

bwp3127y2

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_gangetic_
↪west_bengal.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index		SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	\
0	552	GANGETIC	WEST BENGAL	1901	37.1	58.4	3.9	64.1	121.7	198.0	
1	553	GANGETIC	WEST BENGAL	1902	0.0	1.2	44.2	103.8	161.6	140.9	
2	554	GANGETIC	WEST BENGAL	1903	17.5	24.6	37.3	30.6	78.5	201.7	
3	555	GANGETIC	WEST BENGAL	1904	0.1	23.9	35.6	17.5	160.2	286.7	
4	556	GANGETIC	WEST BENGAL	1905	30.9	49.6	84.7	84.9	156.8	70.9	
..			
110	662	GANGETIC	WEST BENGAL	2011	2.5	2.7	40.5	75.0	132.6	434.5	
111	663	GANGETIC	WEST BENGAL	2012	40.7	15.3	4.4	57.7	44.2	146.6	
112	664	GANGETIC	WEST BENGAL	2013	2.5	10.0	4.8	45.6	195.9	233.4	
113	665	GANGETIC	WEST BENGAL	2014	0.9	42.2	19.9	1.9	124.4	193.6	
114	666	GANGETIC	WEST BENGAL	2015	12.9	5.5	19.3	88.7	57.6	247.2	
		JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	\
0	280.8	275.7	313.5	51.1	83.4	0.0	1487.6		95.5	189.7	
1	347.8	264.8	230.5	32.5	10.4	9.9	1347.7		1.2	309.6	
2	179.6	277.6	300.7	198.0	8.2	0.0	1354.3		42.1	146.4	
3	435.3	241.7	142.8	35.1	4.1	0.1	1383.1		24.0	213.3	
4	525.5	263.6	287.6	107.3	0.0	5.2	1666.9		80.5	326.3	
..				

110	219.9	443.2	295.9	36.9	1.3	1.4	1686.5	5.2	248.1
111	315.0	261.4	246.9	64.2	47.0	24.6	1268.0	56.0	106.3
112	263.2	401.4	254.0	353.2	0.0	0.0	1764.1	12.5	246.3
113	298.7	292.6	229.5	56.9	0.1	0.6	1261.4	43.1	146.2
114	633.1	260.6	164.0	32.7	2.3	6.3	1530.3	18.4	165.6

	Jun-Sep	Oct-Dec
0	1068.0	134.4
1	984.0	52.9
2	959.6	206.2
3	1106.4	39.3
4	1147.6	112.6
..
110	1393.6	39.6
111	969.9	135.8
112	1152.0	353.3
113	1014.4	57.7
114	1304.9	41.3

[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
```

```

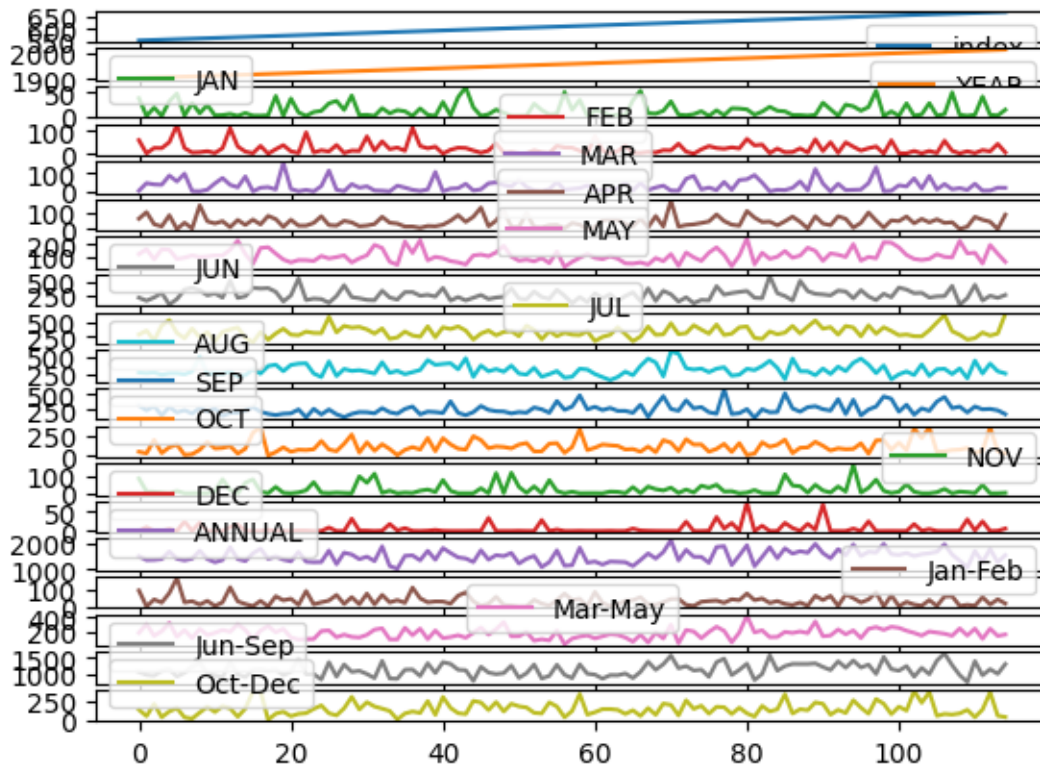
7   MAY          115 non-null    float64
8   JUN          115 non-null    float64
9   JUL          115 non-null    float64
10  AUG          115 non-null    float64
11  SEP          115 non-null    float64
12  OCT          115 non-null    float64
13  NOV          115 non-null    float64
14  DEC          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)
```

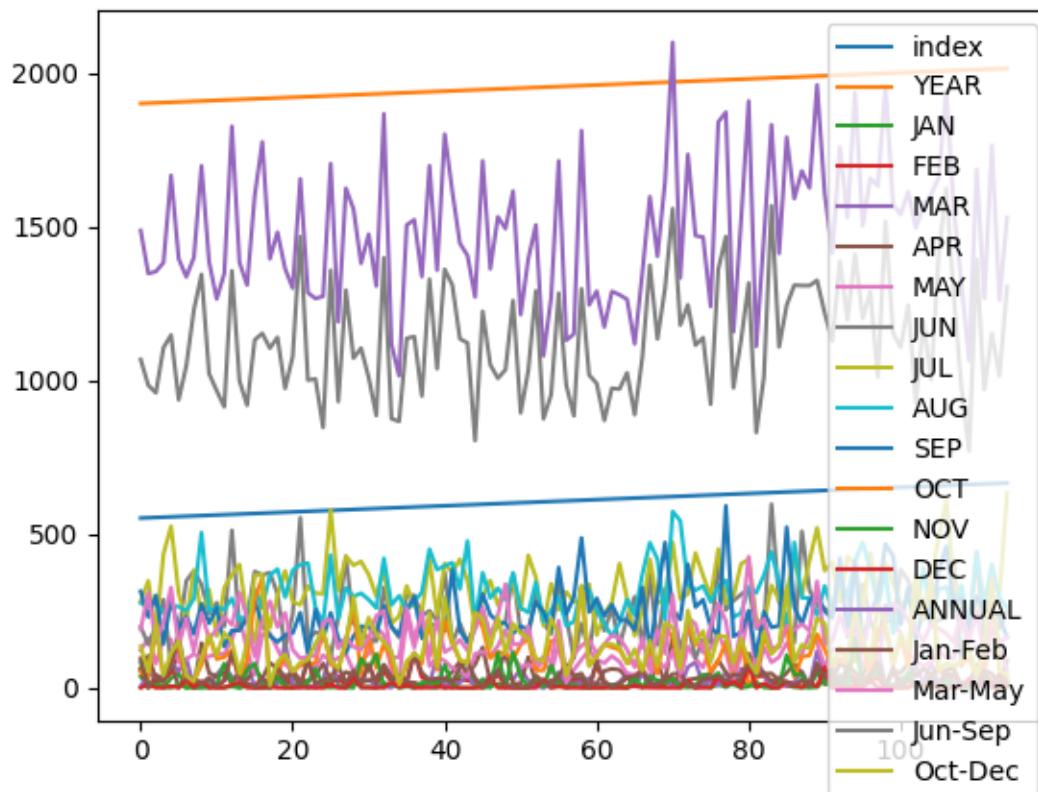
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



5 Line chart

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

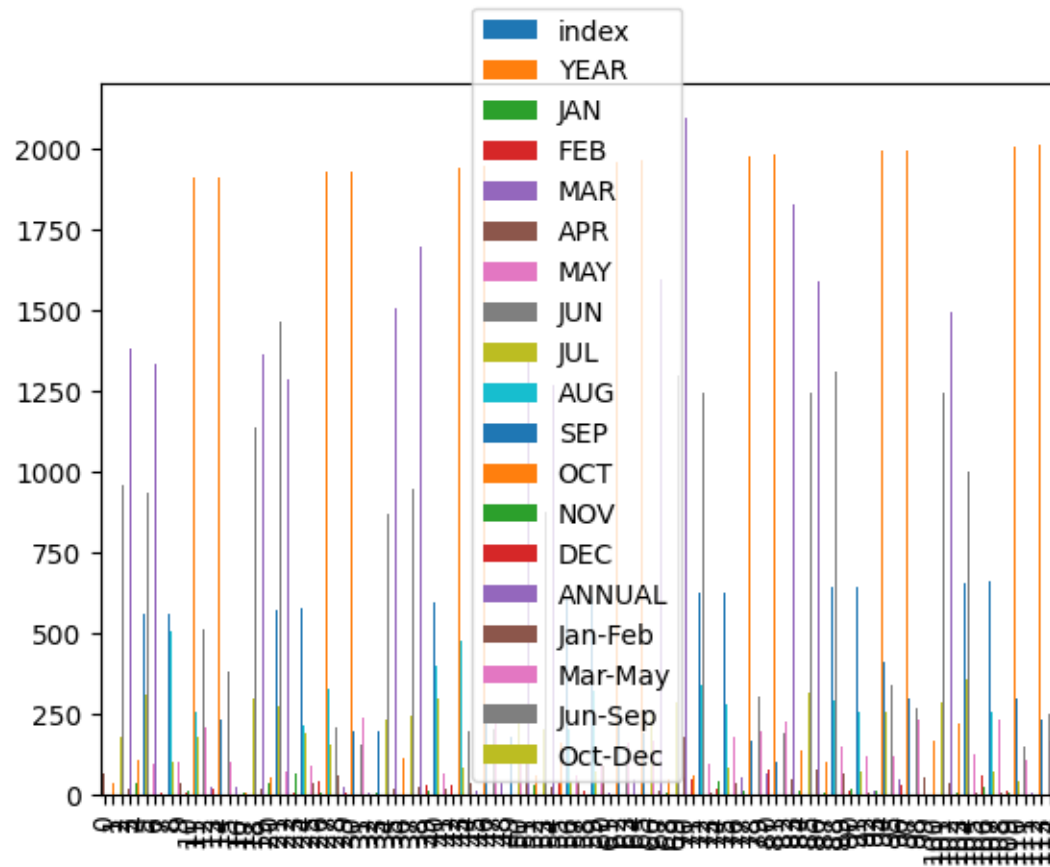
```
[7]: <Axes: >
```



6 Bar chart

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

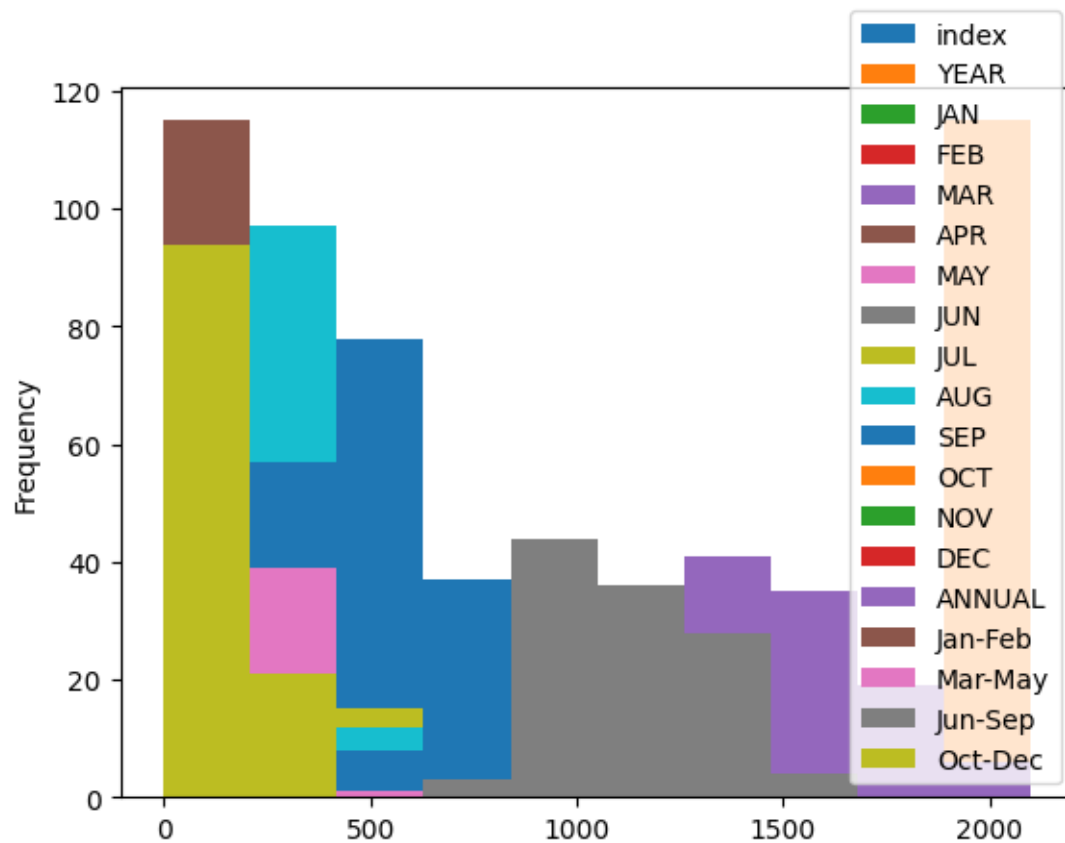
```
[8]: <Axes: >
```



7 Histogram

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

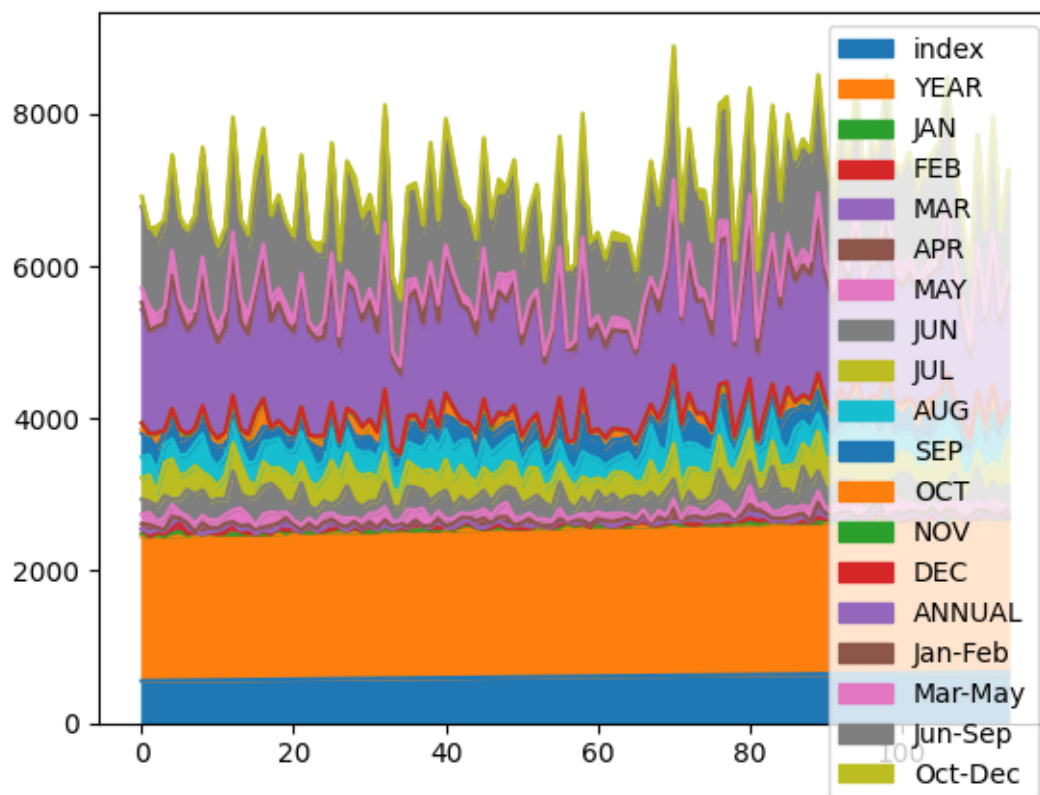
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

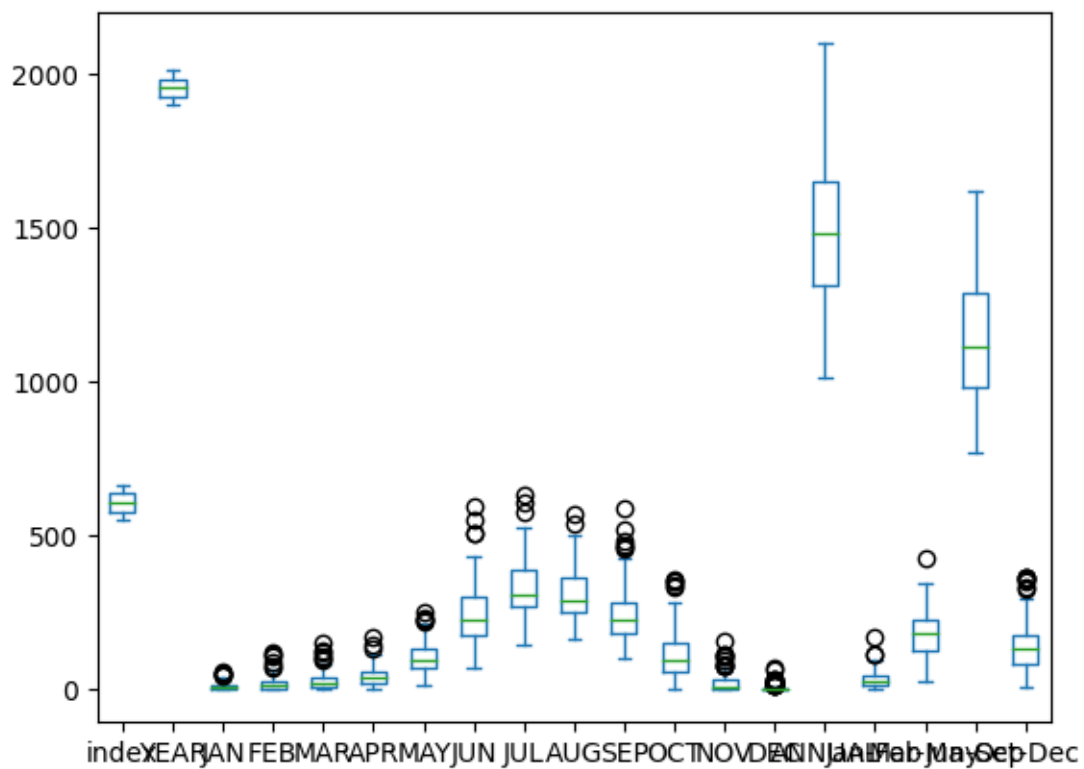
```
[10]: <Axes: >
```



9 Box chart

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

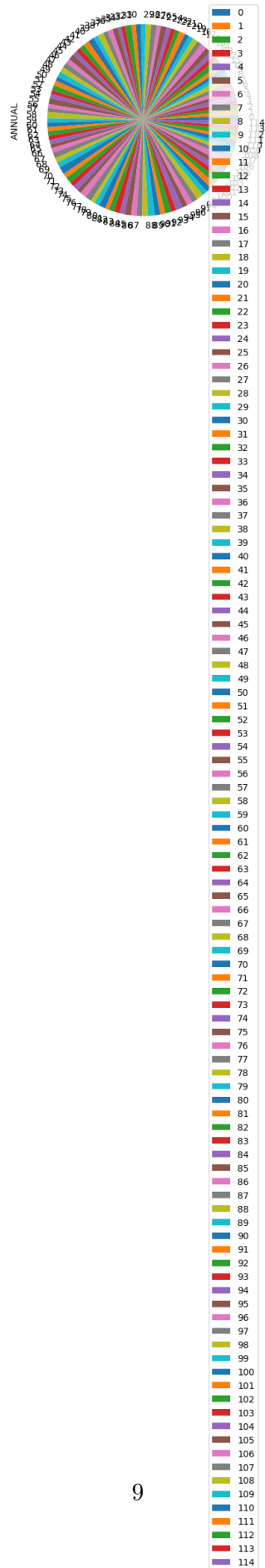
```
[11]: <Axes: >
```



10 Pie chart

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

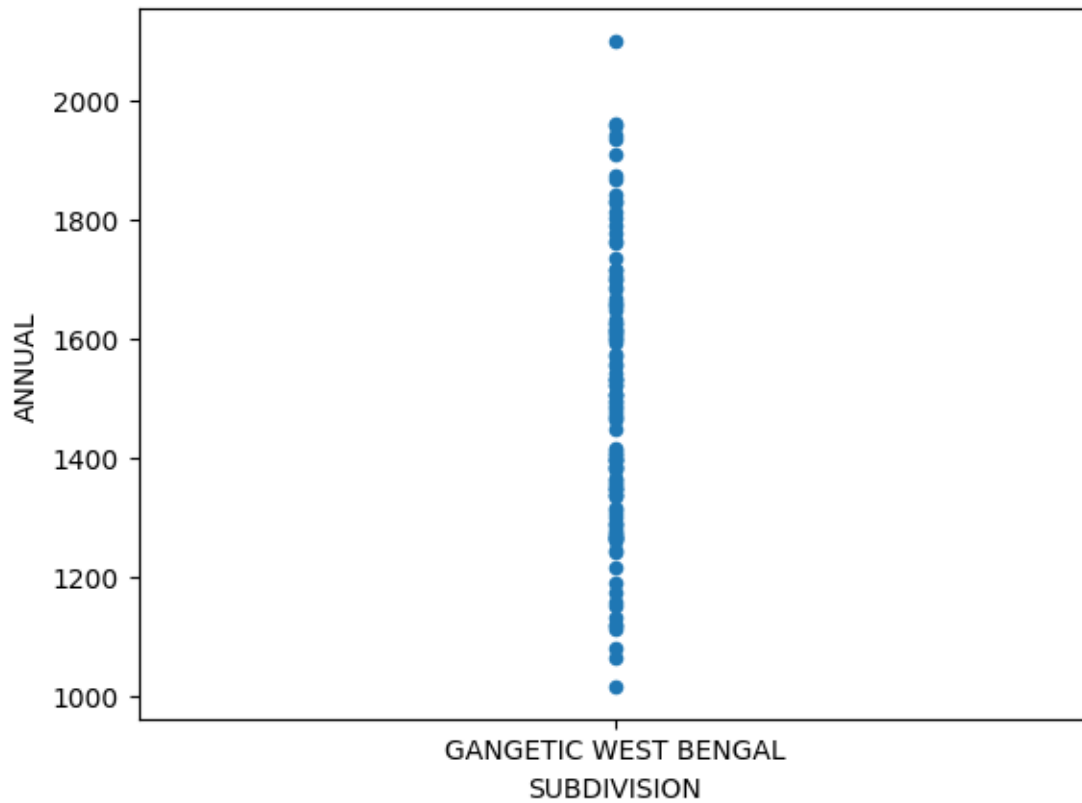
```
[12]: <Axes: ylabel='ANNUAL'>
```

11 Scatter chart

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

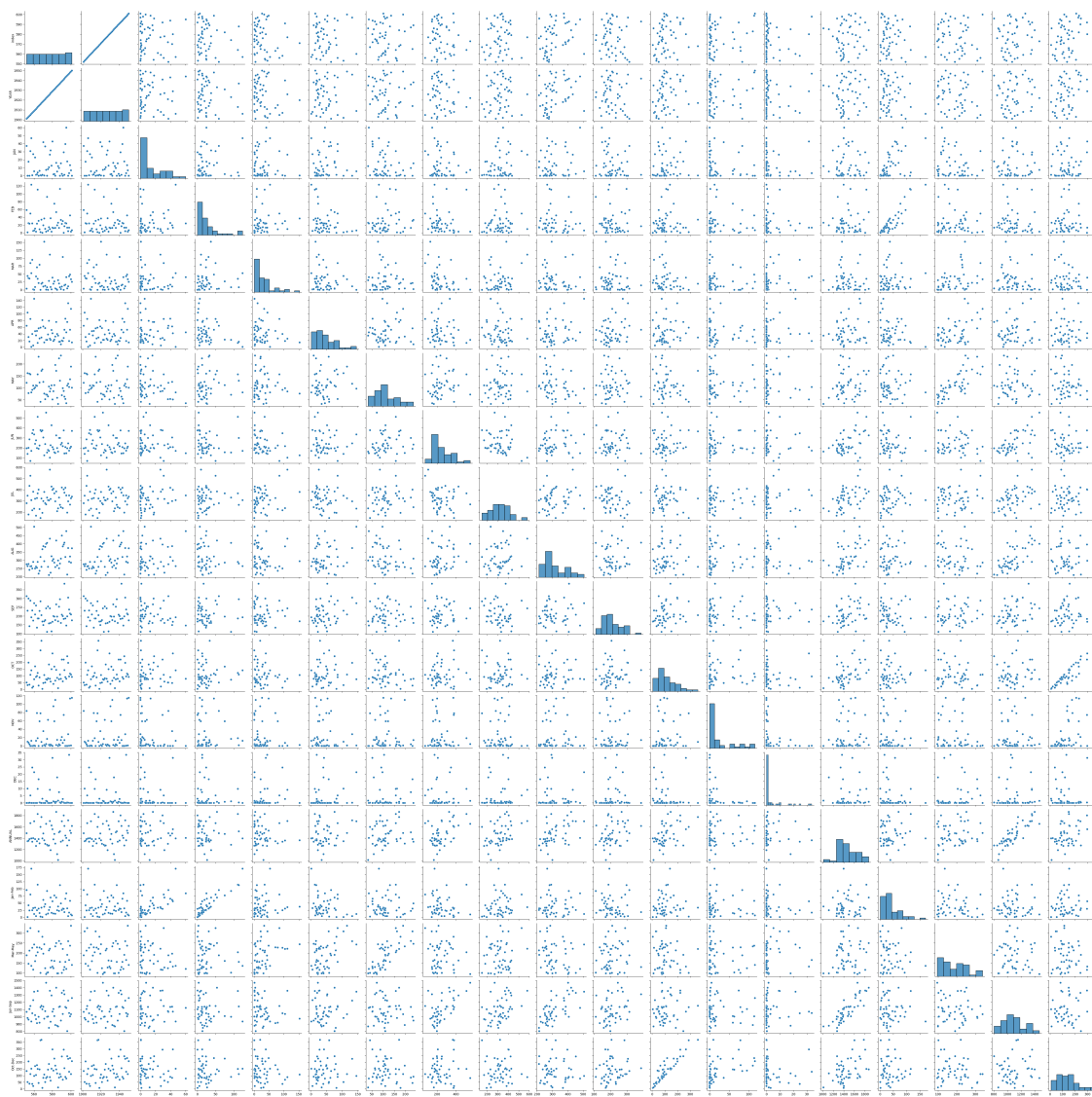
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7c6f11bef3a0>
```



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

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

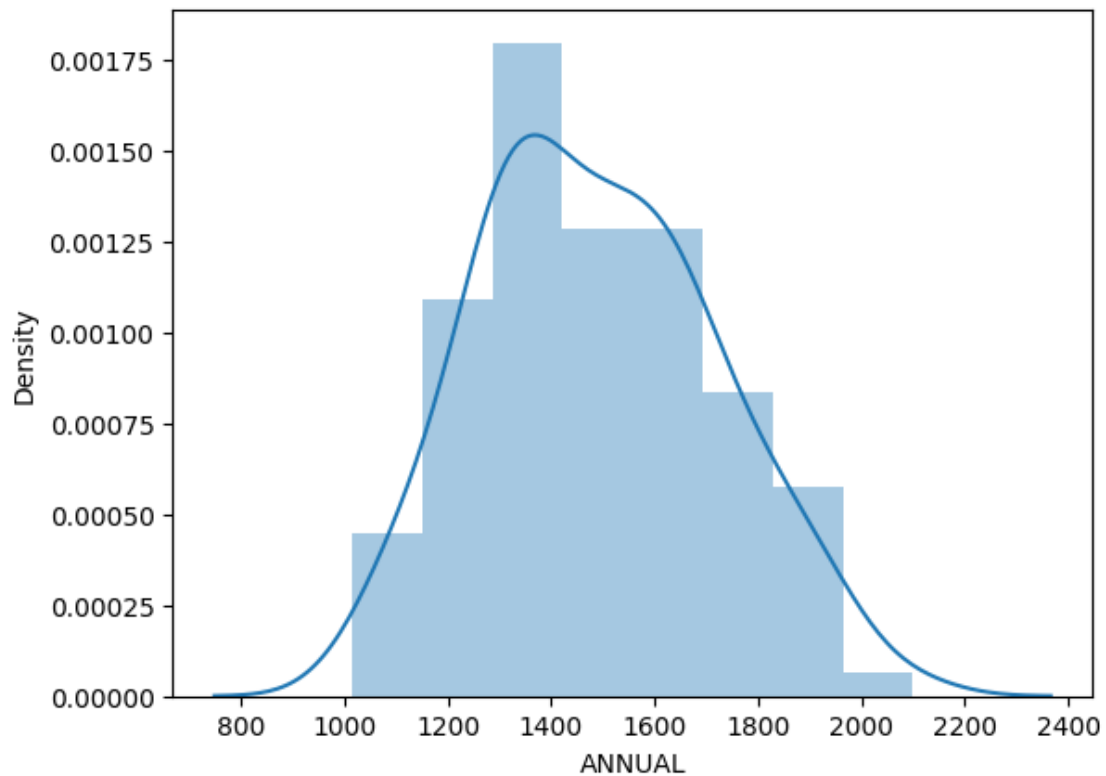
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

