```
#Supervised Machine Learning
#Sai kumar Murarishetti
#Week 6: Final Project Part 1


(https:/https://www.kaggle.com/datasets/asishpandey/crop-production-in
data/)
```

Supervised Machine Learning

Sai kumar Murarishetti

Week 6: Final Project Part 1

(https:/https://www.kaggle.com/datasets/asishpandey/crop-production-in-india/data/)

```
# Used for Importing of libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from google.colab import files
```

```
# This is used to upload the dataset
df=files.upload()
df = pd.read_csv('Crop_production.csv')
```

Choose Files Crop_production.csv

- **Crop_production.csv**(text/csv) - 9887752 bytes, last modified: 10/8/2023 - 100% done

```
Saving Crop_production.csv to Crop_production (3).csv
```

```
# This is used to display first 10 lines from the dataset
print(df.head(10))
```

```
   Unnamed: 0      State_Name Crop_Type       Crop    N   P   K    pH  \
0           0  andhra pradesh    kharif     cotton  120  40  20  5.46
1           1  andhra pradesh    kharif  horsegram   20  60  20  6.18
2           2  andhra pradesh    kharif      jowar   80  40  40  5.42
3           3  andhra pradesh    kharif      maize   80  40  20  5.62
4           4  andhra pradesh    kharif      moong   20  40  20  5.68
5           5  andhra pradesh    kharif       ragi   50  40  20  5.64
6           6  andhra pradesh    kharif       rice   80  40  40  5.54
7           7  andhra pradesh    kharif  sunflower   50  60  30  5.36
8           8  andhra pradesh      rabi  horsegram   20  60  20  6.00
9           9  andhra pradesh      rabi      jowar   80  40  40  5.50

   rainfall  temperature  Area_in_hectares  Production_in_tons  \
0    654.34    29.266667            7300.0              9400.0
1    654.34    29.266667            3300.0              1000.0
2    654.34    29.266667           10100.0             10200.0
3    654.34    29.266667            2800.0              4900.0
4    654.34    29.266667            1300.0               500.0
5    654.34    29.266667            6700.0             11800.0
6    654.34    29.266667           35600.0             75400.0
7    654.34    29.266667           35900.0             11100.0
8    288.30    25.460000             600.0               200.0
9    288.30    25.460000           18800.0              9400.0

   Yield_ton_per_hec
0           1.287671
1           0.303030
2           1.009901
3           1.750000
4           0.384615
5           1.761194
6           2.117978
7           0.309192
8           0.333333
9           0.500000
```

```
# This is used to Display a summary of DataFrame 'df' for data inspection and understanding.
print(df.info)
```

```
<bound method DataFrame.info of        Unnamed: 0      State_Name Crop_Type       Crop    N   P   K    pH  \
0               0  andhra pradesh    kharif     cotton  120  40  20  5.46   
1               1  andhra pradesh    kharif  horsegram   20  60  20  6.18   
2               2  andhra pradesh    kharif      jowar   80  40  40  5.42   
3               3  andhra pradesh    kharif      maize   80  40  20  5.62   
4               4  andhra pradesh    kharif      moong   20  40  20  5.68   
...           ...             ...       ...        ...  ...  ..  ..   ...   
99844       99844     west bengal      rabi      wheat   60  30  30  6.70   
99845       99845     west bengal    summer      maize   80  40  20  5.68   
99846       99846     west bengal    summer       rice   80  40  40  5.64   
99847       99847     west bengal      rabi       rice   80  40  40  5.42   
99848       99848     west bengal      rabi    sesamum   30  15  30  6.54   

       rainfall  temperature  Area_in_hectares  Production_in_tons  \
0        654.34    29.266667            7300.0              9400.0   
1        654.34    29.266667            3300.0              1000.0   
2        654.34    29.266667           10100.0             10200.0   
3        654.34    29.266667            2800.0              4900.0   
4        654.34    29.266667            1300.0               500.0   
...         ...          ...               ...                 ...   
99844    152.54    22.280000            2013.0              5152.0   
99845    182.50    29.200000             258.0               391.0   
99846    182.50    29.200000             105.0               281.0   
99847    152.54    22.280000          152676.0            261435.0   
99848    152.54    22.280000             244.0                95.0   

       Yield_ton_per_hec  
0               1.287671  
1               0.303030  
2               1.009901  
3               1.750000  
4               0.384615  
...                  ...  
99844           2.559364  
99845           1.515504  
99846           2.676190  
99847           1.712352  
99848           0.389344  

[99849 rows x 13 columns]>
```

```
#this is used to display rows and colums
df.shape
```

```
(99849, 13)
```

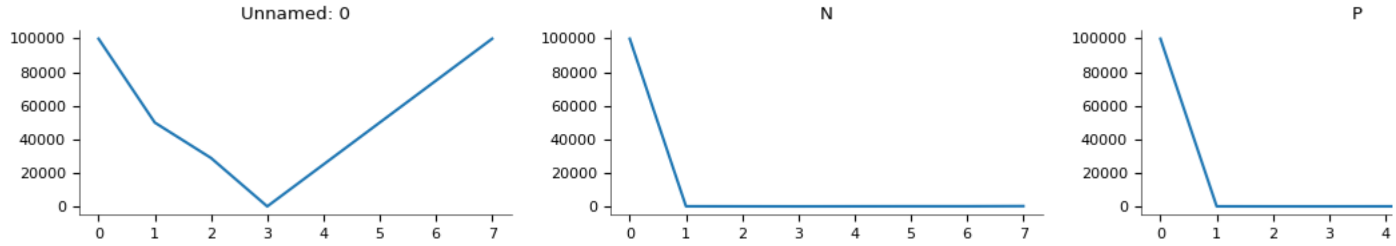
```
#This is Generate descriptive statistics for numeric columns in DataFrame.
df.describe()
```

index	Unnamed: 0	N	P	K	pH	rainfall
count	99849.0	99849.0	99849.0	99849.0	99849.0	998
mean	49924.0	69.81682340333904	41.59365642119601	42.037827118949615	5.643624272651705	701.1510848590
std	28824.06785136338	39.571469100527054	15.056507756682123	28.430263218391968	0.5052825728335681	604.7015524947
min	0.0	10.0	10.0	10.0	3.82	3.274
25%	24962.0	50.0	40.0	20.0	5.36	15
50%	49924.0	75.0	40.0	30.0	5.54	57
75%	74886.0	80.0	60.0	50.0	5.96	111
max	99848.0	180.0	125.0	200.0	7.0	3322.0599999999

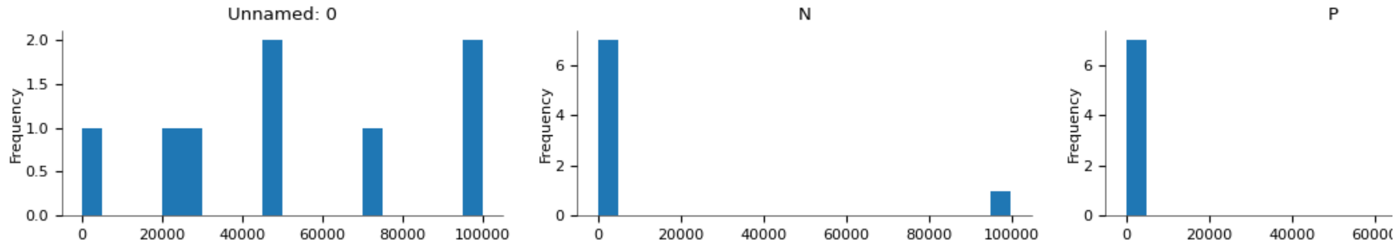
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

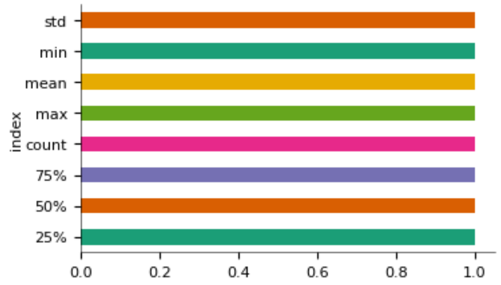
Values



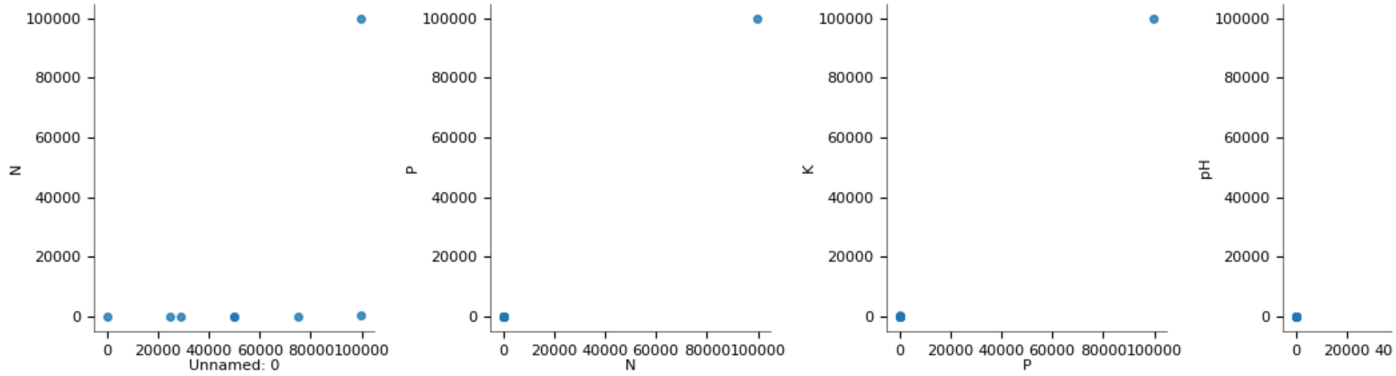
Distributions



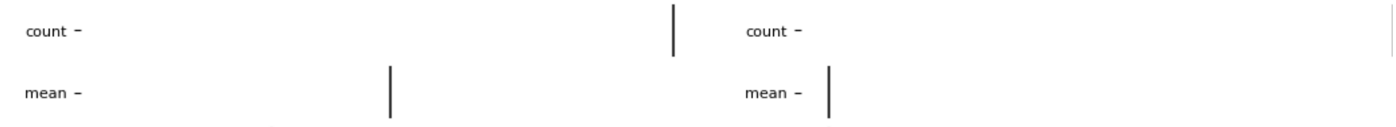
Categorical distributions

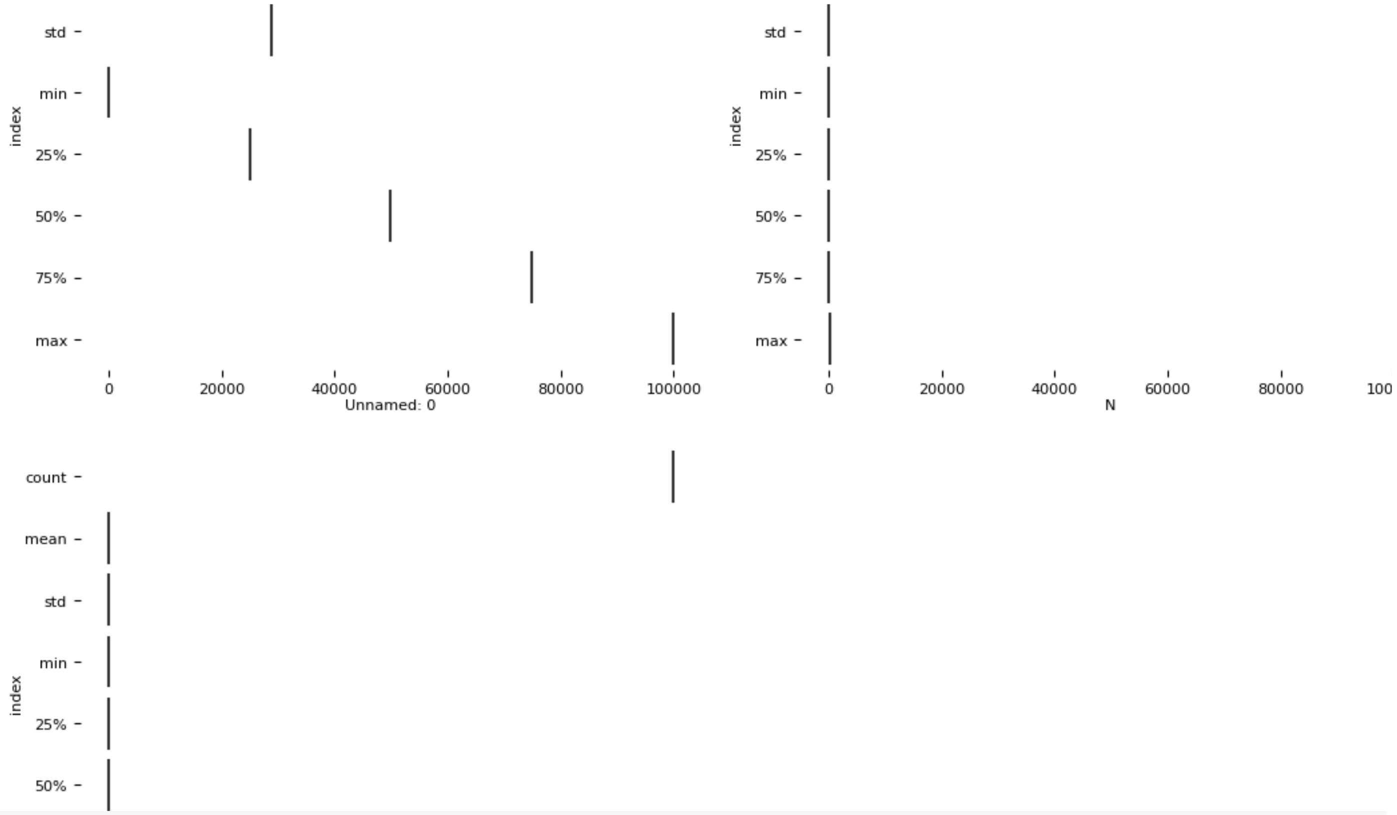


2-d distributions



Faceted distributions





```
#This is a square DataFrame where each cell contains the correlation coefficient between two columns
data.corr()
```

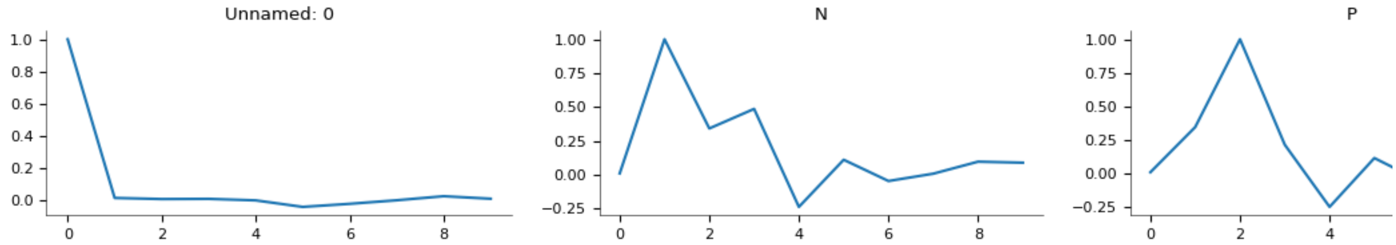
```
<ipython-input-25-34400226bd36>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
  data.corr()
```

index	Unnamed: 0	N	P	K	pH	rainfa
Unnamed: 0	1.0	0.011617835252197839	0.004737001136601322	0.005913171875390982	-0.0034184834244922937	-0.044061321
N	0.011617835252197839	1.0	0.34252121767707605	0.48665034117593303	-0.23543679940310416	0.1119003
P	0.004737001136601322	0.34252121767707605	1.0	0.2103491823801325	-0.2547760999622341	0.1106863
K	0.005913171875390982	0.48665034117593303	0.2103491823801325	1.0	-0.24670848196499137	0.3691730
pH	-0.0034184834244922937	-0.23543679940310416	-0.2547760999622341	-0.24670848196499137	1.0	-0.021997402
rainfall	-0.044061321895436587	0.11190034413008412	0.11068634876320008	0.3691730501864741	-0.021997402678227956	
temperature	-0.02492186752427526	-0.044754647171951316	-0.056987582535855760	-0.07860617255874301	0.01220263385845147	0.03447891
Area_in_hectares	-0.0027550568849649968	0.009286557435104452	-0.05751426914280822	-0.12038827234221887	0.06013557688634771	-0.13526111
Production_in_tons	0.02250865662629071	0.09788833146675463	-0.010697861601572543	-0.026378692315398695	0.07365417145109862	-0.0979477
Yield_ton_per_hec	0.006756084351576290	0.09022286552635281	0.07680552857027549	0.07619791335773496	0.006427378581637349	0.02649170

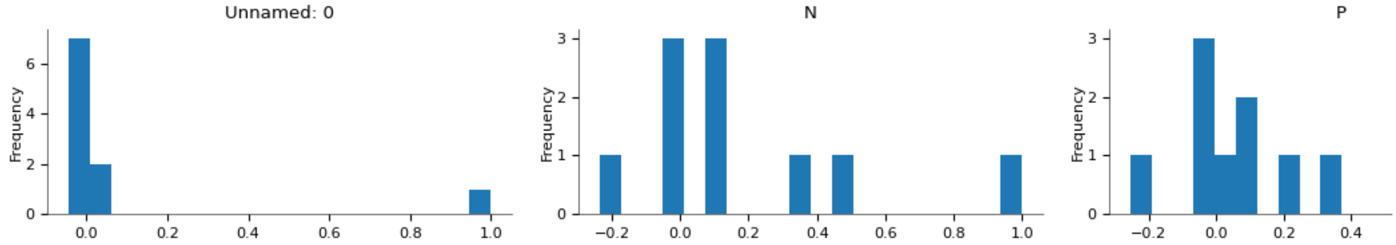
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Values

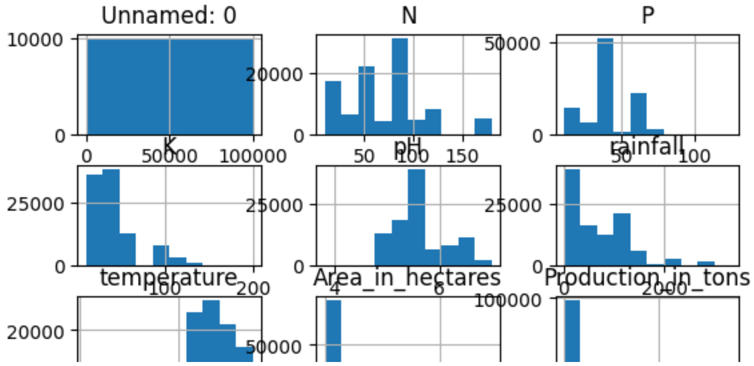


Distributions



```
# This is used to create histograms for all the numeric columns
data.hist()
```

```
array([[<Axes: title={'center': 'Unnamed: 0'}>,
        <Axes: title={'center': 'N'}>, <Axes: title={'center': 'P'}>],
       [<Axes: title={'center': 'K'}>, <Axes: title={'center': 'pH'}>,
        <Axes: title={'center': 'rainfall'}>],
       [<Axes: title={'center': 'temperature'}>,
        <Axes: title={'center': 'Area_in_hectares'}>,
        <Axes: title={'center': 'Production_in_tons'}>],
       [<Axes: title={'center': 'Yield_ton_per_hec'}>, <Axes: >,
        <Axes: >]], dtype=object)
```


Unnamed: 0
N
P
K
pH
rainfall
temperature
Area_in_hectares
Production_in_tons