

b02cllcmo

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_naga_mani_
↳mizo_tripura.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index		SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	\		
0	322	NAGA MANI MIZO TRIPURA	1901	11.7	18.1	29.4	206.2	124.0				
1	323	NAGA MANI MIZO TRIPURA	1902	4.8	0.5	36.3	297.8	215.5				
2	324	NAGA MANI MIZO TRIPURA	1903	6.5	40.5	139.8	45.5	159.9				
3	325	NAGA MANI MIZO TRIPURA	1904	2.3	46.9	47.5	290.3	230.5				
4	326	NAGA MANI MIZO TRIPURA	1905	9.1	35.3	306.5	161.7	193.6				
..				
110	432	NAGA MANI MIZO TRIPURA	2011	12.6	3.6	51.4	81.1	334.9				
111	433	NAGA MANI MIZO TRIPURA	2012	24.5	10.2	20.3	243.5	163.5				
112	434	NAGA MANI MIZO TRIPURA	2013	0.2	5.7	19.7	60.3	348.9				
113	435	NAGA MANI MIZO TRIPURA	2014	1.2	21.0	25.4	49.6	192.5				
114	436	NAGA MANI MIZO TRIPURA	2015	14.4	14.2	21.6	253.5	198.3				
		JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	\
0	443.3	331.4	466.0	304.1	166.7	67.4	0.0	2168.4	29.8	359.6		
1	480.1	392.4	312.8	318.7	102.4	8.9	4.7	2175.0	5.2	549.7		
2	458.6	300.2	470.6	366.1	166.4	76.7	0.1	2230.7	47.0	345.1		
3	455.3	423.5	423.6	375.8	128.9	90.0	5.0	2519.8	49.3	568.3		
4	339.7	450.1	429.9	320.1	246.4	8.0	27.1	2527.6	44.5	661.8		
..		

110	374.2	313.3	367.6	258.3	92.6	2.4	0.2	1892.1	16.1	467.4
111	396.2	280.1	342.7	248.7	160.9	32.0	0.4	1922.9	34.7	427.3
112	206.6	255.9	291.3	241.4	125.6	0.3	1.2	1557.2	5.9	428.9
113	268.3	295.7	372.3	300.9	69.6	3.3	0.1	1599.9	22.2	267.5
114	283.9	413.6	334.2	255.9	118.7	3.9	10.0	1922.4	28.7	473.4

	Jun-Sep	Oct-Dec
0	1544.8	234.2
1	1504.0	116.0
2	1595.4	243.2
3	1678.2	224.0
4	1539.8	281.5
..
110	1313.4	95.2
111	1267.6	193.3
112	995.2	127.1
113	1237.2	73.0
114	1287.7	132.6

[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
```

```

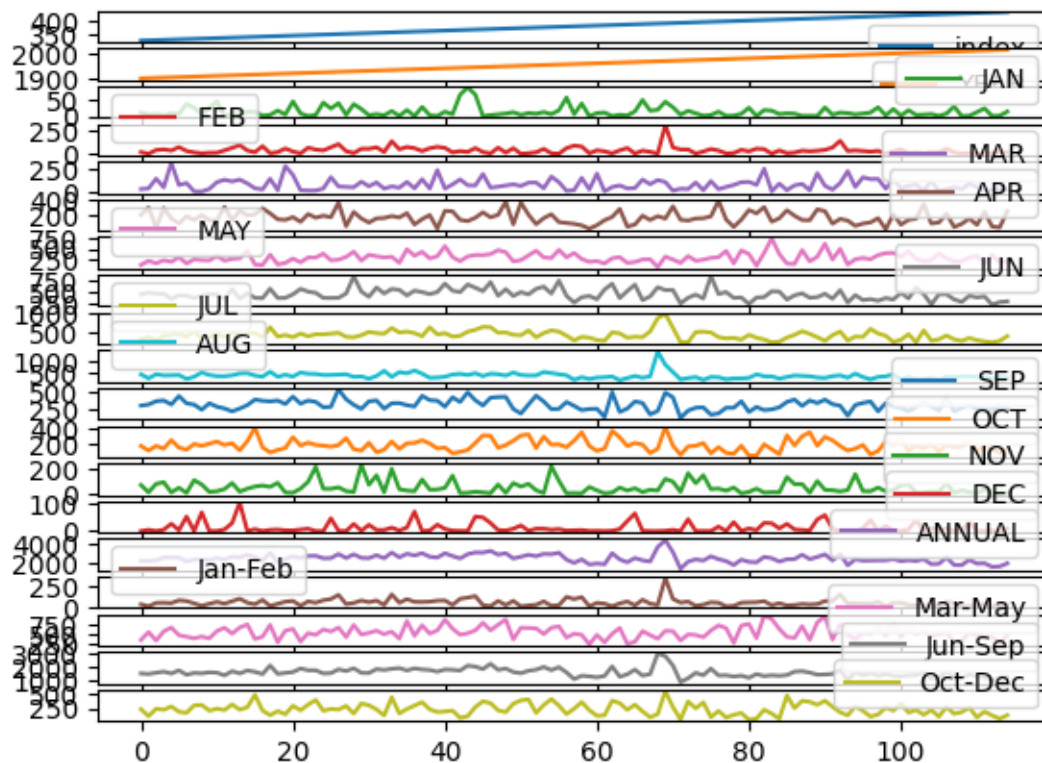
7  MAY          115 non-null    float64
8  JUN          115 non-null    float64
9  JUL          115 non-null    float64
10 AUG          115 non-null    float64
11 SEP          115 non-null    float64
12 OCT          115 non-null    float64
13 NOV          115 non-null    float64
14 DEC          115 non-null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

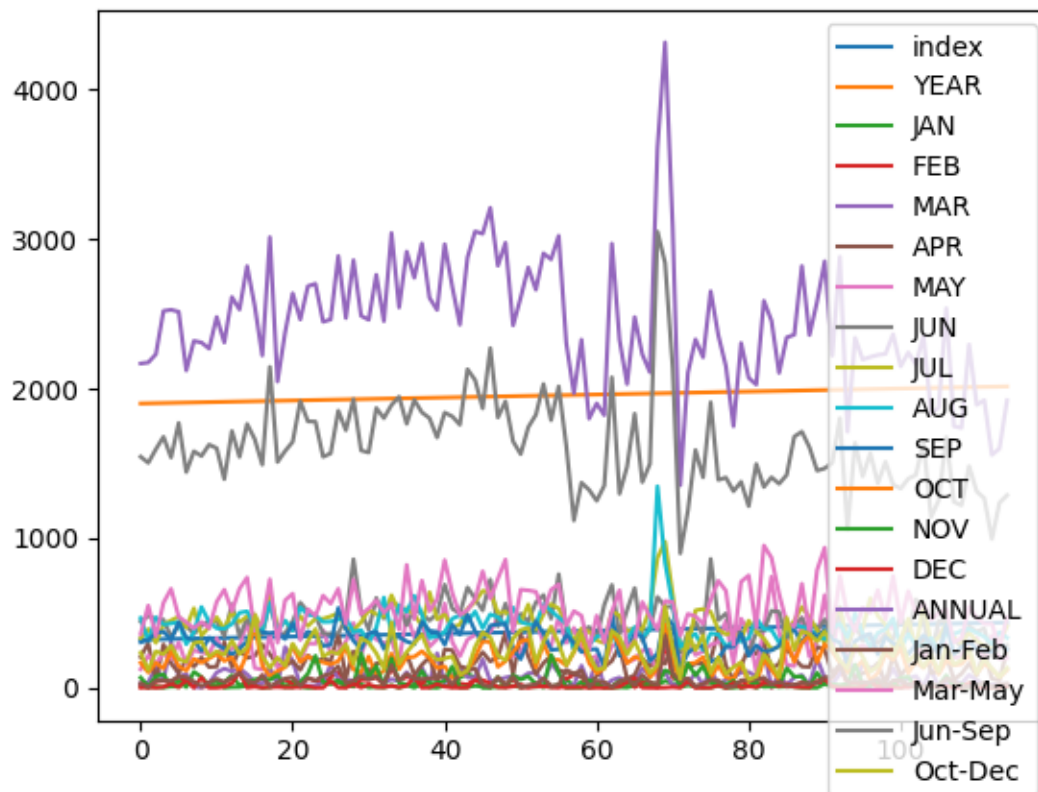
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



5 Line chart

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

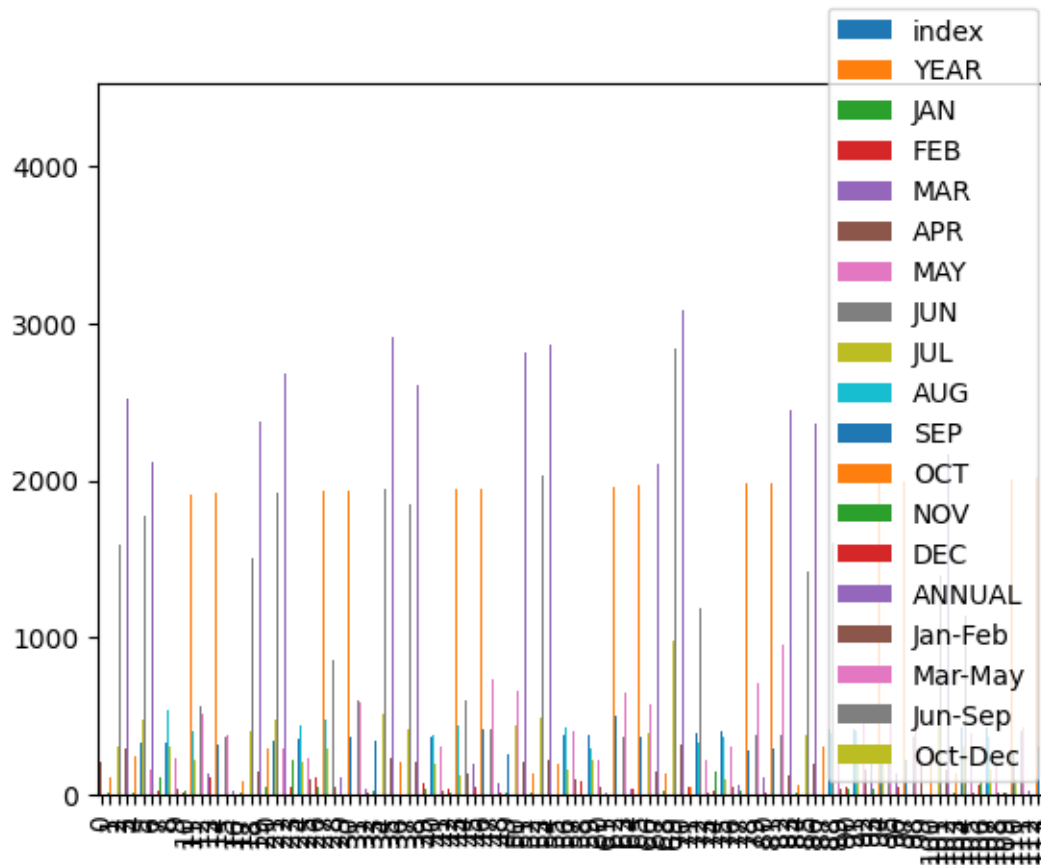
```
[7]: <Axes: >
```



6 Bar chart

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

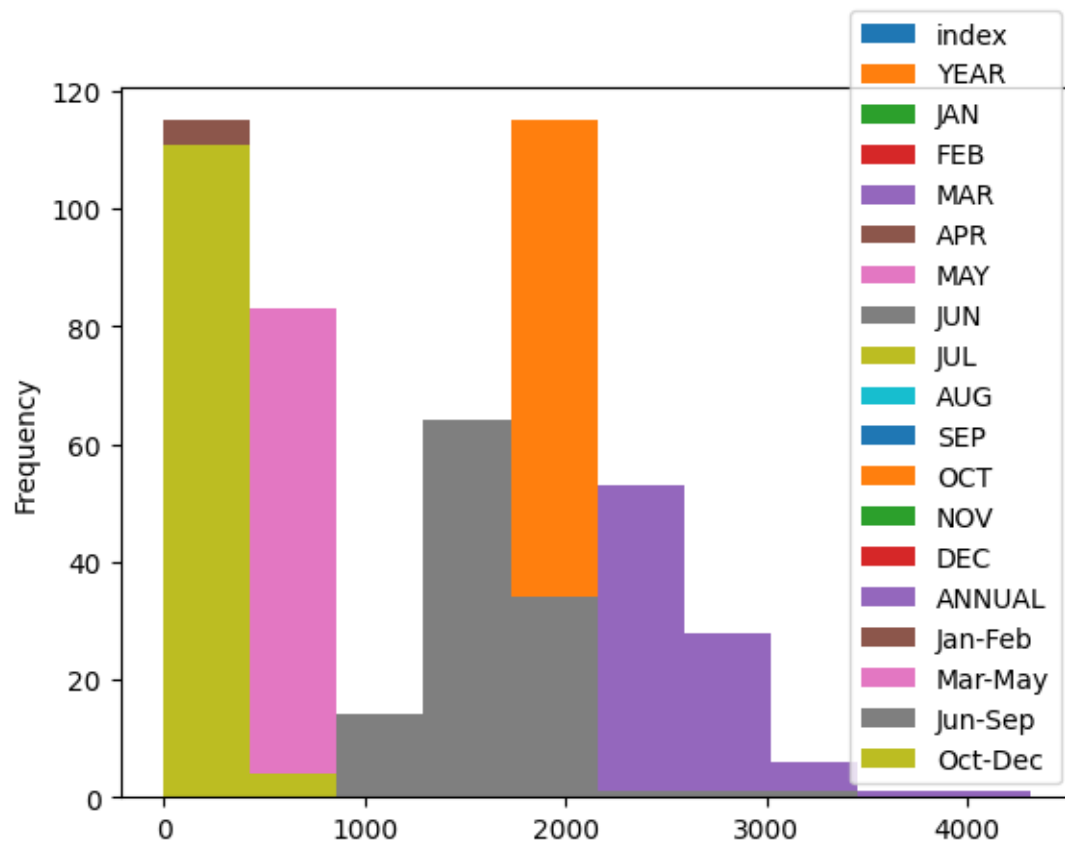
```
[8]: <Axes: >
```



7 Histogram

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

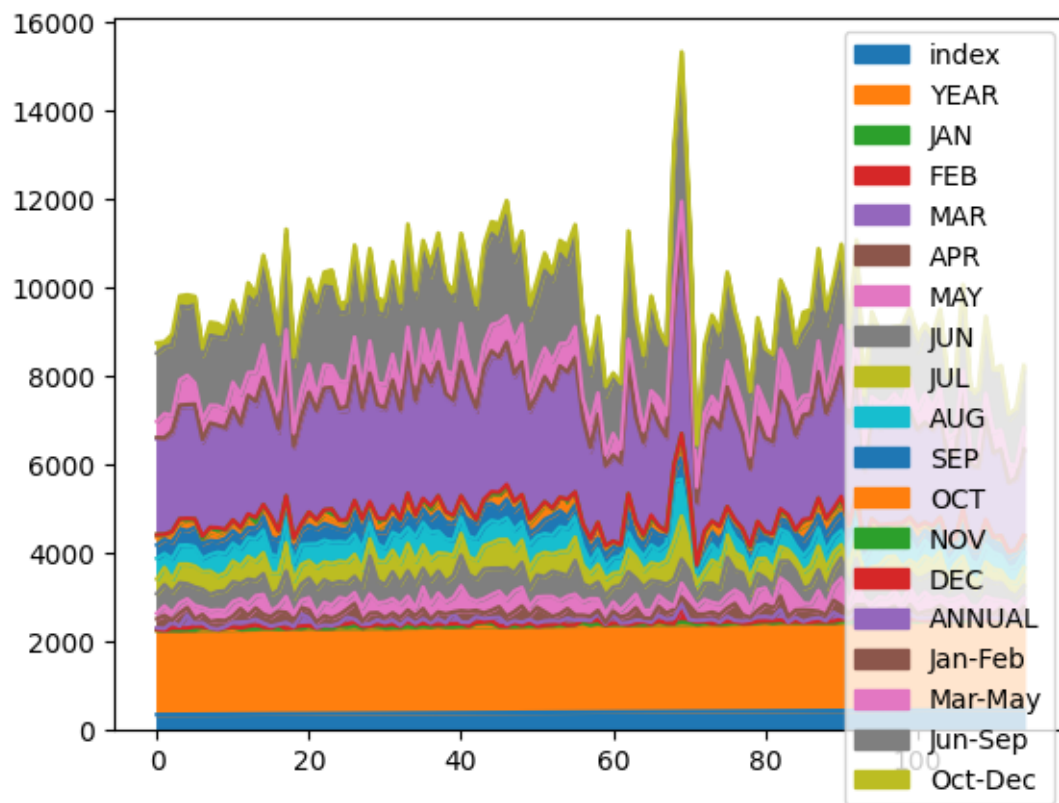
```
[9]: <Axes: ylabel='Frequency'>
```



8 Area chart

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

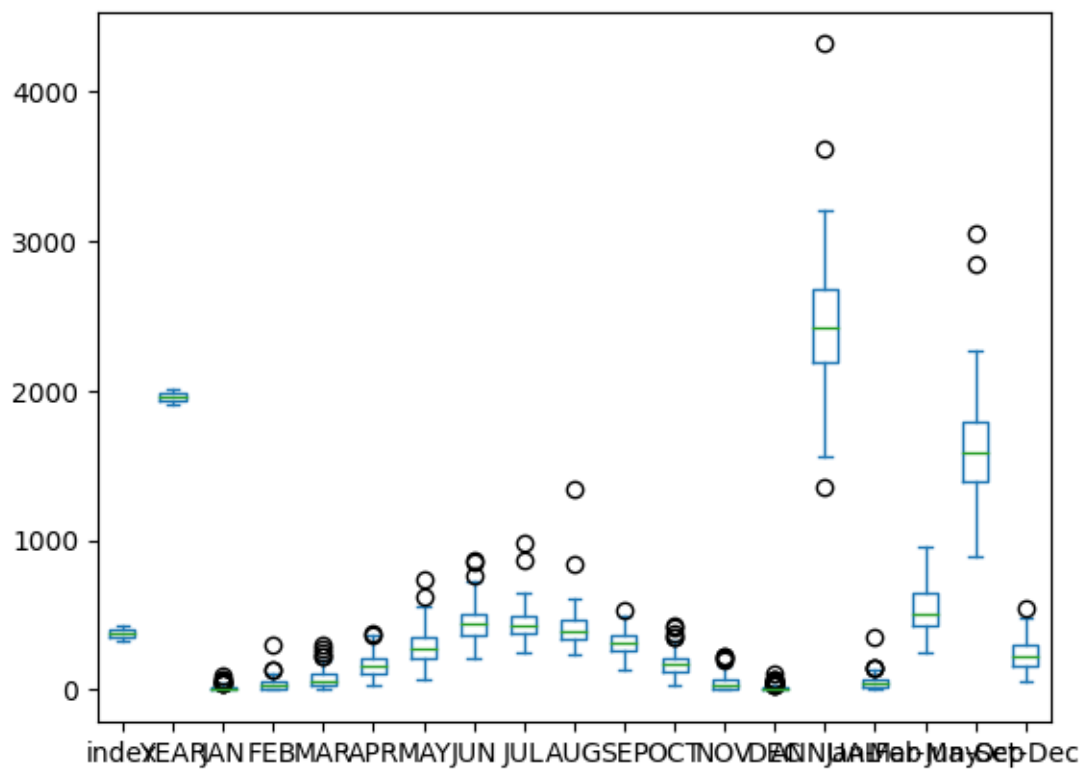
```
[10]: <Axes: >
```



9 Box chart

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

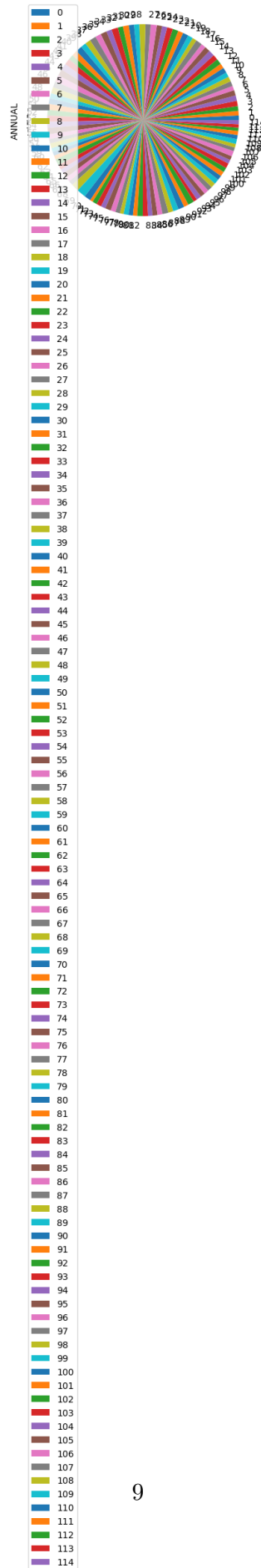
```
[11]: <Axes: >
```



10 Pie chart

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

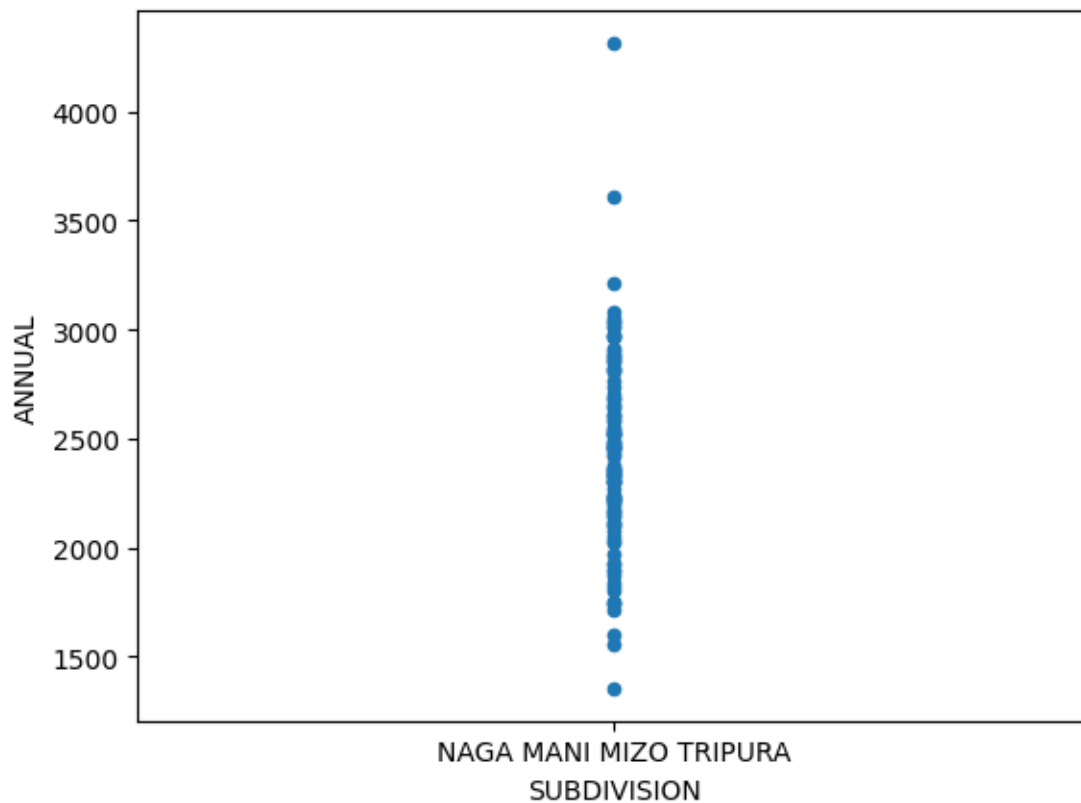
```
[12]: <Axes: ylabel='ANNUAL'>
```

11 Scatter chart

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

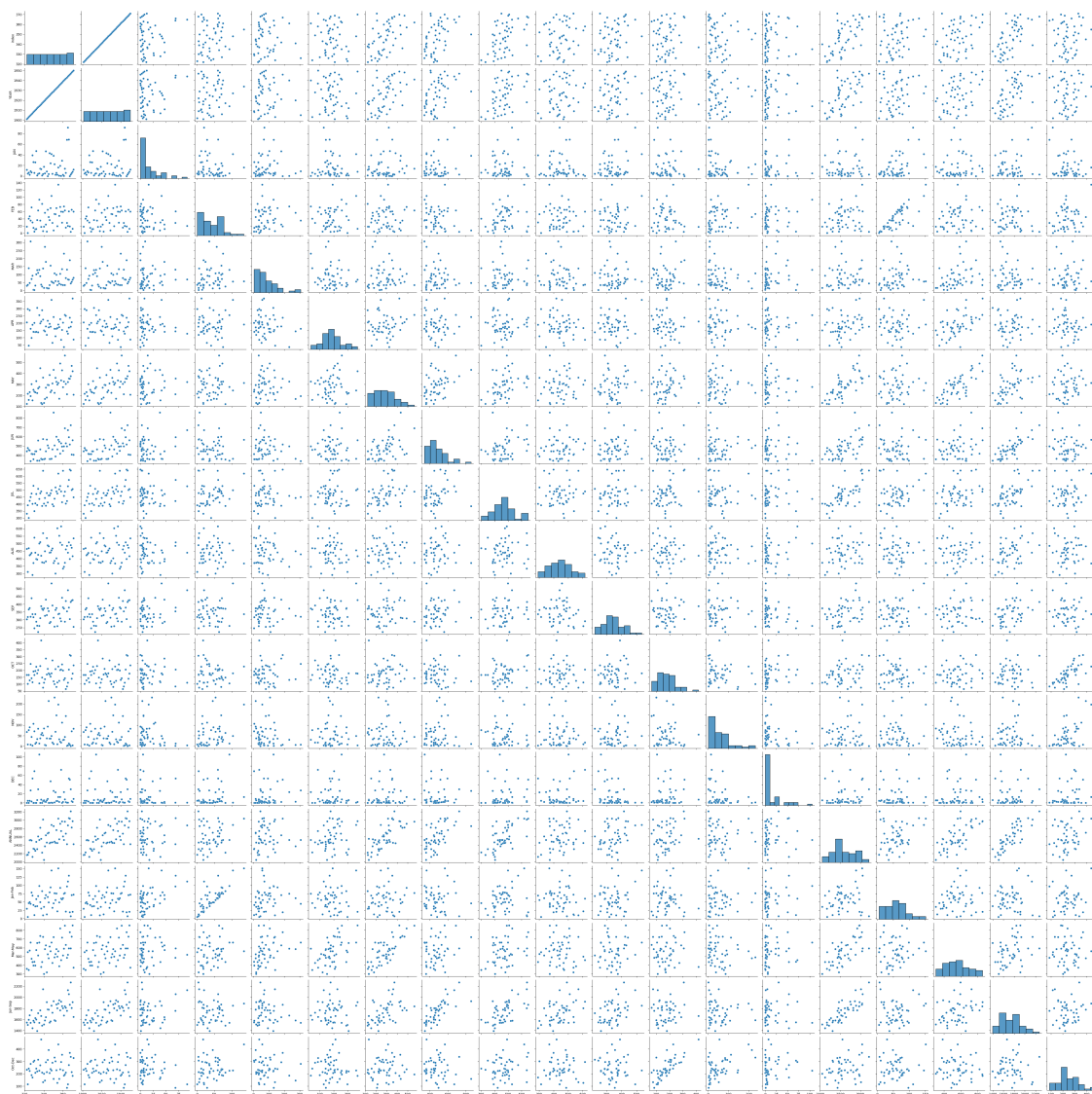
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7b0894d59ed0>
```



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

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

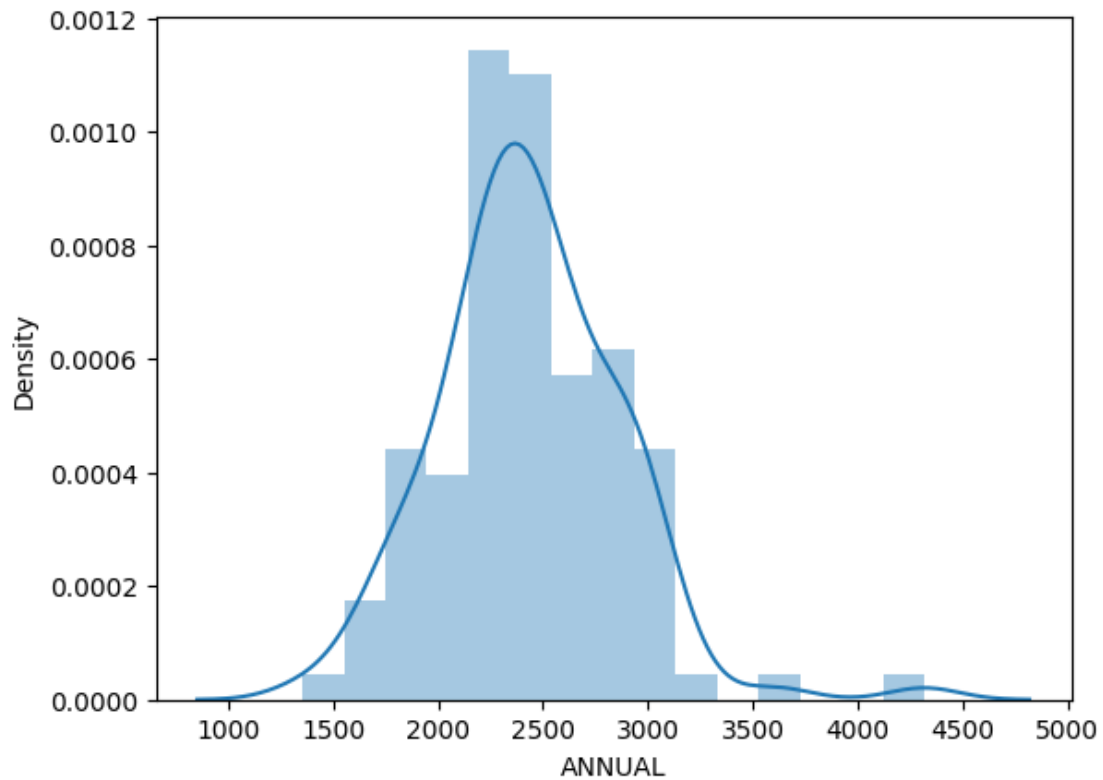
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

