t5uz3bbic

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_uttarakhand.
      ↪csv")
     df
```

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

```
[2]:      index SUBDIVISION  YEAR    JAN    FEB    MAR   APR   MAY    JUN    JUL  \
     0     1242 UTTARAKHAND  1901  134.5   81.4   44.5   5.9  60.8   33.6  381.1
     1     1243 UTTARAKHAND  1902    0.0   17.0   52.2  63.7  52.1  113.1  444.1
     2     1244 UTTARAKHAND  1903   68.0    7.9   87.6  10.3  37.5   83.0  251.6
     3     1245 UTTARAKHAND  1904   40.0    5.2   78.3  13.6  61.1  180.1  449.6
     4     1246 UTTARAKHAND  1905  115.4   80.7   99.8  26.1  70.3  111.5  299.9
     ..     …          …   …     …     …    …    …    …    …    …
     110   1352 UTTARAKHAND  2011   30.9   65.2   18.0  30.9  84.2  223.1  433.3
     111   1353 UTTARAKHAND  2012   38.8   11.9   28.1  39.2   9.1   46.0  387.1
     112   1354 UTTARAKHAND  2013   73.0  188.3   22.0  24.7  18.2  488.9  413.4
     113   1355 UTTARAKHAND  2014   45.9   99.9   68.4  37.6  52.9   62.9  462.7
     114   1356 UTTARAKHAND  2015   54.5   62.6  127.3  57.3  38.0  186.6  337.0

            AUG    SEP   OCT   NOV   DEC  ANNUAL  Jan-Feb  Mar-May  Jun-Sep  \
     0    612.3  167.1  16.3   0.0  24.9  1562.5    215.9    111.3   1194.1
     1    327.5  220.4  31.9   2.1   0.0  1324.2     17.1    168.1   1105.1
     2    442.7  249.3  57.5   0.0  11.3  1306.5     75.9    135.4   1026.5
     3    417.2  174.1   6.3  35.6  31.0  1492.0     45.2    153.0   1221.0
     4    349.5  129.5   0.0   1.0  18.5  1302.1    196.0    196.2    890.4
     ..     …     …    …    …    …      …        …        …        …
```

```
110  523.7  148.4   3.4   1.2   2.3  1564.7     96.1     133.1    1328.5
111  419.5  220.6   4.7   3.4  15.5  1223.9     50.8      76.4    1073.1
112  359.4  111.3  29.1   3.2   3.8  1735.4    261.3      65.0    1373.0
113  264.2  107.9  40.8   0.0  44.3  1287.4    145.8     158.8     897.7
114  305.3   52.6  16.8   2.4   7.2  1247.6    117.0     222.6     881.5

     Oct-Dec
0       41.3
1       34.0
2       68.7
3       72.9
4       19.5
..       …
110      6.9
111     23.6
112     36.2
113     85.1
114     26.4

[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'>
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
```

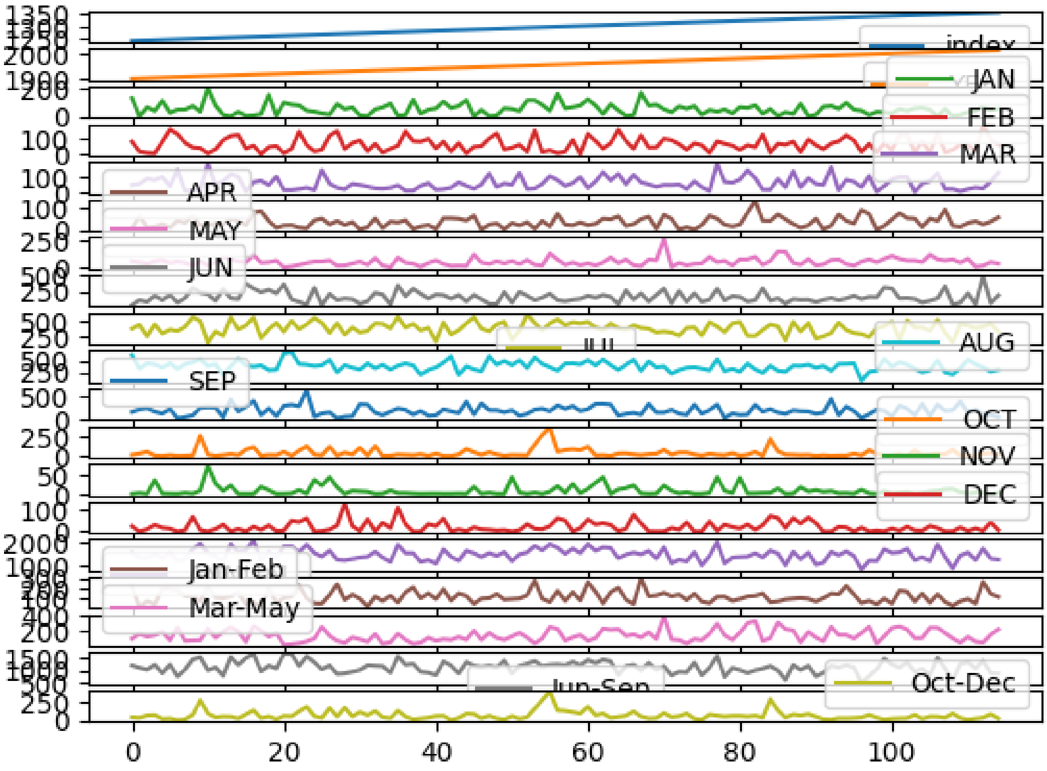
4 Line chart

[6]:

```
df.plot.line(subplots=True)
```

[6]:

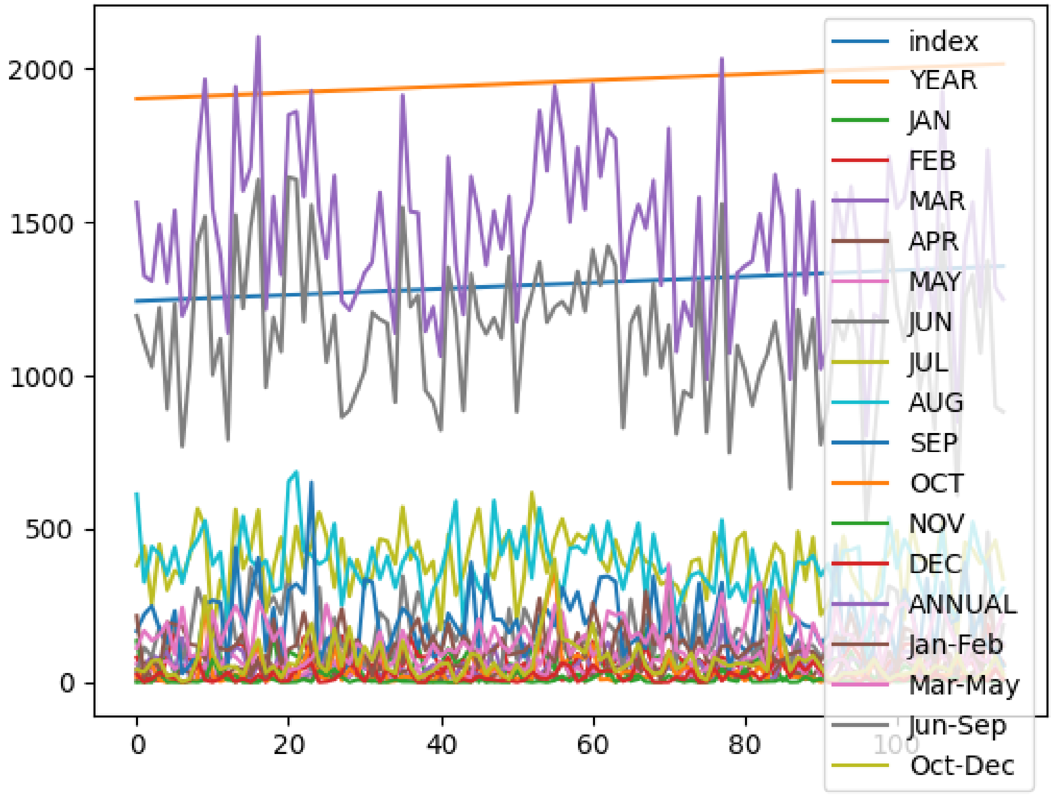
```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >], dtype=object)
```



5 Line chart

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

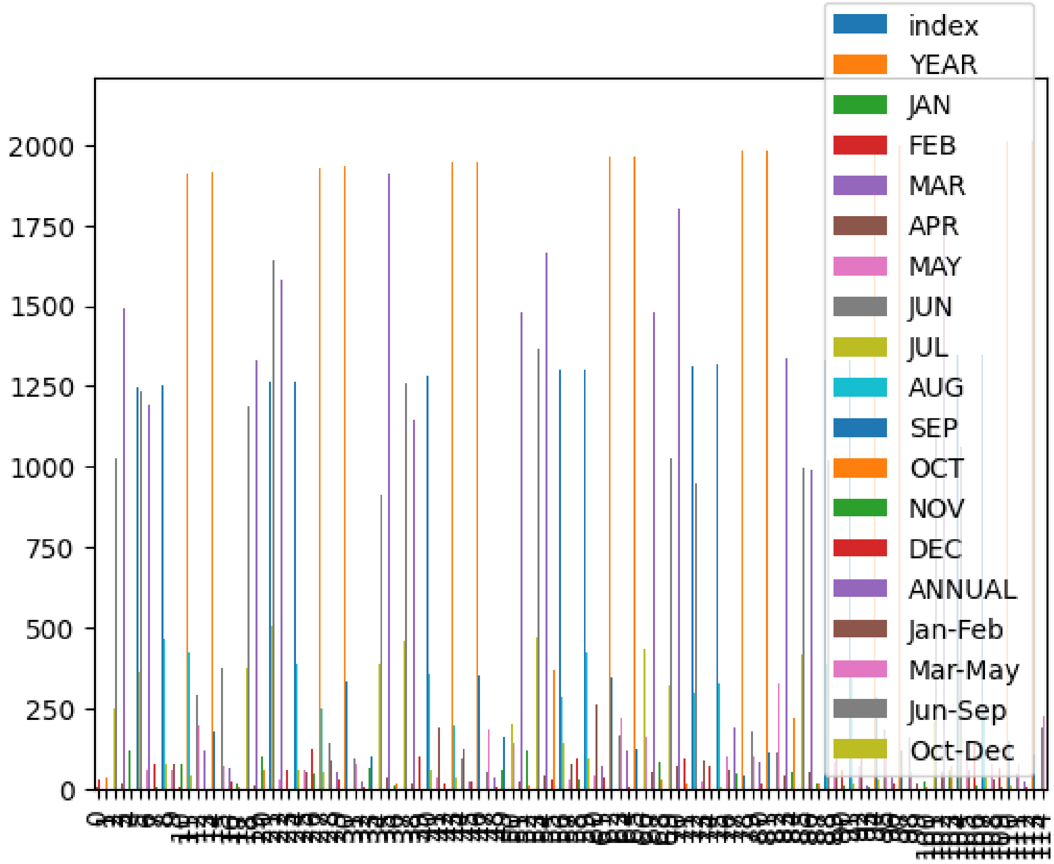
[7]:
```
<Axes: >
```



6 Bar chart

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

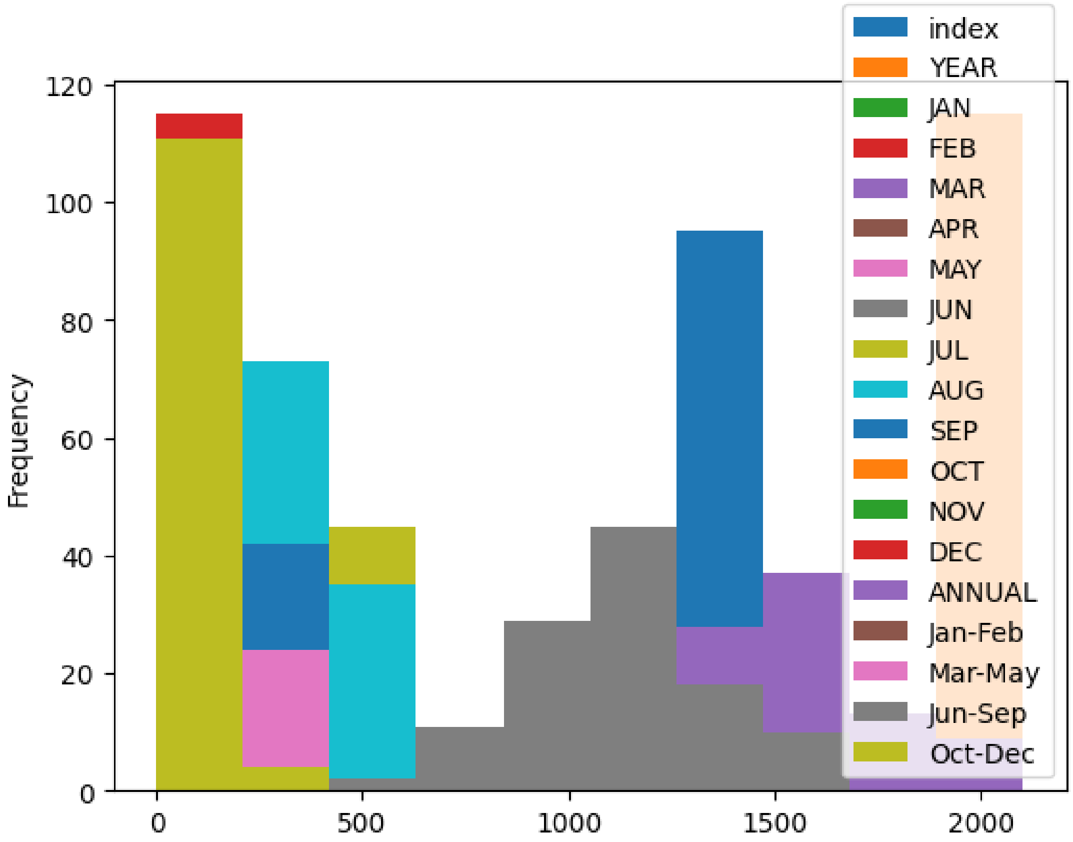
[8]:
```
<Axes: >
```



7 Histogram

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

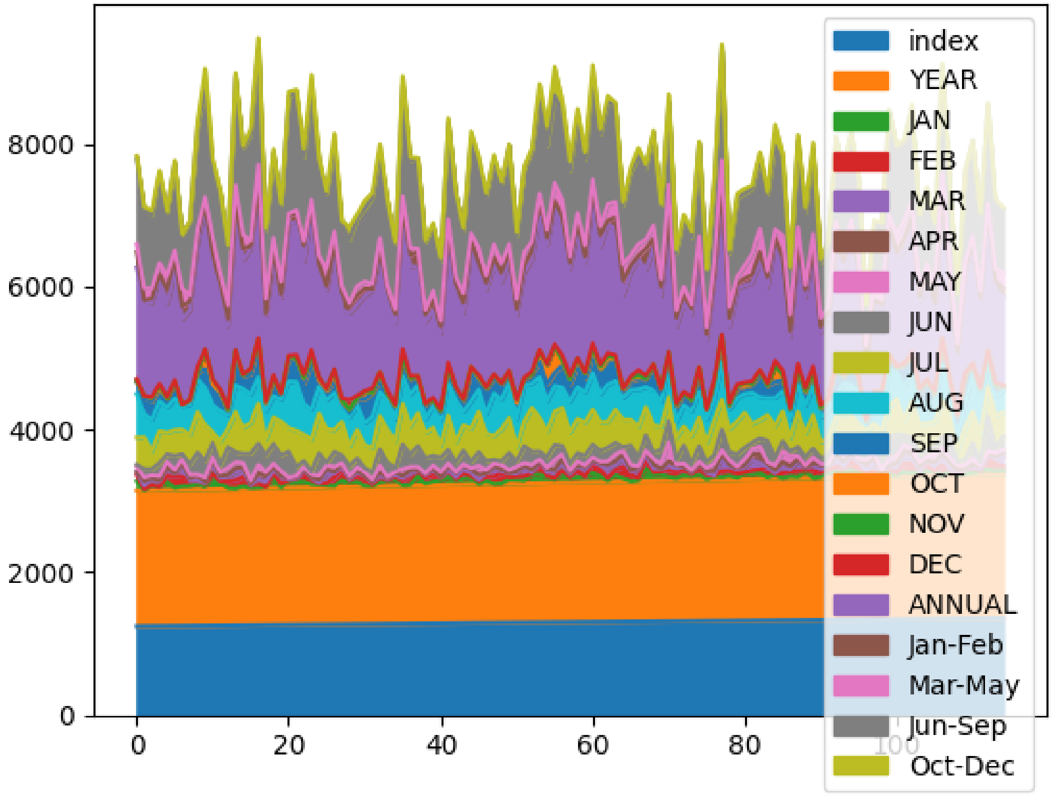
[9]:
```
<Axes: ylabel='Frequency'>
```



8 Area chart

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

```
[10]: <Axes: >
```



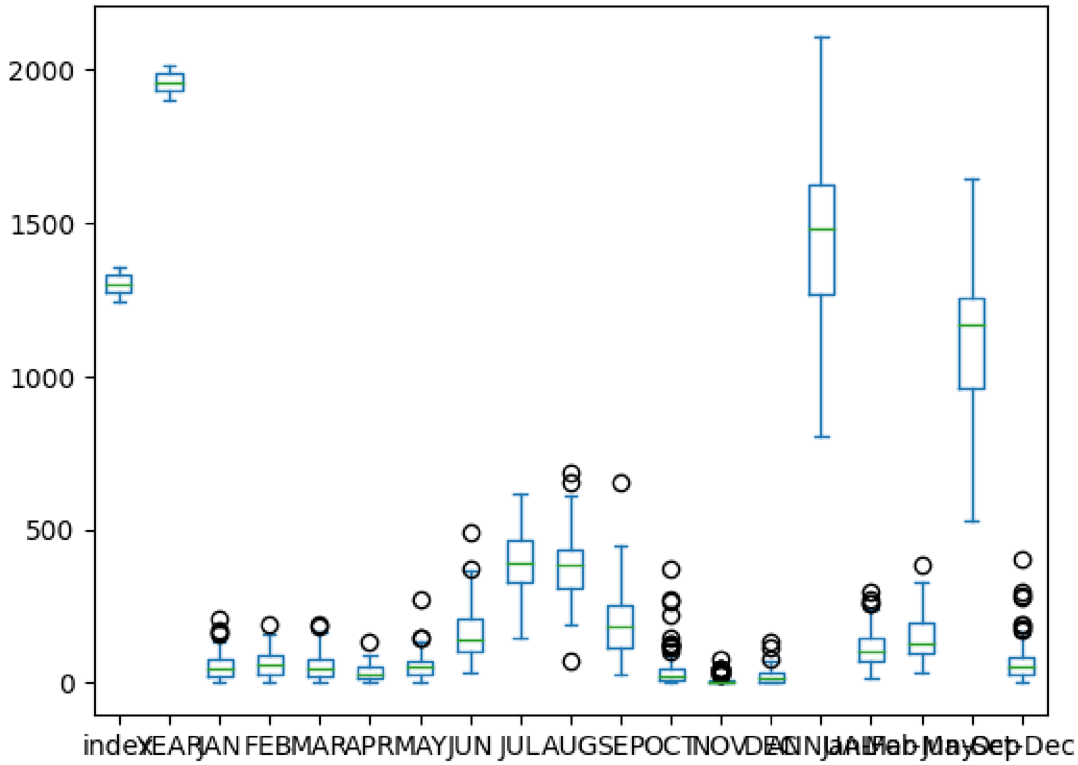
9 Box chart

[11]:

```
df.plot.box()
```

[11]:

```
<Axes: >
```



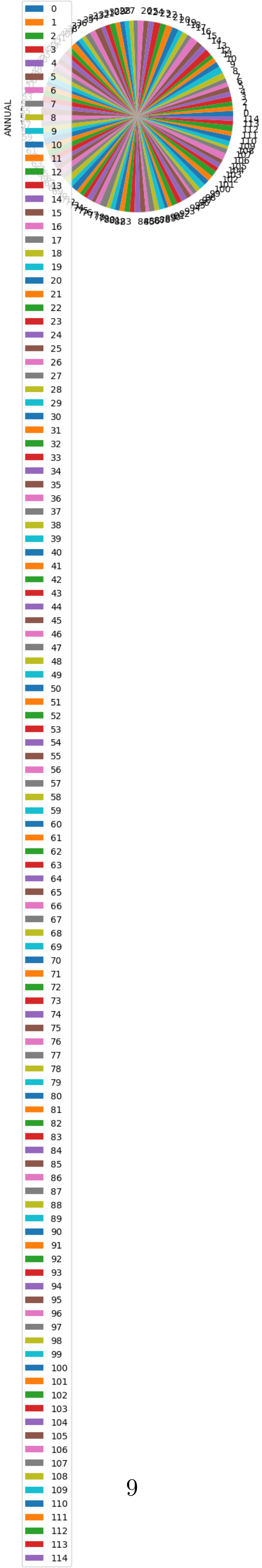
10 Pie chart

[12]:

```
df.plot.pie(y='ANNUAL' )
```

[12]:

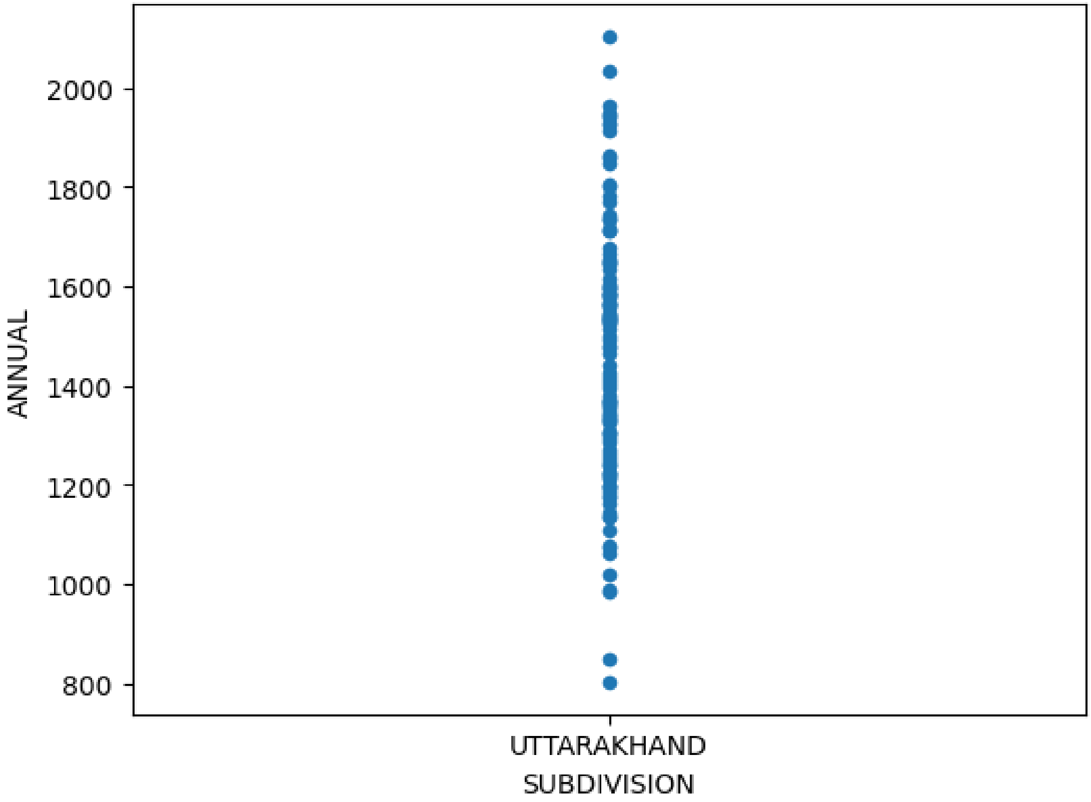
```
<Axes: ylabel='ANNUAL'>
```


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')
```

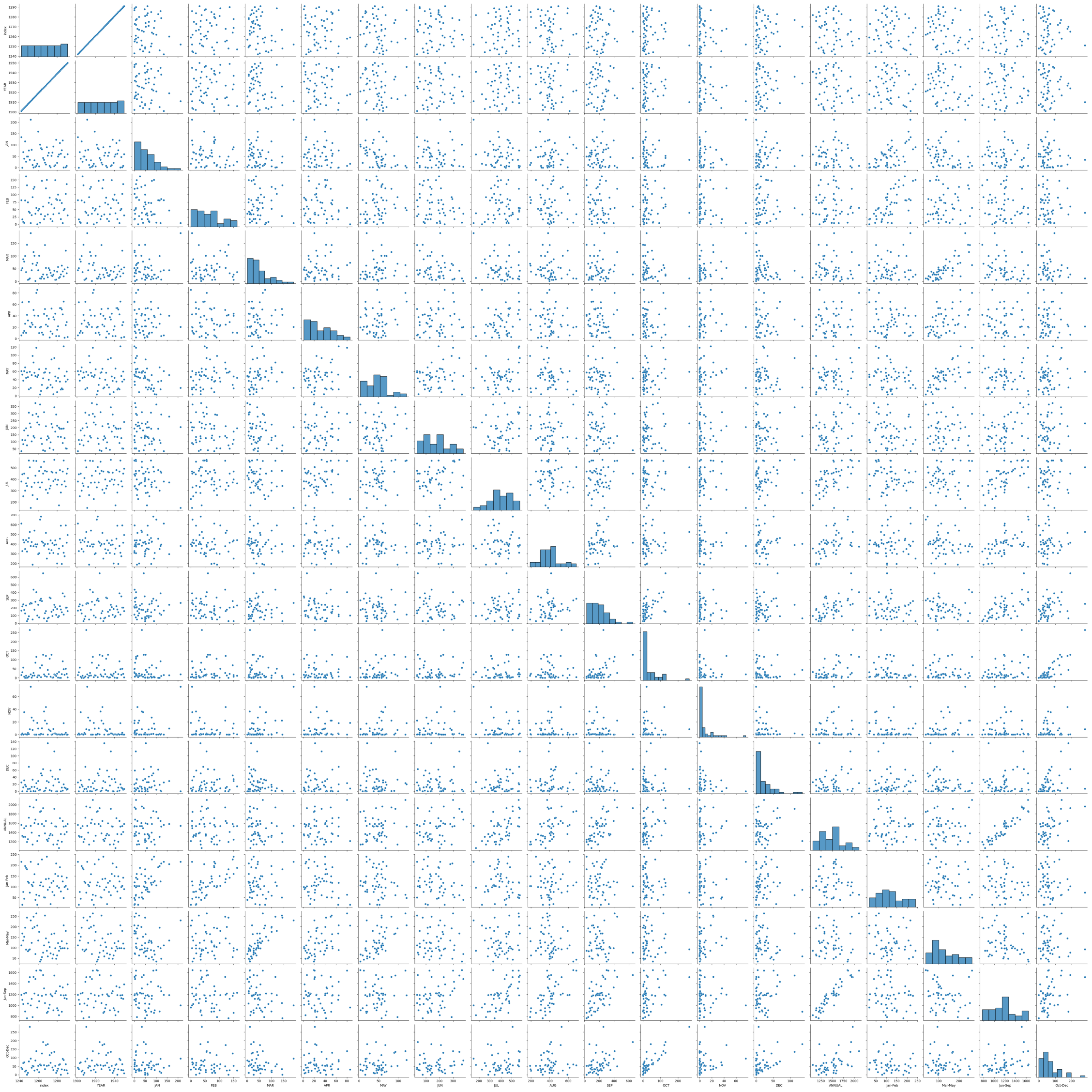
[13]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

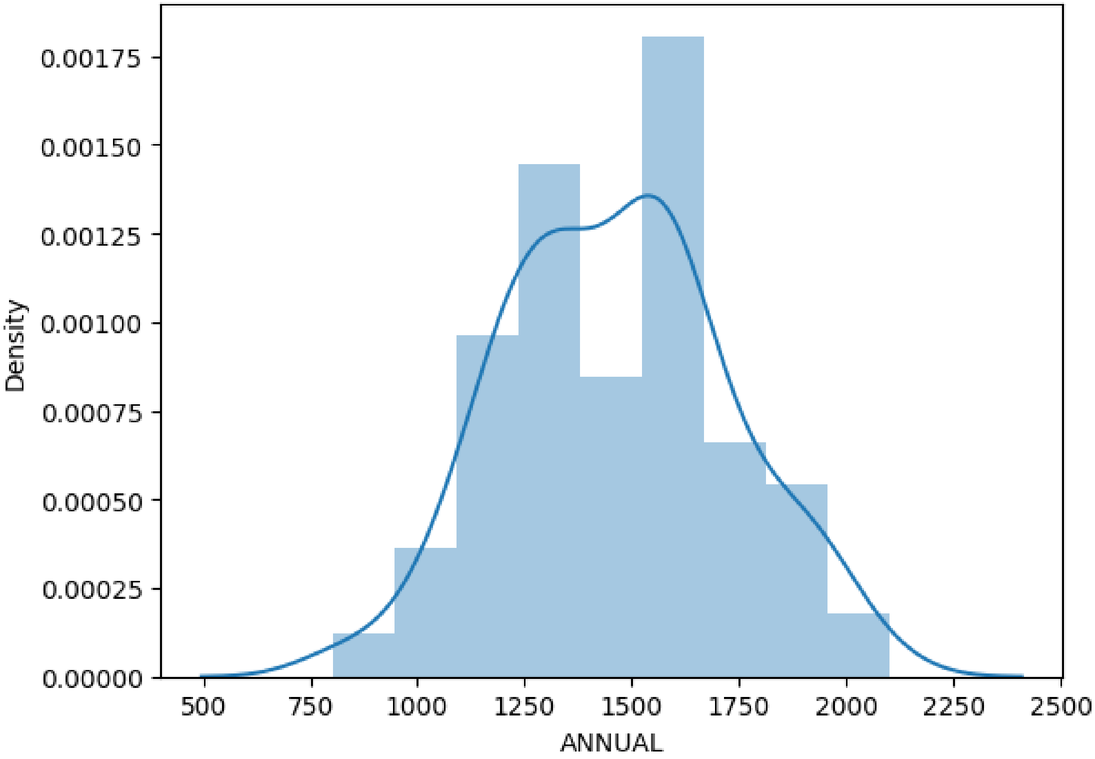
[14]:
```
<seaborn.axisgrid.PairGrid at 0x7ad57bdff490>
```



```
[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'])
```

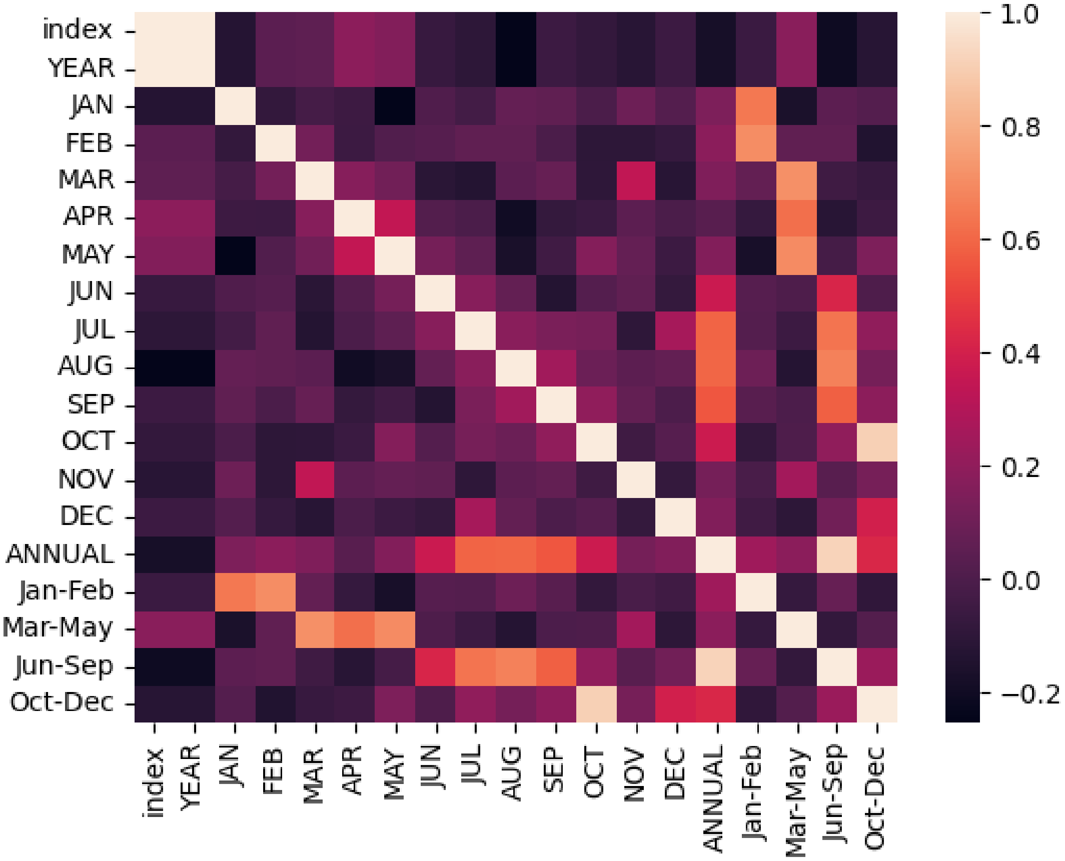
```
[15]: <Axes: xlabel='ANNUAL', ylabel='Density'>
```



```
[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