e2ybfxrpf

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_south␣
      ↪interior karnataka.csv")
     df
```

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

```
[2]:      index               SUBDIVISION  YEAR  JAN   FEB   MAR   APR    MAY  \
     0     3772  SOUTH INTERIOR KARNATAKA  1901  4.9  31.8   3.0  32.7  109.6
     1     3773  SOUTH INTERIOR KARNATAKA  1902  1.9   0.5   6.7  42.6   97.7
     2     3774  SOUTH INTERIOR KARNATAKA  1903  0.3   0.0   1.1  11.6  125.1
     3     3775  SOUTH INTERIOR KARNATAKA  1904  1.0   0.5   5.2  43.5  144.7
     4     3776  SOUTH INTERIOR KARNATAKA  1905  1.7   7.9  14.2  23.6  118.6
     ..     …                         …    …    …    …     …     …     …
     110   3882  SOUTH INTERIOR KARNATAKA  2011  2.1  12.4  12.4  80.2   83.5
     111   3883  SOUTH INTERIOR KARNATAKA  2012  4.6   5.5   8.1  99.0   45.6
     112   3884  SOUTH INTERIOR KARNATAKA  2013  0.5  10.1  11.7  34.6   95.6
     113   3885  SOUTH INTERIOR KARNATAKA  2014  0.4   2.4  17.7  46.7  130.5
     114   3886  SOUTH INTERIOR KARNATAKA  2015  1.7   0.2  24.4  80.5  125.3

            JUN    JUL    AUG    SEP    OCT    NOV   DEC  ANNUAL  Jan-Feb  Mar-May  \
     0    106.0  210.0  109.2  140.8  170.1   72.5  12.3  1003.0     36.8    145.4
     1     91.7  210.0   82.1  138.4  219.1   44.6  84.9  1020.1      2.4    147.0
     2    129.7  284.4  155.7  197.1  154.2  186.6  24.1  1269.9      0.3    137.7
     3    167.9  197.1   73.2   89.6  120.4    2.5   0.3   845.8      1.5    193.3
     4     95.9  148.4  140.6   43.1  142.8   22.4   0.3   759.4      9.5    156.5
     ..     …      …      …      …      …      …     …       …        …        …
```

```
110  177.1  202.4  199.5  111.2  144.8   56.7   5.0  1087.4     14.5     176.1
111   81.8  144.7  236.5  100.6   62.8   82.6   6.2   877.8     10.1     152.6
112  176.2  307.4  151.7  191.8  103.7   24.9   2.4  1110.7     10.6     142.0
113  106.8  271.6  254.6  161.6  152.9   20.2  18.7  1184.2      2.8     195.0
114  218.7  112.0  136.6  164.5  106.1  138.1   4.4  1112.5      1.9     230.2

     Jun-Sep  Oct-Dec
0      566.0    254.9
1      522.3    348.5
2      766.9    364.9
3      527.8    123.2
4      427.9    165.5
..       …        …
110    690.2    206.5
111    563.5    151.6
112    827.1    131.0
113    794.5    191.8
114    631.8    248.6

[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
```

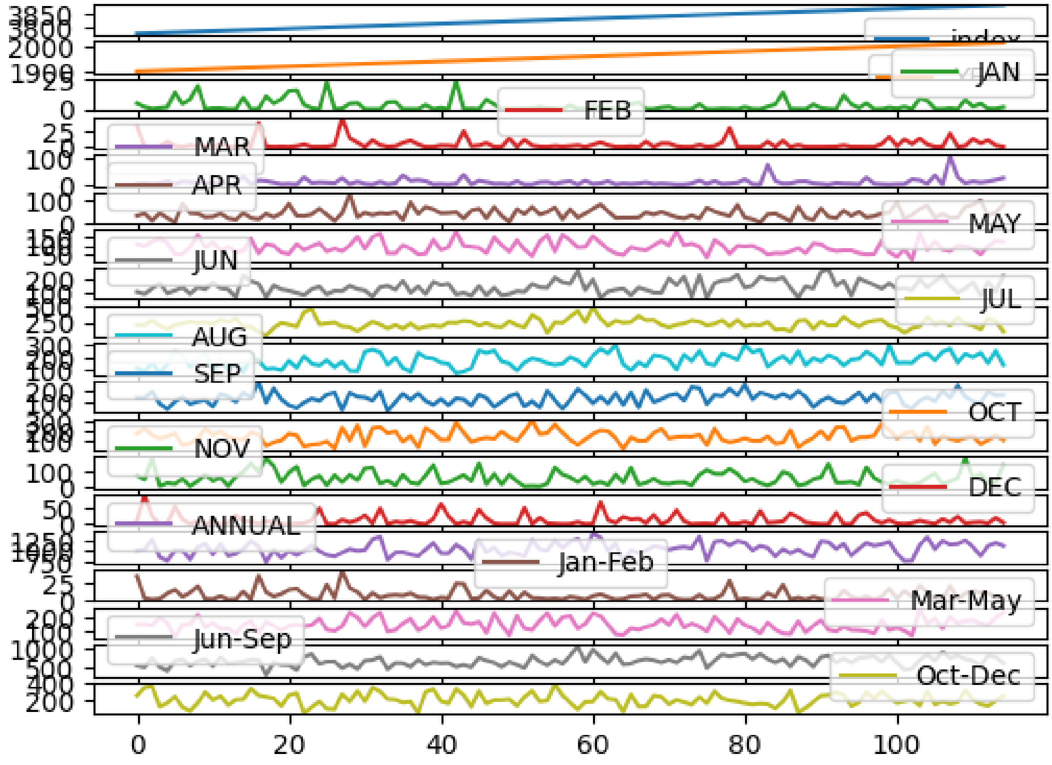
4 Line chart

[6]:

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

[6]:

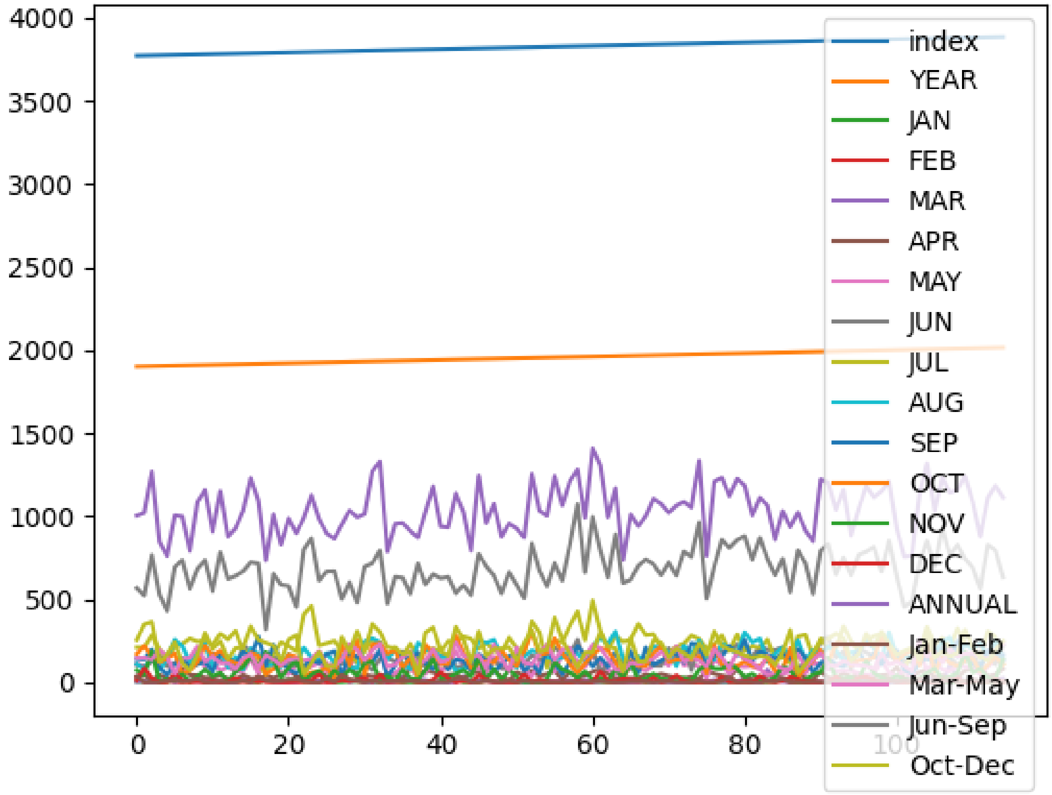
```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >], dtype=object)
```



5 Line chart

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

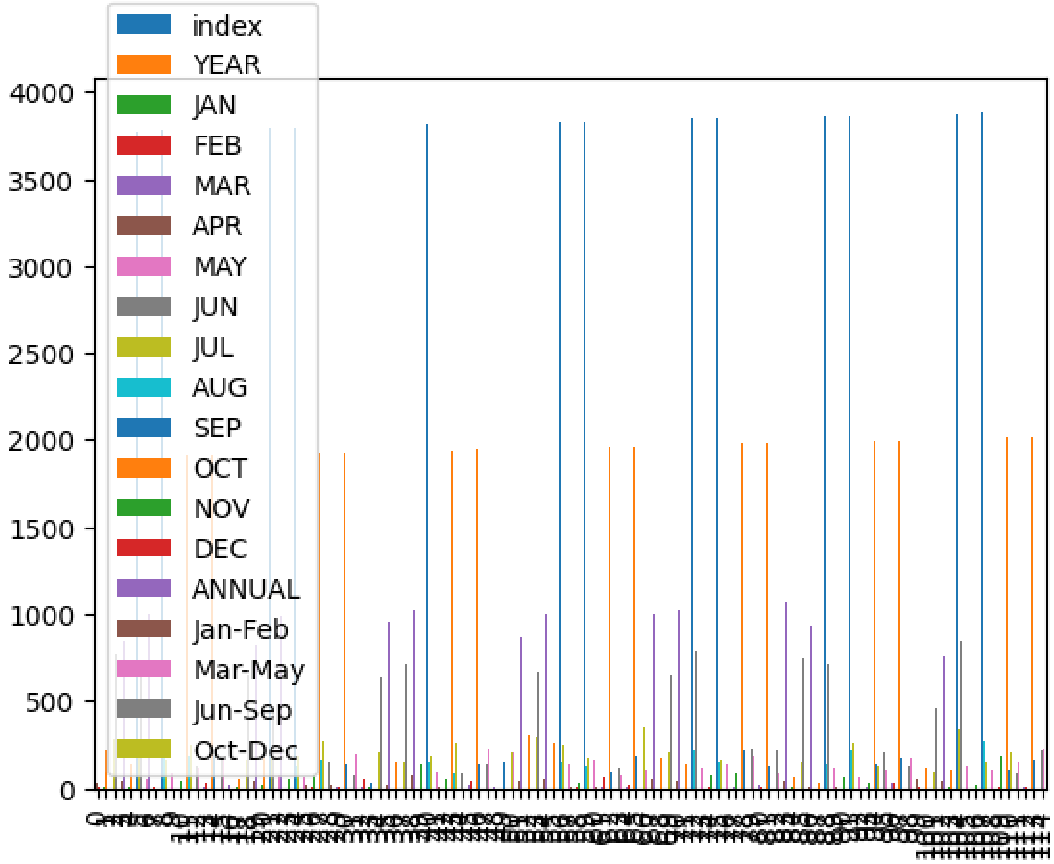
[7]:
```
<Axes: >
```



6 Bar chart

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

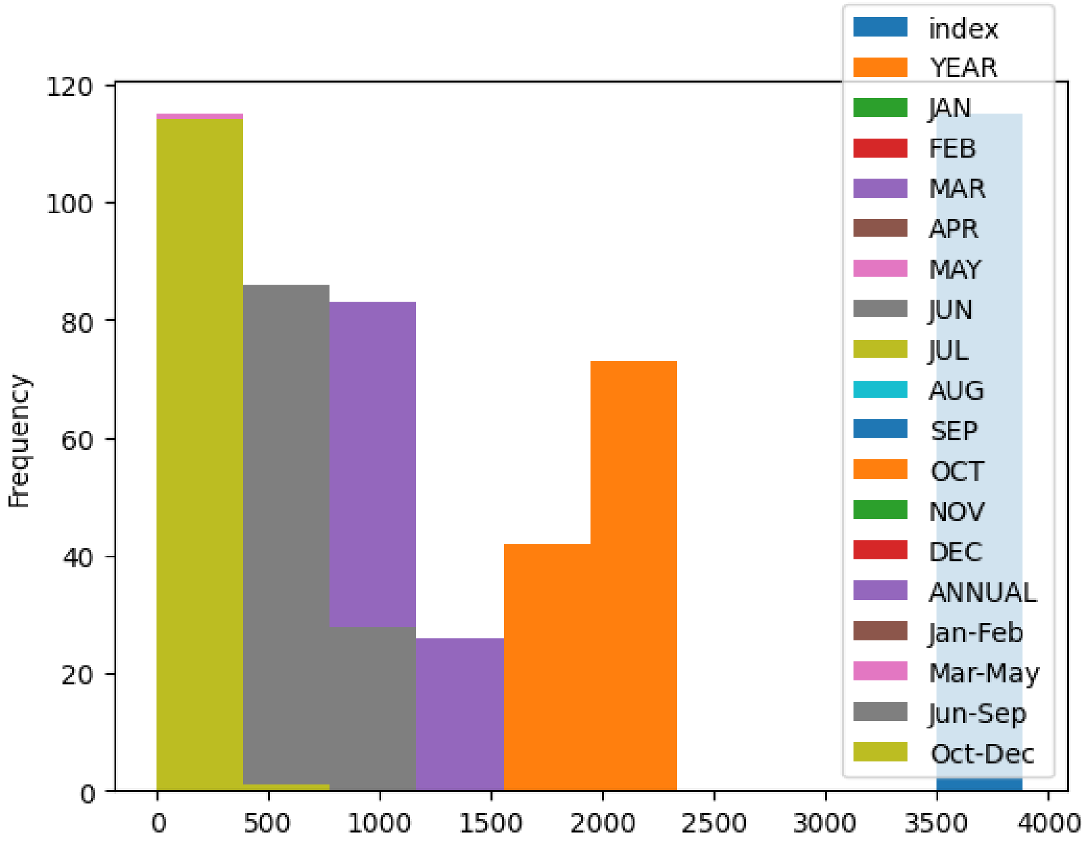
[8]:
```
<Axes: >
```



7 Histogram

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

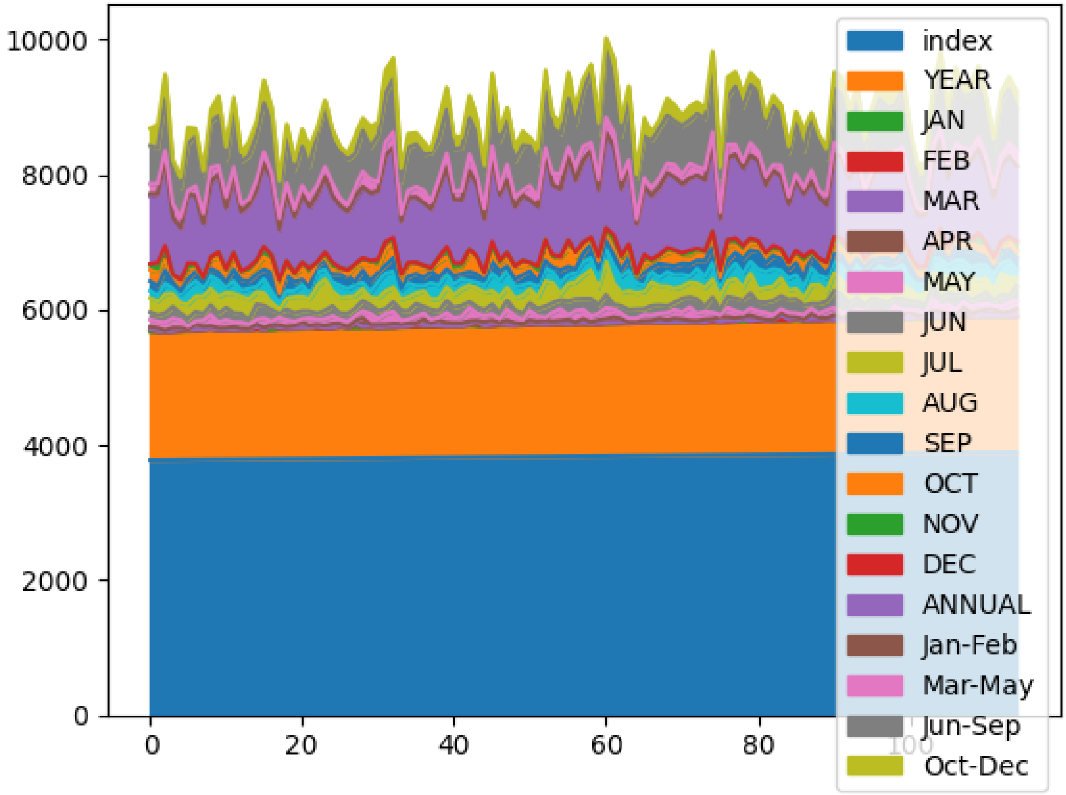
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

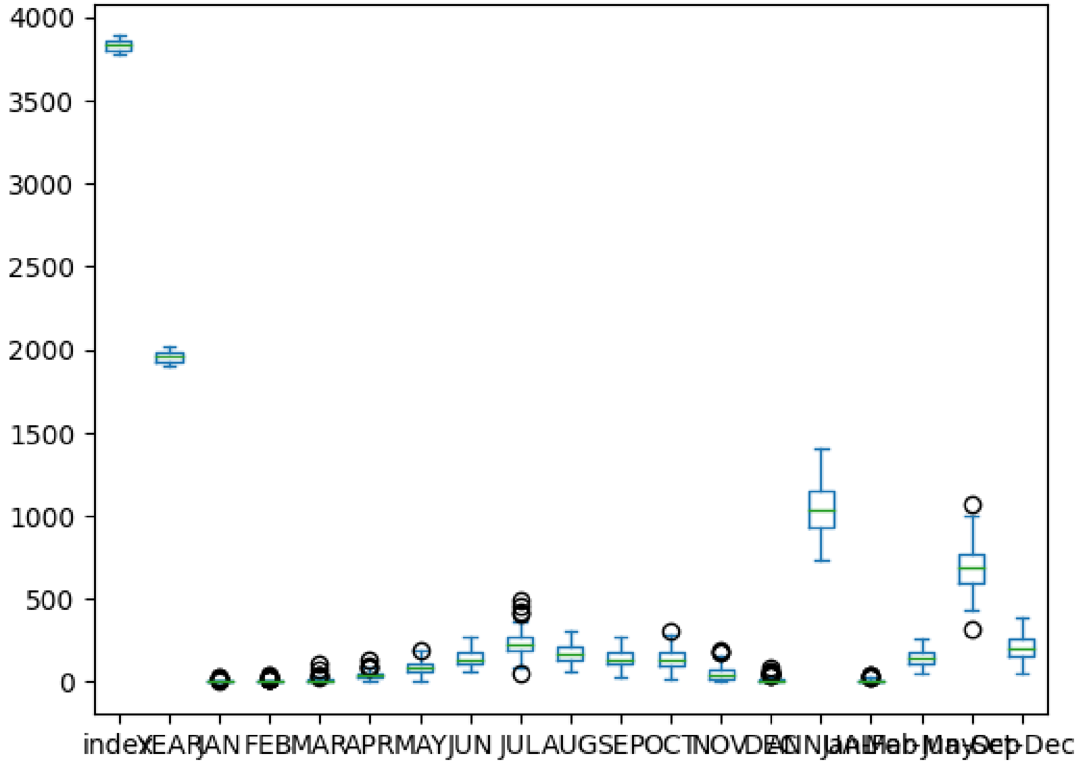
```
[10]: <Axes: >
```



9 Box chart

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

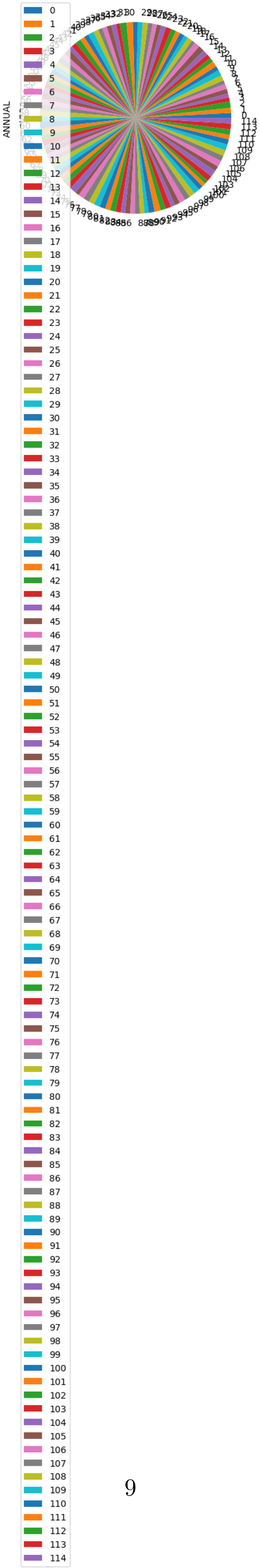
```
[11]: <Axes: >
```



10 Pie chart

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

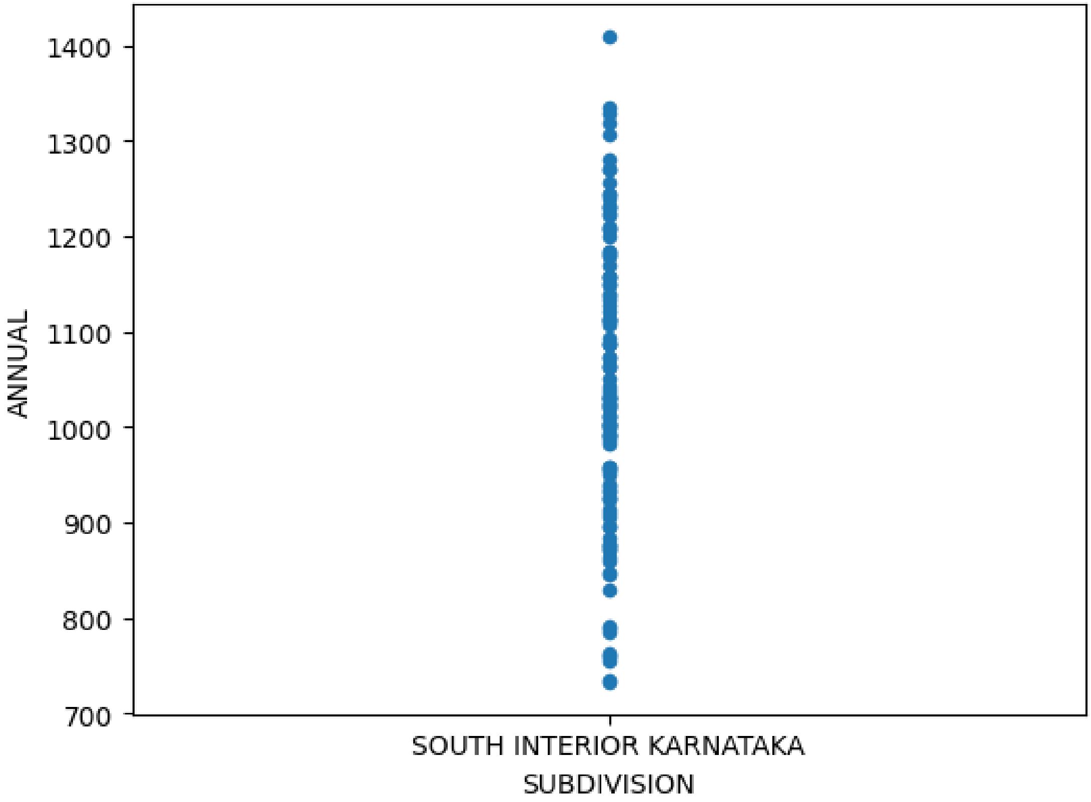
[12]:
```
<Axes: ylabel='ANNUAL'>
```


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')
```

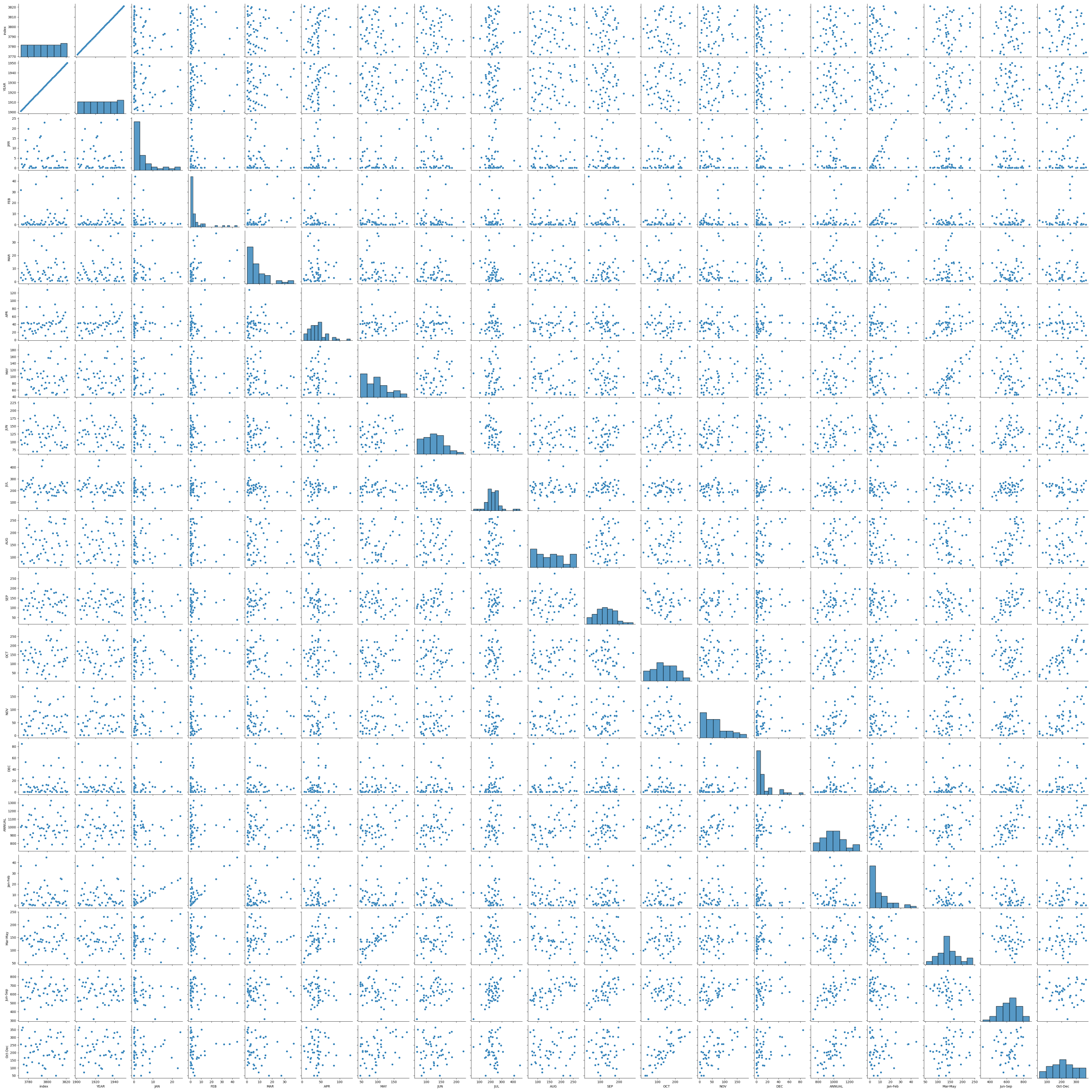
[13]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

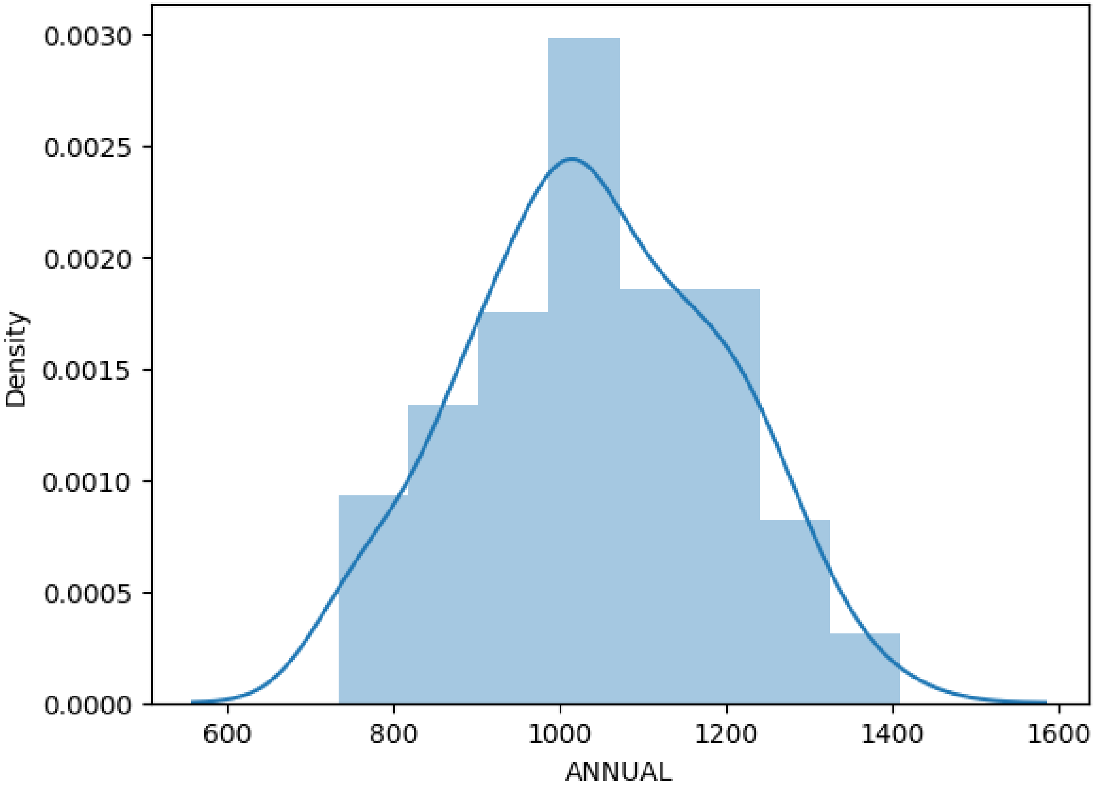
[14]:
```
<seaborn.axisgrid.PairGrid at 0x7926ad2d3a00>
```



[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'])
```

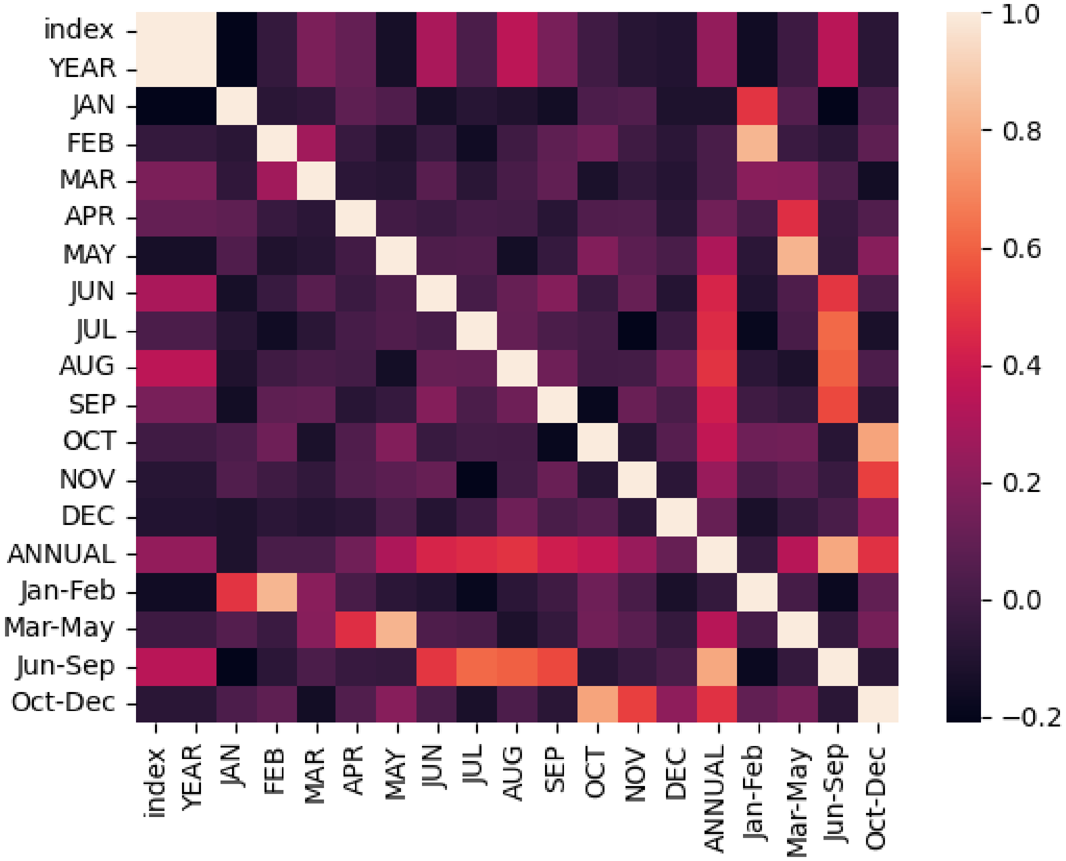
```
[15]: <Axes: xlabel='ANNUAL', ylabel='Density'>
```



```
[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