wcimvyi0n

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_west madhya␣
 ↪pradesh.csv")
df
```

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

[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	\
0	2047	WEST MADHYA PRADESH	1901	25.8	5.8	5.8	2.8	2.1	41.2	
1	2048	WEST MADHYA PRADESH	1902	22.1	8.4	0.0	2.0	5.9	35.9	
2	2049	WEST MADHYA PRADESH	1903	5.3	0.0	0.0	0.0	22.3	50.6	
3	2050	WEST MADHYA PRADESH	1904	3.2	15.5	14.8	0.0	12.0	96.6	
4	2051	WEST MADHYA PRADESH	1905	3.5	4.4	1.1	0.8	3.0	36.1	
..	…	…	…	…	…	…	…	…	…	
110	2157	WEST MADHYA PRADESH	2011	0.0	1.7	0.1	1.8	3.6	241.5	
111	2158	WEST MADHYA PRADESH	2012	6.2	0.0	0.0	0.9	3.1	48.2	
112	2159	WEST MADHYA PRADESH	2013	1.7	31.1	8.5	2.8	0.4	263.7	
113	2160	WEST MADHYA PRADESH	2014	25.6	34.4	4.6	1.4	1.4	30.6	
114	2161	WEST MADHYA PRADESH	2015	40.2	6.4	53.5	13.3	2.0	154.1	

	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	\
0	228.9	349.9	47.9	5.6	0.0	2.4	718.2	31.6	10.7	667.9	
1	401.9	179.4	194.1	37.9	10.0	14.2	911.7	30.5	8.0	811.2	
2	304.9	261.1	250.2	55.1	0.0	0.0	949.6	5.3	22.3	866.8	
3	273.0	218.6	125.9	3.3	1.8	9.6	774.4	18.7	26.9	714.1	
4	326.3	137.6	183.5	0.3	0.0	0.0	696.5	7.9	4.9	683.5	
..	…	…	…	…	…	…	…	…	…	…	

```
110  306.7  343.3  165.0   0.2   0.0   0.0  1063.9      1.7      5.5   1056.5
111  439.2  341.2  194.3   2.1   0.0   0.0  1035.2      6.2      4.0   1023.0
112  485.1  432.6   98.9  68.7   0.3   2.4  1396.3     32.8     11.7   1280.4
113  337.4  211.0  192.6   7.0   3.0  15.8   864.9     60.0      7.5    771.6
114  428.2  276.6   55.6  11.0   0.3   1.0  1042.3     46.6     68.9    914.5
```

```
     Oct-Dec
0        7.9
1       62.0
2       55.1
3       14.7
4        0.3
..       …
110      0.2
111      2.1
112     71.4
113     25.8
114     12.4

[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()
```

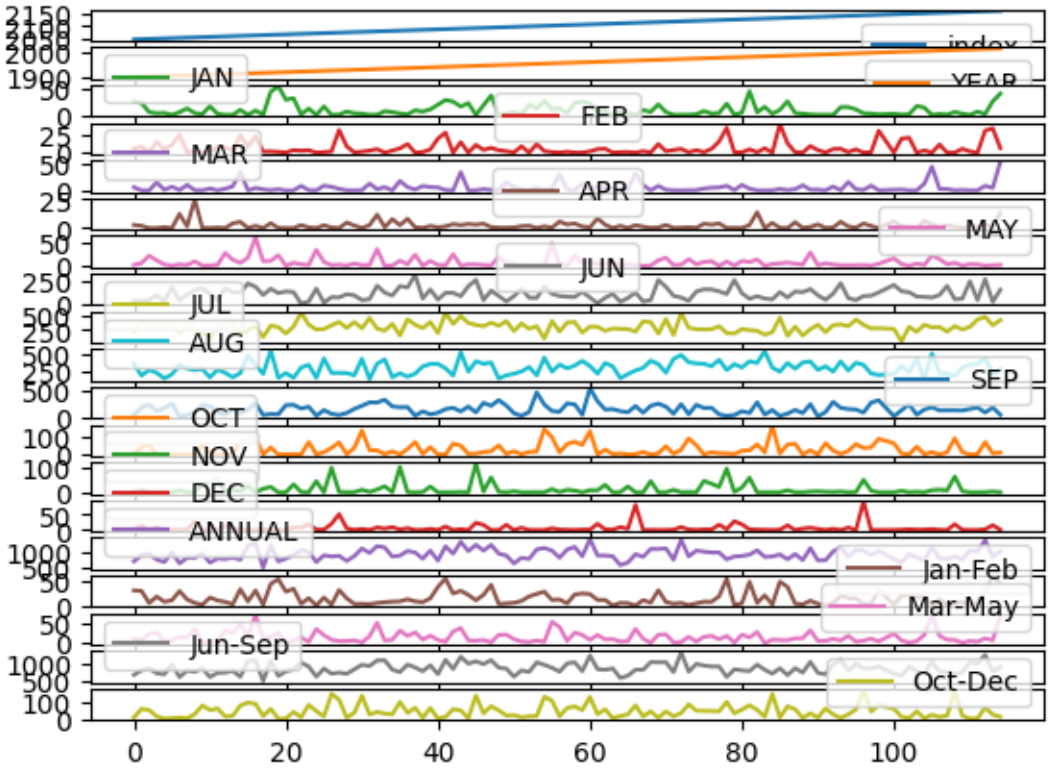
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 114 entries, 0 to 114
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   index        114 non-null    int64
 1   SUBDIVISION  114 non-null    object
 2   YEAR         114 non-null    int64
 3   JAN          114 non-null    float64
 4   FEB          114 non-null    float64
 5   MAR          114 non-null    float64
 6   APR          114 non-null    float64
```

```
 7   MAY          114 non-null    float64
 8   JUN          114 non-null    float64
 9   JUL          114 non-null    float64
 10  AUG          114 non-null    float64
 11  SEP          114 non-null    float64
 12  OCT          114 non-null    float64
 13  NOV          114 non-null    float64
 14  DEC          114 non-null    float64
 15  ANNUAL       114 non-null    float64
 16  Jan-Feb      114 non-null    float64
 17  Mar-May      114 non-null    float64
 18  Jun-Sep      114 non-null    float64
 19  Oct-Dec      114 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 18.7+ KB
```

4 Line chart

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

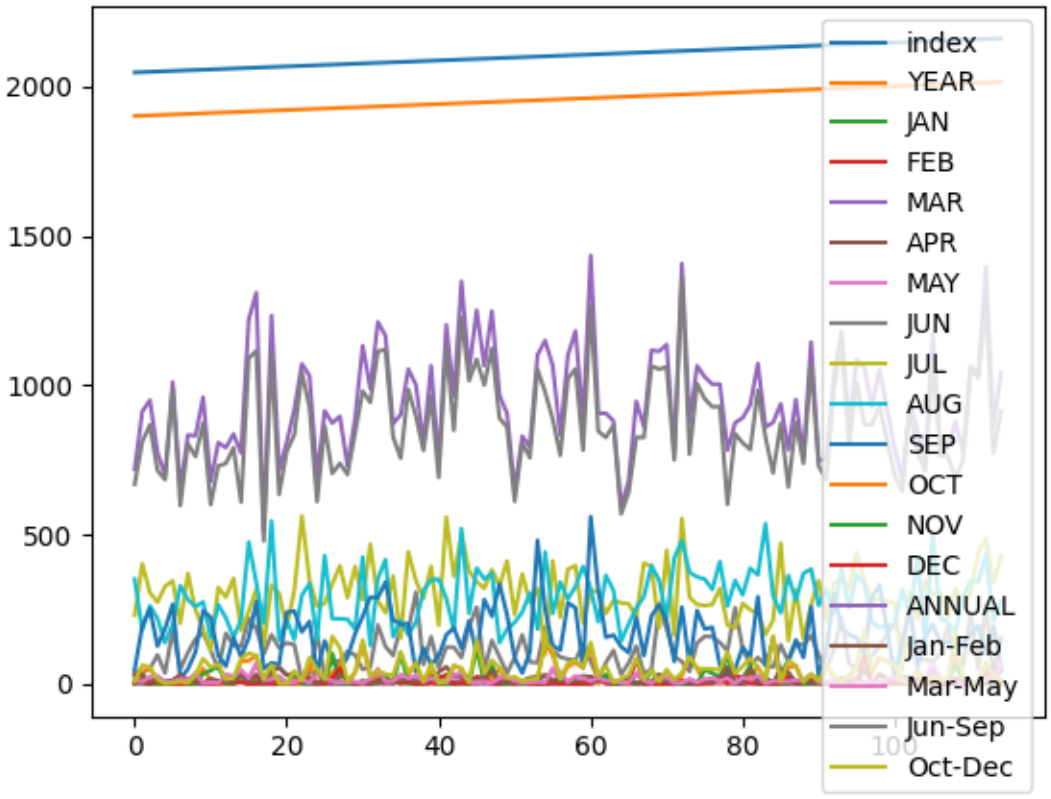
[6]:
```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >], dtype=object)
```



5 Line chart

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

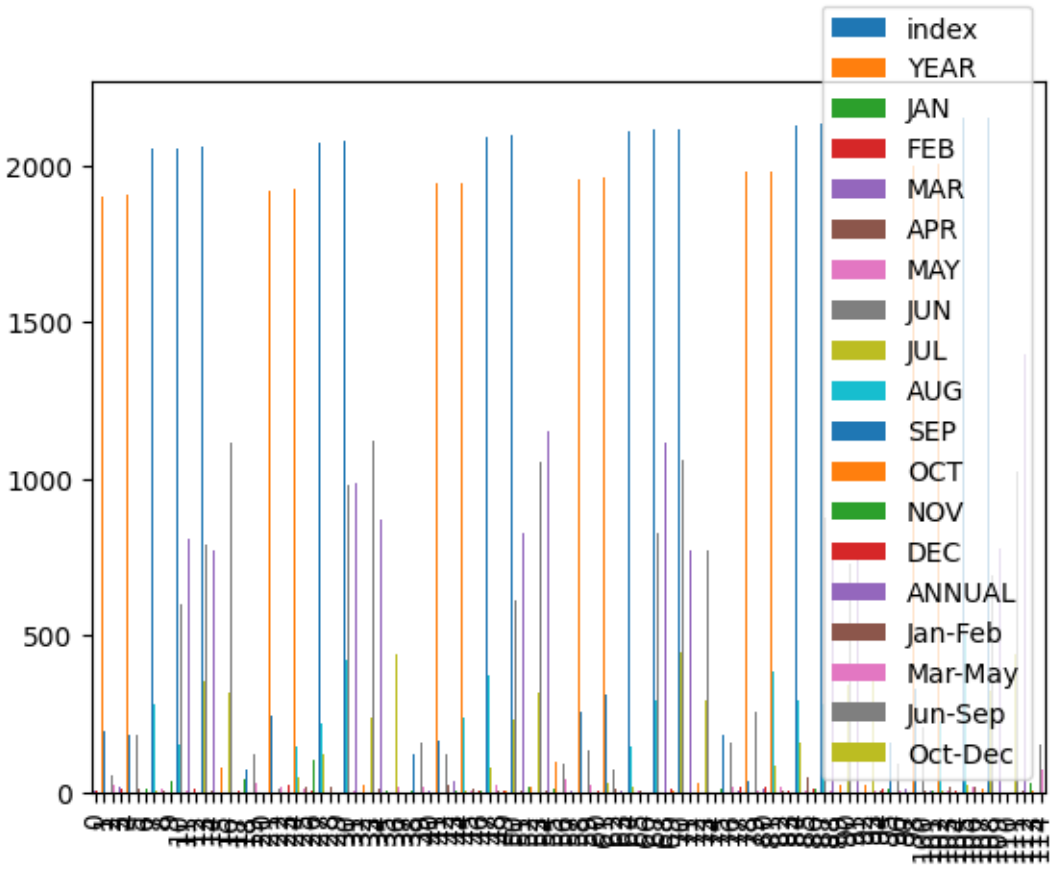
[7]: <Axes: >



6 Bar chart

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

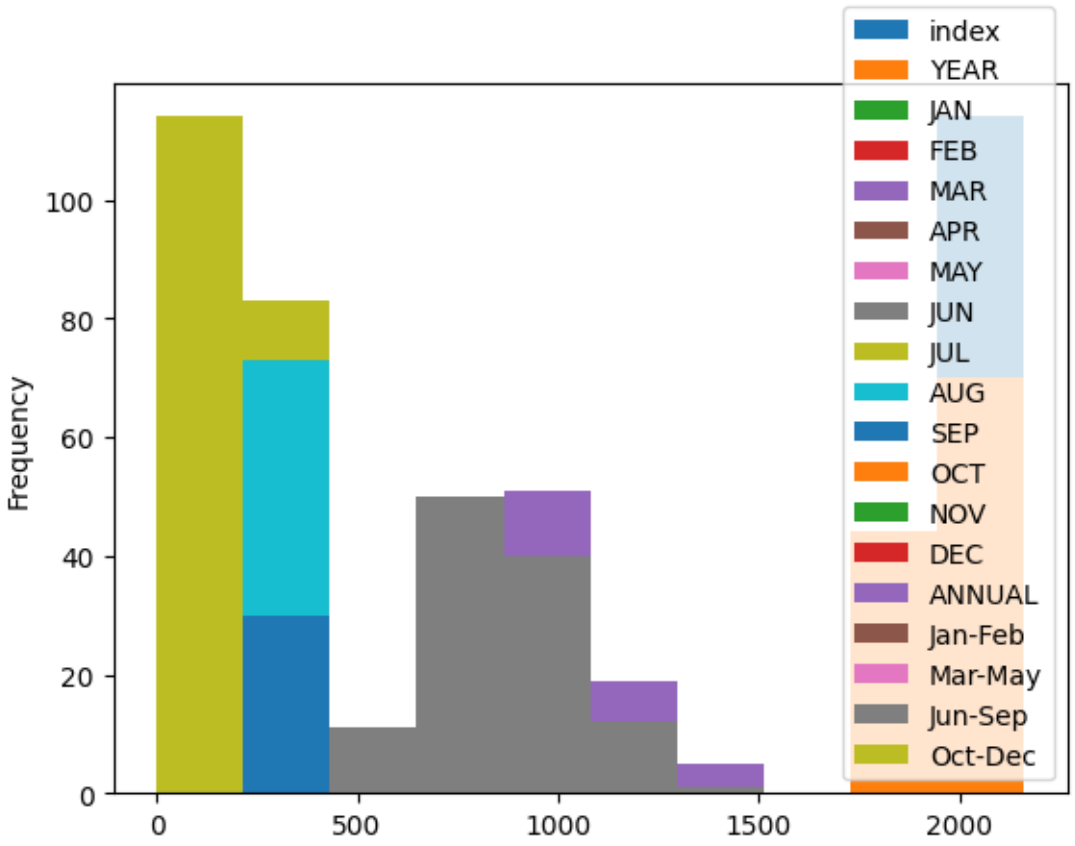
[8]: <Axes: >



7 Histogram

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

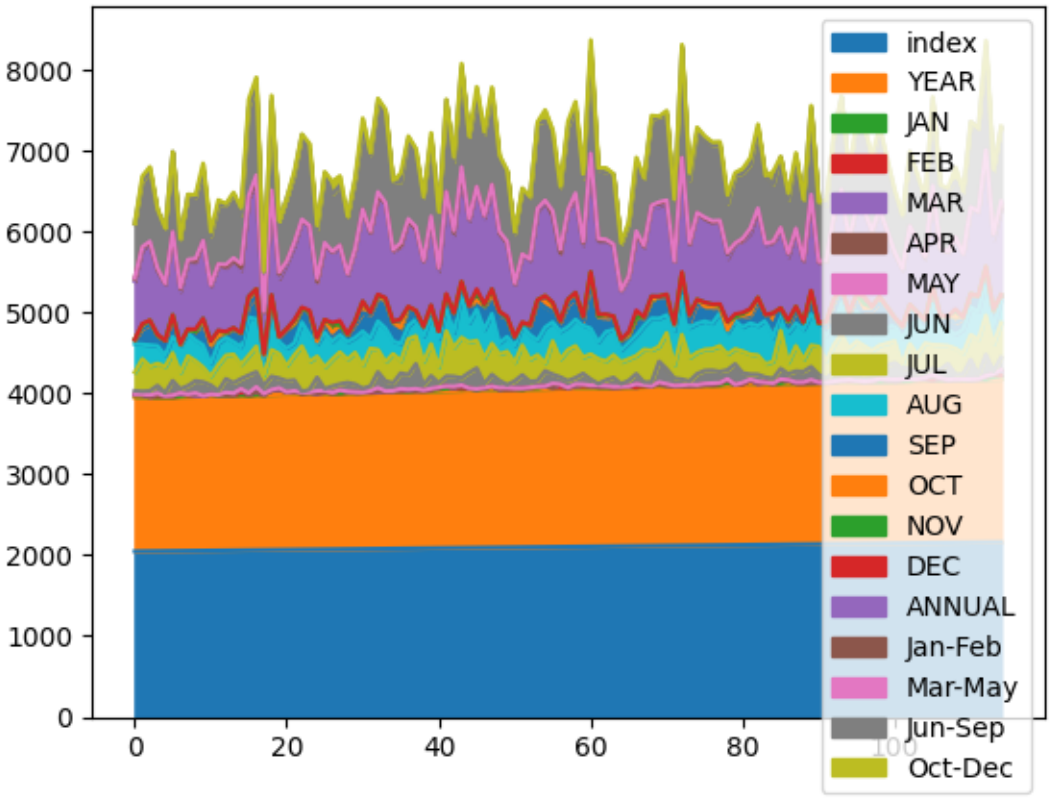
[9]: <Axes: ylabel='Frequency'>



8 Area chart

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

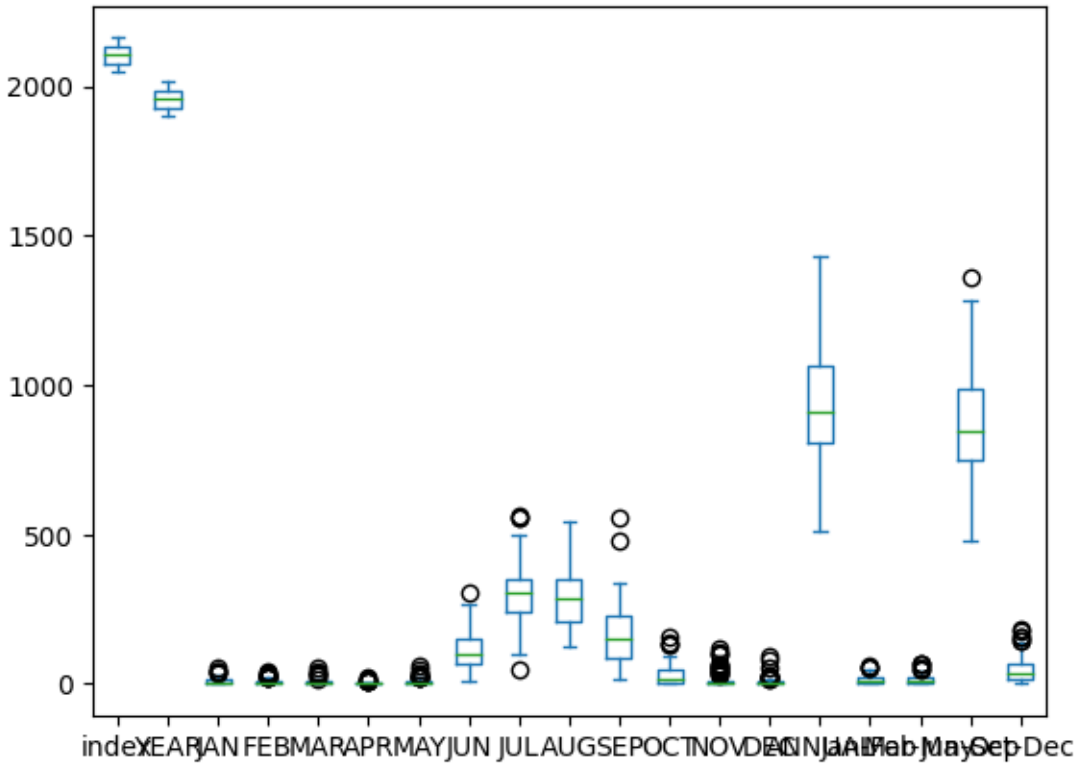
[10]: <Axes: >



9 Box chart

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

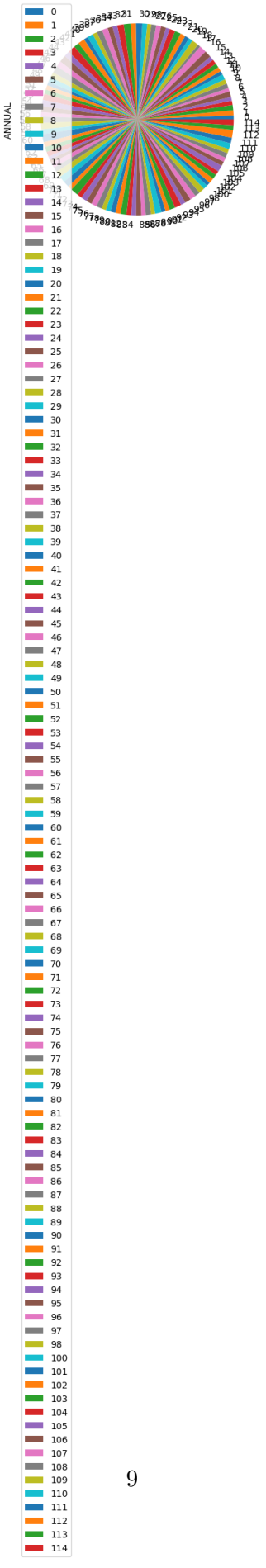
[11]: <Axes: >



10 Pie chart

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

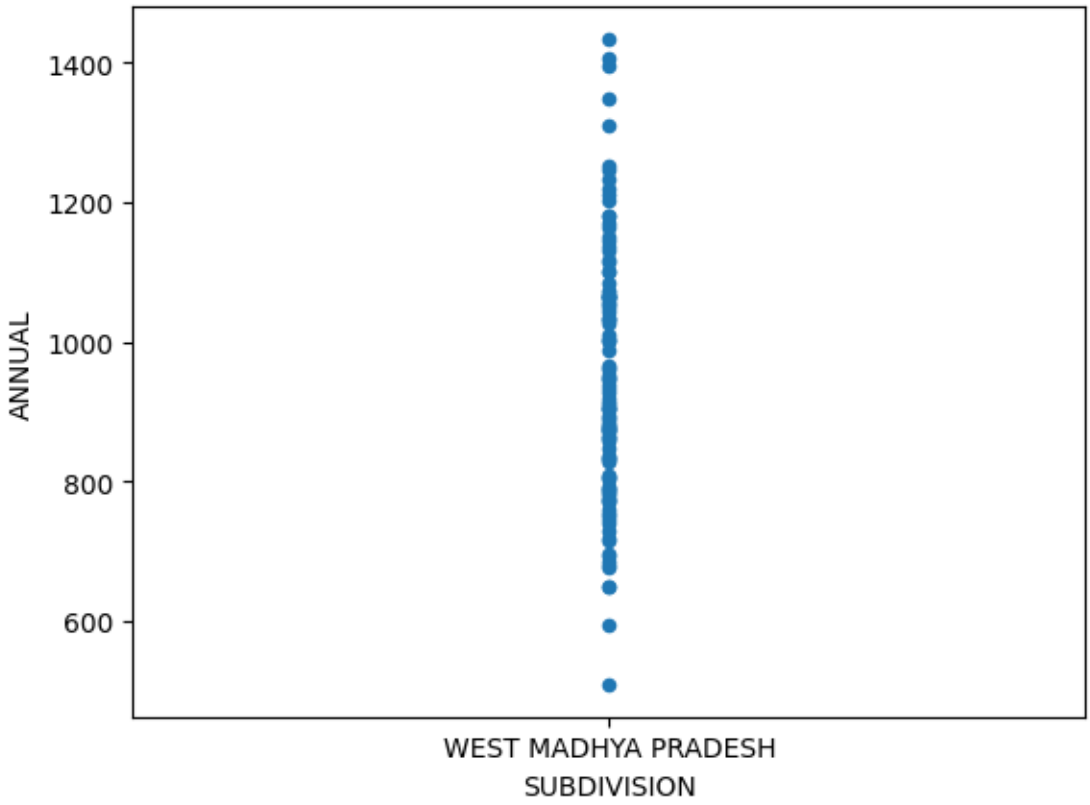
[12]: <Axes: ylabel='ANNUAL'>


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
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')
```

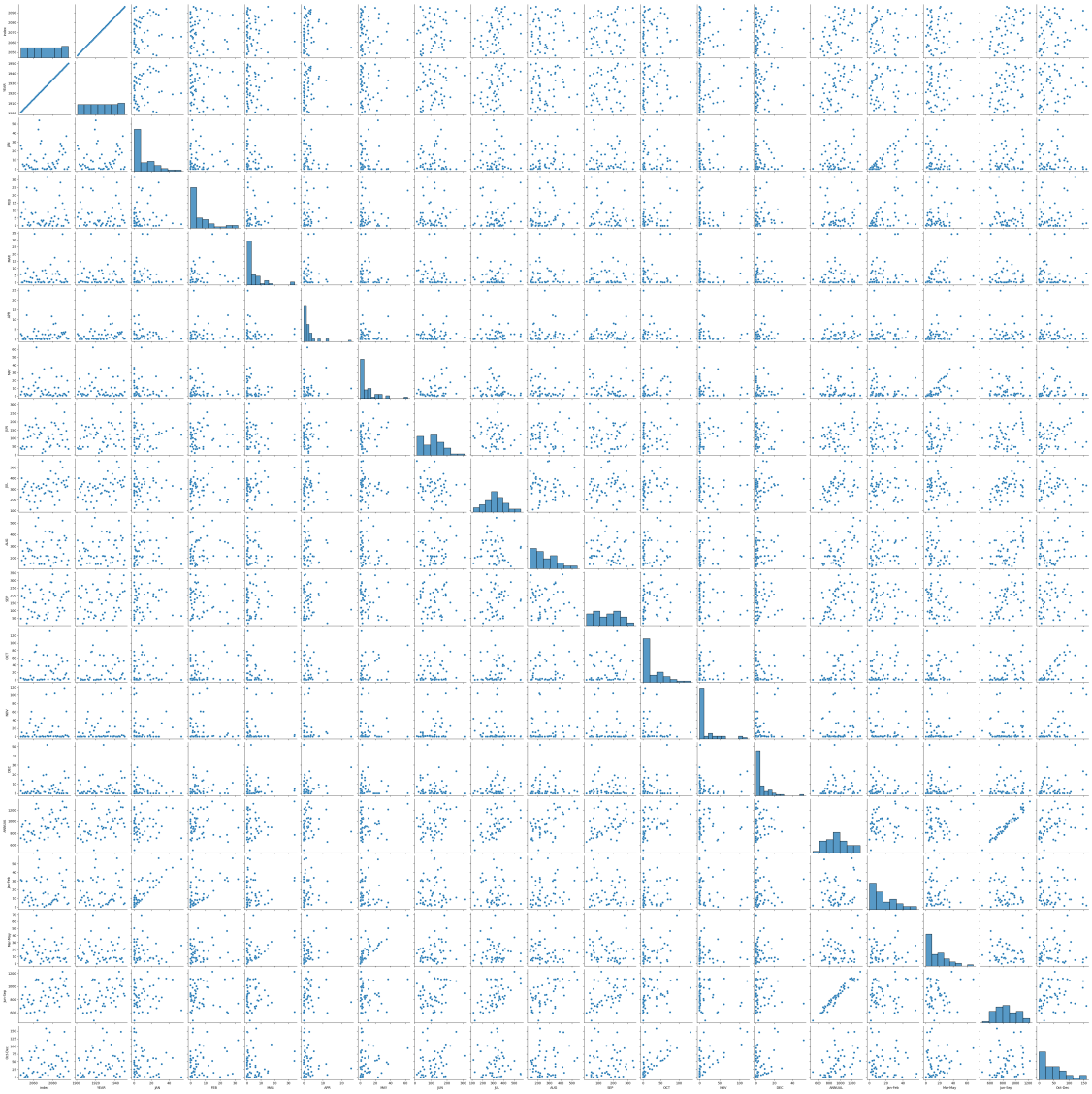
[13]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

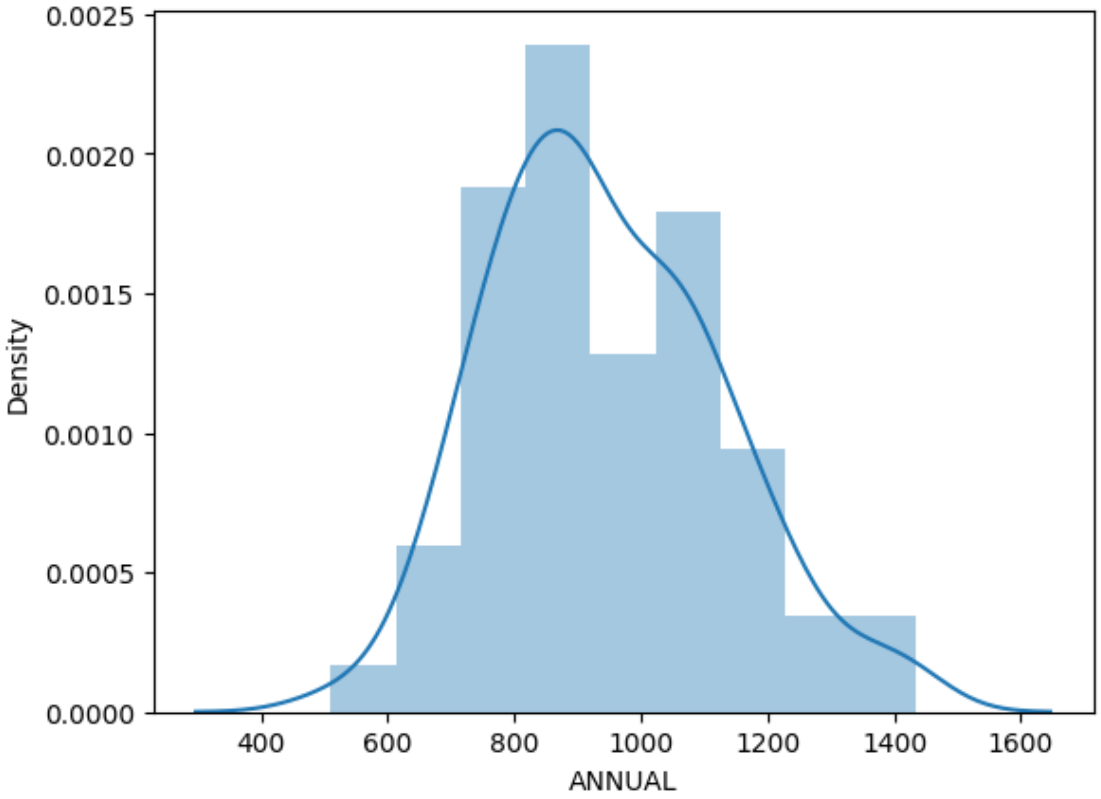
[14]:
```
<seaborn.axisgrid.PairGrid at 0x7cc998c26a70>
```



[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'])
```

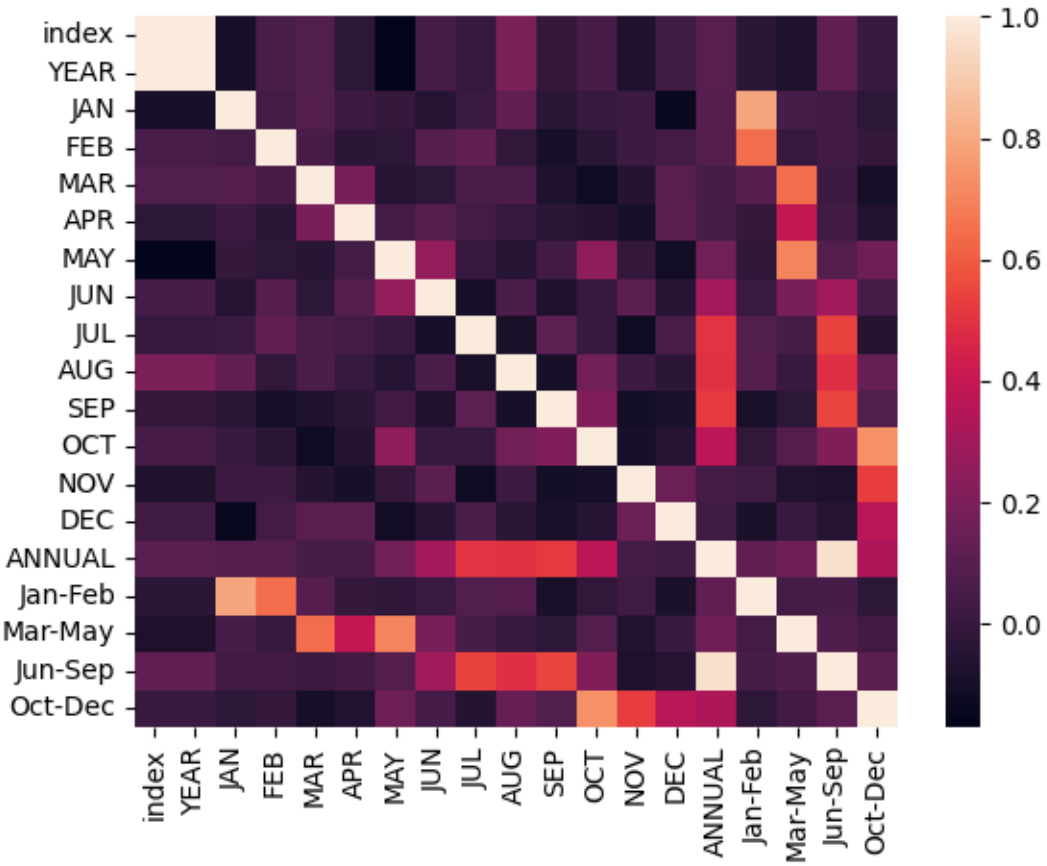
[15]: <Axes: xlabel='ANNUAL', ylabel='Density'>



[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