j9pyy65lr

August 4, 2023

1 20104169 - SUMESH R

2 Importing Libraries

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```
from google.colab import drive
drive.mount('/content/drive')
df=pd.read_csv("/content/drive/MyDrive/mydatasets/rainfall/rainfall_east uttar␣
 ↪pradesh.csv")
df
```

```
Mounted at /content/drive
```

[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	\
0	1012	EAST UTTAR PRADESH	1901	62.6	31.3	8.2	1.1	13.6	21.8	
1	1013	EAST UTTAR PRADESH	1902	6.1	2.3	2.4	2.0	21.4	32.5	
2	1014	EAST UTTAR PRADESH	1903	8.2	0.4	1.3	0.7	15.3	71.6	
3	1015	EAST UTTAR PRADESH	1904	7.3	1.5	8.3	0.4	28.7	148.0	
4	1016	EAST UTTAR PRADESH	1905	16.8	23.6	20.0	5.4	15.4	17.3	
..	…	…	…	…	…	…	…	…	…	
110	1122	EAST UTTAR PRADESH	2011	1.0	2.7	1.6	2.9	32.2	163.8	
111	1123	EAST UTTAR PRADESH	2012	20.3	1.2	3.4	2.8	0.2	18.5	
112	1124	EAST UTTAR PRADESH	2013	6.1	59.6	2.7	8.7	1.1	309.7	
113	1125	EAST UTTAR PRADESH	2014	47.4	25.8	15.4	1.7	10.7	47.8	
114	1126	EAST UTTAR PRADESH	2015	30.0	4.1	48.2	23.2	8.6	95.3	

	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	\
0	226.5	285.6	215.4	4.9	0.1	2.1	873.2	93.9	22.9	
1	411.5	155.4	257.2	13.2	1.2	0.0	905.2	8.3	25.9	
2	115.3	420.2	258.7	324.7	0.0	0.0	1216.4	8.6	17.3	
3	359.4	328.8	95.0	50.6	17.0	26.3	1071.2	8.8	37.4	
4	302.4	316.2	169.5	3.3	0.0	1.6	891.6	40.5	40.9	
..	…	…	…	…	…	…	…	…	…	

```
110  197.9  232.1  146.4    0.6   0.0   0.0   781.2      3.7     36.7
111  234.2  156.0  164.4    0.7   0.3   0.7   602.7     21.5      6.4
112  230.0  246.1   78.2   97.4   0.5   1.1  1041.4     65.8     12.6
113  224.5  138.1  106.7   74.7   0.0   8.4   701.2     73.3     27.7
114  179.0  175.8   21.9   11.8   0.5   4.9   603.3     34.1     80.0

     Jun-Sep  Oct-Dec
0      749.3      7.1
1      856.6     14.5
2      865.8    324.7
3      931.1     93.9
4      805.4      4.9
..       …        …
110    740.2      0.6
111    573.1      1.8
112    864.0     99.0
113    517.1     83.1
114    472.0     17.2

[115 rows x 20 columns]
```

3 Data Cleaning and Data Preprocessing

[3]:
```
df=df.dropna()
```

[4]:
```
df.columns
```

[4]:
```
Index(['index', 'SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY',
       'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL', 'Jan-Feb',
       'Mar-May', 'Jun-Sep', 'Oct-Dec'],
      dtype='object')
```

[5]:
```
df.info()
```

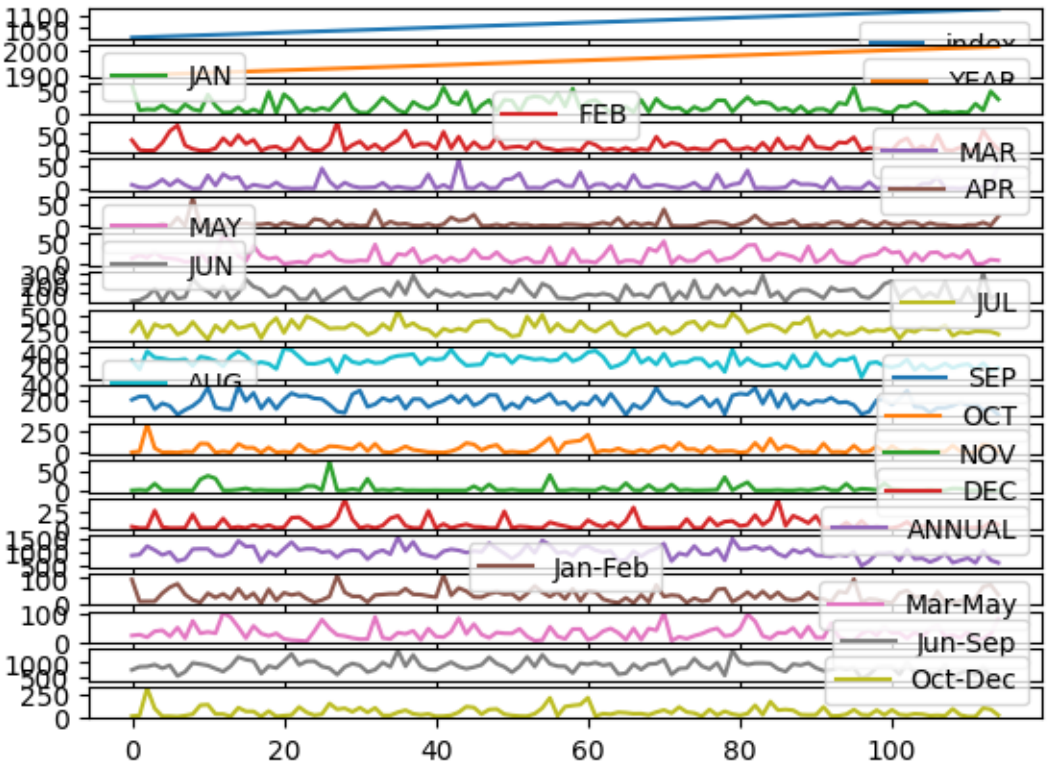
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   index        115 non-null    int64
 1   SUBDIVISION  115 non-null    object
 2   YEAR         115 non-null    int64
 3   JAN          115 non-null    float64
 4   FEB          115 non-null    float64
 5   MAR          115 non-null    float64
 6   APR          115 non-null    float64
```

```
 7   MAY          115 non-null    float64
 8   JUN          115 non-null    float64
 9   JUL          115 non-null    float64
 10  AUG          115 non-null    float64
 11  SEP          115 non-null    float64
 12  OCT          115 non-null    float64
 13  NOV          115 non-null    float64
 14  DEC          115 non-null    float64
 15  ANNUAL       115 non-null    float64
 16  Jan-Feb      115 non-null    float64
 17  Mar-May      115 non-null    float64
 18  Jun-Sep      115 non-null    float64
 19  Oct-Dec      115 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 18.1+ KB
```

4 Line chart

[6]:
```
df.plot.line(subplots=True)
```

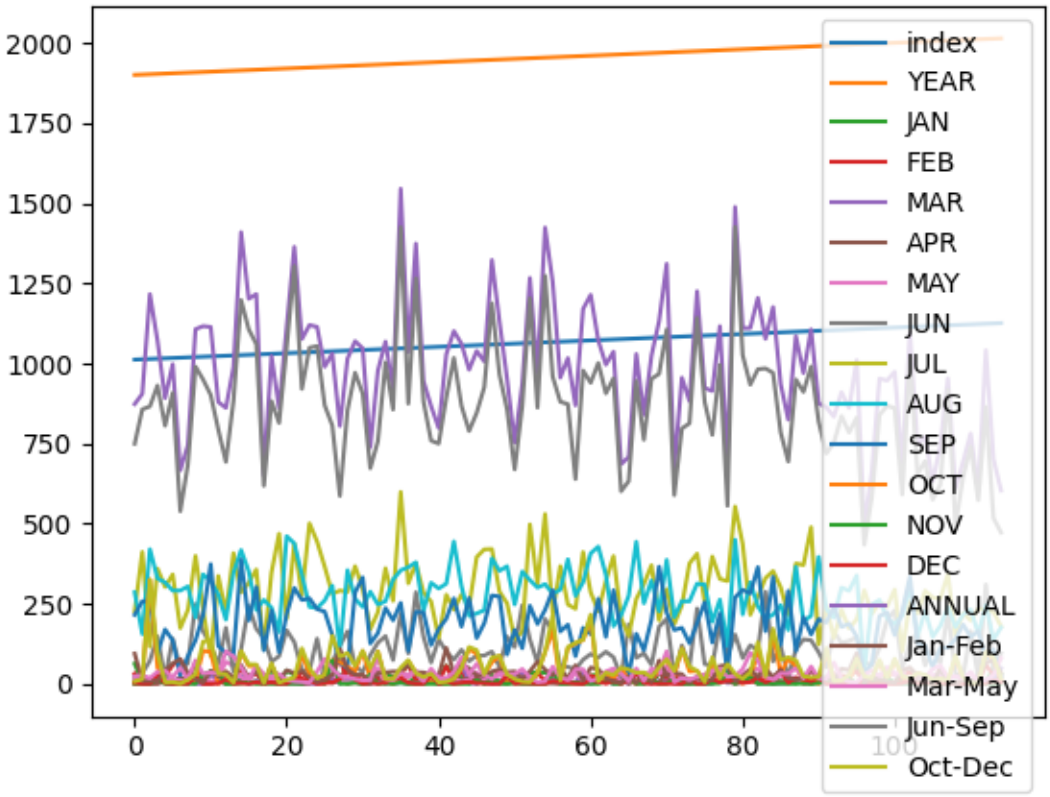
[6]:
```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >], dtype=object)
```



5 Line chart

[7]:
```
df.plot.line()
```

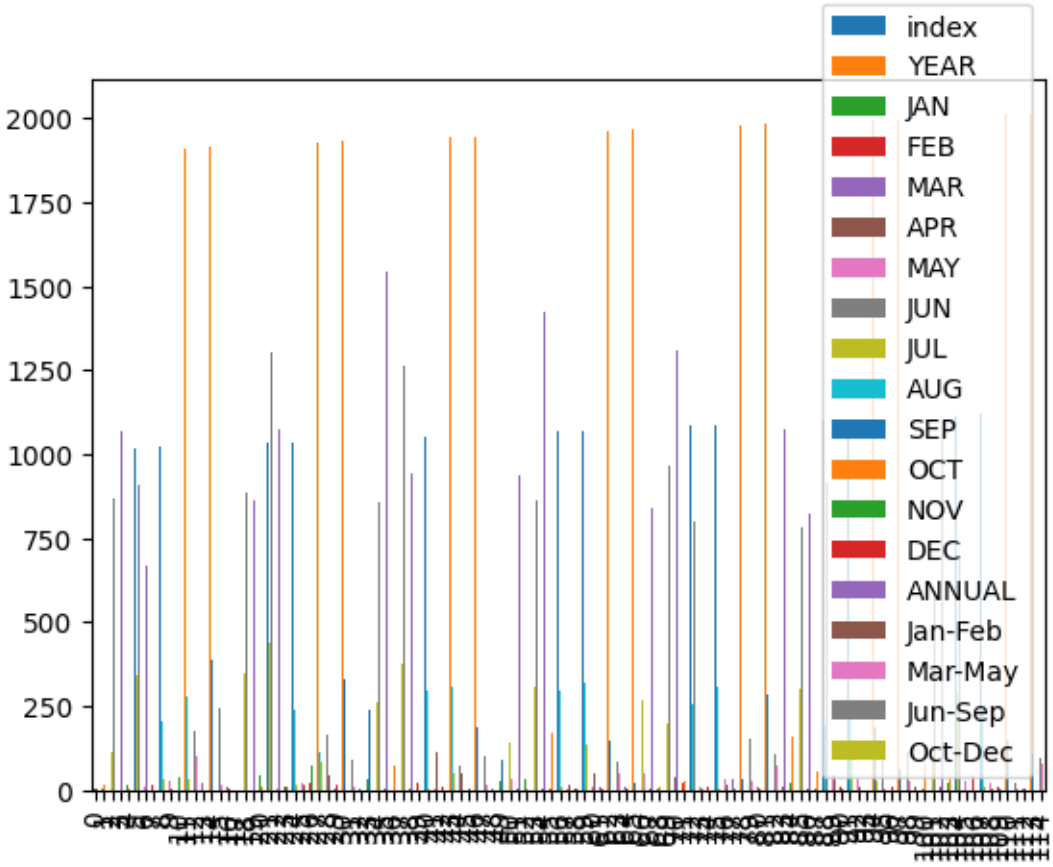
[7]: <Axes: >



6 Bar chart

[8]:
```
df.plot.bar()
```

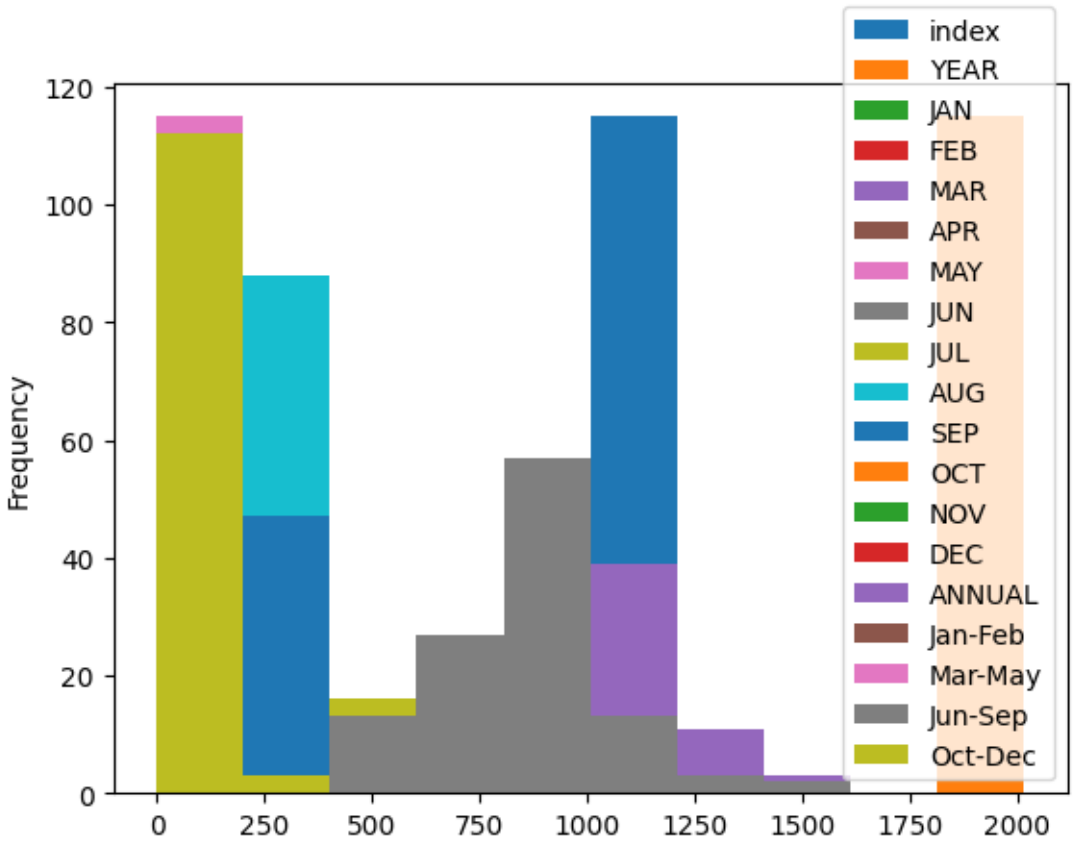
[8]: <Axes: >



7 Histogram

[9]:
```
df.plot.hist()
```

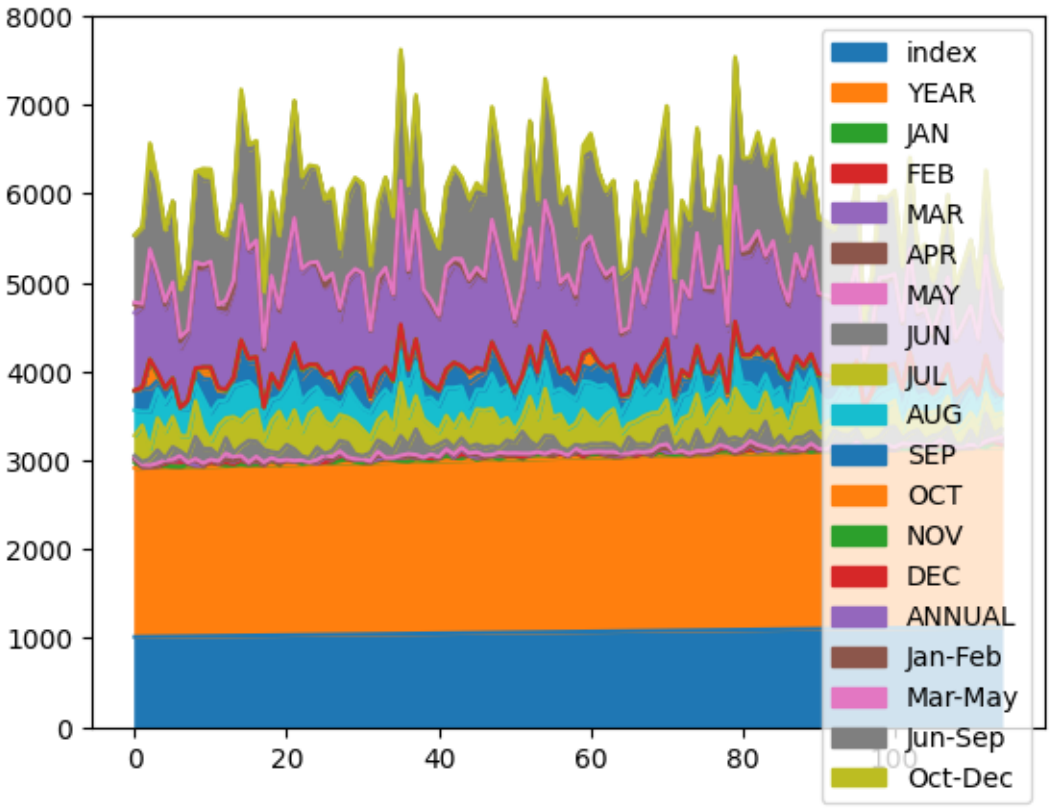
[9]: <Axes: ylabel='Frequency'>



8 Area chart

[10]:
```
df.plot.area()
```

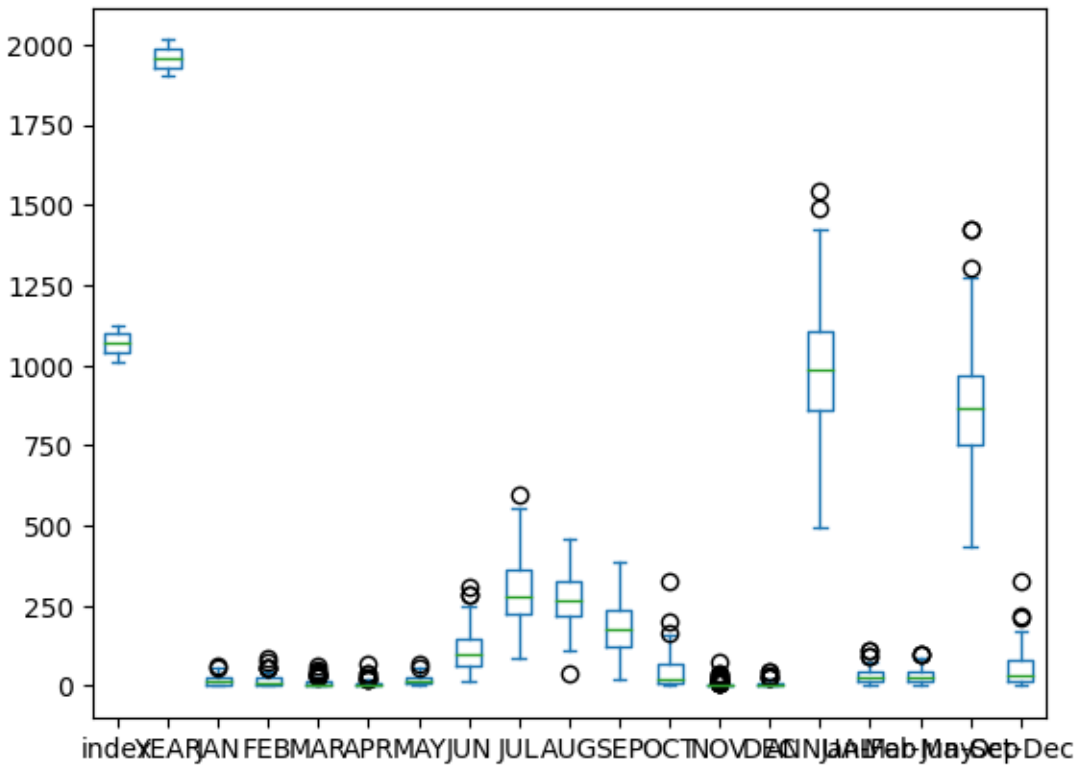
[10]: <Axes: >



9 Box chart

[11]:
```
df.plot.box()
```

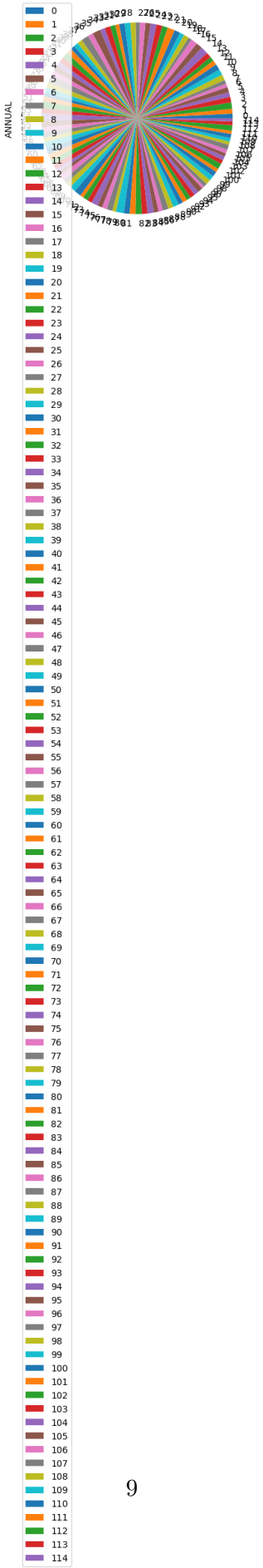
[11]: <Axes: >



10 Pie chart

[12]:
```
df.plot.pie(y='ANNUAL' )
```

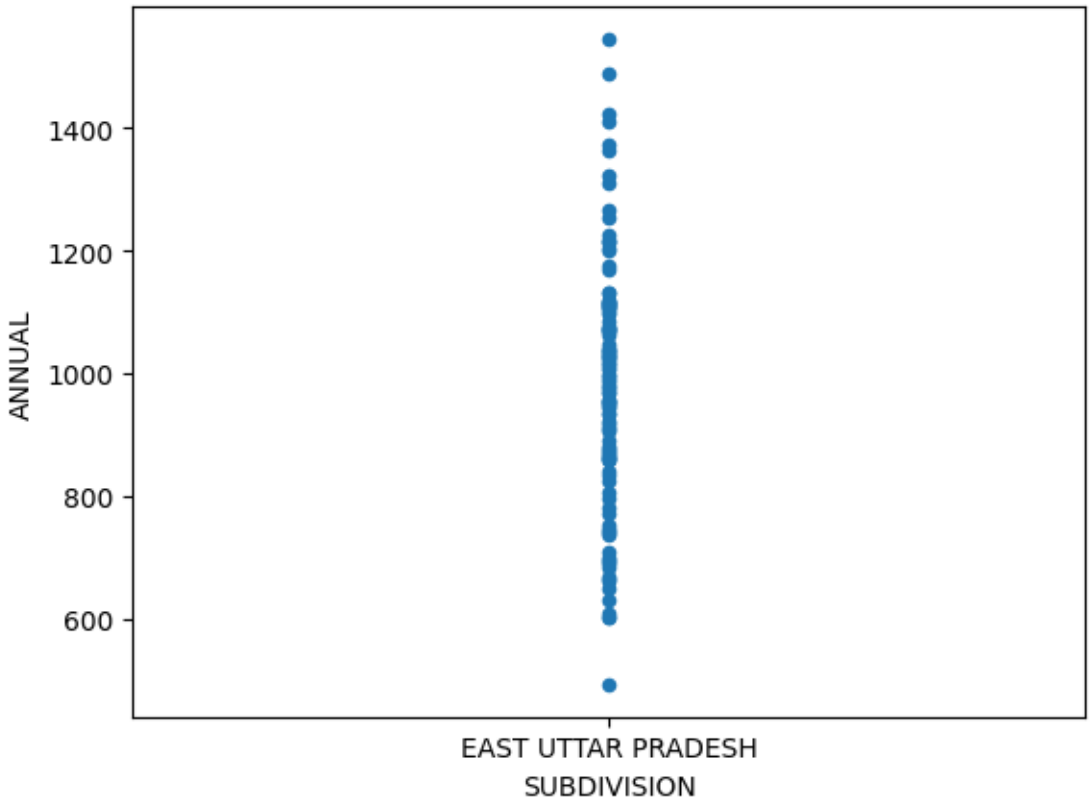
[12]:
```
<Axes: ylabel='ANNUAL'>
```


ANNUAL
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

11 Scatter chart

[13]:
```
df.plot.scatter(x='SUBDIVISION' ,y='ANNUAL')
```

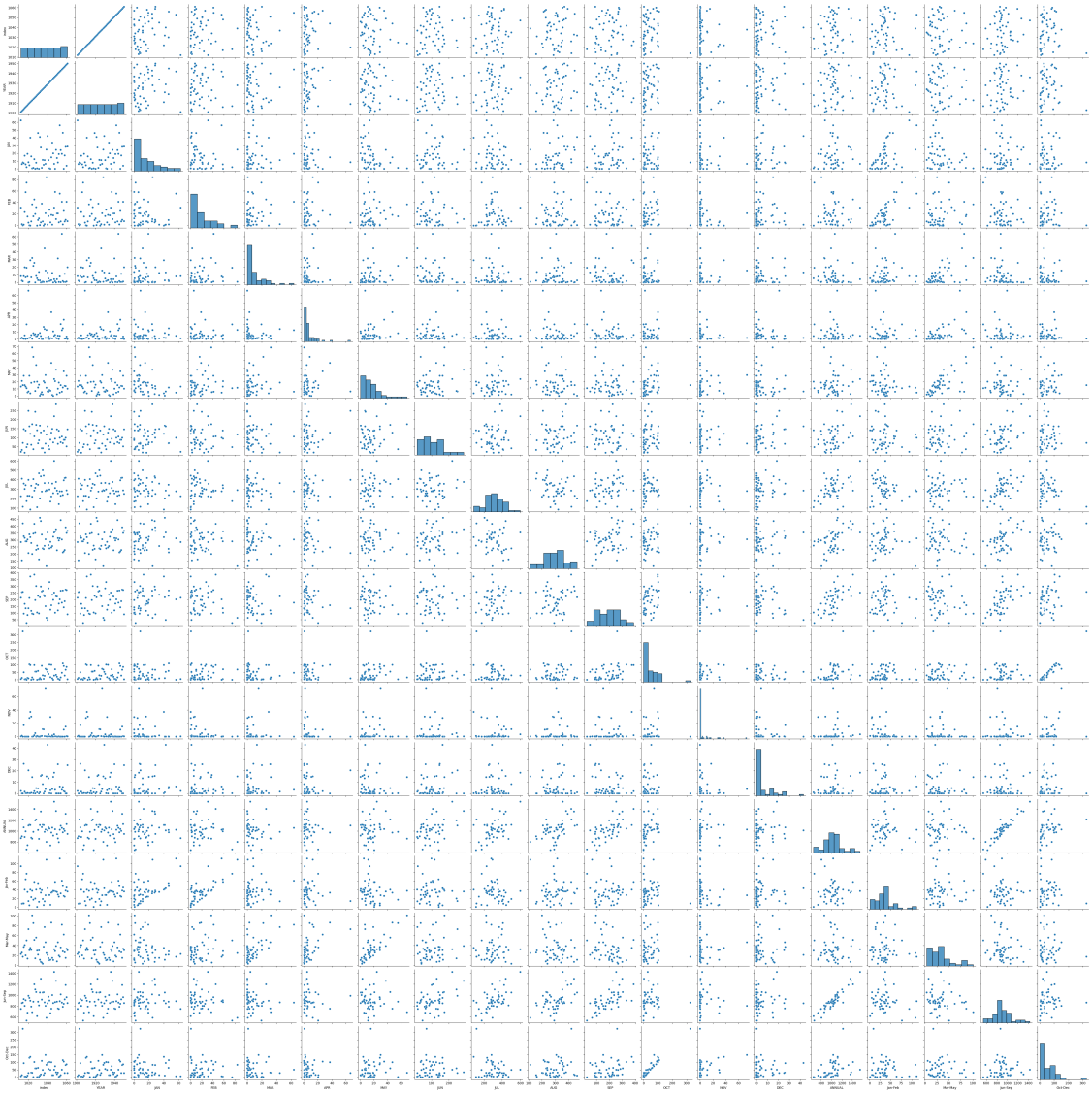
[13]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

[14]:
```
sns.pairplot(df[0:50])
```

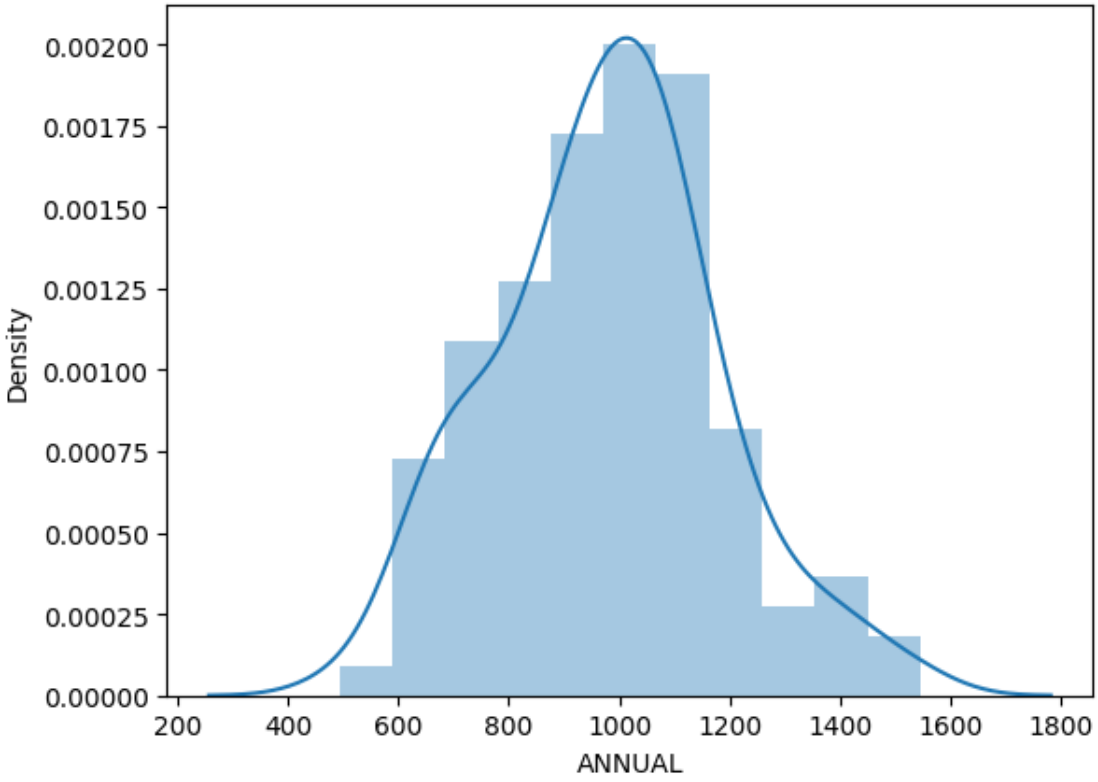
[14]:
```
<seaborn.axisgrid.PairGrid at 0x78c7f6b0a320>
```



[15]:
```
sns.distplot(df['ANNUAL'])
```

```
<ipython-input-15-5daa97052ca5>:1: 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

  sns.distplot(df['ANNUAL'])
```

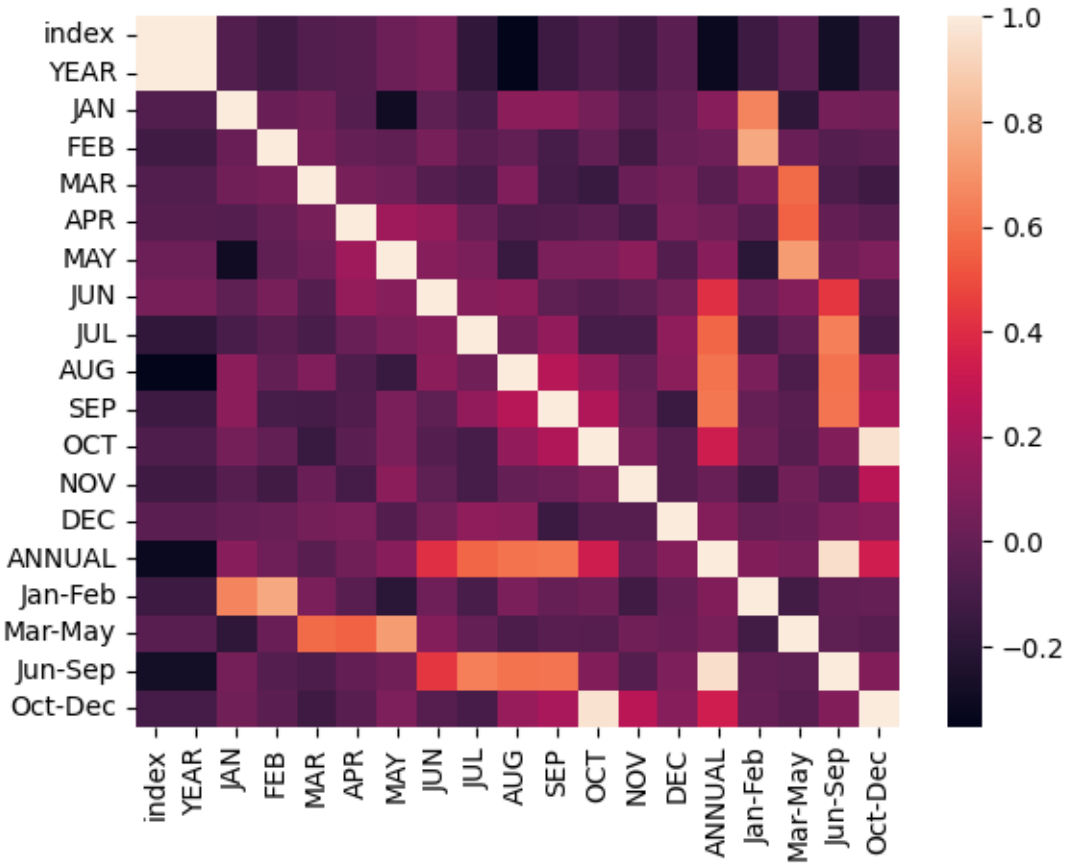
[15]: <Axes: xlabel='ANNUAL', ylabel='Density'>



[16]:
```
sns.heatmap(df.corr())
```

```
<ipython-input-16-aa4f4450a243>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(df.corr())
```

[16]: <Axes: >


index
YEAR
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
ANNUAL
Jan-Feb
Mar-May
Jun-Sep
Oct-Dec
1.0
0.8
0.6
0.4
0.2
0.0
−0.2