

aseqge6ku

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_chhattisgarh.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	\
0	2967	CHHATTISGARH	1901	48.9	116.5	27.8	5.5	18.4	101.6	381.0	
1	2968	CHHATTISGARH	1902	0.6	6.5	0.4	13.9	10.3	37.2	403.8	
2	2969	CHHATTISGARH	1903	6.2	13.9	0.4	6.8	51.1	110.7	365.9	
3	2970	CHHATTISGARH	1904	0.0	8.6	32.3	0.2	77.5	369.5	303.6	
4	2971	CHHATTISGARH	1905	50.3	22.6	19.0	24.6	31.8	40.4	443.7	
..	...	...	...	...	...	...	...	...	...	...	
110	3077	CHHATTISGARH	2011	0.3	11.5	2.6	35.0	16.8	183.5	272.6	
111	3078	CHHATTISGARH	2012	36.6	4.8	1.1	14.9	9.4	147.3	430.6	
112	3079	CHHATTISGARH	2013	2.8	19.7	4.9	45.8	5.7	263.6	418.8	
113	3080	CHHATTISGARH	2014	2.3	29.0	21.4	17.3	25.0	104.9	416.7	
114	3081	CHHATTISGARH	2015	15.8	1.2	21.2	37.0	13.0	257.6	248.6	
..	...	...	...	...	...	...	...	...	...	...	
	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	\	
0	476.7	182.8	27.3	0.4	0.0	1387.0	165.4	51.7	1142.2		
1	236.6	198.1	4.7	8.1	3.7	923.9	7.1	24.6	875.7		
2	396.0	212.0	168.0	0.1	0.0	1331.2	20.1	58.3	1084.7		
3	483.6	86.8	129.3	1.0	0.0	1492.4	8.6	110.0	1243.5		
4	270.8	338.8	8.9	0.0	0.0	1251.1	72.9	75.5	1093.8		
..	...	...	...	...	...	...	...	...	...		

```

110  379.8  382.2   15.5   0.0  2.8  1302.7    11.8    54.4  1218.2
111  442.2  245.3   19.8  20.4  5.0  1377.4    41.5    25.4  1265.4
112  336.6  140.9  180.9   0.3  0.0  1420.0    22.5    56.4  1159.9
113  327.7  252.7   77.9   2.6  1.1  1278.7    31.3    63.7  1102.1
114  286.6  216.9   17.7   0.6  1.5  1117.6    17.0    71.2  1009.7

```

```

      Oct-Dec
0      27.7
1      16.5
2     168.1
3     130.4
4       8.9
..      ...
110     18.3
111     45.1
112    181.2
113     81.6
114     19.7

```

```
[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

```

```

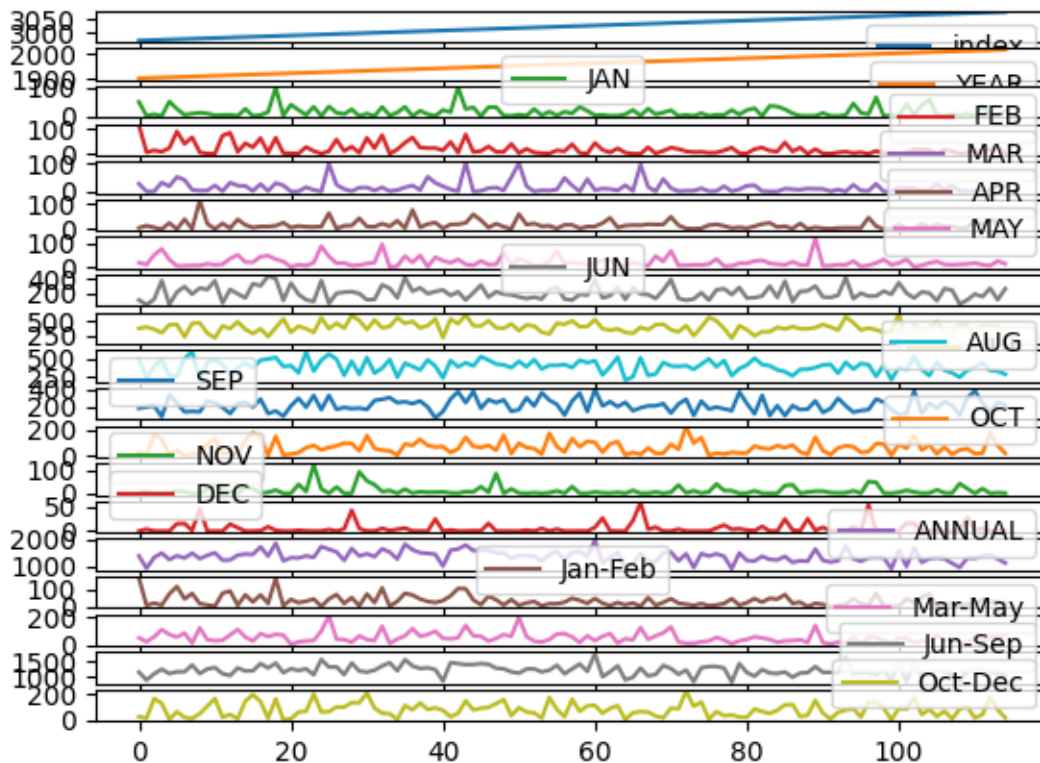
7  MAY          115 non-null    float64
8  JUN          115 non-null    float64
9  JUL          115 non-null    float64
10 AUG          115 non-null    float64
11 SEP          115 non-null    float64
12 OCT          115 non-null    float64
13 NOV          115 non-null    float64
14 DEC          115 non-null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

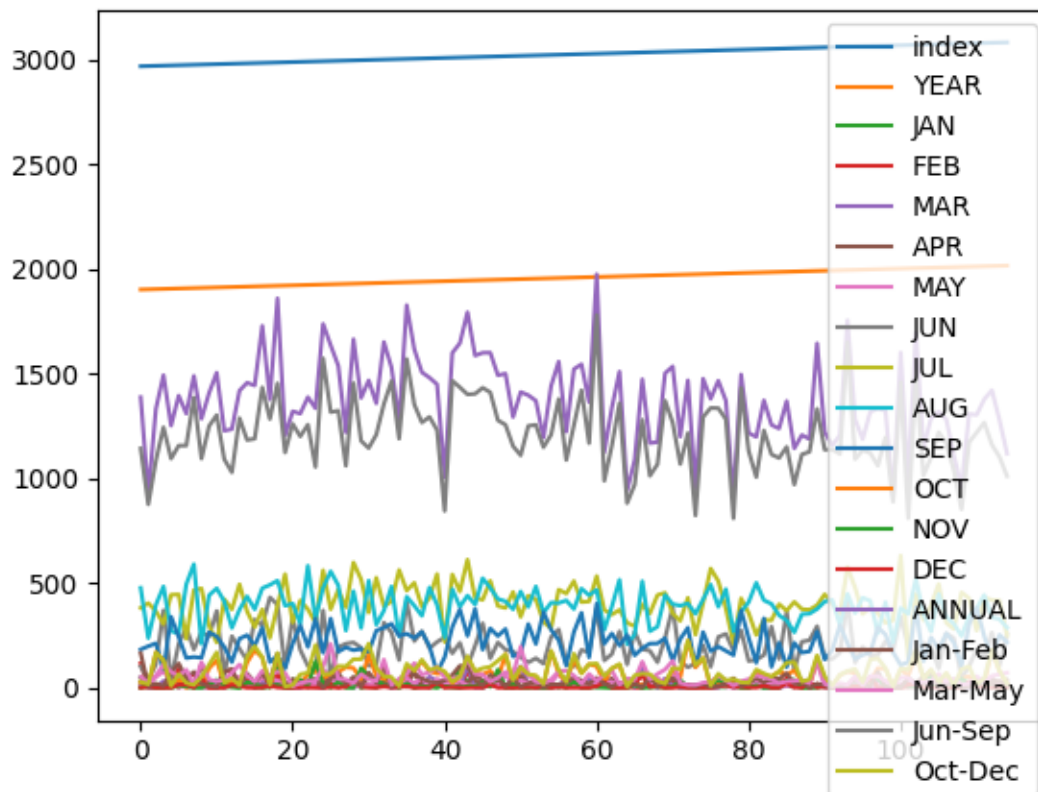
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



## 5 Line chart

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

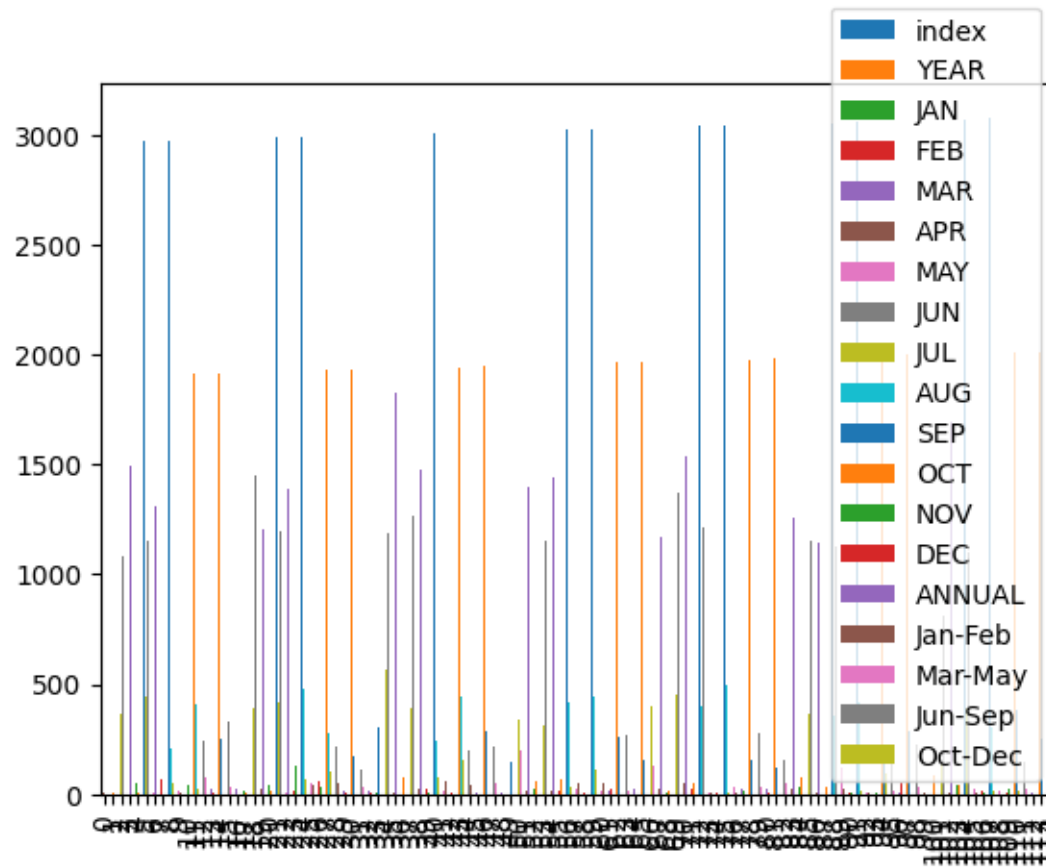
```
[7]: <Axes: >
```



## 6 Bar chart

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

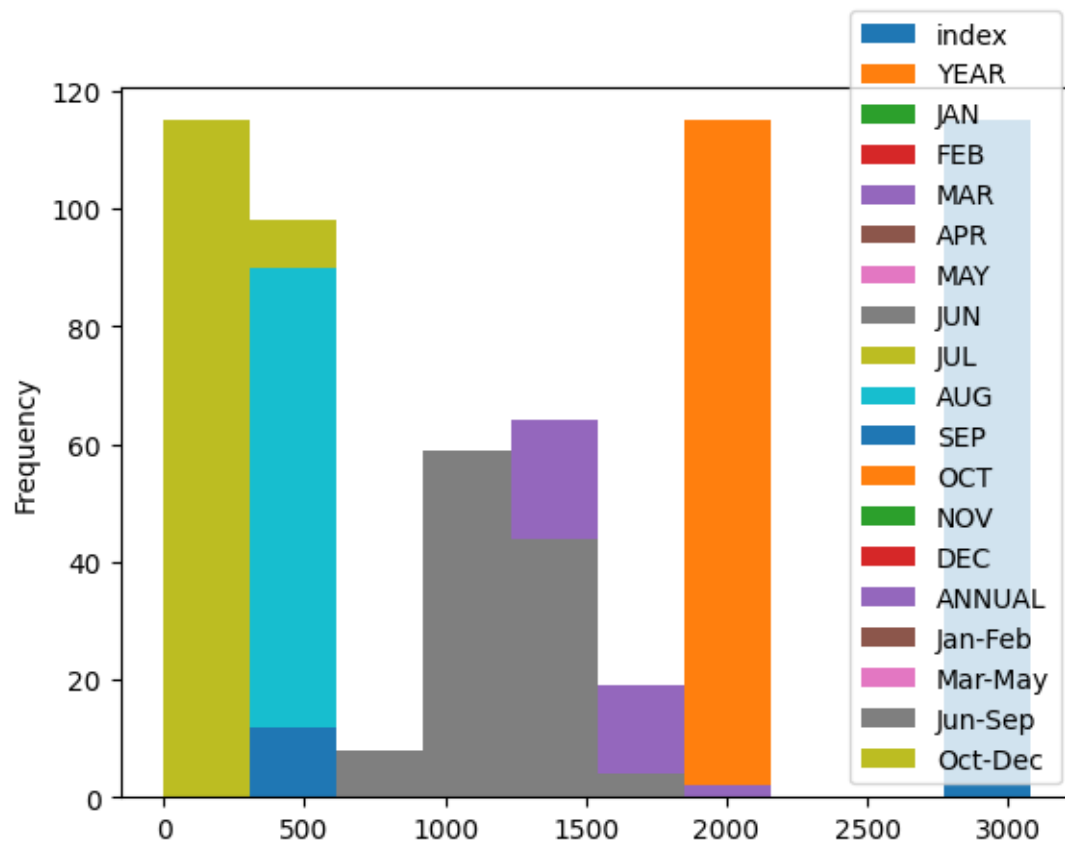
```
[8]: <Axes: >
```



## 7 Histogram

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

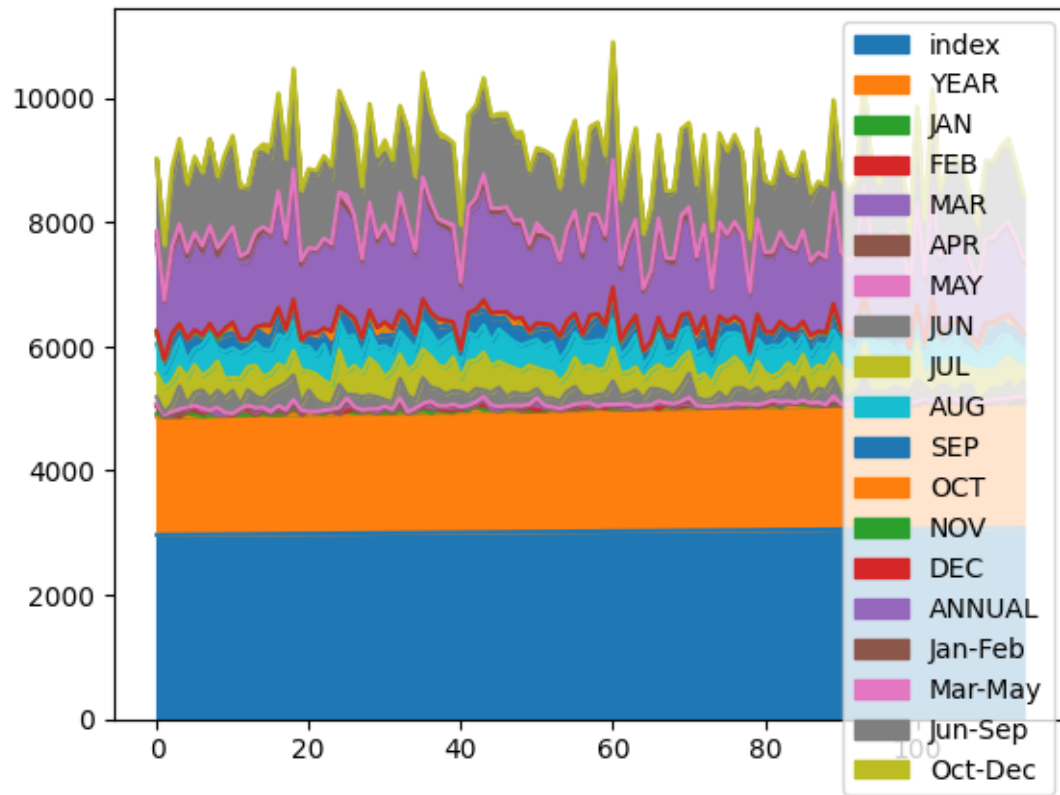
```
[9]: <Axes: ylabel='Frequency'>
```



## 8 Area chart

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

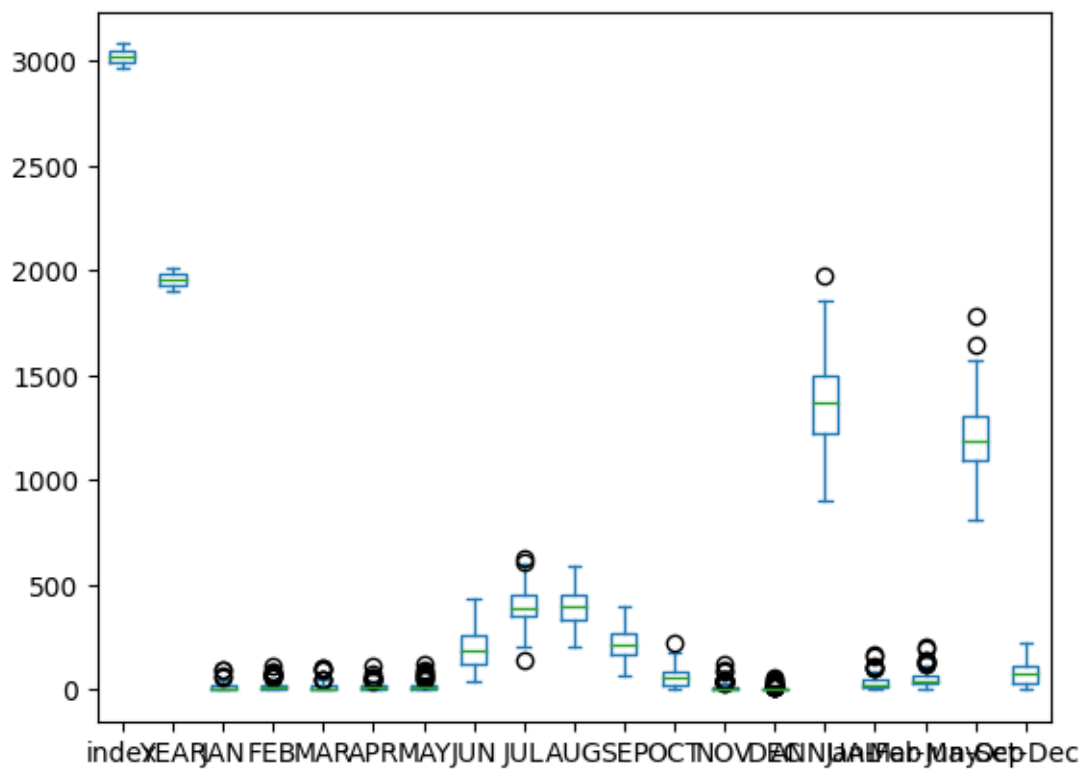
```
[10]: <Axes: >
```



## 9 Box chart

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

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

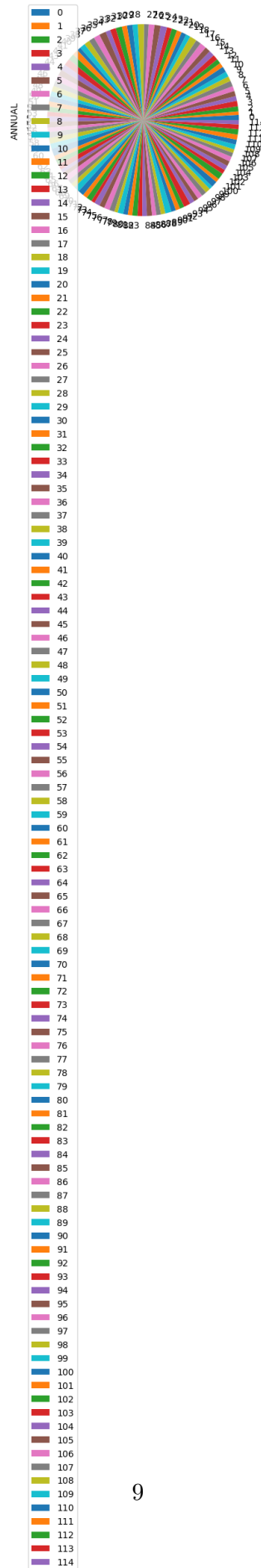


## 10 Pie chart

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

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

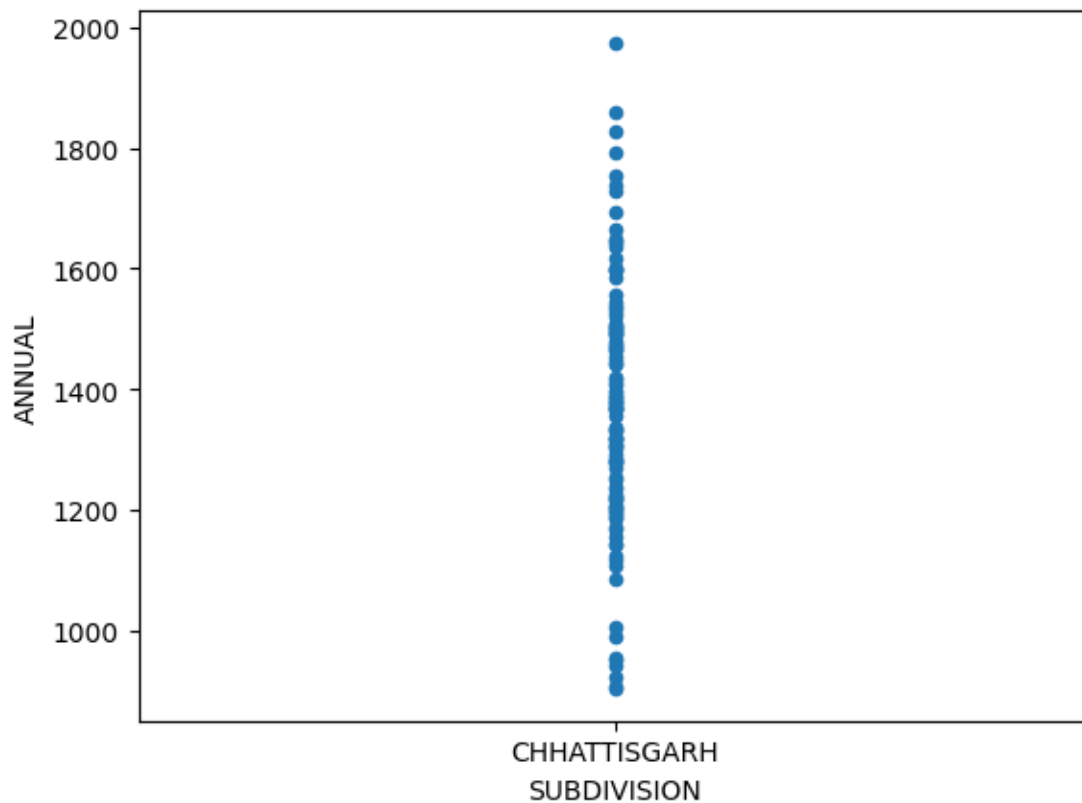




## 11 Scatter chart

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

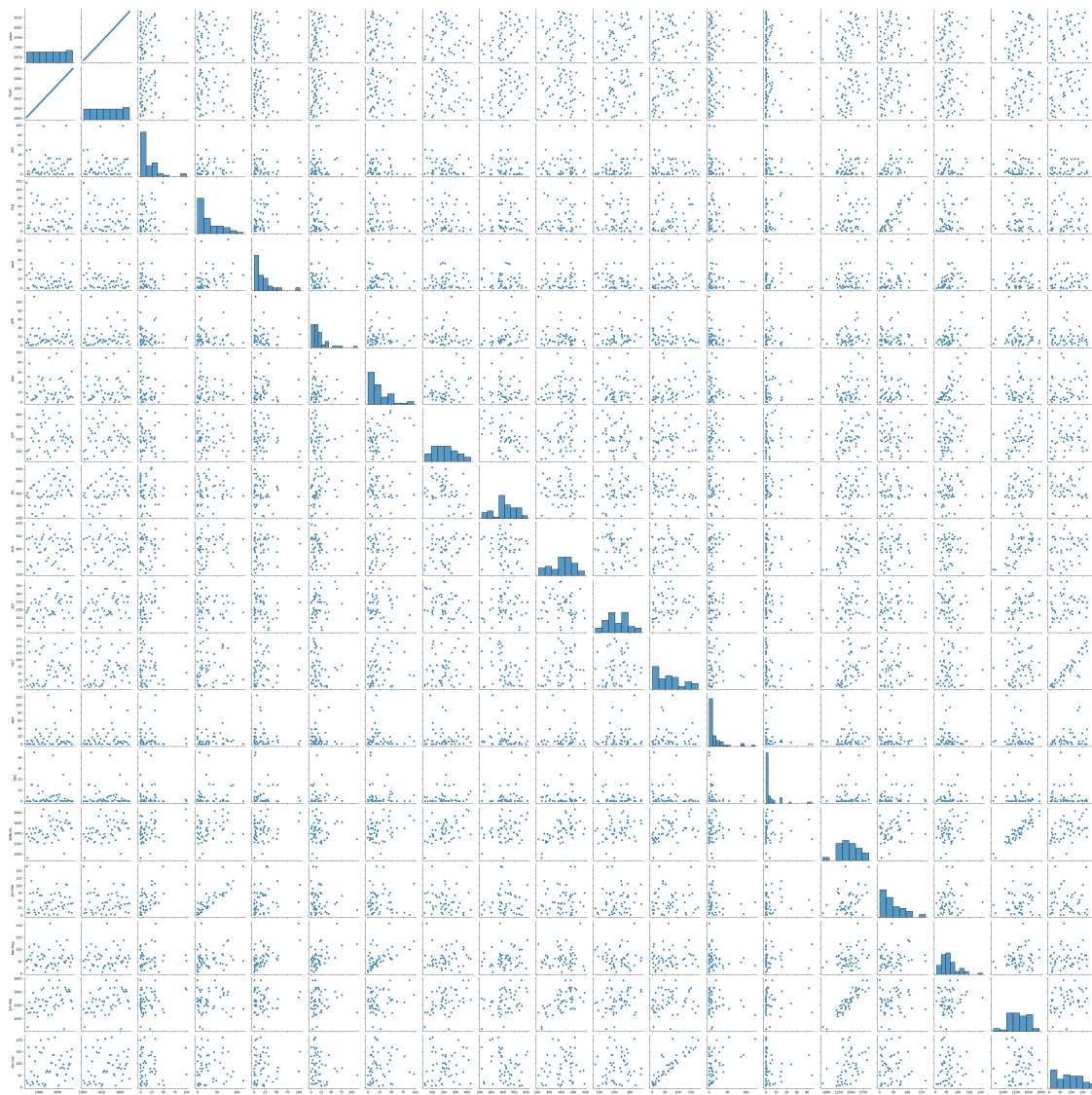
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



## 12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7bf191174ee0>
```



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

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

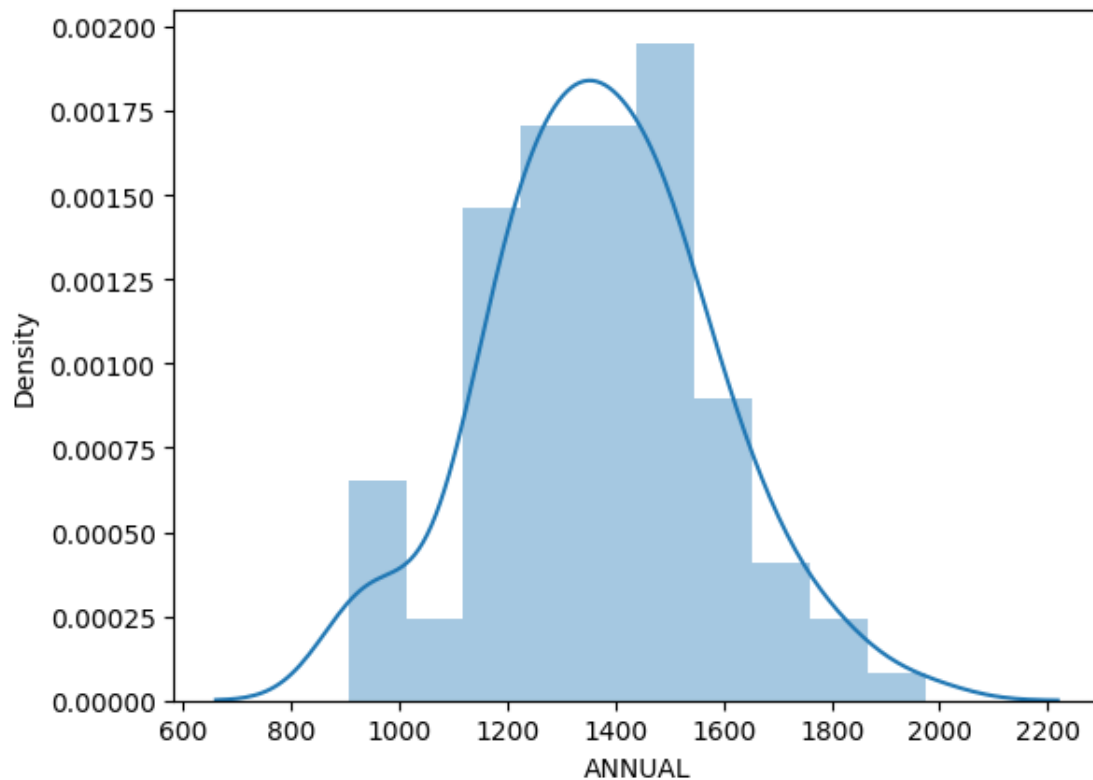
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

