aw8di0txr

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_punjab.csv")
df
```

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

[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	\
0	1472	PUNJAB	1901	55.7	50.1	25.2	2.1	25.2	10.4	178.2	
1	1473	PUNJAB	1902	0.0	0.8	9.9	10.9	29.6	49.9	125.6	
2	1474	PUNJAB	1903	29.5	0.5	45.0	1.3	9.2	5.2	212.2	
3	1475	PUNJAB	1904	24.2	1.7	87.8	1.2	13.8	22.0	59.9	
4	1476	PUNJAB	1905	53.0	40.3	24.3	0.5	2.2	19.2	122.6	
..	…	…	…	…	…	…	…	…	…	…	
110	1582	PUNJAB	2011	3.5	35.6	8.2	17.8	18.9	162.9	120.9	
111	1583	PUNJAB	2012	62.6	3.2	1.9	31.1	1.6	11.9	120.2	
112	1584	PUNJAB	2013	9.3	50.1	11.6	3.4	3.6	120.3	117.9	
113	1585	PUNJAB	2014	21.8	20.1	30.3	24.5	20.8	20.6	76.3	
114	1586	PUNJAB	2015	17.7	31.3	68.5	29.8	16.7	48.3	130.2	

	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	145.0	24.4	3.7	0.0	3.3	523.5	105.9	52.5	358.1	7.0
1	94.9	67.2	9.0	0.0	0.1	398.0	0.8	50.4	337.7	9.1
2	119.1	132.5	6.9	0.0	9.5	571.0	29.9	55.5	469.1	16.4
3	124.0	73.8	7.4	9.8	25.9	451.5	25.9	102.9	279.7	43.1
4	50.3	111.1	1.2	0.0	9.4	434.3	93.3	27.0	303.2	10.7
..	…	…	…	…	…	…	…	…	…	…
110	193.5	140.2	0.0	1.0	2.6	705.2	39.2	44.9	617.5	3.6

```
111  135.1  112.3   2.2  0.4  11.0   493.6     65.8     34.7    379.6     13.6
112  217.1   24.4  16.2  6.1   6.6   586.6     59.4     18.6    479.7     28.9
113   41.9  105.8   6.0  0.7  14.1   382.7     41.9     75.5    244.6     20.7
114   88.6   69.2   9.0  0.8   0.7   510.8     49.0    115.0    336.3     10.5

[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()
```

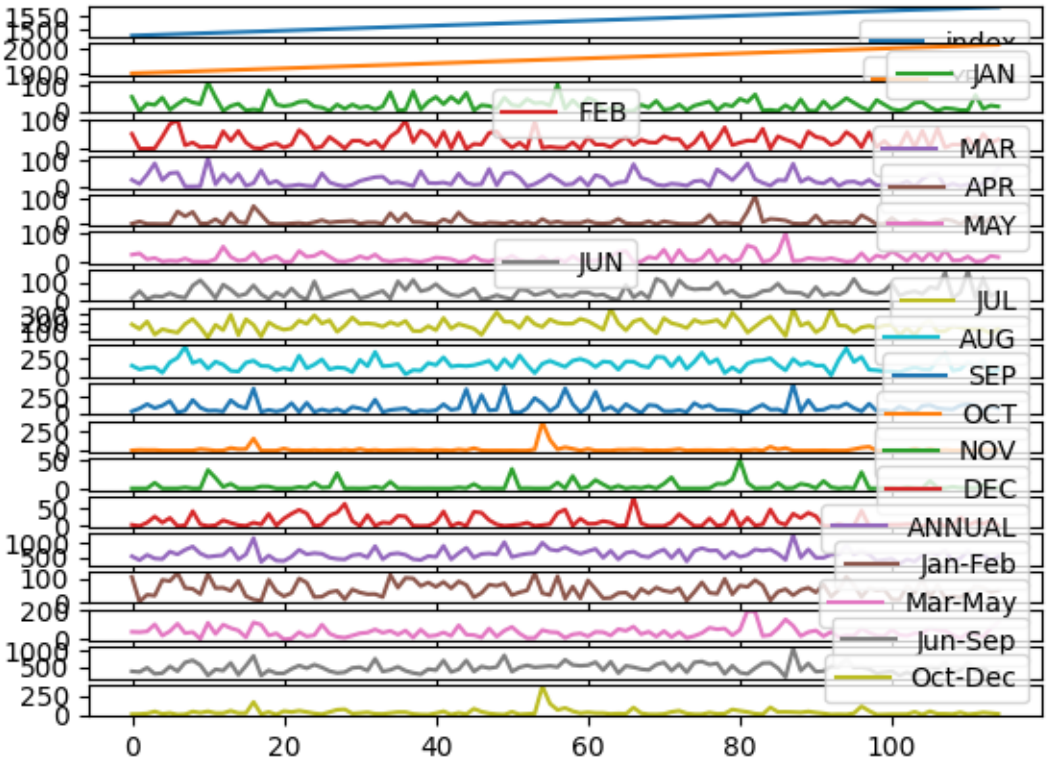
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   index        115 non-null    int64
 1   SUBDIVISION  115 non-null    object
 2   YEAR         115 non-null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

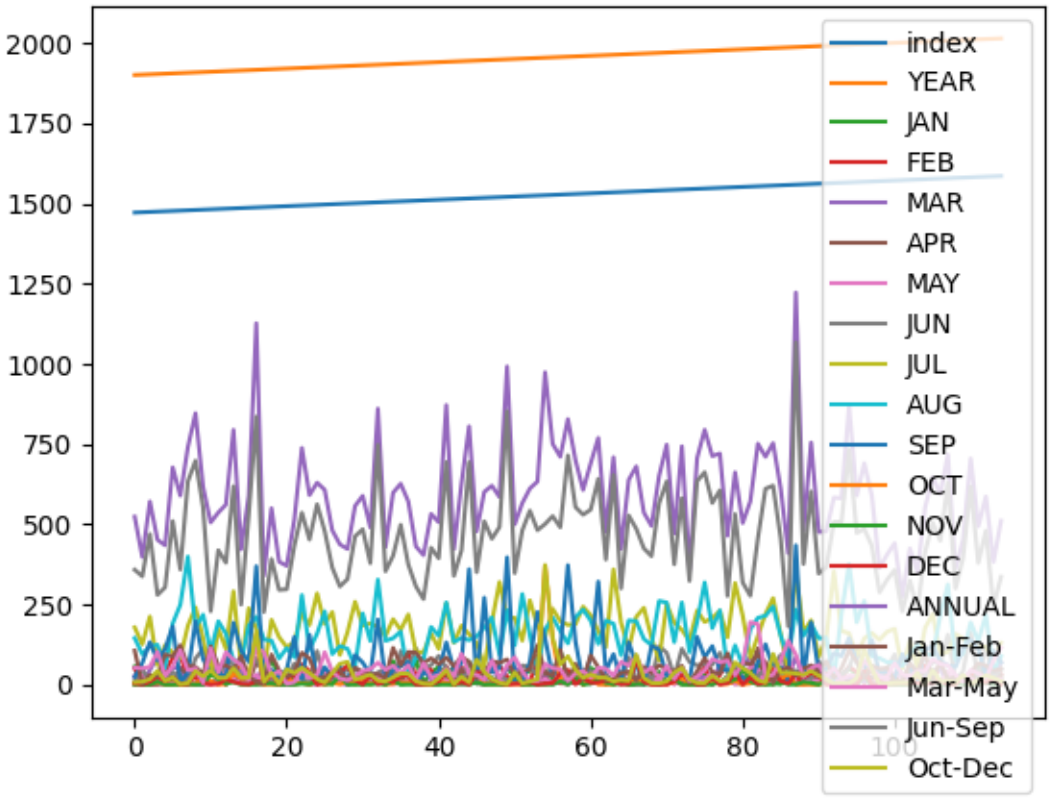
[6]:
```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >], dtype=object)
```



5 Line chart

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

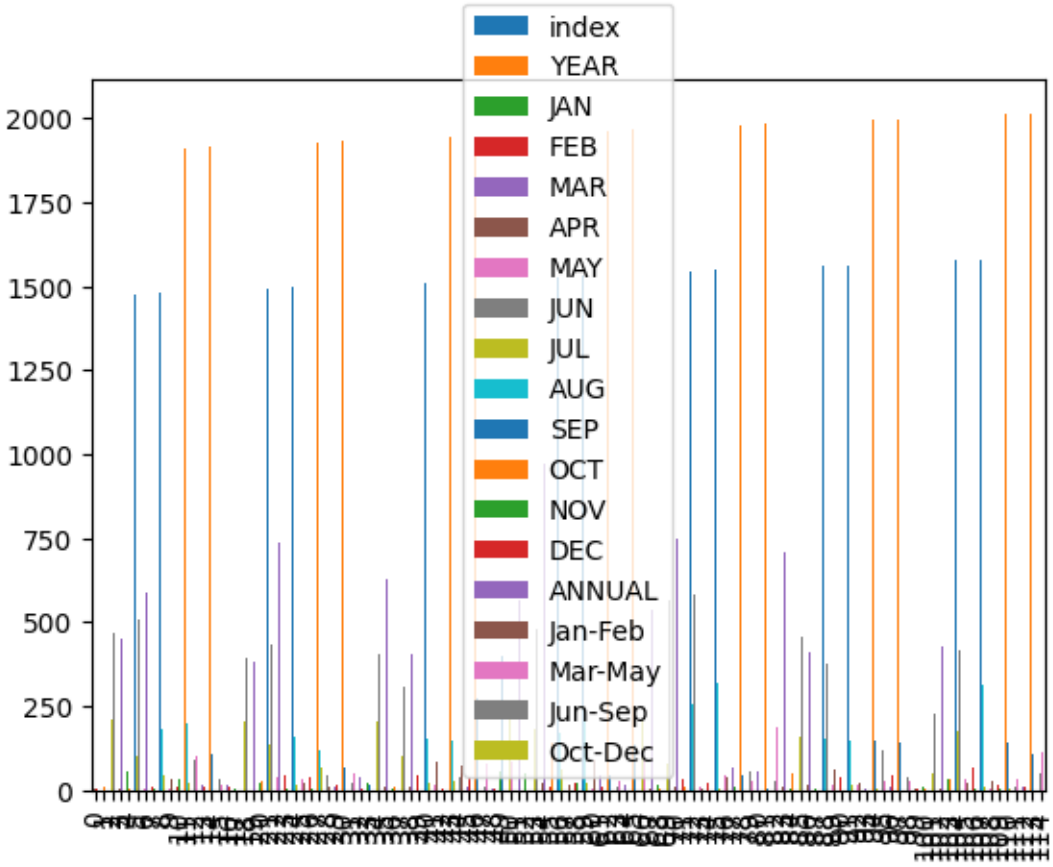
[7]:
```
<Axes: >
```



6 Bar chart

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

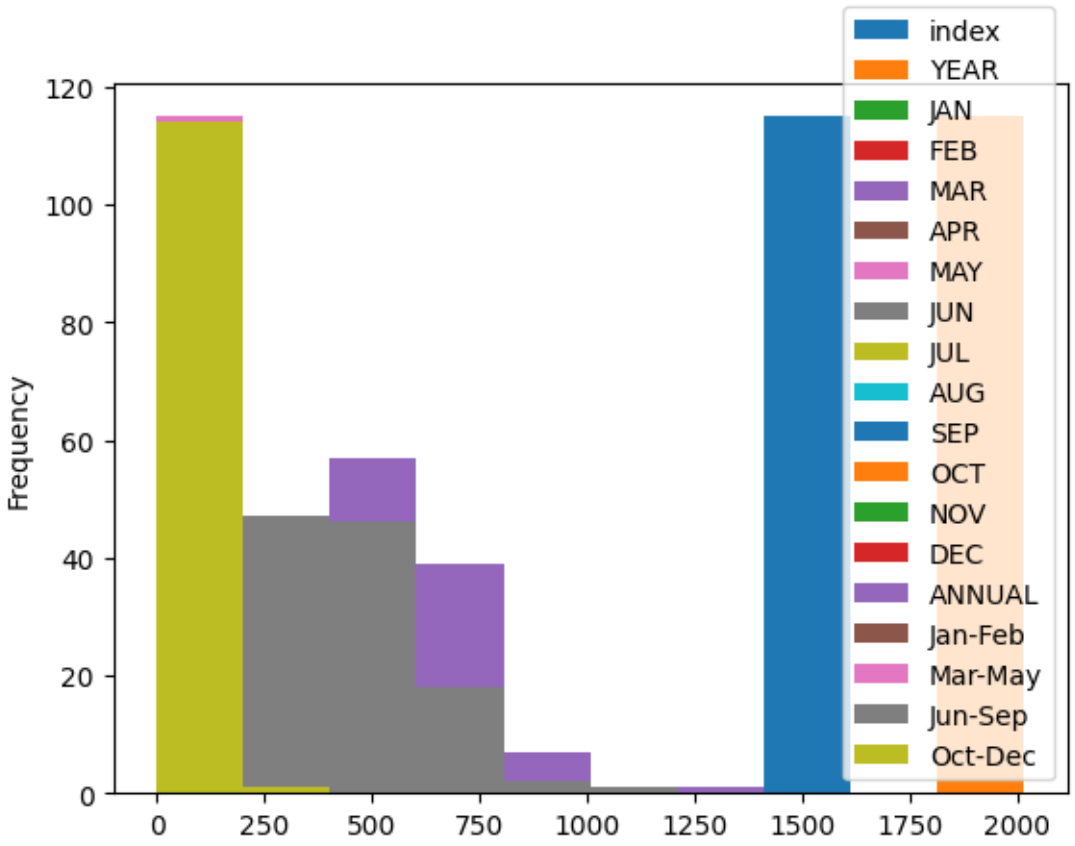
[8]: <Axes: >



7 Histogram

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

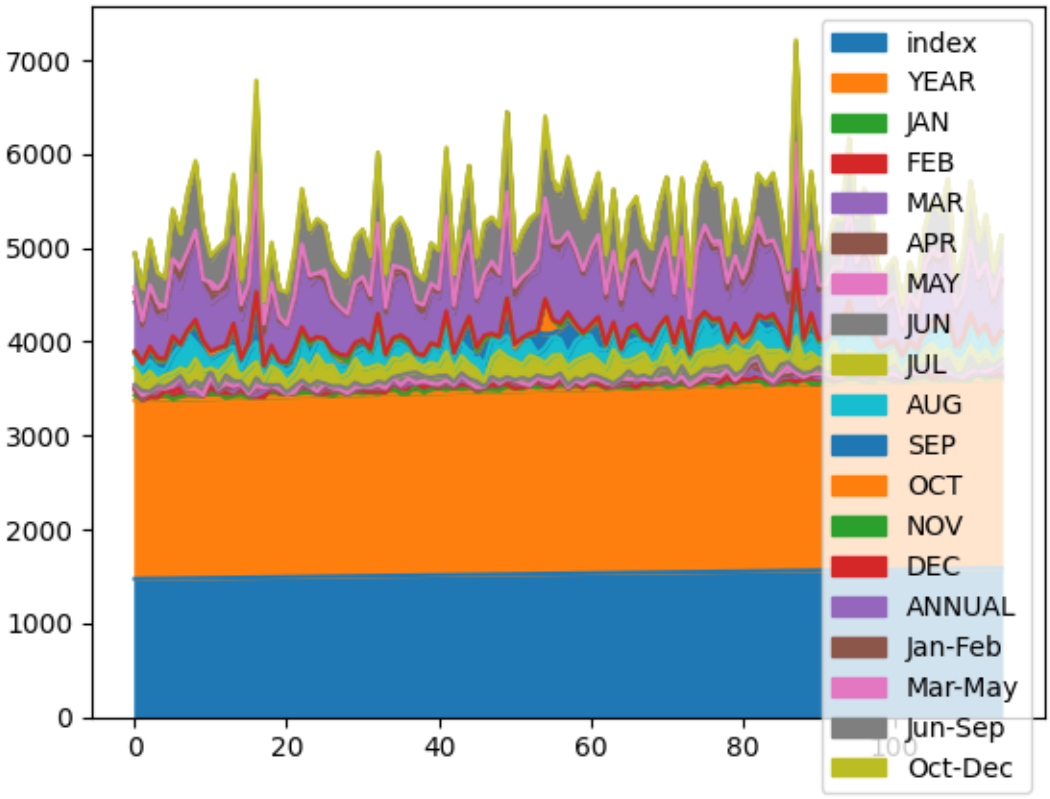
[9]:
```
<Axes: ylabel='Frequency'>
```



8 Area chart

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

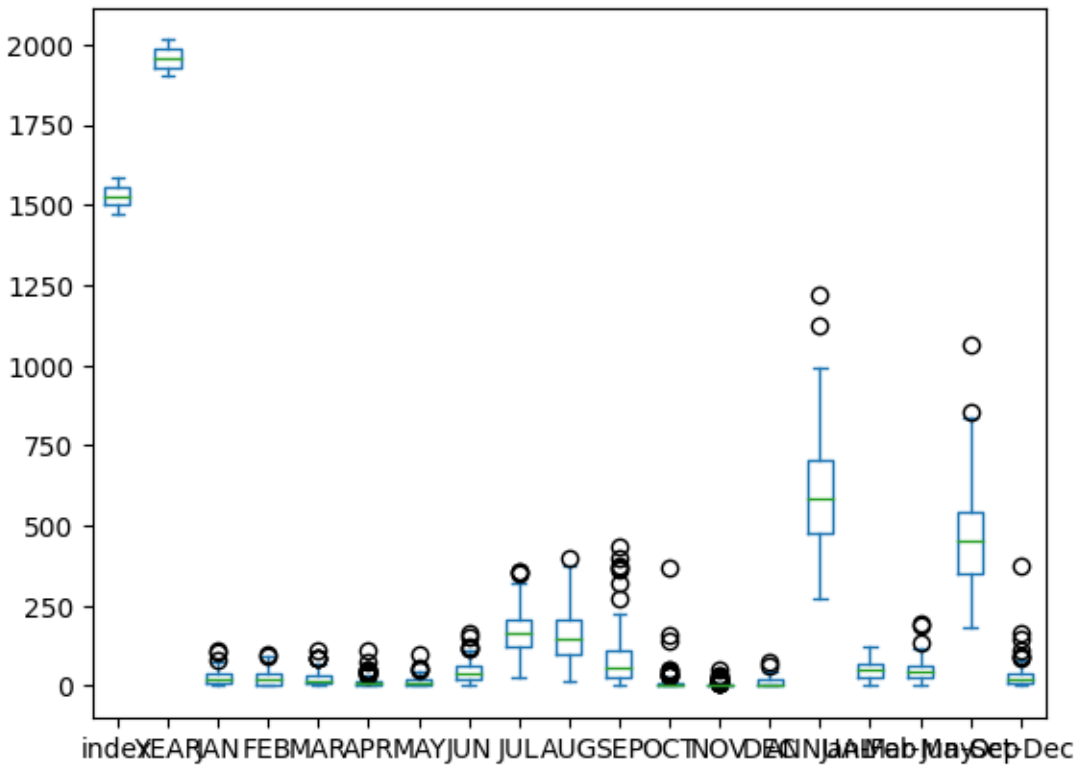
[10]: <Axes: >



9 Box chart

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

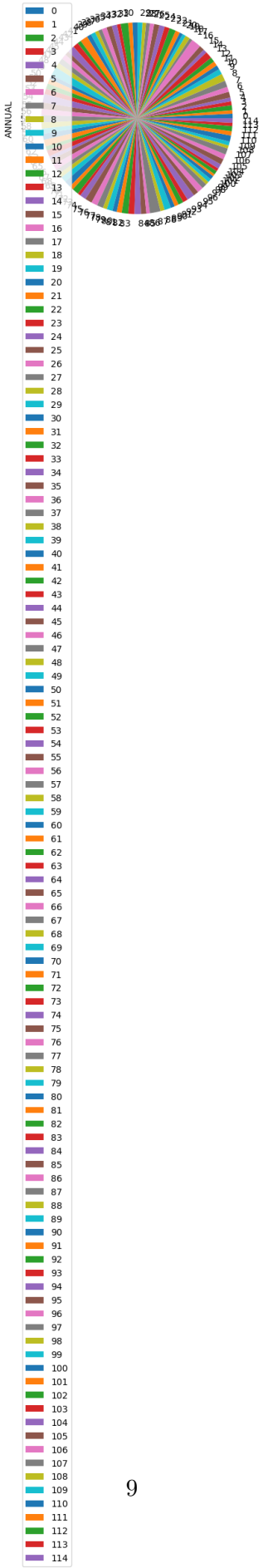
[11]: <Axes: >



10 Pie chart

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

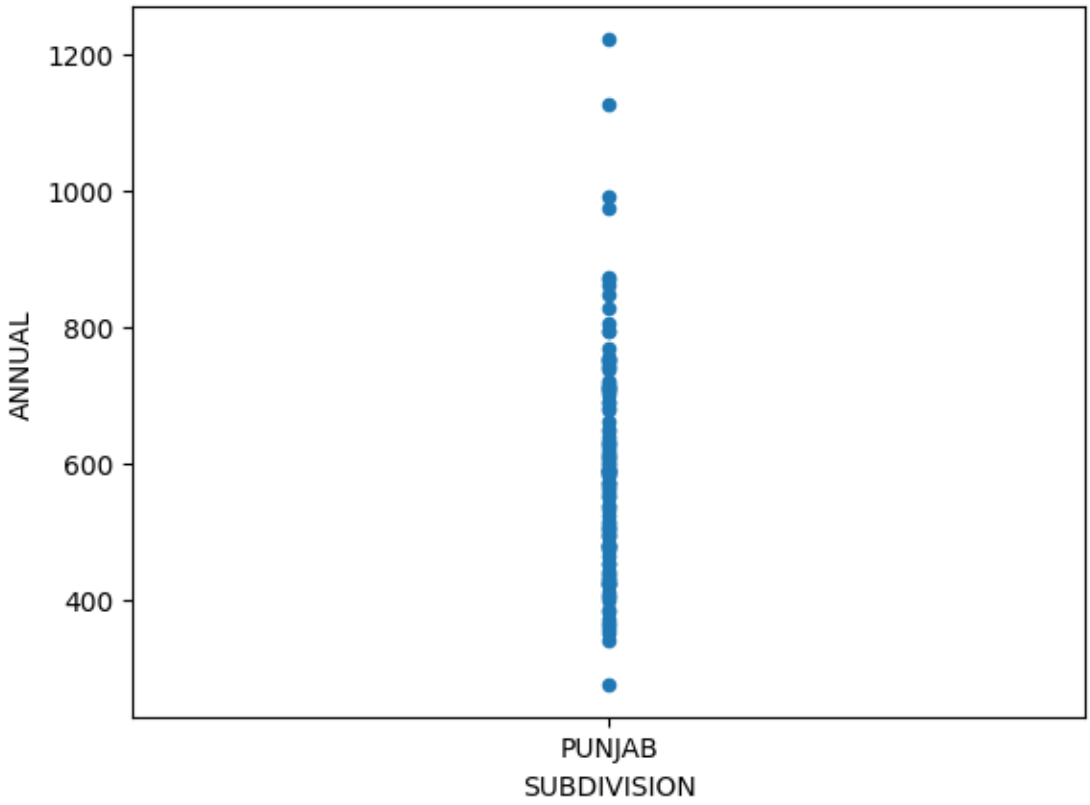
[12]:
```
<Axes: ylabel='ANNUAL'>
```


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')
```

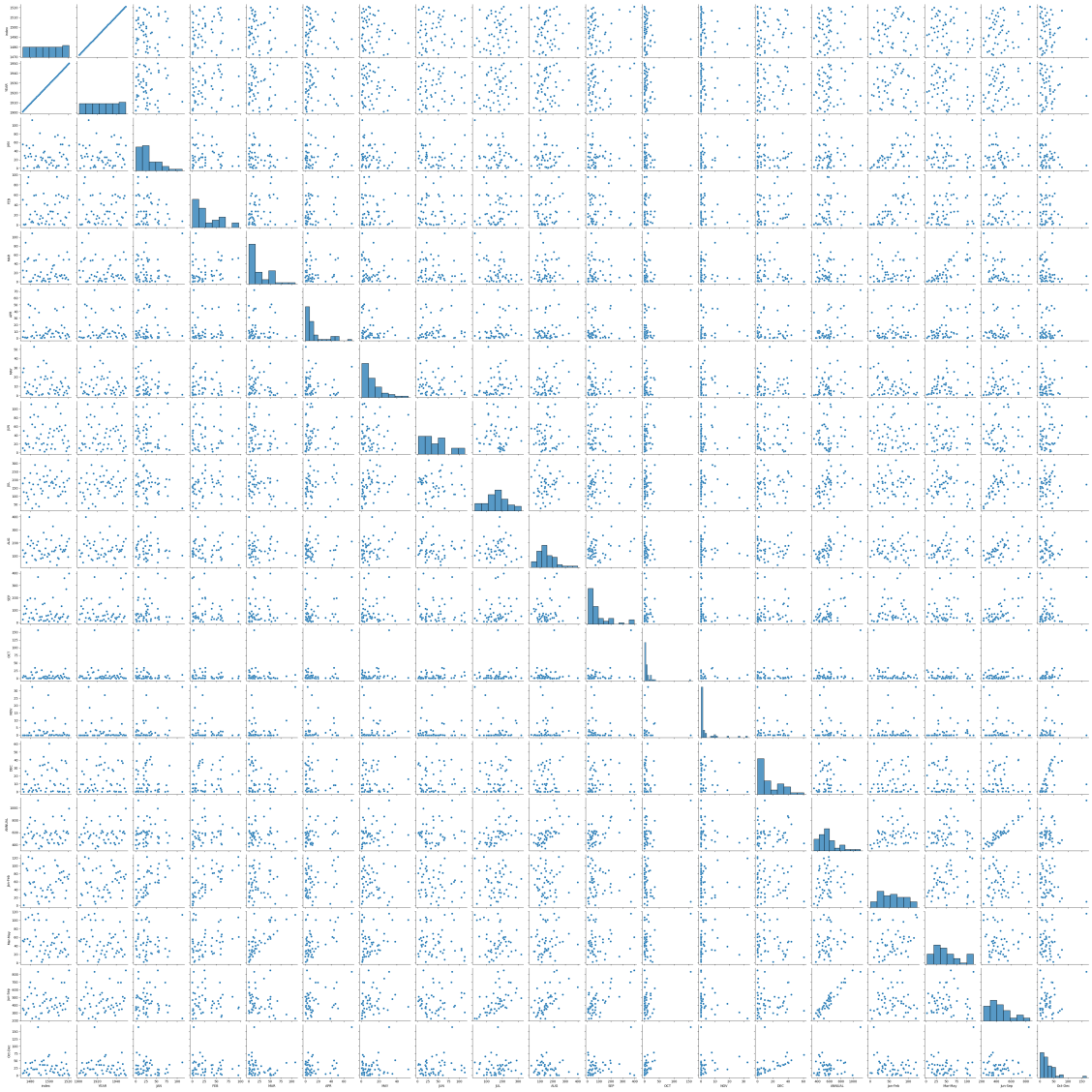
[13]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

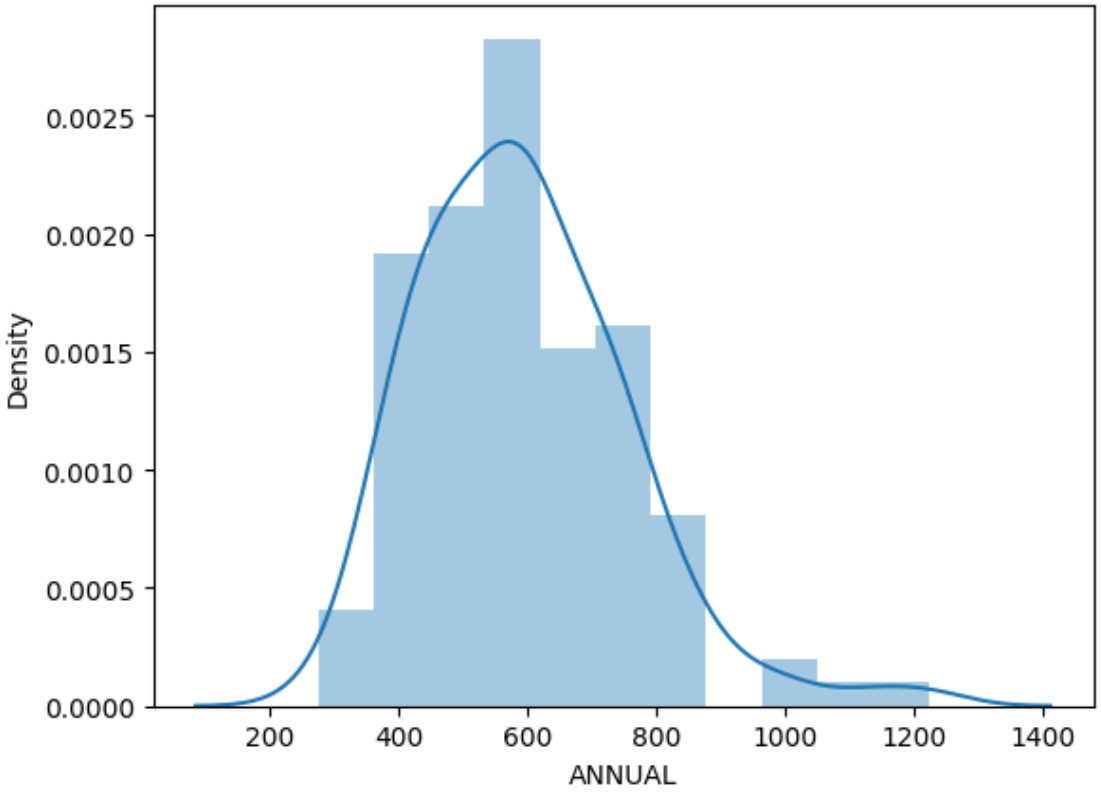
[14]:
```
<seaborn.axisgrid.PairGrid at 0x7e2359963250>
```



[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'])
```

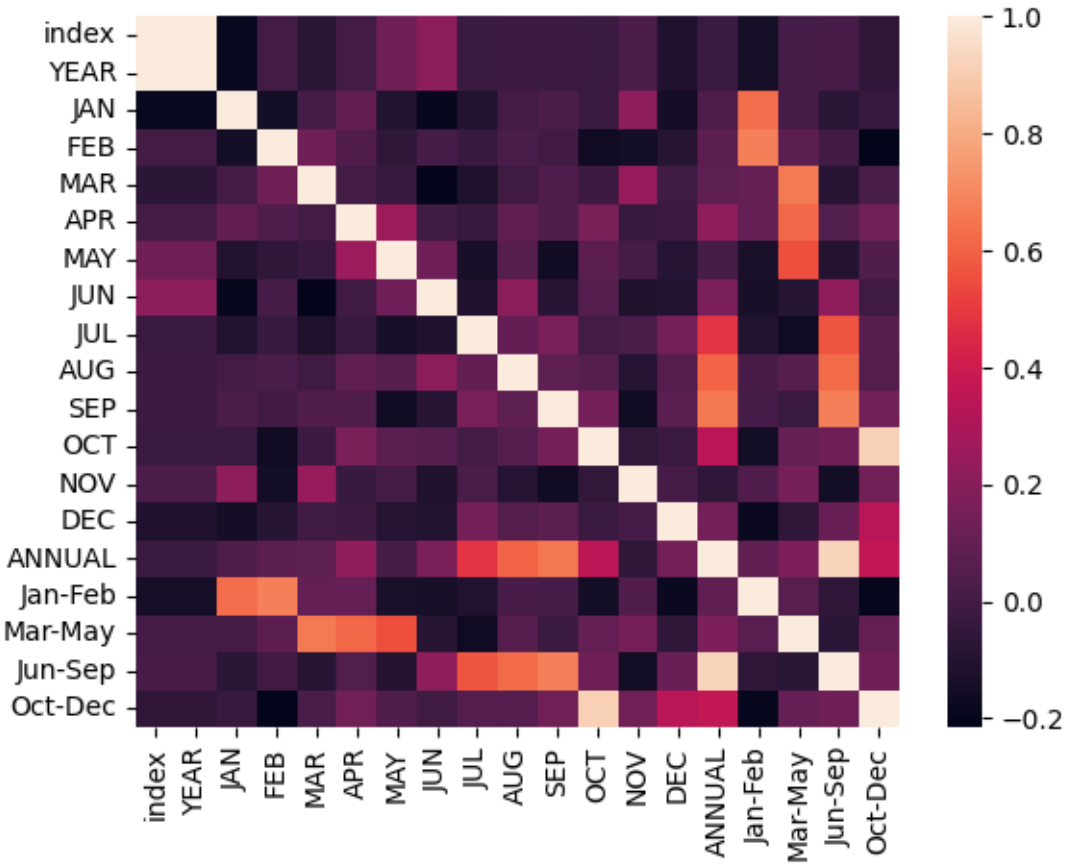
[15]: <Axes: xlabel='ANNUAL', ylabel='Density'>



[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