CHAPTER-I

DOMAIN UNDERSTANDING

AGRICULTURE

Agriculture is the backbone of the Indian economy, providing employment to millions of people and contributing significantly to the country's GDP. However, the sector is faced with several challenges, including climate change, low productivity, and food security issues. To address these challenges, there is a need for data-driven solutions that can inform policy and decision-making.

Creating a dataset that captures agricultural production statistics in India can help in this regard. The dataset can provide valuable insights into crop yields, area under cultivation, and other metrics that can inform agricultural policies and practices. It can also be used to identify trends and patterns in agricultural production, helping farmers and policymakers make informed decisions about crop selection, irrigation, and other important factors that affect agricultural productivity.

Additionally, the dataset can be used for machine learning and predictive modeling to generate insights and make accurate predictions about crop production in different parts of the country. Overall, the dataset has the potential to contribute significantly to the development of the agriculture sector in India and help address some of the challenges faced by the sector.

AGRICULTURE IN INDIA

Agriculture is a major contributor to the Indian economy, with over 58% of the rural households depending on it for their livelihood. India is the world's second-largest producer of food and agricultural commodities, and agriculture accounts for around 18% of the country's GDP.

The agricultural sector in India is diverse and ranges from subsistence farming to commercial agriculture. The major crops grown in India include rice, wheat, maize, cotton, sugarcane, oilseeds, pulses, and fruits and vegetables. Livestock, dairy, and fisheries also form an important part of the agricultural sector.

However, the agriculture sector in India is facing several challenges such as climate change, water scarcity, land degradation, low productivity, and fragmented land holdings. To address these challenges, the government of India has implemented several initiatives and schemes such as the Pradhan Mantri Fasal Bima Yojana, Pradhan Mantri Krishi Sinchai Yojana, and Soil Health Card Scheme, among others.

In recent years, there has been a shift towards organic farming, sustainable agriculture, and use of modern technology and precision agriculture techniques in Indian agriculture. The government of India is also promoting initiatives such as e-NAM (National Agricultural Market) to create a unified market for agricultural commodities and increase the income of farmers.

India is a diverse country with different agro-climatic zones and soil types, which support the cultivation of a wide variety of crops. The major crops grown in India include:

Rice - India is the second-largest producer of rice in the world after China, and rice is a staple food crop in most parts of the country.

Wheat - Wheat is the second-most important cereal crop in India after rice and is primarily grown in the northern and north western parts of the country.

Maize - Maize is the third-most important cereal crop in India after rice and wheat and is used for both food and industrial purposes.

Cotton - India is the world's largest producer of cotton, and cotton cultivation is primarily concentrated in the central and southern states of the country.

Sugarcane - Sugarcane is an important cash crop grown in India and is used for sugar production as well as for the production of ethanol and biofuels.

Oilseeds - India is a major producer of oilseeds such as groundnut, mustard, soybean, sesame, ` and sunflower, which are used for oil extraction.

Pulses - Pulses are an important source of protein in the Indian diet, and India is the world's largest producer of pulses such as chickpeas, lentils, and pigeon peas.

Fruits and vegetables - India is a major producer of fruits and vegetables, including mangoes, bananas, grapes, tomatoes, onions, and potatoes.

CHAPTER - II

DATA UNDERSTANDING

ABOUT DATASET

The dataset contains comprehensive data on crop production statistics for India, categorized by state and district. The dataset covers four major crop seasons, namely kharif, rabbi, summer, and autumn, from the year 1997 to 2023. The data provides information on the annual production and yield of crops grown in different parts of the country.

The dataset will be useful for researchers, policymakers, and farmers who are interested in understanding crop production patterns in different regions of India. By analysing the data, researchers can identify the factors that influence crop yields and production and can make informed decisions on how to improve agricultural productivity in the country. Policymakers can use the data to design and implement agricultural policies that promote sustainable farming practices and improve food security.

Farmers can also benefit from the dataset by gaining insights into the best crops to grow in their region and making informed decisions on crop management practices. Additionally, the dataset can be used to train machine learning models to predict crop yields and production in different parts of the country, which can be valuable for agricultural businesses and organizations. Overall, the dataset provides a comprehensive overview of crop production statistics in India, which is essential for understanding the country's agricultural landscape and developing effective strategies for sustainable agriculture.

EXPLORATION OF DATA

Importing libraries and load the dataset:

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

```
In [2]: crop = pd.read_csv('C:\\Users\\Student\\Desktop\\APYC.csv')
```

```
In [3]: crop
```

Out[3]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
0	Andaman and Nicobar Island	NICOBARS	Arecanut	2007	Kharif	2439.6	3415.0	1.40
1	Andaman and Nicobar Island	NICOBARS	Arecanut	2007	Rabi	1626.4	2277.0	1.40
2	Andaman and Nicobar Island	NICOBARS	Arecanut	2008	Autumn	4147.0	3060.0	0.74
3	Andaman and Nicobar Island	NICOBARS	Arecanut	2008	Summer	4147.0	2660.0	0.64
4	Andaman and Nicobar Island	NICOBARS	Arecanut	2009	Autumn	4153.0	3120.0	0.75
...	...	...	...	...	...	...	...	...
345331	West Bengal	PURULIA	Wheat	2015	Rabi	855.0	1241.0	1.45
345332	West Bengal	PURULIA	Wheat	2016	Rabi	1366.0	2415.0	1.77
345333	West Bengal	PURULIA	Wheat	2017	Rabi	1052.0	2145.0	2.04
345334	West Bengal	PURULIA	Wheat	2018	Rabi	833.0	2114.0	2.54
345335	West Bengal	PURULIA	Wheat	2019	Rabi	516.0	931.0	1.80

Import libraries and read the dataset into the jupyter notebook and view the dataset.

head ( )

```
In [6]: crop.head()
```

Out[6]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
0	Andaman and Nicobar Island	NICOBARS	Arecanut	2007	Kharif	2439.6	3415.0	1.40
1	Andaman and Nicobar Island	NICOBARS	Arecanut	2007	Rabi	1626.4	2277.0	1.40
2	Andaman and Nicobar Island	NICOBARS	Arecanut	2008	Autumn	4147.0	3060.0	0.74
3	Andaman and Nicobar Island	NICOBARS	Arecanut	2008	Summer	4147.0	2660.0	0.64
4	Andaman and Nicobar Island	NICOBARS	Arecanut	2009	Autumn	4153.0	3120.0	0.75

Top 5 rows of the given crops dataset.

Tail ( )

```
In [7]: crop.tail()
```

Out[7]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
345331	West Bengal	PURULIA	Wheat	2015	Rabi	855.0	1241.0	1.45
345332	West Bengal	PURULIA	Wheat	2016	Rabi	1366.0	2415.0	1.77
345333	West Bengal	PURULIA	Wheat	2017	Rabi	1052.0	2145.0	2.04
345334	West Bengal	PURULIA	Wheat	2018	Rabi	833.0	2114.0	2.54
345335	West Bengal	PURULIA	Wheat	2019	Rabi	516.0	931.0	1.80

Last 5 rows of the given dataset.

info ( )

```
In [4]: crop.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 345336 entries, 0 to 345335
Data columns (total 8 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   State       345336 non-null  object
 1   District    345336 non-null  object
 2   Crop        345327 non-null  object
 3   Crop_Year   345336 non-null  int64
 4   Season      345336 non-null  object
 5   Area        345336 non-null  float64
 6   Production  340388 non-null  float64
 7   Yield       345336 non-null  float64
dtypes: float64(3), int64(1), object(4)
memory usage: 21.1+ MB
```

Variables and information about the given dataset.

shape

```
In [5]: crop.shape
Out[5]: (345336, 8)
```

There are 345336 rows and 8 columns in the given dataset.

describe ( )

```
In [142]: crop.describe()
```

Out[142]:

	Crop_Year	Area	Production	Yield
count	338918.00	338918.00	338918.00	338918.00
mean	2008.91	11876.57	962629.85	80.93
std	6.57	46219.45	21577225.65	925.25
min	1997.00	0.01	1.00	0.00
25%	2003.00	81.00	90.00	0.57
50%	2009.00	573.00	731.00	1.03
75%	2015.00	4300.00	7273.00	2.51
max	2020.00	8580100.00	1597800000.00	43958.33

Detailed description about the given dataset like minimum & .maximum values.

value_counts( )

```
In [22]: crop['Season'].value_counts()
```

```
Out[22]: Kharif         135460
         Rabi            99444
         Whole Year      67039
         Summer          21899
         Winter           8192
         Autumn           6884
         Name: Season, dtype: int64
```

Total value counts of the seoson variable from the given dataset.

columns

```
In [144]: crop.columns
```

```
Out[144]: Index(['State', 'District ', 'Crop', 'Crop_Year', 'Season', 'Area ',
                 'Production', 'Yield'],
                dtype='object')
```

Total column variable names from the given dataset.

dtypes

```
In [145]: crop.dtypes

Out[145]: State          object
          District       object
          Crop           object
          Crop_Year       int64
          Season         object
          Area          float64
          Production    float64
          Yield         float64
          dtype: object
```

To know the variable data types from the given dataset.

CHAPTER-III

EXPLORATORY DATA ANALYSIS

DIFFERENT SEASONS OF CROP PRODUCTION

Exploring Different Seasons

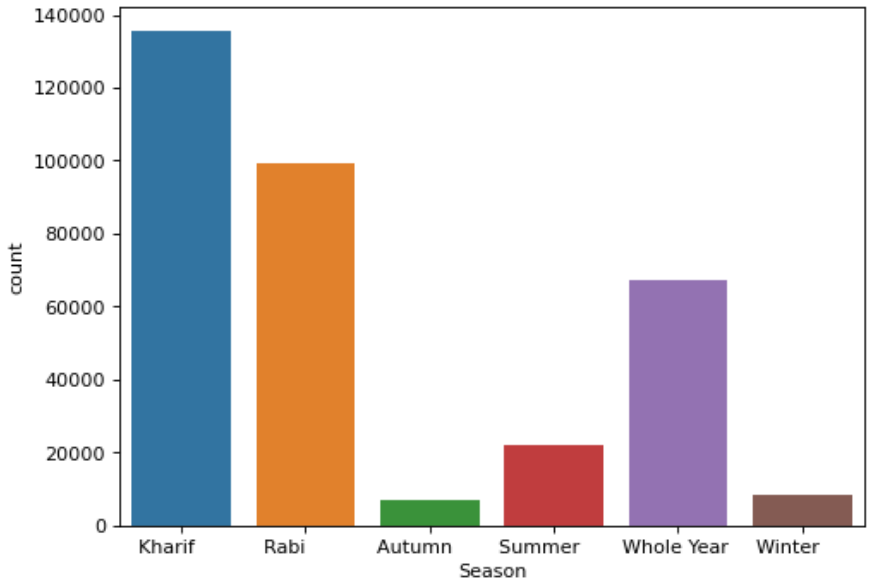
Lets see the different seasons

```
In [22]: crop['Season'].value_counts()
```

```
Out[22]: Kharif         135460
         Rabi            99444
         Whole Year      67039
         Summer          21899
         Winter           8192
         Autumn           6884
         Name: Season, dtype: int64
```

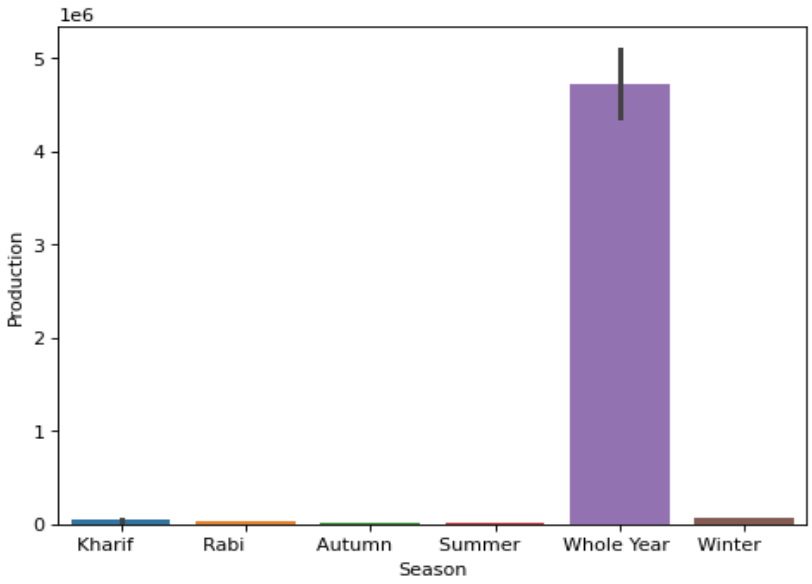
Different seasons and their counts from the given dataset.

```
In [150]: plt.figure(figsize=(7,5),dpi=80)
          sns.countplot(data=crop,x='Season');
```



Kharif season is the highest crops yielding season.

```
In [151]: plt.figure(figsize=(7,5),dpi=80)
          sns.barplot(data=crop,x='Season',y='Production');
```



Whole Year season seems to have yielded more crops compared to other seasons in a year.

```
In [27]: state_prod
```

Out[27]:

	State	Production
17	Kerala	129700649853.00
31	Tamil Nadu	78051759253.00
16	Karnataka	63772797366.00
1	Andhra Pradesh	26076218605.00
36	West Bengal	8941179120.00
34	Uttar Pradesh	4442585302.00
3	Assam	3637714928.00
10	Goa	2193998349.00
0	Andaman and Nicobar Island	2053349886.00
20	Maharashtra	1878564915.00
19	Madhya Pradesh	834490323.00
11	Gujarat	807581678.00
27	Punjab	781551409.00
12	Haryana	589739640.00
28	Rajasthan	589164328.00
4	Bihar	544953534.00
26	Puducherry	493815573.00
25	Odisha	194080325.00
35	Uttarakhand	179697348.00
32	Telangana	147700034.00
6	Chhattisgarh	143096606.00
15	Jharkhand	43793850.00
13	Himachal Pradesh	32297151.00
14	Jammu and Kashmir	30298377.00
21	Manipur	18764635.00
24	Nagaland	18748702.00
33	Tripura	16952054.00
22	Meghalaya	16516629.00
2	Arunachal Pradesh	9522010.00
23	Mizoram	2769208.00
29	Sikkim	2744927.00
9	Delhi	2666022.00
30	THE DADRA AND NAGAR HAVELI	2222055.00
7	Dadra and Nagar Haveli	337093.00
18	Laddak	114584.00
5	CHANDIGARH	89782.00
8	Daman and Diu	59268.00

We see Kerela has more productions among other states In India Followed by the Tamil Nadu, Karnataka and Andhra Pradesh (South Indian States are Having More Production).

Top 5 States with more Production

Let's see the top 5 states with most production over the years

```
In [28]: crop.groupby('State').sum()['Production'].nlargest()

Out[28]: State
         Kerala           129700649853.00
         Tamil Nadu        78051759253.00
         Karnataka         63772797366.00
         Andhra Pradesh    26076218605.00
         West Bengal        8941179120.00
         Name: Production, dtype: float64
```

Kerala is the highest crop producing state.

States with Least Production

Let's see the states with least production over the years

```
In [29]: crop.groupby("State").sum()['Production'].nsmallest()

Out[29]: State
         Daman and Diu                    59268.00
         CHANDIGARH                       89782.00
         Laddak                          114584.00
         Dadra and Nagar Haveli          337093.00
         THE DADRA AND NAGAR HAVELI     2222055.00
         Name: Production, dtype: float64
```

Daman and Diu is the least crop producing state.

Top 5 Crops with more Production

Let's see the top 5 crops with production over the years

```
In [31]: crop.groupby('Crop').sum()['Production'].nlargest()
```

```
Out[31]: Crop
         Coconut     310804772578.00
         Sugarcane     7249507133.00
         Rice          2236428183.00
         Wheat         2007360244.00
         Potato         632315694.00
         Name: Production, dtype: float64
```

Coconut is the large production crop over the years.

Crops with Least Production

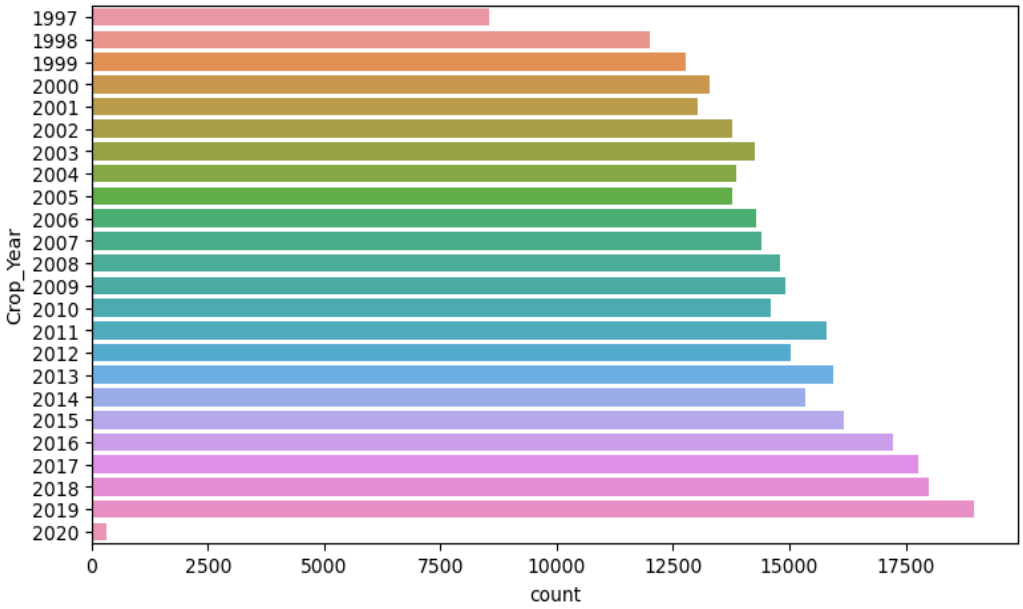
Let's see the crops with least production over the years

```
In [32]: crop.groupby('Crop').sum()['Production'].nsmallest()
```

```
Out[32]: Crop
         Other Summer Pulses      8394.00
         Cardamom               255497.00
         Sannhamp               433530.00
         Cowpea(Lobia)          745568.00
         Other Cereals         1682056.00
         Name: Production, dtype: float64
```

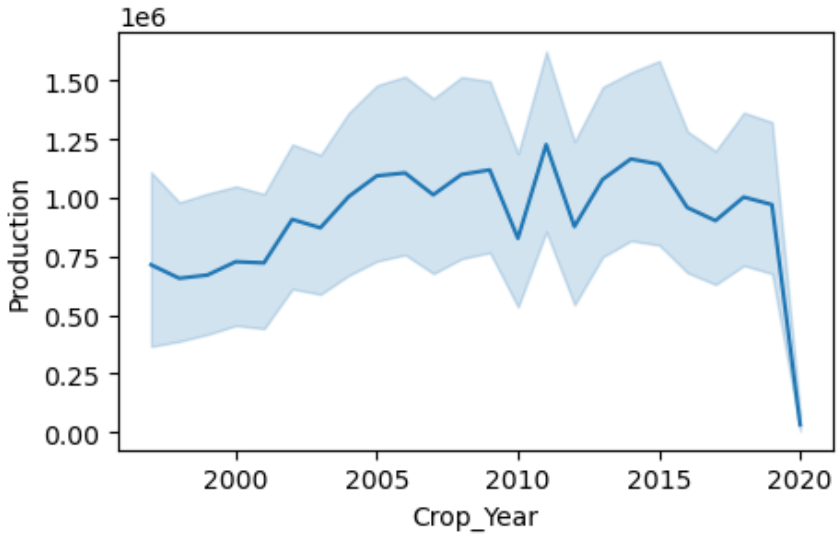
Some other summer pulses are the least produced crops over the years.

```
In [124]: plt.figure(figsize=(9,5),dpi=100)
          sns.countplot(data=crop,y='Crop_Year');
```



2018 & 2019 has maximum Crop Production.

```
In [136]: plt.figure(figsize=(5,3),dpi=100)
          sns.lineplot(data=crop,x='Crop_Year',y='Production');
```



Though **2018-2019** was the year most crops were cultivated, the period between **2010-2015** and then **2012-2013** happens to be the year which saw highest yield for the crops.

STATEWISE PRODUCTION STATISTICS

TOP 5 STATES OF LARGE PRODUCTION

1) Kerala

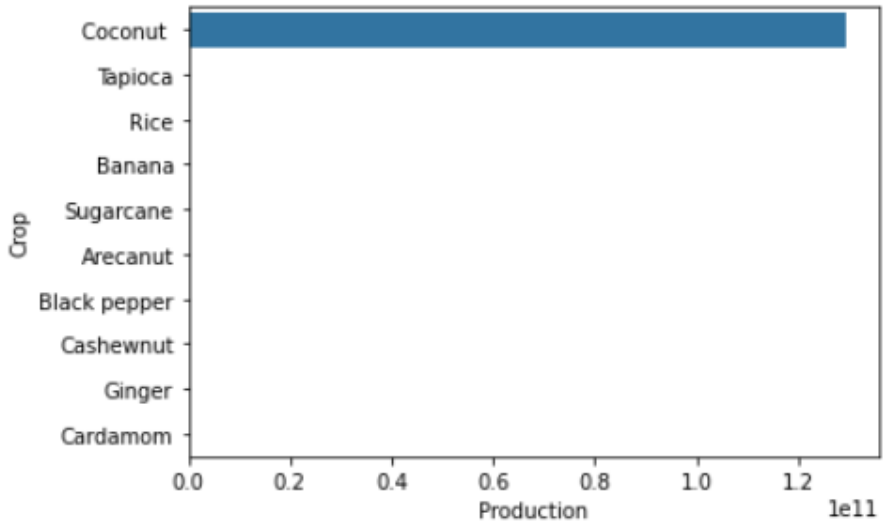
```
In [39]: kerala_data = crop[crop['State'] == 'Kerala']
         kerala_data.head()
```

Out[39]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
144517	Kerala	ALAPPUZHA	Arecanut	1997	Whole Year	2253.00	1518.00	0.67
144518	Kerala	ALAPPUZHA	Arecanut	1998	Whole Year	2067.00	766.00	0.37
144519	Kerala	ALAPPUZHA	Arecanut	1999	Whole Year	2308.00	1043.00	0.45
144520	Kerala	ALAPPUZHA	Arecanut	2000	Whole Year	2205.00	917.00	0.42
144521	Kerala	ALAPPUZHA	Arecanut	2001	Whole Year	2389.00	792.00	0.33

```
In [41]: sns.barplot(data=top_prod_kerala,y='Crop',x='Production')
```

Out[41]: <AxesSubplot:xlabel='Production', ylabel='Crop'>



Coconut ranks First in Production (Kerala) followed by Tapioca, Rice, Banana, Sugarcane, Areca nut.

2) Tamil Nadu

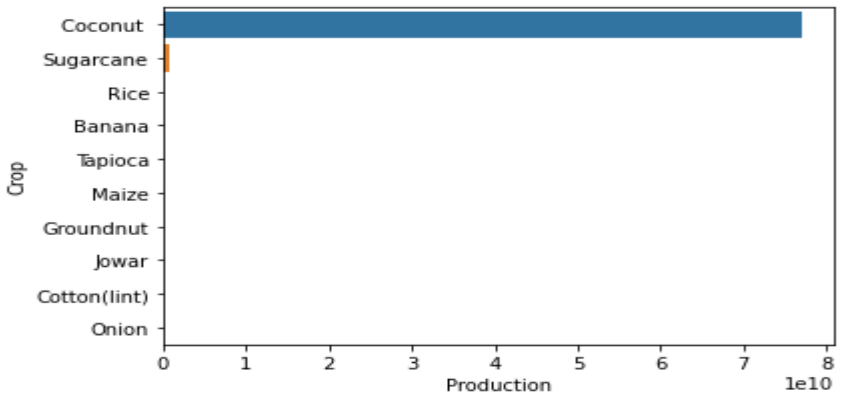
```
In [42]: Tamilnadu = crop[crop['State'] == 'Tamil Nadu']
         Tamilnadu.head()
```

Out[42]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
255243	Tamil Nadu	ARIYALUR	Arecanut	2018	Whole Year	1.00	2.00	2.00
255244	Tamil Nadu	ARIYALUR	Arecanut	2019	Whole Year	1.00	2.00	2.00
255247	Tamil Nadu	COIMBATORE	Arecanut	2004	Whole Year	1574.00	1844.00	1.17
255248	Tamil Nadu	COIMBATORE	Arecanut	2005	Whole Year	1556.00	4469.00	2.87
255249	Tamil Nadu	COIMBATORE	Arecanut	2006	Whole Year	1602.00	3828.00	2.39

```
In [45]: sns.barplot(data=top_prod_tamil,y='Crop',x='Production')
```

```
Out[45]: <AxesSubplot:xlabel='Production', ylabel='Crop'>
```



In Tamil Nadu Coconut, Sugarcane, rice, Banana are produced in Maximum.

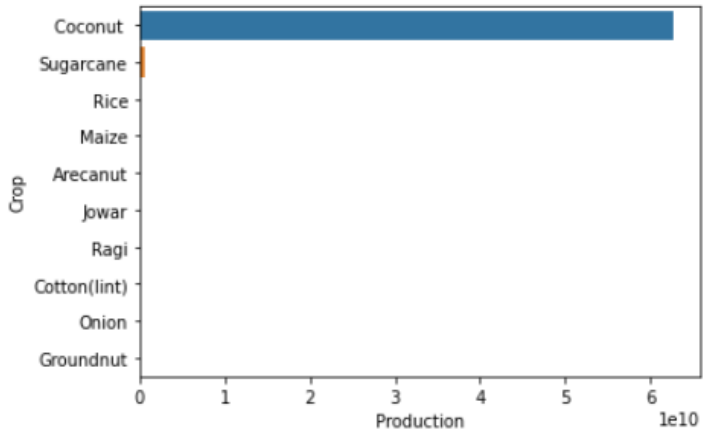
3) Karnataka

```
In [46]: Karnataka = crop[crop['State'] == 'Karnataka']
         Karnataka.head()
```

Out[46]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
117024	Karnataka	BAGALKOTE	Arecanut	1998	Whole Year	1.00	7.00	7.00
117025	Karnataka	BAGALKOTE	Arecanut	2000	Whole Year	4.00	6.00	1.50
117026	Karnataka	BAGALKOTE	Arecanut	2001	Whole Year	4.00	6.00	1.50
117027	Karnataka	BAGALKOTE	Arecanut	2003	Whole Year	6.00	8.00	1.33
117028	Karnataka	BAGALKOTE	Arecanut	2004	Whole Year	2.00	3.00	1.50

```
In [49]: sns.barplot(data=top_prod_Karnataka,y='Crop',x='Production')
Out[49]: <AxesSubplot:xlabel='Production', ylabel='Crop'>
```



In Karnataka Coconut is most produced followed by sugarcane, Rice and Maize.

Now, Lets analyse the production data based on the products

```
In [50]: crop.groupby('Season').sum()['Production'].nlargest()
Out[50]: Season
         Whole Year    316563640348.00
         Kharif          5622334808.00
         Rabi            3150454633.00
         Winter           587750509.00
         Summer           243762911.00
         Name: Production, dtype: float64
```

Whole year has the highest production.

```
In [51]: crop['Crop'].value_counts()

Out[51]: Rice                    21566
         Maize                   20267
         Moong(Green Gram)       14713
         Urad                    14296
         Sesamum                 12541
         Groundnut               12470
         Wheat                   11204
         Rapeseed &Mustard       10873
         Sugarcane               10812
         Arhar/Tur               10737
         Potato                  10725
         Onion                   10616
         Gram                    10286
         Jowar                    9653
         Dry chillies             8868
         Bajra                    8063
         Peas & beans (Pulses)    7147
         Sunflower                7077
         Small millets            6815
         Cotton(lint)             6284
         Masoor                   6267
         Barley                   5769
         Linseed                  5758
         Ragi                     5679
         Soyabean                 4947
         Other  Rabi pulses       4694
         Castor seed              4476
         Ginger                   4406
         Banana                   4013
         Tobacco                  3851
         Coconut                  2890
         Niger seed               2809
         Sannhamp                 2694
         Mesta                    2373
         Tapioca                  2263
         Arecanut                 2150
         Guar seed                2045
         Jute                     1837
         Khesari                  1753
         Safflower                1700
         Cowpea(Lobia)            1696
         Cashewnut                1515
         Black pepper             1363
         Other Cereals            1332
         Moth                     1312
         other oilseeds           1161
         Cardamom                  476
         Oilseeds total            435
         Other Summer Pulses        66
         Name: Crop, dtype: int64
```

Over the year Rice, Maize, Moong, Urad are cultivated Max.

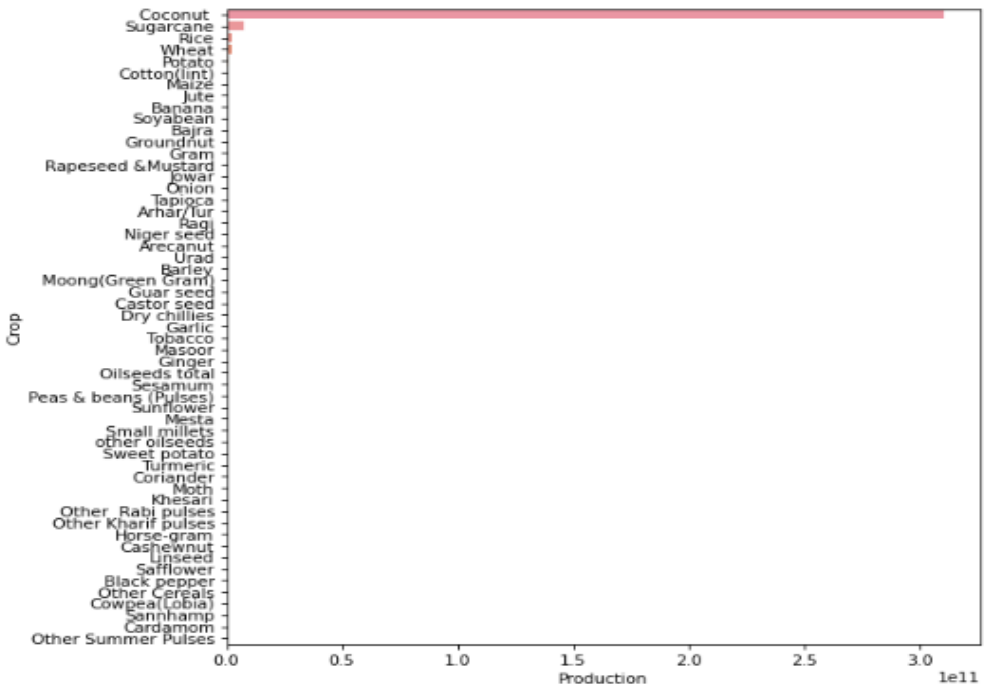
```
In [52]: india_top_prod = crop.groupby('Crop').sum()['Production'].nlargest(55).reset_index()
         india_top_prod
```

	Crop	Production
0	Coconut	310804772578.00
1	Sugarcane	7249507133.00
2	Rice	2236428183.00
3	Wheat	2007360244.00
4	Potato	632315694.00
5	Cotton(lint)	483907997.00
6	Maize	444000679.00
7	Jute	230423821.00
8	Banana	226763294.00
9	Soyabean	211796467.00
10	Bajra	201100380.00
11	Groundnut	163832092.00
12	Gram	160256419.00
13	Rapeseed &Mustard	149836139.00
14	Jowar	149255891.00
15	Onion	133343945.00
16	Tapioca	130918132.00
17	Arhar/Tur	61260778.00
18	Ragi	44253722.00
19	Niger seed	40643631.00
20	Arecanut	39299349.00
21	Urad	37888016.00
22	Barley	35069331.00
23	Moong(Green Gram)	32398991.00
24	Guar seed	31321927.00
25	Castor seed	27949346.00
26	Dry chillies	26534403.00
28	Tobacco	21102473.00
29	Masoor	20412728.00
30	Ginger	17717345.00
31	Oilseeds total	17535566.00
32	Sesamum	15742227.00
33	Peas & beans (Pulses)	14800028.00
34	Sunflower	14645676.00
35	Mesta	14053610.00
36	Small millets	13561916.00
37	other oilseeds	11818281.00
38	Sweet potato	10553575.00
39	Turmeric	9706411.00
40	Coriander	7355907.00
41	Moth	7251608.00
42	Khesari	7115452.00
43	Other Rabi pulses	6664952.00
44	Other Kharif pulses	6133466.00
45	Horse-gram	5276790.00
46	Cashewnut	3740786.00
47	Linseed	3298069.00
48	Safflower	3241790.00
49	Black pepper	2097317.00
50	Other Cereals	1682056.00
51	Cowpea(Lobia)	745568.00
52	Sannhamp	433530.00
53	Cardamom	255497.00
54	Other Summer Pulses	8394.00

Coconut, Sugarcane, Rice, Wheat and Potato are (the top 5 crops produced more in India over the years)

```
In [153]: plt.figure(figsize=(8,8),dpi=80)
          sns.barplot(data=india_top_prod,x='Production',y='Crop')
```

```
Out[153]: <AxesSubplot:xlabel='Production', ylabel='Crop'>
```



Coconut is the crop with the largest production.

Let do Crop wise analyses based on the Production

1) Coconut

```
In [55]: coconut_prod = crop[crop['Crop'] == 'Coconut ']
```

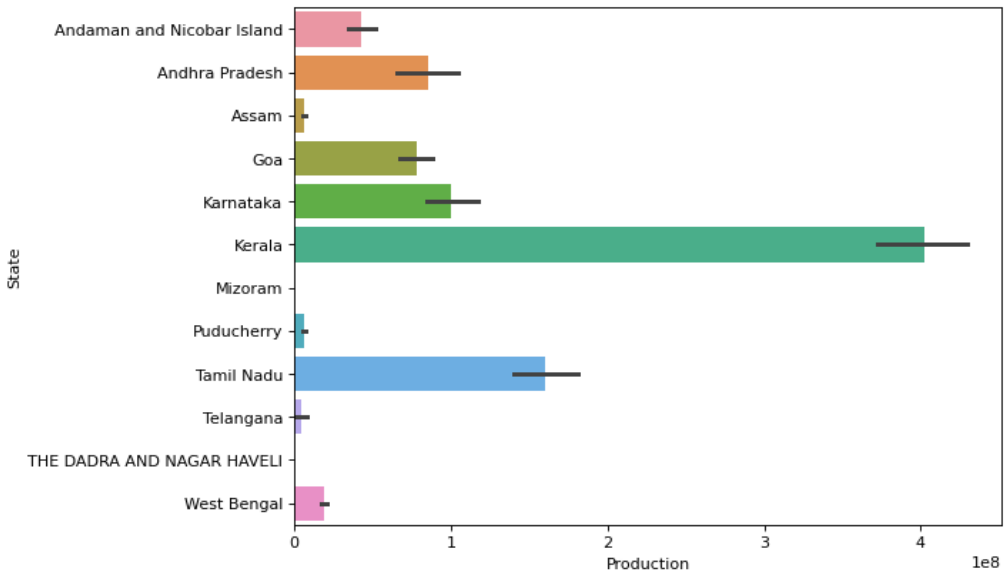
```
In [56]: coconut_prod
```

```
Out[56]:
```

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
207	Andaman and Nicobar Island	NICOBARS	Coconut	2007	Whole Year	21636.00	80640000.00	3727.12
208	Andaman and Nicobar Island	NICOBARS	Coconut	2008	Whole Year	43380.00	81900000.00	1887.97
209	Andaman and Nicobar Island	NICOBARS	Coconut	2009	Whole Year	43520.00	84970000.00	1952.44
210	Andaman and Nicobar Island	NICOBARS	Coconut	2000	Whole Year	18168.00	65100000.00	3583.22
211	Andaman and Nicobar Island	NICOBARS	Coconut	2001	Whole Year	18190.00	64430000.00	3542.06
...	...	...	...	...	...	...	...	...
333865	West Bengal	PURULIA	Coconut	2015	Whole Year	65.00	723000.00	11123.08
333866	West Bengal	PURULIA	Coconut	2016	Whole Year	65.00	754900.00	11613.85
333867	West Bengal	PURULIA	Coconut	2017	Whole Year	68.00	785500.00	11551.47
333868	West Bengal	PURULIA	Coconut	2018	Whole Year	75.00	857500.00	11433.33
333869	West Bengal	PURULIA	Coconut	2019	Whole Year	78.00	858100.00	11001.28

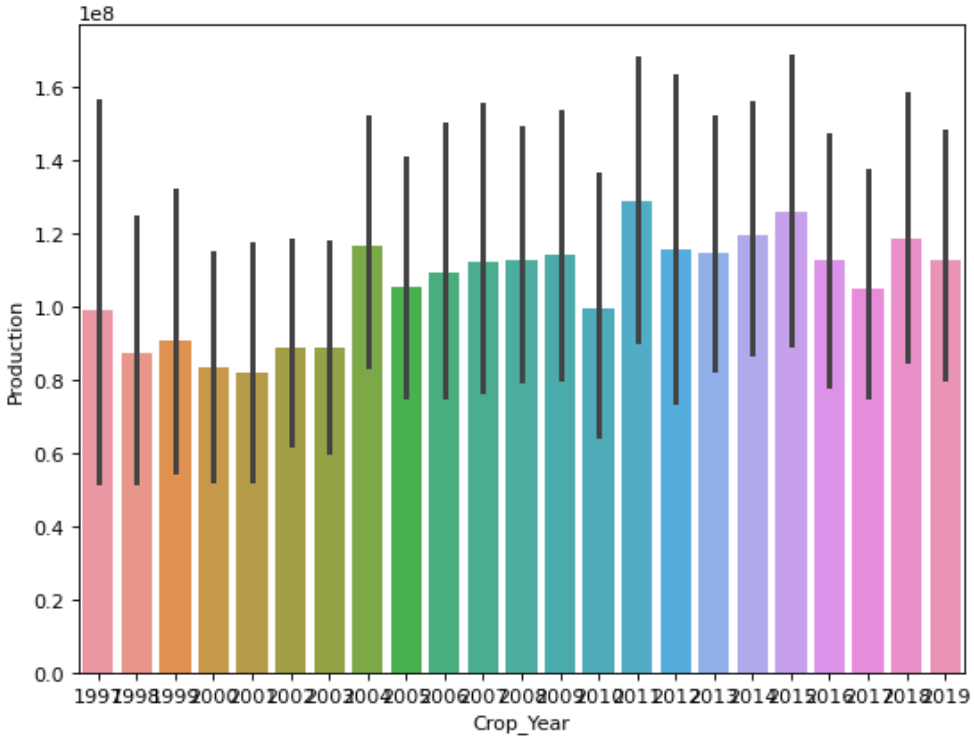
2890 rows × 8 columns

```
In [154]: plt.figure(figsize=(8,6),dpi=80)
          sns.barplot(data=coconut_prod,x='Production',y='State');
```



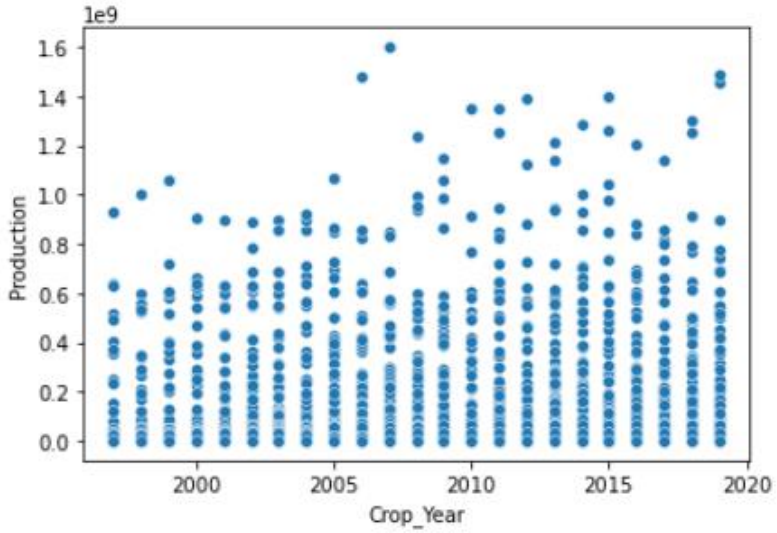
Kerala, Tamil Nadu, followed by Andhra pradesh are high in the production.

```
In [157]: plt.figure(figsize=(8,6),dpi=80)
          sns.barplot(data=coconut_prod,x='Crop_Year',y='Production');
```



The Year 2015 has More Coconut production.

```
In [62]: sns.scatterplot(data=coconut_prod,x='Crop_Year',y='Production');
```



The Production of the coconut have been stable from the Year of 2013-2015.

2008 has Maximum Production

2018< 2019< 2020 have drastic increase in the production.

2) Sugarcane

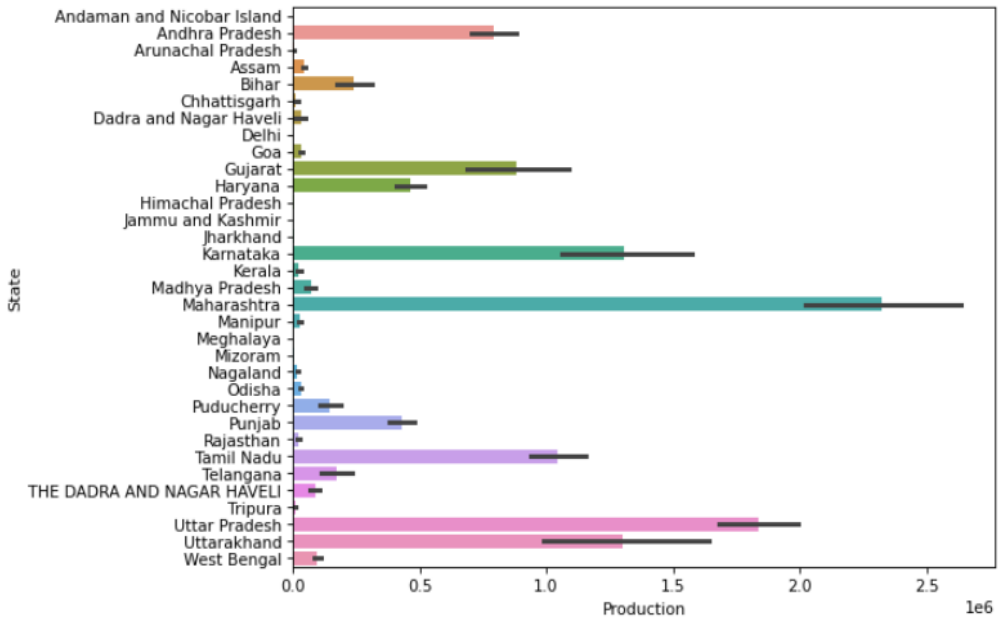
```
In [63]: sugarcane_prod = crop[crop['Crop'] == 'Sugarcane']
```

```
In [64]: sugarcane_prod.head()
```

Out[64]:

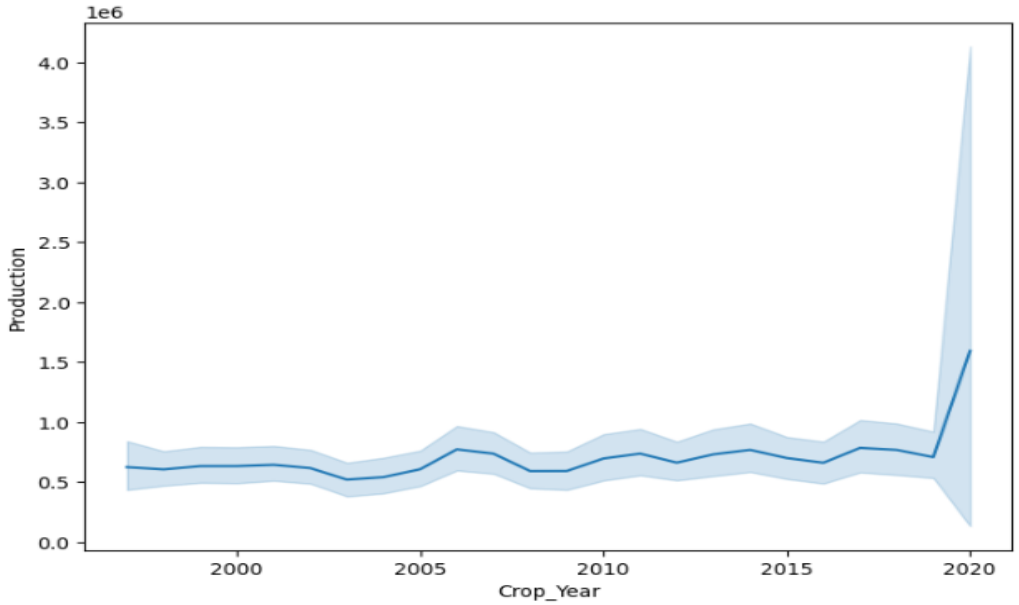
	State	District	Crop	Crop_Year	Season	Area	Production	Yield
528	Andaman and Nicobar Island	NICOBARS	Sugarcane	2007	Kharif	106.50	2129.00	19.99
529	Andaman and Nicobar Island	NICOBARS	Sugarcane	2007	Rabi	70.93	1419.00	20.01
530	Andaman and Nicobar Island	NICOBARS	Sugarcane	2008	Autumn	117.00	2260.00	19.32
531	Andaman and Nicobar Island	NICOBARS	Sugarcane	2008	Summer	39.00	760.00	19.49
532	Andaman and Nicobar Island	NICOBARS	Sugarcane	2009	Autumn	113.00	1610.00	14.25

```
In [160]: plt.figure(figsize=(8,6),dpi=80)
          sns.barplot(data=sugarcane_prod,x='Production',y='State');
```



Maharashtra, Uttar Pradesh, Karnataka followed by Uttarakhand has the highest sugarcane production.

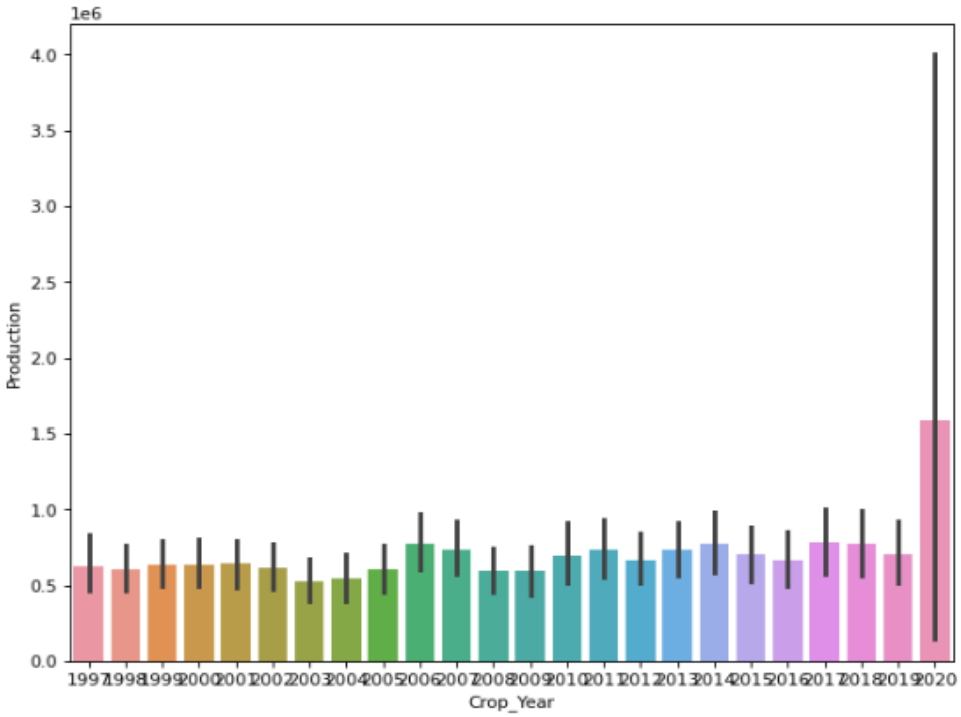
```
In [67]: plt.figure(figsize=(8,6),dpi=100)
         sns.lineplot(data=sugarcane_prod,x='Crop_Year',y='Production');
```



The Production is stable form 2006- 2018 and having drastic increase in 2019-2020.

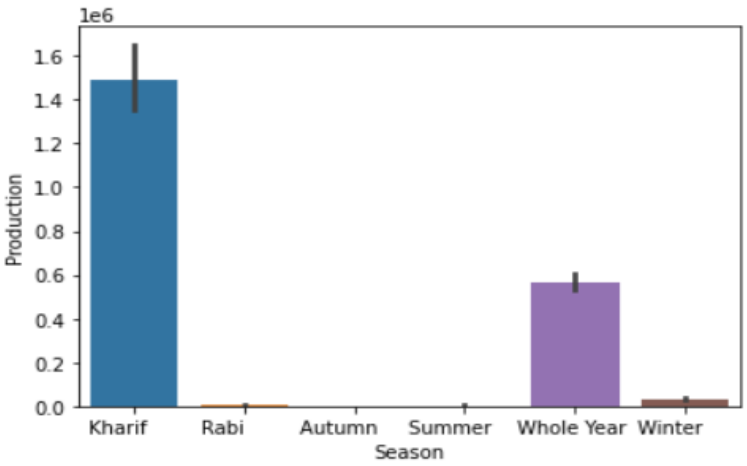
```
In [164]: plt.figure(figsize=(9,7),dpi=80)
          sns.barplot(data=sugarcane_prod,x='Crop_Year',y='Production');
          print('We can also see that result in the following graph')
```

```
We can also see that result in the following graph
```



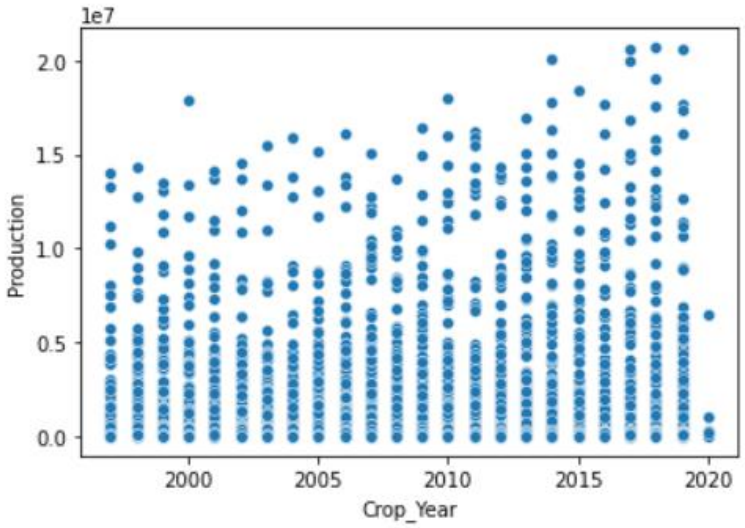
In 2020, there is a drastic increase in sugarcane production.

```
In [70]: sns.barplot(data=sugarcane_prod,x='Season',y='Production');
```



Sugarcane production increase in the whole year and kharif season.

```
In [71]: sns.scatterplot(data=sugarcane_prod,x='Crop_Year',y='Production');
```



The North and South west regions of India have more sugarcane production, which makes sense because they have tropical and sub-tropical climates. The warm and moist weather in these areas favours sugarcane growth.

3) Rice

```
In [72]: rice_prod = crop[crop['Crop'] == 'Rice']
```

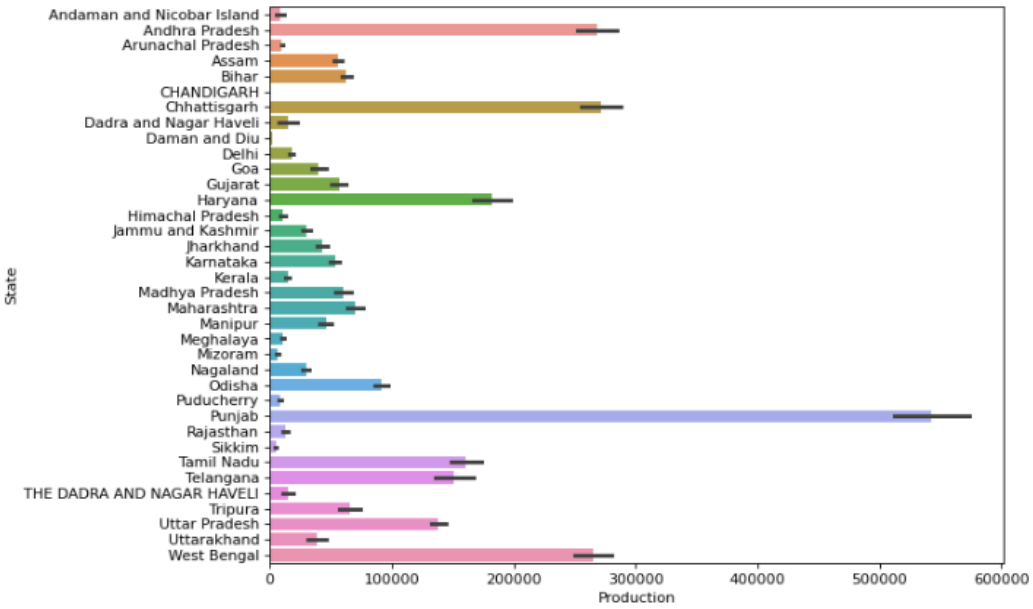
```
In [73]: rice_prod
```

Out[73]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
468	Andaman and Nicobar Island	NICOBARS	Rice	2007	Kharif	7333.75	21864.00	2.98
469	Andaman and Nicobar Island	NICOBARS	Rice	2008	Autumn	7900.00	14730.00	1.86
470	Andaman and Nicobar Island	NICOBARS	Rice	2009	Autumn	8140.00	16600.00	2.04
471	Andaman and Nicobar Island	NICOBARS	Rice	2000	Kharif	102.00	321.00	3.15
472	Andaman and Nicobar Island	NICOBARS	Rice	2001	Kharif	83.00	300.00	3.61
...	...	...	...	...	...	...	...	...
342053	West Bengal	PURULIA	Rice	2018	Summer	378.00	1144.00	3.03
342054	West Bengal	PURULIA	Rice	2018	Winter	281236.00	615025.00	2.19
342055	West Bengal	PURULIA	Rice	2019	Autumn	346.00	801.00	2.32
342056	West Bengal	PURULIA	Rice	2019	Summer	544.00	1644.00	3.02
342057	West Bengal	PURULIA	Rice	2019	Winter	285631.00	794280.00	2.78

21566 rows × 8 columns

```
In [166]: plt.figure(figsize=(9,7),dpi=80)
          sns.barplot(data=rice_prod,x='Production',y='State');
```



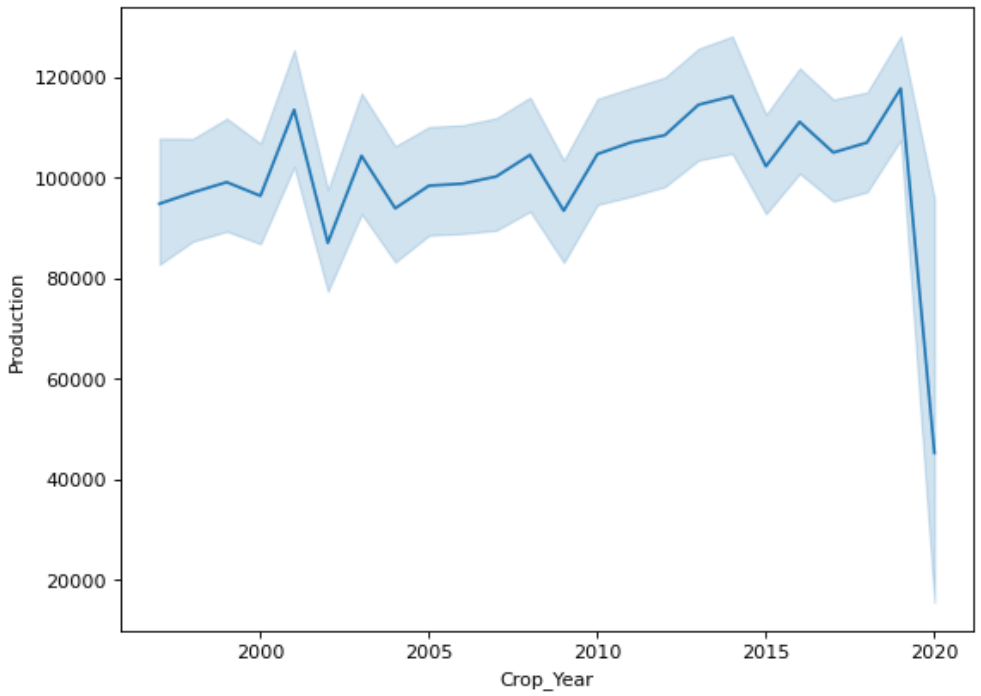
Punjab, Uttar Pradesh, Andhra Pradesh, followed by West Bengal has the highest Rice production.

```
In [75]: rice_prod.groupby('State').sum()['Production'].nlargest(10)

Out[75]: State
         West Bengal      338984869.00
         Uttar Pradesh    296108280.00
         Punjab           243042000.00
         Andhra Pradesh   239361201.00
         Odisha           153257182.00
         Tamil Nadu       132142328.00
         Bihar            131489958.00
         Chhattisgarh     116274060.00
         Assam             97989518.00
         Karnataka         82049324.00
         Name: Production, dtype: float64
```

West Bengal, UP, Punjab, AP are ranking high in Rice Production.

```
In [167]: plt.figure(figsize=(8,6),dpi=80)
          sns.lineplot(data=rice_prod,x='Crop_Year',y='Production');
```



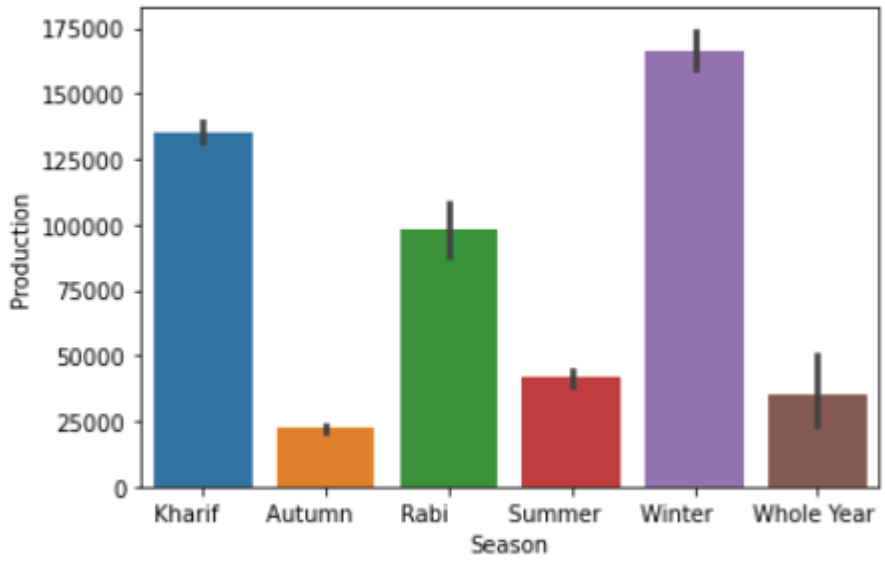
The year 2020, have the drastically fallen Production.

```
In [77]: rice_prod.groupby('Crop_Year').sum()['Production'].nsmallest(10)
```

```
Out[77]: Crop_Year
         2020     724429.00
         1997   61317562.00
         2002   71603326.00
         1998   83065010.00
         2000   83248654.00
         1999   83511141.00
         2004   84202290.00
         2003   88027062.00
         2009   88907038.00
         2005   89613933.00
         Name: Production, dtype: float64
```

The Year 2020, 1997, and 2002 have the lowest production of the rice.

```
In [78]: sns.barplot(data=rice_prod,x='Season',y='Production');
```



Winter is the best season for rice production.

4) Turmeric

```
In [79]: Turmeric_prod =crop[crop['Crop'] == 'Turmeric']
```

```
In [80]: Turmeric_prod
```

Out[80]:

	State	District	Crop	Crop_Year	Season	Area	Production	Yield
665	Andaman and Nicobar Island	NICOBARS	Turmeric	2007	Rabi	80.50	474.00	5.89
666	Andaman and Nicobar Island	NICOBARS	Turmeric	2008	Summer	82.00	381.00	4.65
667	Andaman and Nicobar Island	NICOBARS	Turmeric	2009	Summer	84.00	384.00	4.57
668	Andaman and Nicobar Island	NICOBARS	Turmeric	2005	Whole Year	10.00	70.00	7.00
669	Andaman and Nicobar Island	NICOBARS	Turmeric	2006	Whole Year	2.50	1.00	0.41
...	...	...	...	...	...	...	...	...
344256	West Bengal	PURULIA	Turmeric	2004	Whole Year	270.00	166.00	0.61
344257	West Bengal	PURULIA	Turmeric	2005	Whole Year	284.00	229.00	0.81
344258	West Bengal	PURULIA	Turmeric	2006	Whole Year	294.00	261.00	0.89
344259	West Bengal	PURULIA	Turmeric	2007	Whole Year	289.00	178.00	0.62
344260	West Bengal	PURULIA	Turmeric	2008	Whole Year	289.00	378.00	1.31

5561 rows × 8 columns

```
In [83]: Turmeric_prod.groupby('State').sum()['Production'].nlargest(10).reset_index()
```

Out[83]:

	State	Production
0	Tamil Nadu	2579682.00
1	Telangana	1984106.00
2	Karnataka	1595798.00
3	Madhya Pradesh	1512800.00
4	Odisha	387938.00
5	West Bengal	304953.00
6	Assam	274525.00
7	Manipur	250880.00
8	Meghalaya	249354.00
9	Kerala	160151.00

Tamil Nadu has the largest production in turmeric production.

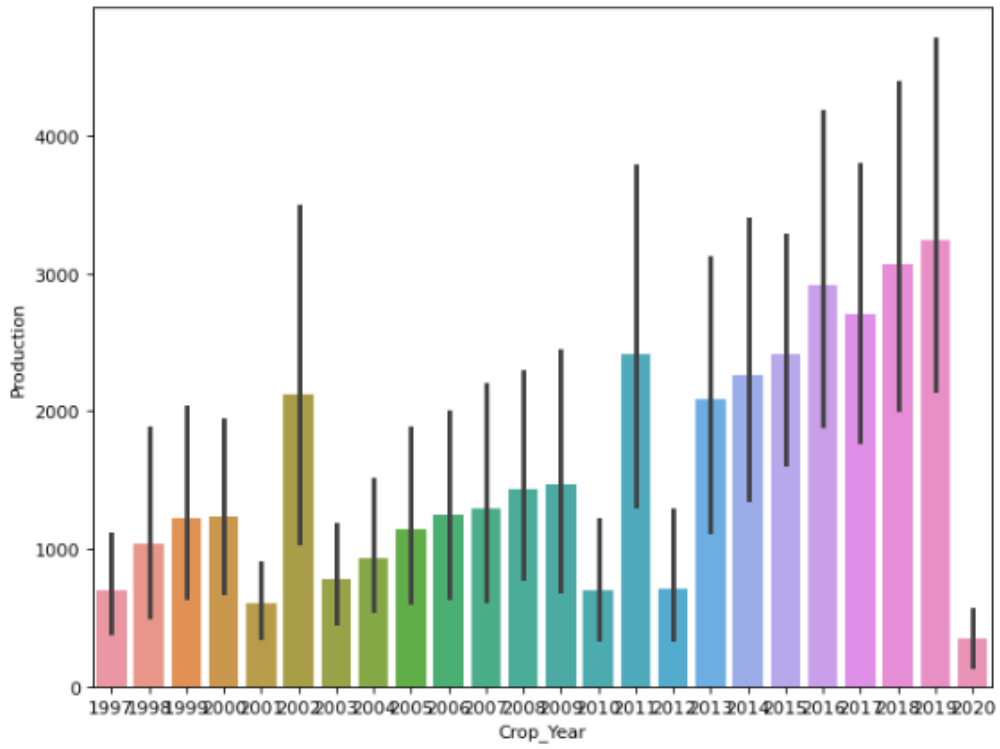
```
In [84]: Turmeric_prod.groupby('State').sum()['Production'].nsmallest(10).reset_index()
```

Out[84]:

	State	Production
0	Jammu and Kashmir	170.00
1	Puducherry	273.00
2	Himachal Pradesh	1868.00
3	Nagaland	4050.00
4	Rajasthan	8447.00
5	Andaman and Nicobar Island	9030.00
6	Chhattisgarh	19164.00
7	Uttarakhand	39440.00
8	Bihar	46063.00
9	Arunachal Pradesh	56220.00

Jammu and Kashmir has the least production in turmeric.

```
In [169]: plt.figure(figsize=(9,7),dpi=80)
          sns.barplot(data=Turmeric_prod,x='Crop_Year',y='Production');
```



Year 2016, 2018, 2019 have higher amount of the production.

CHAPTER – 4

DATA PRE-PROCESING

REMOVING NULL VALUES

Removing the null values

```
In [8]: crop.isnull().sum()
Out[8]: State           0
        District        0
        Crop            9
        Crop_Year       0
        Season          0
        Area            0
        Production    4948
        Yield           0
        dtype: int64

In [9]: crop.dropna(inplace=True)

In [10]: crop.isnull().sum()
Out[10]: State         0
         District      0
         Crop          0
         Crop_Year     0
         Season        0
         Area          0
         Production    0
         Yield         0
         dtype: int64
```

Removing the null values of the production variable.

CHECKING FOR DUPLICATES

Finding the Years of the data

```
In [13]: crop["Crop_Year"].unique()
Out[13]: array([2007, 2008, 2009, 2000, 2001, 2002, 2003, 2004, 2006, 2010, 2011,
                2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2005, 1997, 1998,
                1999, 2020], dtype=int64)
```

Checking for duplicates in the Crop year variable from the given dataset.

Finding the Seasons of the data

```
In [15]: crop["Season"].unique()
Out[15]: array(['Kharif     ', 'Rabi       ', 'Autumn     ', 'Summer     ',
                'Whole Year ', 'Winter     '], dtype=object)

In [16]: crop["Crop"].nunique()
Out[16]: 55

In [17]: crop["State"].nunique()
Out[17]: 37
```

Checking for duplicates in the season variable from the given dataset.

Finding the State of the data

```
In [18]: crop["State"].unique()
Out[18]: array(['Andaman and Nicobar Island', 'Andhra Pradesh',
                'Arunachal Pradesh', 'Assam', 'Bihar', 'CHANDIGARH',
                'Chhattisgarh', 'Dadra and Nagar Haveli', 'Daman and Diu', 'Delhi',
                'Goa', 'Gujarat', 'Haryana', 'Himachal Pradesh',
                'Jammu and Kashmir', 'Jharkhand', 'Karnataka', 'Kerala', 'Laddak',
                'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram',
                'Nagaland', 'Odisha', 'Puducherry', 'Punjab', 'Rajasthan',
                'Sikkim', 'Tamil Nadu', 'Telangana', 'THE DADRA AND NAGAR HAVELI',
                'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West Bengal'],
               dtype=object)
```

Checking for duplicates in the State variable from the given dataset.

```
In [34]: crop.groupby(['State','Crop','Crop_Year']).sum()['Production']
Out[34]: State                         Crop      Crop_Year
         Andaman and Nicobar Island    Arecanut  2000          7200.00
                                                 2001          7300.00
                                                 2002          7350.00
                                                 2003          6707.00
                                                 2004          4781.00
                                                                  ...
         West Bengal                   Wheat     2015        788503.00
                                                 2016        862712.00
                                                 2017        362744.00
                                                 2018        337751.00
                                                 2019        509970.00
         Name: Production, Length: 17705, dtype: float64
```

Year-wise crop production using group-by function.

CHAPTER – V

MODEL BUILDING

Feature Selection & Creating dummy variables for object datatype within the dataset

```
In [86]: crop_data = pd.get_dummies(data=crop)

In [87]: crop_data.head()
```

Out[87]:

	Crop_Year	Area	Production	Yield	State_Andaman and Nicobar Island	State_Andhra Pradesh	State_Arunachal Pradesh	State_Assam	State_Bihar	State_CHANDIGARH	...
0	2007	2439.60	3415.00	1.40	1	0	0	0	0	0	...
1	2007	1626.40	2277.00	1.40	1	0	0	0	0	0	...
2	2008	4147.00	3060.00	0.74	1	0	0	0	0	0	...
3	2008	4147.00	2660.00	0.64	1	0	0	0	0	0	...
4	2009	4153.00	3120.00	0.75	1	0	0	0	0	0	...

5 rows × 809 columns

Top rows of the given dataset.

```
In [88]: from sklearn.model_selection import train_test_split

In [89]: X = crop_data.drop('Production',axis=1)
         X.head()
```

Out[89]:

	Crop_Year	Area	Yield	State_Andaman and Nicobar Island	State_Andhra Pradesh	State_Arunachal Pradesh	State_Assam	State_Bihar	State_CHANDIGARH	State_Chhattisgarh
0	2007	2439.60	1.40	1	0	0	0	0	0	0
1	2007	1626.40	1.40	1	0	0	0	0	0	0
2	2008	4147.00	0.74	1	0	0	0	0	0	0
3	2008	4147.00	0.64	1	0	0	0	0	0	0
4	2009	4153.00	0.75	1	0	0	0	0	0	0

5 rows × 808 columns

Dropping the variable production and showing the result of top rows.

```
In [90]: y = crop_data['Production']
         y.head()

Out[90]: 0    3415.00
         1    2277.00
         2    3060.00
         3    2660.00
         4    3120.00
         Name: Production, dtype: float64
```

Production is the dependent or target variable.

```
In [91]: X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.33, random_state=42)

In [140]: X_train.shape
Out[140]: (227075, 808)

In [92]: X_test.shape
Out[92]: (111843, 808)

In [138]: y_train.shape
Out[138]: (227075,)

In [93]: y_test.shape
Out[93]: (111843,)
```

Creating Model

```
In [94]: from sklearn.linear_model import LinearRegression

In [95]: crop_model = LinearRegression()
```

Creating a linear regression model.

Training the Model

```
In [96]: crop_model.fit(X_train,y_train)
Out[96]: LinearRegression()
```

Fit the model with the train data.

Prediction

```
In [97]: crop_predictions = crop_model.predict(X_test)
         crop_predictions
Out[97]: array([ -162068.90023768, -1208872.76231915,   755736.96225636, ...,
                -2564235.82947292,  -466267.00357063,   352481.33998434])
```

Predicting the test data.

```
In [98]: crop_model.coef_
```

```
Out[98]: array([ 3.91113155e+03,  2.68731612e+01,  4.42557694e+03, -1.23511839e+06,
                -6.47858685e+05,  4.50962467e+05, -1.14584698e+07,  1.91787123e+05,
                 3.14331475e+05,  3.18976977e+04,  1.17970545e+06,  7.66528180e+05,
                 2.26145304e+05,  4.17970797e+05, -8.07891204e+04, -9.10868980e+04,
                 2.21925029e+05,  5.45271869e+04,  7.62640212e+04,  2.04908582e+05,
                 2.07741628e+07,  1.40144893e+05, -1.67436495e+05, -3.52144276e+05,
                 4.13806397e+05,  8.22323595e+05,  4.07917892e+05,  1.71761763e+05,
                 2.31159531e+05, -6.76353342e+06, -4.21592416e+05,  7.49790704e+04,
                 3.12701868e+05, -1.88059505e+06, -6.53591586e+05,  3.36630810e+05,
                 1.09124210e+05, -2.08030343e+05, -2.66515694e+05, -3.70490381e+06,
                 2.55881088e+05,  1.64105317e+05, -8.97274464e+03, -8.13119342e+04,
                -4.71802080e+05, -2.09873763e+05, -2.74555749e+05, -1.96952594e+05,
                -8.55428845e+06, -1.01910382e+05,  8.73781432e+05,  4.12138032e+05,
                -1.53097541e+05,  3.47884581e+05, -5.60971701e+05,  1.26382336e+05,
                 2.76043164e+04, -9.40893179e+04, -3.20789270e+05, -2.79064760e+05,
                -2.44892014e+05,  2.42135831e+05,  1.72114366e+05, -1.47513961e+06,
                -1.34362906e+05,  2.01522163e+05,  7.22759763e+04,  3.02365891e+05,
                -1.40563899e+05,  9.24909061e+04, -1.83740681e+06, -1.33250654e+05,
                -6.34863992e+03,  6.91465045e+04, -2.40877107e+05, -1.40877877e+05,
```

```
In [99]: crop_model.intercept_
```

```
Out[99]: -7334731.76254124
```

Now even though we have a set of predictions with us we need a way to actually tell if the model is accurate enough, let's test that using metrics.

```
In [100]: predicted_crop_val = pd.DataFrame({'Actual':y_test,'Predicted':crop_predictions})
          predicted_crop_val
```

Out[100]:

	Actual	Predicted
219700	397.00	-162068.90
230130	1000.00	-1208872.76
330217	14970.00	755736.96
141812	9058.00	-2239271.60
211044	100.00	132324.32
...	...	...
332610	65373.00	-203406.59
219398	1.00	491195.42
11372	125.00	-2564235.83
234020	2815.00	-466267.00
48886	15473.00	352481.34

111843 rows × 2 columns

We can see that from above table the predicted and actual values don't match.

Model evaluation

```
In [101]: from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score

In [102]: crop['Production'].mean()
Out[102]: 962629.8535988056

In [103]: crop_predictions.mean()
Out[103]: 952012.7160808861

In [104]: mean_absolute_error(y_test,crop_predictions)
Out[104]: 2612618.905912449

In [105]: mean_squared_error(y_test,crop_predictions)
Out[105]: 375037273142427.94

In [106]: np.sqrt(mean_squared_error(y_test,crop_predictions))
Out[106]: 19365879.095523342

In [107]: def mape(actual, pred):
              actual, pred = np.array(actual), np.array(pred)
              return np.mean(np.abs((actual - pred) / actual)) * 100

          mape(y_test,crop_predictions)
Out[107]: 6594084.634129278
```

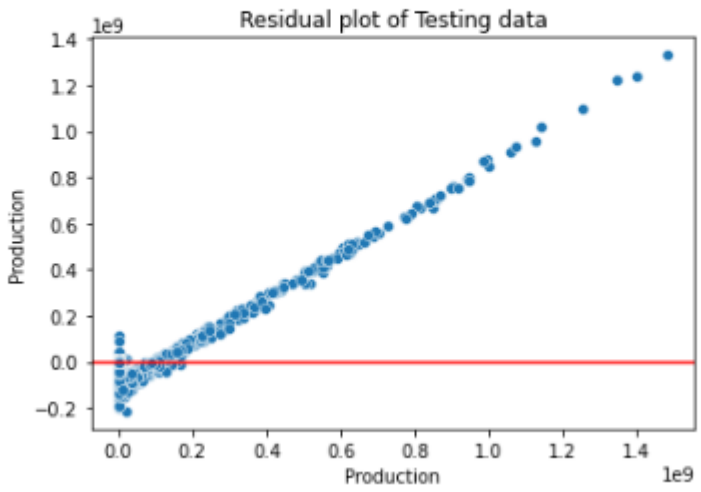
Checking the residual plots

```
In [108]: test_residuals = y_test - crop_predictions

In [109]: test_residuals
Out[109]: 219700    162465.90
          230130   1209872.76
          330217   -740766.96
          141812   2248329.60
          211044   -132224.32
                      ...
          332610    268779.59
          219398   -491194.42
          11372    2564360.83
          234020    469082.00
          48886    -337008.34
          Name: Production, Length: 111843, dtype: float64
```

Checking the residual plots error. The residues form a pattern which means as data spreads, the residuals also spreads across uniformly.

In [110]:
```
sns.scatterplot(x=y_test,y=test_residuals)
plt.axhline(y=0,color='red')
plt.title('Residual plot of Testing data');
```



In [111]:
```
r = r2_score(y_test,crop_predictions)
print("R2score to predict using Linear Regression is ",r)
```

```
R2score to predict using Linear Regression is  0.25358497319619233
```

Results of training data

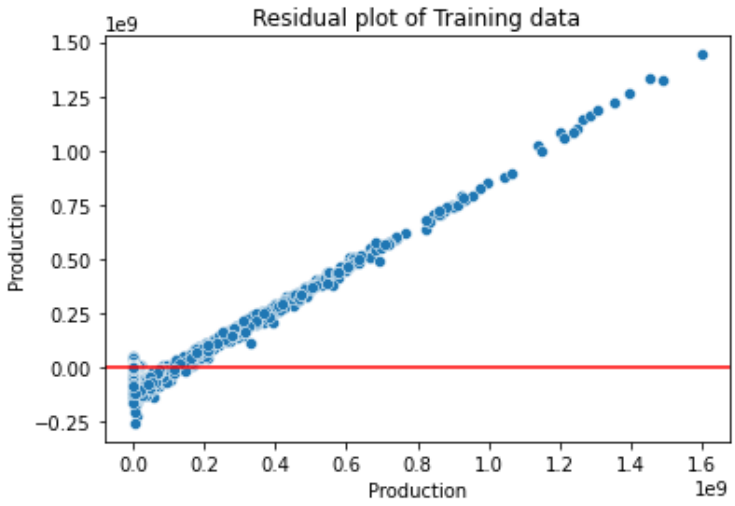
In [112]:
```
train_set_predictions = crop_model.predict(X_train)
train_set_predictions
```

Out[112]:
```
array([   80367.79229452,  -158750.09355451, -1112349.41674228, ...,
        6592543.1027372 ,   190595.96283247,  -444052.67719723])
```

In [113]:
```
train_set_predictions.mean()
```

Out[113]:
```
936148.5508180129
```

```
In [114]: sns.scatterplot(x=y_train,y=y_train-train_set_predictions)
          plt.axhline(y=0,color='red')
          plt.title('Residual plot of Training data');
```




```
In [115]: def mape(actual, pred):
              actual, pred = np.array(actual), np.array(pred)
              return np.mean(np.abs((actual - pred) / actual)) * 100

          mape(y_train,train_set_predictions)
```

```
Out[115]: 6514852.641021204
```

From the metrics that we have used to test our model we can straight away say that Linear Regression was not a good choice of algorithm for this data set.

The residual plot is a clear explanation of this, we can see the that the residual points are not randomly distributed around the actual points themselves.

Decision Tree model

```
In [117]: from sklearn.tree import DecisionTreeRegressor

In [118]: dtree = DecisionTreeRegressor()

In [119]: dtree.fit(X_train, y_train)

Out[119]: DecisionTreeRegressor()
```

```
In [120]: y_pred = dtree.predict(X_test)

In [121]: mean_absolute_error(y_test,y_pred)
Out[121]: 64786.01462764769

In [122]: y_pred
Out[122]: array([  393.,  1000., 15267., ...,   125.,  2805., 15652.])

In [123]: pd.DataFrame({"Y_test": y_test,
                        "Y_pred": y_pred})
```

Out[123]:

	Y_test	Y_pred
219700	397.00	393.00
230130	1000.00	1000.00
330217	14970.00	15267.00
141812	9058.00	9300.00
211044	100.00	100.00
...	...	...
332610	65373.00	64812.00
219398	1.00	1.00
11372	125.00	125.00
234020	2815.00	2805.00
48886	15473.00	15652.00

111843 rows × 2 columns

The decision tree model seems to be performing well for this data as the data contains large range of values in the production variable and hence, the regression model is unable to perform well whereas the decision tree has the prediction as close as to the original values.