agriculture-2-project

November 18, 2024

1 Smart Agricultural Analysis

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
df=pd.read_csv('agriculture.csv')
```

[3]:
```
df.head()
```

[3]:

	Unnamed: 0	Crop_Type	Crop	N	P	K	pH	rainfall	temperature	\
0	0	kharif	cotton	120	40	20	5.46	654.34	29.266667	
1	1	kharif	horsegram	20	60	20	6.18	654.34	29.266667	
2	2	kharif	jowar	80	40	40	5.42	654.34	29.266667	
3	3	kharif	maize	80	40	20	5.62	654.34	29.266667	
4	4	kharif	moong	20	40	20	5.68	654.34	29.266667	

	Area_in_hectares	Production_in_tons	target
0	7300	9400	1.287671
1	3300	1000	0.303030
2	10100	10200	1.009901
3	2800	4900	1.750000
4	1300	500	0.384615

[4]:
```
df.tail()
```

[4]:

	Unnamed: 0	Crop_Type	Crop	N	P	K	pH	rainfall	\
9996	9996	summer	maize	80	40	20	5.40	34.81	
9997	9997	summer	moong	20	40	20	5.60	34.81	
9998	9998	whole year	onion	120	60	65	5.94	689.88	
9999	9999	whole year	potato	180	60	90	5.02	689.88	
10000	10000	kharif	maize	80	40	20	5.48	579.75	

```
       temperature  Area_in_hectares  Production_in_tons    target
9996     34.666667               152                 154   1.013158
9997     34.666667               488                 211   0.432377
9998     29.037273               752                9080  12.074468
9999     29.037273              7595              167455  22.048058
10000    34.010000             11247                3385   0.300969
```

[5]:
```
df.shape
```

[5]:
```
(10001, 12)
```

[6]:
```
df.describe()
```

[6]:
```
         Unnamed: 0             N             P             K            pH  \
count  10001.000000  10001.000000  10001.000000  10001.000000  10001.000000
mean    5000.000000     69.146585     41.527847     39.709029      5.645313
std     2887.184355     37.197031     13.998587     26.615039      0.487916
min        0.000000     10.000000     10.000000     20.000000      4.820000
25%     2500.000000     50.000000     40.000000     20.000000      5.360000
50%     5000.000000     80.000000     40.000000     30.000000      5.540000
75%     7500.000000     80.000000     50.000000     40.000000      5.900000
max    10000.000000    180.000000     75.000000    150.000000      7.000000

           rainfall   temperature  Area_in_hectares  Production_in_tons  \
count  10001.000000  10001.000000      10001.000000        1.000100e+04
mean     670.237772     26.752053      18956.858714        3.886877e+04
std      604.413140      5.078345      45938.016774        1.134659e+05
min        3.274569      1.180000          1.000000        1.000000e+00
25%      157.310000     23.106000        193.000000        2.000000e+02
50%      579.750000     27.333333       1638.000000        2.000000e+03
75%     1011.490000     29.566667      11435.000000        1.810000e+04
max     3041.400000     35.346667     726300.000000        1.823000e+06

            target
count  10001.000000
mean       4.128522
std       30.233001
min        0.000514
25%        0.571429
50%        1.180132
75%        2.393728
max     1494.000000
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10001 entries, 0 to 10000
```

```
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          10001 non-null  int64
 1   Crop_Type           10001 non-null  object
 2   Crop                10001 non-null  object
 3   N                   10001 non-null  int64
 4   P                   10001 non-null  int64
 5   K                   10001 non-null  int64
 6   pH                  10001 non-null  float64
 7   rainfall            10001 non-null  float64
 8   temperature         10001 non-null  float64
 9   Area_in_hectares    10001 non-null  int64
 10  Production_in_tons  10001 non-null  int64
 11  target              10001 non-null  float64
dtypes: float64(4), int64(6), object(2)
memory usage: 937.7+ KB
```

[8]:
```
df.isnull().sum()
```

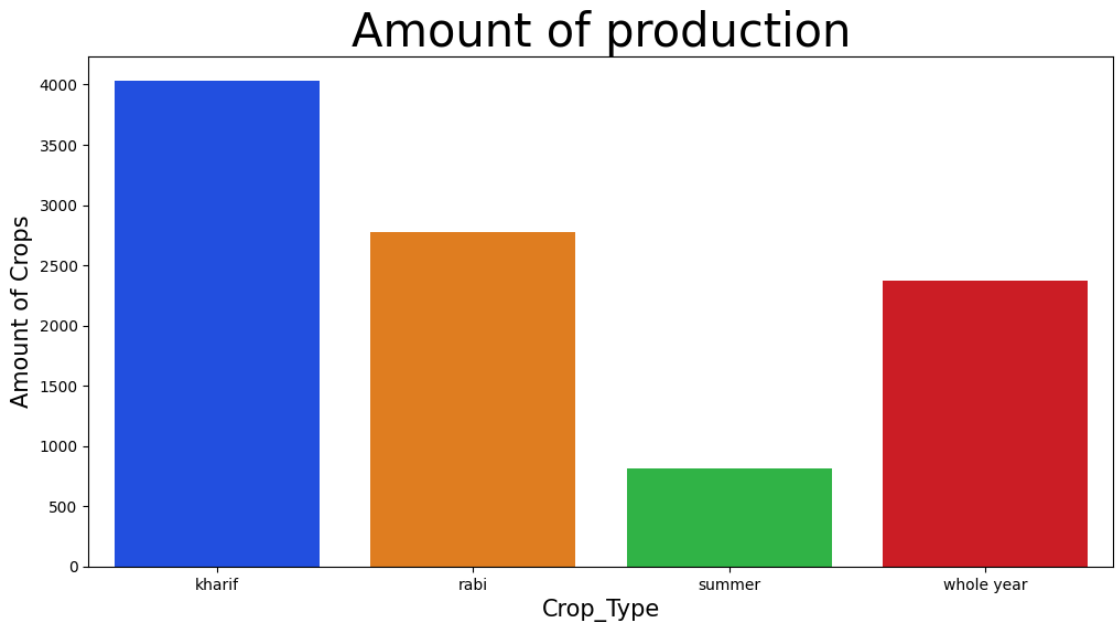
[8]:
```
Unnamed: 0            0
Crop_Type             0
Crop                  0
N                     0
P                     0
K                     0
pH                    0
rainfall              0
temperature           0
Area_in_hectares      0
Production_in_tons    0
target                0
dtype: int64
```

[9]:
```
df.duplicated().sum()
```

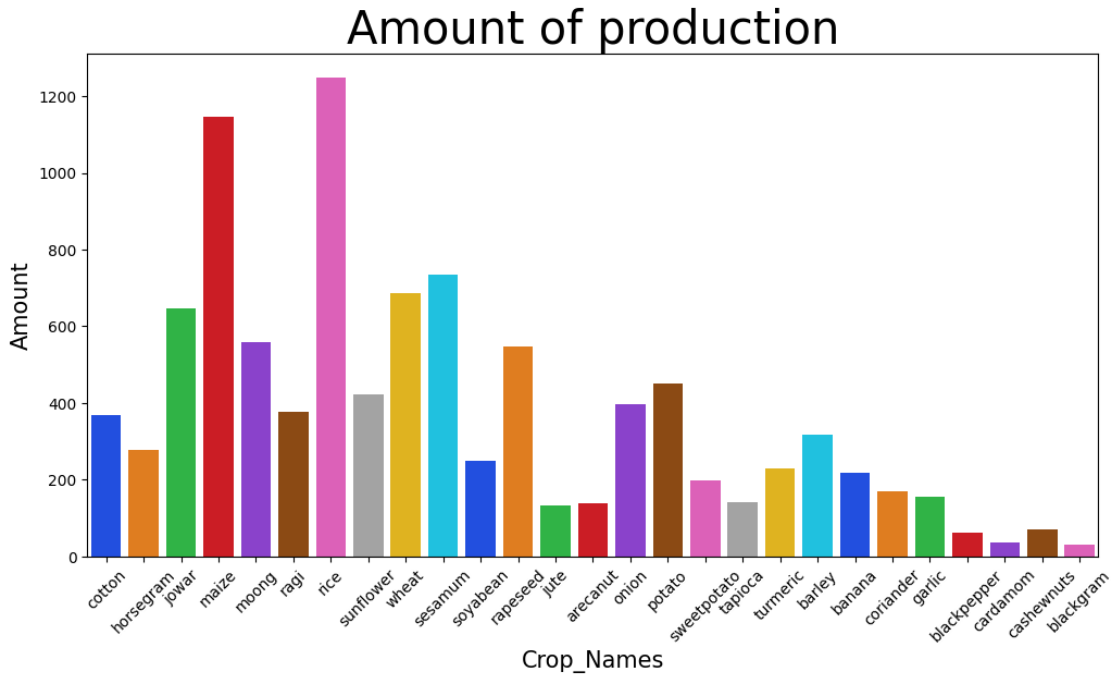
[9]:
```
np.int64(0)
```

1.1 visual representaion

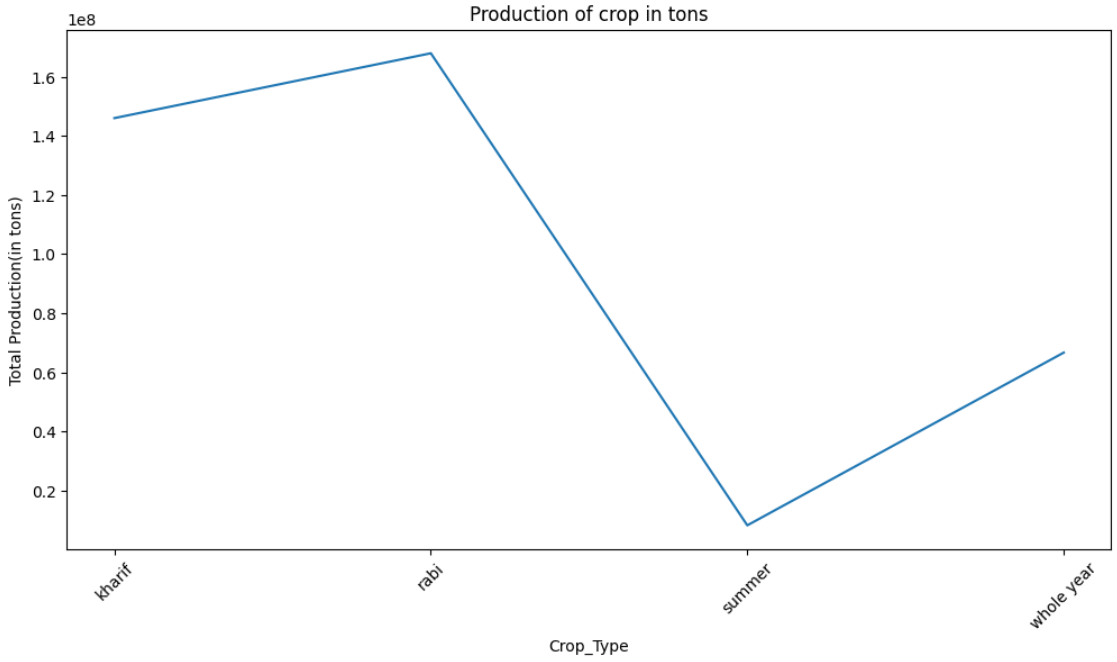
[10]:
```
plt.figure(figsize=(12,6))
sns.countplot(data=df, x='Crop_Type', hue="Crop_Type",palette='bright')
plt.title('Amount of production', fontsize=30)
plt.xlabel('Crop_Type', fontsize=15)
plt.ylabel('Amount of Crops', fontsize=15)
plt.show()
```



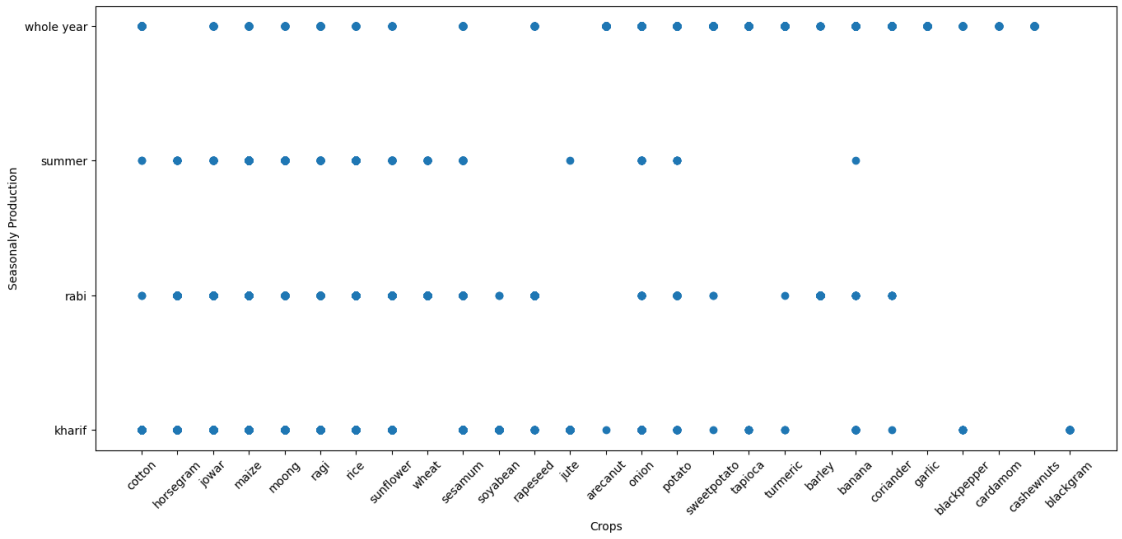
[11]:
```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Crop', hue="Crop",palette='bright')
plt.title('Amount of production', fontsize=30)
plt.xlabel('Crop_Names', fontsize=15)
plt.ylabel('Amount', fontsize=15)
plt.xticks(rotation=45)
plt.show()
```



[12]:
```
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='Crop_Type', y='Production_in_tons', estimator='sum',␣
 ↪ci=None)
plt.title('Production of crop in tons')
plt.xlabel('Crop_Type')
plt.ylabel('Total Production(in tons)')
plt.xticks(rotation=45)
plt.show()
```

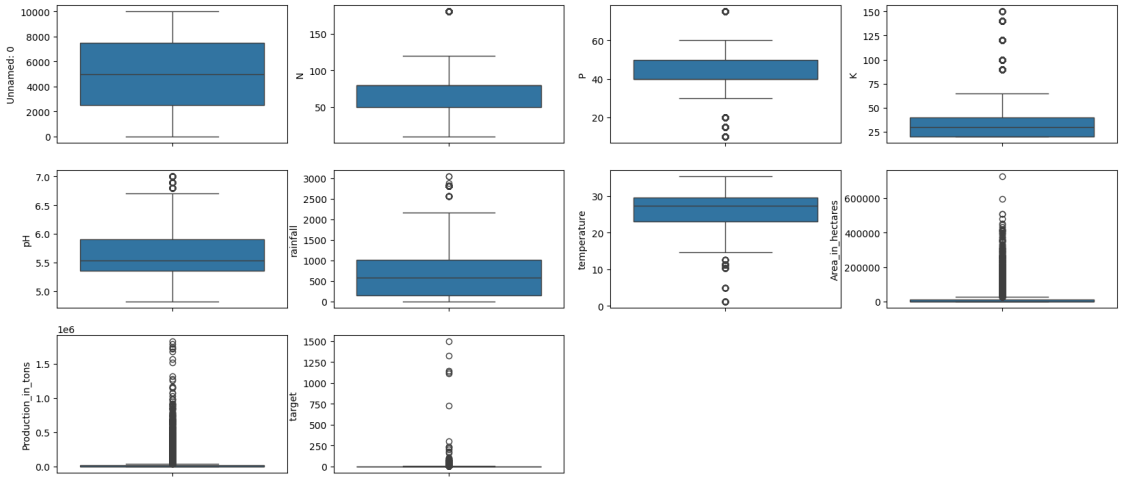


[13]:
```
plt.figure(figsize=(16,7))
plt.scatter(x=df['Crop'],y=df['Crop_Type'])
plt.xlabel('Crops')
plt.ylabel('Seasonaly Production')
plt.xticks(rotation=45)
plt.show()
```

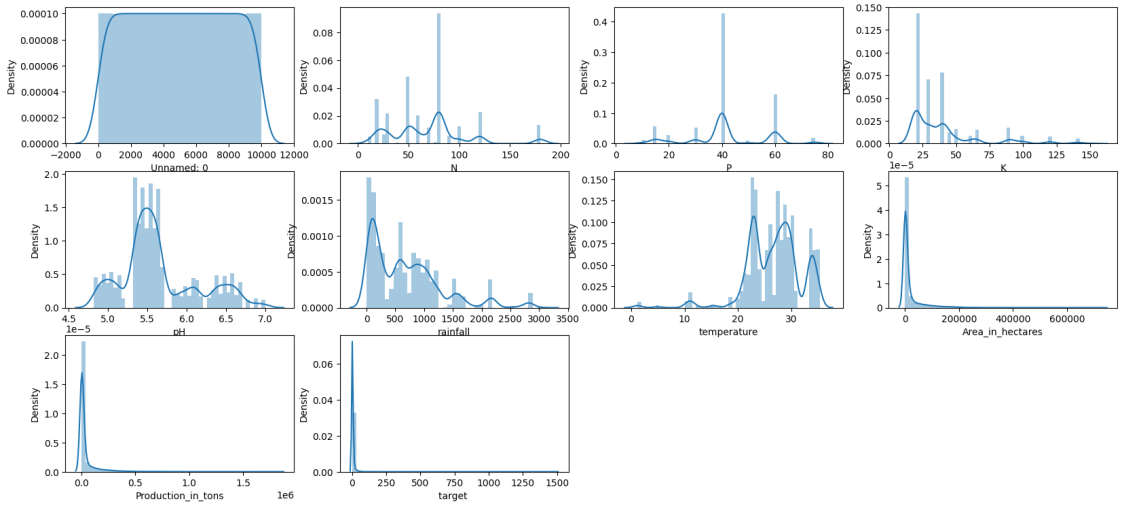


1.2 Univariate Analysis

[14]:
```
plt.figure(figsize=(20, 12))
for i, column in enumerate(df.select_dtypes(include='number'),1):
    plt.subplot(4,4,i)
    sns.boxplot(df[column])
```



[15]:
```
plt.figure(figsize=(20, 12))
for i, column in enumerate(df.select_dtypes(include='number'),1):
    plt.subplot(4,4,i)
    sns.distplot(df[column])
```



1.3 Skewness

[16]:
```
for i in df.select_dtypes(include='number'):
    skewness=df[i].skew()
    print("The skewness of",i,"is",skewness)
```

```
The skewness of Unnamed: 0 is 0.0
The skewness of N is 0.9547164852502035
The skewness of P is -0.01643855245891474
The skewness of K is 1.989466468450195
The skewness of pH is 0.7599319343153695
The skewness of rainfall is 1.268531509612503
The skewness of temperature is -1.0836069029134547
The skewness of Area_in_hectares is 4.463813201575672
The skewness of Production_in_tons is 6.311786054554366
The skewness of target  is 37.45537628503799
```

1.4 Correlation and heatmap

[17]:
```
df.describe(include='object')
```

[17]:

	Crop_Type	Crop
count	10001	10001
unique	4	27
top	kharif	rice
freq	4034	1250

[18]:
```
df=df.drop(["Crop_Type","Crop"],axis=1)
df.head()
```

[18]:

	Unnamed: 0	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	target
0	0	120	40	20	5.46	654.34	29.266667	7300	9400	1.287671
1	1	20	60	20	6.18	654.34	29.266667	3300	1000	0.303030
2	2	80	40	40	5.42	654.34	29.266667	10100	10200	1.009901
3	3	80	40	20	5.62	654.34	29.266667	2800	4900	1.750000
4	4	20	40	20	5.68	654.34	29.266667	1300	500	0.384615

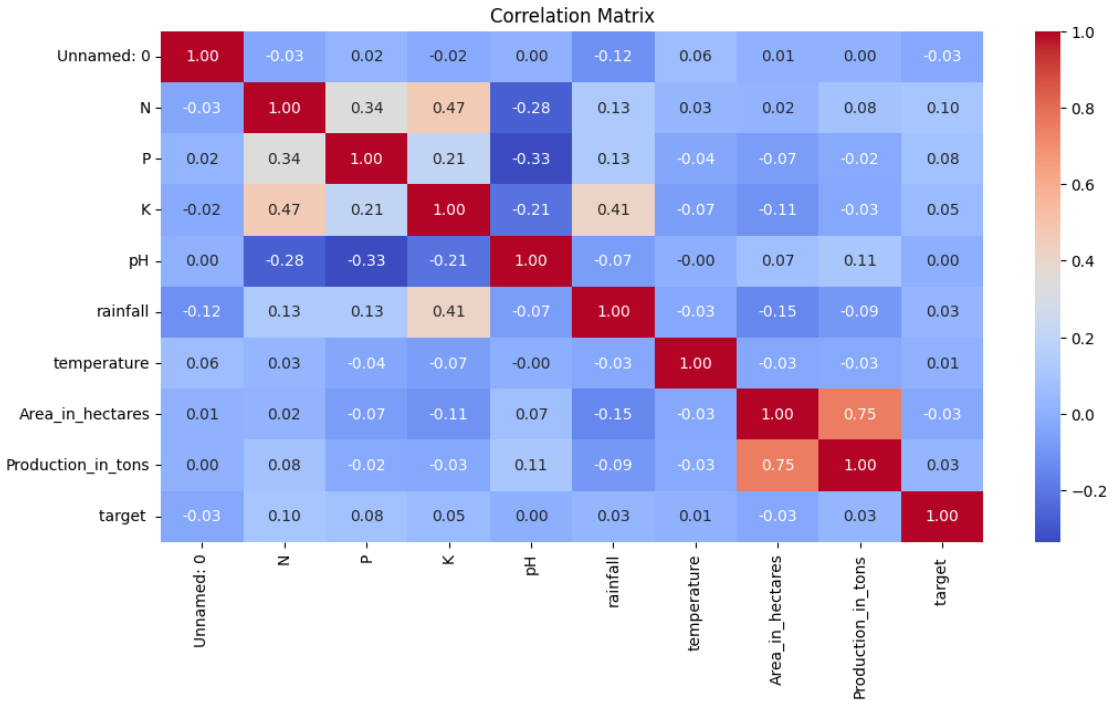
[19]:
```
df.corr()
```

[19]:

	Unnamed: 0	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	target
Unnamed: 0	1.000000	-0.033189	0.020015	-0.018716	0.004477	-0.117434	0.056472	0.006091	0.004372	-0.027724
N	-0.033189	1.000000	0.335317	0.467259	-0.277163	0.128159	0.028687	0.016556	0.082932	0.099843
P	0.020015	0.335317	1.000000	0.205663	-0.334898	0.126305	-0.037438	-0.069552	-0.022940	0.078102
K	-0.018716	0.467259	0.205663	1.000000	-0.211495	0.411469	-0.065351	-0.111029	-0.028182	0.049799
pH	0.004477	-0.277163	-0.334898	-0.211495	1.000000	-0.069599	-0.000626	0.070010	0.111721	0.002668
rainfall	-0.117434	0.128159	0.126305	0.411469	-0.069599	1.000000	-0.030709	-0.148148	-0.092841	0.027720
temperature	0.056472	0.028687	-0.037438	-0.065351	-0.000626	-0.030709	1.000000	-0.028585	-0.025893	0.007835
Area_in_hectares	0.006091	0.016556	-0.069552	-0.111029	0.070010	-0.148148	-0.028585	1.000000	0.753248	-0.028368
Production_in_tons	0.004372	0.082932	-0.022940	-0.028182	0.111721	-0.092841	-0.025893	0.753248	1.000000	-0.027724
target	-0.027724	0.099843	0.078102	0.049799	0.002668	0.027720	0.007835	-0.028368	-0.027724	1.000000

```
temperature                -0.025893   0.007835
Area_in_hectares            0.753248  -0.028368
Production_in_tons          1.000000   0.029977
target                      0.029977   1.000000
```

[20]:
```
plt.figure(figsize=(12, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



1.5 Checking for Outliers

[21]:
```
df.dtypes
```

[21]:
```
Unnamed: 0              int64
N                       int64
P                       int64
K                       int64
pH                    float64
rainfall              float64
temperature           float64
Area_in_hectares        int64
Production_in_tons      int64
target                float64
```

```
dtype: object
```

[22]:
```
df.duplicated().sum()
```

[22]:
```
np.int64(0)
```

[23]:
```
df.describe()
```

[23]:
```
         Unnamed: 0             N             P             K            pH  \
count  10001.000000  10001.000000  10001.000000  10001.000000  10001.000000
mean    5000.000000     69.146585     41.527847     39.709029      5.645313
std     2887.184355     37.197031     13.998587     26.615039      0.487916
min        0.000000     10.000000     10.000000     20.000000      4.820000
25%     2500.000000     50.000000     40.000000     20.000000      5.360000
50%     5000.000000     80.000000     40.000000     30.000000      5.540000
75%     7500.000000     80.000000     50.000000     40.000000      5.900000
max    10000.000000    180.000000     75.000000    150.000000      7.000000

           rainfall   temperature  Area_in_hectares  Production_in_tons  \
count  10001.000000  10001.000000      10001.000000        1.000100e+04
mean     670.237772     26.752053      18956.858714        3.886877e+04
std      604.413140      5.078345      45938.016774        1.134659e+05
min        3.274569      1.180000          1.000000        1.000000e+00
25%      157.310000     23.106000        193.000000        2.000000e+02
50%      579.750000     27.333333       1638.000000        2.000000e+03
75%     1011.490000     29.566667      11435.000000        1.810000e+04
max     3041.400000     35.346667     726300.000000        1.823000e+06

            target
count  10001.000000
mean       4.128522
std       30.233001
min        0.000514
25%        0.571429
50%        1.180132
75%        2.393728
max     1494.000000
```

1.6 Treatement of Outliers

[24]:
```
q1=df.quantile(0.25)
q1
```

[24]:
```
Unnamed: 0          2500.000000
N                     50.000000
P                     40.000000
K                     20.000000
```

```
pH                        5.360000
rainfall                157.310000
temperature              23.106000
Area_in_hectares        193.000000
Production_in_tons      200.000000
target                    0.571429
Name: 0.25, dtype: float64
```

[25]:
```
q3=df.quantile(0.75)
q3
```

[25]:
```
Unnamed: 0             7500.000000
N                        80.000000
P                        50.000000
K                        40.000000
pH                        5.900000
rainfall               1011.490000
temperature              29.566667
Area_in_hectares      11435.000000
Production_in_tons    18100.000000
target                    2.393728
Name: 0.75, dtype: float64
```

[26]:
```
IQR = q3-q1
IQR
```

[26]:
```
Unnamed: 0             5000.000000
N                        30.000000
P                        10.000000
K                        20.000000
pH                        0.540000
rainfall                854.180000
temperature               6.460667
Area_in_hectares      11242.000000
Production_in_tons    17900.000000
target                    1.822300
dtype: float64
```

[27]:
```
high_out_n=q3.N+1.5*IQR.N
high_out_n
```

[27]:
```
np.float64(125.0)
```

[28]:
```
low_out_n=q1.N-1.5*IQR.N
low_out_n
```

[28]:
```
np.float64(5.0)
```

[29]:
```
high_out_p=q3.P+1.5*IQR.P
high_out_p
```

[29]: np.float64(65.0)

[30]:
```
low_out_p=q1.P-1.5*IQR.P
low_out_p
```

[30]: np.float64(25.0)

[31]:
```
high_out_k=q3.K+1.5*IQR.K
high_out_k
```

[31]: np.float64(70.0)

[32]:
```
low_out_k=q1.K-1.5*IQR.K
low_out_k
```

[32]: np.float64(-10.0)

[33]:
```
high_out_ph=q3.pH+1.5*IQR.pH
high_out_ph
```

[33]: np.float64(6.710000000000001)

[34]:
```
low_out_ph=q1.pH-1.5*IQR.pH
low_out_ph
```

[34]: np.float64(4.550000000000001)

[35]:
```
high_out_rainfall=q3.rainfall+1.5*IQR.rainfall
high_out_rainfall
```

[35]: np.float64(2292.76)

[36]:
```
low_out_rainfall=q1.rainfall-1.5*IQR.rainfall
low_out_rainfall
```

[36]: np.float64(-1123.96)

[37]:
```
high_out_temp=q3.temperature+1.5*IQR.temperature
high_out_temp
```

[37]: np.float64(39.257666674999996)

[38]:
```
low_out_temp=q1.temperature-1.5*IQR.temperature
low_out_temp
```

[38]:
```
np.float64(13.414999995000004)
```

[39]:
```
high_out_Area=q3.Area_in_hectares+1.5*IQR.Area_in_hectares
high_out_Area
```

[39]:
```
np.float64(28298.0)
```

[40]:
```
low_out_Area=q1.Area_in_hectares-1.5*IQR.Area_in_hectares
low_out_Area
```

[40]:
```
np.float64(-16670.0)
```

[41]:
```
high_out_Production=q3.Production_in_tons+1.5*IQR.Production_in_tons
high_out_Production
```

[41]:
```
np.float64(44950.0)
```

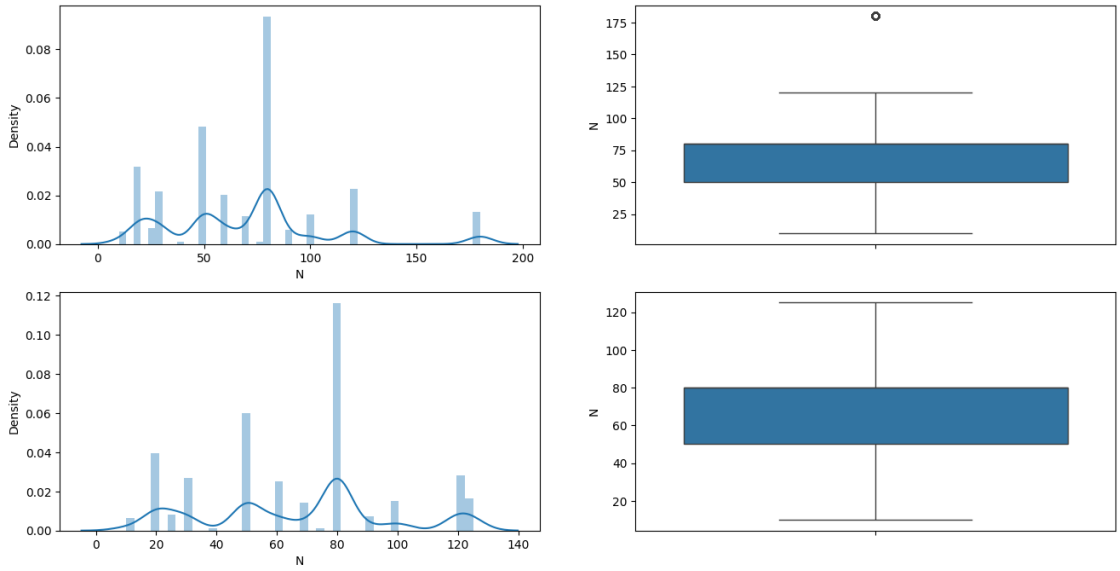
[42]:
```
low_out_Production=q1.Production_in_tons-1.5*IQR.Production_in_tons
low_out_Production
```

[42]:
```
np.float64(-26650.0)
```

1.7 Capping

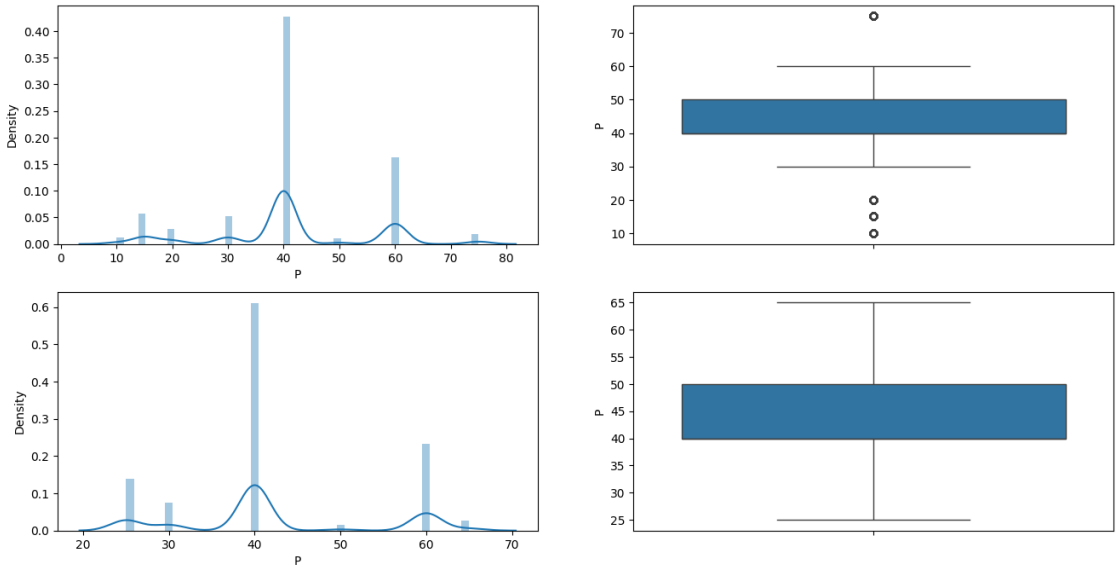
[43]:
```
new_df = df.copy()
new_df['N']=np.where(new_df['N']>high_out_n,
                     high_out_n,
                     np.where(new_df['N']<low_out_n,
                     low_out_n,
                     new_df['N']
    )
)
```

[44]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['N'])

plt.subplot(2,2,2)
sns.boxplot(df['N'])

plt.subplot(2,2,3)
sns.distplot(new_df['N'])

plt.subplot(2,2,4)
sns.boxplot(new_df['N'])

plt.show()
```



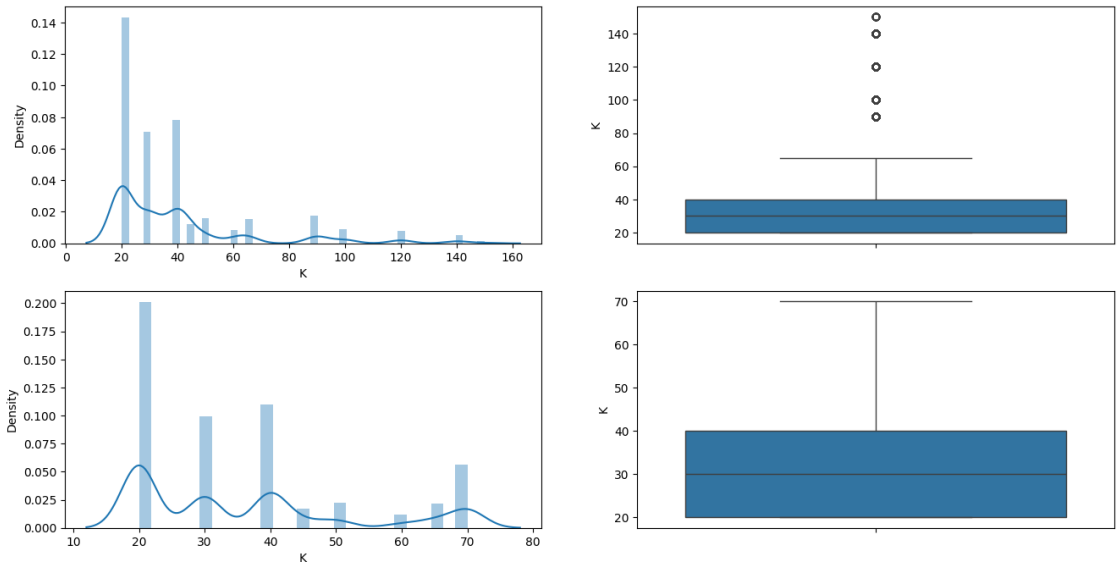
[45]:
```
new_df['P']=np.where(new_df['P']>high_out_p,
                     high_out_p,
                     np.where(new_df['P']<low_out_p,
                     low_out_p,
                     new_df['P']
    )
)
```

[46]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['P'])

plt.subplot(2,2,2)
sns.boxplot(df['P'])

plt.subplot(2,2,3)
sns.distplot(new_df['P'])

plt.subplot(2,2,4)
sns.boxplot(new_df['P'])

plt.show()
```



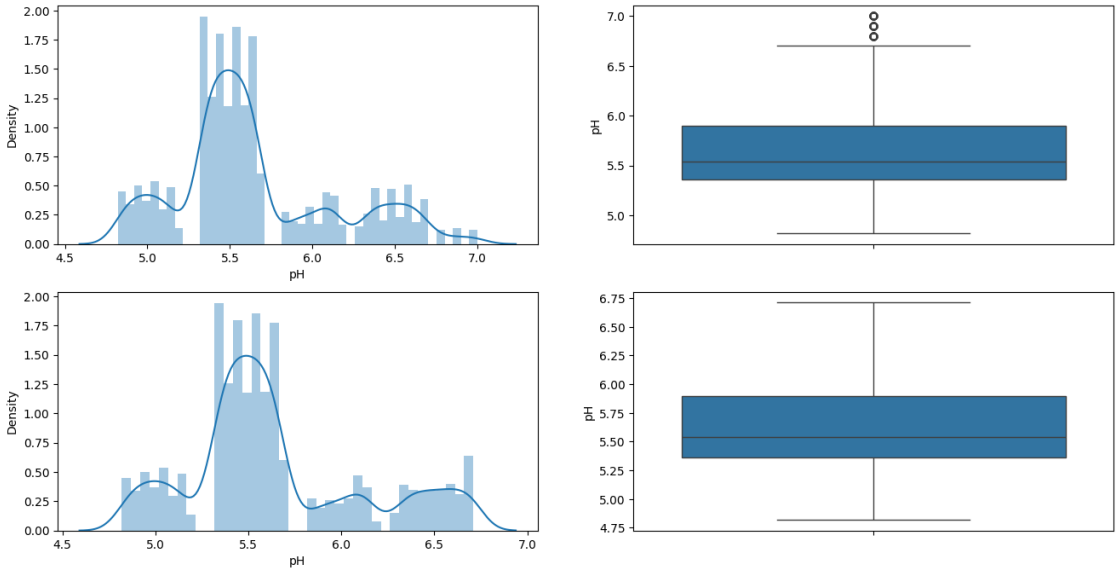
[47]:
```
new_df['K']=np.where(new_df['K']>high_out_k,
                     high_out_k,
                     np.where(new_df['K']<low_out_k,
                     low_out_k,
                     new_df['K']
    )
)
```

[48]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['K'])

plt.subplot(2,2,2)
sns.boxplot(df['K'])

plt.subplot(2,2,3)
sns.distplot(new_df['K'])

plt.subplot(2,2,4)
sns.boxplot(new_df['K'])

plt.show()
```



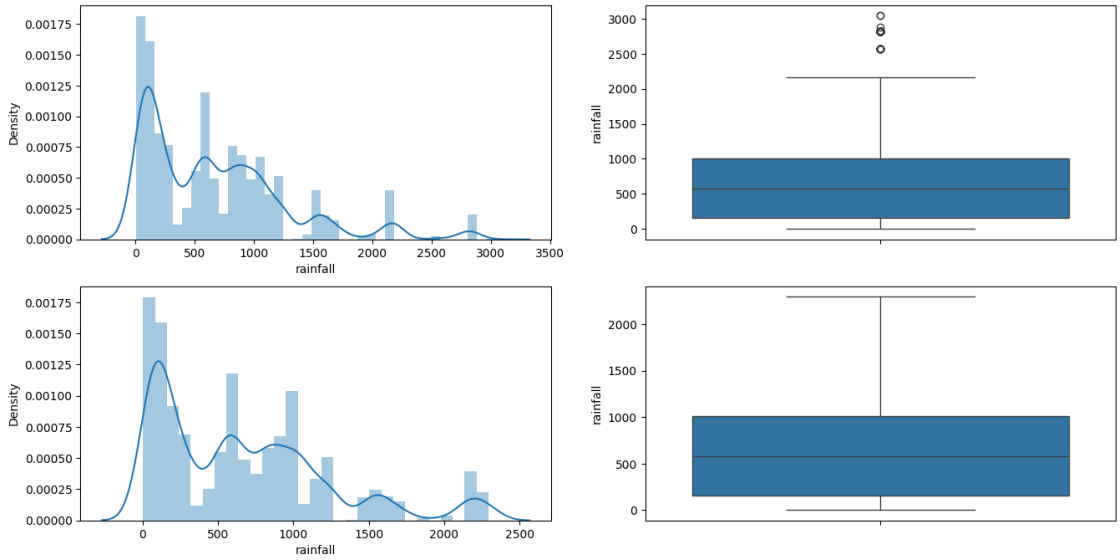
[49]:
```
new_df['pH']=np.where(new_df['pH']>high_out_ph,
                      high_out_ph,
                      np.where(new_df['pH']<low_out_ph,
                      low_out_ph,
                      new_df['pH']
    )
)
```

[50]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['pH'])

plt.subplot(2,2,2)
sns.boxplot(df['pH'])

plt.subplot(2,2,3)
sns.distplot(new_df['pH'])

plt.subplot(2,2,4)
sns.boxplot(new_df['pH'])

plt.show()
```



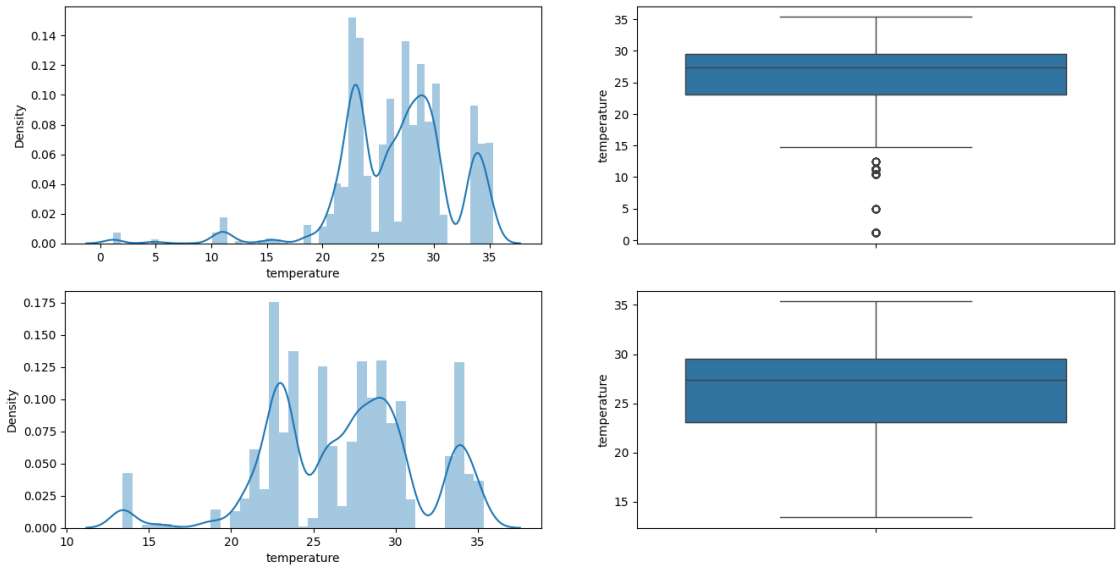
[51]:
```
new_df['rainfall']=np.where(new_df['rainfall']>high_out_rainfall,
                    high_out_rainfall,
                    np.where(new_df['rainfall']<low_out_rainfall,
                    low_out_rainfall,
                    new_df['rainfall']
    )
)
```

[52]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['rainfall'])

plt.subplot(2,2,2)
sns.boxplot(df['rainfall'])

plt.subplot(2,2,3)
sns.distplot(new_df['rainfall'])

plt.subplot(2,2,4)
sns.boxplot(new_df['rainfall'])

plt.show()
```



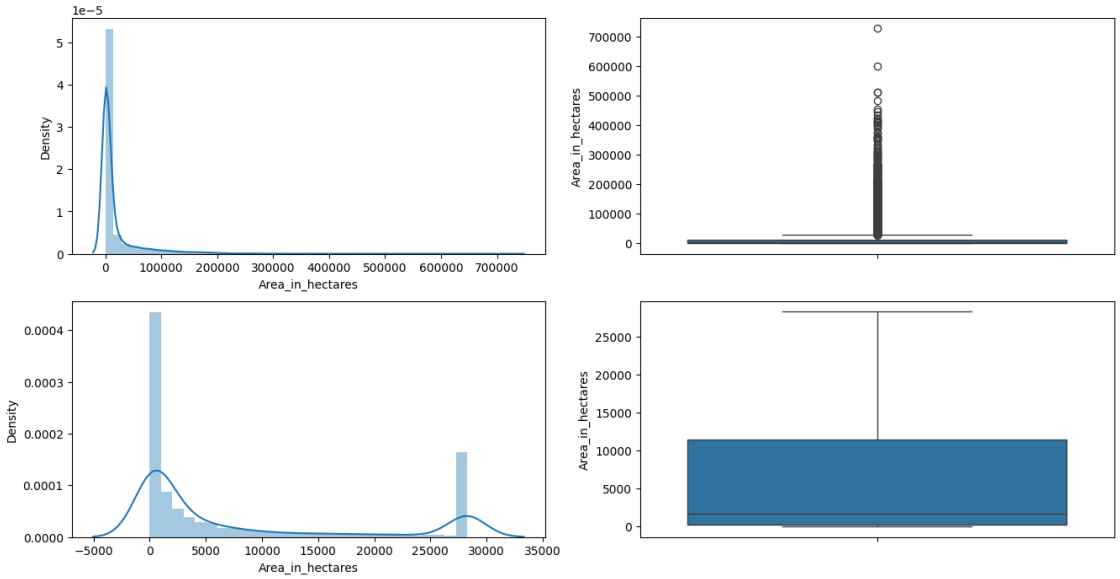
[53]:
```
new_df['temperature']=np.where(new_df['temperature']>high_out_temp,
                      high_out_temp,
                      np.where(new_df['temperature']<low_out_temp,
                      low_out_temp,
                      new_df['temperature']
    )
)
```

[54]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['temperature'])

plt.subplot(2,2,2)
sns.boxplot(df['temperature'])

plt.subplot(2,2,3)
sns.distplot(new_df['temperature'])

plt.subplot(2,2,4)
sns.boxplot(new_df['temperature'])

plt.show()
```



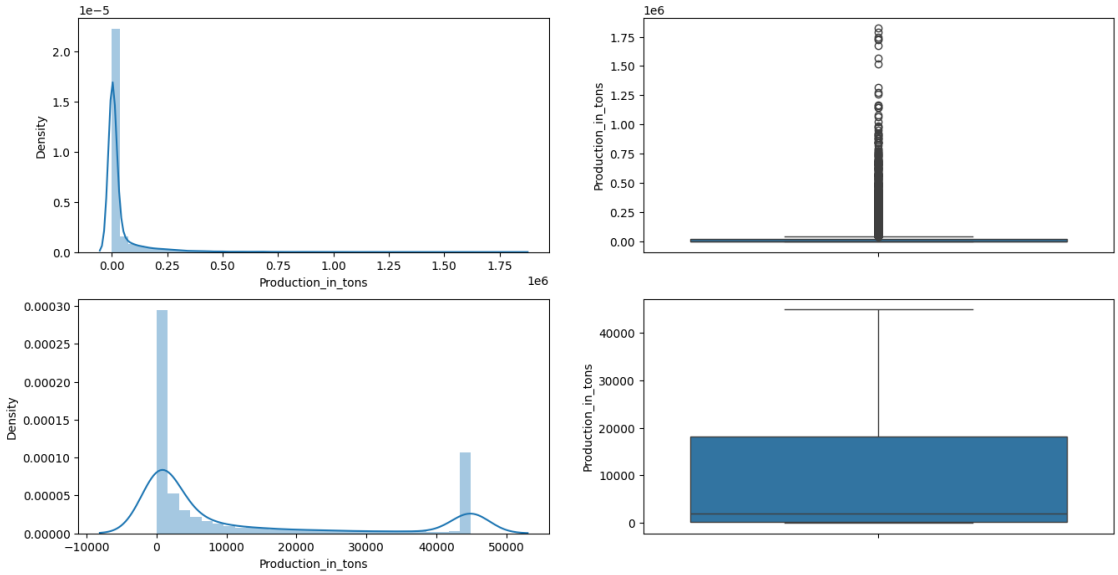
[55]:
```
new_df['Area_in_hectares']=np.where(new_df['Area_in_hectares']>high_out_Area,
                    high_out_Area,
                    np.where(new_df['Area_in_hectares']<low_out_Area,
                    low_out_Area,
                    new_df['Area_in_hectares']
    )
)
```

[56]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['Area_in_hectares'])

plt.subplot(2,2,2)
sns.boxplot(df['Area_in_hectares'])

plt.subplot(2,2,3)
sns.distplot(new_df['Area_in_hectares'])

plt.subplot(2,2,4)
sns.boxplot(new_df['Area_in_hectares'])

plt.show()
```



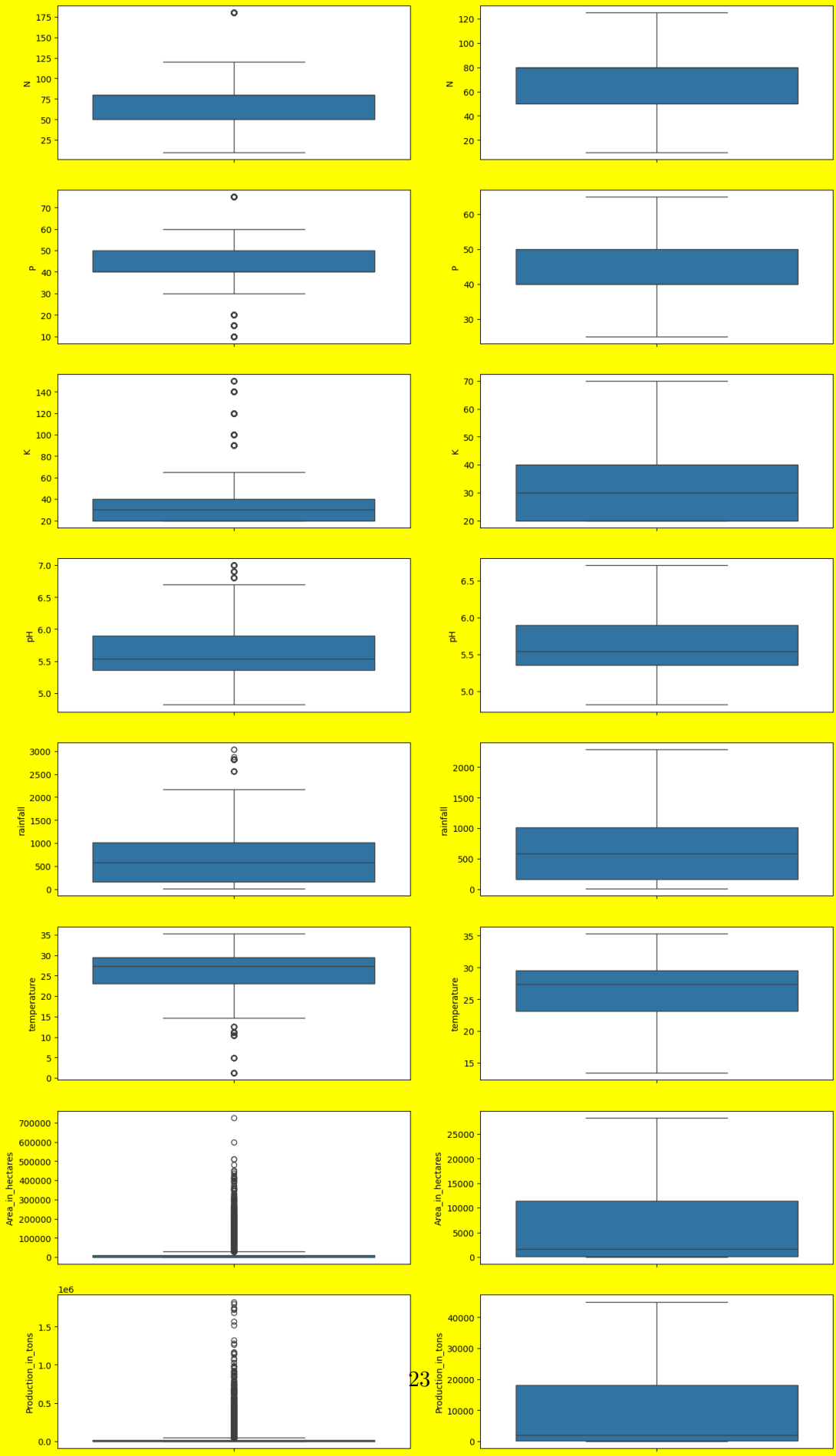
[57]:
```
new_df['Production_in_tons']=np.
  ↪where(new_df['Production_in_tons']>high_out_Production,
                    high_out_Production,
                    np.where(new_df['Production_in_tons']<low_out_Production,
                    low_out_Production,
                    new_df['Production_in_tons']
    )
)
```

[58]:
```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['Production_in_tons'])

plt.subplot(2,2,2)
sns.boxplot(df['Production_in_tons'])

plt.subplot(2,2,3)
sns.distplot(new_df['Production_in_tons'])

plt.subplot(2,2,4)
sns.boxplot(new_df['Production_in_tons'])

plt.show()
```



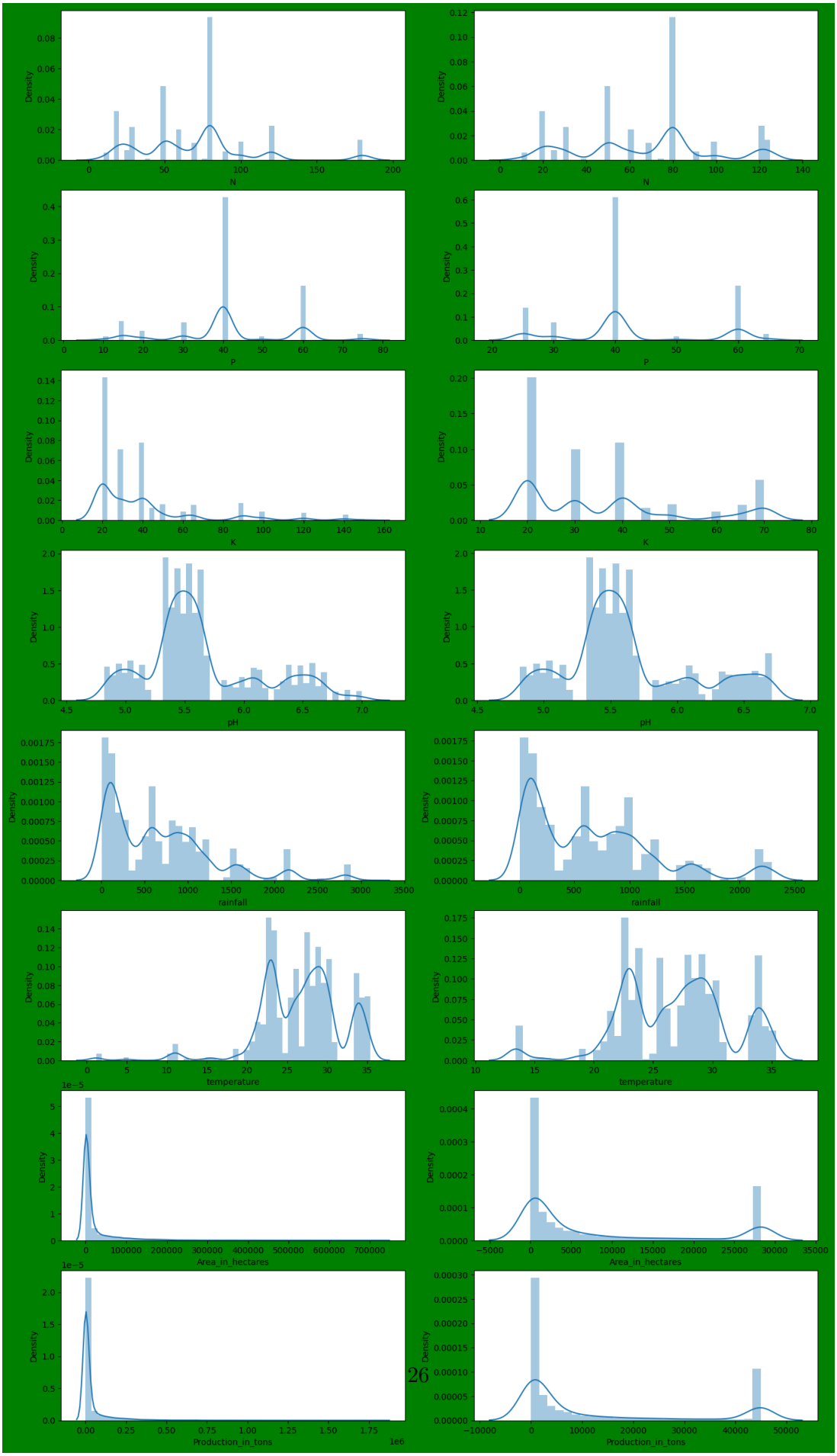
[59]:
```
plt.figure(figsize=(16,30),facecolor="yellow")
plt.subplot(8,2,1)
sns.boxplot(df['N'])

plt.subplot(8,2,2)
sns.boxplot(new_df['N'])

plt.subplot(8,2,3)
sns.boxplot(df['P'])

plt.subplot(8,2,4)
sns.boxplot(new_df['P'])

plt.subplot(8,2,5)
sns.boxplot(df['K'])

plt.subplot(8,2,6)
sns.boxplot(new_df['K'])

plt.subplot(8,2,7)
sns.boxplot(df['pH'])

plt.subplot(8,2,8)
sns.boxplot(new_df['pH'])

plt.subplot(8,2,9)
sns.boxplot(df['rainfall'])
```

```
plt.subplot(8,2,10)
sns.boxplot(new_df['rainfall'])

plt.subplot(8,2,11)
sns.boxplot(df['temperature'])

plt.subplot(8,2,12)
sns.boxplot(new_df['temperature'])

plt.subplot(8,2,13)
sns.boxplot(df['Area_in_hectares'])

plt.subplot(8,2,14)
sns.boxplot(new_df['Area_in_hectares'])

plt.subplot(8,2,15)
sns.boxplot(df['Production_in_tons'])

plt.subplot(8,2,16)
sns.boxplot(new_df['Production_in_tons'])

plt.show()
```

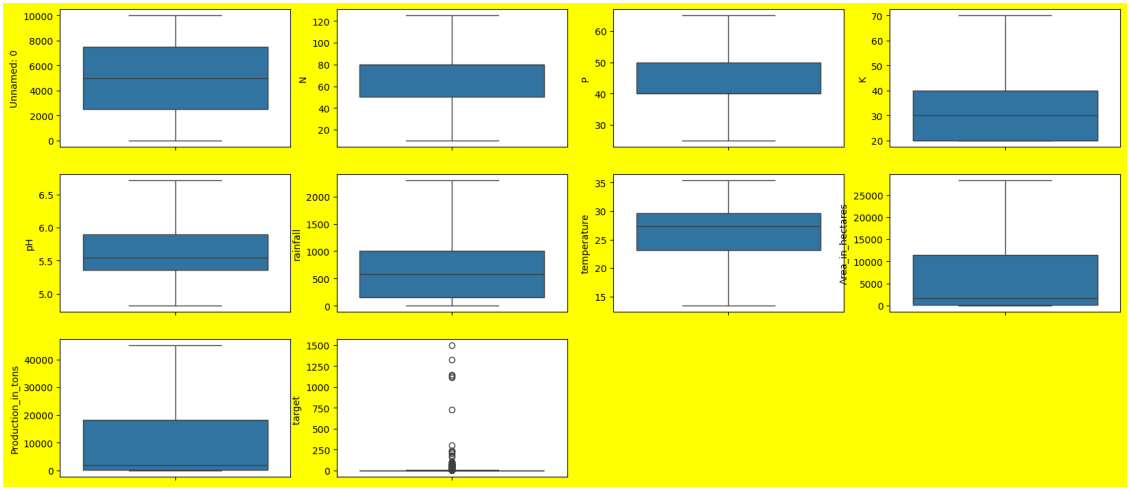

N
P
K
pH
rainfall
temperature
Area_in_hectares
Production_in_tons
1e6

[60]:
```
plt.figure(figsize=(16,30),facecolor="green")
plt.subplot(8,2,1)
sns.distplot(df['N'])

plt.subplot(8,2,2)
sns.distplot(new_df['N'])

plt.subplot(8,2,3)
sns.distplot(df['P'])

plt.subplot(8,2,4)
sns.distplot(new_df['P'])

plt.subplot(8,2,5)
sns.distplot(df['K'])

plt.subplot(8,2,6)
sns.distplot(new_df['K'])

plt.subplot(8,2,7)
sns.distplot(df['pH'])

plt.subplot(8,2,8)
sns.distplot(new_df['pH'])

plt.subplot(8,2,9)
sns.distplot(df['rainfall'])

plt.subplot(8,2,10)
sns.distplot(new_df['rainfall'])

plt.subplot(8,2,11)
sns.distplot(df['temperature'])

plt.subplot(8,2,12)
sns.distplot(new_df['temperature'])

plt.subplot(8,2,13)
sns.distplot(df['Area_in_hectares'])

plt.subplot(8,2,14)
sns.distplot(new_df['Area_in_hectares'])

plt.subplot(8,2,15)
sns.distplot(df['Production_in_tons'])
```

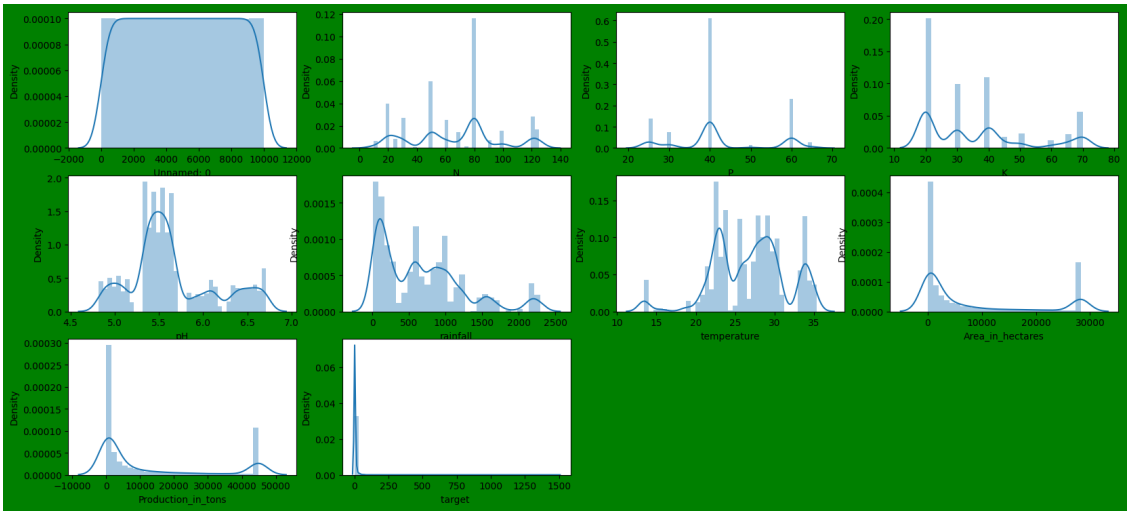
```
plt.subplot(8,2,16)
sns.distplot(new_df['Production_in_tons'])

plt.show()
```


Density
N
P
K
pH
rainfall
temperature
Area_in_hectares
Production_in_tons
1e−5
1e6

[61]:
```
plt.figure(figsize=(20,12),facecolor="yellow")
for i, column in enumerate(new_df.select_dtypes(include='number'),1):
    plt.subplot(4,4,i)
    sns.boxplot(new_df[column])
```



[62]:
```
plt.figure(figsize=(20, 12),facecolor="green")
for i, column in enumerate(new_df.select_dtypes(include='number'),1):
    plt.subplot(4,4,i)
    sns.distplot(new_df[column])
```



[63]:
```
for i in new_df.select_dtypes(include='number'):
    skewness=new_df[i].skew()
    print("The skewness of",i,"is",skewness)
```

```
The skewness of Unnamed: 0 is 0.0
The skewness of N is 0.09496749839203446
The skewness of P is 0.4255254081825508
The skewness of K is 0.8564864932522908
The skewness of pH is 0.6888507424055778
The skewness of rainfall is 1.0132776884705987
The skewness of temperature is -0.35609624356641484
The skewness of Area_in_hectares is 1.1903582088137628
The skewness of Production_in_tons is 1.19659013731798
The skewness of target  is 37.45537628503799
```

2 Encoding

2.1 Binary Encoder

[64]:
```
from category_encoders import BinaryEncoder
```

[65]:
```
bi_enc = BinaryEncoder()
```

[66]:
```
df_1=pd.read_csv('agriculture.csv')
df_1.head()
```

[66]:

	Unnamed: 0	Crop_Type	Crop	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	target
0	0	kharif	cotton	120	40	20	5.46	654.34	29.266667	7300	9400	1.287671
1	1	kharif	horsegram	20	60	20	6.18	654.34	29.266667	3300	1000	0.303030
2	2	kharif	jowar	80	40	40	5.42	654.34	29.266667	10100	10200	1.009901
3	3	kharif	maize	80	40	20	5.62	654.34	29.266667	2800	4900	1.750000
4	4	kharif	moong	20	40	20	5.68	654.34	29.266667	1300	500	0.384615

[67]:
```
df_1_bincode=bi_enc.fit_transform(df_1[['Crop','Crop_Type']])
```

[68]:
```
df_1_bincode
```

[68]:

	Crop_0	Crop_1	Crop_2	Crop_3	Crop_4	Crop_Type_0	Crop_Type_1
0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0

```
2           0       0       0       1       1           0           0
3           0       0       1       0       0           0           0
4           0       0       1       0       1           0           0
…       …       …       …       …       …           …           …
9996        0       0       1       0       0           0           1
9997        0       0       1       0       1           0           1
9998        0       1       1       1       1           1           0
9999        1       0       0       0       0           1           0
10000       0       0       1       0       0           0           0

       Crop_Type_2
0                1
1                1
2                1
3                1
4                1
…              …
9996             1
9997             1
9998             0
9999             0
10000            1

[10001 rows x 8 columns]
```

[69]:
```
df_1conc=pd.concat([df_1,df_1_bincode],axis=1)
df_1conc.head()
```

[69]:
```
   Unnamed: 0 Crop_Type       Crop    N   P   K    pH  rainfall  temperature  \
0           0    kharif     cotton  120  40  20  5.46    654.34    29.266667
1           1    kharif  horsegram   20  60  20  6.18    654.34    29.266667
2           2    kharif      jowar   80  40  40  5.42    654.34    29.266667
3           3    kharif      maize   80  40  20  5.62    654.34    29.266667
4           4    kharif      moong   20  40  20  5.68    654.34    29.266667

   Area_in_hectares  Production_in_tons    target   Crop_0  Crop_1  Crop_2  \
0              7300                9400  1.287671       0       0       0
1              3300                1000  0.303030       0       0       0
2             10100               10200  1.009901       0       0       0
3              2800                4900  1.750000       0       0       1
4              1300                 500  0.384615       0       0       1

   Crop_3  Crop_4  Crop_Type_0  Crop_Type_1  Crop_Type_2
0       0       1            0            0            1
1       1       0            0            0            1
2       1       1            0            0            1
3       0       0            0            0            1
```

```
4       0       1              0              0              1
```

3 Feature Scaling

[70]:
```
from sklearn.preprocessing import StandardScaler
```

[71]:
```
import seaborn as sns
```

[72]:
```
df_1=df_1.drop(['Crop','Crop_Type'],axis=1)
```

[73]:
```
df_1.head()
```

[73]:

	Unnamed: 0	N	P	K	pH	rainfall	temperature	Area_in_hectares	Production_in_tons	target
0	0	120	40	20	5.46	654.34	29.266667	7300	9400	1.287671
1	1	20	60	20	6.18	654.34	29.266667	3300	1000	0.303030
2	2	80	40	40	5.42	654.34	29.266667	10100	10200	1.009901
3	3	80	40	20	5.62	654.34	29.266667	2800	4900	1.750000
4	4	20	40	20	5.68	654.34	29.266667	1300	500	0.384615

[82]:
```
print(df_1.columns)
```

```
Index(['Unnamed: 0', 'N', 'P', 'K', 'pH', 'rainfall', 'temperature',
       'Area_in_hectares', 'Production_in_tons', 'target '],
      dtype='object')
```

[83]:
```
y = df_1['target ']
y
```

[83]:
```
0          1.287671
1          0.303030
2          1.009901
3          1.750000
4          0.384615
             ...
9996       1.013158
9997       0.432377
9998      12.074468
9999      22.048058
10000      0.300969
```

```
Name: target , Length: 10001, dtype: float64
```

[85]:
```
x = df_1.drop('target ',axis=1)
x
```

[85]:
```
       Unnamed: 0    N   P   K    pH  rainfall  temperature  Area_in_hectares  \
0               0  120  40  20  5.46    654.34    29.266667              7300
1               1   20  60  20  6.18    654.34    29.266667              3300
2               2   80  40  40  5.42    654.34    29.266667             10100
3               3   80  40  20  5.62    654.34    29.266667              2800
4               4   20  40  20  5.68    654.34    29.266667              1300
…             …  …  ..  ..   …       …          …                 …
9996         9996   80  40  20  5.40     34.81    34.666667               152
9997         9997   20  40  20  5.60     34.81    34.666667               488
9998         9998  120  60  65  5.94    689.88    29.037273               752
9999         9999  180  60  90  5.02    689.88    29.037273              7595
10000       10000   80  40  20  5.48    579.75    34.010000             11247

       Production_in_tons
0                    9400
1                    1000
2                   10200
3                    4900
4                     500
…                    …
9996                  154
9997                  211
9998                 9080
9999               167455
10000                3385

[10001 rows x 9 columns]
```

[87]:
```
scaler = StandardScaler()
x_scaler = scaler.fit_transform(x)
x_scaler
```

[87]:
```
array([[-1.73187763,  1.36720475, -0.10914841, …,  0.49518872,
        -0.25376458, -0.25972778],
       [-1.73153125, -1.32131628,  1.31963861, …,  0.49518872,
        -0.34084278, -0.33376255],
       [-1.73118488,  0.29179634, -0.10914841, …,  0.49518872,
        -0.19280984, -0.25267685],
       …,
       [ 1.73118488,  1.36720475,  1.31963861, …,  0.45001545,
        -0.3963116 , -0.26254815],
       [ 1.73153125,  2.98031736,  1.31963861, …,  0.45001545,
```

```
        -0.24734256,  1.13331571],
       [ 1.73187763,  0.29179634, -0.10914841, …,  1.42926675,
        -0.16784016, -0.31274196]])
```

[88]:
```
Df=pd.DataFrame(x_scaler)
Df
```

[88]:

	0	1	2	3	4	5	6	7	8
0	-1.731878	1.367205	-0.109148	-0.740559	-0.379825	-0.026304	0.495189	-0.253765	-0.259728
1	-1.731531	-1.321316	1.319639	-0.740559	1.095913	-0.026304	0.495189	-0.340843	-0.333763
2	-1.731185	0.291796	-0.109148	0.010933	-0.461811	-0.026304	0.495189	-0.192810	-0.252677
3	-1.730839	0.291796	-0.109148	-0.740559	-0.051883	-0.026304	0.495189	-0.351728	-0.299389
4	-1.730492	-1.321316	-0.109148	-0.740559	0.071095	-0.026304	0.495189	-0.384382	-0.338169
…	…	…	…	…	…	…	…	…	…
9996	1.730492	0.291796	-0.109148	-0.740559	-0.502803	-1.051366	1.558580	-0.409373	-0.341219
9997	1.730839	-1.321316	-0.109148	-0.740559	-0.092876	-1.051366	1.558580	-0.402059	-0.340717
9998	1.731185	1.367205	1.319639	0.950299	0.604000	0.032500	0.450015	-0.396312	-0.262548
9999	1.731531	2.980317	1.319639	1.889664	-1.281665	0.032500	0.450015	-0.247343	1.133316
10000	1.731878	0.291796	-0.109148	-0.740559	-0.338832	-0.149719	1.429267	-0.167840	-0.312742

[10001 rows x 9 columns]

[90]:
```
Df.corr()
```

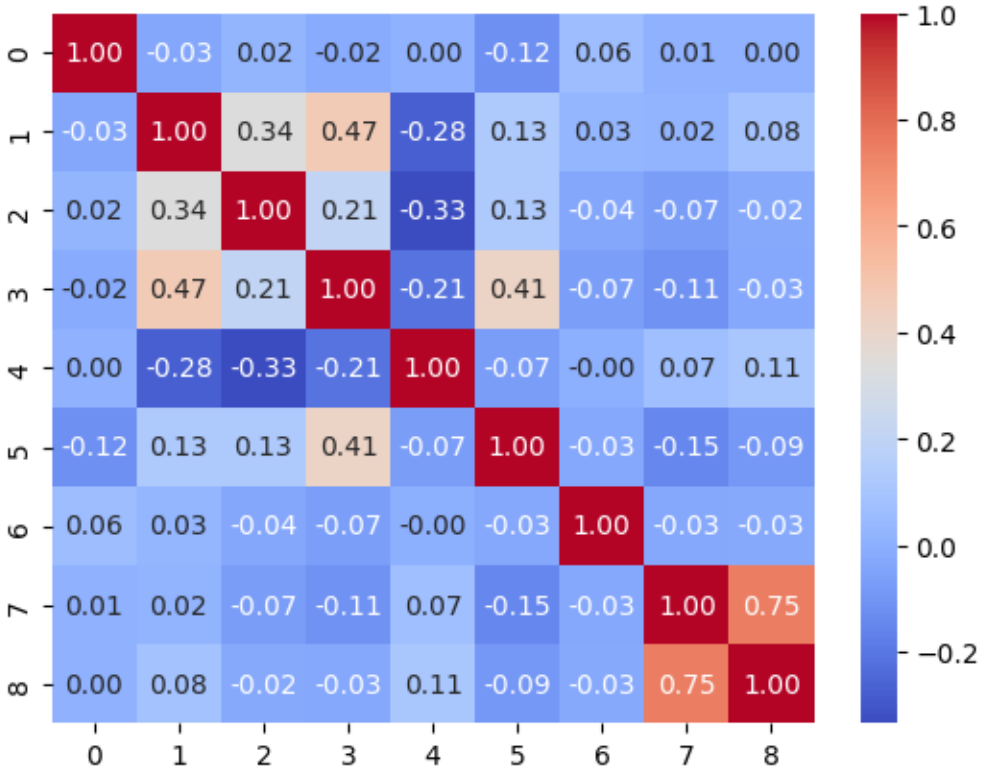
[90]:

	0	1	2	3	4	5	6	7	8
0	1.000000	-0.033189	0.020015	-0.018716	0.004477	-0.117434	0.056472	0.006091	0.004372
1	-0.033189	1.000000	0.335317	0.467259	-0.277163	0.128159	0.028687	0.016556	0.082932
2	0.020015	0.335317	1.000000	0.205663	-0.334898	0.126305	-0.037438	-0.069552	-0.022940
3	-0.018716	0.467259	0.205663	1.000000	-0.211495	0.411469	-0.065351	-0.111029	-0.028182
4	0.004477	-0.277163	-0.334898	-0.211495	1.000000	-0.069599	-0.000626	0.070010	0.111721
5	-0.117434	0.128159	0.126305	0.411469	-0.069599	1.000000	-0.030709	-0.148148	-0.092841
6	0.056472	0.028687	-0.037438	-0.065351	-0.000626	-0.030709	1.000000	-0.028585	-0.025893
7	0.006091	0.016556	-0.069552	-0.111029	0.070010	-0.148148	-0.028585	1.000000	0.000000
8	0.004372	0.082932	-0.022940	-0.028182	0.111721	-0.092841	-0.025893	0.000000	1.000000

```
          7         8
0  0.006091  0.004372
1  0.016556  0.082932
2 -0.069552 -0.022940
3 -0.111029 -0.028182
4  0.070010  0.111721
5 -0.148148 -0.092841
6 -0.028585 -0.025893
7  1.000000  0.753248
8  0.753248  1.000000
```

[91]:
```
sns.heatmap(Df.corr( ),annot=True, cmap='coolwarm', fmt=".2f")
```

[91]: <Axes: >



4 Train test split

[92]:
```
from sklearn.model_selection import train_test_split
```

[94]:
```
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.30,␣
↪random_state=22)
X_train
```

[94]:
```
      Unnamed: 0    N   P    K    pH  rainfall  temperature  Area_in_hectares  \
9889        9889   80  40   20  5.38    34.810    34.666667                47
3681        3681   50  60   30  5.40   770.440    28.680000              1500
1879        1879   70  40   60  5.88  2817.860    27.909091              1244
6742        6742   80  40   20  5.40  1246.715    22.600000              5751
1261        1261   80  40   20  5.58   810.260    29.956364              4700
…            …   …   ..  …   …       …           …                 …
4587        4587  100  40  140  5.96  1501.980    25.818182              1345
6646        6646   20  60   20  5.54  1246.715    22.600000             53052
5478        5478   40  60   20  5.06   840.460    33.583333                26
8548        8548  180  60   90  4.92   167.380    23.560000               196
6276        6276  100  75   50  6.62   810.260    29.956364              2100

      Production_in_tons
9889                  48
3681                 600
1879                 626
6742               21373
1261                7700
…                   …
4587                1274
6646               58866
5478                  18
8548                2670
6276              135500

[7000 rows x 9 columns]
```

[95]:
```
X_test
```

[95]:
```
      Unnamed: 0    N   P    K    pH  rainfall  temperature  Area_in_hectares  \
710          710   50  10   60  5.34  1026.640    29.186364                 7
6211        6211  100  75   50  6.52   810.260    29.956364               400
4567        4567   20  40   20  5.54    34.810    34.666667               427
199          199   20  60   20  5.36  1712.660    11.200000               187
4422        4422   50  40   20  5.04    75.320    22.676000              1645
…            …   …   ..  …   …       …           …                 …
6191        6191   60  30   30  6.12    15.340    27.276000             20300
1813        1813   50  40   20  5.38  1246.715    22.600000                84
5889        5889   80  40   40  5.44    98.980    34.923333              1254
6567        6567   20  60   20  5.36   664.940    22.033333                22
4592        4592   25  60  100  4.92  1501.980    25.818182               820
```

```
      Production_in_tons
710                    9
6211               29500
4567                 181
199                  405
4422                 794
…                    …
6191               51800
1813                  92
5889                1326
6567                  25
4592                1701

[3001 rows x 9 columns]
```

[96]:
```
Y_train
```

[96]:
```
9889     1.021277
3681     0.400000
1879     0.503215
6742     3.716397
1261     1.638298
          …
4587     0.947212
6646     1.109591
5478     0.692308
8548    13.622449
6276    64.523810
Name: target , Length: 7000, dtype: float64
```

[97]:
```
Y_test
```

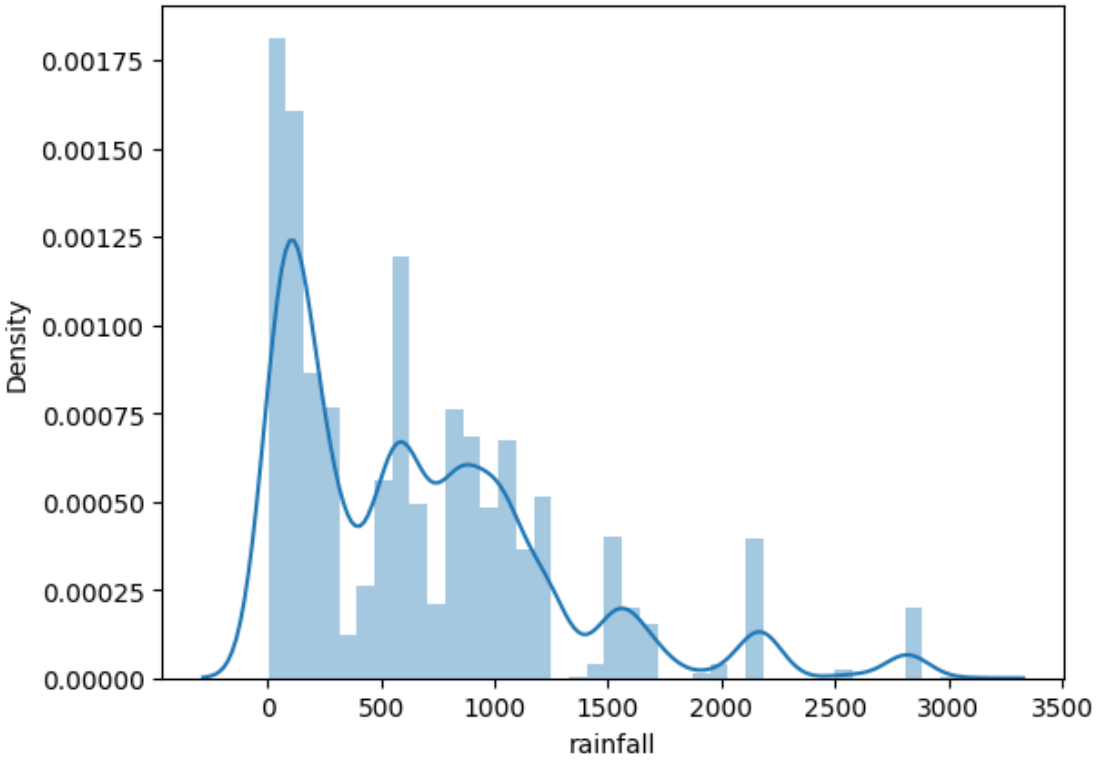
[97]:
```
710      1.285714
6211    73.750000
4567     0.423888
199      2.165775
4422     0.482675
          …
6191     2.551724
1813     1.095238
5889     1.057416
6567     1.136364
4592     2.074390
Name: target , Length: 3001, dtype: float64
```

5 Data Transformation

5.1 Log Transformation = np.log(df[”])

[114]:
```
sns.distplot(x['rainfall'])
```

[114]:
```
<Axes: xlabel='rainfall', ylabel='Density'>
```



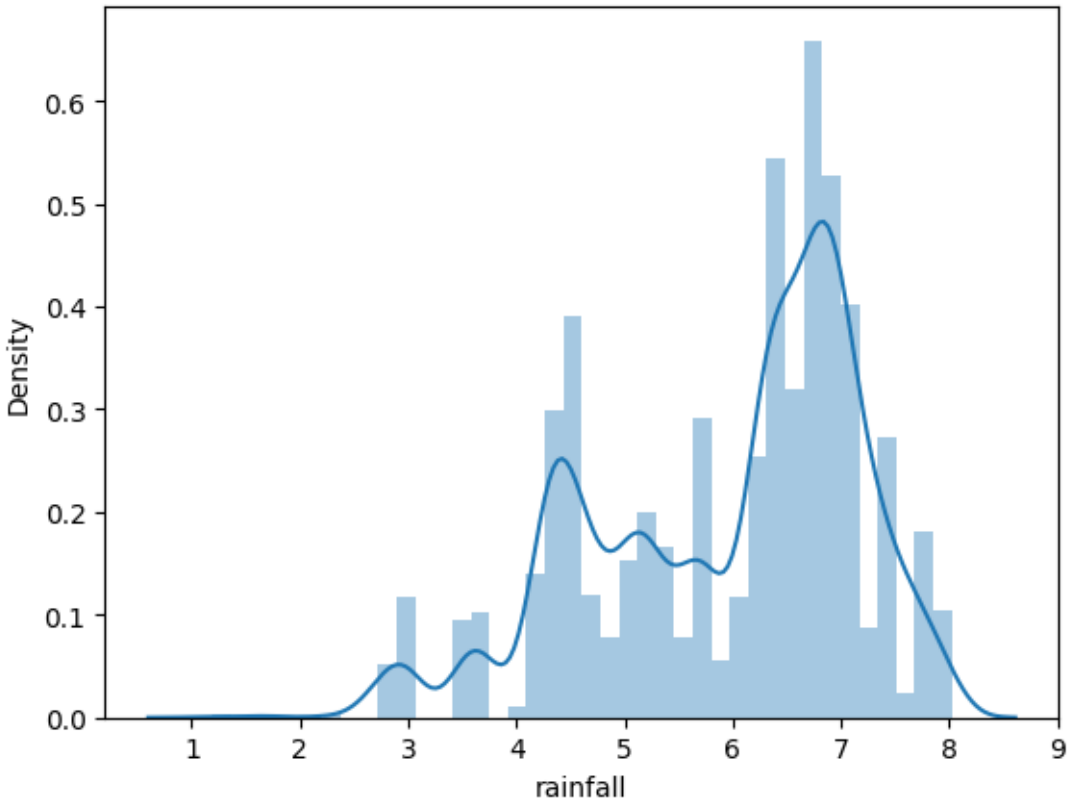
[115]:
```
x["rainfall"].skew()
```

[115]:
```
np.float64(1.268531509612503)
```

[116]:
```
log_dis = np.log(x['rainfall'])
```

[117]:
```
sns.distplot(log_dis)
```

[117]:
```
<Axes: xlabel='rainfall', ylabel='Density'>
```



[118]:
```
log_dis.skew()
```

[118]: np.float64(-0.6724806345315684)