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

```
df=pd.read_csv("/content/Crop Production data.csv")
```

```
df.head()
```

	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
	Andaman and			Whole			

```
df.shape
```

```
    (246091, 7)
```

```
df.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
```

```
df.isnull().sum()
```

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

```
df.dropna(subset=["Production"],axis=0,inplace=True)
```

```
df.shape
```

```
    (242361, 7)
```

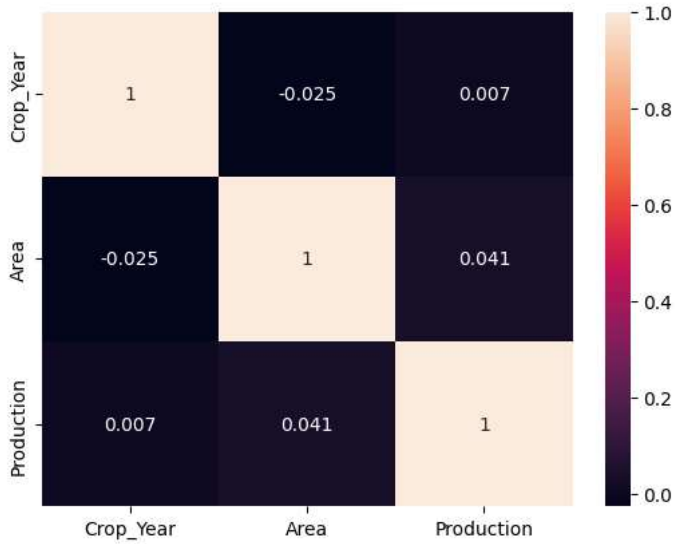
```
df.isnull().sum()
```

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

Checking for Correlation between variables

```
plt.tick_params(labelsize=10)
sns.heatmap(df.corr(),annot=True)
```

```
<ipython-input-63-d21a8883155f>:2: FutureWarning: The default value of numeric_only in Dat
  sns.heatmap(df.corr(),annot=True)
<Axes: >
```



```
df.columns
```

```
Index(['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
       'Production'],
      dtype='object')
```

```
df.State_Name.unique()
```

```
array(['Andaman and Nicobar Islands', 'Andhra Pradesh',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
       'Chhattisgarh', 'Dadra and Nagar Haveli', 'Goa', 'Gujarat',
       'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir ', 'Jharkhand',
       'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
       'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Puducherry',
       'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Telangana ',
       'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West Bengal'],
      dtype=object)
```

```
df.State_Name.nunique()
```

```
33
```

```
df.State_Name.value_counts()
```

```
Uttar Pradesh                  33189
Madhya Pradesh                 22604
Karnataka                      21079
Bihar                          18874
Assam                          14622
Odisha                         13524
Tamil Nadu                     13266
Maharashtra                    12496
Rajasthan                      12066
Chhattisgarh                   10368
West Bengal                     9597
Andhra Pradesh                  9561
Gujarat                         8365
Telangana                       5591
Uttarakhand                     4825
Haryana                         4540
Kerala                          4003
Nagaland                        3904
Punjab                          3143
Meghalaya                       2867
Arunachal Pradesh               2545
Himachal Pradesh                2456
Jammu and Kashmir               1632
```

```
Tripura                         1412
Manipur                         1266
Jharkhand                       1266
Mizoram                          954
Puducherry                       872
Sikkim                           714
Dadra and Nagar Haveli           263
Goa                              207
Andaman and Nicobar Islands      201
Chandigarh                        89
Name: State_Name, dtype: int64
```

```
df.District_Name.nunique()
```

```
646
```

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

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

Crop Year Variable

```
df.describe()
```

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

```
df.Crop_Year.nunique()
```

```
19
```

```
df.Crop_Year.min()
```

```
1997
```

```
df.Crop_Year.max()
```

```
2015
```

```
df.Crop_Year.value_counts()
```

```
2003    17139
2002    16536
2007    14269
2008    14230
2006    13976
2004    13858
2010    13793
2011    13791
2009    13767
```

```
2000    13553
2005    13519
2013    13475
2001    13293
2012    13184
1999    12441
1998    11262
2014    10815
1997     8899
2015      561
Name: Crop_Year, dtype: int64
```

Double-click (or enter) to edit

```
print("Number of Unique Season is --> ",df.Season.nunique())
print("Unique Season is --> \n ", df.Season.unique())
print("Maximum Season is -->", df.Season.max())
print("Number of value in each season is --> \n", df.Season.value_counts())
```

```
Number of Unique Season is -->  6
Unique Season is -->
  ['Kharif     ' 'Whole Year ' 'Autumn     ' 'Rabi       ' 'Summer     '
 'Winter     ']
Maximum Season is --> Winter
Number of value in each season is -->
 Kharif         94283
Rabi           66160
Whole Year     56127
Summer         14811
Winter          6050
Autumn          4930
Name: Season, dtype: int64
```

```
print(df.Crop.nunique())
```

```
124
```

```
print(df.Crop.unique())
```

```
['Arecanut' 'Other Kharif pulses' 'Rice' 'Banana' 'Cashewnut' 'Coconut '
 'Dry ginger' 'Sugarcane' 'Sweet potato' 'Tapioca' 'Black pepper'
 'Dry chillies' 'other oilseeds' 'Turmeric' 'Maize' 'Moong(Green Gram)'
 'Urad' 'Arhar/Tur' 'Groundnut' 'Sunflower' 'Bajra' 'Castor seed'
 'Cotton(lint)' 'Horse-gram' 'Jowar' 'Korra' 'Ragi' 'Tobacco' 'Gram'
 'Wheat' 'Masoor' 'Sesamum' 'Linseed' 'Safflower' 'Onion'
 'other misc. pulses' 'Samai' 'Small millets' 'Coriander' 'Potato'
 'Other  Rabi pulses' 'Soyabean' 'Beans & Mutter(Vegetable)' 'Bhindi'
 'Brinjal' 'Citrus Fruit' 'Cucumber' 'Grapes' 'Mango' 'Orange'
 'other fibres' 'Other Fresh Fruits' 'Other Vegetables' 'Papaya'
 'Pome Fruit' 'Tomato' 'Mesta' 'Cowpea(Lobia)' 'Lemon' 'Pome Granet'
 'Sapota' 'Cabbage' 'Rapeseed &Mustard' 'Peas  (vegetable)' 'Niger seed'
 'Bottle Gourd' 'Varagu' 'Garlic' 'Ginger' 'Oilseeds total' 'Pulses total'
 'Jute' 'Peas & beans (Pulses)' 'Blackgram' 'Paddy' 'Pineapple' 'Barley'
 'Sannhamp' 'Khesari' 'Guar seed' 'Moth' 'Other Cereals & Millets'
 'Cond-spcs other' 'Turnip' 'Carrot' 'Redish' 'Arcanut (Processed)'
 'Atcanut (Raw)' 'Cashewnut Processed' 'Cashewnut Raw' 'Cardamom' 'Rubber'
 'Bitter Gourd' 'Drum Stick' 'Jack Fruit' 'Snak Guard' 'Tea' 'Coffee'
 'Cauliflower' 'Other Citrus Fruit' 'Water Melon' 'Total foodgrain'
 'Kapas' 'Colocosia' 'Lentil' 'Bean' 'Jobster' 'Perilla' 'Rajmash Kholar'
 'Ricebean (nagadal)' 'Ash Gourd' 'Beet Root' 'Lab-Lab' 'Ribed Guard'
 'Yam' 'Pump Kin' 'Apple' 'Peach' 'Pear' 'Plums' 'Litchi' 'Ber'
 'Other Dry Fruit' 'Jute & mesta']
```

```
print(df.Crop.max())
```

```
other oilseeds
```

```
print(df.Crop.value_counts().head(20))
```

```
Rice                     15082
Maize                    13787
Moong(Green Gram)        10106
Urad                      9710
Sesamum                   8821
Groundnut                 8770
```

```
Wheat                     7878
Sugarcane                 7827
Rapeseed &Mustard         7533
Arhar/Tur                 7476
Gram                      7227
Jowar                     6990
Onion                     6984
Potato                    6914
Dry chillies              6421
Sunflower                 5483
Bajra                     5379
Small millets             4593
Peas & beans (Pulses)     4447
Cotton(lint)              4382
Name: Crop, dtype: int64
```

Area Variable

```
print("Number of Different unique area is -->",df.Area.nunique())
print("Maximum area is -->", df.Area.max())
```

```
Number of Different unique area is --> 38391
Maximum area is --> 8580100.0
```

```
print(df.Area.value_counts().head(10))
```

```
1.0      3573
2.0      3140
100.0    2621
3.0      2478
4.0      2182
5.0      2090
6.0      1750
200.0    1671
10.0     1590
7.0      1555
Name: Area, dtype: int64
```

```
print(df.Area.value_counts().tail(10))
```

```
63107.0     1
13655.0     1
95399.0     1
71644.0     1
17459.0     1
25569.0     1
19349.0     1
90302.0     1
39698.0     1
279151.0    1
Name: Area, dtype: int64
```

Production Variable

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

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

```
df.Production.max()
```

```
1250800000.0
```

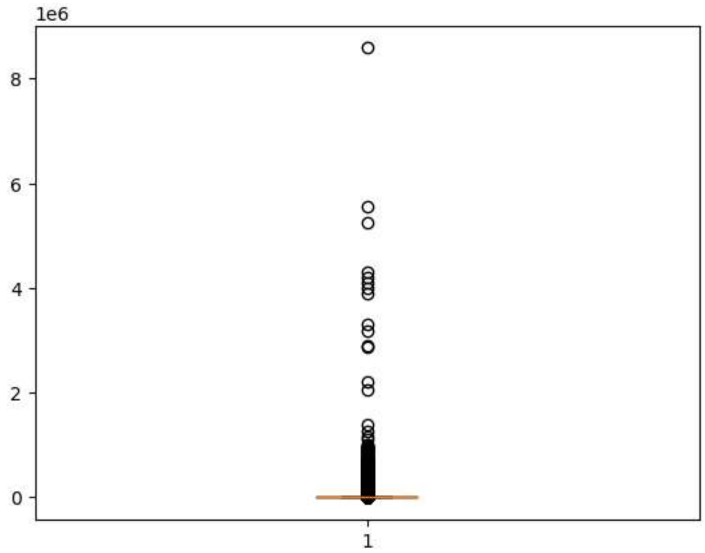
```
df.Production.value_counts(ascending = False)
```

```
1.000000e+00    4028
0.000000e+00    3523
1.000000e+02    3521
2.000000e+00    2964
3.000000e+00    2311
                ...
2.120000e+08       1
1.070000e+00       1
2.293410e+05       1
1.870600e+04       1
5.978990e+05       1
Name: Production, Length: 51627, dtype: int64
```

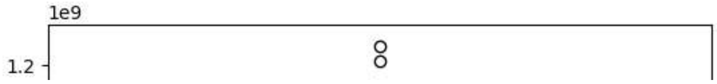
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 242361 entries, 0 to 246090
Data columns (total 7 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   State_Name     242361 non-null  object 
 1   District_Name  242361 non-null  object 
 2   Crop_Year      242361 non-null  int64  
 3   Season         242361 non-null  object 
 4   Crop           242361 non-null  object 
 5   Area           242361 non-null  float64
 6   Production     242361 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 14.8+ MB
```

```
plt.boxplot(df.Area);
```



```
plt.boxplot(df.Production);
```



Bivariate Anaysis

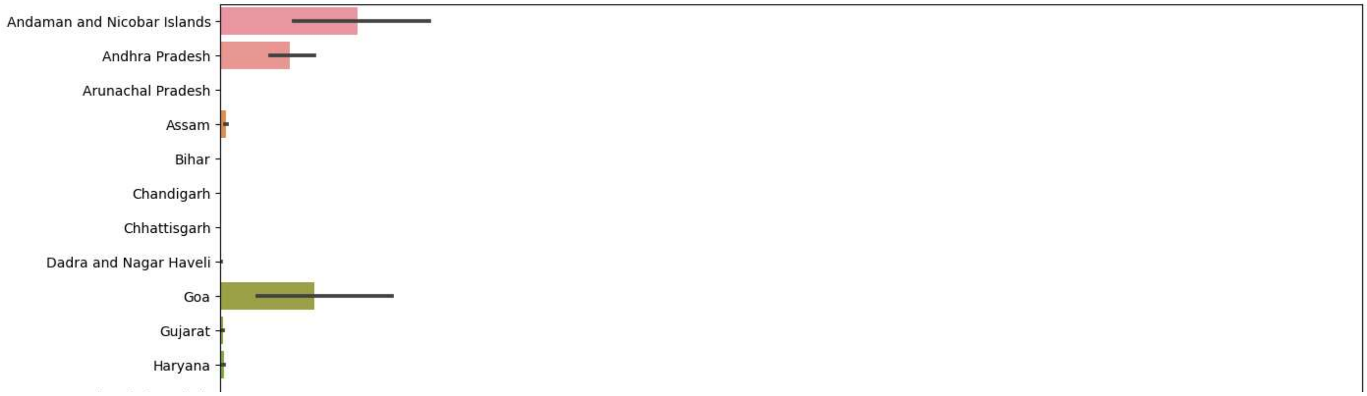
```
Prod = df.groupby(by = df.State_Name)['Production','State_Name'].sum().reset_index().sort_values(by = 'Production', ascending = False).head(1
Prod
```

```
<ipython-input-89-b30c6e812fa8>:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecat
  Prod = df.groupby(by = df.State_Name)['Production','State_Name'].sum().reset_index().sort_values(by = 'Production', ascending = False)
<ipython-input-89-b30c6e812fa8>:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future v
  Prod = df.groupby(by = df.State_Name)['Production','State_Name'].sum().reset_index().sort_values(by = 'Production', ascending = False)
```

	State_Name	Production
15	Kerala	9.788005e+10
1	Andhra Pradesh	1.732459e+10
27	Tamil Nadu	1.207644e+10
30	Uttar Pradesh	3.234493e+09
3	Assam	2.111752e+09
32	West Bengal	1.397904e+09
17	Maharashtra	1.263641e+09
14	Karnataka	8.634298e+08
0	Andaman and Nicobar Islands	7.182232e+08
24	Punjab	5.863850e+08

- Kerala is the Top State in production as we can see in the above

```
plt.figure(figsize= (15,15))
sns.barplot(x=df['Production'],y= df["State_Name"], orient='h');
```



```
df.describe()
```

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

```
#Zone-Wise Production - 1997-2014
north_india = ['Jammu and Kashmir', 'Punjab', 'Himachal Pradesh', 'Haryana', 'Uttarakhand', 'Uttar Pradesh', 'Chandigarh']
east_india = ['Bihar', 'Odisha', 'Jharkhand', 'West Bengal']
south_india = ['Andhra Pradesh', 'Karnataka', 'Kerala' ,'Tamil Nadu', 'Telangana']
west_india = ['Rajasthan' , 'Gujarat', 'Goa','Maharashtra']
central_india = ['Madhya Pradesh', 'Chhattisgarh']
north_east_india = ['Assam', 'Sikkim', 'Nagaland', 'Meghalaya', 'Manipur', 'Mizoram', 'Tripura', 'Arunachal Pradesh']
ut_india = ['Andaman and Nicobar Islands', 'Dadra and Nagar Haveli', 'Puducherry']
```

```
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

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

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

```
df.Zones.value_counts()
```

```
South Zone      53500
North Zone      49874
East Zone       43261
West Zone       33134
```

```
     Central Zone    32972
     NE Zone         28284
     Union Terr       1336
     Name: Zones, dtype: int64
```

```
crop=df['Crop']
def cat_crop(crop):
    for i in ['Rice','Maize','Wheat','Barley','Varagu','Other Cereals & Millets','Ragi','Small millets','Bajra','Jowar', 'Paddy','Total foodg
        if crop==i:
            return 'Cereal'
    for i in ['Moong','Urad','Arhar/Tur','Peas & beans','Masoor',
              'Other Kharif pulses','other misc. pulses','Ricebean (nagadal)',
              'Rajmash Kholar','Lentil','Samai','Blackgram','Korra','Cowpea(Lobia)',
              'Other  Rabi pulses','Other Kharif pulses','Peas & beans (Pulses)','Pulses total','Gram']:
        if crop==i:
            return 'Pulses'
    for i in ['Peach','Apple','Litchi','Pear','Plums','Ber','Sapota','Lemon','Pome Granet',
               'Other Citrus Fruit','Water Melon','Jack Fruit','Grapes','Pineapple','Orange',
               'Pome Fruit','Citrus Fruit','Other Fresh Fruits','Mango','Papaya','Coconut','Banana']:
        if crop==i:
            return 'Fruits'
    for i in ['Bean','Lab-Lab','Moth','Guar seed','Soyabean','Horse-gram']:
        if crop==i:
            return 'Beans'
    for i in ['Turnip','Peas','Beet Root','Carrot','Yam','Ribed Guard','Ash Gourd ','Pump Kin','Redish','Snak Guard','Bottle Gourd',
              'Bitter Gourd','Cucumber','Drum Stick','Cauliflower','Beans & Mutter(Vegetable)','Cabbage',
              'Bhindi','Tomato','Brinjal','Khesari','Sweet potato','Potato','Onion','Tapioca','Colocosia']:
              if crop==i:
                return 'Vegetables'
    for i in ['Perilla','Ginger','Cardamom','Black pepper','Dry ginger','Garlic','Coriander','Turmeric','Dry chillies','Cond-spcs other']:
        if crop==i:
            return 'spices'
    for i in ['other fibres','Kapas','Jute & mesta','Jute','Mesta','Cotton(lint)','Sannhamp']:
        if crop==i:
            return 'fibres'
    for i in ['Arcanut (Processed)','Atcanut (Raw)','Cashewnut Processed','Cashewnut Raw','Cashewnut','Arecanut','Groundnut']:
        if crop==i:
            return 'Nuts'
    for i in ['other oilseeds','Safflower','Niger seed','Castor seed','Linseed','Sunflower','Rapeseed &Mustard','Sesamum','Oilseeds total']:
        if crop==i:
            return 'oilseeds'
    for i in ['Tobacco','Coffee','Tea','Sugarcane','Rubber']:
        if crop==i:
            return 'Commercial'

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

```
df["cat_crop"].value_counts()
```

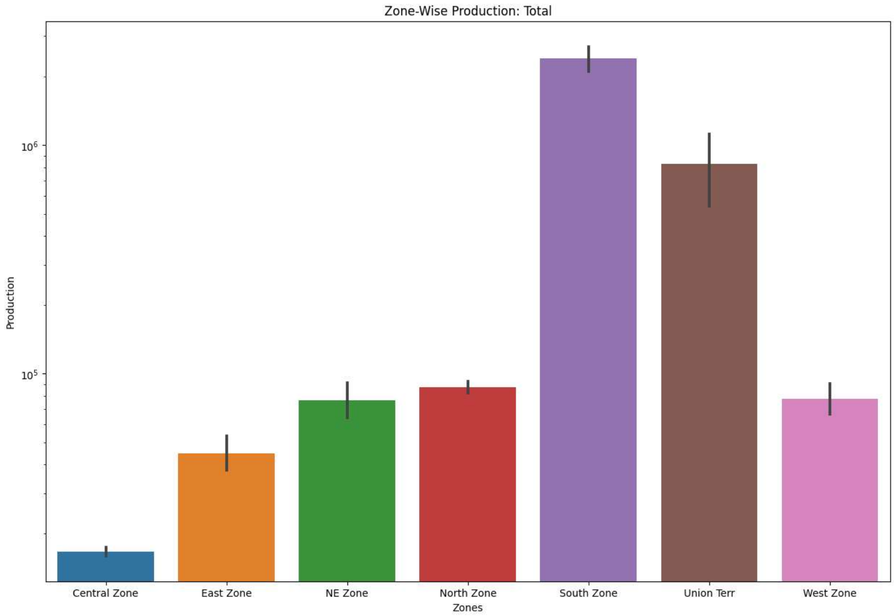
```
     Cereal        63283
     Pulses        40898
     oilseeds      33801
     Vegetables    23154
     spices        21638
     Nuts          11472
     Commercial    10561
     fibres         9785
     Beans          9115
     Fruits         6153
     Name: cat_crop, dtype: int64
```

```
data_explore = df.copy()
```

Zonal distribution of crops:

```
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(x = data_explore.Zones.sort_values(ascending=True), y = data_explore.Production)
plt.yscale('log')
plt.title('Zone-Wise Production: Total')
```

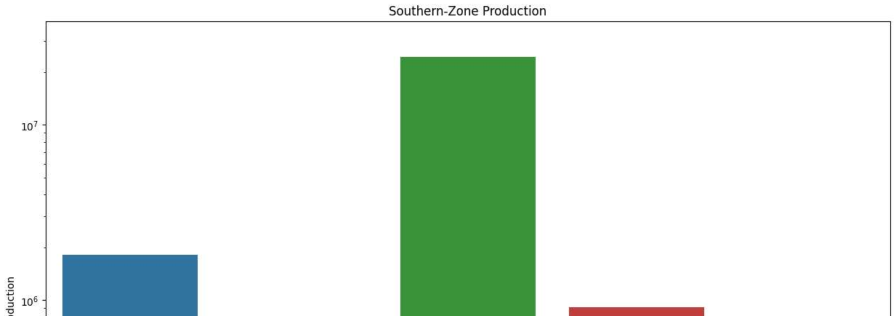
```
Text(0.5, 1.0, 'Zone-Wise Production: Total')
```



- South Zone has maximum production

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

```
Text(0.5, 1.0, 'Southern-Zone Production')
```

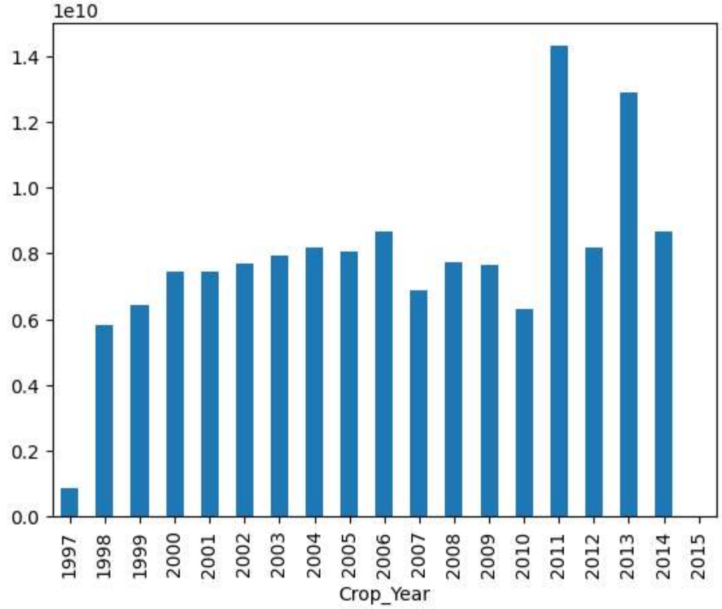


- In South zone, Kerala stata has more dominant in production

Yearwise Production Status

```
plt.tick_params(labelsize=10)
data_explore.groupby("Crop_Year")["Production"].agg("sum").plot.bar()
```

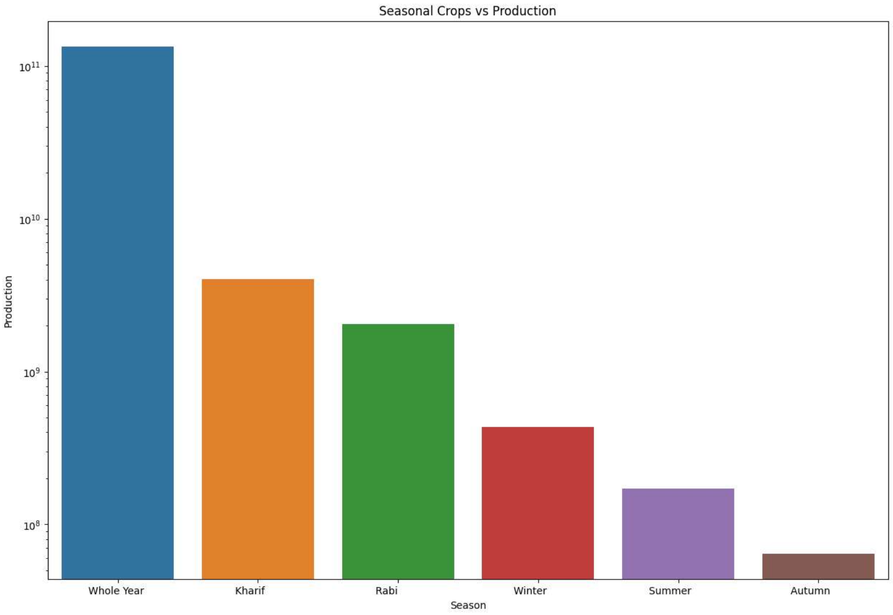
```
<Axes: xlabel='Crop_Year'>
```



Season wise Production Status:

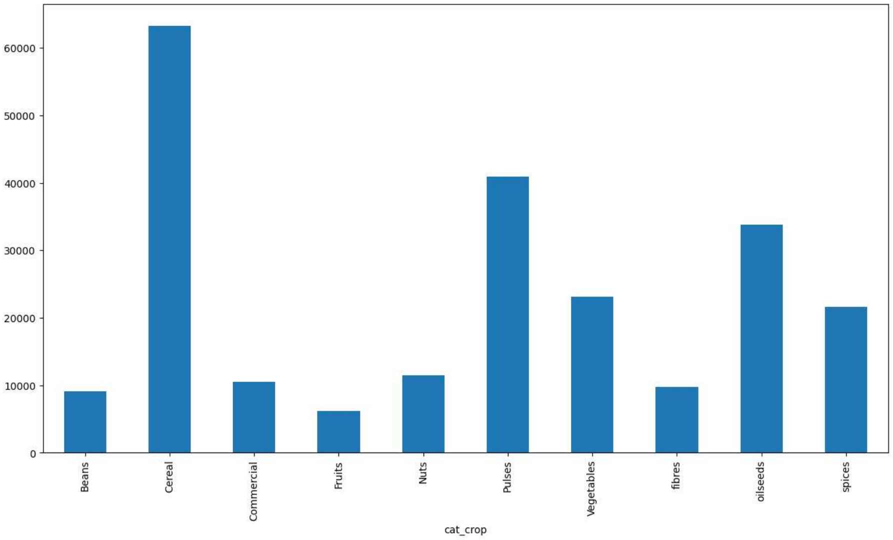
```
#Season vs Production
df_season=data_explore.copy()
season = df_season.groupby(by='Season')['Production'].sum().reset_index().sort_values(by='Production', ascending=False).head(10)
season
fig, ax = plt.subplots(figsize=(15,10))
sns.barplot(x = season.Season, y = season.Production,errwidth=0)
plt.yscale('log')
plt.title('Seasonal Crops vs Production')
```

```
Text(0.5, 1.0, 'Seasonal Crops vs Production')
```



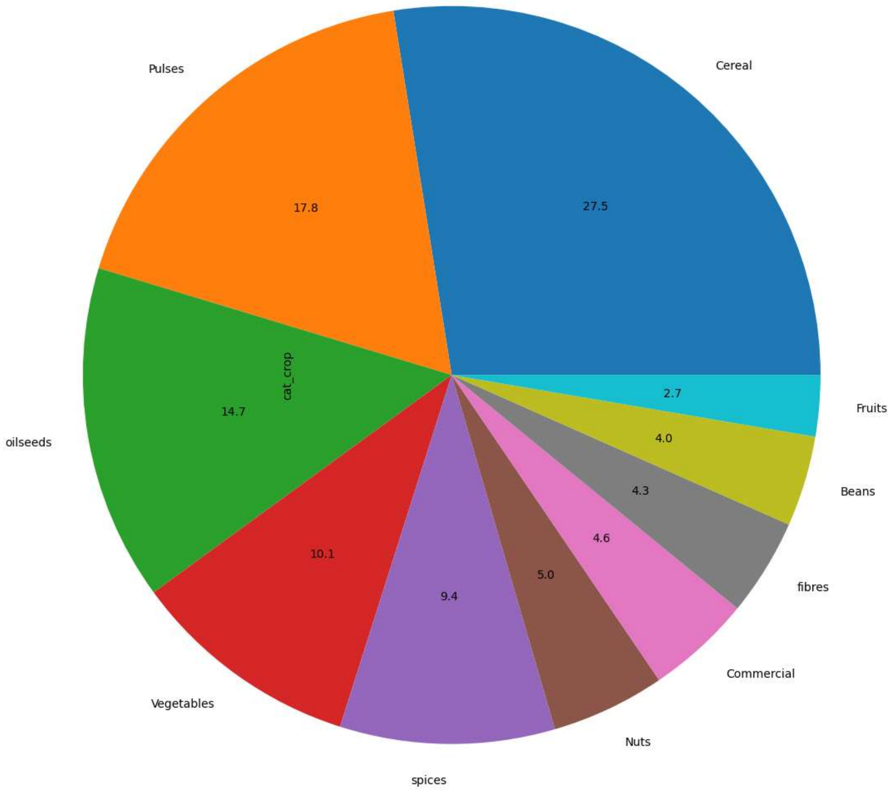
Crop wise Production plot describing production values for all crop types.

```
plt.figure(figsize=(15,8))
plt.tick_params(labelsize=10)
data_explore.groupby("cat_crop")["Production"].agg("count").plot.bar()
plt.show()
```



Different proportion of Crop Categories for India:

```
df1=data_explore["cat_crop"].value_counts()
df1.plot(radius=3,kind="pie",autopct="%1.1f",pctdistance=0.6)
plt.tick_params(labelsize=5)
```



```
df_2 = pd.crosstab(data_explore['State_Name'], data_explore['cat_crop'])
df_2
```

cat_crop State_Name	Beans	Cereal	Commercial	Fruits	Nuts	Pulses	Vegetables	fibres	oilseeds	spices
Andaman and Nicobar Islands	0	20	15	16	37	9	20	0	11	52
Andhra Pradesh	386	2264	474	502	674	1336	1046	333	1101	802
Arunachal Pradesh	26	1021	168	0	26	67	257	0	343	637
Assam	0	2952	854	920	400	2234	1781	1284	2097	1338
Bihar	280	6108	756	226	130	3731	1775	924	2504	1396
Chandigarh	0	39	0	0	0	14	26	0	7	0
Chhattisgarh	646	1805	316	264	261	2087	1143	535	1496	1288
Dadra and Nagar Haveli	0	116	12	9	9	64	0	13	30	1
Goa	0	62	22	16	47	32	0	0	0	12
Gujarat	403	2466	372	157	683	1521	473	327	1029	512
Haryana	108	1427	259	52	126	860	463	257	543	248
Himachal Pradesh	179	726	67	0	54	530	214	37	236	345
Jammu and Kashmir	12	562	42	24	7	307	196	44	233	115
Jharkhand	0	575	16	0	0	304	247	0	124	0
Karnataka	1096	5295	615	598	1470	2776	1763	605	3135	2588
Kerala	3	819	236	437	536	13	636	12	168	863
Madhya Pradesh	962	5115	826	659	768	3993	2738	922	3281	2739
Maharashtra	477	4009	458	83	868	2326	56	465	3189	0
Manipur	31	151	40	228	4	160	347	12	49	226
Meghalaya	113	606	182	162	143	314	399	177	329	442
Mizoram	42	230	123	0	15	213	96	64	143	0
Nagaland	211	1054	160	0	144	873	302	197	718	131
Odisha	629	3871	607	0	1156	1760	909	284	2335	912

- **Uttar Pradesh** is topping in producing more crop categories than any other Indian state.

```
data_explore["Crop"].value_counts()[:5]
```

```
Rice                15082
Maize               13787
Moong(Green Gram)   10106
Urad                 9710
Sesamum              8821
Name: Crop, dtype: int64
```

Uttar Pradesh 1112 9719 1741 269 958 6549 3734 724 4028 2529

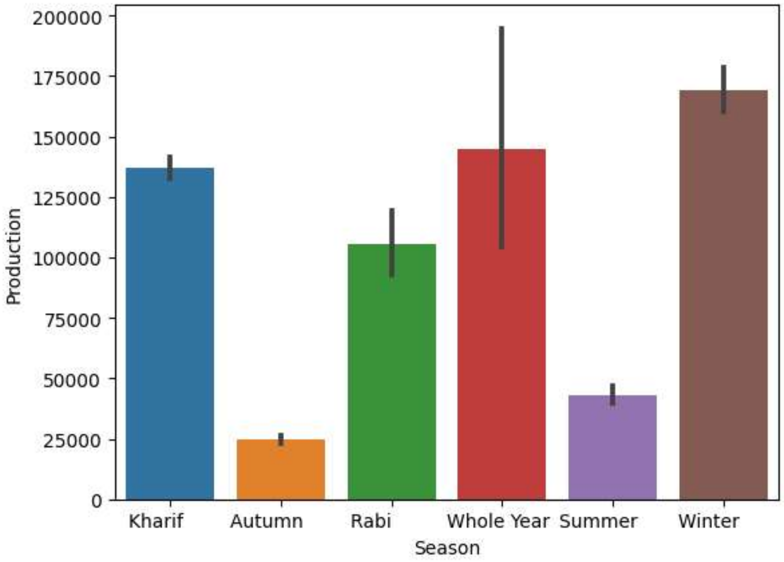
- High Frequency Crop in the dataset

```
rice_df = data_explore[data_explore["Crop"]=="Rice"]
print(rice_df.shape)
rice_df[:3]
```

```
(15082, 9)
```

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

```
sns.barplot(x ="Season",y = "Production",data=rice_df);
```



✓ 0s completed at 11:15 AM