xrx3hptg3

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_sub␣
  ↪himalayan west bengal & sikkim.csv")
df
```

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

[2]:

```
     index                     SUBDIVISION  YEAR   JAN   FEB   MAR    APR  \
0      437  SUB HIMALAYAN WEST BENGAL & SIKKIM  1901  26.5  14.8  14.1   29.2
1      438  SUB HIMALAYAN WEST BENGAL & SIKKIM  1902   1.2   0.7  87.1  126.1
2      439  SUB HIMALAYAN WEST BENGAL & SIKKIM  1903   5.5   8.7  19.6   18.6
3      440  SUB HIMALAYAN WEST BENGAL & SIKKIM  1904   3.4  29.2   0.9  124.3
4      441  SUB HIMALAYAN WEST BENGAL & SIKKIM  1905  12.0  31.2  51.9  104.4
..     …                                 …    …     …     …     …      …
110    547  SUB HIMALAYAN WEST BENGAL & SIKKIM  2011   8.5  19.9  71.2  135.0
111    548  SUB HIMALAYAN WEST BENGAL & SIKKIM  2012  15.3  13.9  45.5  159.8
112    549  SUB HIMALAYAN WEST BENGAL & SIKKIM  2013   3.0  23.6  32.1  114.7
113    550  SUB HIMALAYAN WEST BENGAL & SIKKIM  2014   0.2  26.6  37.7   47.9
114    551  SUB HIMALAYAN WEST BENGAL & SIKKIM  2015  15.7  15.0  64.8  149.0

       MAY    JUN    JUL     AUG    SEP    OCT   NOV   DEC  ANNUAL  Jan-Feb  \
0    195.5  488.4  524.8   501.1  242.7   55.5  17.9   2.6  2113.2     41.3
1    271.3  539.2  671.0   603.8  799.9   74.4   5.6   0.0  3180.4      1.9
2    163.6  541.2  431.5   708.8  365.2  141.3   0.3   0.0  2404.5     14.3
3    333.6  274.2  500.4   468.5  260.6  164.8   8.9   1.1  2169.9     32.5
4    290.6  524.8  523.1  1036.6  321.1   87.9   2.7  18.7  3005.0     43.2
..     …      …      …       …      …      …     …     …       …        …
```

```
110  247.8  419.8  612.3   470.3  356.3   46.7  26.7   4.3  2418.7     28.4
111  202.4  604.2  684.5   332.7  434.7  119.4  12.5   7.4  2632.2     29.2
112  296.5  404.9  588.4   416.3  308.0  199.8  16.1   2.7  2406.1     26.7
113  308.6  543.2  384.6   563.3  371.5   31.2   5.3   2.4  2322.6     26.9
114  304.6  508.2  393.3   626.6  354.9   53.6  23.8   9.0  2518.6     30.7

     Mar-May  Jun-Sep  Oct-Dec
0      238.9   1757.0     76.1
1      484.6   2613.9     80.1
2      201.9   2046.7    141.6
3      458.8   1503.7    174.8
4      447.0   2405.6    109.3
..       …        …      …
110    453.9   1858.6     77.7
111    407.7   2056.0    139.3
112    443.4   1717.6    218.5
113    394.2   1862.6     38.9
114    518.5   1883.0     86.4

[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
```

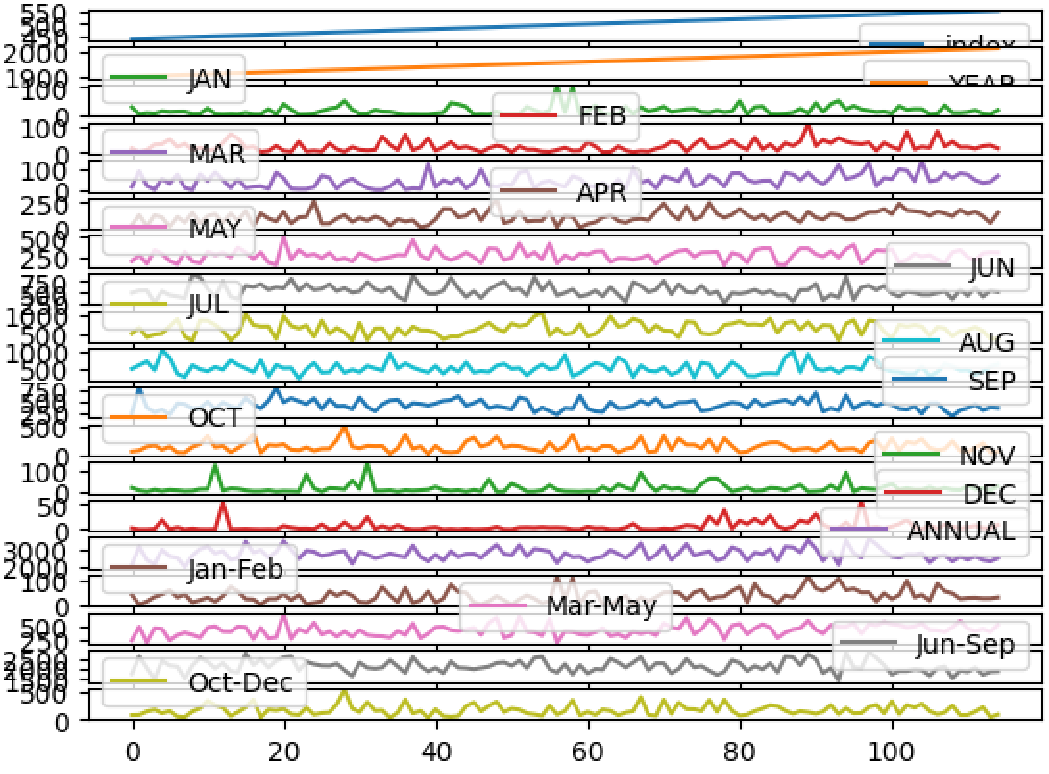
4 Line chart

[6]:

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

[6]:

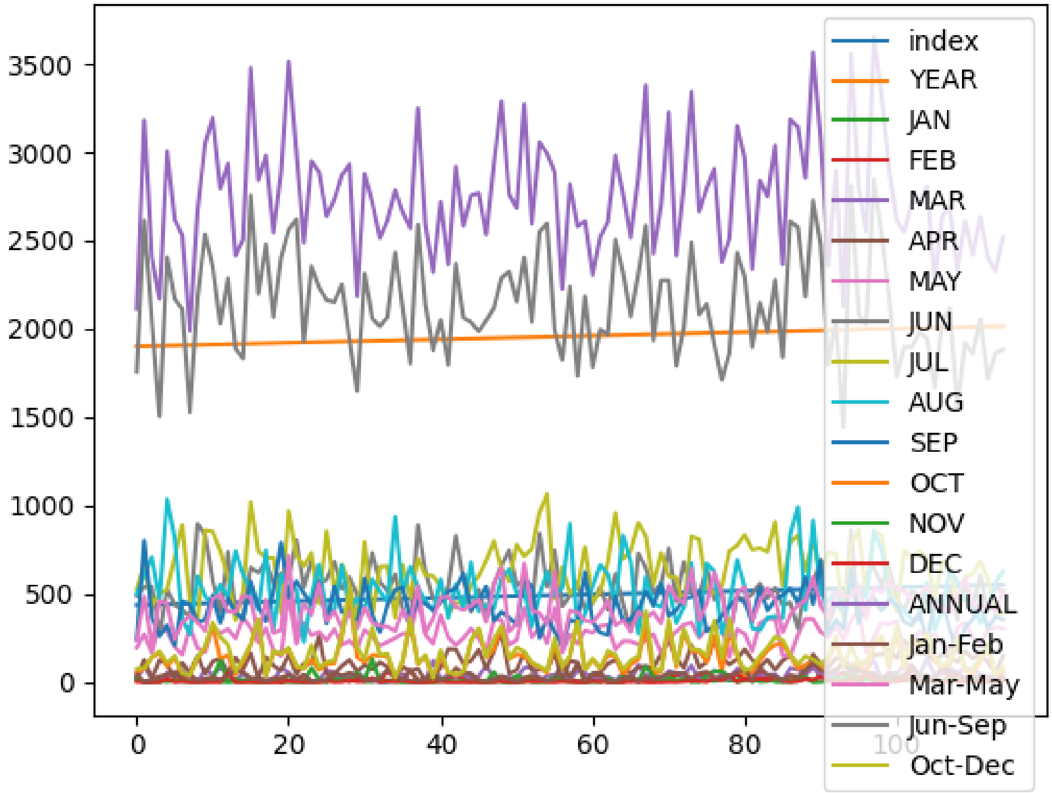
```
array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
       <Axes: >], dtype=object)
```



5 Line chart

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

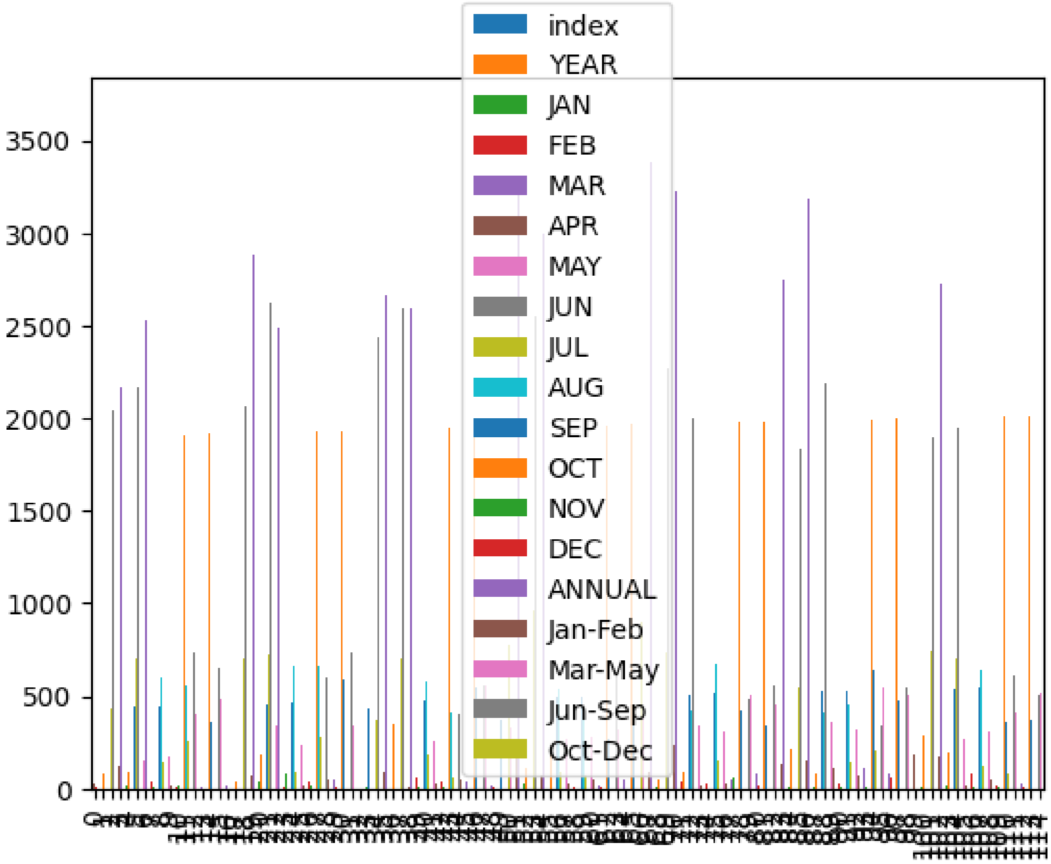
[7]:
```
<Axes: >
```



6 Bar chart

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

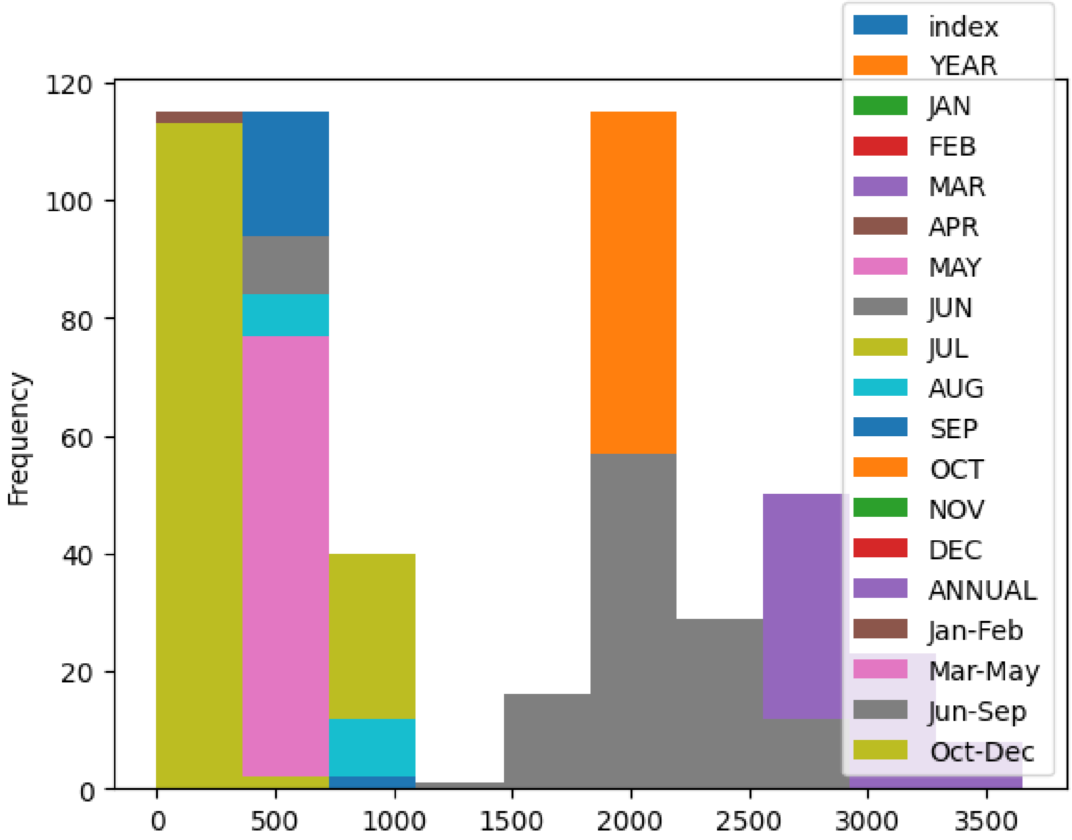
[8]:
```
<Axes: >
```



7 Histogram

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

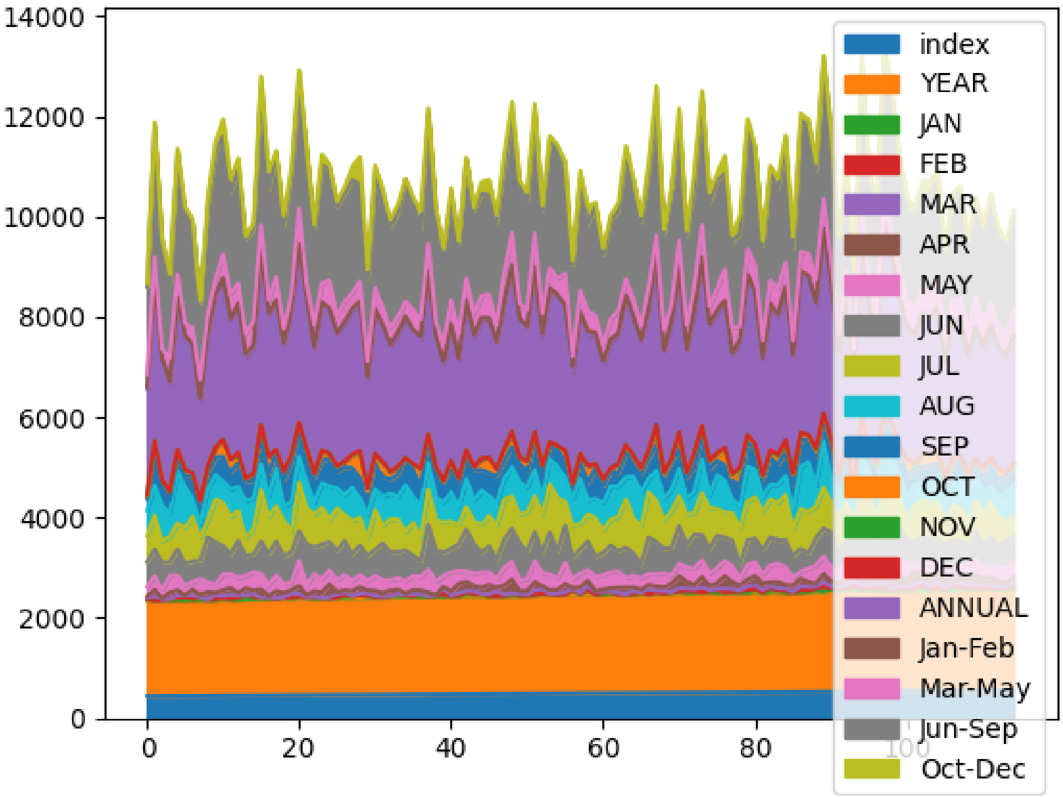
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

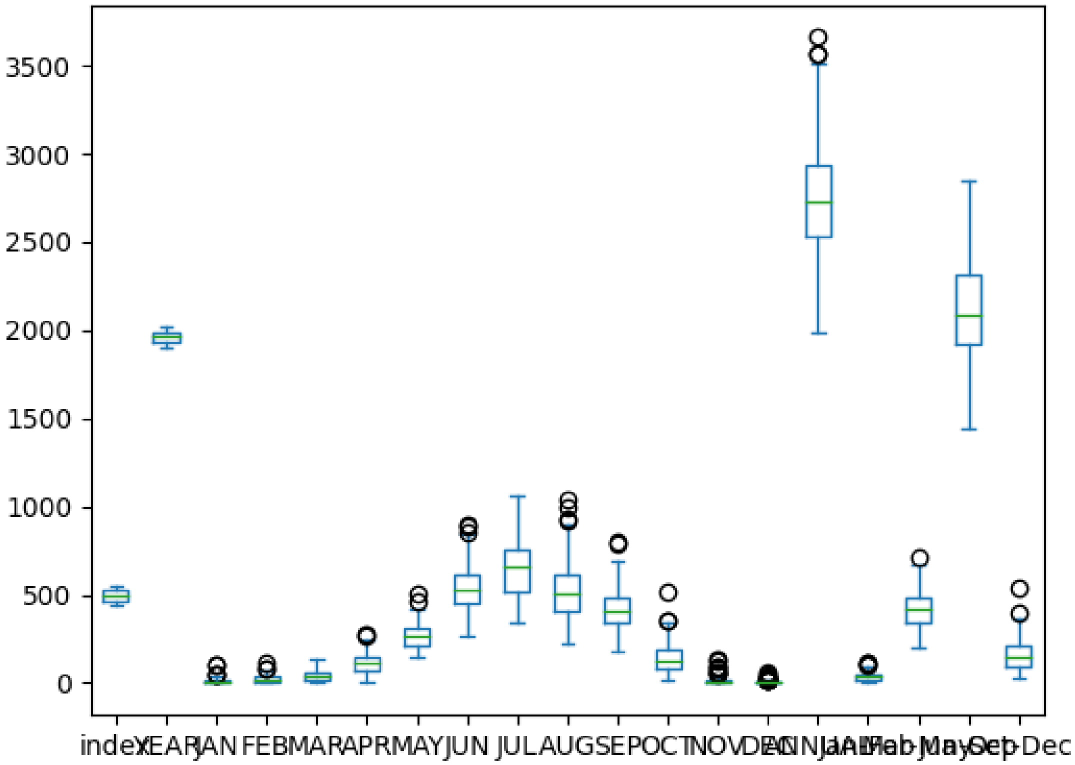
```
[10]: <Axes: >
```



9 Box chart

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

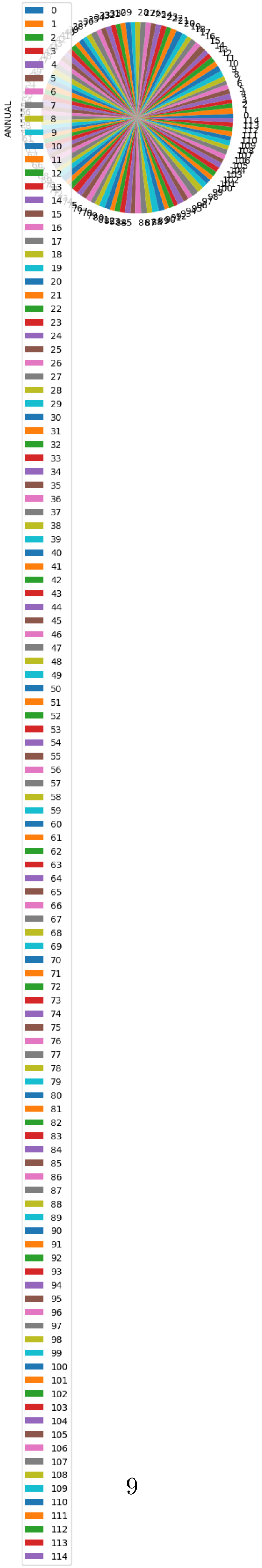
[11]:
```
<Axes: >
```



10 Pie chart

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

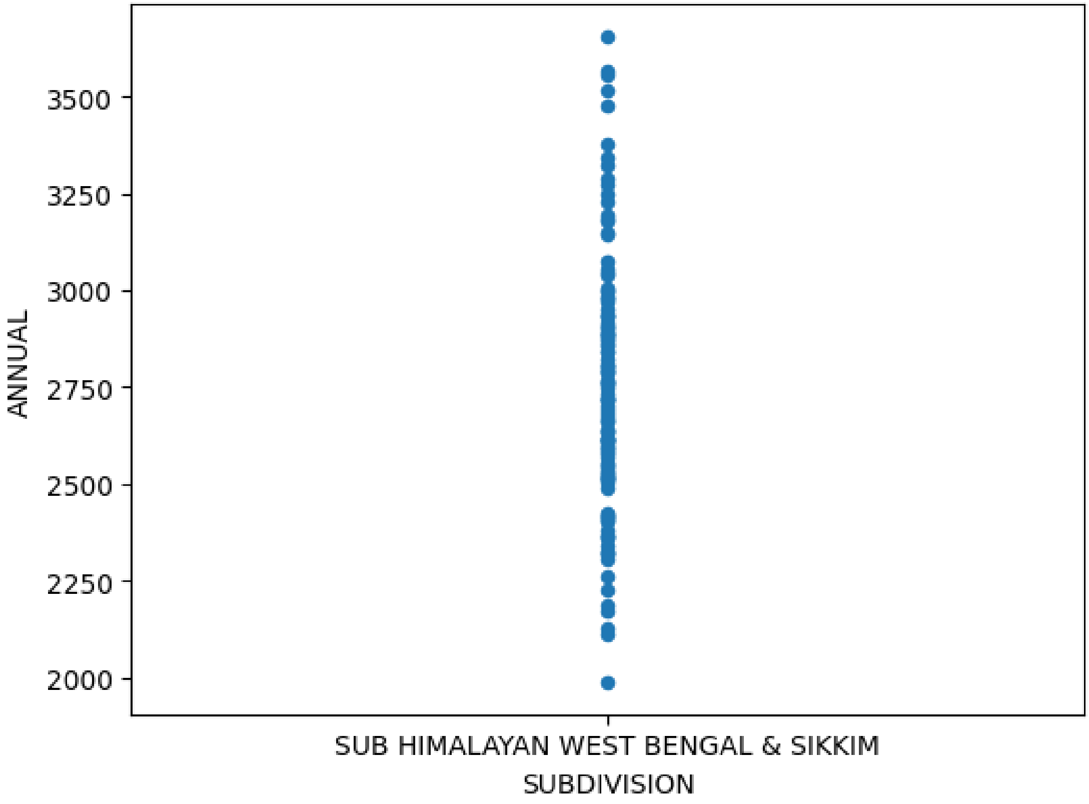
[12]:
```
<Axes: ylabel='ANNUAL'>
```


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')
```

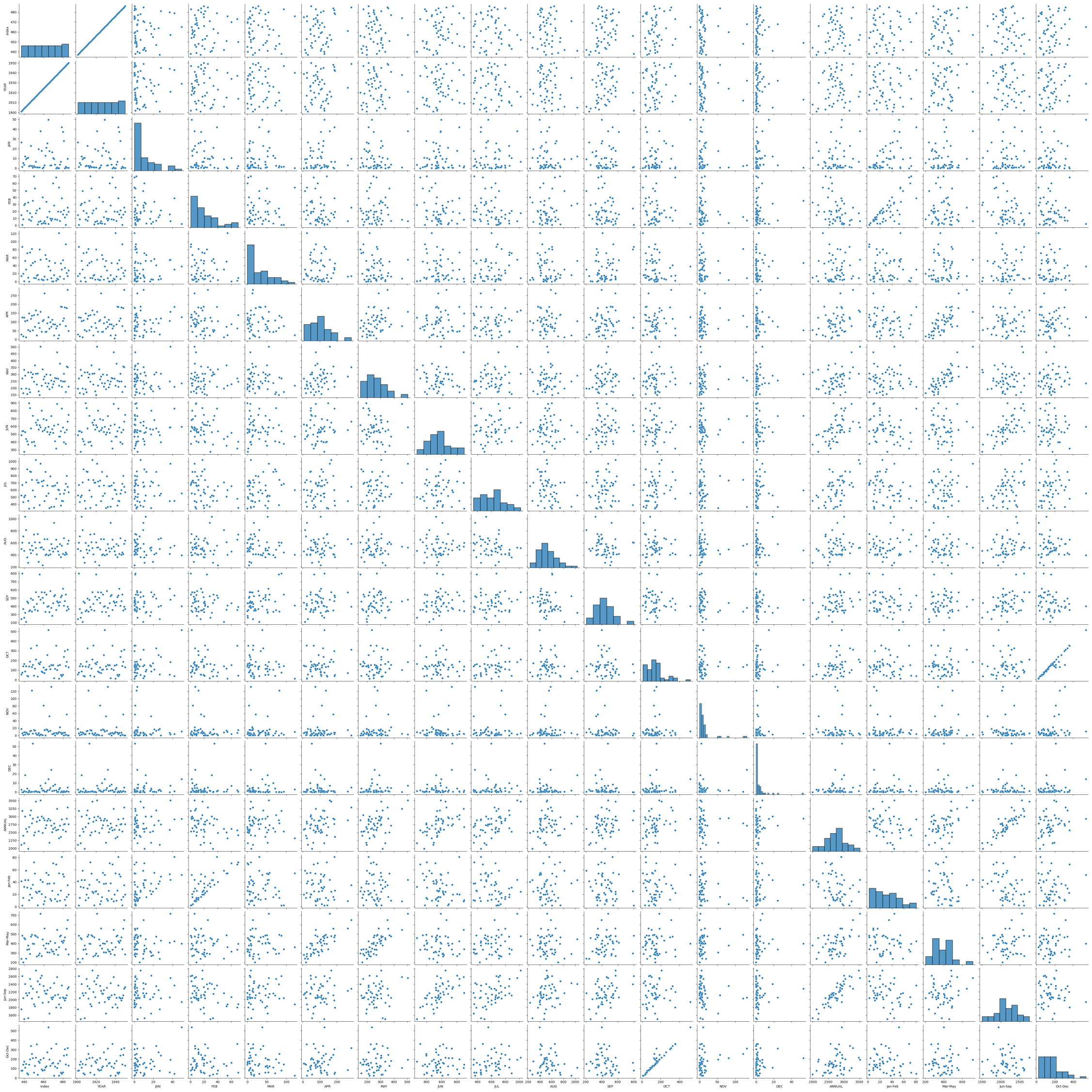
[13]:
```
<Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

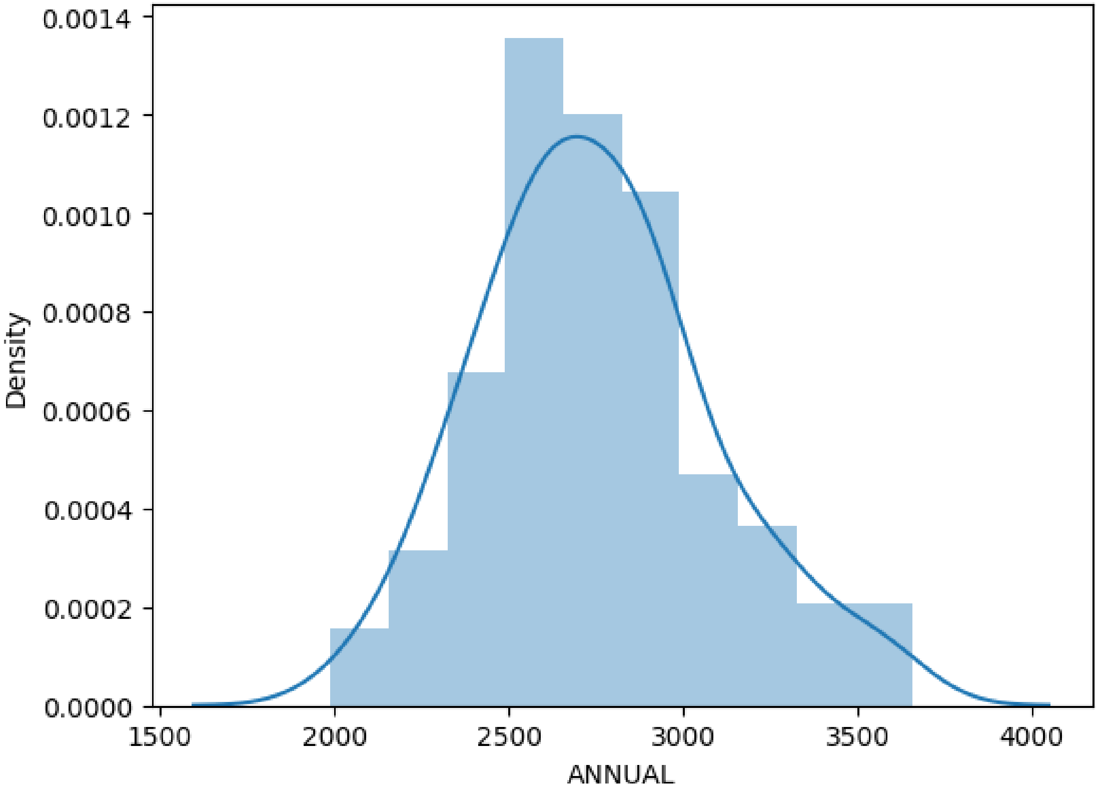
[14]:
```
<seaborn.axisgrid.PairGrid at 0x7b05976a69b0>
```



```
[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'])
```

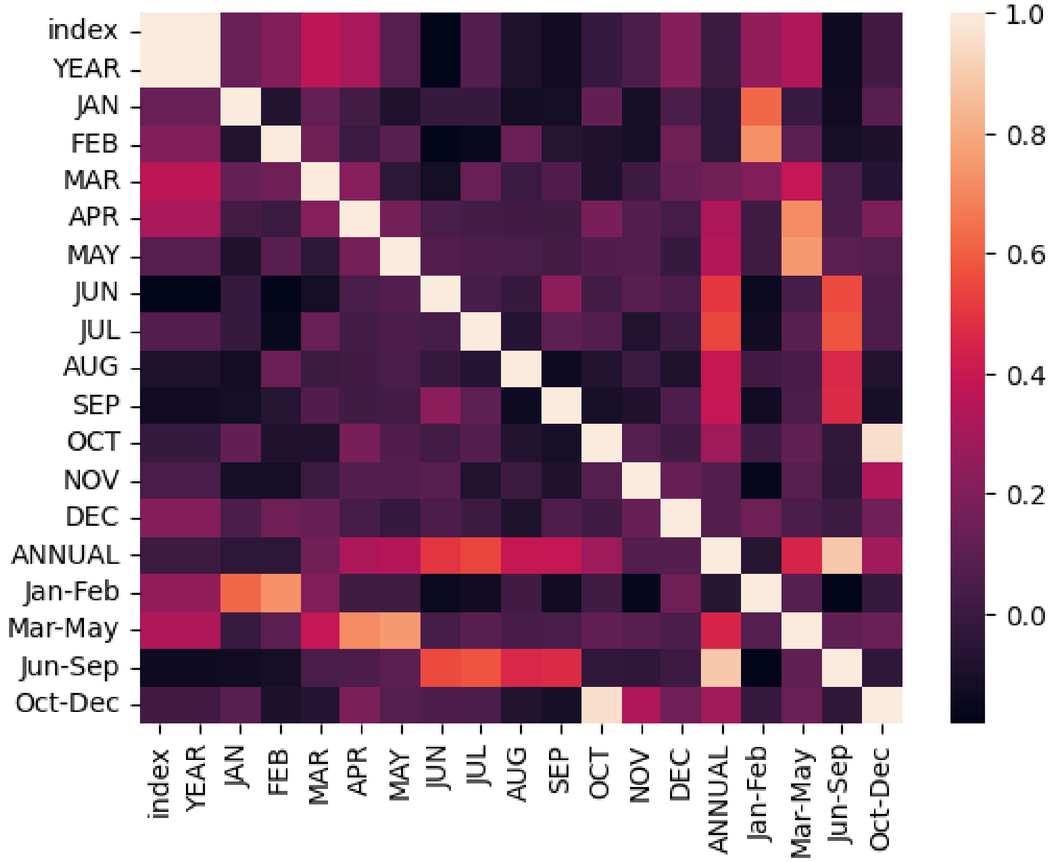
```
[15]: <Axes: xlabel='ANNUAL', ylabel='Density'>
```



```
[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