

# DATA ANALYSIS AND VISUALIZATION DONE BY JATIN KUMAR(18BCB0072)

## Importing libraries

In [50]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style='whitegrid')
sns.set_context("paper")
```

In [2]:

```
#reading data csv
crop_df = pd.read_csv("crop_production.csv")
crop_df.head()
```

Out[2]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0

## State-Wise Production - 1997-2014

In [3]:

```

fig, ax = plt.subplots(figsize=(25,65), sharex='col')
count = 1

for state in crop_df.State_Name.unique():
    plt.subplot(len(crop_df.State_Name.unique()),1,count)
    sns.lineplot(crop_df[crop_df.State_Name==state]['Crop_Year'],crop_df[crop_df.
    plt.subplots_adjust(hspace=2.2)
    plt.title(state)
    count+=1

```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword

## Zone-Wise Production - 1997-2014

In [4]:

```

north_india = ['Jammu and Kashmir', 'Punjab', 'Himachal Pradesh', 'Haryana', 'Utt
east_india = ['Bihar', 'Odisha', 'Jharkhand', 'West Bengal']
south_india = ['Andhra Pradesh', 'Karnataka', 'Kerala', 'Tamil Nadu', 'Telangana'
west_india = ['Rajasthan', 'Gujarat', 'Goa', 'Maharashtra', 'Goa']
central_india = ['Madhya Pradesh', 'Chhattisgarh']
north_east_india = ['Assam', 'Sikkim', 'Nagaland', 'Meghalaya', 'Manipur', 'Mizor
ut_india = ['Andaman and Nicobar Islands', 'Dadra and Nagar Haveli', 'Puducherry']

```

```
In [5]: def get_zonal_names(row):
        if row['State_Name'].strip() in north_india:
            val = 'North Zone'
        elif row['State_Name'].strip() in south_india:
            val = 'South Zone'
        elif row['State_Name'].strip() in east_india:
            val = 'East Zone'
        elif row['State_Name'].strip() in west_india:
            val = 'West Zone'
        elif row['State_Name'].strip() in central_india:
            val = 'Central Zone'
        elif row['State_Name'].strip() in north_east_india:
            val = 'NE Zone'
        elif row['State_Name'].strip() in ut_india:
            val = 'Union Terr'
        else:
            val = 'No Value'
        return val

crop_df['Zones'] = crop_df.apply(get_zonal_names, axis=1)
crop_df['Zones'].unique()
```

```
Out[5]: array(['Union Terr', 'South Zone', 'NE Zone', 'East Zone', 'North Zone',
              'Central Zone', 'West Zone'], dtype=object)
```

```
In [6]: fig, ax = plt.subplots(figsize=(25,30), sharex='col')
count = 1

for zone in crop_df.Zones.unique():
    plt.subplot(len(crop_df.Zones.unique()),1,count)
    sns.lineplot(crop_df[crop_df.Zones==zone]['Crop_Year'],crop_df[crop_df.Zones==zone]['Crop_Yield'])
    plt.subplots_adjust(hspace=0.6)
    plt.title(zone)
    count+=1
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

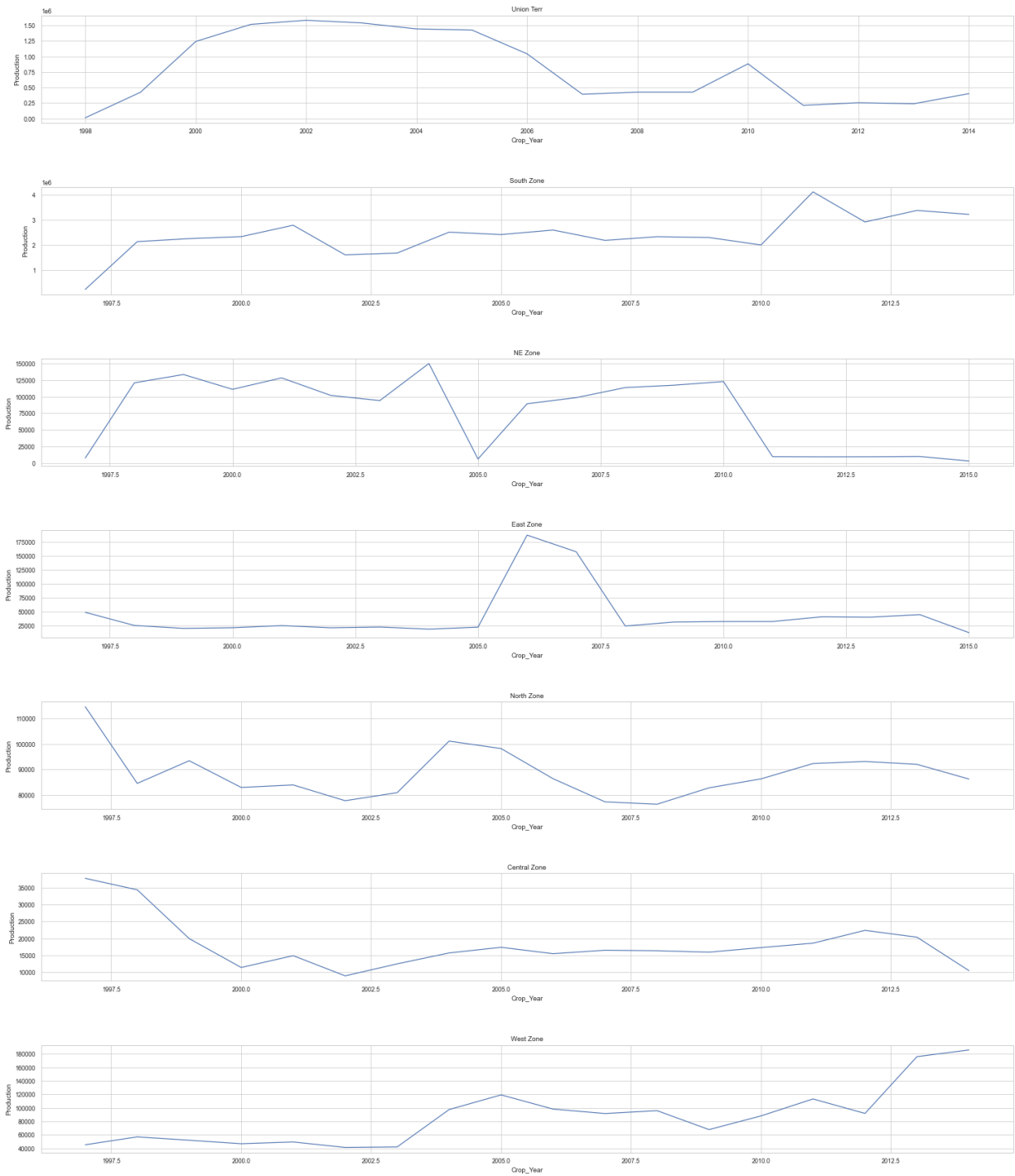
warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



We see 2005 to be of some significance as there's activity happening before and after 2005. Similarly we see slight changes around 2011 for Central, West and North Zone which differ for East and North East Zone.

Let's see what happened in 2005 and 2011. Excerpts from the reports published by the IMD for those respective years.

#### 2005

During the season, rainfall was not well distributed in time. Rainfall over the country was below normal

(12% below LPA) in June. However, monsoon was active in July with excess rainfall (14% above LPA).

Monsoon was subdued in August with a large deficiency of 28% of LPA. In September (rainfall +17% above LPA), monsoon became active again helping a timely revival and improving the seasonal rainfall situation over the country.

#### 2011

Out of the total 36 meteorological subdivisions, 33 subdivisions constituting 92% of the total area of the country received excess/normal season rainfall and the remaining 3 subdivisions (Arunachal Pradesh, Assam & Meghalaya, and NMMT constituting 8% of the total area of the country) received deficient season rainfall.

Ref:

<https://reliefweb.int/report/india/india-meteorological-department-southwest-monsoon-2005-end-season-report>

<https://reliefweb.int/report/india/southwest-monsoon-2011-end-season-report>

## Zone-Wise Production - Total

In [11]:

```
zone_df = crop_df.groupby(by='Zones')['Production'].sum().reset_index().sort_values(
zone_df.head()
```

Out[11]:

	Zones	Production
4	South Zone	1.284797e+11
3	North Zone	4.365489e+09
6	West Zone	2.575008e+09
2	NE Zone	2.165305e+09
1	East Zone	1.936070e+09

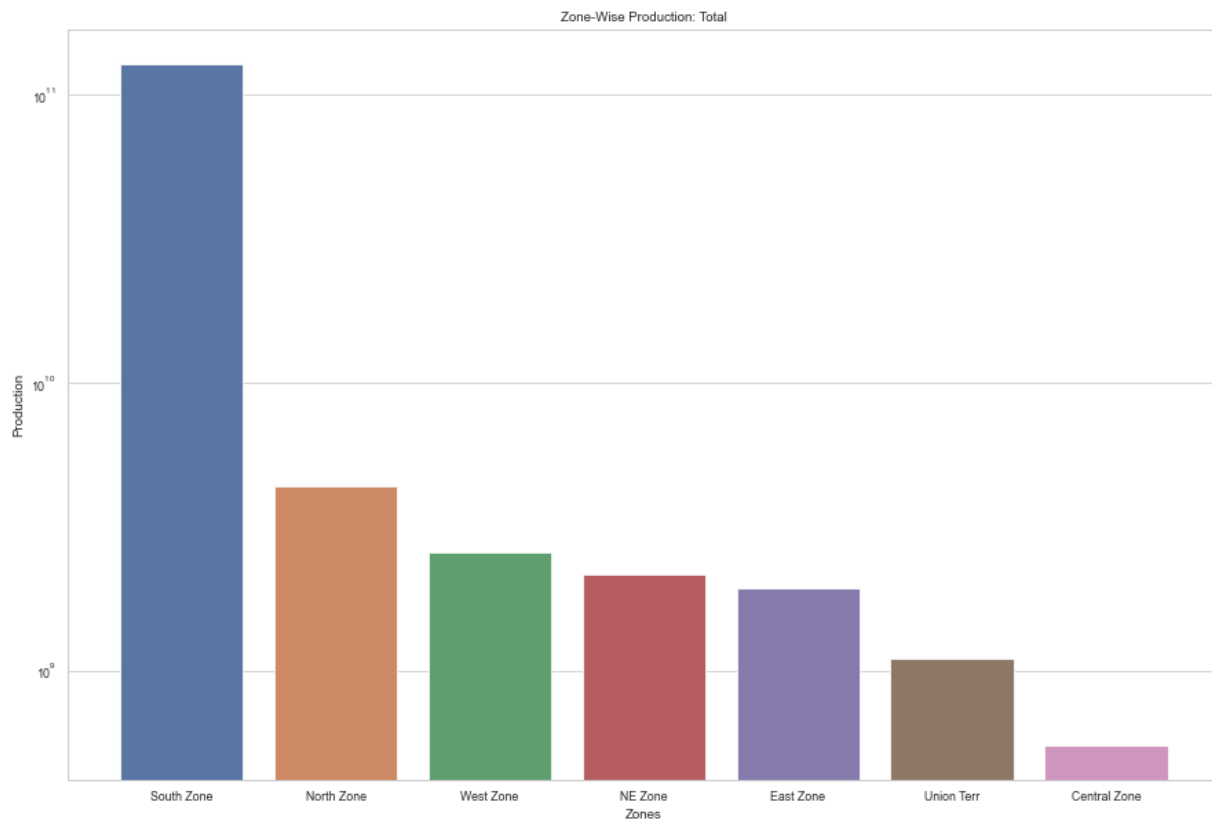
In [12]:

```
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(zone_df.Zones, zone_df.Production)
plt.yscale('log')
plt.title('Zone-Wise Production: Total')
plt.plot()
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[12]: []



# South Zone Production

In [13]:

```
south_zone = crop_df[(crop_df["Zones"] == 'South Zone')]
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(south_zone.State_Name, south_zone.Production, errwidth=0)
plt.yscale('log')
plt.title('Southern-Zone Production')

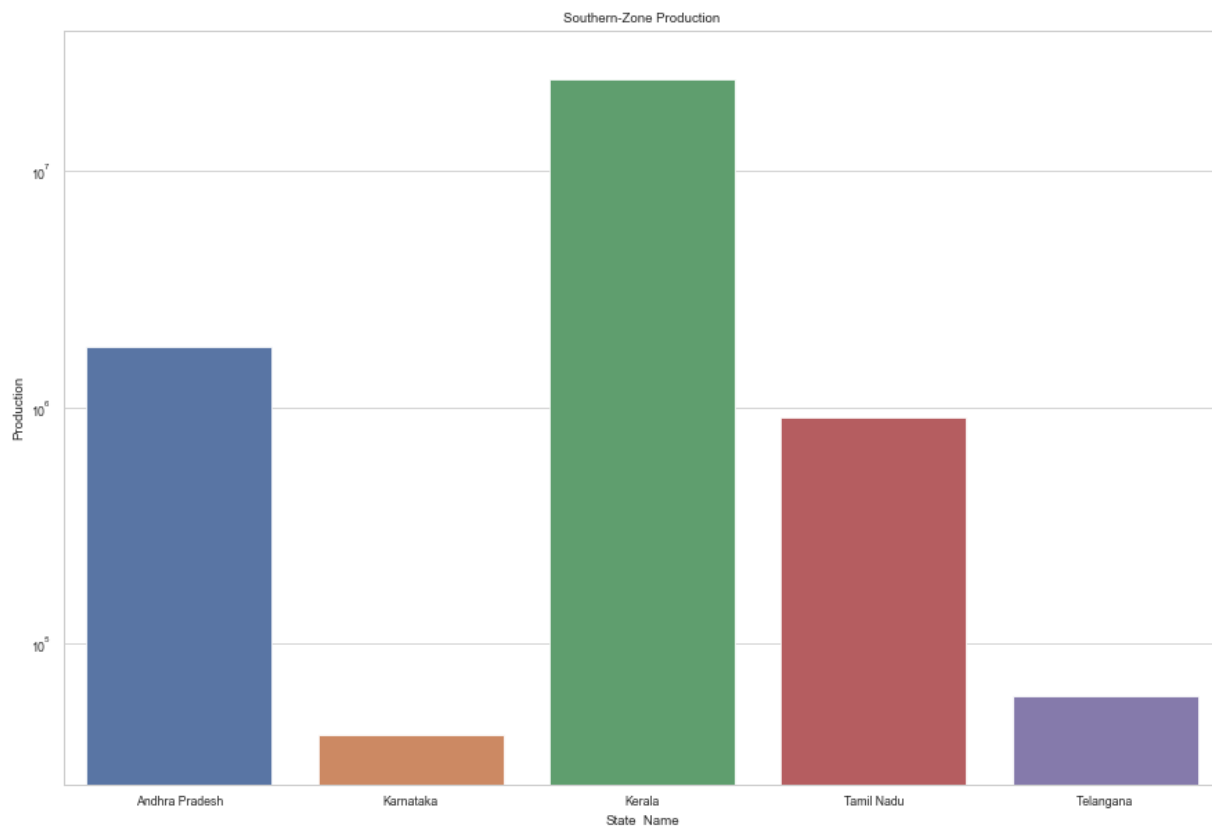
south_zone.groupby(by='State_Name')['Production'].sum().reset_index().sort_values
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

Out[13]:

	State_Name	Production
2	Kerala	9.788005e+10
0	Andhra Pradesh	1.732459e+10
3	Tamil Nadu	1.207644e+10
1	Karnataka	8.634298e+08
4	Telangana	3.351479e+08





## South Zone - Highest Produced Crops

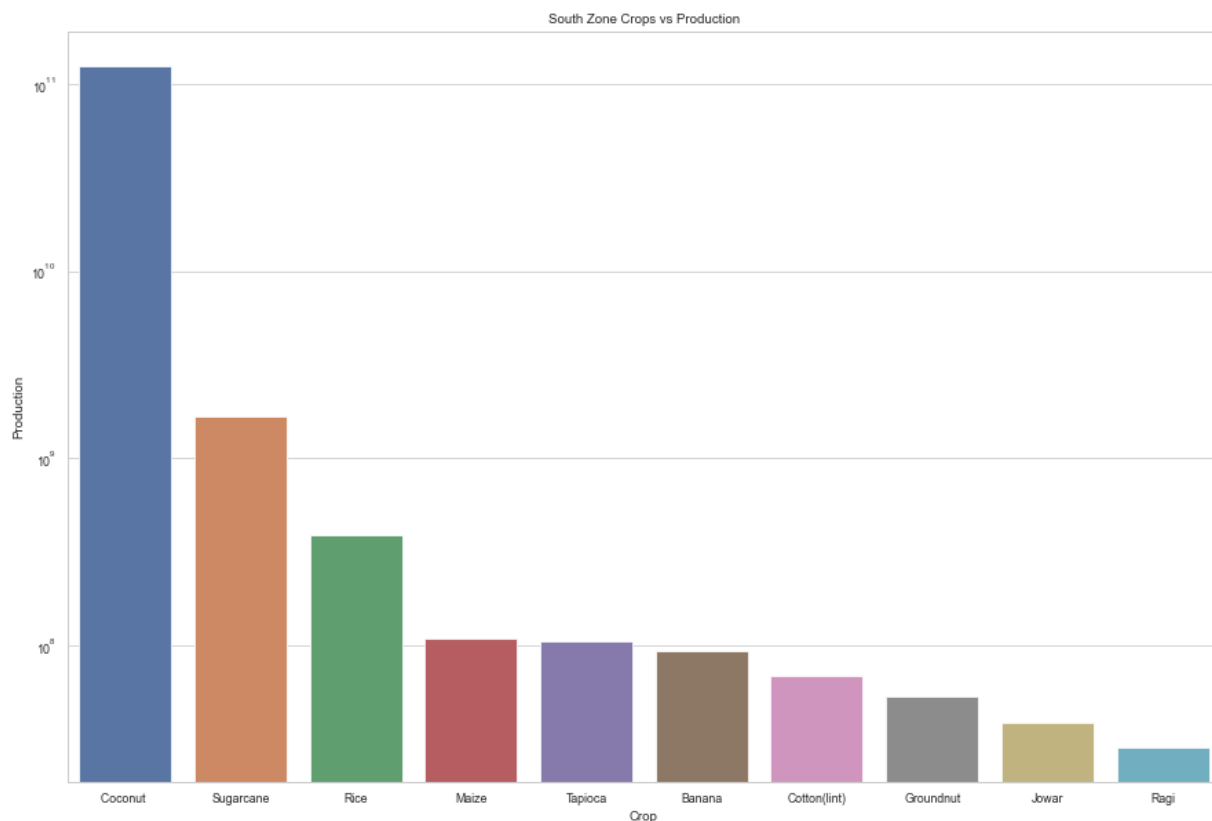
In [14]:

```
df = south_zone.groupby(by='Crop')['Production'].sum().reset_index().sort_values(
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.Crop, df.Production, errwidth=0)
plt.yscale('log')
plt.title('South Zone Crops vs Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[14]: Text(0.5, 1.0, 'South Zone Crops vs Production')



As we can see coconut is produced mostly in the south zone.

## Overall Crop Production

In [19]:

```

crop = crop_df.groupby(by='Crop')['Production'].sum().reset_index().sort_values(
crop
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(crop.Crop, crop.Production,errwidth=0)
plt.yscale('log')
plt.title('Overall Crops vs Production')

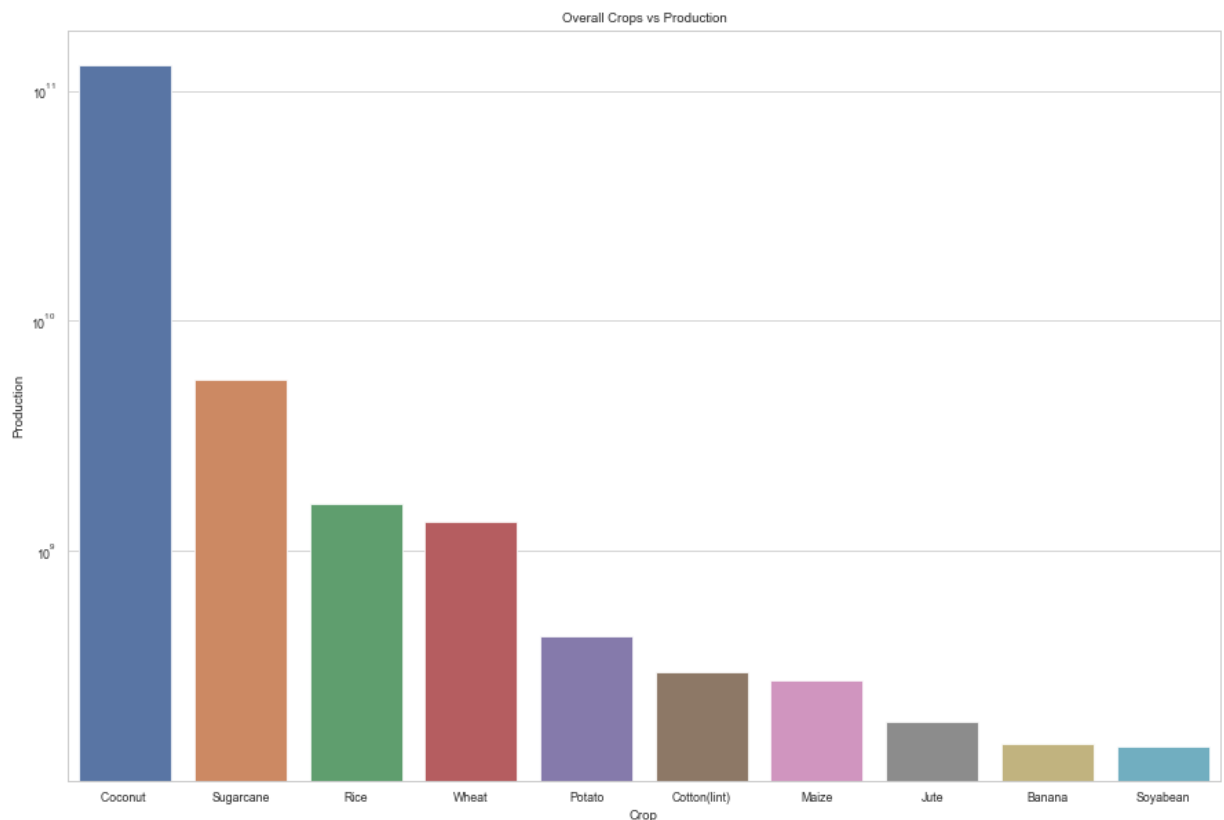
```

```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn
_decorators.py:36: FutureWarning: Pass the following variables as keyword arg
s: x, y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in an error
or misinterpretation.
  warnings.warn(

```

Out[19]: Text(0.5, 1.0, 'Overall Crops vs Production')



The numbers are displayed for all crops. Also, Coconut can be harvested year long depending on its use. More details on cocunut harvesting can be found here <http://vikaspedia.in/agriculture/crop-production/package-of-practices/plantation-crops/coconut/coconut-cultivation-practices> (<http://vikaspedia.in/agriculture/crop-production/package-of-practices/plantation-crops/coconut/coconut-cultivation-practices>)

```
In [20]: (crop_df[(crop_df['Season'] == 'Whole Year')].Crop.unique()) & set(crop_df[(crop_
```

```
Out[20]: {'Arecanut',  
          'Arhar/Tur',  
          'Bajra',  
          'Banana',  
          'Barley',  
          'Black pepper',  
          'Blackgram',  
          'Brinjal',  
          'Cabbage',  
          'Cardamom',  
          'Cashewnut',  
          'Castor seed',  
          'Coconut ',  
          'Coriander',  
          'Cotton(lint)',  
          'Cowpea(Lobia)',  
          'Dry chillies',  
          'Dry ginger',  
          'Garlic',  
          'Ginger',  
          'Gram',  
          'Grapes',  
          'Groundnut',  
          'Guar seed',  
          'Horse-gram',  
          'Jowar',  
          'Jute & mesta',  
          'Kapas',  
          'Khesari',  
          'Linseed',  
          'Maize',  
          'Mango',  
          'Masoor',  
          'Mesta',  
          'Moong(Green Gram)',  
          'Moth',  
          'Niger seed',  
          'Oilseeds total',  
          'Onion',  
          'Orange',  
          'Papaya',  
          'Peas & beans (Pulses)',  
          'Pome Granet',  
          'Potato',  
          'Pulses total',  
          'Ragi',  
          'Rapeseed &Mustard',  
          'Rice',  
          'Safflower',  
          'Sannhamp',  
          'Sesamum',  
          'Small millets',  
          'Soyabean',  
          'Sugarcane',
```

```
'Sunflower',
'Sweet potato',
'Tapioca',
'Tea',
'Tobacco',
'Tomato',
'Total foodgrain',
'Turmeric',
'Urad',
'Wheat',
'other oilseeds'}
```

## Classifying some of the most important crops into the either Kharif or Rabi, and remaining clubbed as 'Others'

```
In [21]: Kharif = ['Bajra', 'Jowar', 'Maize', 'Millet', 'Rice', 'Soybean', 'Fruits', 'Muskmelon',
'Urad', 'Cotton(lint)', 'Cowpea(Lobia)', 'Moong(Green Gram)', 'Guar seed', 'Moth', 'Tor
Rabi = ['Barley', 'Gram', 'Rapeseed & Mustard', 'Masoor', 'Coriander', 'Sunflower',
'Onion', 'Sweet potato', 'Potato', 'Peas & beans (Pulses)', 'Oilseeds total',
'Sesamum', 'Safflower', 'Castor seed', 'Linseed', 'Soyabean']

def change_crop_seasons(row):
    if row['Crop'].strip() in Kharif:
        val = 'Kharif'
    elif row['Crop'].strip() in Rabi:
        val = 'Rabi'
    else:
        val = 'Others'
    return val
crop_df['Updated_Crop_Season'] = crop_df.apply(change_crop_seasons, axis=1)
crop_df['Updated_Crop_Season'].unique()
```

```
Out[21]: array(['Others', 'Kharif', 'Rabi'], dtype=object)
```

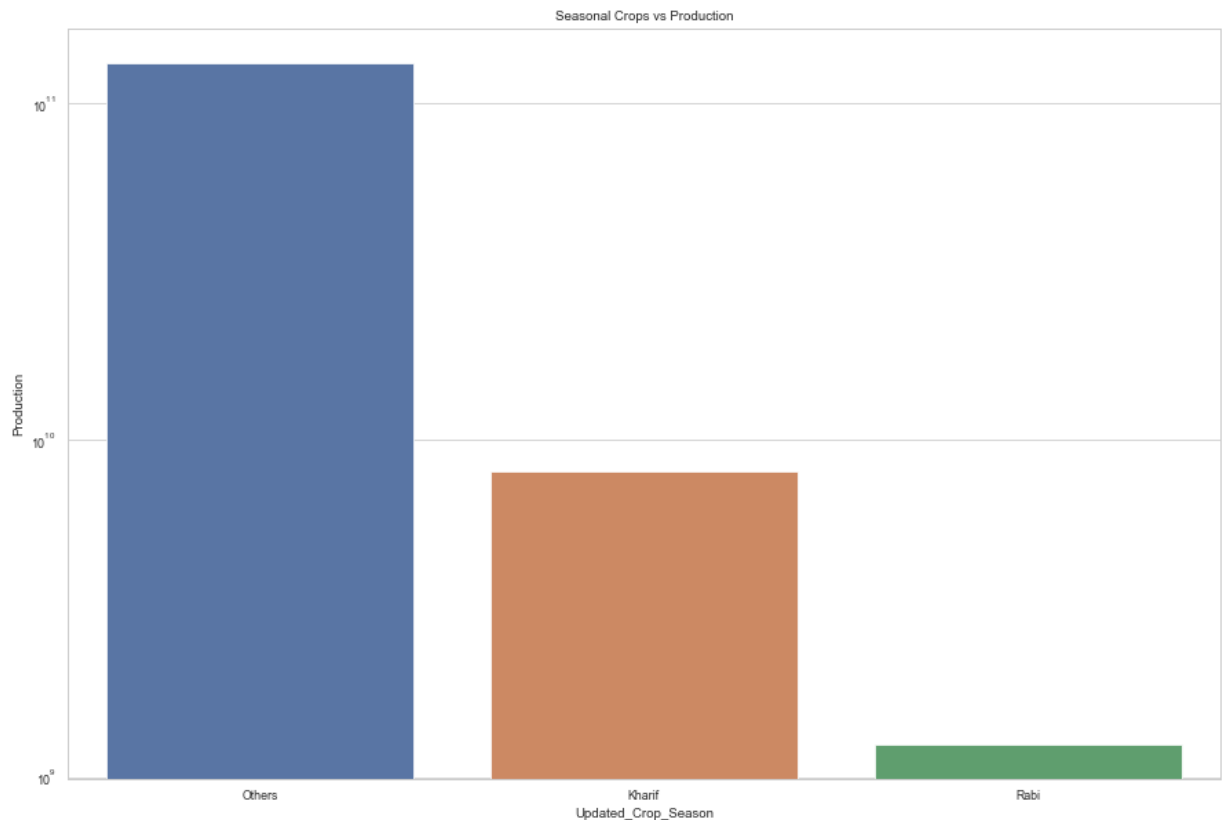
## Season vs Production

```
In [22]: season = crop_df.groupby(by='Updated_Crop_Season')['Production'].sum().reset_index()
season
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(season.Updated_Crop_Season, season.Production, errwidth=0)
plt.yscale('log')
plt.title('Seasonal Crops vs Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[22]: Text(0.5, 1.0, 'Seasonal Crops vs Production')



**Since The Indian cropping season is classified into two main seasons-(i) Kharif and (ii) Rabi based on the monsoon, Let's look at them**

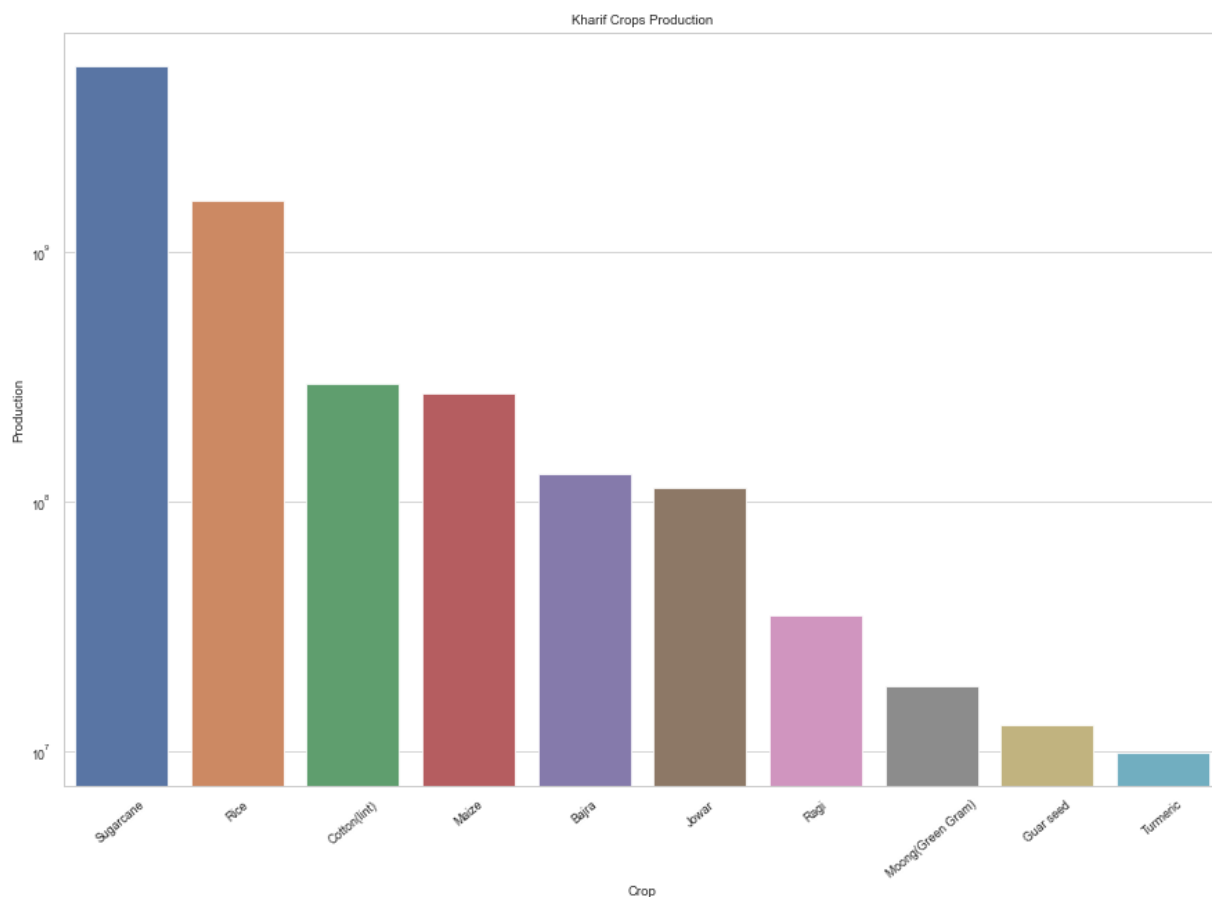
## A. Kharif

```
In [23]: kharif_df = crop_df[(crop_df['Updated_Crop_Season'] == 'Kharif')]
df = kharif_df.groupby(by='Crop')['Production'].sum().reset_index().sort_values(
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.Crop, df.Production, errwidth=0)
plt.yscale('log')
plt.xticks(rotation=40)
plt.title('Kharif Crops Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

Out[23]: Text(0.5, 1.0, 'Kharif Crops Production')



Sugarcane, Rice and Cotton lead in terms of Kharif crop production.

# ZONE WISE PRODUCTION OF SUGARCANE

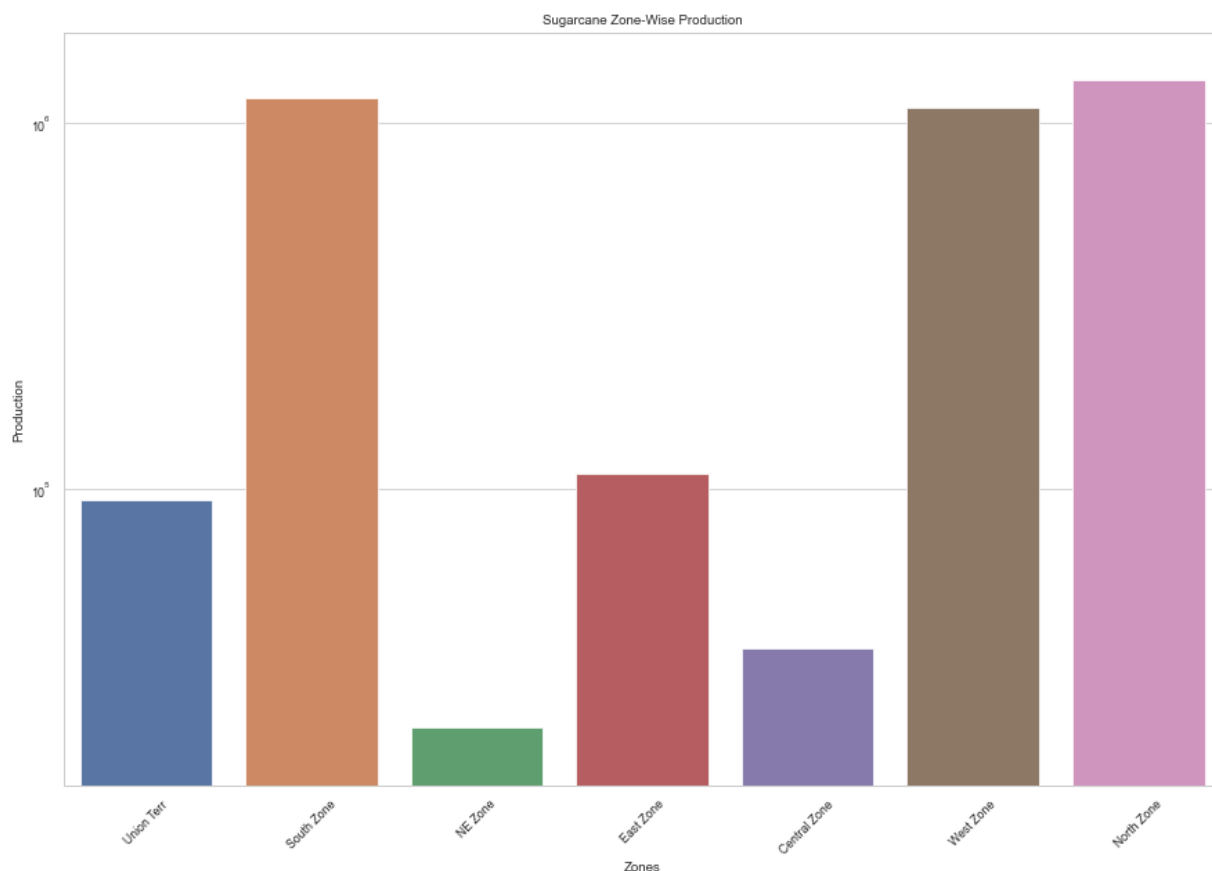
```
In [25]: sugarcane_df = kharif_df[(kharif_df['Crop'] == 'Sugarcane')]
sugarcane_df.head()

fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(sugarcane_df.Zones, sugarcane_df.Production, errwidth=0)
plt.yscale('log')
plt.xticks(rotation=45)
plt.title('Sugarcane Zone-Wise Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[25]: Text(0.5, 1.0, 'Sugarcane Zone-Wise Production')
```



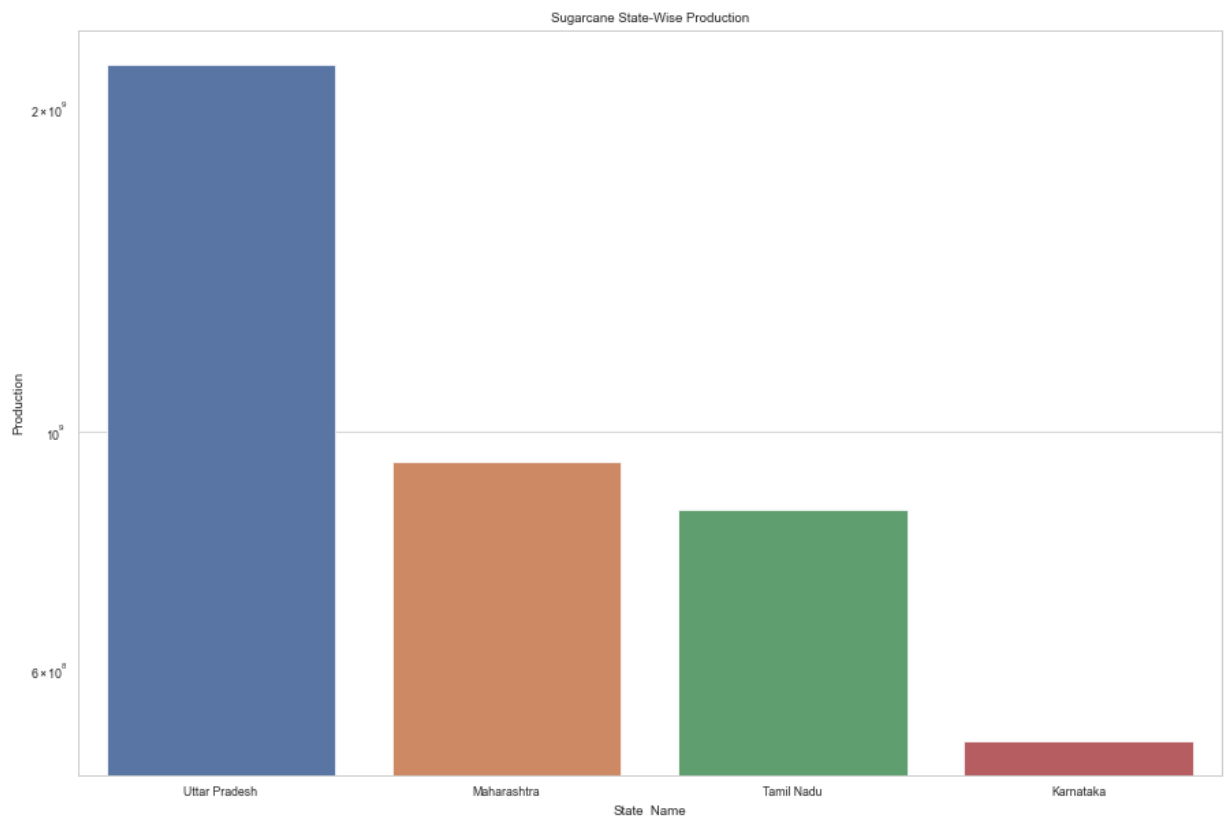
## STATE WISE PRODUCTION OF SUGARCANE

```
In [26]: df = sugarcane_df.groupby(by='State_Name')['Production'].sum().reset_index().sort
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.State_Name.head(4), df.Production.head(4),errwidth=0)
plt.yscale('log')
plt.title('Sugarcane State-Wise Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[26]: Text(0.5, 1.0, 'Sugarcane State-Wise Production')



**Since Uttar Pradesh leads in this, lets look at the agricultural area distribution.**



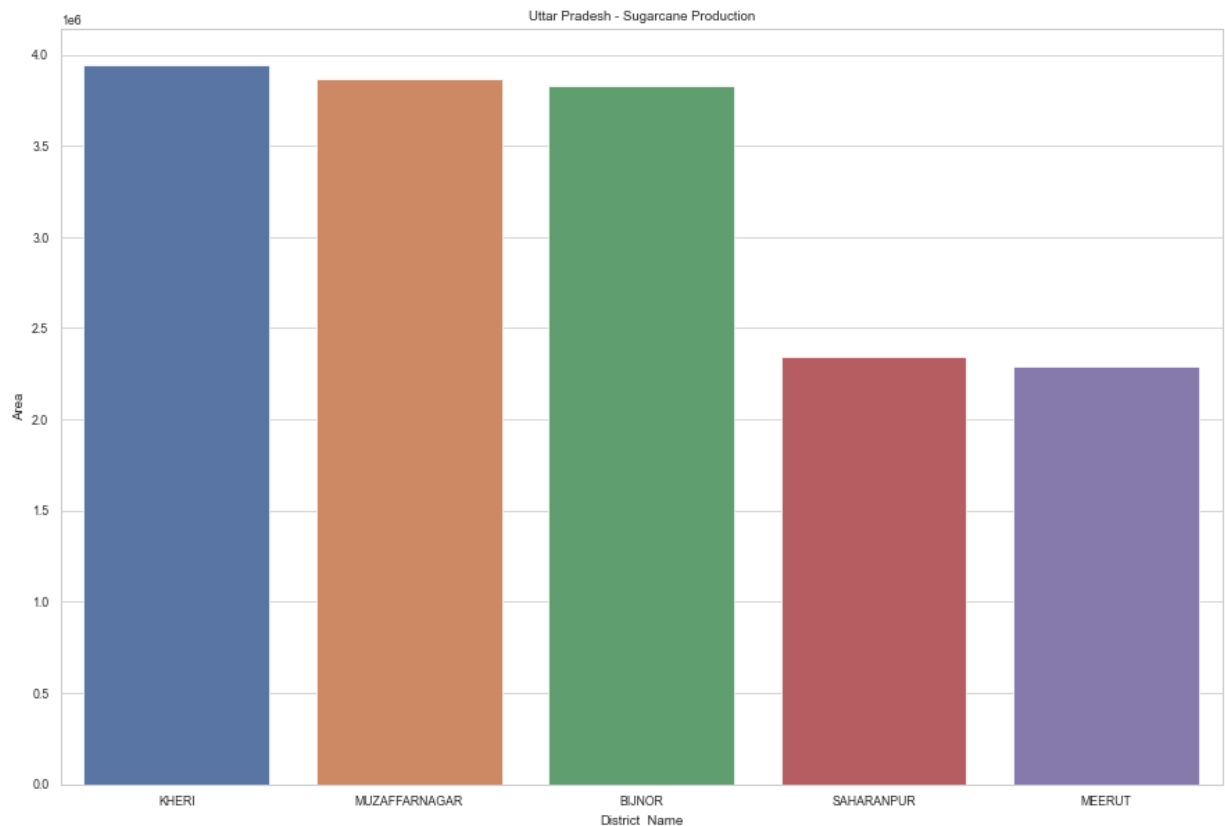
```
In [27]: uttarpr_df = sugarcane_df[(sugarcane_df['State_Name'] == 'Uttar Pradesh')]
df = uttarpr_df.groupby(by=['District_Name', 'Crop'])['Area'].sum().reset_index()
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.District_Name.head(5), df.Area.head(5),errwidth=0)
plt.title('Uttar Pradesh - Sugarcane Production')
df.head(5)
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[27]:

	District_Name	Crop	Area
45	KHERI	Sugarcane	3944916.0
57	MUZAFFARNAGAR	Sugarcane	3867761.0
16	BIJNOR	Sugarcane	3827478.0
62	SAHARANPUR	Sugarcane	2340820.0
54	MEERUT	Sugarcane	2288688.0



This data will help our admins to grab more sugarcane based on demand in market.

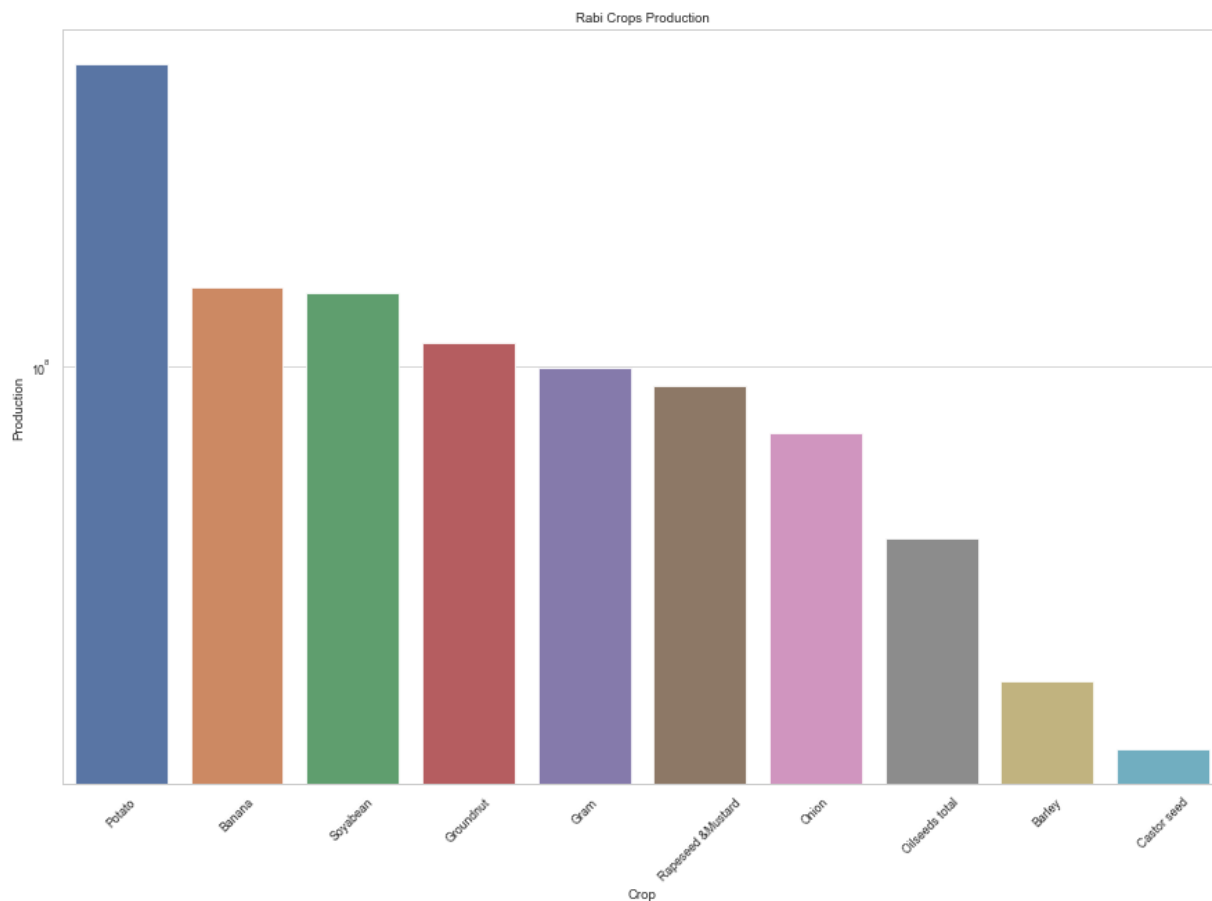
## B. Rabi

```
In [28]: rabi_df = crop_df[(crop_df['Updated_Crop_Season'] == 'Rabi')]
df = rabi_df.groupby(by='Crop')['Production'].sum().reset_index().sort_values(by=
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.Crop, df.Production, errwidth=0)
plt.yscale('log')
plt.xticks(rotation=45)
plt.title('Rabi Crops Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[28]: Text(0.5, 1.0, 'Rabi Crops Production')
```



Potato, Banana and Soyabean lead in terms of Rabi crop production.

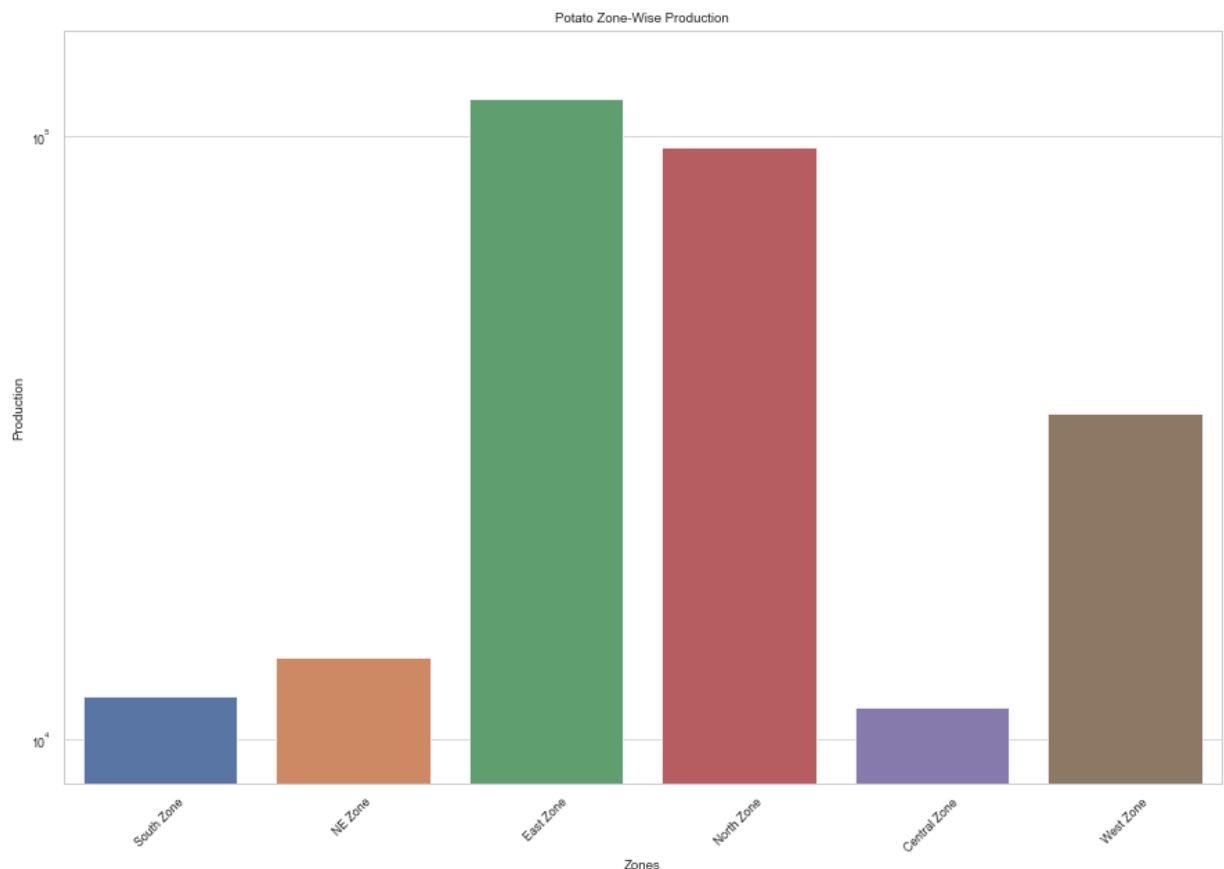
## Zonal distribution of Potato

```
In [29]: potato_df = rabi_df[(rabi_df['Crop'] == 'Potato')]
potato_df.head()

fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(potato_df.Zones, potato_df.Production, errwidth=0)
plt.yscale('log')
plt.xticks(rotation=45)
plt.title('Potato Zone-Wise Production')
```

```
c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg
s: x, y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in an error
or misinterpretation.
  warnings.warn(
```

```
Out[29]: Text(0.5, 1.0, 'Potato Zone-Wise Production')
```



East leads in Potato production followed closely by North. Let's take a look at the state-wise distribution and see if we have data that confirms the stats provided in this link -

<https://www.mapsofindia.com/top-ten/india-crops/potato.html> (<https://www.mapsofindia.com/top-ten/india-crops/potato.html>)

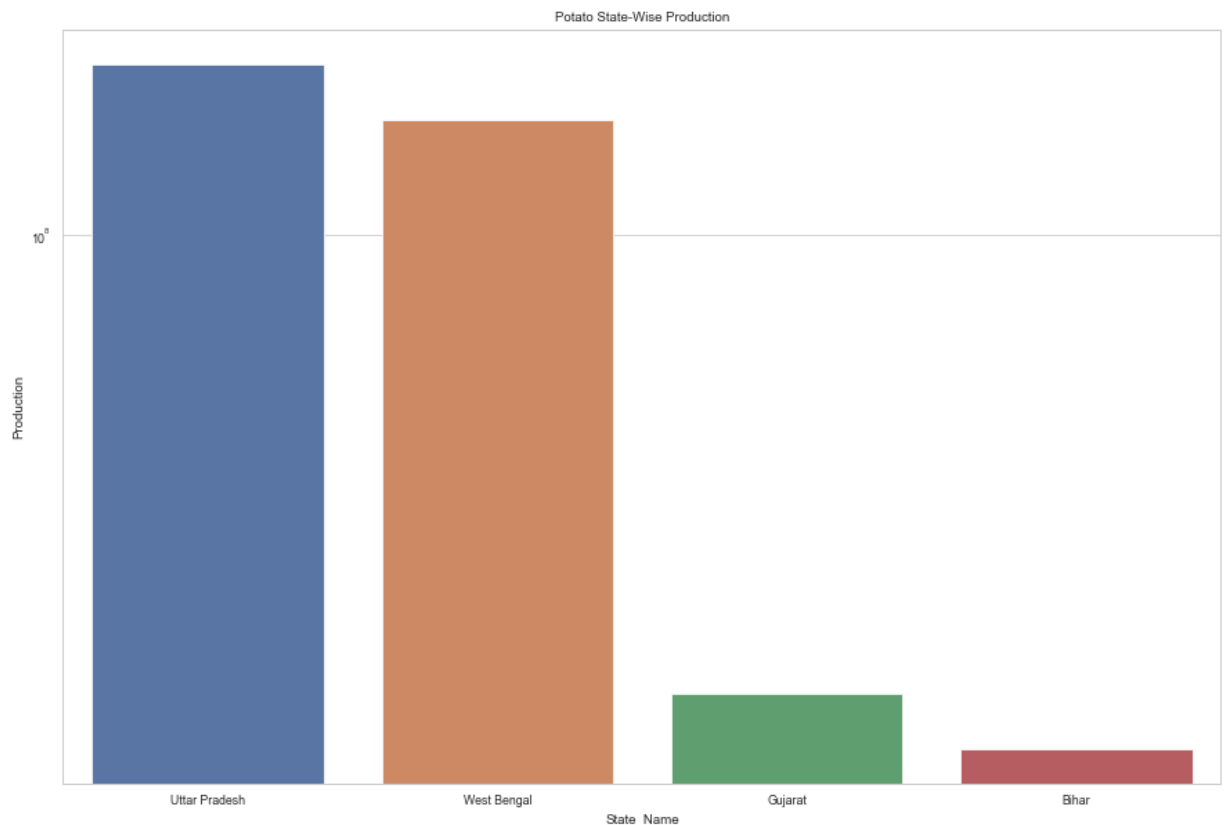
## State wise potato distribution

```
In [30]: df = potato_df.groupby(by='State_Name')['Production'].sum().reset_index().sort_values(
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.State_Name.head(4), df.Production.head(4),errwidth=0)
plt.yscale('log')
plt.title('Potato State-Wise Production')
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[30]: Text(0.5, 1.0, 'Potato State-Wise Production')



## Potato production in Uttar Pradesh

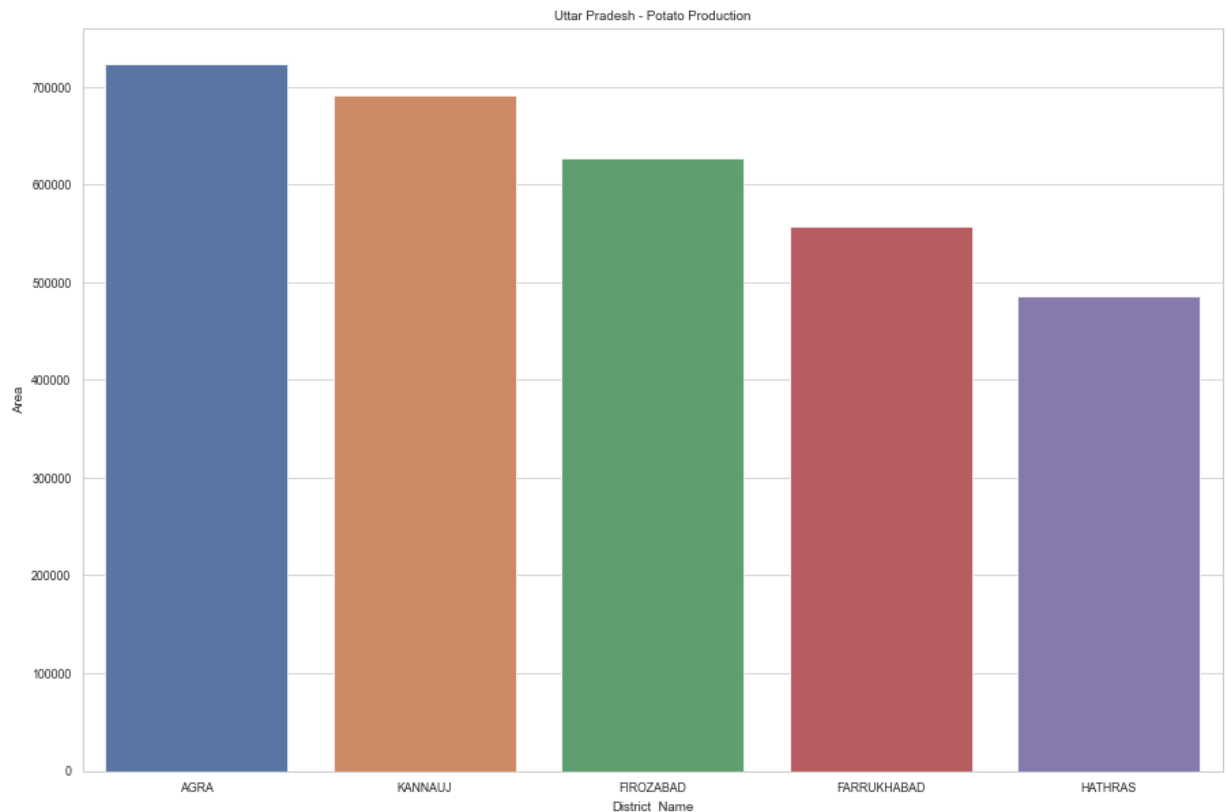
```
In [31]: uttarpr_df = potato_df[(potato_df['State_Name'] == 'Uttar Pradesh')]
df = uttarpr_df.groupby(by=['District_Name', 'Crop'])['Area'].sum().reset_index()
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.District_Name.head(5), df.Area.head(5), errwidth=0)
plt.title('Uttar Pradesh - Potato Production')
df.head(5)
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

Out[31]:

	District_Name	Crop	Area
0	AGRA	Potato	723954.0
40	KANNAUJ	Potato	691435.0
27	FIROZABAD	Potato	627652.0
25	FARRUKHABAD	Potato	557135.0
36	HATHRAS	Potato	485722.0



Thus our management team can always rely on these above districts in UP for potato collection and distribution.

## **Agricultural Area in India statewise**

```
In [32]: df = crop_df.groupby(by='State_Name')['Area'].sum().reset_index().sort_values(by='Area', ascending=False)
df.head()

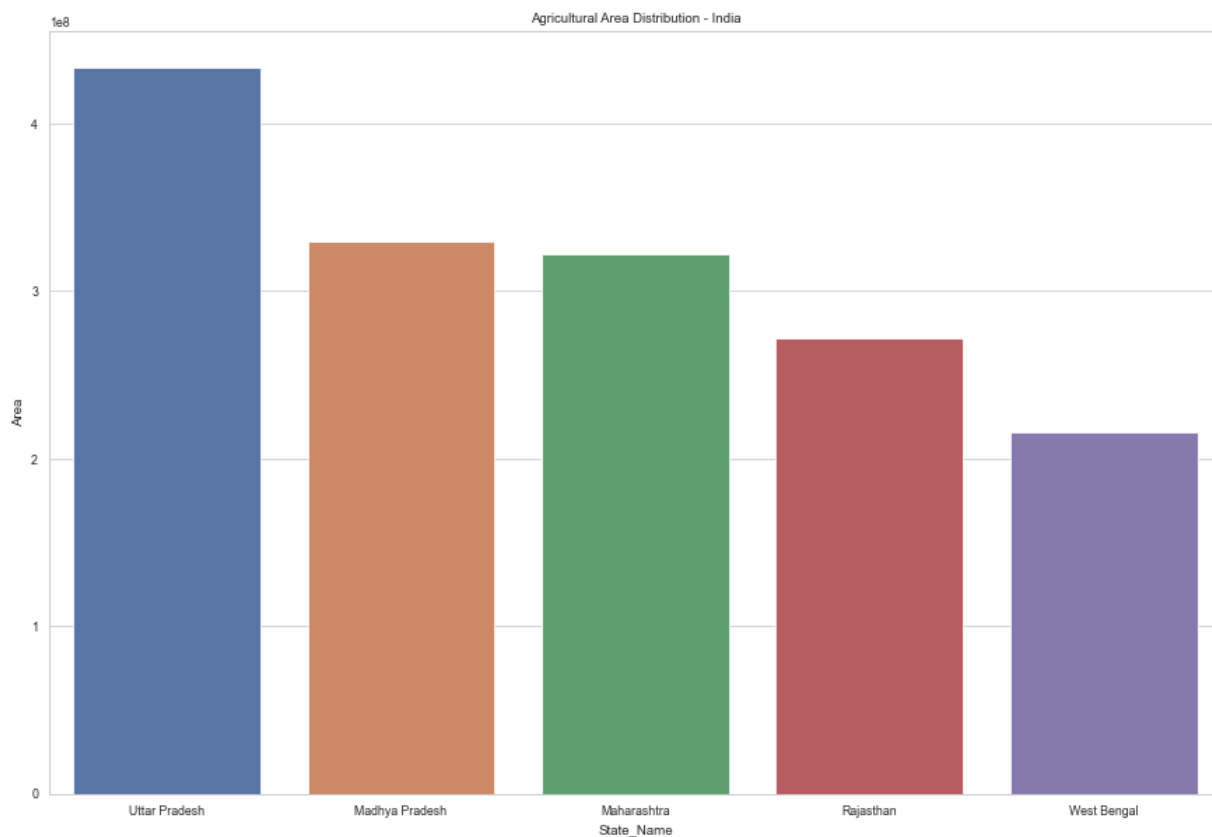
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(df.State_Name.head(5), df.Area.head(5), errwidth=0)
plt.title('Agricultural Area Distribution - India')
df.head(5)
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[32]:

	State_Name	Area
30	Uttar Pradesh	4.336316e+08
16	Madhya Pradesh	3.298131e+08
17	Maharashtra	3.222062e+08
25	Rajasthan	2.720249e+08
32	West Bengal	2.154052e+08



UP leads the nation in agriculture.

## Production of the top 5 agricultural states in last

# 5 years



```
In [33]: df = crop_df.groupby(by='State_Name')['Area'].sum().reset_index().sort_values(by='Area')
df = df.head(5)

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

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

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



And we see how the top 5 states w.r.t Production filtered around 2005. The biggest hit was felt by Maharashtra because of the 2005 Maharashtra floods. Ref -

[https://en.wikipedia.org/wiki/Maharashtra\\_floods\\_of\\_2005](https://en.wikipedia.org/wiki/Maharashtra_floods_of_2005)  
[https://en.wikipedia.org/wiki/Maharashtra\\_floods\\_of\\_2005](https://en.wikipedia.org/wiki/Maharashtra_floods_of_2005).

**Now, we will focus on generating data related to a specific product for collection and to meet market demand.**

## 1. RICE

```
In [36]: data = crop_df.dropna()
rice_df = data[data["Crop"]=="Rice"]
print(rice_df.shape)
rice_df[:3]
```

(15082, 9)

Out[36]:

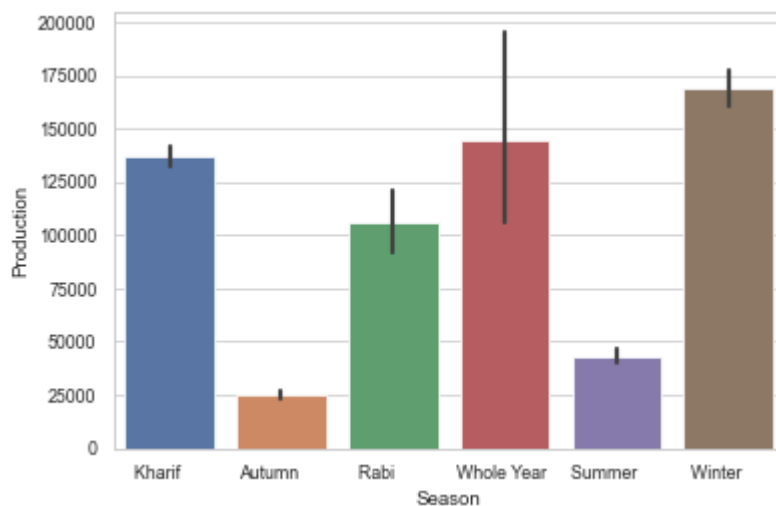
	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Zones	Updated_Crop
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.00	Union Terr	
12	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Rice	83.0	300.00	Union Terr	
18	Andaman and Nicobar Islands	NICOBARS	2002	Kharif	Rice	189.2	510.84	Union Terr	

```
In [37]: sns.barplot("Season", "Production", data=rice_df)
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

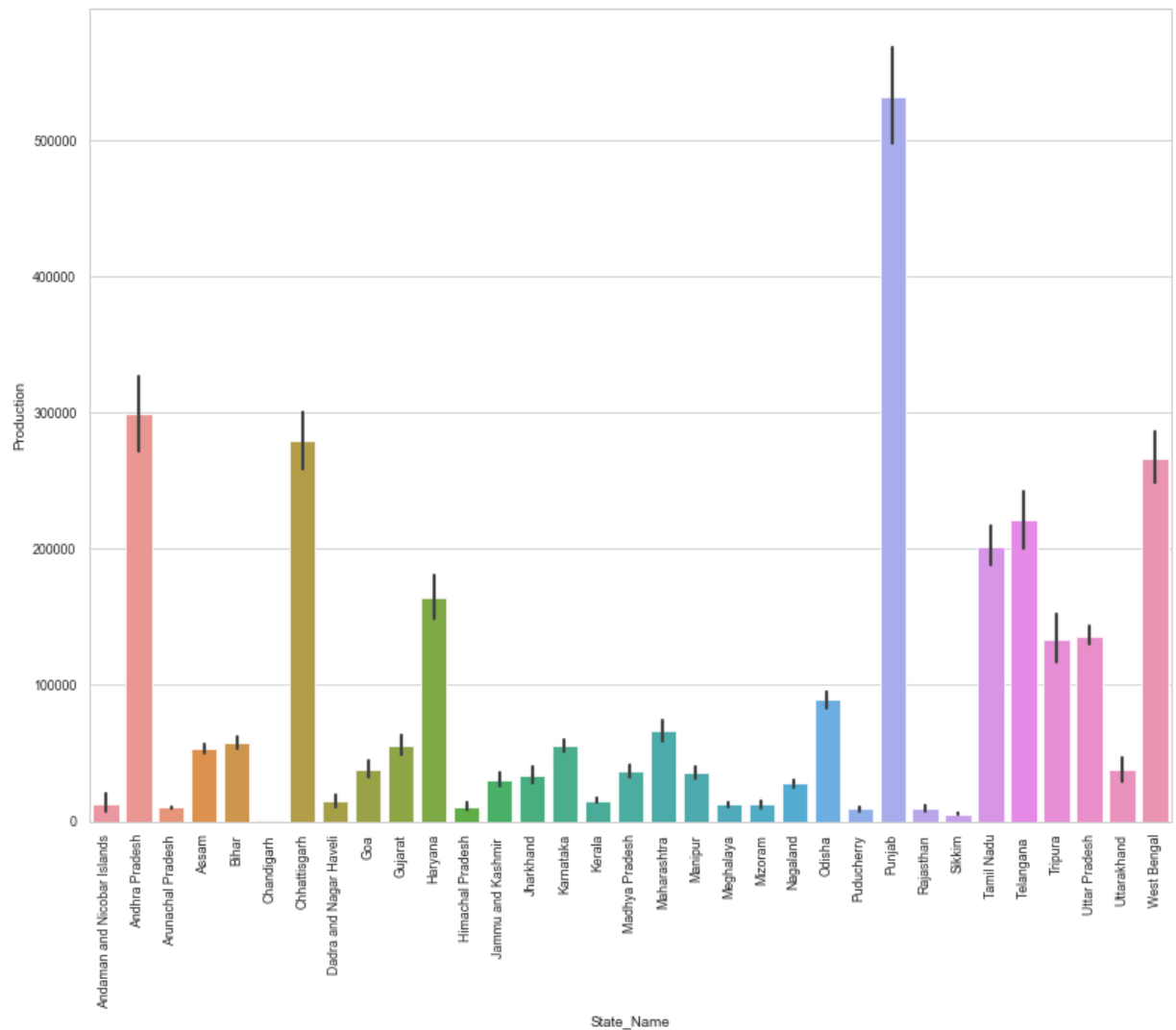
Out[37]: <AxesSubplot:xlabel='Season', ylabel='Production'>



```
In [38]: plt.figure(figsize=(13,10))
sns.barplot("State_Name", "Production", data=rice_df)
plt.xticks(rotation=90)
plt.show()
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



```
In [39]: top_rice_pro_dis = rice_df.groupby("District_Name")["Production"].sum().reset_index(
        by='Production', ascending=False)
top_rice_pro_dis[:5]
sum_max = top_rice_pro_dis["Production"].sum()
top_rice_pro_dis["precent_of_pro"] = top_rice_pro_dis["Production"].map(lambda x:
top_rice_pro_dis[:5]
```

Out[39]:

	District_Name	Production	precent_of_pro
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

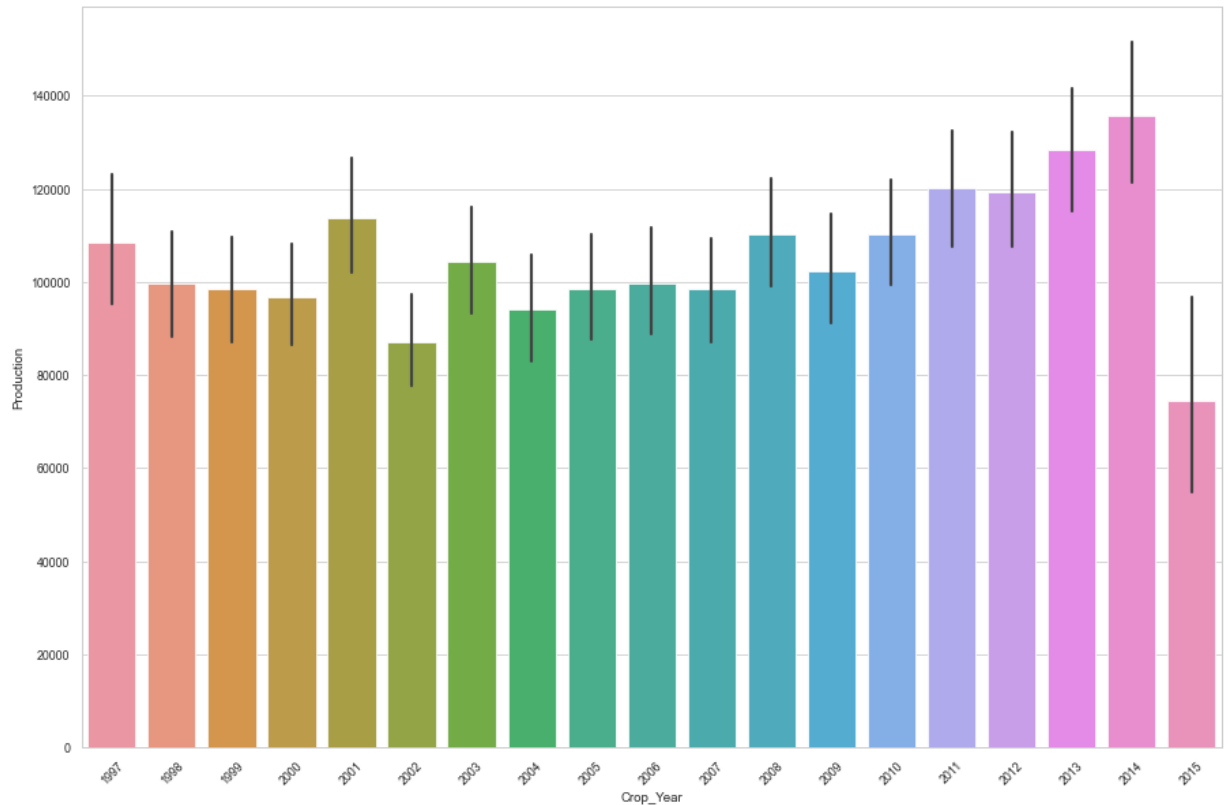
Top 5 rice producing districts.

Year wise rice production

```
In [41]: plt.figure(figsize=(15,10))
sns.barplot("Crop_Year","Production",data=rice_df)
plt.xticks(rotation=45)
#plt.legend(rice_df['State_Name'].unique())
plt.show()
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



## 2.Sugarcane

```
In [42]: sug_df = data[data["Crop"]=="Sugarcane"]
print(sug_df.shape)
sug_df[:3]
```

(7827, 9)

Out[42]:

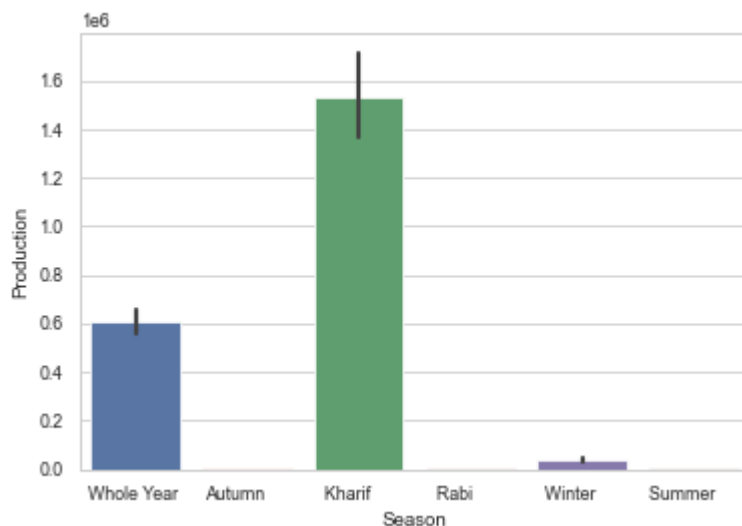
	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Zones	Updated_
7	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1.0	2.0	Union Terr	
16	Andaman and Nicobar Islands	NICOBARS	2001	Whole Year	Sugarcane	1.0	1.0	Union Terr	
26	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Sugarcane	5.0	40.0	Union Terr	

Seasonwise sugarcane product

```
In [43]: sns.barplot("Season", "Production", data=sug_df)
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.  
warnings.warn(

Out[43]: <AxesSubplot:xlabel='Season', ylabel='Production'>

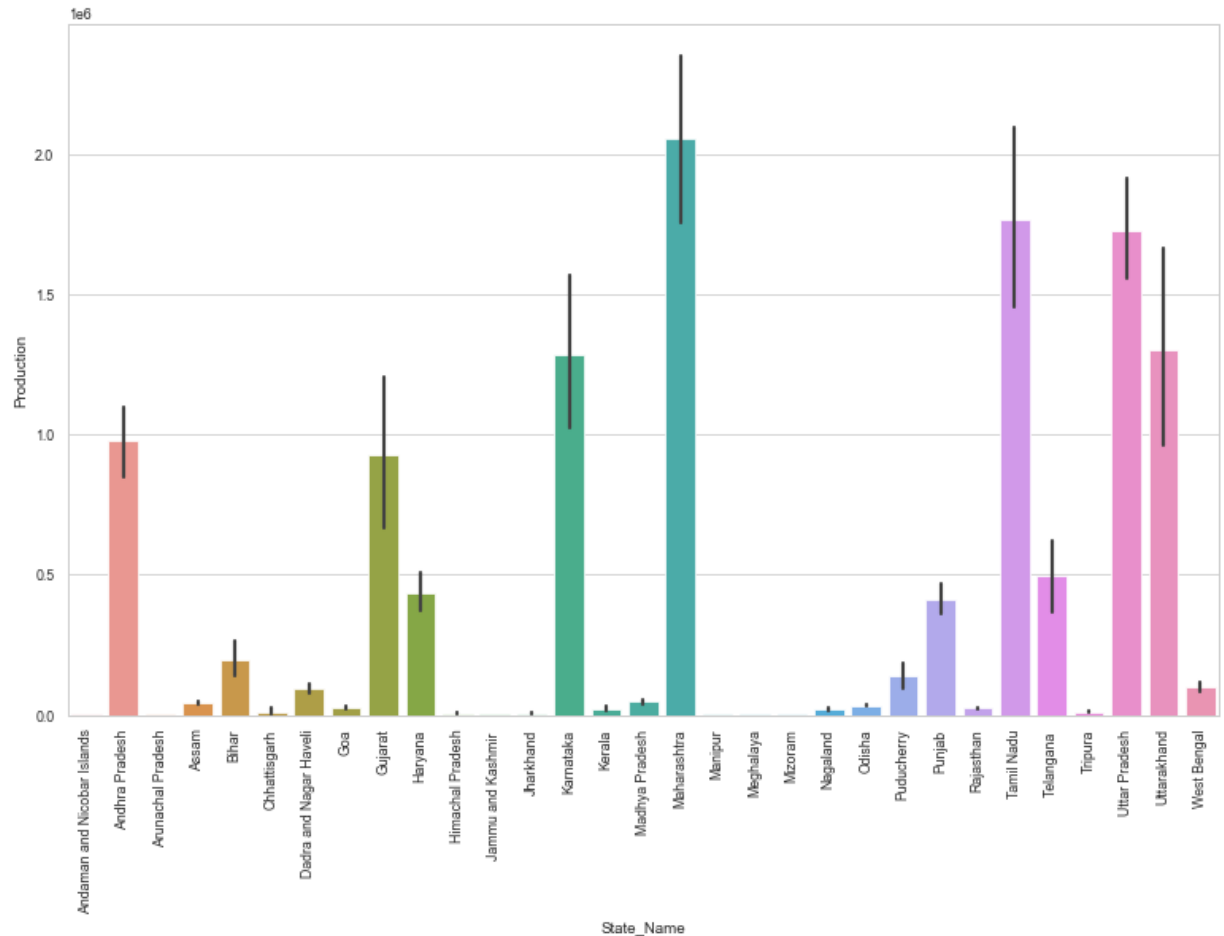


State wise sugarcane production

```
In [44]: plt.figure(figsize=(13,8))
sns.barplot("State_Name", "Production", data=sug_df)
plt.xticks(rotation=90)
plt.show()
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arg s: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



Top 5 Districts in sugarcane production



```
In [45]: top_sug_pro_dis = sug_df.groupby("District_Name")["Production"].sum().reset_index  
        by='Production',ascending=False)  
top_sug_pro_dis[:5]  
sum_max = top_sug_pro_dis["Production"].sum()  
top_sug_pro_dis["precent_of_pro"] = top_sug_pro_dis["Production"].map(lambda x:(>  
top_sug_pro_dis[:5])
```

Out[45]:

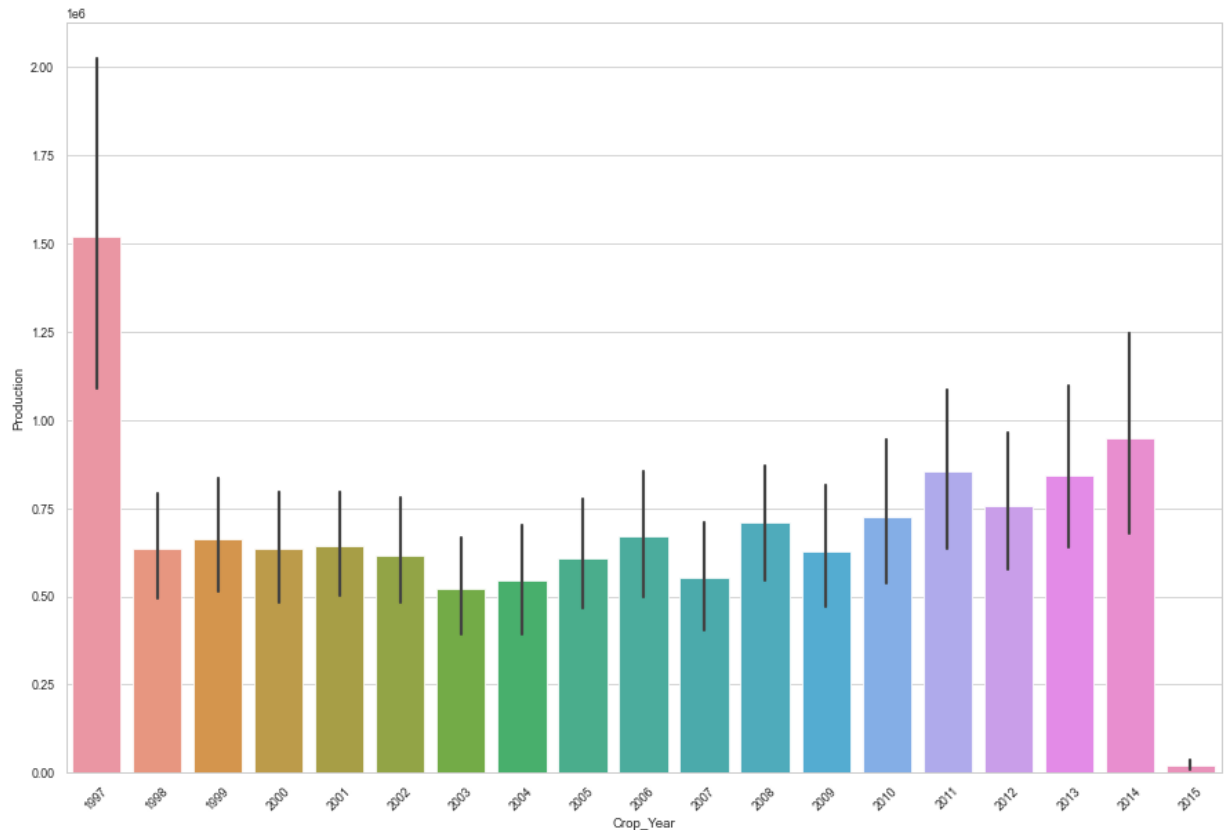
	District_Name	Production	precent_of_pro
360	MUZAFFARNAGAR	258698795.0	4.673296
81	BIJNOR	230907185.0	4.171251
283	KHERI	224383685.0	4.053407
63	BELGAUM	198207649.0	3.580546
294	KOLHAPUR	157210426.0	2.839947

Yearwise sugarcane production

```
In [47]: plt.figure(figsize=(15,10))
sns.barplot("Crop_Year", "Production", data=sug_df)
plt.xticks(rotation=45)
#plt.legend(rice_df['State_Name'].unique())
plt.show()
```

c:\users\jatin\appdata\local\programs\python\python39\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variables as keyword arguments: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(



Insights: Sugarcane production is directly proportional to area And the production is high in some state only.

In [ ]: