

oktmvtzzp

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_rajasthan.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	\
0	1817	WEST RAJASTHAN	1901	6.7	0.0	1.1	0.0	6.1	3.0	79.0	
1	1818	WEST RAJASTHAN	1902	0.0	0.0	0.0	0.5	4.0	49.1	27.0	
2	1819	WEST RAJASTHAN	1903	1.7	1.3	5.5	0.0	4.2	2.7	154.8	
3	1820	WEST RAJASTHAN	1904	3.8	2.9	16.3	0.7	11.4	14.6	39.8	
4	1821	WEST RAJASTHAN	1905	6.3	4.8	0.7	1.3	0.3	4.9	30.1	
..	
110	1927	WEST RAJASTHAN	2011	0.0	11.8	1.5	1.5	7.8	24.4	88.5	
111	1928	WEST RAJASTHAN	2012	0.5	0.0	0.0	9.5	10.4	5.3	40.4	
112	1929	WEST RAJASTHAN	2013	8.6	21.8	4.2	3.1	1.7	37.6	104.5	
113	1930	WEST RAJASTHAN	2014	0.8	2.2	4.7	8.4	23.0	13.8	94.3	
114	1931	WEST RAJASTHAN	2015	1.4	0.9	30.3	25.2	15.5	53.2	234.6	
		AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	59.2	1.0	2.1	0.0	0.6	158.9	6.7	7.2	142.2	2.7	
1	71.3	41.8	1.8	0.0	0.0	195.6	0.0	4.5	189.2	1.8	
2	87.1	49.3	0.1	0.0	0.5	307.0	3.0	9.7	293.8	0.5	
3	45.6	21.4	1.4	2.9	7.1	167.9	6.6	28.5	121.4	11.4	
4	0.6	64.5	0.0	0.0	0.9	114.4	11.0	2.4	100.1	0.9	
..	

110	166.8	116.3	0.1	0.0	0.0	418.7	11.8	10.7	396.1	0.1
111	166.7	92.0	1.9	0.0	0.6	327.3	0.5	19.9	304.3	2.5
112	138.2	58.7	10.1	1.0	0.0	389.4	30.4	8.9	339.0	11.1
113	69.6	84.9	0.5	0.2	0.0	302.4	3.0	36.0	262.6	0.7
114	60.5	35.7	1.1	0.1	0.0	458.4	2.3	71.0	384.0	1.2

[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()
```

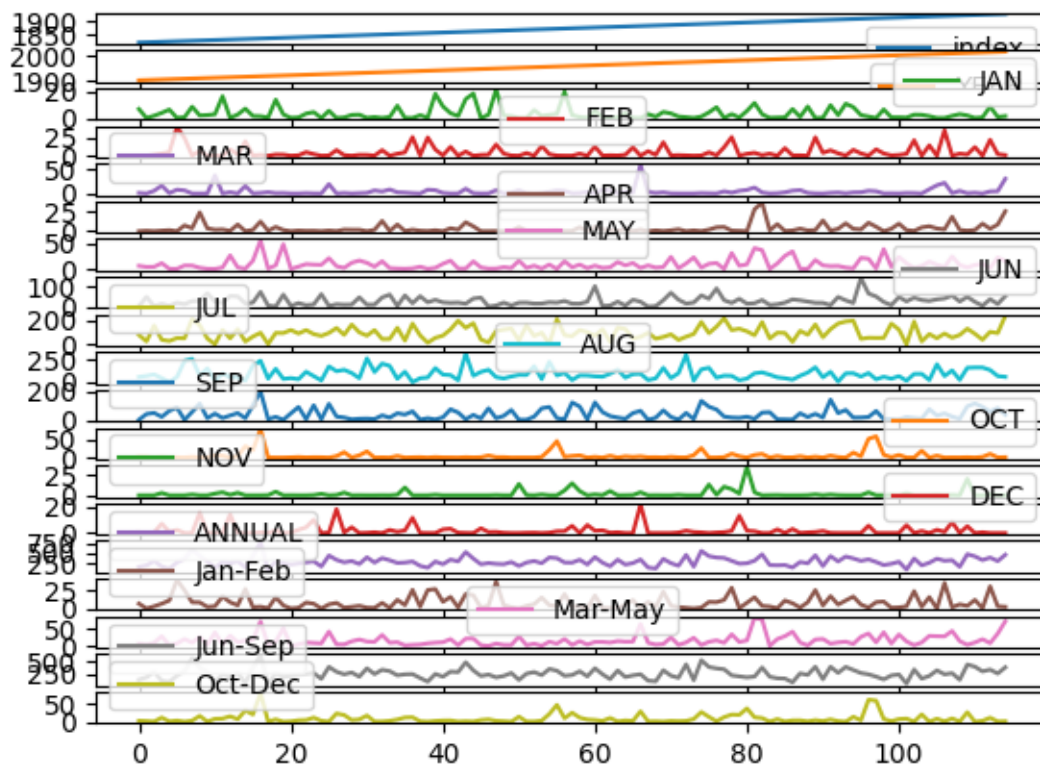
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 20 columns):
#   Column          Non-Null Count  Dtype
---  -
0   index           115 non-null   int64
1   SUBDIVISION     115 non-null   object
2   YEAR            115 non-null   int64
3   JAN             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)
```

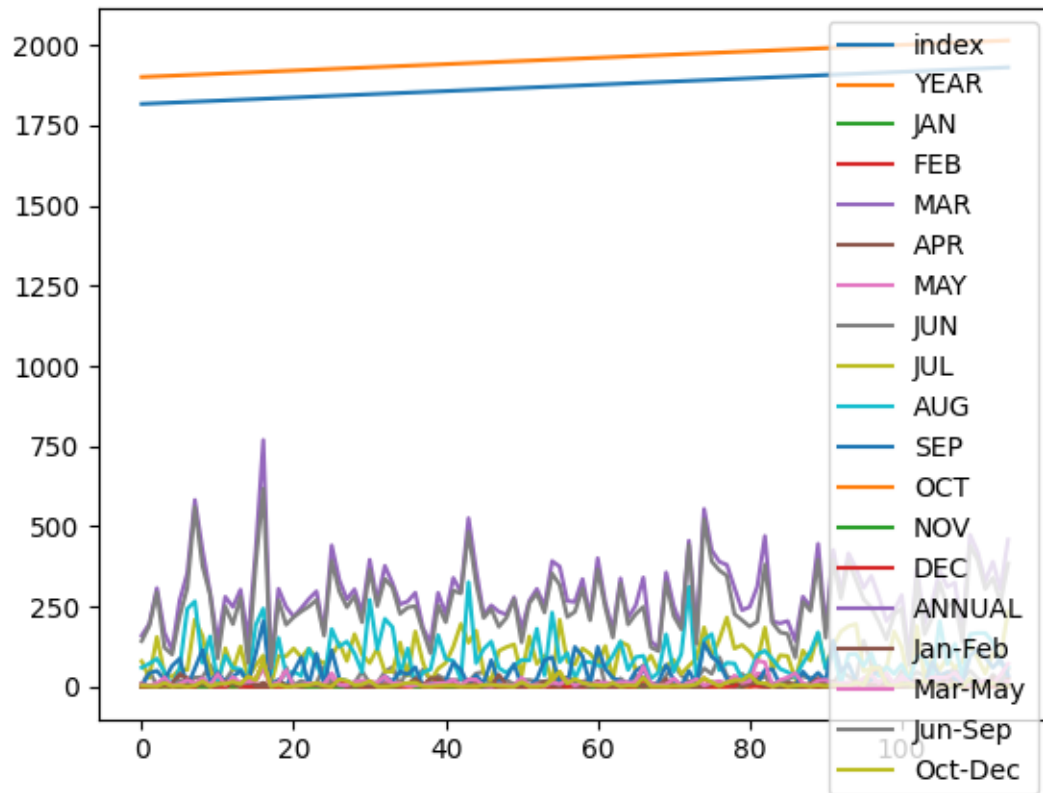
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



5 Line chart

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

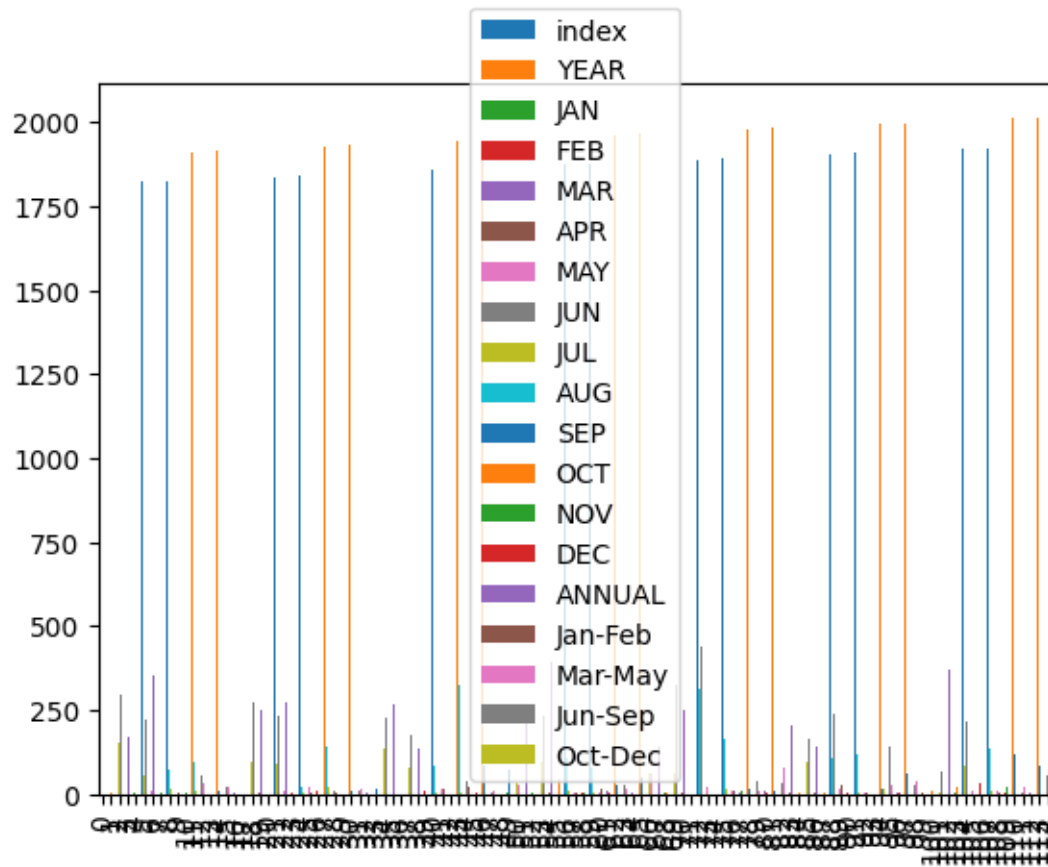
```
[7]: <Axes: >
```



6 Bar chart

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

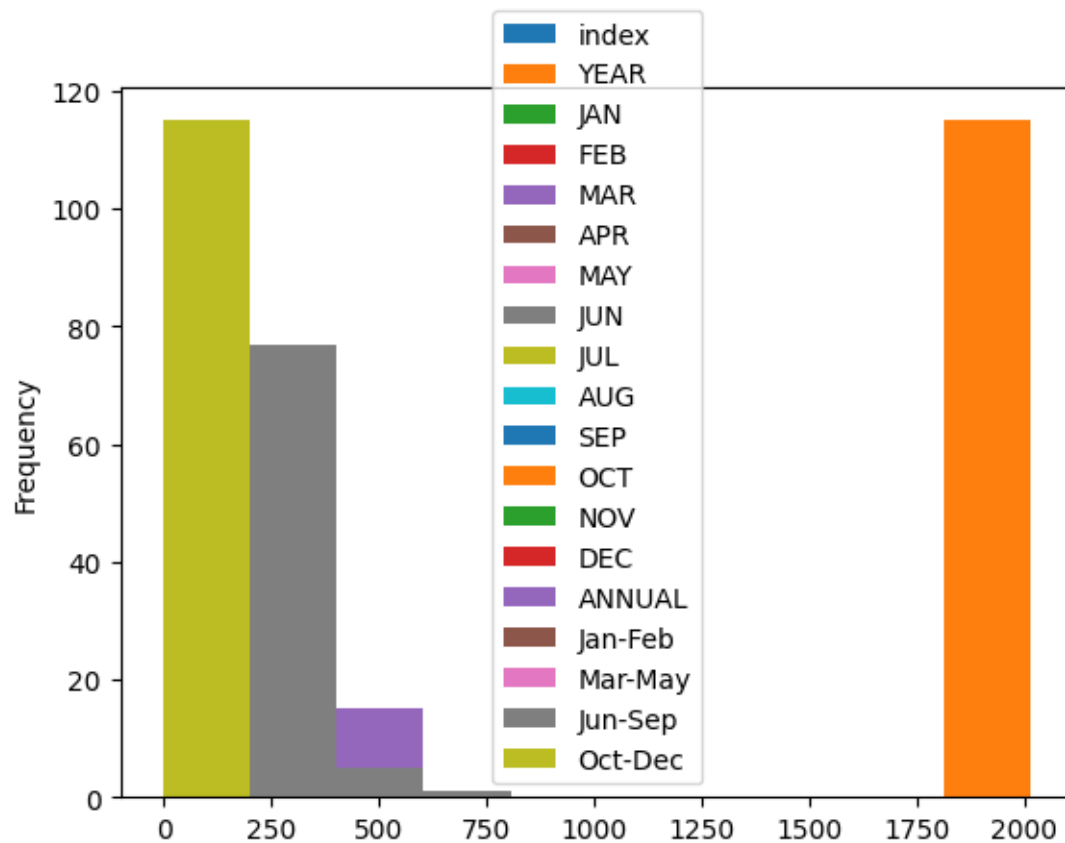
```
[8]: <Axes: >
```



7 Histogram

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

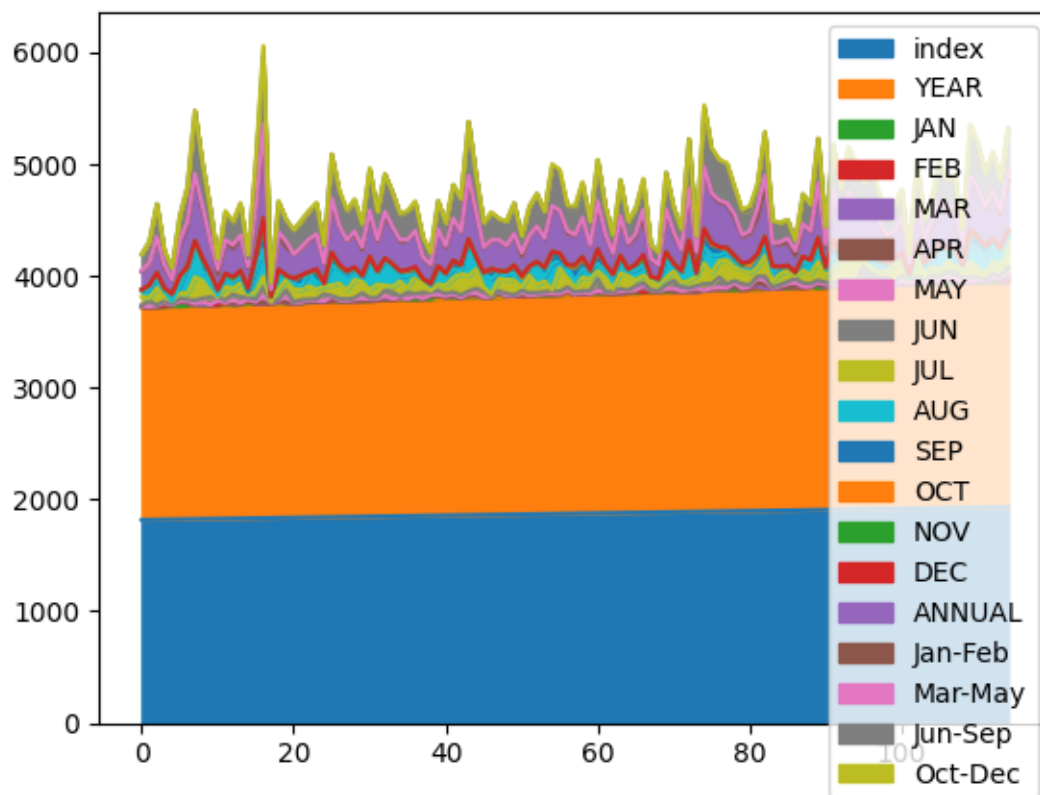
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

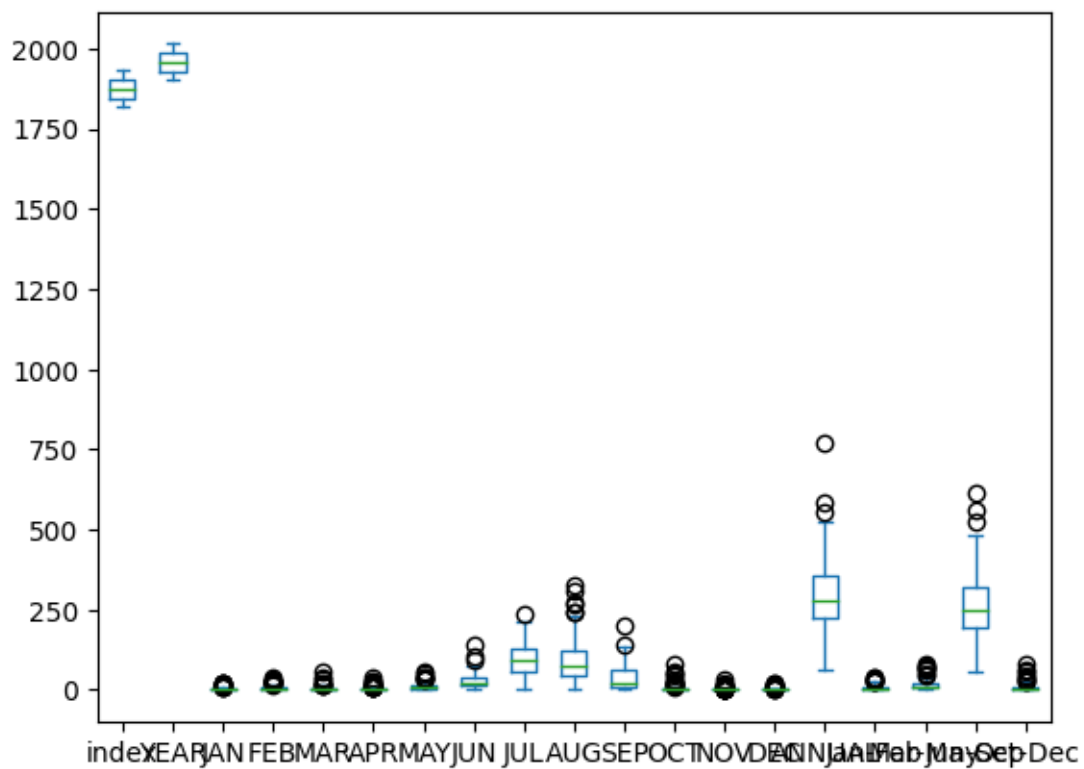
```
[10]: <Axes: >
```



9 Box chart

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

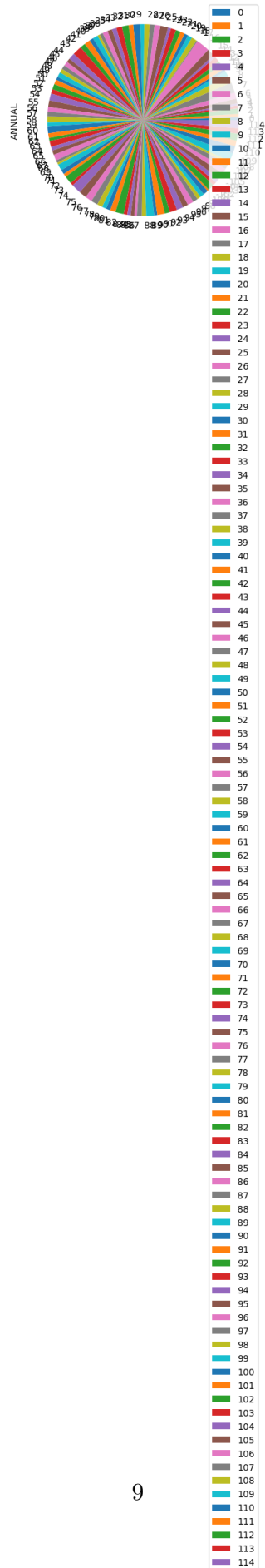
```
[11]: <Axes: >
```



10 Pie chart

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

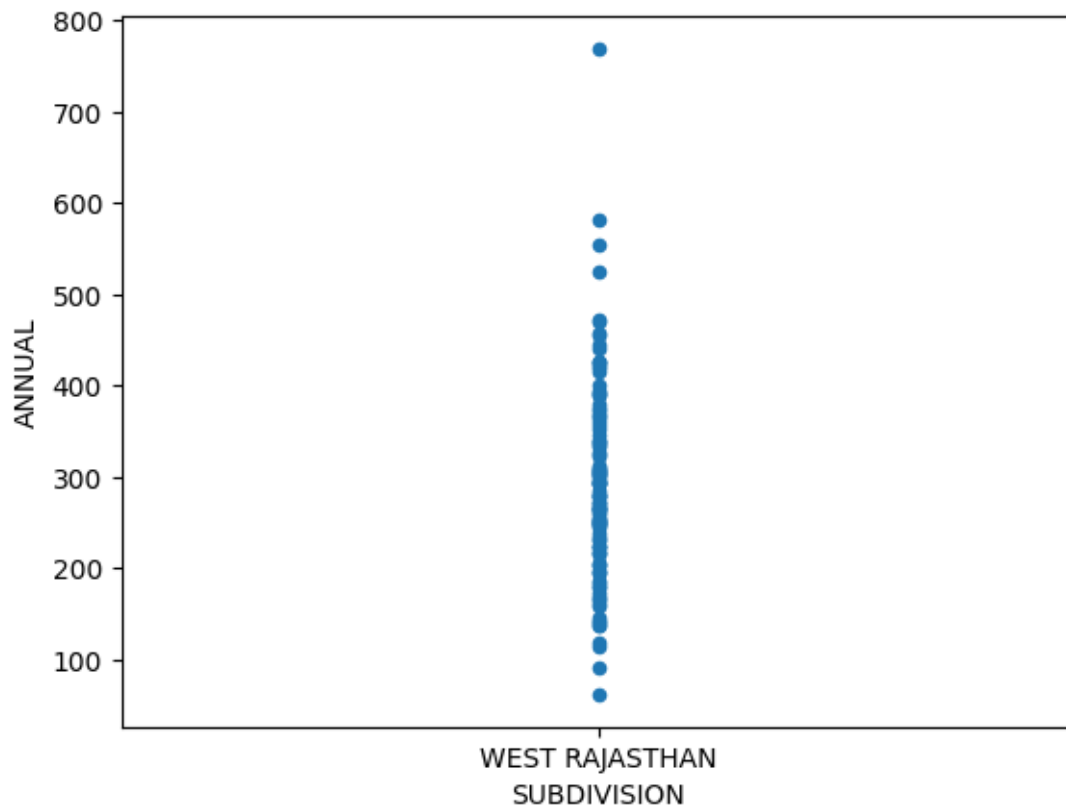
```
[12]: <Axes: ylabel='ANNUAL'>
```

11 Scatter chart

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

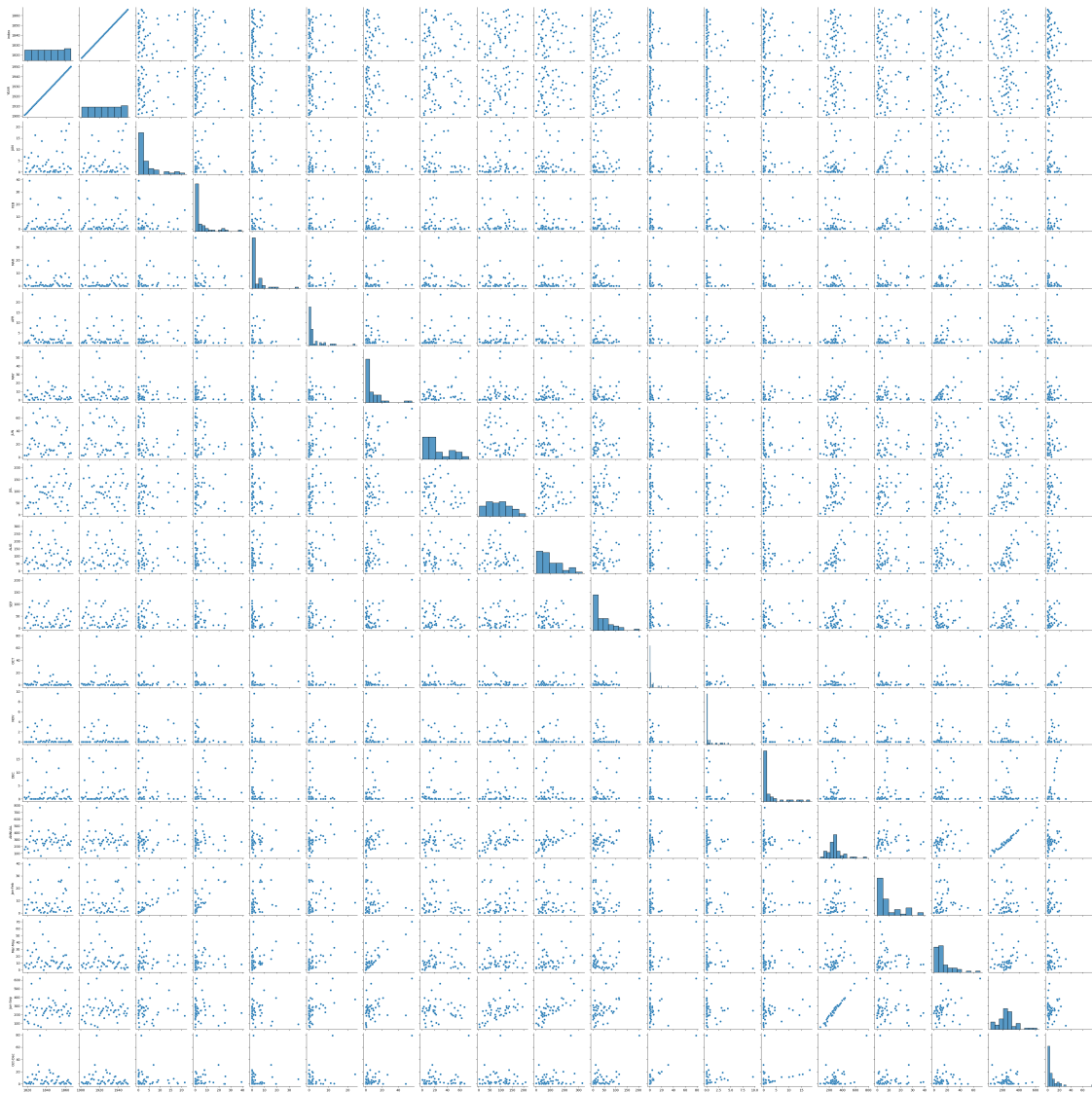
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7e3e035bb610>
```



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

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

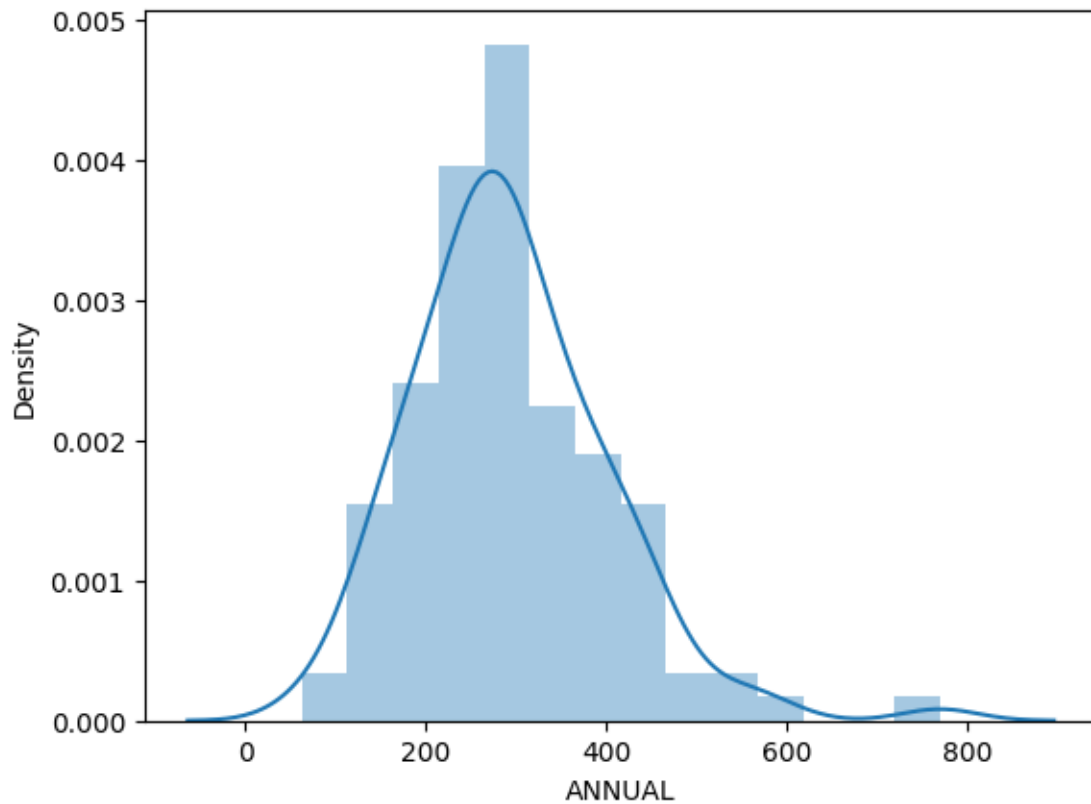
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

