

qojxtiiec

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_jammu &
↳kashmir.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	\		
0	1702	JAMMU & KASHMIR	1901	66.4	69.3	69.6	132.2	105.8	53.4			
1	1703	JAMMU & KASHMIR	1902	6.5	9.7	91.3	100.5	70.7	113.3			
2	1704	JAMMU & KASHMIR	1903	96.2	21.5	238.6	58.7	57.3	18.9			
3	1705	JAMMU & KASHMIR	1904	110.6	17.3	145.2	64.5	67.8	25.9			
4	1706	JAMMU & KASHMIR	1905	146.7	76.3	161.4	71.7	65.2	43.3			
..			
110	1812	JAMMU & KASHMIR	2011	43.4	211.6	97.8	89.0	32.4	72.5			
111	1813	JAMMU & KASHMIR	2012	150.9	95.8	45.2	86.6	48.9	32.6			
112	1814	JAMMU & KASHMIR	2013	52.2	136.4	41.9	47.4	47.4	80.5			
113	1815	JAMMU & KASHMIR	2014	75.8	64.0	153.1	76.1	52.7	25.3			
114	1816	JAMMU & KASHMIR	2015	27.9	187.2	341.4	173.3	64.6	121.4			
		JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	\
0	171.7	181.3	101.8	24.1	0.0	4.4	979.9	135.6	307.6	508.2		
1	108.4	136.9	62.2	15.1	44.0	1.8	760.5	16.2	262.5	420.9		
2	332.5	218.6	176.9	10.7	15.0	41.8	1286.6	117.7	354.6	746.9		
3	182.3	132.2	62.3	50.0	24.8	99.2	982.2	128.0	277.5	402.7		
4	145.2	111.5	239.7	5.8	0.6	90.2	1157.7	223.0	298.4	539.7		
..		

110	81.6	131.2	72.0	19.4	12.9	23.8	887.6	254.9	219.2	357.3
111	118.8	264.9	106.7	15.7	10.8	57.8	1034.7	246.7	180.7	523.0
112	125.1	219.1	41.2	34.4	13.4	20.3	859.3	188.6	136.7	465.8
113	100.5	134.6	362.8	32.2	14.1	2.3	1093.4	139.8	281.9	623.2
114	233.2	129.2	130.2	87.1	38.1	39.3	1572.8	215.1	579.3	614.0

	Oct-Dec
0	28.5
1	60.9
2	67.5
3	174.0
4	96.7
..	...
110	56.1
111	84.3
112	68.1
113	48.5
114	164.5

[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
```

```

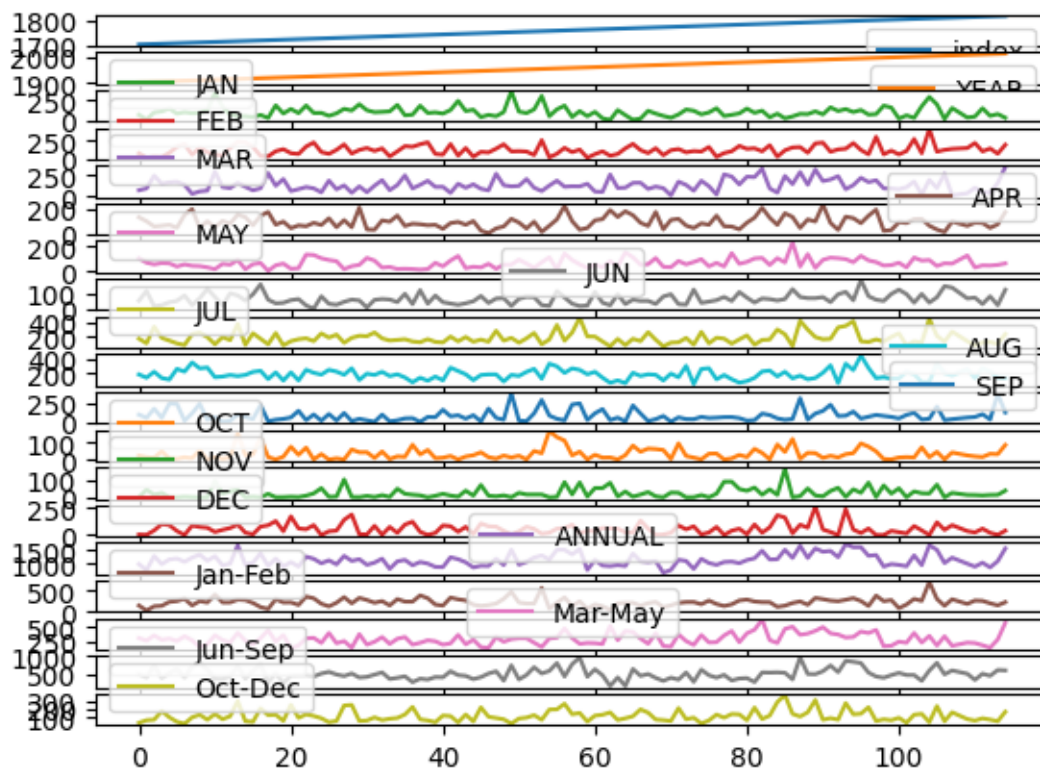
7  MAY          114 non-null    float64
8  JUN          114 non-null    float64
9  JUL          114 non-null    float64
10 AUG          114 non-null    float64
11 SEP          114 non-null    float64
12 OCT          114 non-null    float64
13 NOV          114 non-null    float64
14 DEC          114 non-null    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)
```

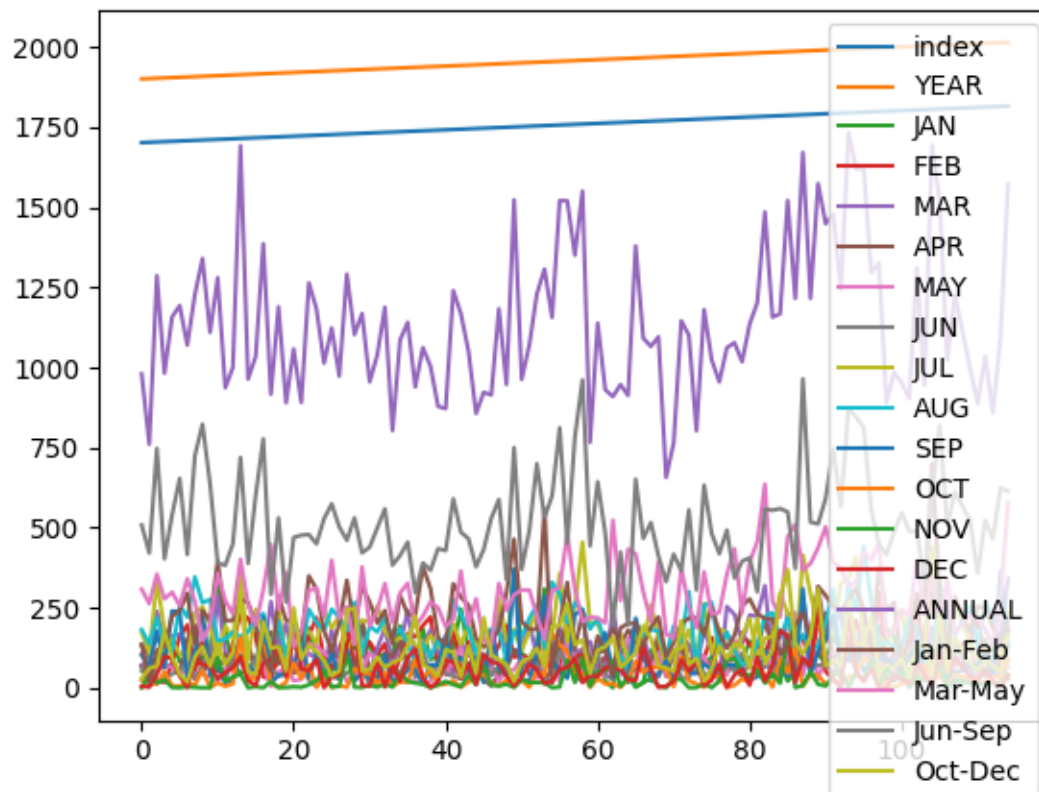
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



5 Line chart

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

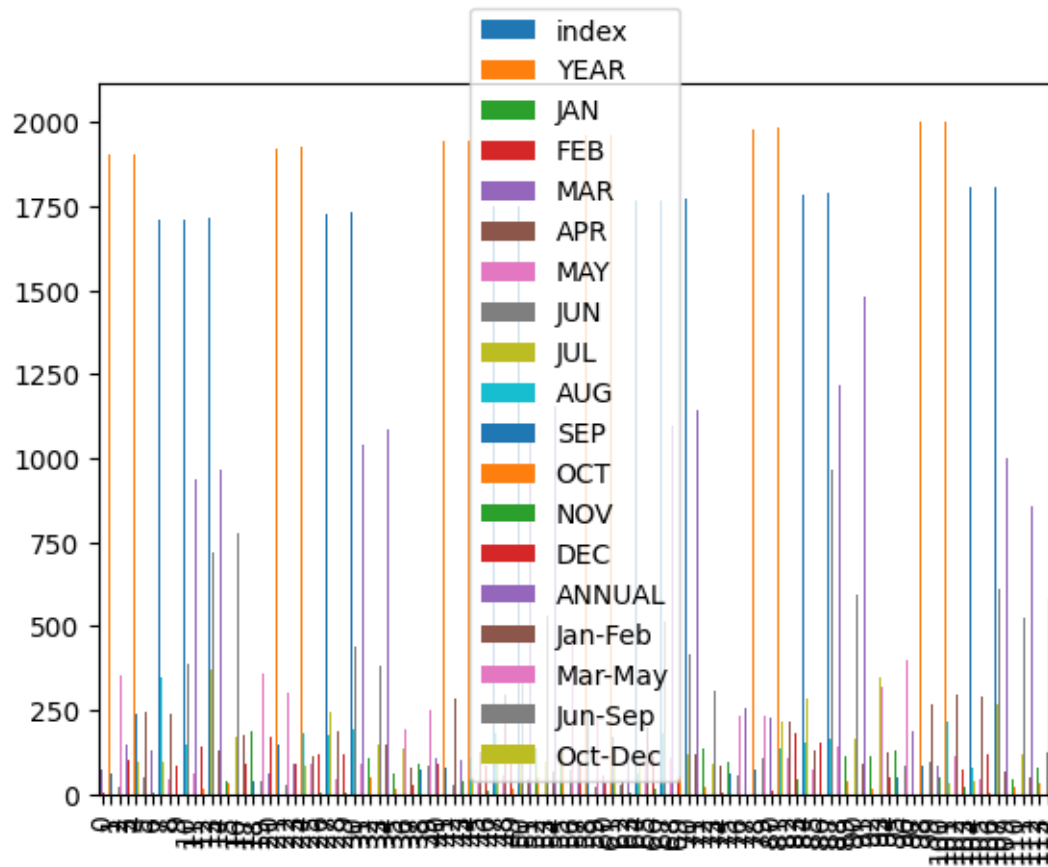
```
[7]: <Axes: >
```



6 Bar chart

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

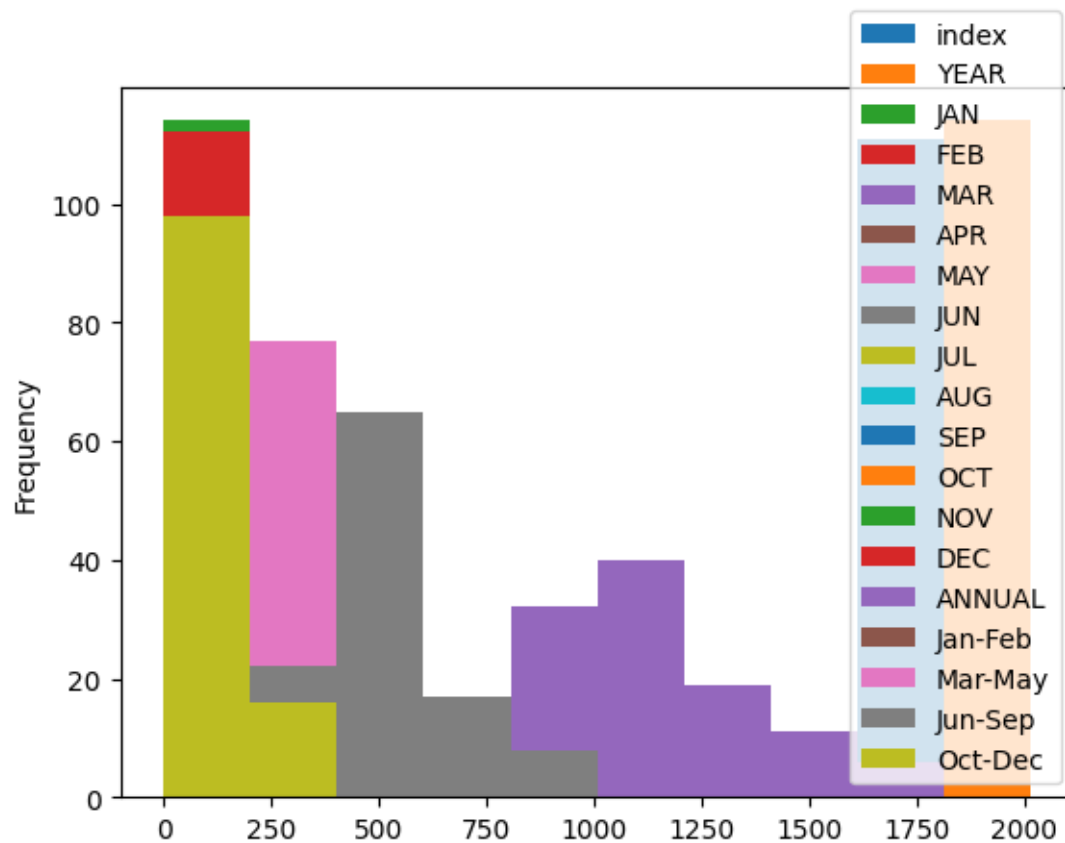
```
[8]: <Axes: >
```



7 Histogram

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

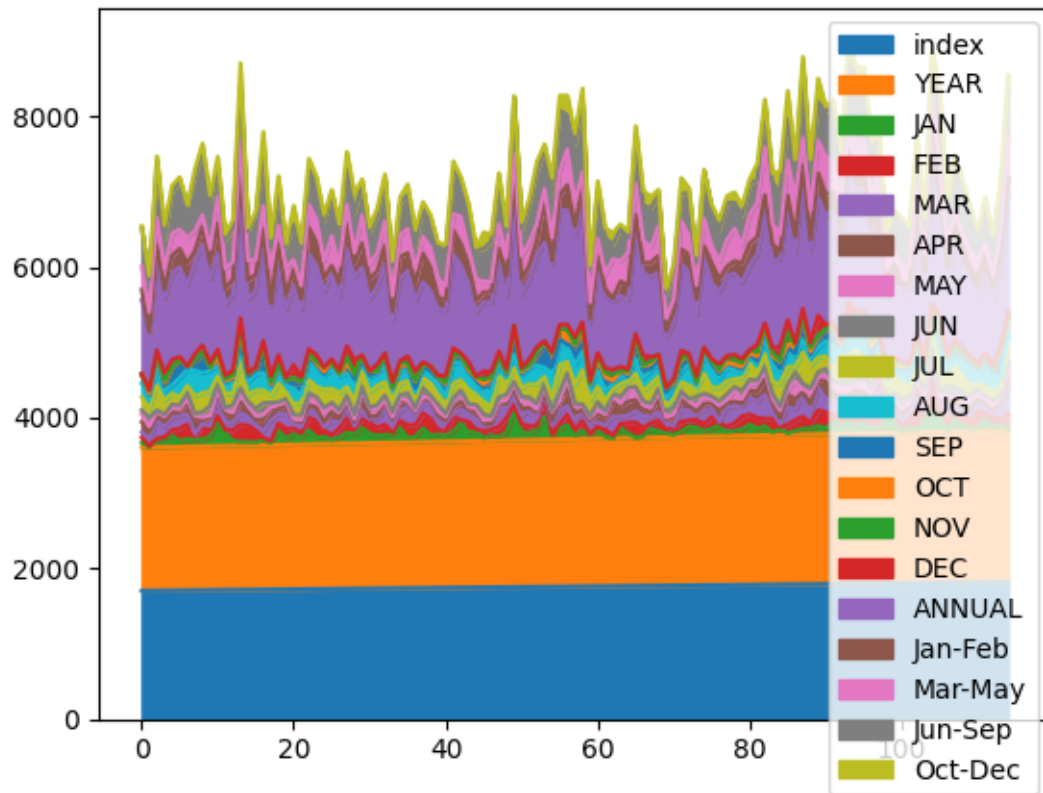
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

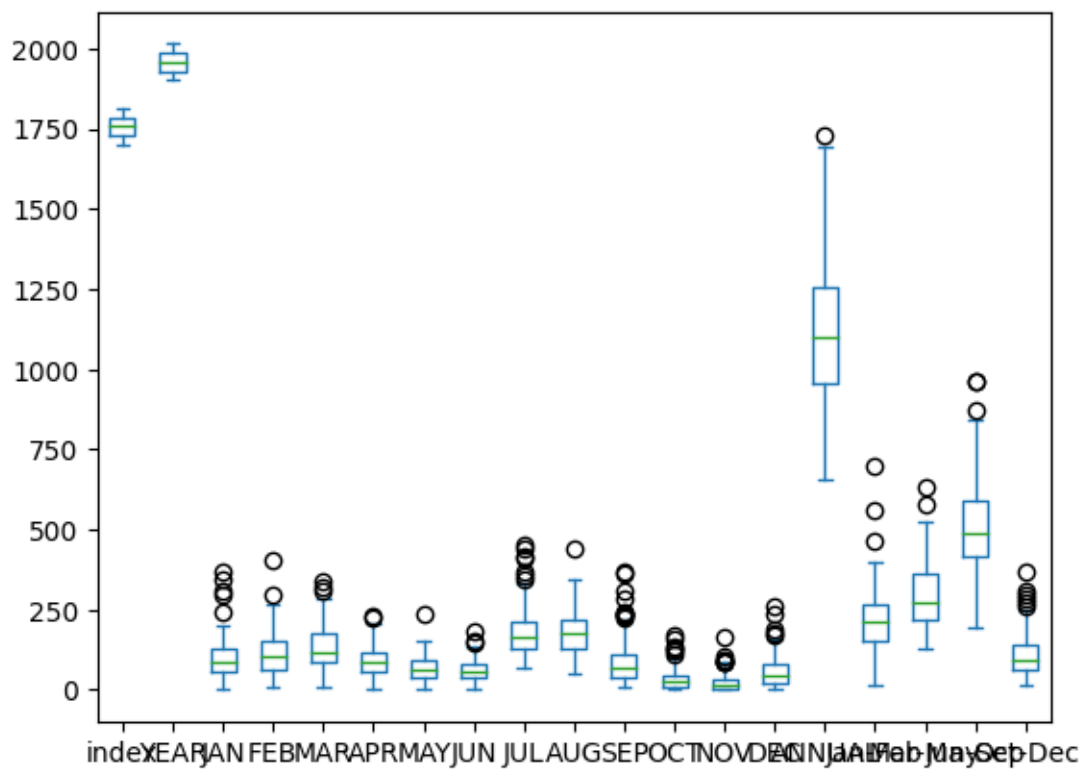
```
[10]: <Axes: >
```



9 Box chart

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

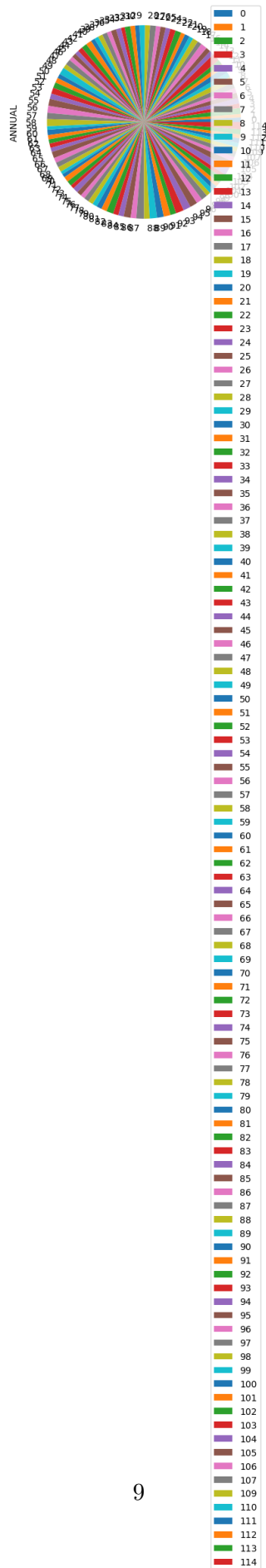
```
[11]: <Axes: >
```



10 Pie chart

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

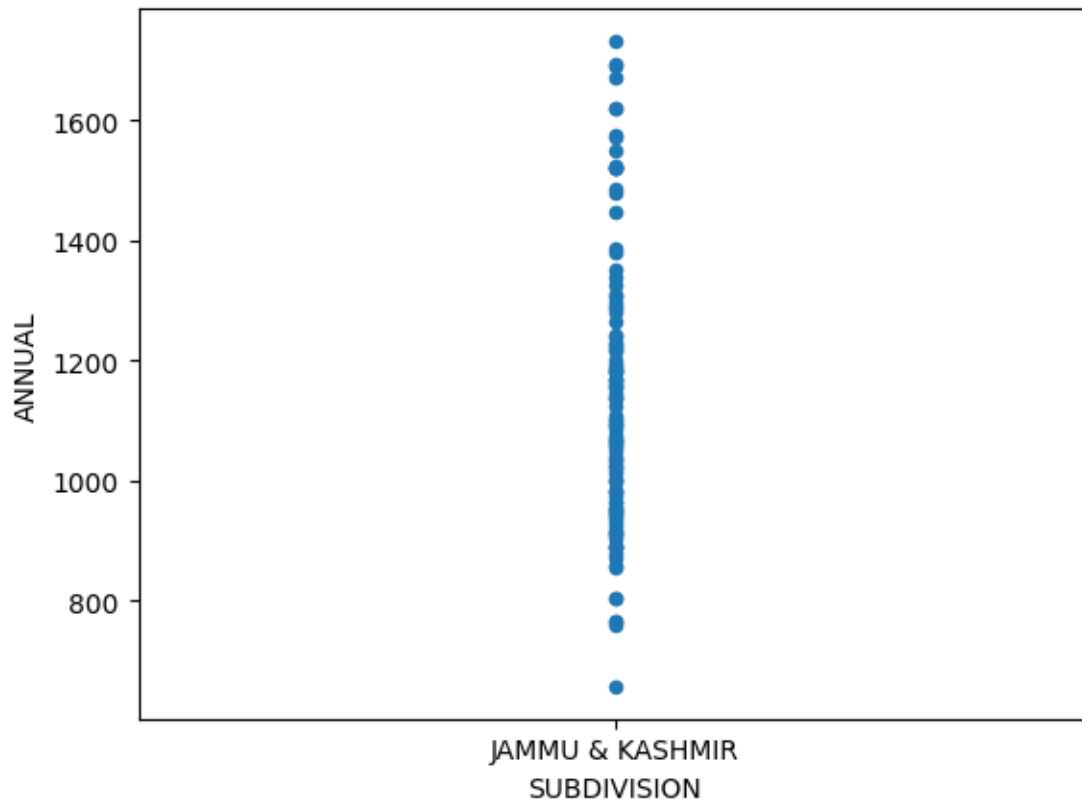
```
[12]: <Axes: ylabel='ANNUAL'>
```

11 Scatter chart

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

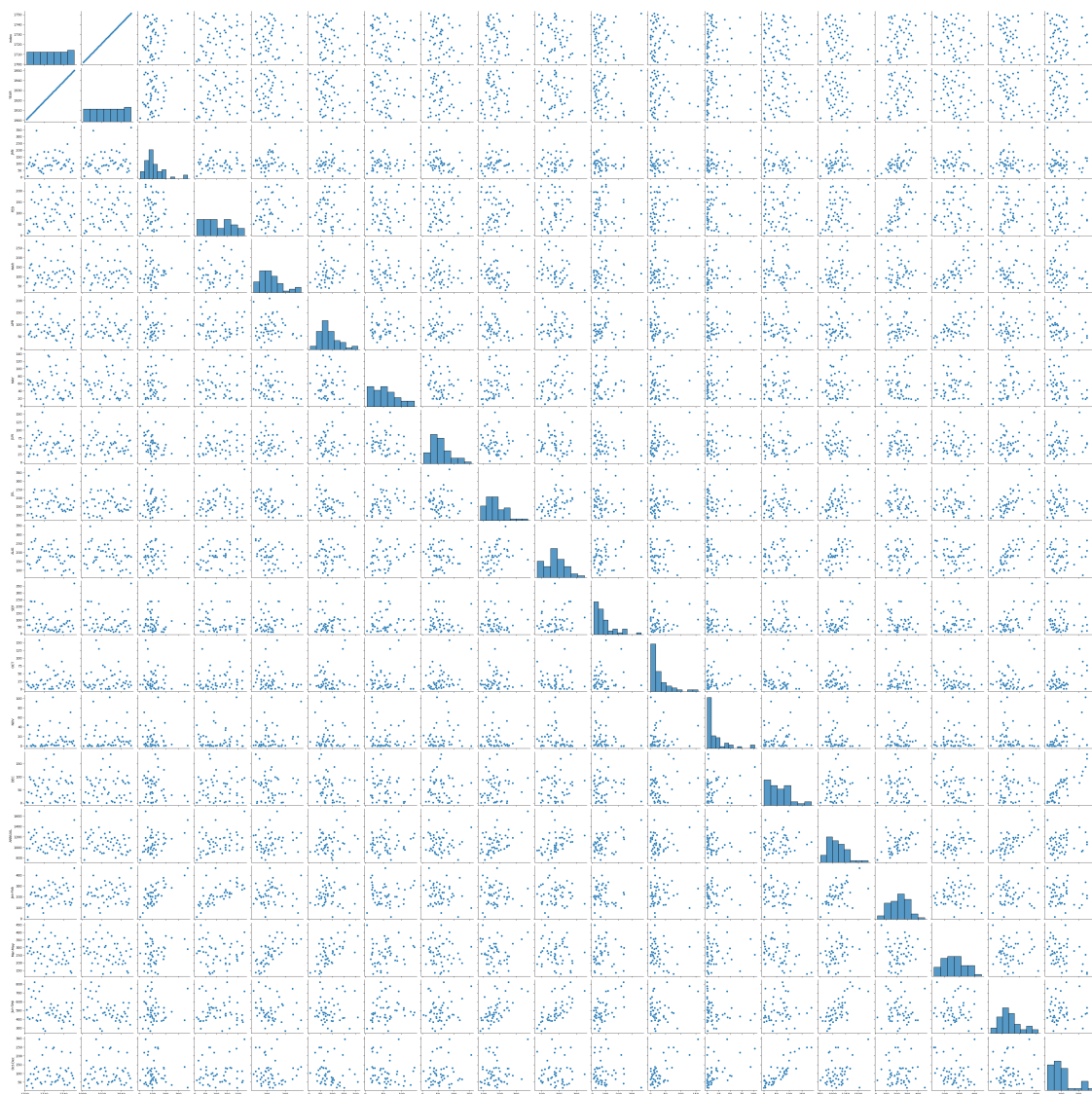
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7ed8a62ffa00>
```



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

<ipython-input-15-5daa97052ca5>:1: UserWarning:

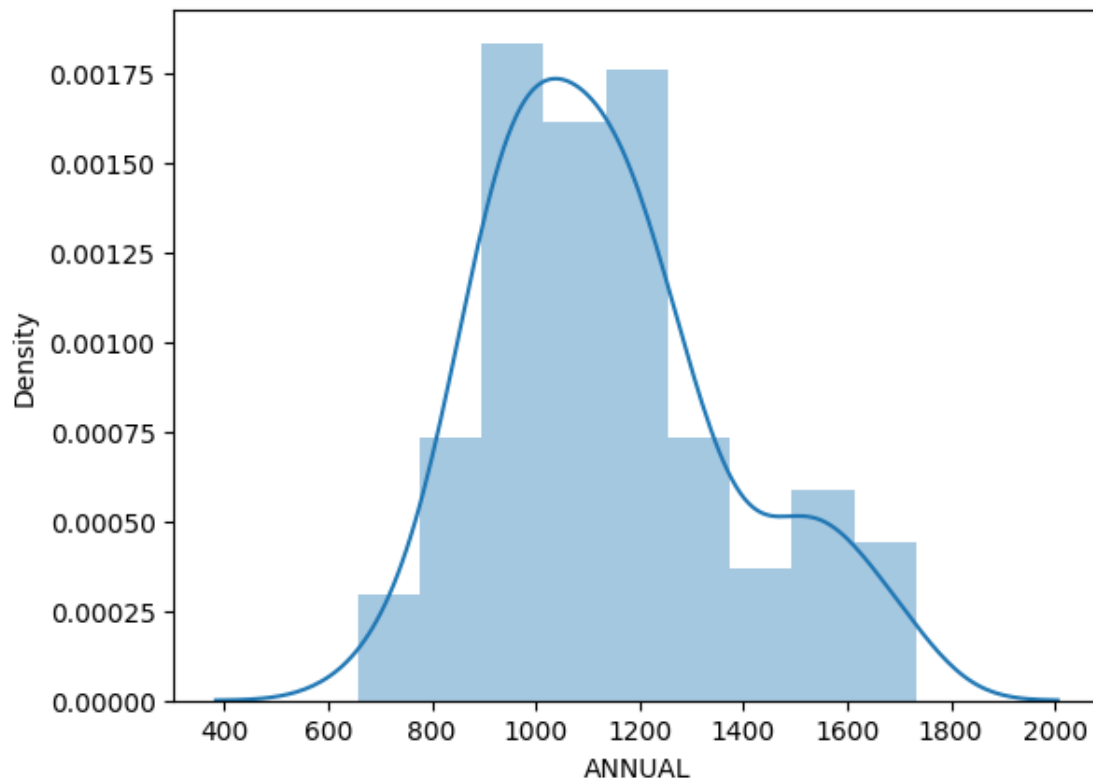
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

