ptrjp3jt9

August 4, 2023

1 20104169 - SUMESH R

2 Importing Libraries

[2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[3]:
```
from google.colab import drive
drive.mount('/content/drive')
df=pd.read_csv("/content/drive/MyDrive/mydatasets/rainfall/rainfall_arunachal␣
 ↪pradesh.csv")
df
```

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

[3]:
```
    index         SUBDIVISION  YEAR   JAN    FEB    MAR    APR    MAY     JUN  \
0     110   ARUNACHAL PRADESH  1916  48.1   69.8   71.1  316.1  424.6  1124.9
1     111   ARUNACHAL PRADESH  1917  21.4  164.5    NaN  269.6  107.9   823.8
2     112   ARUNACHAL PRADESH  1918  10.4   11.0  191.2  144.6  861.1  1609.9
3     113   ARUNACHAL PRADESH  1919  34.5   67.8   28.5  256.9  420.6   973.6
4     114   ARUNACHAL PRADESH  1920  14.0  196.3  605.6  364.7  173.6   840.6
..    …                  …    …     …     …      …      …      …      …
92    202   ARUNACHAL PRADESH  2011  40.0   51.3  174.5  240.8  219.6   288.4
93    203   ARUNACHAL PRADESH  2012  57.8   35.8  134.2  403.4  187.4   645.8
94    204   ARUNACHAL PRADESH  2013  18.5   40.5  115.1  175.1  335.8   290.0
95    205   ARUNACHAL PRADESH  2014  19.0  101.9   80.3   86.7  299.0   415.8
96    206   ARUNACHAL PRADESH  2015  30.8   47.5   97.5  287.1  238.9   637.9

       JUL    AUG    SEP    OCT   NOV   DEC  ANNUAL  Jan-Feb  Mar-May  \
0      NaN  629.7  333.9    NaN   NaN   NaN     NaN    117.9    811.8
1    909.1  628.4  411.5  199.3  63.5   0.0     NaN    185.9      NaN
2   1303.0  692.6  515.8  125.2   7.8  13.7  5486.3     21.4   1196.9
3    999.0  286.7  628.7  948.3  40.7   8.6  4693.9    102.3    706.0
4    535.4  896.5  376.7  103.3   0.0   0.0  4106.7    210.3   1143.9
..     …      …      …      …     …     …       …        …        …
```

```
92   531.4  277.6  286.7   51.9  16.2  15.2  2193.7     91.4    634.9
93   638.9  316.0  724.9  248.1  22.0  26.2  3440.3     93.6    724.9
94   329.6  230.2  316.1  164.1  13.3  14.6  2042.9     59.0    626.0
95   392.4  599.6  343.0   35.1  20.1  10.2  2403.2    120.9    466.0
96   329.3  595.5  374.2   65.2  33.8  29.8  2767.5     78.3    623.5

    Jun-Sep  Oct-Dec
0       NaN      NaN
1    2772.8    262.8
2    4121.3    146.7
3    2888.0    997.6
4    2649.2    103.3
..      …        …
92   1384.2     83.3
93   2325.6    296.3
94   1165.9    192.0
95   1750.8     65.4
96   1936.9    128.8

[97 rows x 20 columns]
```

3 Data Cleaning and Data Preprocessing

[4]:

```
df=df.dropna()
```

[5]:

```
df.columns
```

[5]:

```
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')
```

[6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 91 entries, 2 to 96
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   index        91 non-null     int64
 1   SUBDIVISION  91 non-null     object
 2   YEAR         91 non-null     int64
 3   JAN          91 non-null     float64
 4   FEB          91 non-null     float64
 5   MAR          91 non-null     float64
 6   APR          91 non-null     float64
 7   MAY          91 non-null     float64
 8   JUN          91 non-null     float64
 9   JUL          91 non-null     float64
 10  AUG          91 non-null     float64
 11  SEP          91 non-null     float64
 12  OCT          91 non-null     float64
 13  NOV          91 non-null     float64
 14  DEC          91 non-null     float64
 15  ANNUAL       91 non-null     float64
 16  Jan-Feb      91 non-null     float64
 17  Mar-May      91 non-null     float64
 18  Jun-Sep      91 non-null     float64
 19  Oct-Dec      91 non-null     float64
dtypes: float64(17), int64(2), object(1)
memory usage: 14.9+ KB
```

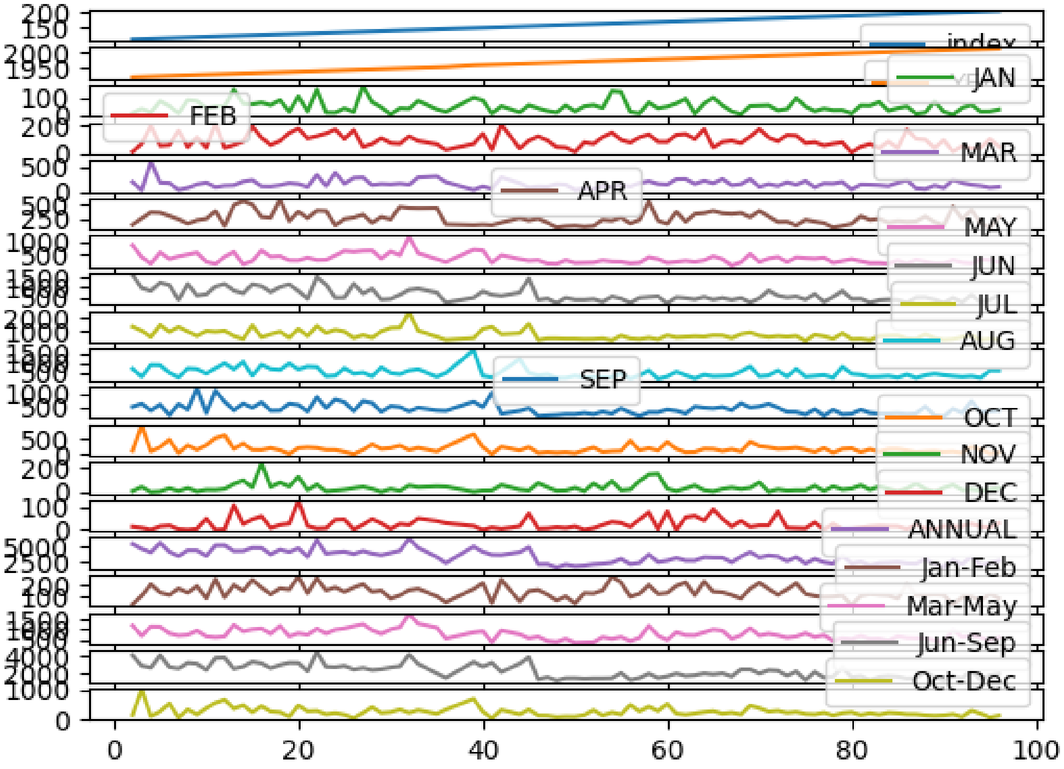
4 Line chart

[7]:

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

[7]:

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



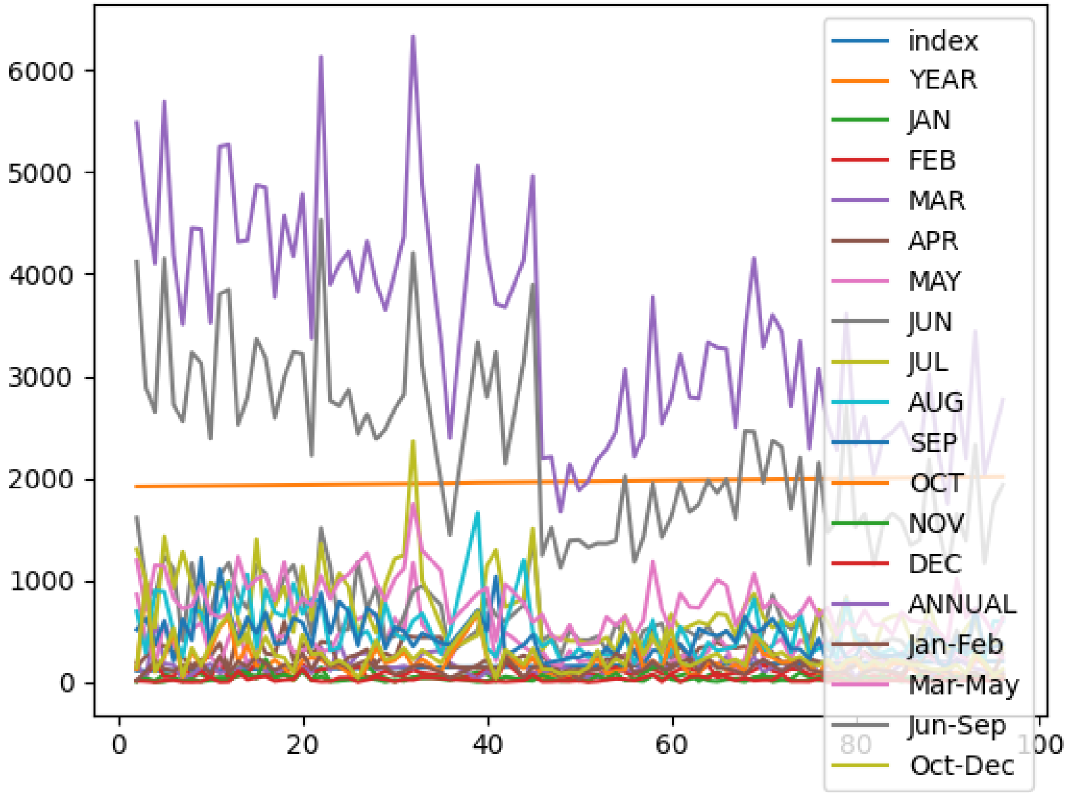
5 Line chart

[8]:

```
df.plot.line()
```

[8]:

```
<Axes: >
```



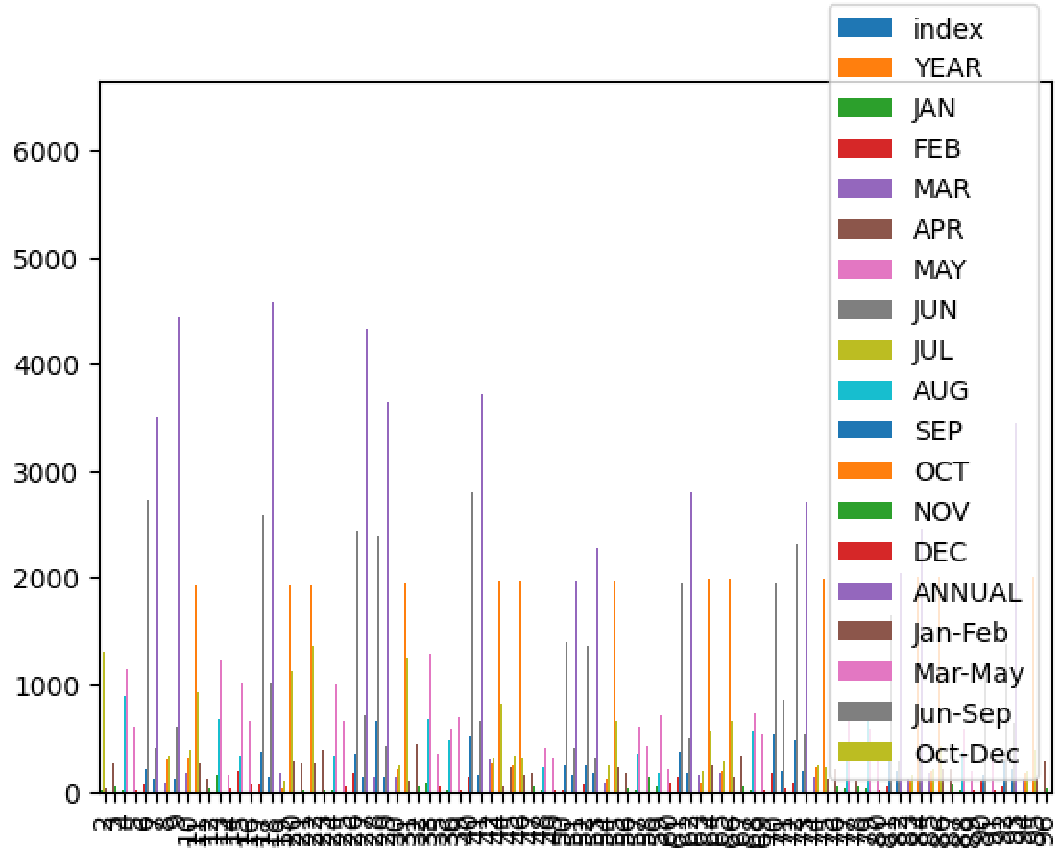
6 Bar chart

[9]:

```
df.plot.bar()
```

[9]:

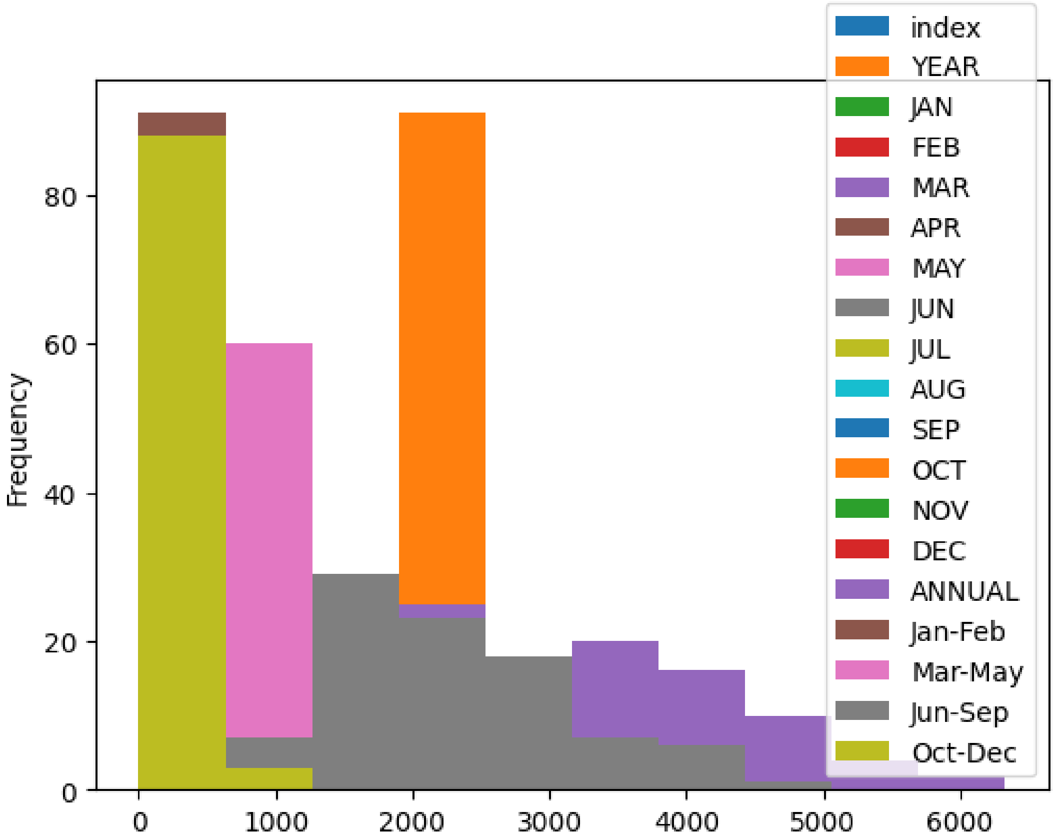
```
<Axes: >
```



7 Histogram

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

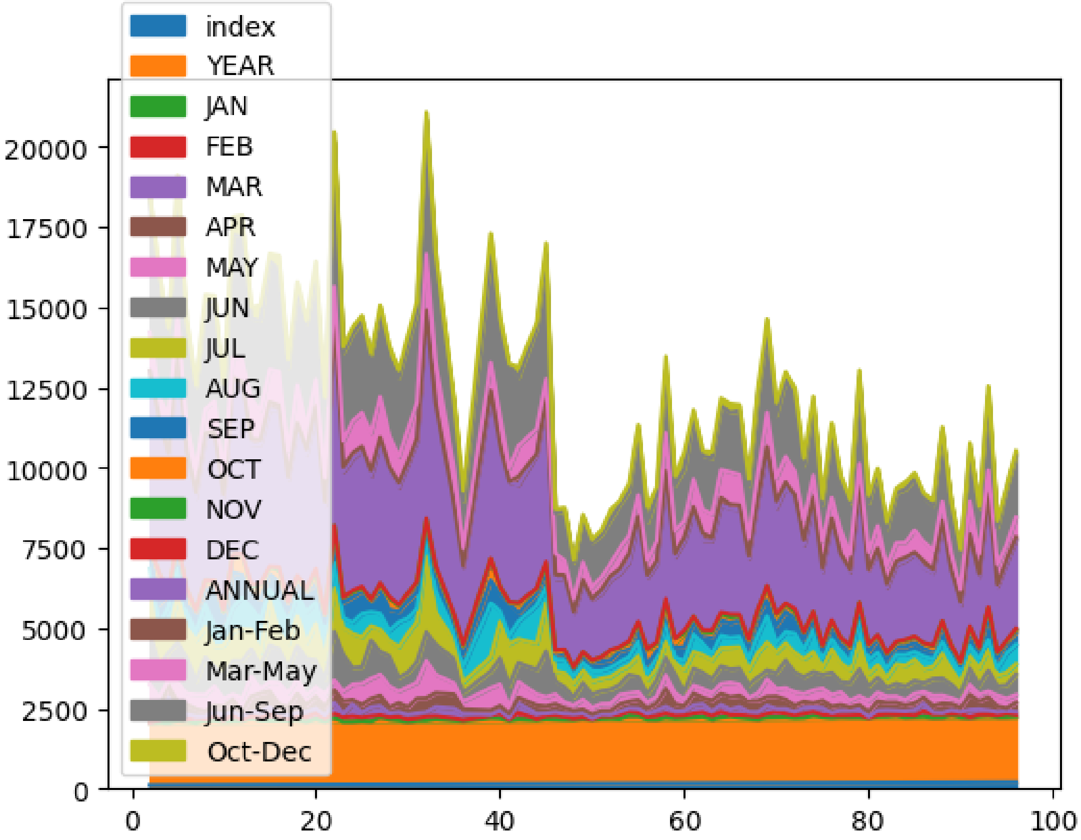
```
[10]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

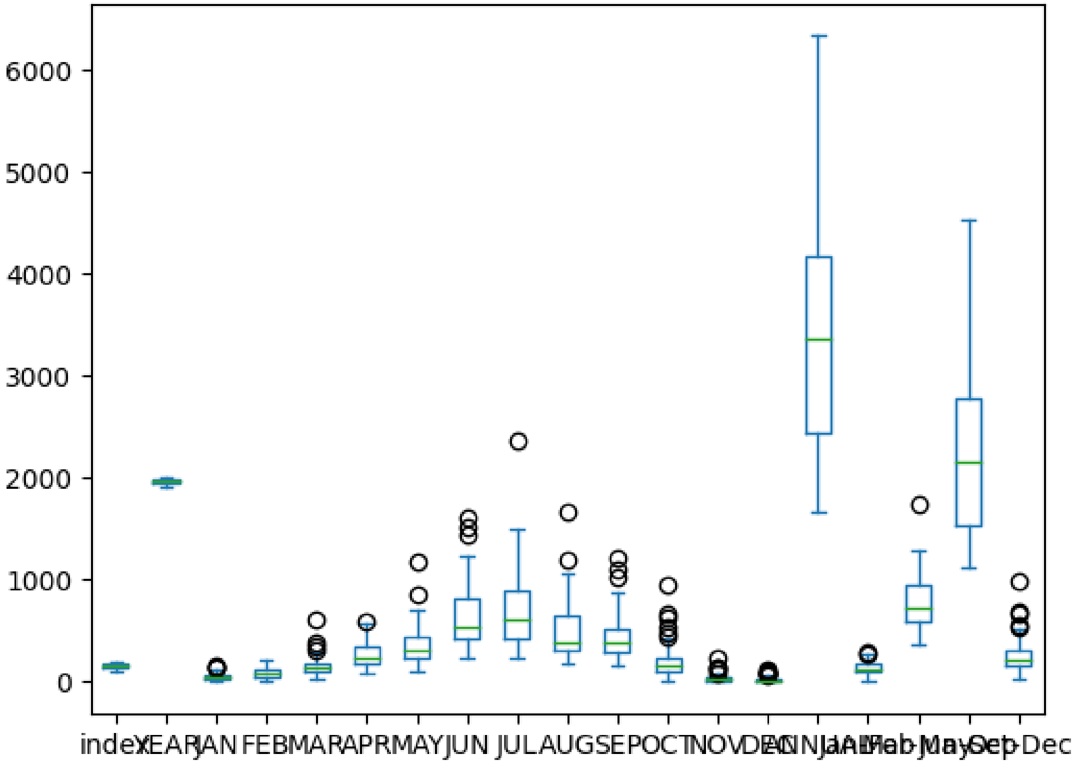
```
[11]: <Axes: >
```



9 Box chart

[12]:
```
df.plot.box()
```

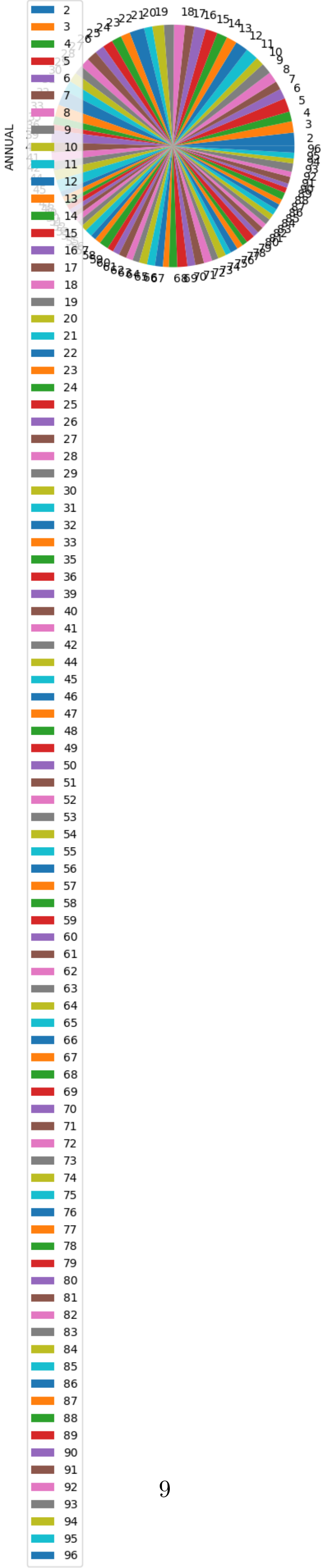
[12]:
```
<Axes: >
```



10 Pie chart

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

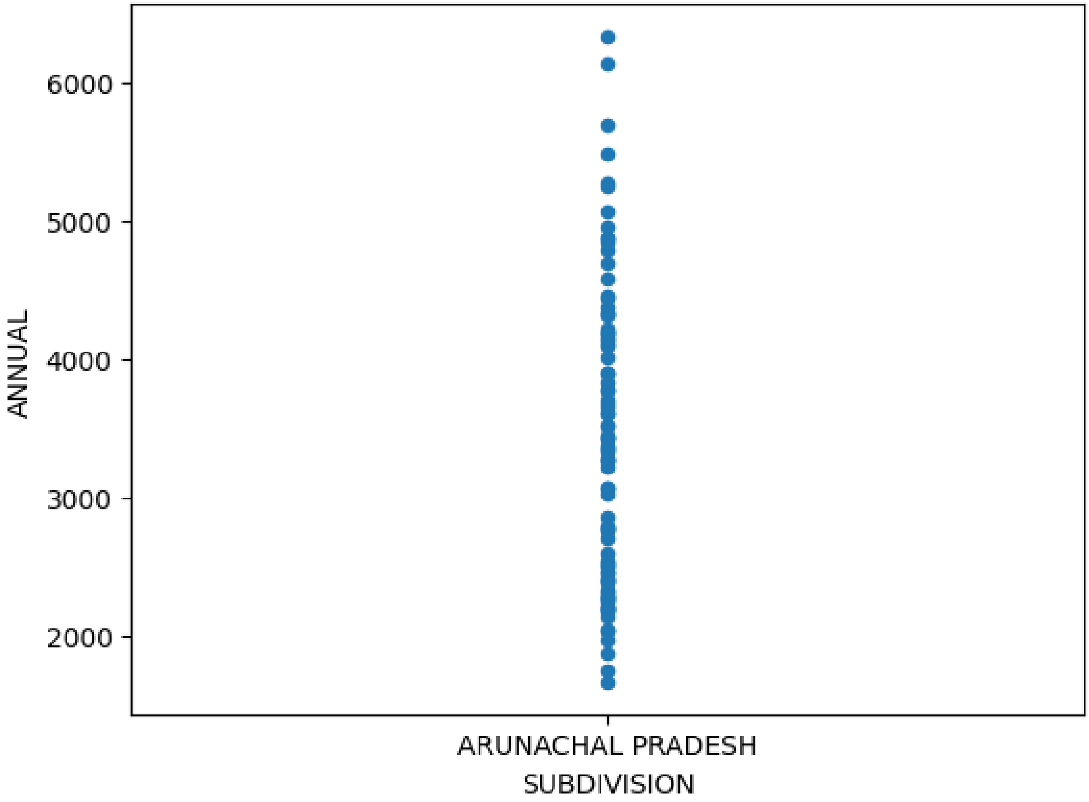
```
[13]: <Axes: ylabel='ANNUAL'>
```


ANNUAL
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
35
36
39
40
41
42
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

11 Scatter chart

[14]:
```
df.plot.scatter(x='SUBDIVISION' ,y='ANNUAL')
```

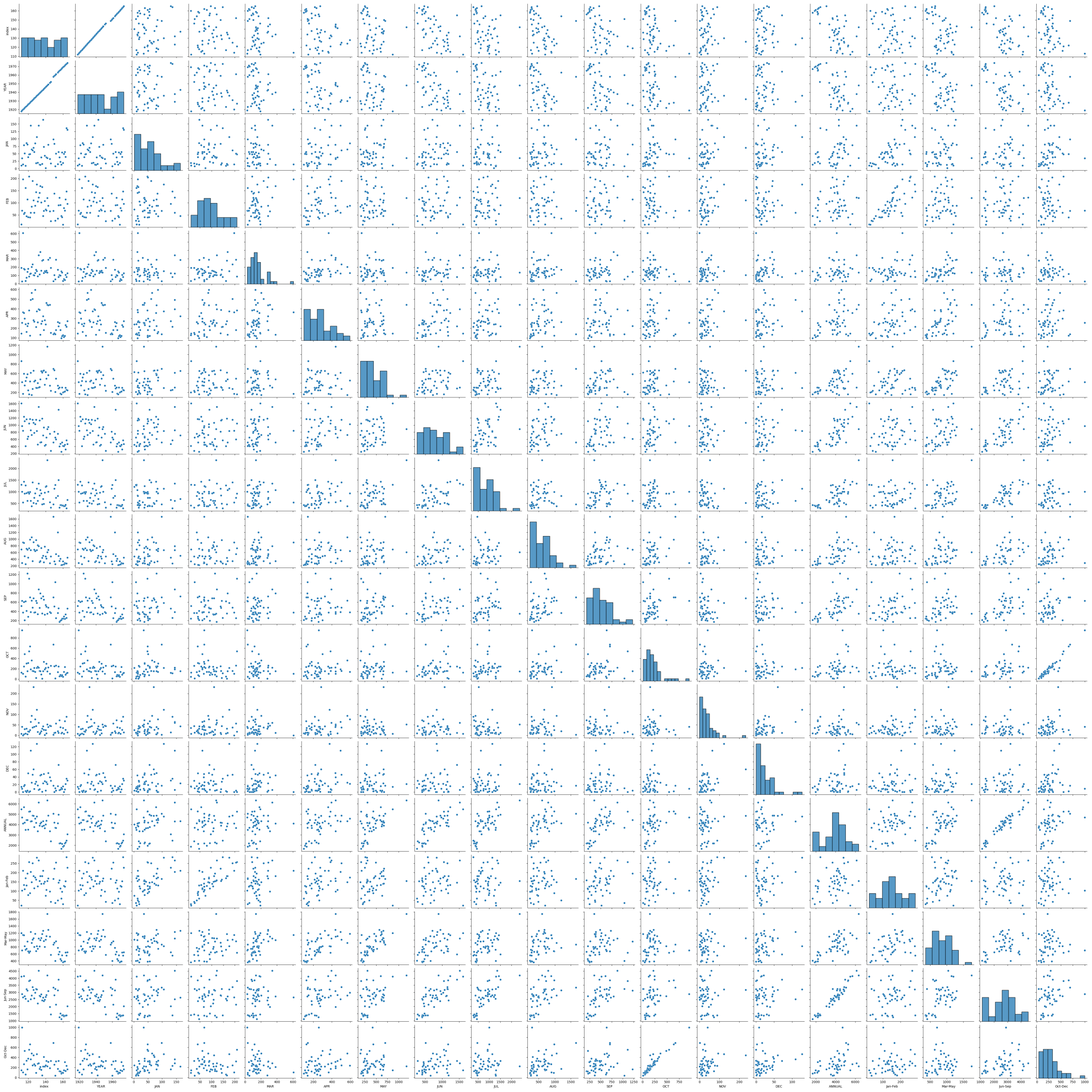
[14]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

[15]:
```
sns.pairplot(df[0:50])
```

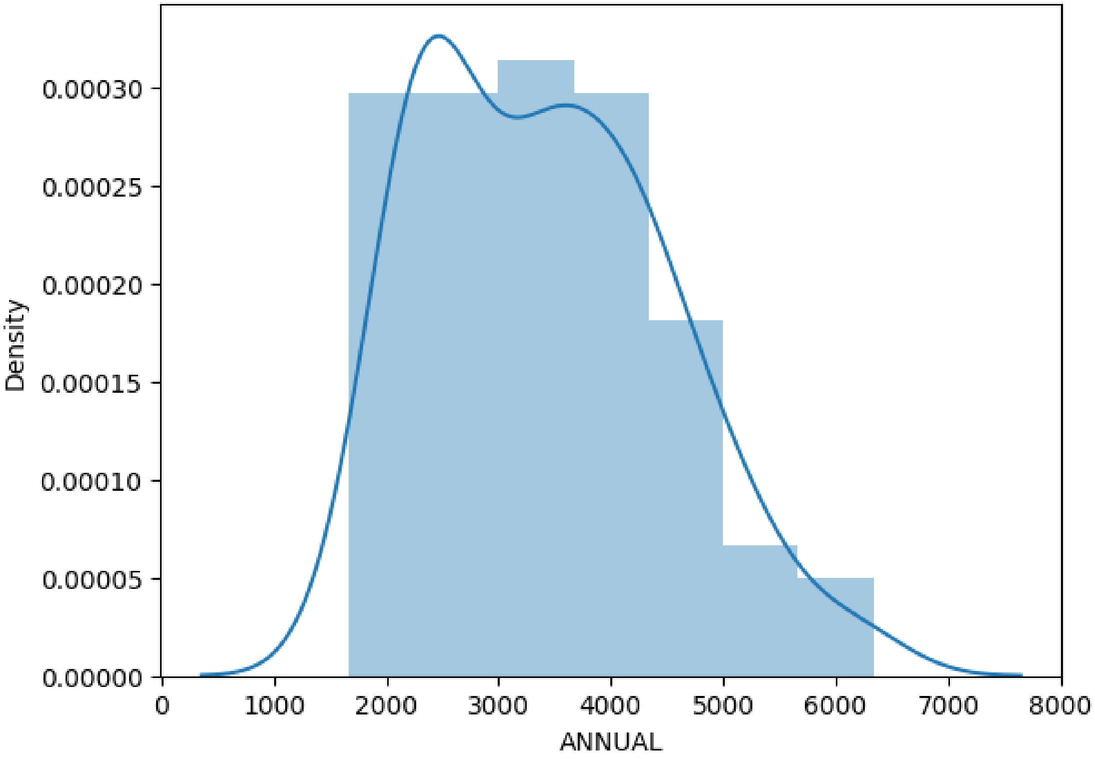
[15]:
```
<seaborn.axisgrid.PairGrid at 0x7a6161d88d00>
```



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

```
<ipython-input-16-5daa97052ca5>: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'])
```

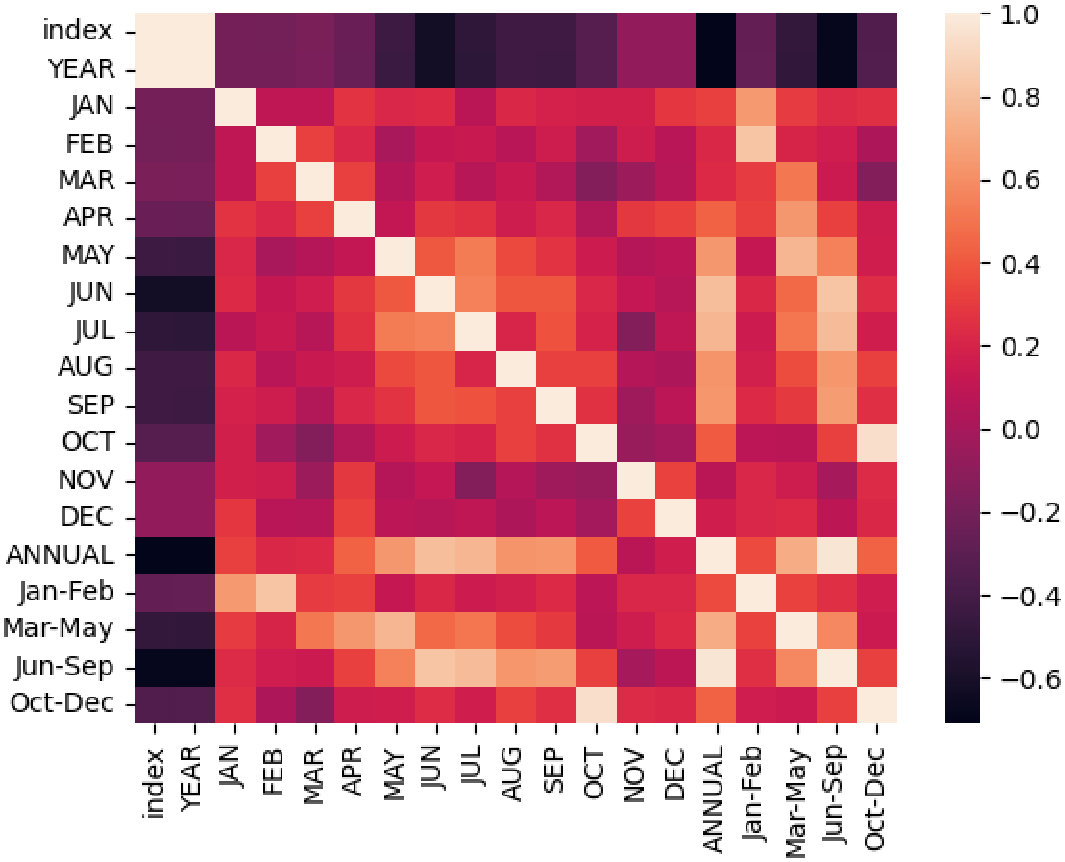
```
[16]: <Axes: xlabel='ANNUAL', ylabel='Density'>
```



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

```
<ipython-input-17-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())
```

```
[17]: <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
−0.4
−0.6