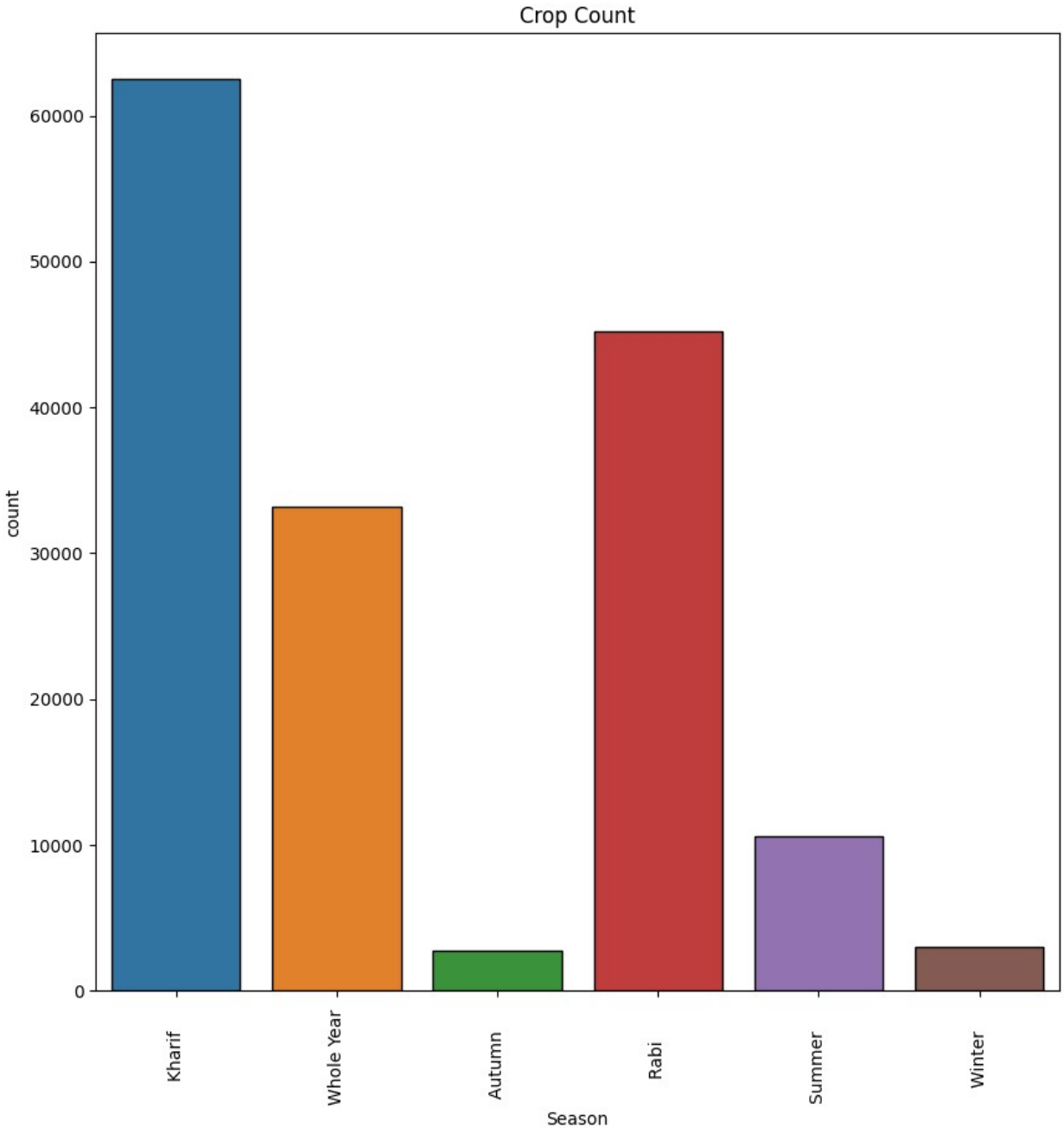


- 2003 & 2002 has the maximum count
- 2015 has the lowest

Season

```
plt.figure(figsize=(10,10))

sns.countplot(data = crop_new_2, x = 'Season', edgecolor = 'black')
```

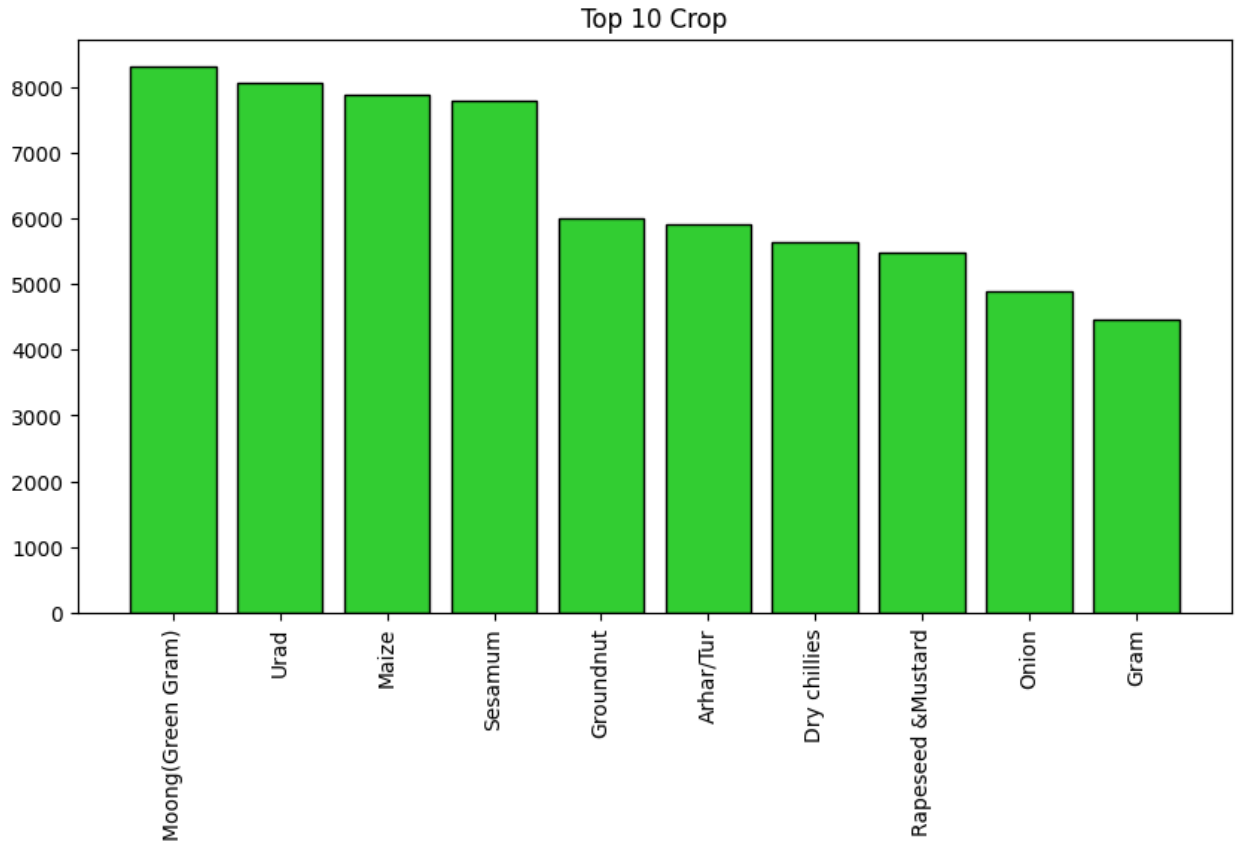
```
plt.title('Crop Count')

plt.xticks(rotation=90)
plt.show()
```



- Kharif Season has the maximum count.
- Autumn Season with the lowest.

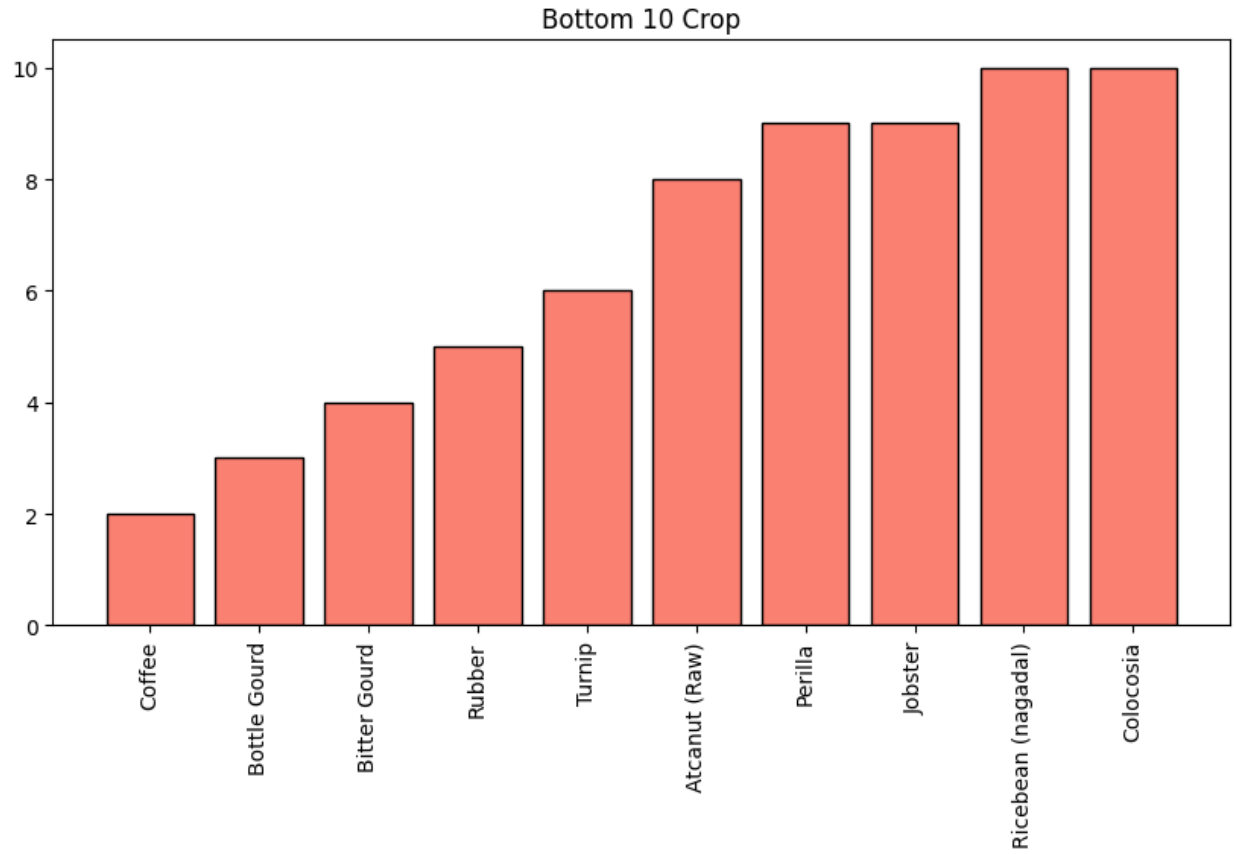
Crop

```
top10crop = crop_new_2.Crop.value_counts(ascending=False).head(10)

bottom10crop = crop_new_2.Crop.value_counts(ascending=True).head(10)

top10crop

plt.figure(figsize=(10,5))

plt.bar(top10crop.keys(),top10crop.values, color = 'limegreen',
edgecolor = 'black')

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




```
bottom10crop

plt.figure(figsize = (10,5))

plt.bar(bottom10crop.keys(),bottom10crop.values, color = 'salmon',
edgecolor = 'black')
```

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



- Moong(Green Gram) has maximum count.
- Coffee with the lowest count.

Bivariate Analysis

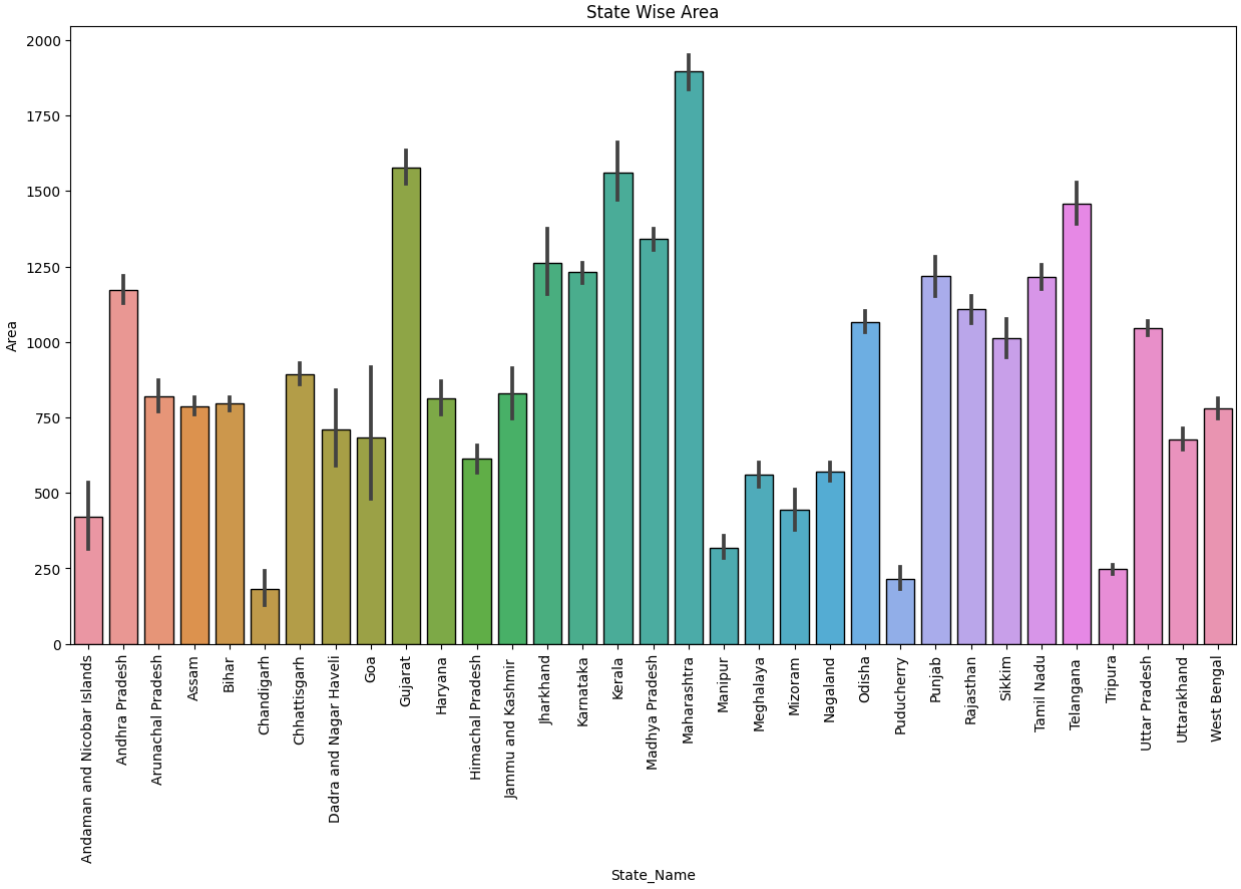
```
crop_new_2.head()
```

```
                   State_Name District_Name Crop_Year       Season  \
0  Andaman and Nicobar Islands      NICOBARS      2000  Kharif       \
2  Andaman and Nicobar Islands      NICOBARS      2000  Kharif
3  Andaman and Nicobar Islands      NICOBARS      2000  Whole Year
4  Andaman and Nicobar Islands      NICOBARS      2000  Whole Year
6  Andaman and Nicobar Islands      NICOBARS      2000  Whole Year

         Crop  Area  Production
0    Arecanut  1254        2000
2        Rice   102         321
```

```
3      Banana   176          641
4   Cashewnut   720          165
6  Dry ginger    36          100
```

State Wise Area

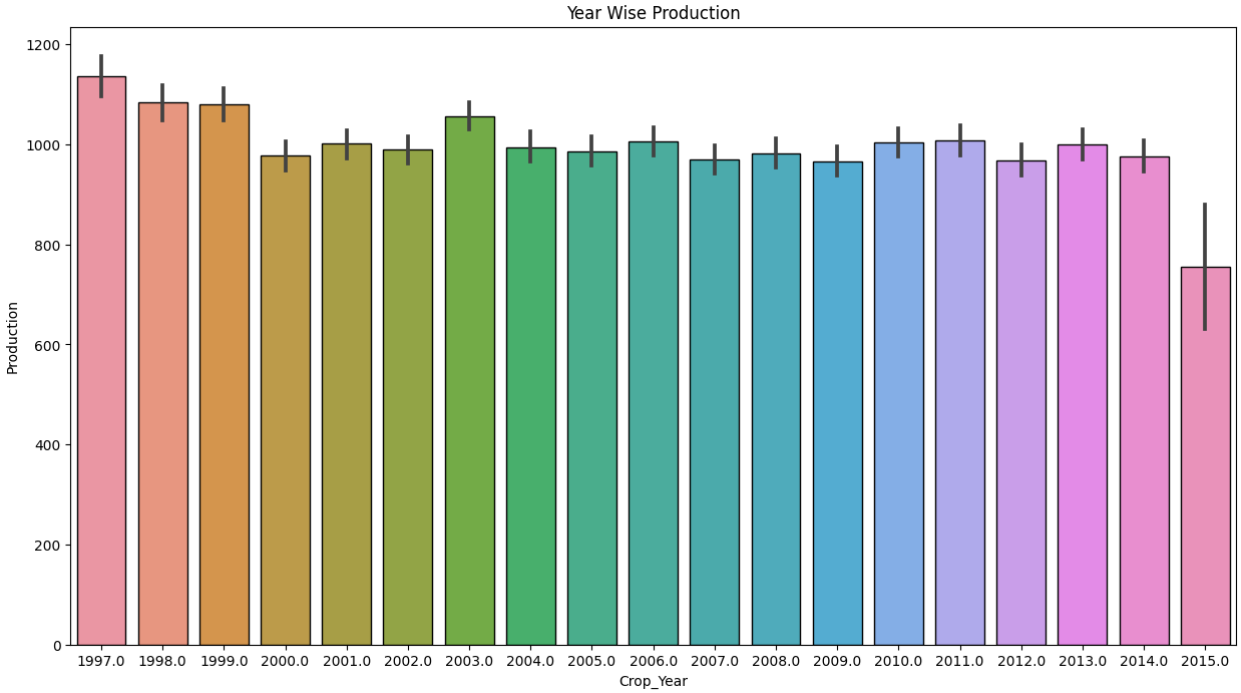
```
crop_new_2.groupby('State_Name')['Area'].sum().sort_values(ascending = False)

sns.barplot(data = crop_new_2, x = 'State_Name', y = 'Area', edgecolor = 'black')

plt.title('State Wise Area',fontdict = {'size':12})
plt.xticks(rotation = 90)
plt.show()
```



- Maharasthtra, Gujrat, Kerala has the maximum agricultural land.
- Chandigarh with the lowest land.

Year Wise Production.

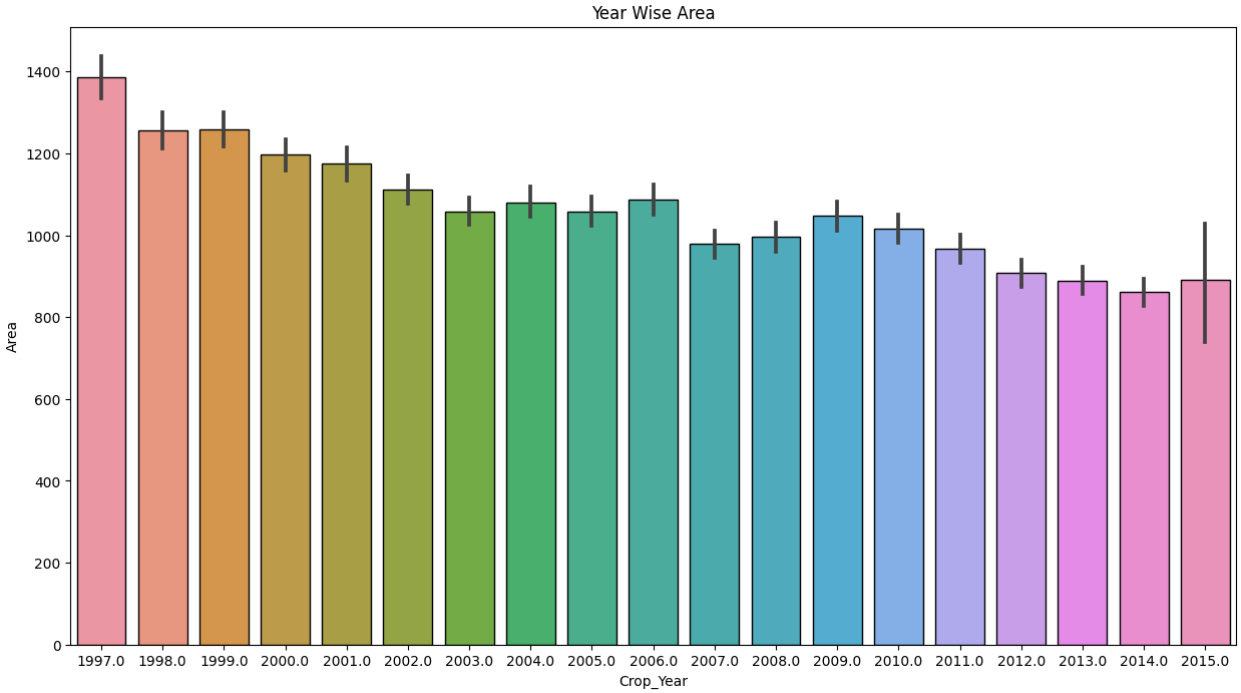
```
sns.barplot(data = crop_new_2, x = 'Crop_Year', y = 'Production',
edgecolor = 'black')
plt.title('Year Wise Production')
plt.show()
```



Since 1997 there is no big difference in Production except 2015.

Year Wise Area.

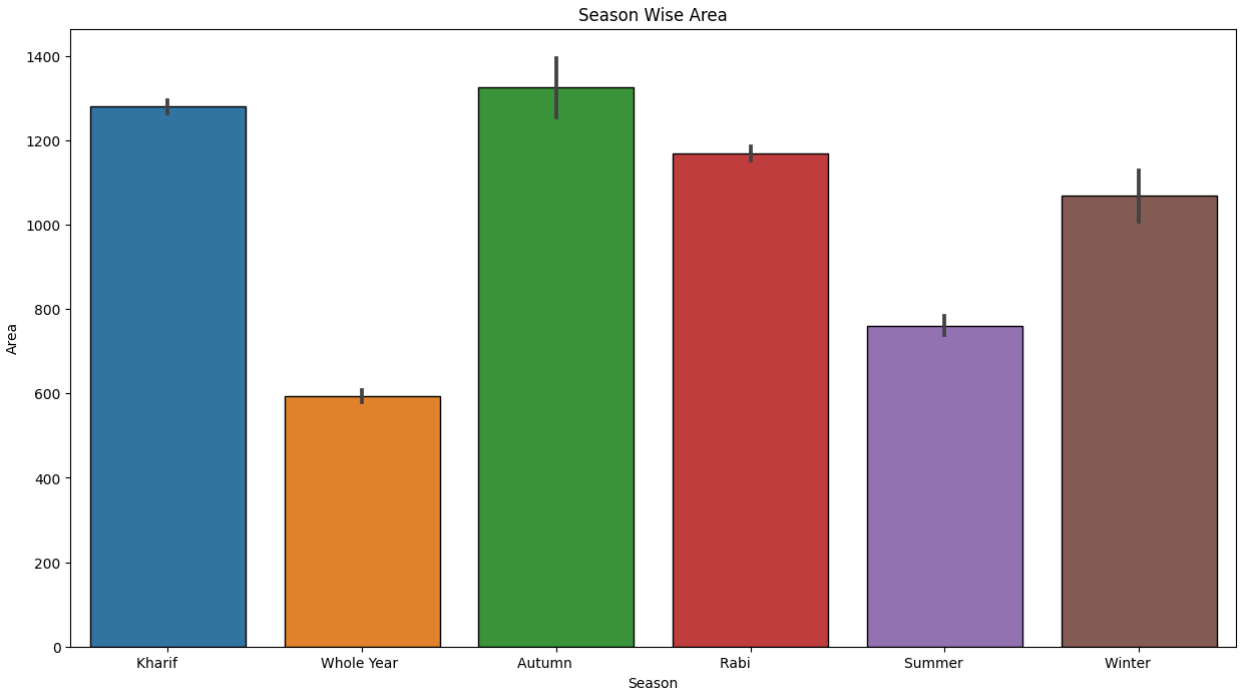
```
sns.barplot(data = crop_new_2, x = 'Crop_Year', y = 'Area', edgecolor
= 'black')
plt.title('Year Wise Area')
plt.show()
```



- 1997 to 2015 there is a decrease in agricultural land.

Season Wise Area.

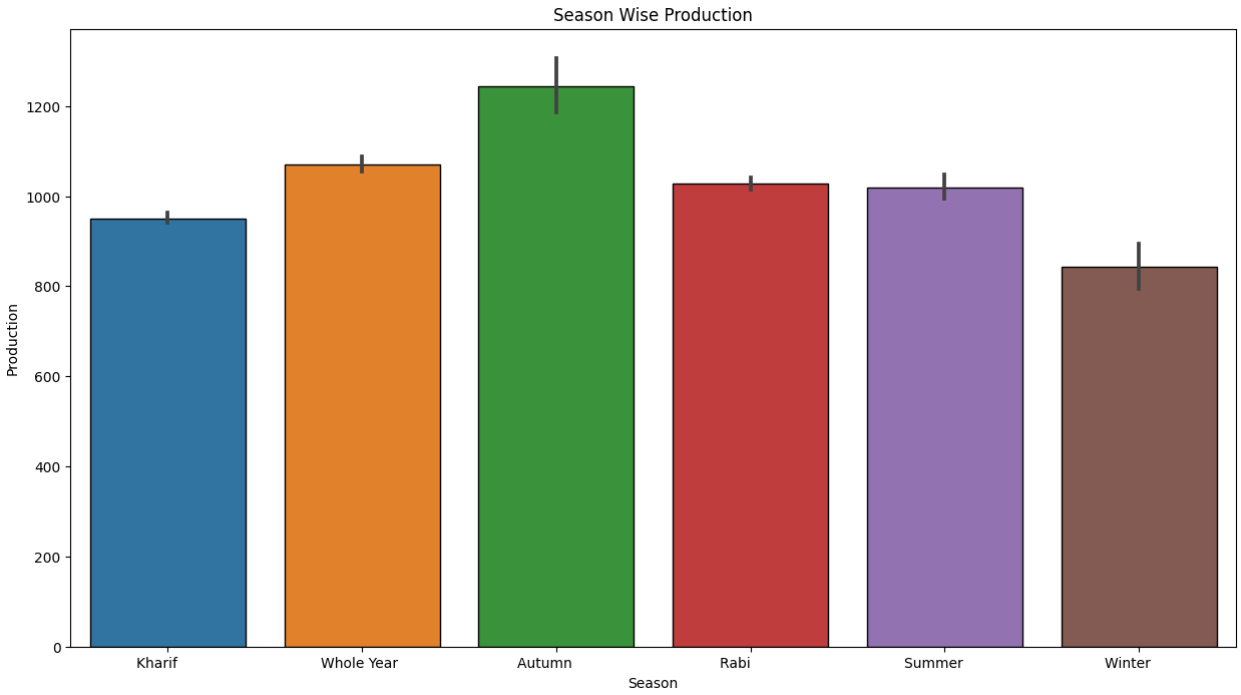
```
sns.barplot(data = crop_new_2, x = 'Season', y = 'Area', edgecolor =
'black')
plt.title('Season Wise Area')
plt.show()
```



- In Autumn agricultural areas are maximum.

Season Wise Production.

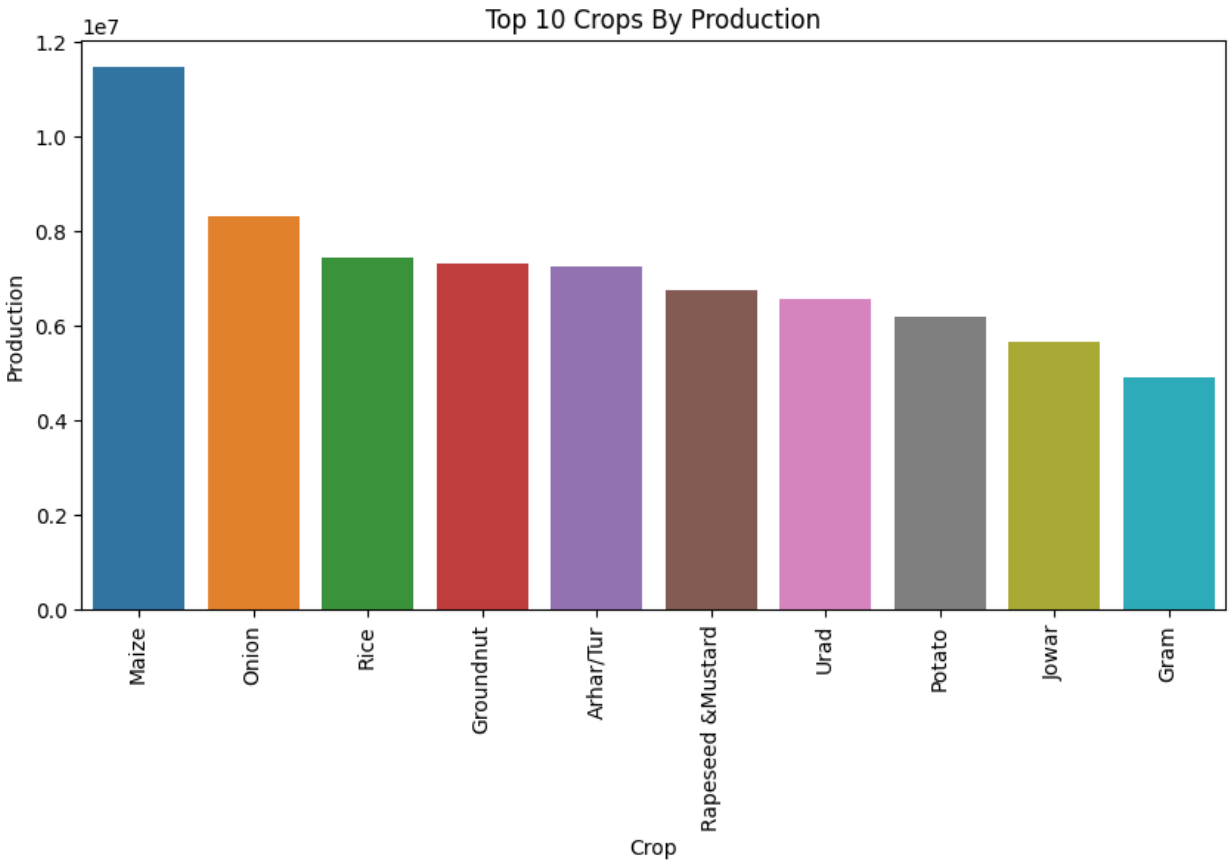
```
sns.barplot(data=crop_new_2, x='Season', y='Production',edgecolor=
'black')
plt.title('Season Wise Production')
plt.show()
```



- Production of crops peaks in Autumn.

Top10 Crops By Production.

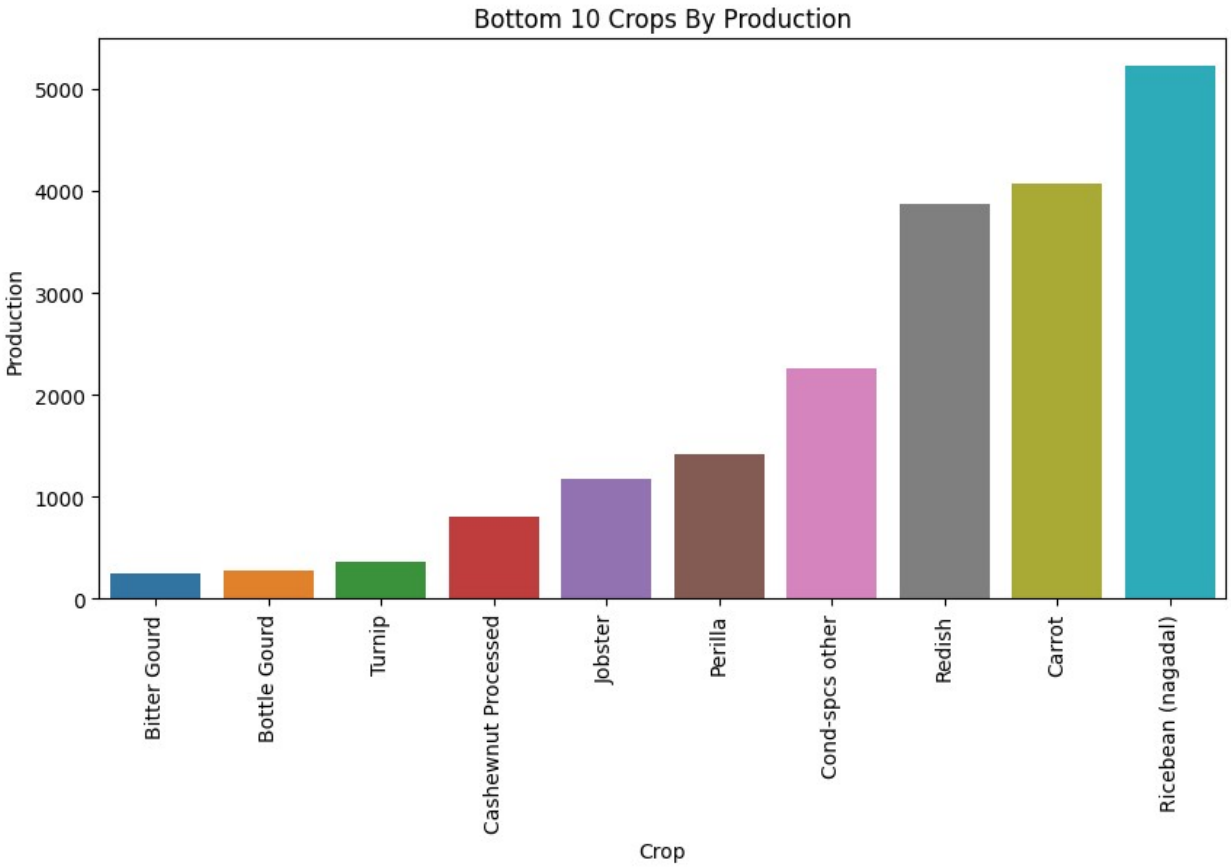
```
top10crp = crop_new_2.groupby('Crop')
['Production'].sum().sort_values(ascending = False).head(10)

plt.figure(figsize=(10, 5))

sns.barplot(x=top10crp.index, y=top10crp.values)

plt.title('Top 10 Crops By Production')

plt.xticks(rotation=90)

plt.xlabel('Crop')
plt.ylabel('Production')

plt.show()
```



- Maize & Onion has the highest production.

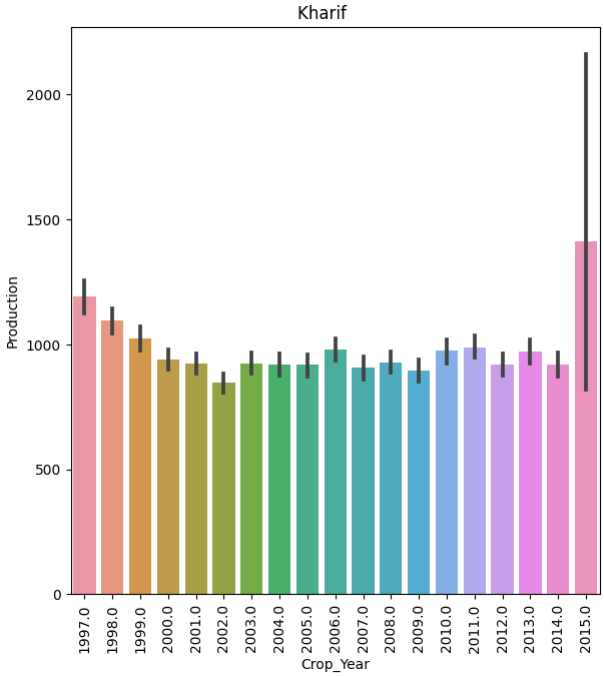
Bottom10 Crops By Production.

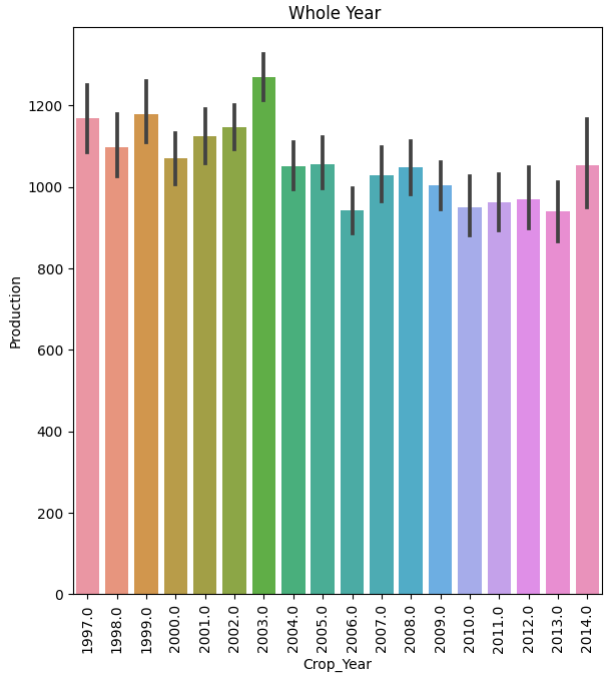
```
Bottom10crp = crop_new_2.groupby('Crop')
['Production'].sum().sort_values(ascending=True).head(10)

plt.figure(figsize=(10,5))
sns.barplot(x= Bottom10crp.index, y=Bottom10crp.values)
plt.title('Bottom 10 Crops By Production')

plt.xticks(rotation=90)
plt.xlabel("Crop")
plt.ylabel('Production')
plt.show()
```

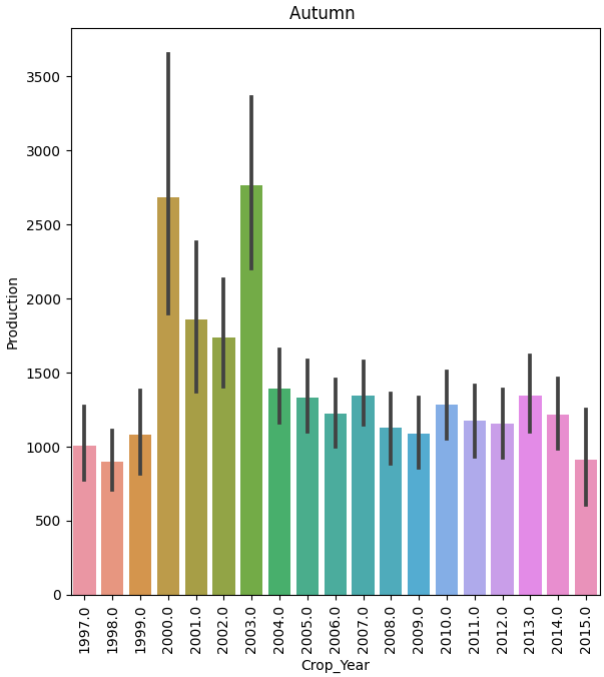


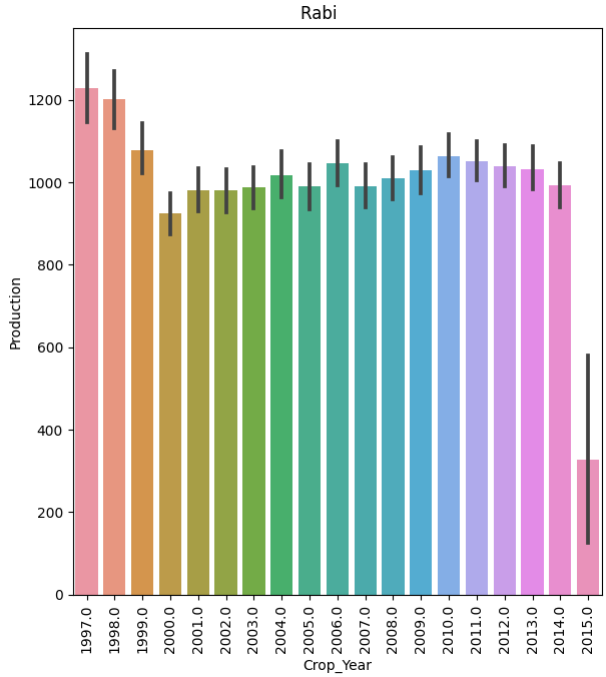
- Bitter Gourd has the lowest production.

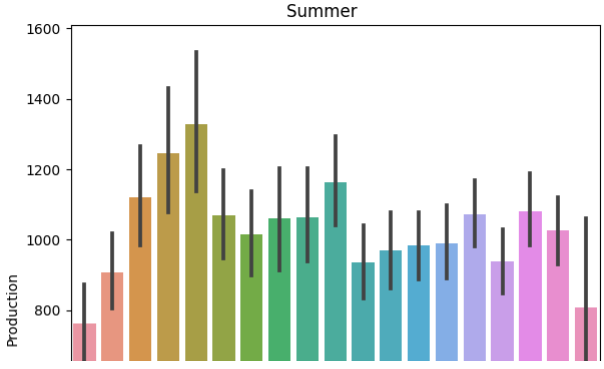
```
num = 1
plt.figure(figsize=(15,25))
for i in crop_new_2.Season.unique():
    plt.subplot(3,2,num)
    sns.barplot(data=crop_new_2[crop_new_2['Season']==i],
x='Crop_Year', y='Production')
    plt.xticks(rotation=90)
    plt.title(i)
    num+=1
```

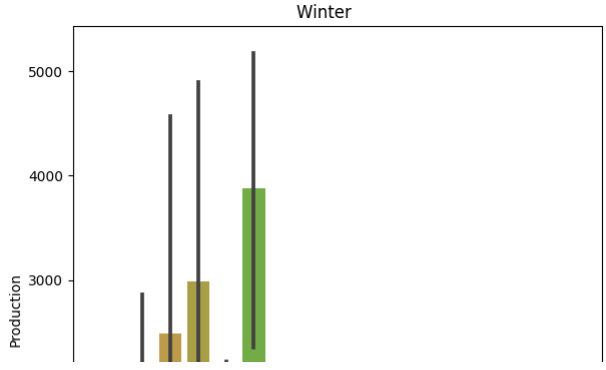

Kharif
Production
Crop_Year


Whole Year
Production
Crop_Year


Autumn
Production
Crop_Year


Rabi
Production
Crop_Year


Summer
Production


Winter
Production

- 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.

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, ie 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 correlated 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.