

▼ CROP PRODUCTION - By Challa Vijaya Sai Pavan Santhosh

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

```
#some additional libraries
import missingno as msno
import plotly.graph_objects as go
import plotly.express as px
```

```
#READING DATA FROM APPS
df = pd.read_csv('/content/Crop Production data.csv')
```

df

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Productio
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.
...

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
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

df.shape

(246091, 7)

```
#Checking Column Names in the Dataset
df.columns
```

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

df['District_Name'].unique()

```
array(['NICOBARS', 'NORTH AND MIDDLE ANDAMAN', 'SOUTH ANDAMANS',
       'ANANTAPUR', 'CHITTOOR', 'EAST GODAVARI', 'GUNTUR', 'KADAPA',
       'KRISHNA', 'KURNOOL', 'PRAKASAM', 'SPSR NELLORE', 'SRIKAKULAM',
       'VISAKHAPATANAM', 'VIZIANAGARAM', 'WEST GODAVARI', 'ANJAW',
       'CHANGLANG', 'DIBANG VALLEY', 'EAST KAMENG', 'EAST SIANG',
       'KURUNG KUMEY', 'LOHIT', 'LONGDING', 'LOWER DIBANG VALLEY',
       'LOWER SUBANSIRI', 'NAMSAI', 'PAPUM PARE', 'TAWANG', 'TIRAP',
       'UPPER SIANG', 'UPPER SUBANSIRI', 'WEST KAMENG', 'WEST SIANG',
       'BAKSA', 'BARPETA', 'BONGAIGAON', 'CACHAR', 'CHIRANG', 'DARRANG',
       'DHEMAJI', 'DHUBRI', 'DIBRUGARH', 'DIMA HASAO', 'GOALPARA',
```

```
'GOLAGHAT', 'HAILAKANDI', 'JORHAT', 'KAMRUP', 'KAMRUP METRO',
'KARBI ANGLONG', 'KARIMGANJ', 'KOKRAJHAR', 'LAKHIMPUR', 'MARIGAON',
'NAGAON', 'NALBARI', 'SIVASAGAR', 'SONITPUR', 'TINSUKIA',
'UDALGURI', 'ARARIA', 'ARWAL', 'AURANGABAD', 'BANKA', 'BEGUSARAI',
'BHAGALPUR', 'BHOJPUR', 'BUXAR', 'DARBHANGA', 'GAYA', 'GOPALGANJ',
'JAMUI', 'JEHANABAD', 'KAIMUR (BHABUA)', 'KATI HAR', 'KHAGARIA',
'KISHANGANJ', 'LAKHISARAI', 'MADHEPURA', 'MADHUBANI', 'MUNGER',
'MUZAFFARPUR', 'NALANDA', 'NAWADA', 'PASHCHIM CHAMPARAN', 'PATNA',
'PURBI CHAMPARAN', 'PURNIA', 'ROHTAS', 'SAHARSA', 'SAMASTIPUR',
'SARAN', 'SHEIKHPURA', 'SHEOHAR', 'SITAMARHI', 'SIWAN', 'SUPAUL',
'VAISHALI', 'CHANDIGARH', 'BALOD', 'BALODA BAZAR', 'BALRAMPUR',
'BASTAR', 'BEMETARA', 'BIJAPUR', 'BILASPUR', 'DANTEWADA',
'DHARTARI', 'DURG', 'GARIYABAND', 'JANJGIR-CHAMPA', 'JASHPUR',
'KABIRDHAM', 'KANKER', 'KONDAGAON', 'KORBA', 'KOREA', 'MAHASAMUND',
'MUNGELI', 'NARAYANPUR', 'RAIGARH', 'RAIPUR', 'RAJNANDGAON',
'SUKMA', 'SURAJPUR', 'SURGUJA', 'DADRA AND NAGAR HAVELI',
'NORTH GOA', 'SOUTH GOA', 'AHMADABAD', 'AMRELI', 'ANAND',
'BANAS KANTHA', 'BHARUCH', 'BHAVNAGAR', 'DANG', 'DOHAD',
'GANDHINAGAR', 'JAMNAGAR', 'JUNAGADH', 'KACHCHH', 'KHEDA',
'MAHESANA', 'NARMADA', 'NAVSARI', 'PANCH MAHALS', 'PATAN',
'PORBANDAR', 'RAJKOT', 'SABAR KANTHA', 'SURAT', 'SURENDRANAGAR',
'TAPI', 'VADODARA', 'VALSAD', 'AMBALA', 'BHIWANI', 'FARIDABAD',
'FATEHABAD', 'GURGAON', 'HISAR', 'JHAJJAR', 'JIND', 'KAITHAL',
'KARNAL', 'KURUKSHETRA', 'MAHENDRAGARH', 'MEWAT', 'PALWAL',
'PANCHKULA', 'PANIPAT', 'REWARI', 'ROHTAK', 'SIRSA', 'SONIPAT',
'YAMUNANAGAR', 'CHAMBA', 'HAMIRPUR', 'KANGRA', 'KINNAUR', 'KULLU',
'LAHUL AND SPITI', 'MANDI', 'SHIMLA', 'SIRMAUR', 'SOLAN', 'UNA',
'ANANTNAG', 'BADGAM', 'BANDIPORA', 'BARAMULLA', 'DODA',
'GANDERBAL', 'JAMMU', 'KARGIL', 'KATHUA', 'KISHTWAR', 'KULGAM',
'KUPWARA', 'LEH LADAKH', 'POONCH', 'PULWAMA', 'RAJOURI', 'RAMBAN',
'REASI', 'SAMBA', 'SHOPIAN', 'SRINAGAR', 'UDHAMPUR', 'BOKARO',
'CHATRA', 'DEOGHAR', 'DHANBAD', 'DUMKA', 'EAST SINGHBUM', 'GARHWA',
'GIRIDIH', 'GODDA', 'GUMLA', 'HAZARIBAGH', 'JAMTARA', 'KHUNTI',
'KODERMA', 'LATEHAR', 'LOHARDAGA', 'PAKUR', 'PALAMU', 'RAMGARH',
'RANCHI', 'SAHEBGANJ', 'SARAIKELA KHARSWAN', 'SIMDEGA',
'WEST SINGHBUM', 'BAGALKOT', 'BANGALORE RURAL', 'BELGAUM',
'BELLARY', 'BENGALURU URBAN', 'BIDAR', 'CHAMARAJANAGAR',
'CHIKBALLAPUR', 'CHIKMAGALUR', 'CHITRADURGA', 'DAKSHIN KANNAD',
'DAVANGERE', 'DHARWAD', 'GADAG', 'GULBARGA', 'HASSAN', 'HAVERI',
'KODAGU', 'KOLAR', 'KOPPAL', 'MANDYA', 'MYSORE', 'RAICHUR',
'RAMANAGARA', 'SHIMOGA', 'TUMKUR', 'UDUPI', 'UTTAR KANNAD',
'YADGIR', 'ALAPPUZHA', 'ERNAKULAM', 'IDUKKI', 'KANNUR',
'KASARAGOD', 'KOLLAM', 'KOTTAYAM', 'KOZHIKODE', 'MALAPPURAM',
'PALAKKAD', 'PATHANAMTHITTA', 'THIRUVANANTHAPURAM', 'THRISSUR',
'WAYANAD', 'AGAR MALWA', 'ALIRAJPUR', 'ANUPPUR', 'ASHOKNAGAR',
'BALAGHAT', 'BARWANI', 'BETUL', 'BHIND', 'BHOPAL', 'BURHANPUR',
'CHHATARPUR', 'CHHINDWARA', 'DAMOH', 'DATIA', 'DEWAS', 'DHAR',
'DINDORI', 'GUNA', 'GWALIOR', 'HARDA', 'HOSHANGABAD', 'INDORE',
```

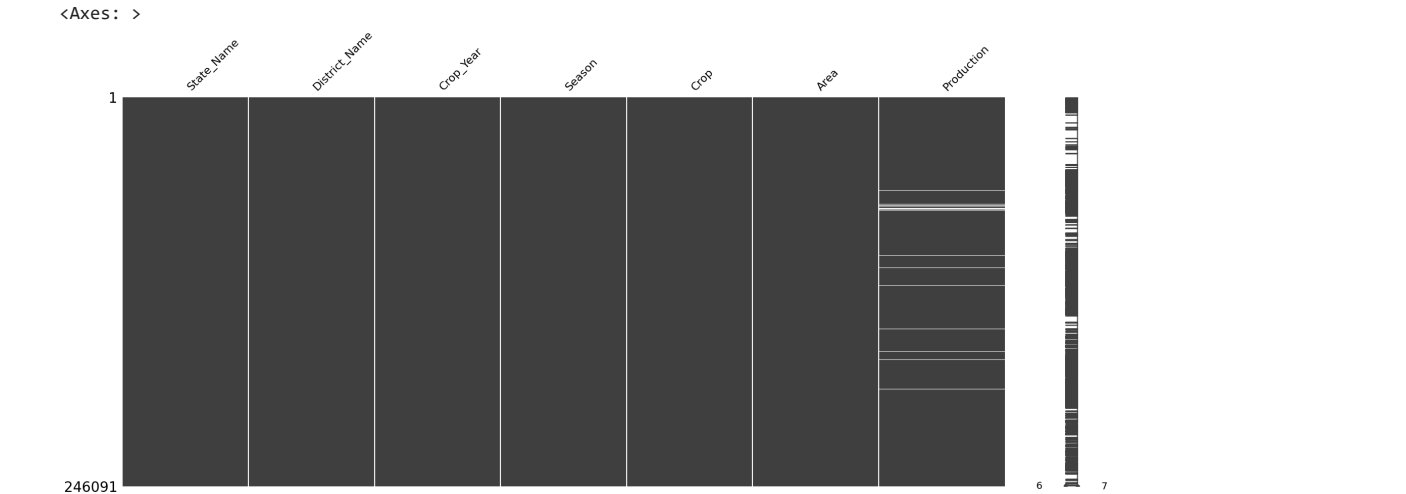
```
df['District_Name'].nunique()
```

646

```
df.info()
```

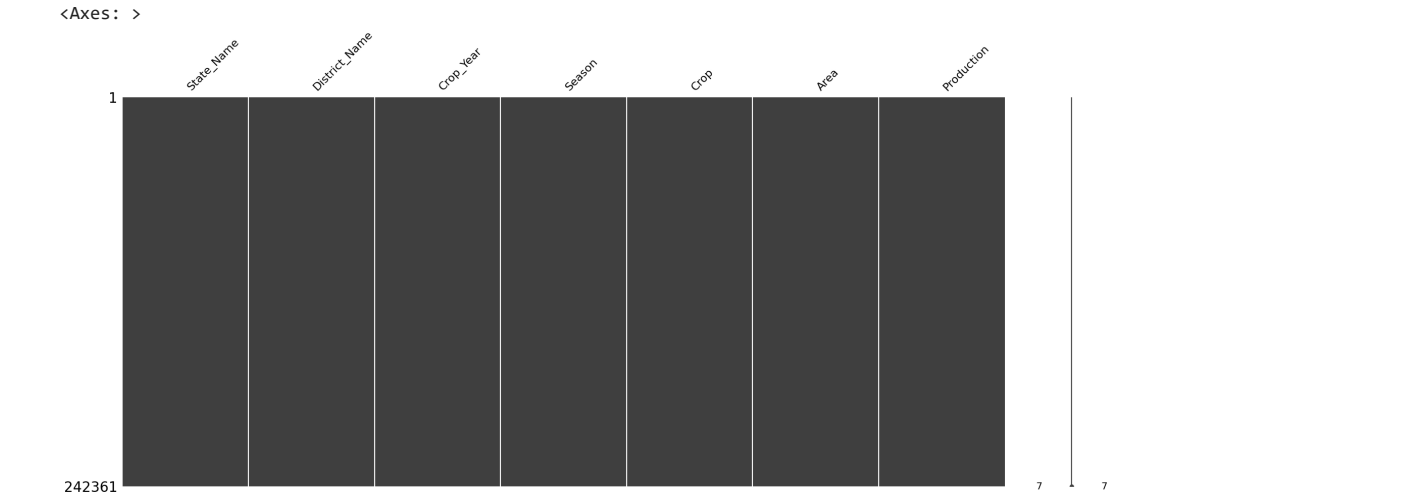
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246091 entries, 0 to 246090
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   State_Name  246091 non-null object
1   District_Name  246091 non-null object
2   Crop_Year    246091 non-null int64
3   Season       246091 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
```

```
msno.matrix(df)
```



```
#dropping nans, null entries in the data
df =df.dropna()
```

```
#no white lines
msno.matrix(df)
```



```
#dropping of columns
df.drop(['State_Name','District_Name'], axis = 1, inplace=True)
```

<ipython-input-18-74b1dfbd6fc8>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
df.drop(['State_Name','District_Name'], axis = 1, inplace=True)

df

	Crop_Year	Season	Crop	Area	Production
0	2000	Kharif	Arecanut	1254.0	2000.0
1	2000	Kharif	Other Kharif pulses	2.0	1.0
2	2000	Kharif	Rice	102.0	321.0
3	2000	Whole Year	Banana	176.0	641.0
4	2000	Whole Year	Cashewnut	720.0	165.0
...
246086	2014	Summer	Rice	306.0	801.0
246087	2014	Summer	Sesamum	627.0	463.0
246088	2014	Whole Year	Sugarcane	324.0	16250.0

```
df.describe()  
#Basic Statistics
```

	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

```
#Checking null values  
df.isnull().sum()
```

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

```
columns = list(df)  
columns
```

```
['Crop_Year', 'Season', 'Crop', 'Area', 'Production']
```

```
(df[columns[1:]]==0).sum()
```

```
Season      0  
Crop        0  
Area        0  
Production  3523  
dtype: int64
```

```
#Replace statement  
df[columns[1:]] = df[columns[1:]].replace(0,np.nan)
```

```
<ipython-input-25-8e73b16d4fe3>:2: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus  
df[columns[1:]] = df[columns[1:]].replace(0,np.nan)
```

```
#before drop statement  
df.shape
```

```
(242361, 5)
```

```
df.dropna(inplace =True)
```

```
<ipython-input-27-bd0d564509cf>:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

df.shape

(238838, 5)

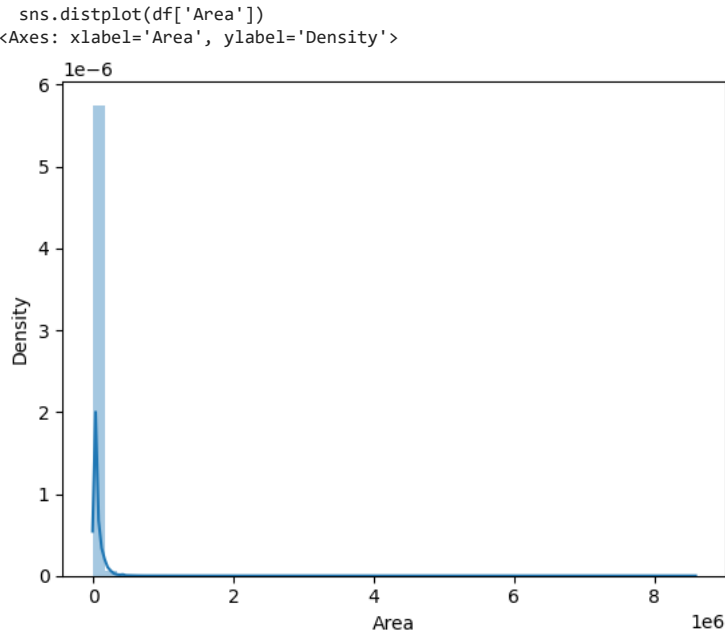
```
#Distribution Plot to Identify which technique is used
sns.distplot(df['Area'])
```

<ipython-input-29-33a5ff344958>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>



df[df.duplicated()]

	Crop_Year	Season	Crop	Area	Production	
	2024	2003	Kharif	Mesta	2.0	16.0
	2080	2004	Kharif	Mesta	2.0	17.0
	2166	2006	Kharif	Mesta	2.0	18.0
	2206	2007	Kharif	Jowar	1000.0	1000.0
	2555	1997	Kharif	Sesamum	400.0	100.0

	245786	2006	Rabi	Peas & beans (Pulses)	1.0	1.0
	245875	2008	Summer	Moong(Green Gram)	5.0	4.0
	245883	2008	Whole Year	Tobacco	7.0	2.0
	245943	2010	Rabi	Masoor	3.0	2.0
	245964	2011	Kharif	Mesta	3.0	3.0

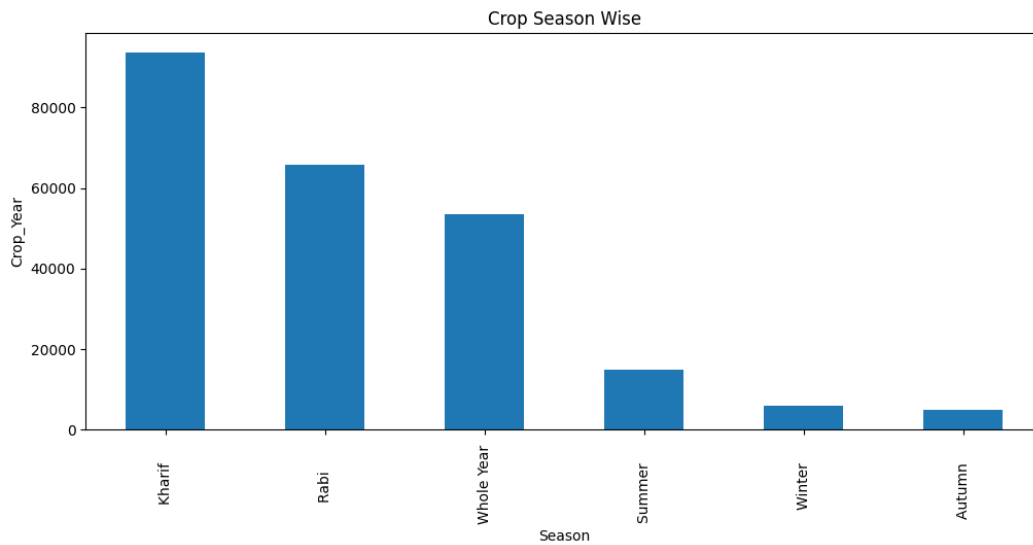
9273 rows × 5 columns

```
category_series = df['Season'].value_counts().head(10)
```

category_series

Kharif 93765
Rabi 65904
Name: Season, dtype: int64

```
#Plot Bar Graph for the no.of Apps in each Category
plt.figure(figsize=(12,5))
plt.title("Crop Season Wise")
plt.ylabel('Crop_Year')
plt.xlabel('Season')
plt.xticks(rotation=60,fontsize=10)
df['Season'].value_counts().head(10).plot(kind='bar')
plt.show()
```



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

```
#dropping of columns
df.drop(['State_Name','District_Name'], axis = 1, inplace=True)
```

```
df.head()
```

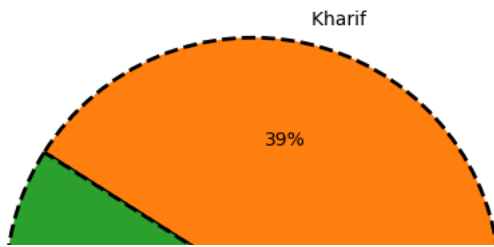
	Crop_Year	Season	Crop	Area	Production
0	2000	Kharif	Areca nut	1254.0	2000.0
1	2000	Kharif	Other Kharif pulses	2.0	1.0
2	2000	Kharif	Rice	102.0	321.0
3	2000	Whole Year	Banana	176.0	641.0
4	2000	Whole Year	Cashewnut	720.0	165.0

```
season_production_df=df.groupby('Season')[['Crop']].count()
```

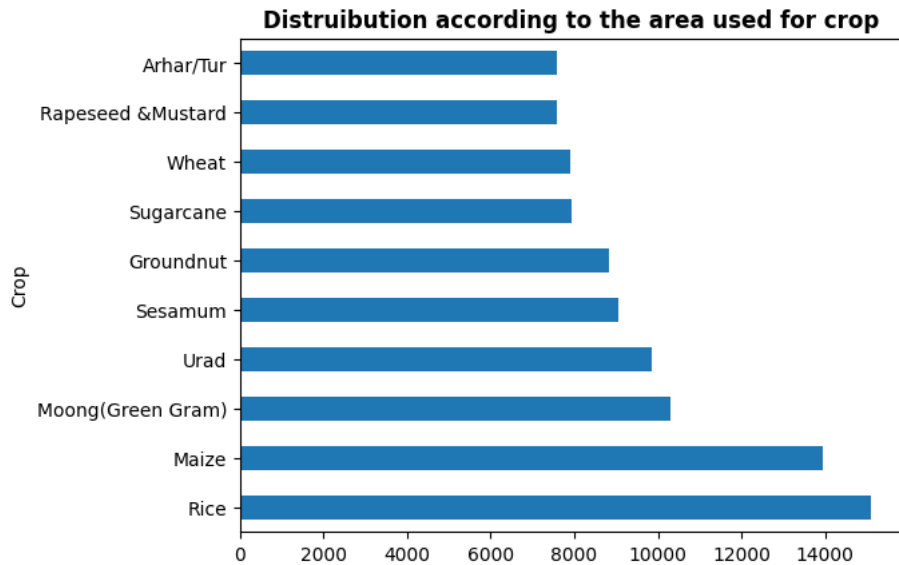
```
season_production_df.plot.pie(subplots=True, figsize=(12, 6), wedgeprops={"edgecolor":"0", 'linewidth': 2,
'linestyle': 'dashed', 'antialiased': True}, autopct='%1.0f%%')
plt.title('Distribution of Crop according to seasons ',fontweight=600)
```

```
Text(0.5, 1.0, 'Distribution of Crop according to seasons ')
```

Distribution of Crop according to seasons

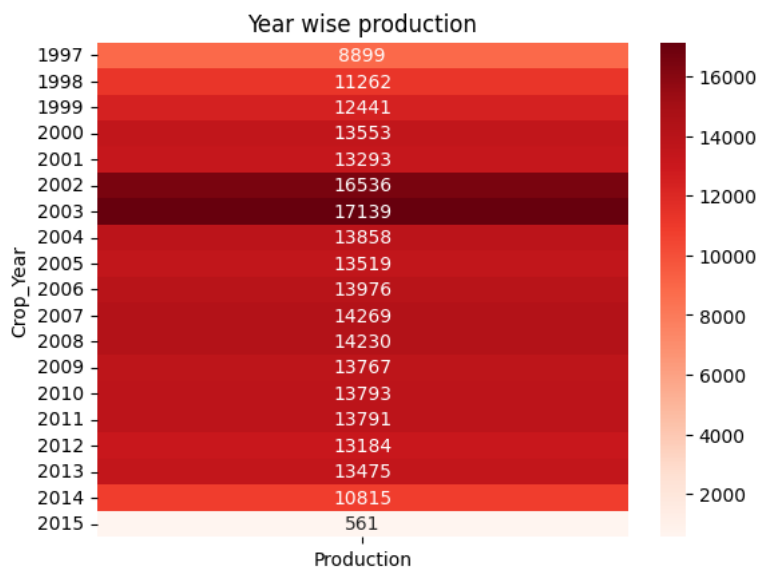


```
plt.title('Distruibution according to the area used for crop',fontweight=600)
plt.ylabel('Crop')
plt.xlabel('Area')
df['Crop'].value_counts().head(10).plot(kind='barh')
plt.show()
```



```
plt.title("Year wise production")
sns.heatmap(df.groupby('Crop_Year')[['Production']].count(),fmt="d", annot=True, cmap='Reds')
```

```
<Axes: title={'center': 'Year wise production'}, ylabel='Crop_Year'>
```



```
## Asking and Answering Questions
df.sort_values(by=['Production'],ascending=False).head(10)
```

	Crop_Year	Season	Crop	Area	Production
178273	2011	Whole Year	Coconut	82704.00	1.250800e+09
178331	2013	Whole Year	Coconut	84531.00	1.212000e+09
100654	2012	Whole Year	Coconut	102417.00	1.125000e+09
100612	2009	Whole Year	Coconut	108380.00	1.063000e+09
100149	1999	Whole Year	Coconut	131061.00	1.059000e+09
100391	2014	Whole Year	Coconut	123066.00	1.001000e+09
100135	1998	Whole Year	Coconut	129506.00	9.999999e+08

df.sort_values(by=['Crop_Year'],ascending=False).head(10)

	Crop_Year	Season	Crop	Area	Production
160379	2015	Summer	Sesamum	20.0	6.7
150898	2015	Summer	Wheat	8.0	9.1
150904	2015	Winter	Rapeseed &Mustard	1.0	0.2
150903	2015	Winter	Potato	170.0	2371.2
150902	2015	Winter	Moong(Green Gram)	235.0	83.9
150901	2015	Winter	Maize	2.0	1.9
150900	2015	Winter	Horse-gram	479.0	190.2
150899	2015	Winter	Groundnut	16.0	30.3
150897	2015	Summer	Sesamum	14.0	4.7
150906	2015	Winter	Sesamum	187.0	31.0

df[df.Production >= 100.0]

	Crop_Year	Season	Crop	Area	Production
0	2000	Kharif	Arecanut	1254.0	2000.0
2	2000	Kharif	Rice	102.0	321.0
3	2000	Whole Year	Banana	176.0	641.0
4	2000	Whole Year	Cashewnut	720.0	165.0
5	2000	Whole Year	Coconut	18168.0	65100000.0
...
246085	2014	Summer	Maize	325.0	2039.0
246086	2014	Summer	Rice	306.0	801.0
246087	2014	Summer	Sesamum	627.0	463.0
246088	2014	Whole Year	Sugarcane	324.0	16250.0
246089	2014	Winter	Rice	279151.0	597899.0

179095 rows × 5 columns

```
pip install squarify

Collecting squarify
  Downloading squarify-0.4.3-py3-none-any.whl (4.3 kB)
Installing collected packages: squarify
Successfully installed squarify-0.4.3
```

```
import matplotlib.pyplot as plt
import squarify
import pandas as pd

plt.figure(figsize = (9,4))

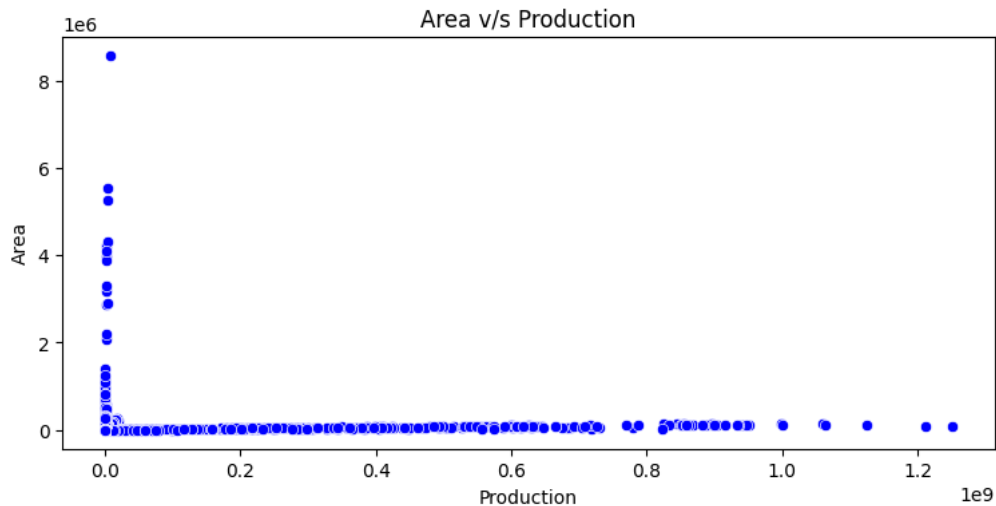
Production=df['Production']
Area=df['Area']

sns.scatterplot(x =Production, y = Area, color = 'blue',)

plt.title("Area v/s Production")
```



```
plt.xlabel('Production')
plt.ylabel('Area')
plt.show()
```

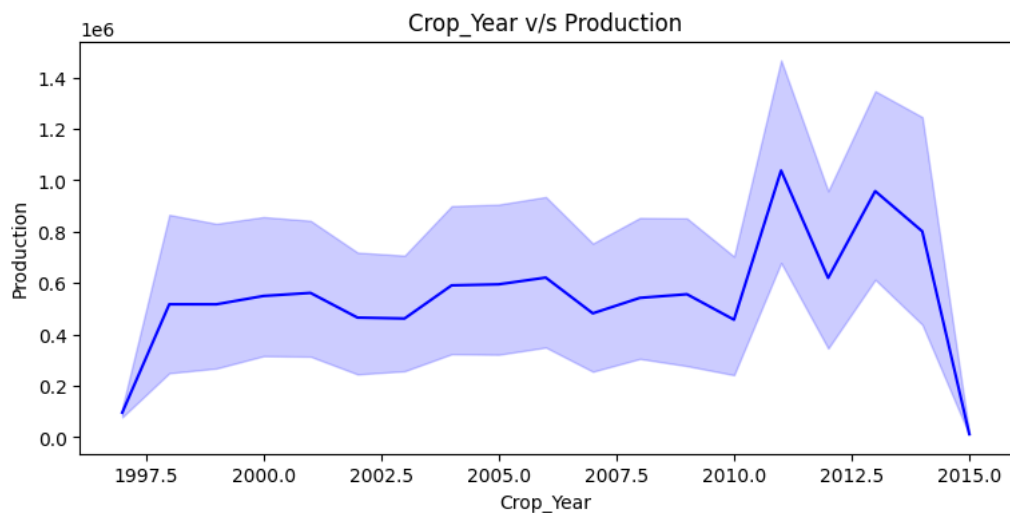


```
plt.figure(figsize = (9,4))

Rating=df['Crop_Year']
Reviews=df['Production']

sns.lineplot(x = Rating, y = Reviews, color = 'blue',)

plt.title("Crop_Year v/s Production")
plt.xlabel('Crop_Year')
plt.ylabel('Production')
plt.show()
```



```
df['Crop'].unique()
```

```
array(['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', 'Rapeseed &Mustard', 'Mesta',
       'Cowpea(Lobia)', 'Lemon', 'Pome Granet', 'Sapota', 'Cabbage',
       'Peas (vegetable)', 'Niger seed', 'Bottle Gourd', 'Sannhamp',
       'Varagu', 'Garlic', 'Ginger', 'Oilseeds total', 'Pulses total',
       'Jute', 'Peas & beans (Pulses)', 'Blackgram', 'Paddy', 'Pineapple',
       'Barley', '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',
```

```
'Pump Kin', 'Tea', 'Coffee', 'Cauliflower', 'Other Citrus Fruit',
'Water Melon', 'Total foodgrain', 'Kapas', 'Colocosia', 'Lentil',
'Bean', 'Jobster', 'Perilla', 'Rajmash Kholan',
'Ricebean (nagadal)', 'Ash Gourd', 'Beet Root', 'Lab-Lab',
'Ribed Guard', 'Yam', 'Apple', 'Peach', 'Pear', 'Plums', 'Litchi',
'Ber', 'Other Dry Fruit', 'Jute & mesta'], dtype=object)
```

```
data=df['Crop']
value=data.value_counts()
print(value)
```

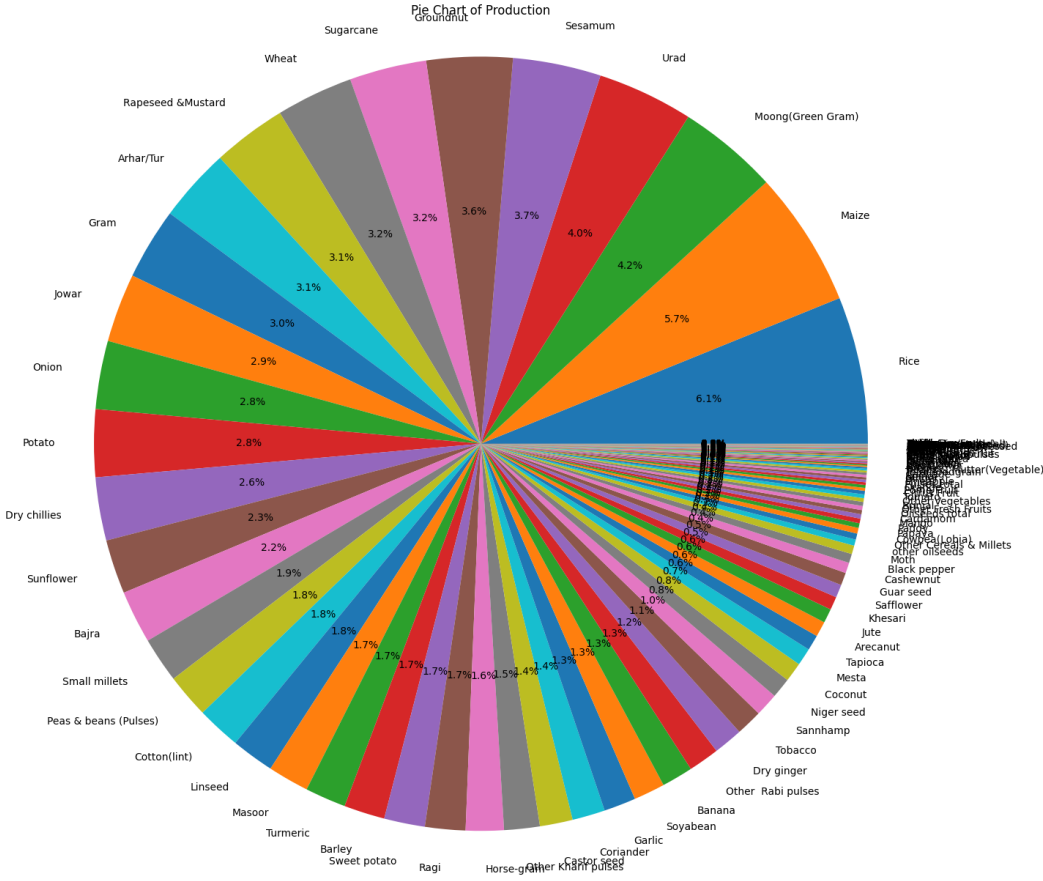
```
Rice          15104
Maize         13947
Moong(Green Gram) 10318
Urad          9850
Sesamum       9046
...
Litchi         6
Coffee         6
Apple          4
Peach          4
Other Dry Fruit 1
Name: Crop, Length: 124, dtype: int64
```

```
plt.figure(figsize = (15,9))
plt.bar(value.index, value.values)
plt.xlabel('Crop')
plt.ylabel('Count of Production')
plt.title('Crop Production Count')
plt.xticks(rotation=90)
plt.show()
```

Crop Production Count

```
plt.figure(figsize=(15,15)) # Optional: Set the figure size
plt.pie(value.values, labels=value.index,autopct='%1.1f%%')
plt.title('Pie Chart of Production')
plt.xticks(rotation=90)
plt.axis('equal')
```

(-1.0999999999999865,
1.0999999999999994,
-1.09999999999936434,
1.0999999999877552)

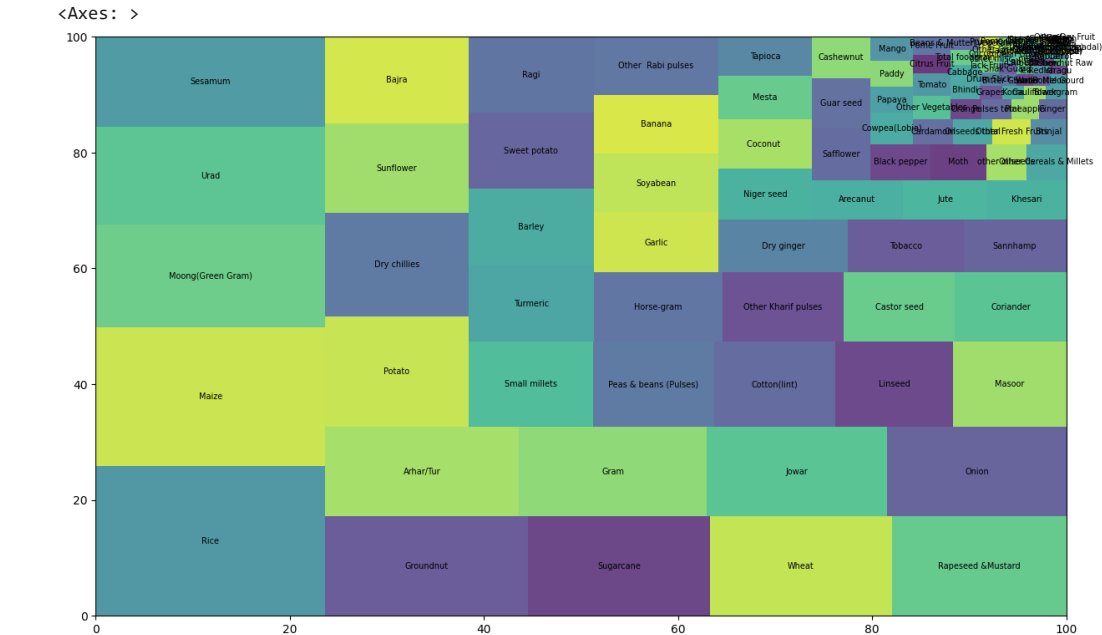


```
df.corr()
```

```
<ipython-input-83-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated.
df.corr()

   Crop_Year  Area  Production
Crop_Year    1.000000 -0.026022  0.006989
Area          0.000000  1.000000  0.010507
Production    0.000000  0.000000  1.000000

plt.figure(figsize=(15, 9))
text_kwargs = {'fontsize': 7, 'fontweight': 'ultralight', 'color': 'black'}
squarify.plot(label=value.index, sizes=value.values,alpha=0.8, text_kwargs=text_kwargs)
```



```
plt.figure(figsize=(21, 20))
sns.displot(df['Crop'], kde=True, bins=50)
plt.xticks(rotation=90)
```

```
([0,  
1,  
2,  
3,  
4,  
5,  
6,  
7,  
8,  
9,  
10,  
11,  
12,  
13,  
14,  
15,  
16,  
17,  
18,  
19,  
20,  
21,  
22,  
23,  
24,  
25,  
26,  
27,  
28,  
29,  
30,  
31,  
32,  
33,  
34,  
35,  
36,  
37,  
38,  
39,  
40,  
41,  
42,  
43,  
44,  
45,  
46,  
47,  
48,  
49,  
50,  
51,  
52,  
53,  
54,  
55,  
56,  
57,  
58,  
59,  
60,  
61,  
62,  
63,  
64,  
65,  
66,  
67,  
68,  
69,  
70,  
71,  
72,  
73,  
74,  
75,  
76,  
77,  
78,  
79,  
80,  
81,  
82,  
83,  
84,  
85,  
86,  
87,  
88,  
89,
```

```

90,
91,
92,
93,
94,
95,
96,
97,
98,
99,
100,
101,
102,
103,
104,
105,
106,
107,
108,
109,
110,
111,
112,
113,
114,
115,
116,
117,
118,
119,
120,
121,
122,
123],
[Text(0, 0, 'Areca nut'),
Text(1, 0, 'Other Kharif pulses'),
Text(2, 0, 'Rice'),
Text(3, 0, 'Banana'),
Text(4, 0, 'Cashewnut'),
Text(5, 0, 'Coconut '),
Text(6, 0, 'Dry ginger'),
Text(7, 0, 'Sugarcane'),
Text(8, 0, 'Sweet potato'),
Text(9, 0, 'Tapioca'),
Text(10, 0, 'Black pepper'),
Text(11, 0, 'Dry chillies'),
Text(12, 0, 'other oilseeds'),
Text(13, 0, 'Turmeric'),
Text(14, 0, 'Maize'),
Text(15, 0, 'Moong(Green Gram)'),
Text(16, 0, 'Urad'),
Text(17, 0, 'Arhar/Tur'),
Text(18, 0, 'Groundnut'),
Text(19, 0, 'Sunflower'),
Text(20, 0, 'Bajra'),
Text(21, 0, 'Castor seed'),
Text(22, 0, 'Cotton(lint)'),
Text(23, 0, 'Horse-gram'),
Text(24, 0, 'Jowar'),
Text(25, 0, 'Korra'),
Text(26, 0, 'Ragi'),
Text(27, 0, 'Tobacco'),
Text(28, 0, 'Gram'),
Text(29, 0, 'Wheat'),
Text(30, 0, 'Masoor'),
Text(31, 0, 'Sesamum'),
Text(32, 0, 'Linseed'),
Text(33, 0, 'Safflower'),
Text(34, 0, 'Onion'),
Text(35, 0, 'other misc. pulses'),
Text(36, 0, 'Samai'),
Text(37, 0, 'Small millets'),
Text(38, 0, 'Coriander'),
Text(39, 0, 'Potato'),
Text(40, 0, 'Other Rabi pulses'),
Text(41, 0, 'Soyabean'),
Text(42, 0, 'Beans & Mutter(Vegetable)'),
Text(43, 0, 'Bhindi'),
Text(44, 0, 'Brinjal'),
Text(45, 0, 'Citrus Fruit'),
Text(46, 0, 'Cucumber'),
Text(47, 0, 'Grapes'),
Text(48, 0, 'Mango'),
Text(49, 0, 'Orange'),
Text(50, 0, 'other fibres'),
Text(51, 0, 'Other Fresh Fruits'),
Text(52, 0, 'Other Vegetables'),
Text(53, 0, 'Papaya'),
Text(54, 0, 'Pome Fruit'),
Text(55, 0, 'Tomato'),

```

```

Text(56, 0, 'Rapeseed &Mustard'),
Text(57, 0, 'Mesta'),
Text(58, 0, 'Cowpea(Lobia)'),
Text(59, 0, 'Lemon'),
Text(60, 0, 'Pome Granet'),
Text(61, 0, 'Sapota'),
Text(62, 0, 'Cabbage'),
Text(63, 0, 'Peas (vegetable)'),
Text(64, 0, 'Niger seed'),
Text(65, 0, 'Bottle Gourd'),
Text(66, 0, 'Sannhamp'),
Text(67, 0, 'Varagu'),
Text(68, 0, 'Garlic'),
Text(69, 0, 'Ginger'),
Text(70, 0, 'Oilseeds total'),
Text(71, 0, 'Pulses total'),
Text(72, 0, 'Jute'),
Text(73, 0, 'Peas & beans (Pulses)'),
Text(74, 0, 'Blackgram'),
Text(75, 0, 'Paddy'),
Text(76, 0, 'Pineapple'),
Text(77, 0, 'Barley'),
Text(78, 0, 'Khesari'),
Text(79, 0, 'Guar seed'),
Text(80, 0, 'Moth'),
Text(81, 0, 'Other Cereals & Millets'),
Text(82, 0, 'Cond-spcs other'),
Text(83, 0, 'Turnip'),
Text(84, 0, 'Carrot'),
Text(85, 0, 'Redish'),
Text(86, 0, 'Arcanut (Processed)'),
Text(87, 0, 'Atcanut (Raw)'),
Text(88, 0, 'Cashewnut Processed'),
Text(89, 0, 'Cashewnut Raw'),
Text(90, 0, 'Cardamom'),
Text(91, 0, 'Rubber'),
Text(92, 0, 'Bitter Gourd'),
Text(93, 0, 'Drum Stick'),
Text(94, 0, 'Jack Fruit'),
Text(95, 0, 'Snak Guard'),
Text(96, 0, 'Pump Kin'),
Text(97, 0, 'Tea'),
Text(98, 0, 'Coffee'),
Text(99, 0, 'Cauliflower'),
Text(100, 0, 'Other Citrus Fruit'),
Text(101, 0, 'Water Melon'),
Text(102, 0, 'Total foodgrain'),
Text(103, 0, 'Kapas'),
Text(104, 0, 'Colocosia'),
Text(105, 0, 'Lentil'),
Text(106, 0, 'Bean'),
Text(107, 0, 'Jobster'),
Text(108, 0, 'Perilla'),
Text(109, 0, 'Rajmash Kholar'),
Text(110, 0, 'Ricebean (nagadal)'),
Text(111, 0, 'Ash Gourd'),
Text(112, 0, 'Beet Root'),
Text(113, 0, 'Lab-Lab'),
Text(114, 0, 'Ribed Guard'),
Text(115, 0, 'Yam'),
Text(116, 0, 'Apple'),
Text(117, 0, 'Peach'),
Text(118, 0, 'Pear'),
Text(119, 0, 'Plums'),
Text(120, 0, 'Litchi'),
Text(121, 0, 'Ber'),
Text(122, 0, 'Other Dry Fruit'),
Text(123, 0, 'Jute & mesta'])

```

<Figure size 2100x2000 with 0 Axes>

