In [21]:
```
import numpy as np
import pandas as pd
```

In [22]:
```
import matplotlib.pyplot as plt
```

In [3]:
```
import seaborn as sns
from ipywidgets import interact
```

In [4]:
```
data = pd.read_csv(r'C:\Users\ranim\Downloads\Crop_recommendation.csv')
```

In [5]:
```
print(data)
```

```
         N   P   K  temperature   humidity        ph    rainfall   label
0       90  42  43    20.879744  82.002744  6.502985  202.935536    rice
1       85  58  41    21.770462  80.319644  7.038096  226.655537    rice
2       60  55  44    23.004459  82.320763  7.840207  263.964248    rice
3       74  35  40    26.491096  80.158363  6.980401  242.864034    rice
4       78  42  42    20.130175  81.604873  7.628473  262.717340    rice
...    ...  ..  ..          ...        ...       ...         ...     ...
2195   107  34  32    26.774637  66.413269  6.780064  177.774507  coffee
2196    99  15  27    27.417112  56.636362  6.086922  127.924610  coffee
2197   118  33  30    24.131797  67.225123  6.362608  173.322839  coffee
2198   117  32  34    26.272418  52.127394  6.758793  127.175293  coffee
2199   104  18  30    23.603016  60.396475  6.779833  140.937041  coffee

[2200 rows x 8 columns]
```

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

Out[6]:

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

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

Out[7]:

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

```
data.shape
```

In [8]:
```
data.isnull().sum()
```

Out[8]:
```
N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64
```

In [9]:
```
data.shape
```

Out[9]:
```
(2200, 8)
```

In [10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   N            2200 non-null   int64  
 1   P            2200 non-null   int64  
 2   K            2200 non-null   int64  
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object 
dtypes: float64(4), int64(3), object(1)
memory usage: 137.6+ KB
```

In [11]:
```
data.describe()
```

Out[11]:

	N	P	K	temperature	humidity	ph	ra
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.0(
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.4(
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.9(
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.2
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.5(
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.8(
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.2(
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.5(

In [12]:
```
data.count()
```

Out[12]:
```
N              2200
P              2200
K              2200
temperature    2200
humidity       2200
ph             2200
rainfall       2200
label          2200
dtype: int64
```

In [13]:
```
data['label'].value_counts()
```

Out[13]:
```
label
rice           100
maize          100
jute           100
cotton         100
coconut        100
papaya         100
orange         100
apple          100
muskmelon      100
watermelon     100
grapes         100
mango          100
banana         100
pomegranate    100
lentil         100
blackgram      100
mungbean       100
mothbeans      100
pigeonpeas     100
kidneybeans    100
chickpea       100
coffee         100
Name: count, dtype: int64
```

In [14]:
```
print("Average Ratio of Nitrogen in the Soil: {0:.2f}".format(data['N'].mean
print("Average Ratio of Posphorous in the Soil: {0:.2f}".format(data['P'].me
print("Average Ratio of Potassium in the Soil: {0:.2f}".format(data['K'].mea
print("Average Tempature in Celsius: {0:.2f}".format(data['temperature'].mea
print("Average Relative Humidity in %: {0:.2f}".format(data['humidity'].mean
print("Average PH value of the Soil: {0:.2f}".format(data['ph'].mean()))
print("Average Rainfall mm: {0:.2f}".format(data['rainfall'].mean()))
```

```
Average Ratio of Nitrogen in the Soil: 50.55
Average Ratio of Posphorous in the Soil: 53.36
Average Ratio of Potassium in the Soil: 48.15
Average Tempature in Celsius: 25.62
Average Relative Humidity in %: 71.48
Average PH value of the Soil: 6.47
Average Rainfall mm: 103.46
```

In [15]:
```
@interact
def summary(crops = list(data['label'].value_counts().index)):
    x=data[data['label']== crops]

    print("-----------------------------------------------------------------

    print("statistics for Nitrogen")
    print("Minimum Nitrogen required:",x['N'].min())
    print("Average Nitrogen required:",x['N'].mean())
    print("Maximum Nitrogen required:",x['N'].max())

    print("-----------------------------------------------------------------

    print("statistics for Phosphorous")
    print("Minimum Phosphorous required:",x['P'].min())
    print("Average Phosphorous required:",x['P'].mean())
    print("Maximum Phosphorous required:",x['P'].max())

    print("-----------------------------------------------------------------


    print("statistics for Potassium")
    print("Minimum Potassium required:",x['K'].min())
    print("Average Potassium required:",x['K'].mean())
    print("Maximum Potassium required:",x['K'].max())


    print("-----------------------------------------------------------------
    print("statistics for Temperature")
    print("Minimum Temperature required: {0:.2f}".format(x['temperature'].mi
    print("Average Temperature required: {0:.2f}".format(x['temperature'].me
    print("Maximum Temperature required: {0:.2f}".format(x['temperature'].ma


    print("-----------------------------------------------------------------

    print("statistics for Humidity")
    print("Minimum Humidity required: {0:.2f}".format(x['humidity'].min()))
    print("Average Humidity required: {0:.2f}".format(x['humidity'].mean()))
    print("Maximum Humidity required: {0:.2f}".format(x['humidity'].max()))


    print("-----------------------------------------------------------------

    print("statistics for PH")
    print("Minimum PH required: {0:.2f}".format(x['ph'].min()))
    print("Average PH required: {0:.2f}".format(x['ph'].mean()))
    print("Maximum PH required: {0:.2f}".format(x['ph'].max()))

    print("-----------------------------------------------------------------

    print("statistics for Rainfall")
    print("Minimum Rainfall required: {0:.2f}".format(x['rainfall'].min()))
    print("Average Rainfall required: {0:.2f}".format(x['rainfall'].mean()))
    print("Maximum Rainfall required: {0:.2f}".format(x['rainfall'].max()))
```

crops rice

```
------------------------------------------------------------------
statistics for Nitrogen
Minimum Nitrogen required: 60
Average Nitrogen required: 79.89
Maximum Nitrogen required: 99
------------------------------------------------------------------
statistics for Phosphorous
Minimum Phosphorous required: 35
Average Phosphorous required: 47.58
Maximum Phosphorous required: 60
------------------------------------------------------------------
statistics for Potassium
Minimum Potassium required: 35
Average Potassium required: 39.87
Maximum Potassium required: 45
------------------------------------------------------------------
statistics for Temperature
Minimum Temperature required: 20.05
Average Temperature required: 23.69
Maximum Temperature required: 26.93
------------------------------------------------------------------
statistics for Humidity
Minimum Humidity required: 80.12
Average Humidity required: 82.27
Maximum Humidity required: 84.97
------------------------------------------------------------------
statistics for PH
Minimum PH required: 5.01
Average PH required: 6.43
Maximum PH required: 7.87
------------------------------------------------------------------
--
statistics for Rainfall
Minimum Rainfall required: 182.56
Average Rainfall required: 236.18
Maximum Rainfall required: 298.56
```

In [16]:
```
@interact
def compare(conditions=['N','P','K','temperature','ph','humidity','rainfall'
    print("Average Value for",conditions,"is {0:.2f}".format(data[conditions
    print("--------------------------------------------------------------------
    print("Rice:{0:.2f}".format(data[(data['label']=='rice')][conditions].me
    print("Blacgram:{0:.2f}".format(data[(data['label']=='blackgram')][condi
    print("Banana{0:.2f}".format(data[(data['label']=='banana')][conditions]
    print("Jute:{0:.2f}".format(data[(data['label']=='jute')][conditions].me
    print("Coconut:{0:.2f}".format(data[(data['label']=='coconut')][conditio
    print("Apple:{0:.2f}".format(data[(data['label']=='apple')][conditions].
    print("Papaya:{0:.2f}".format(data[(data['label']=='papaya')][conditions
    print("Muskmelon:{0:.2f}".format(data[(data['label']=='muskmelon')][conc
    print("Grapes:{0:.2f}".format(data[(data['label']=='grapes')][conditions
    print("Watermelon:{0:.2f}".format(data[(data['label']=='watermelon')][cc
    print("Kidneybeans:{0:.2f}".format(data[(data['label']=='kidneybeans')][
    print("Mungbean:{0:.2f}".format(data[(data['label']=='mungbean')][condit
    print("Orange:{0:.2f}".format(data[(data['label']=='orange')][conditions
    print("Chickpea:{0:.2f}".format(data[(data['label']=='chickpea')][condit
    print("Lentil:{0:.2f}".format(data[(data['label']=='lentil')][conditions
    print("Cotten:{0:.2f}".format(data[(data['label']=='cotton')][conditions
    print("Maize:{0:.2f}".format(data[(data['label']=='maize')][conditions].
    print("Mothbeans:{0:.2f}".format(data[(data['label']=='mothbeans')][conc
    print("Pigeonpeas:{0:.2f}".format(data[(data['label']=='pigeonpeas')][cc
    print("Mango:{0:.2f}".format(data[(data['label']=='mango')][conditions].
    print("Pomegranate:{0:.2f}".format(data[(data['label']=='pomegranate')][
    print("Coffee:{0:.2f}".format(data[(data['label']=='coffee')][conditions
```

conditions N

```
Average Value for N is 50.55
--------------------------------------------------------------------------
----------
Rice:79.89
Blacgram:40.02
Banana100.23
Jute:78.40
Coconut:21.98
Apple:20.80
Papaya:49.88
Muskmelon:100.32
Grapes:23.18
Watermelon:99.42
Kidneybeans:20.75
Mungbean:20.99
Orange:19.58
Chickpea:40.09
Lentil:18.77
Cotten:117.77
Maize:77.76
Mothbeans:21.44
Pigeonpeas:20.73
Mango:20.07
Pomegranate:18.87
Coffee:101.20
```

In [17]:
```
@interact
def compare(conditions=['N','P','K','temperature','ph','humidity','rainfall'
    print("Crops which require greter than average",conditions,'\n')
    print(data[data[conditions]>data[conditions].mean()]['label'].unique())
    print("----------------------------------------------------")

    print("Crops which require Less than average",conditions,'\n')
    print(data[data[conditions]<=data[conditions].mean()]['label'].unique())
```

conditions N

```
Crops which require greter than average N

['rice' 'maize' 'chickpea' 'blackgram' 'banana' 'watermelon' 'muskmelon'
 'papaya' 'cotton' 'jute' 'coffee']
----------------------------------------------------
Crops which require Less than average N

['chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans' 'mungbean' 'blackgram'
 'lentil' 'pomegranate' 'mango' 'grapes' 'apple' 'orange' 'papaya'
 'coconut']
```

In [18]:
```
plt.subplot(2,4,1)
sns.distplot(data['N'],color='pink')
plt.xlabel('Ratio of Nitrogen',fontsize=12)
plt.grid()

plt.subplot(2,4,2)
sns.distplot(data['P'],color='yellow')
plt.xlabel('Ratio of Phosphorous',fontsize=12)
plt.grid()

plt.subplot(2,4,3)
sns.distplot(data['K'],color='darkblue')
plt.xlabel('Ratio of Pottasium',fontsize=12)
plt.grid()


plt.subplot(2,4,4)
sns.distplot(data['temperature'],color='blue')
plt.xlabel('Temperature',fontsize=12)
plt.grid()


plt.subplot(2,4,5)
sns.distplot(data['rainfall'],color='black')
plt.xlabel('Rainfall',fontsize=12)
plt.grid()

plt.subplot(2,4,6)
sns.distplot(data['humidity'],color='grey')
plt.xlabel('Humidity',fontsize=12)
plt.grid()

plt.subplot(2,4,7)
sns.distplot(data['ph'],color='darkgrey')
plt.xlabel('pH Level',fontsize=12)
plt.grid()

plt.suptitle('Distribution For Agricultural Conditions',fontsize=20)
plt.show()
```

```
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data['temperature'],color='blue')
C:\Users\ranim\AppData\Local\Temp\ipykernel_11316\994104938.py:24: UserW
arning:

`distplot` is a deprecated function and will be removed in seaborn v0.1
4.0.

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

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(data['rainfall'],color='black')
```

In [19]:
```
print("Some Interesting Patterns")
print("------------------------------------")
print("Crops which requires very high ratio of nitrogen content in soil:",da
print("Crops which requires very high ratio of Phosphorous content in soil:"
print("Crops which requires very high ratio of potassium content in soil:",d
print("Crops which requires very high rainfall:",data[data['rainfall']>200][
print("Crops which requires very low temperature:",data[data['temperature']<
print("Crops which requires very high temperature:",data[data['temperature']
print("Crops which requires very low humidity:",data[data['humidity']>20]['l
print("Crops which requires very low pH:",data[data['ph']<4]['label'].unique
print("Crops which requires very high pH:",data[data['ph']>9]['label'].uniqu
```

```
Some Interesting Patterns
------------------------------------
Crops which requires very high ratio of nitrogen content in soil: ['cotto
n']
Crops which requires very high ratio of Phosphorous content in soil: ['gra
pes' 'apple']
Crops which requires very high ratio of potassium content in soil: ['grape
s' 'apple']
Crops which requires very high rainfall: ['rice' 'papaya' 'coconut']
Crops which requires very low temperature: ['grapes']
Crops which requires very high temperature: ['grapes' 'papaya']
Crops which requires very low humidity: ['rice' 'maize' 'kidneybeans' 'pig
eonpeas' 'mothbeans' 'mungbean'
 'blackgram' 'lentil' 'pomegranate' 'banana' 'mango' 'grapes' 'watermelon'
 'muskmelon' 'apple' 'orange' 'papaya' 'coconut' 'cotton' 'jute' 'coffee']
Crops which requires very low pH: ['mothbeans']
Crops which requires very high pH: ['mothbeans']
```

In [20]:
```
print("Summer Crops")
print(data[(data['temperature']>30) & (data['humidity']>50)]['label'].unique

print("-------------------------------------------------")

print("Winter Crops")
print(data[(data['temperature']<20) & (data['humidity']>30)]['label'].unique

print("-------------------------------------------------")

print("Rainy Crops")
print(data[(data['temperature']>200) & (data['humidity']>30)]['label'].uniqu
```

```
Summer Crops
['pigeonpeas' 'mothbeans' 'blackgram' 'mango' 'grapes' 'orange' 'papaya']
-------------------------------------------------
Winter Crops
['maize' 'pigeonpeas' 'lentil' 'pomegranate' 'grapes' 'orange']
-------------------------------------------------
Rainy Crops
[]
```

In [ ]: