

pnpf8tozy

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_west uttar_
↳pradesh.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	\	
0	1127	WEST UTTAR PRADESH	1901	51.4	25.6	9.5	0.7	5.6	23.8		
1	1128	WEST UTTAR PRADESH	1902	4.6	4.6	0.6	4.8	7.2	54.5		
2	1129	WEST UTTAR PRADESH	1903	13.4	0.4	1.2	0.0	8.2	32.7		
3	1130	WEST UTTAR PRADESH	1904	6.3	2.0	29.7	0.4	24.8	68.5		
4	1131	WEST UTTAR PRADESH	1905	32.3	26.6	14.8	3.6	7.1	18.9		
..		
110	1237	WEST UTTAR PRADESH	2011	2.1	10.4	3.9	2.8	29.6	175.9		
111	1238	WEST UTTAR PRADESH	2012	14.5	0.1	1.4	4.7	0.3	4.0		
112	1239	WEST UTTAR PRADESH	2013	20.4	69.5	3.5	1.6	2.1	190.6		
113	1240	WEST UTTAR PRADESH	2014	48.3	29.4	22.6	5.3	11.0	22.0		
114	1241	WEST UTTAR PRADESH	2015	31.6	7.2	66.8	21.0	8.1	72.0		
		JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	\
0	201.9	374.3	67.7	7.6	0.0	7.9	775.9	76.9	15.8		
1	325.9	180.6	143.1	9.6	0.9	0.2	736.6	9.2	12.7		
2	145.4	279.1	150.4	177.3	0.0	0.4	808.4	13.7	9.4		
3	358.8	311.1	97.1	2.7	15.7	28.2	945.2	8.2	54.9		
4	139.8	95.0	92.2	0.2	0.0	2.9	433.5	59.0	25.5		
..		

110	215.9	232.3	101.7	0.7	0.5	1.5	777.4	12.5	36.3
111	145.1	149.1	67.8	0.5	0.1	2.0	389.6	14.5	6.4
112	233.9	287.1	52.2	61.2	1.7	8.9	932.8	89.9	7.2
113	151.6	81.0	84.7	14.6	0.0	16.3	486.9	77.7	39.0
114	194.2	143.5	26.5	6.9	2.0	3.0	582.7	38.8	95.9

	Jun-Sep	Oct-Dec
0	667.7	15.4
1	704.1	10.6
2	607.6	177.6
3	835.5	46.6
4	345.9	3.1
..
110	725.9	2.7
111	366.0	2.7
112	763.8	71.8
113	339.4	30.8
114	436.1	11.9

[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
```

```

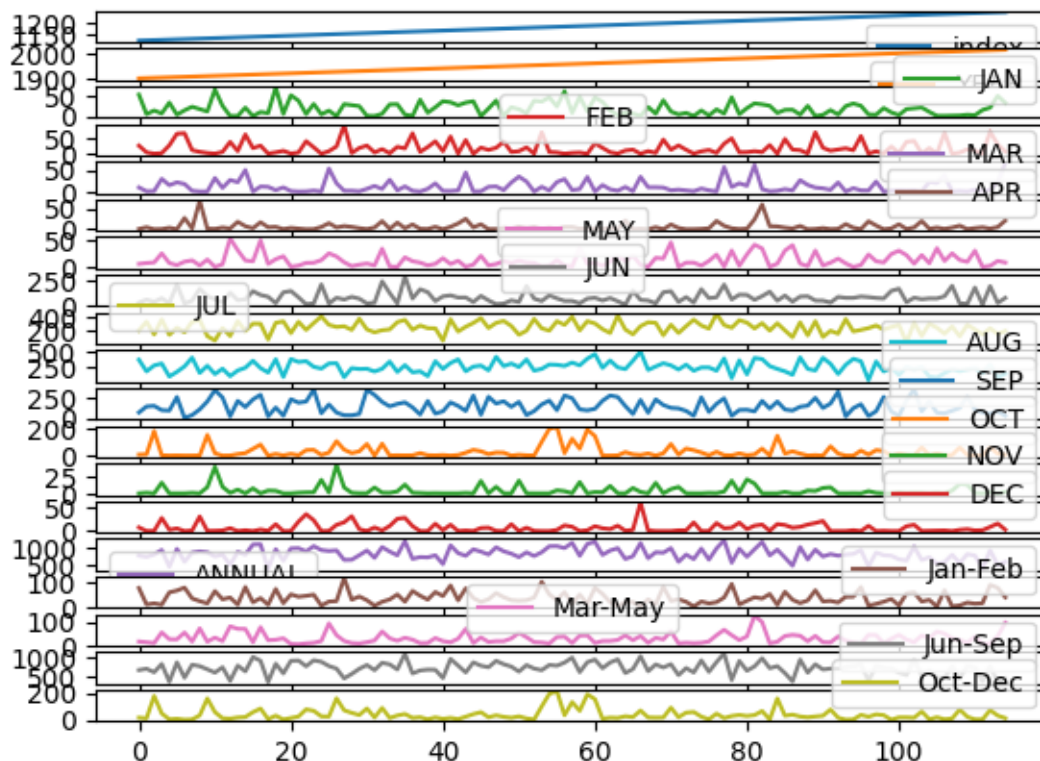
7  MAY          115 non-null    float64
8  JUN          115 non-null    float64
9  JUL          115 non-null    float64
10 AUG          115 non-null    float64
11 SEP          115 non-null    float64
12 OCT          115 non-null    float64
13 NOV          115 non-null    float64
14 DEC          115 non-null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

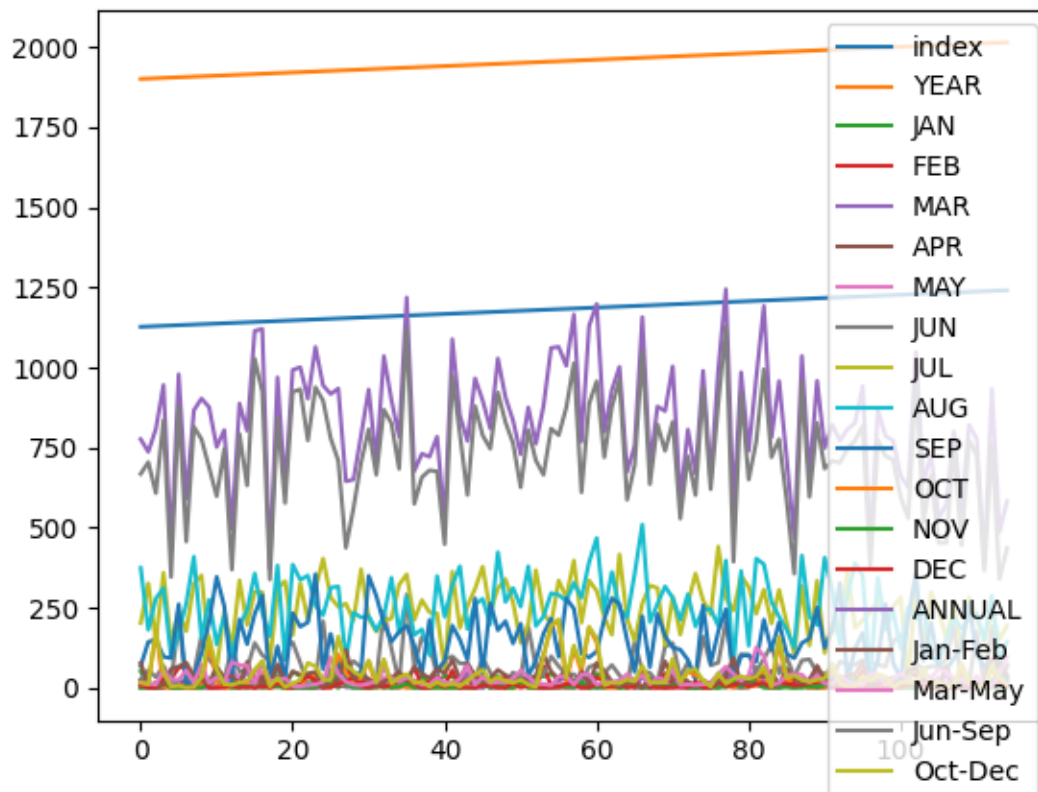
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



5 Line chart

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

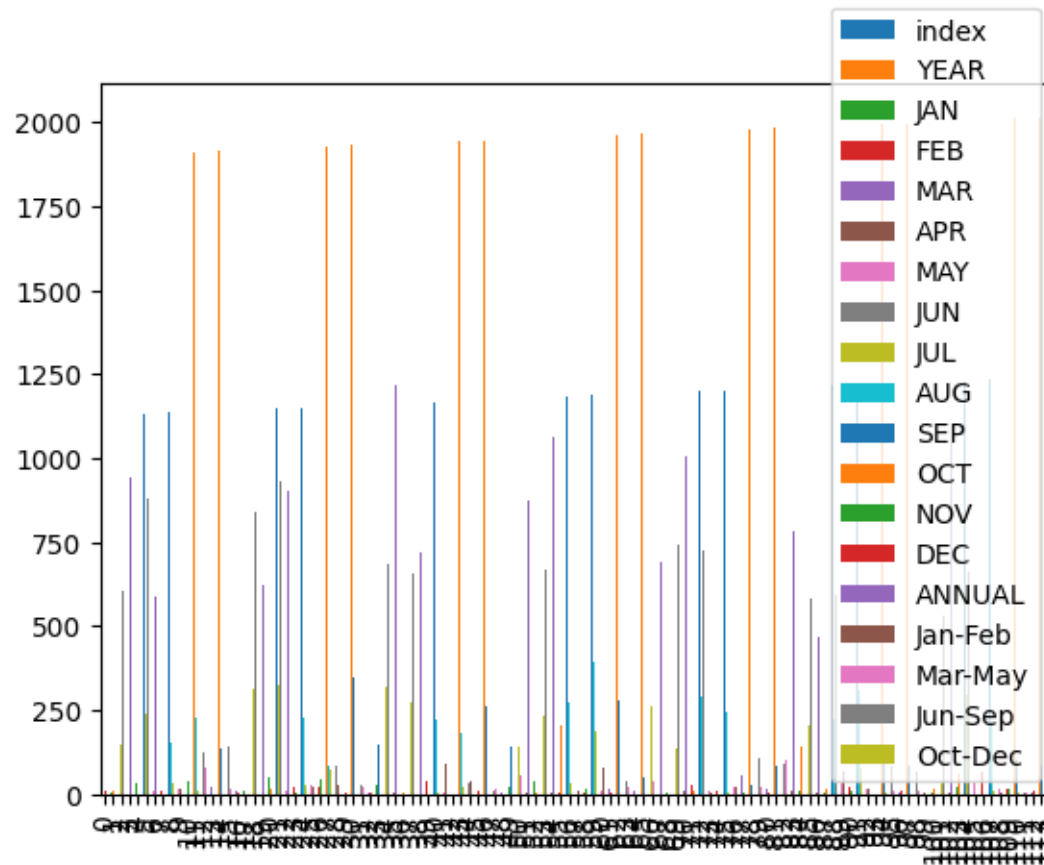
```
[7]: <Axes: >
```



6 Bar chart

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

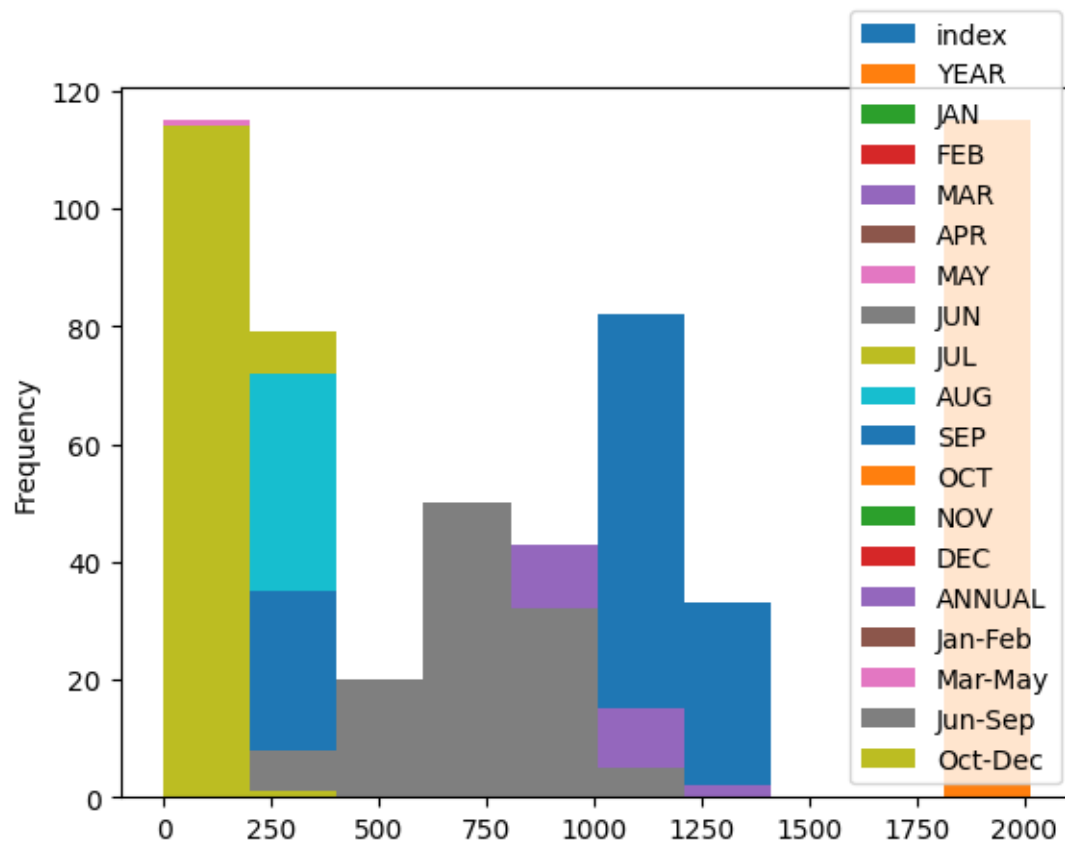
```
[8]: <Axes: >
```



7 Histogram

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

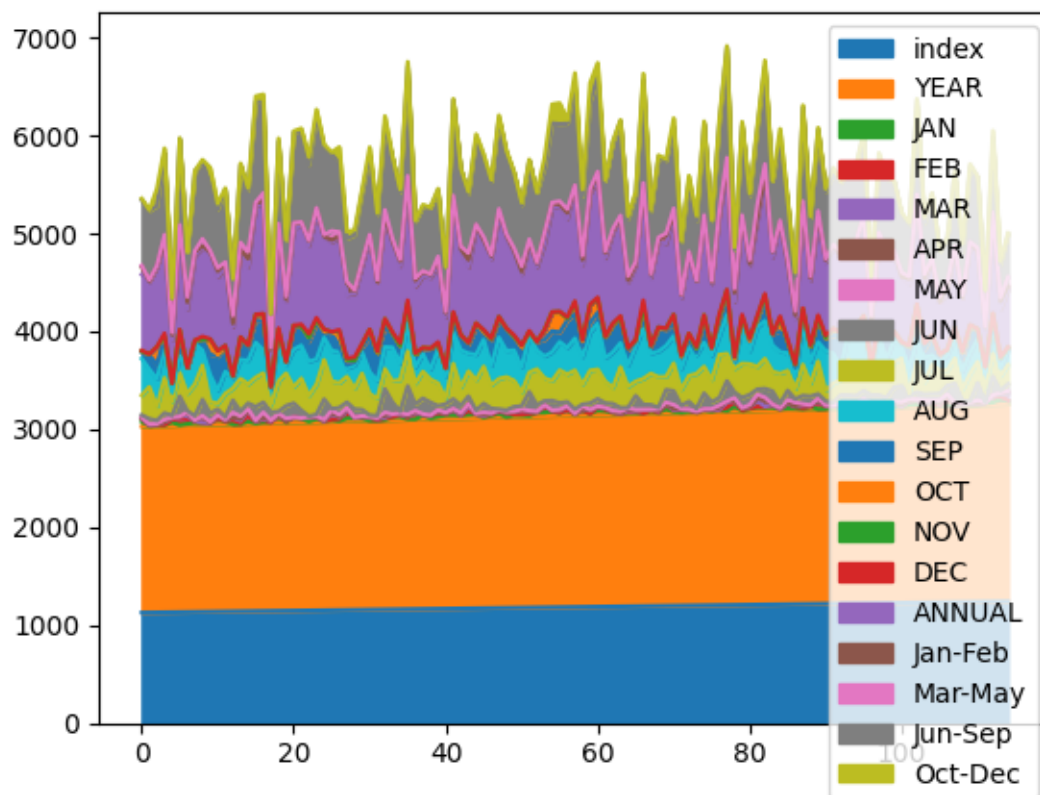
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

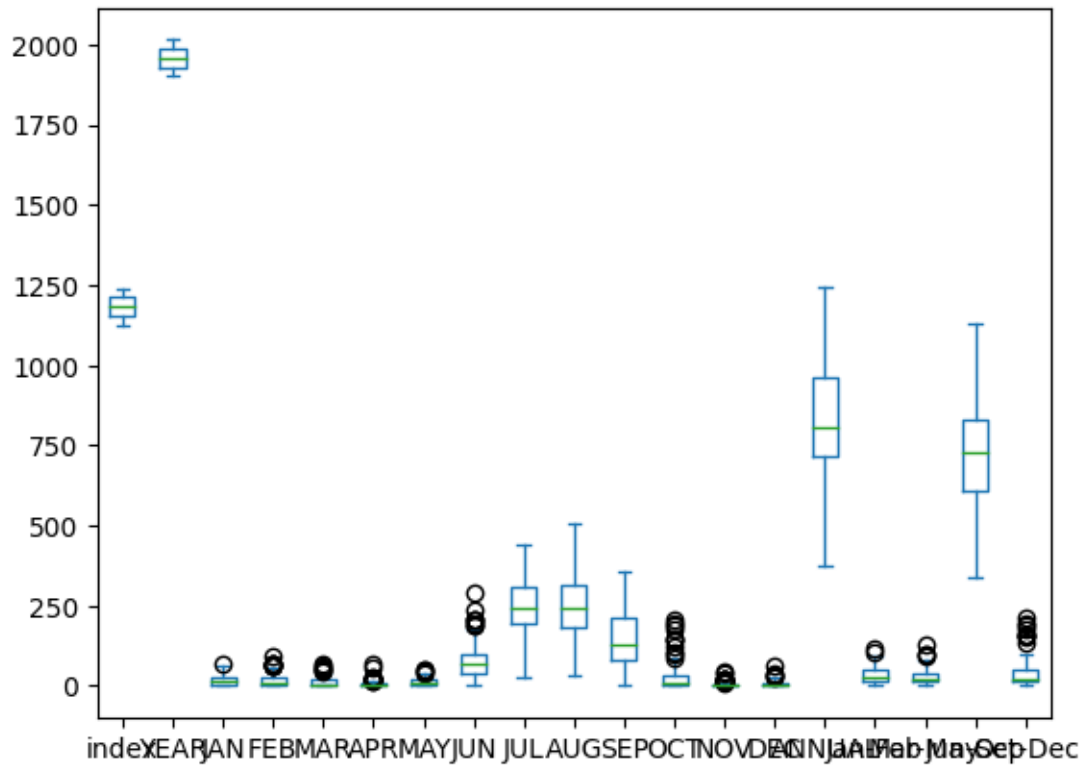
```
[10]: <Axes: >
```



9 Box chart

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

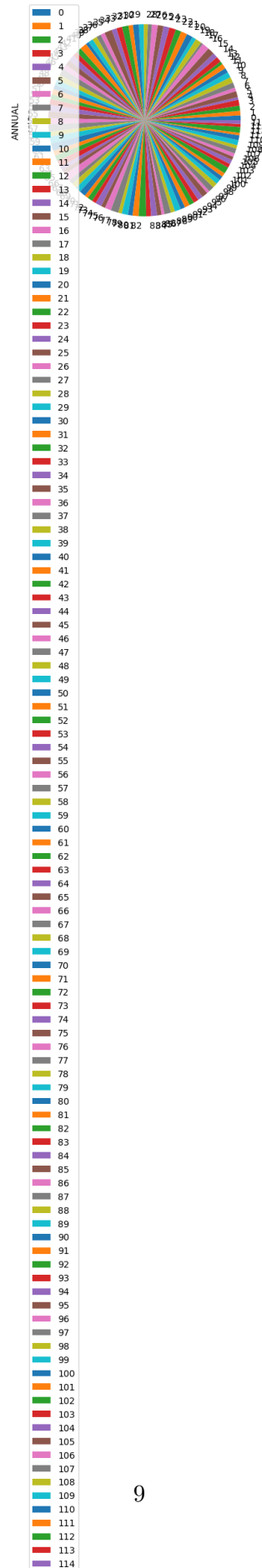
```
[11]: <Axes: >
```



10 Pie chart

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

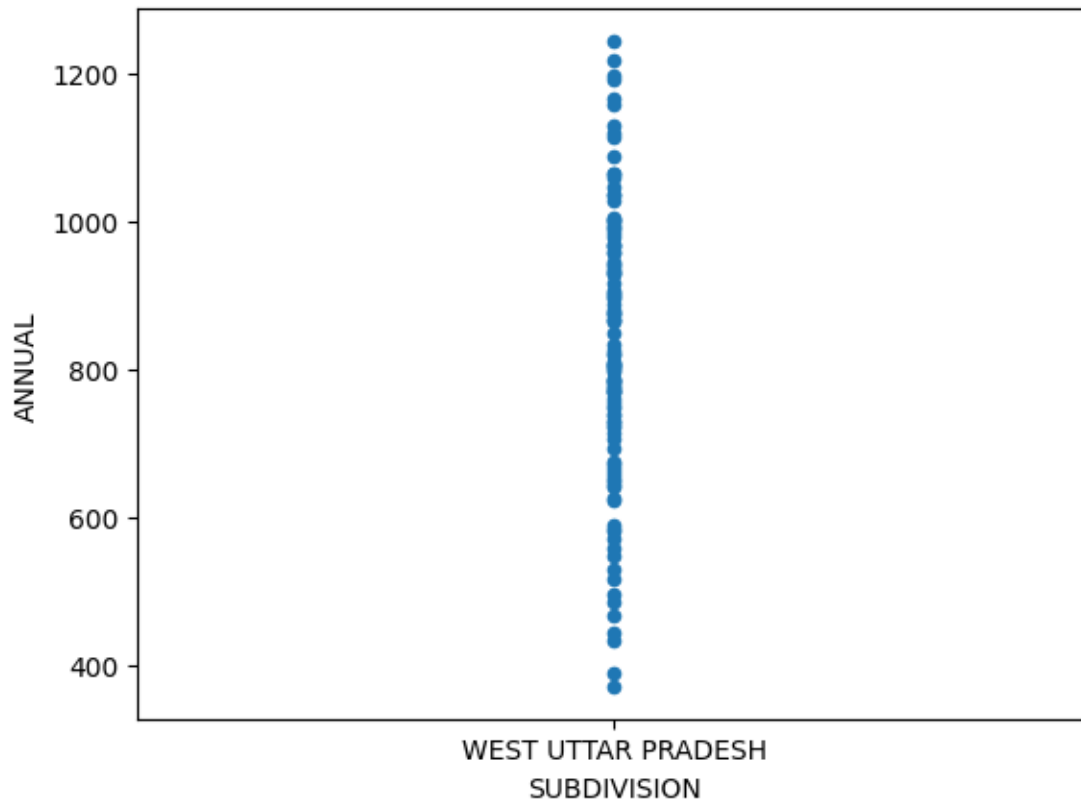
```
[12]: <Axes: ylabel='ANNUAL'>
```

11 Scatter chart

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

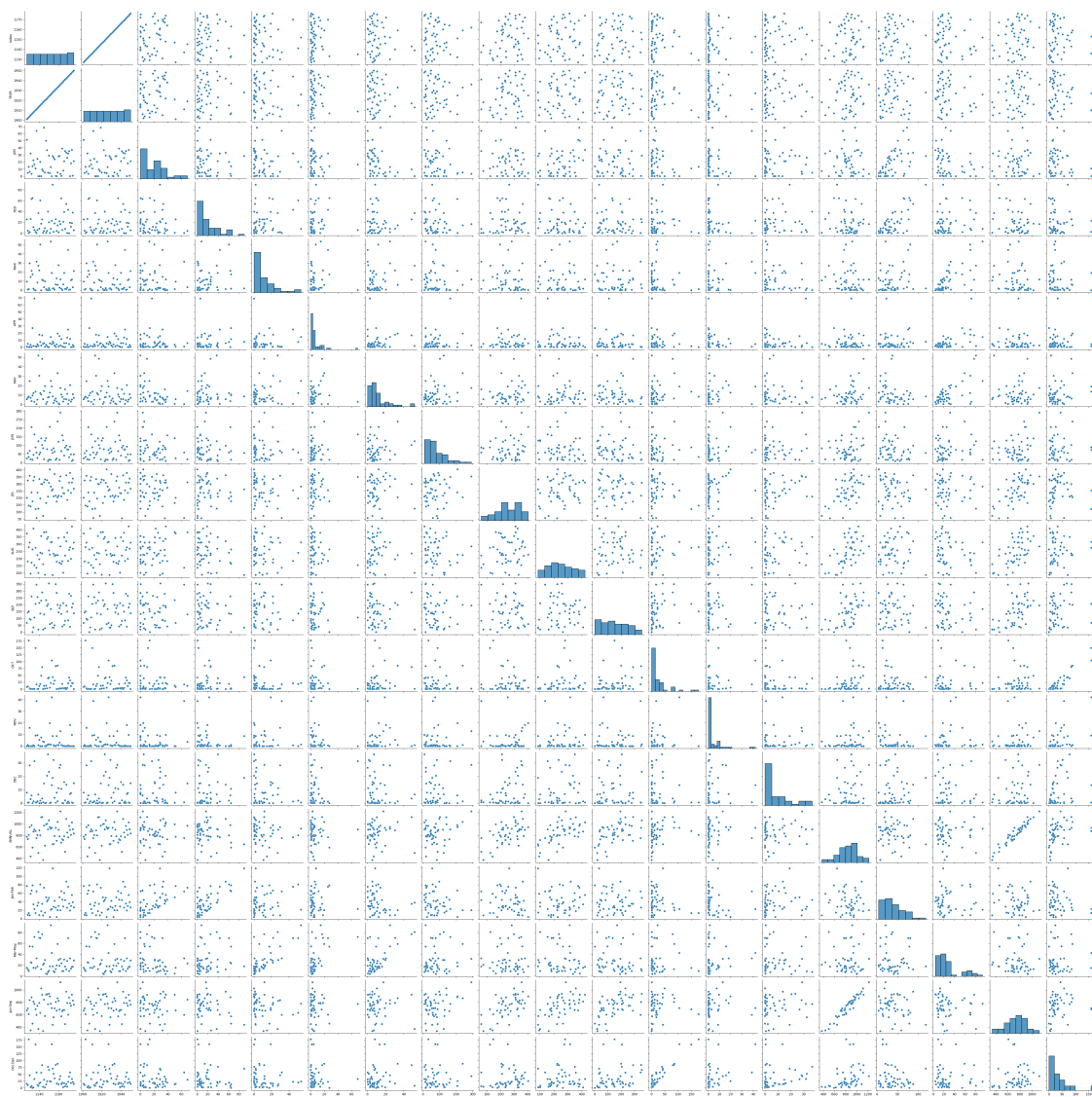
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7bec3cc32fe0>
```



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

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

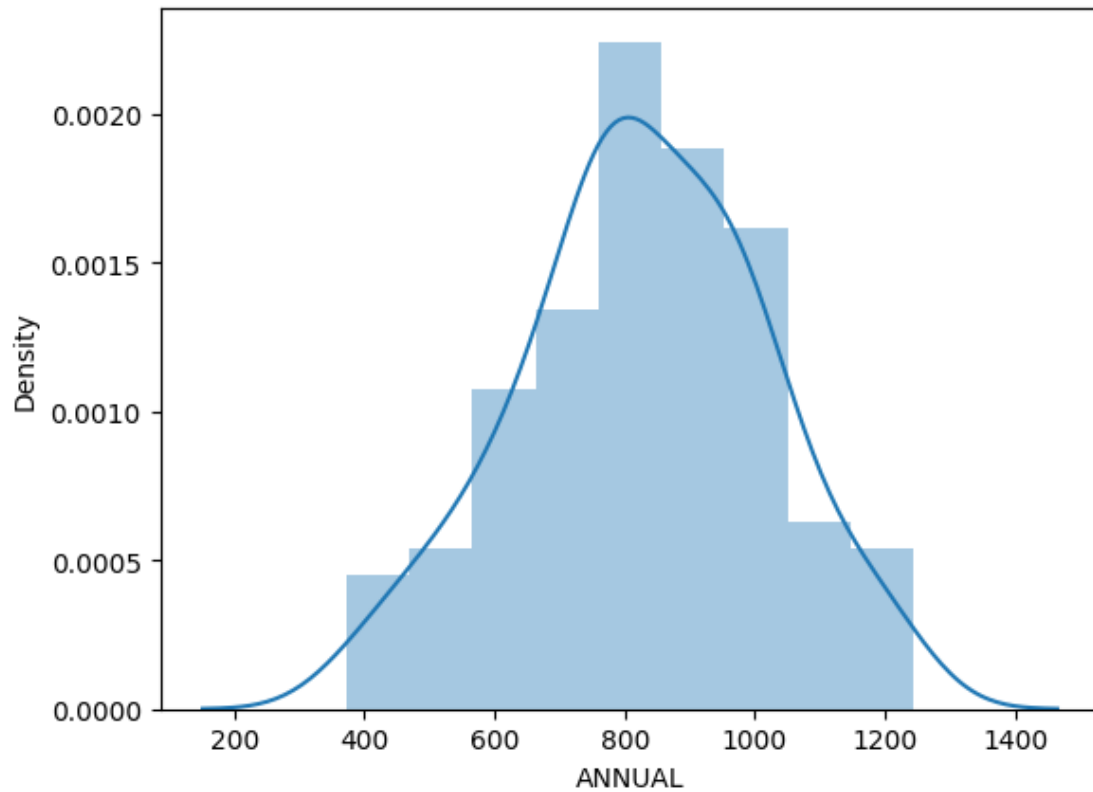
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

