

ltgziz0rq

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_north_
↪interior_karnataka.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index		SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	\		
0	3657	NORTH	INTERIOR KARNATAKA	1901	3.5	18.8	7.1	67.2	65.5			
1	3658	NORTH	INTERIOR KARNATAKA	1902	0.0	0.0	0.3	22.5	34.4			
2	3659	NORTH	INTERIOR KARNATAKA	1903	3.5	0.0	0.1	6.9	53.4			
3	3660	NORTH	INTERIOR KARNATAKA	1904	0.2	0.3	8.5	11.0	46.3			
4	3661	NORTH	INTERIOR KARNATAKA	1905	0.0	6.0	2.6	16.0	51.2			
..	...			...	...	...	...	...	...			
110	3767	NORTH	INTERIOR KARNATAKA	2011	0.5	7.2	7.2	41.2	46.8			
111	3768	NORTH	INTERIOR KARNATAKA	2012	28.5	6.2	0.4	35.4	19.5			
112	3769	NORTH	INTERIOR KARNATAKA	2013	1.2	6.1	3.0	25.4	47.4			
113	3770	NORTH	INTERIOR KARNATAKA	2014	0.0	6.1	29.2	26.4	93.0			
114	3771	NORTH	INTERIOR KARNATAKA	2015	2.4	0.0	27.5	50.8	45.3			
		JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	\
0	120.5	151.9	115.1	128.8	80.0	13.6	0.9	773.0	22.3	139.8		
1	111.3	83.2	78.1	146.7	118.8	35.7	85.1	716.1	0.0	57.1		
2	102.8	209.4	146.4	189.3	166.4	34.3	16.0	928.5	3.5	60.3		
3	120.6	91.6	48.5	165.1	86.5	0.0	0.0	578.6	0.5	65.8		
4	99.6	60.1	139.2	42.2	85.0	4.4	0.0	506.2	6.0	69.8		
..	...	...	...	...	...	...	...	...	...	...		

110	101.3	150.8	152.0	69.0	73.4	5.7	0.0	655.2	7.7	95.2
111	60.0	114.5	105.5	79.2	85.2	46.5	2.9	583.8	34.7	55.3
112	99.4	160.7	73.9	201.0	101.0	4.2	0.1	723.2	7.3	75.7
113	50.4	136.8	205.2	90.2	80.3	25.0	14.1	756.8	6.1	148.7
114	89.6	38.5	78.4	150.8	61.2	5.7	1.7	551.9	2.4	123.6

	Jun-Sep	Oct-Dec
0	516.4	94.5
1	419.3	239.6
2	647.9	216.8
3	425.8	86.5
4	341.1	89.4
..	...	...
110	473.1	79.2
111	359.3	134.5
112	534.9	105.2
113	482.5	119.5
114	357.2	68.6

[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
```

```

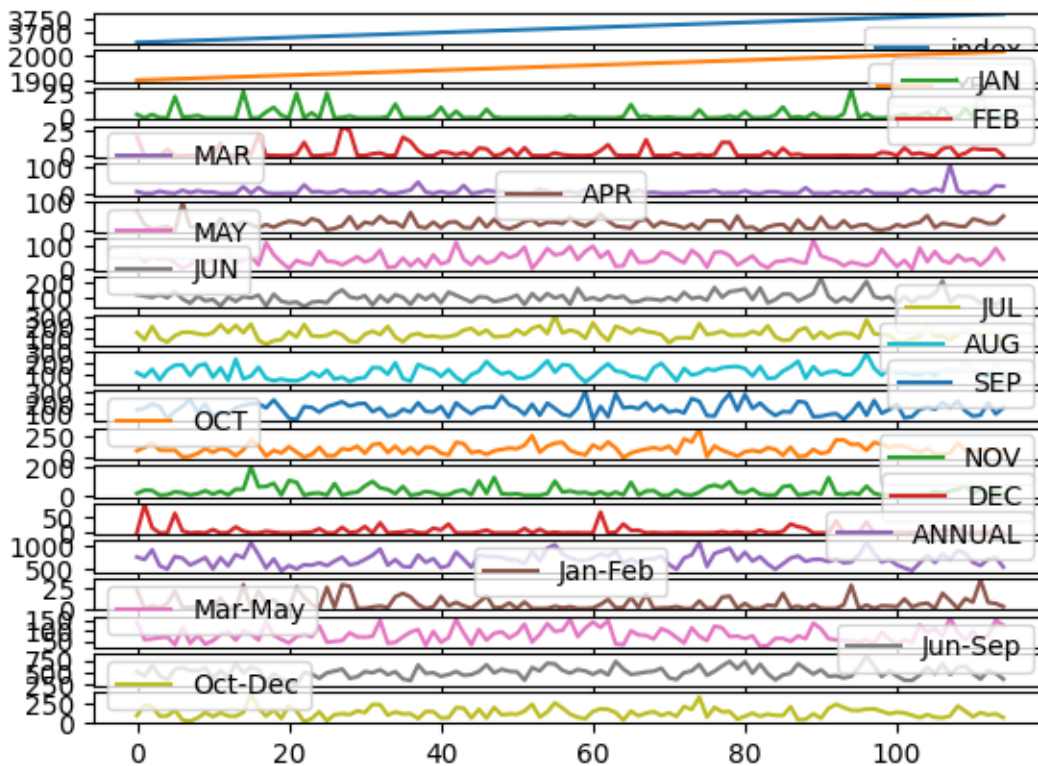
7 MAY 115 non-null float64
8 JUN 115 non-null float64
9 JUL 115 non-null float64
10 AUG 115 non-null float64
11 SEP 115 non-null float64
12 OCT 115 non-null float64
13 NOV 115 non-null float64
14 DEC 115 non-null float64
15 ANNUAL 115 non-null float64
16 Jan-Feb 115 non-null float64
17 Mar-May 115 non-null 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)
```

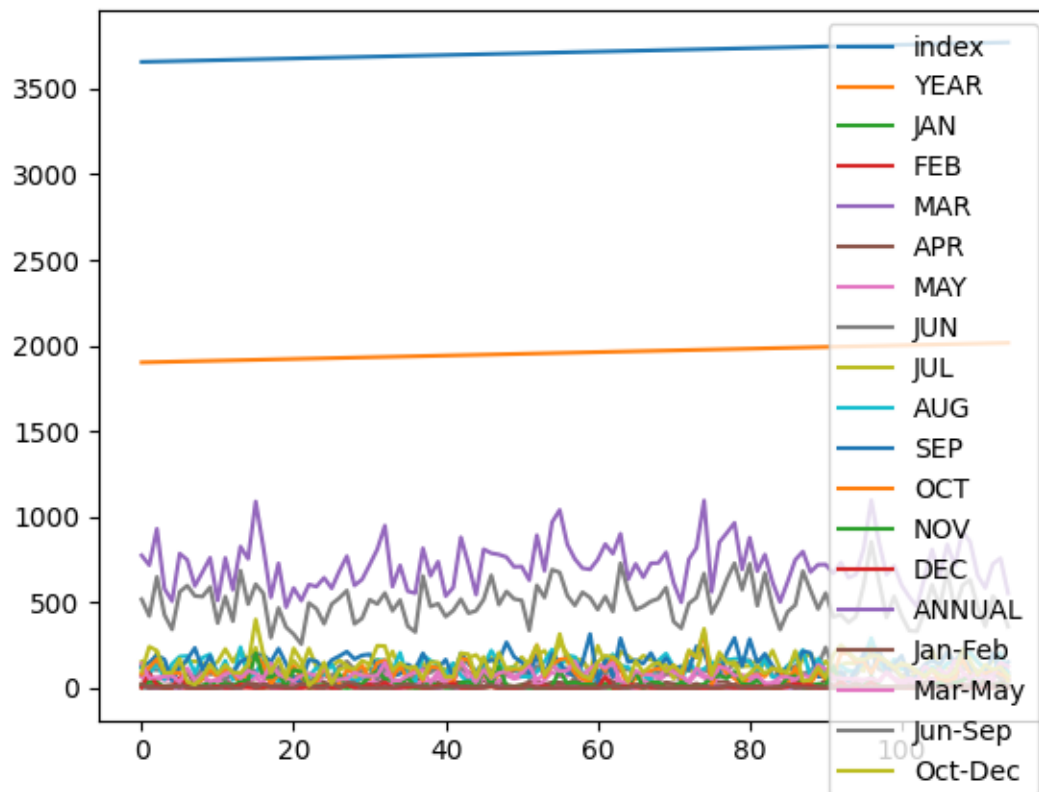
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



## 5 Line chart

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

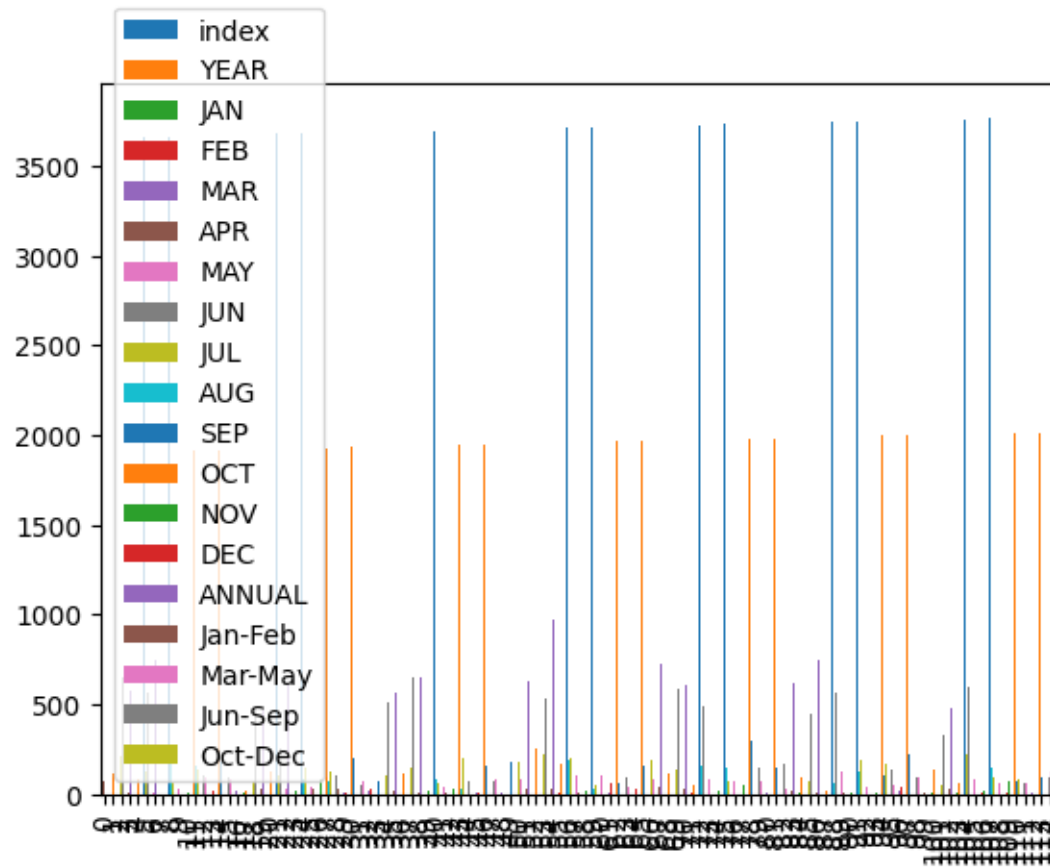
```
[7]: <Axes: >
```



## 6 Bar chart

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

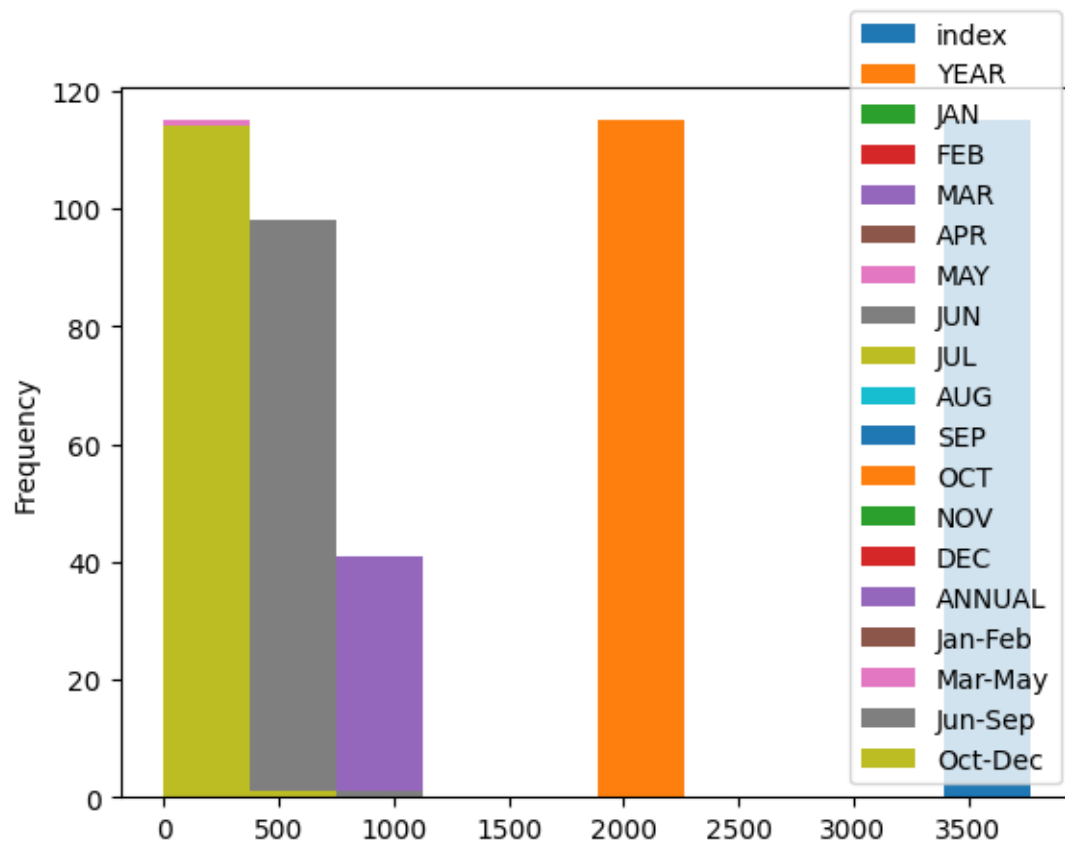
```
[8]: <Axes: >
```



## 7 Histogram

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

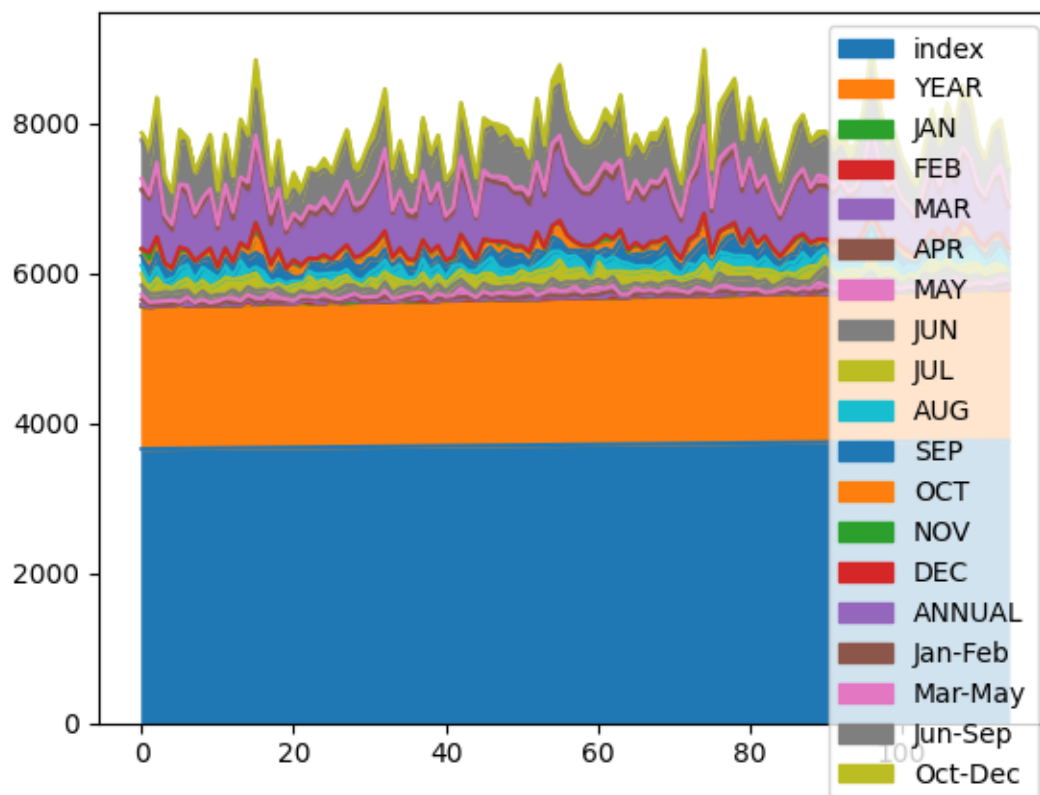
```
[9]: <Axes: ylabel='Frequency'>
```



## 8 Area chart

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

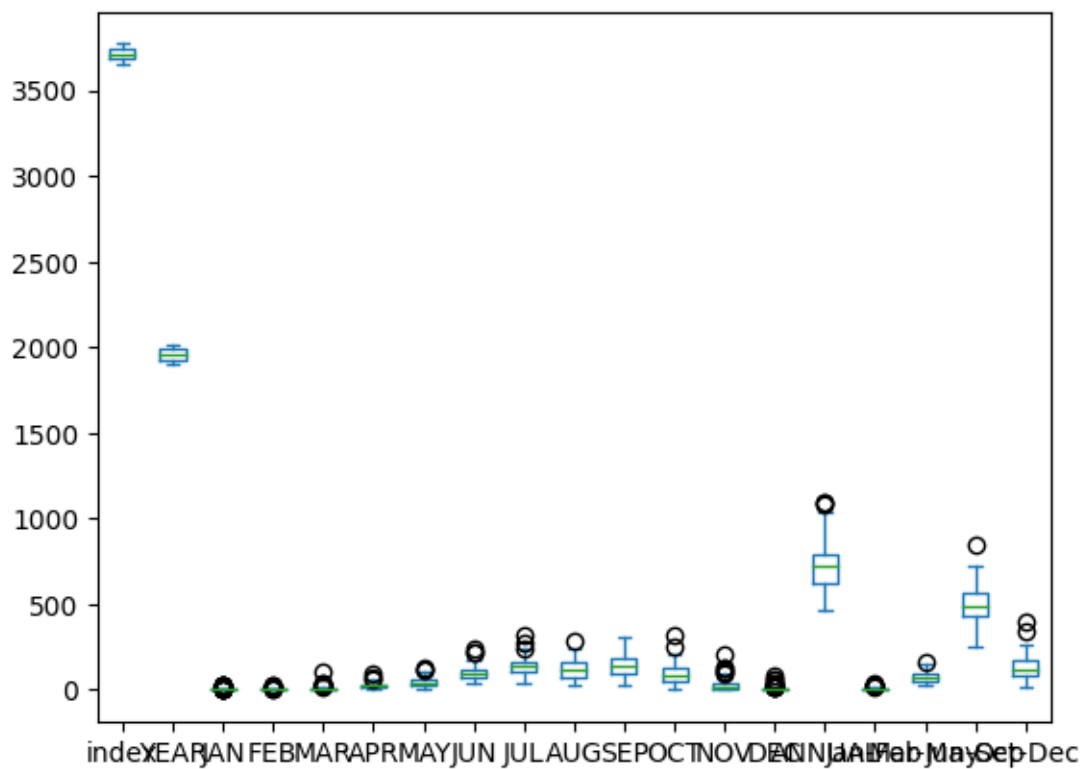
```
[10]: <Axes: >
```



## 9 Box chart

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

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

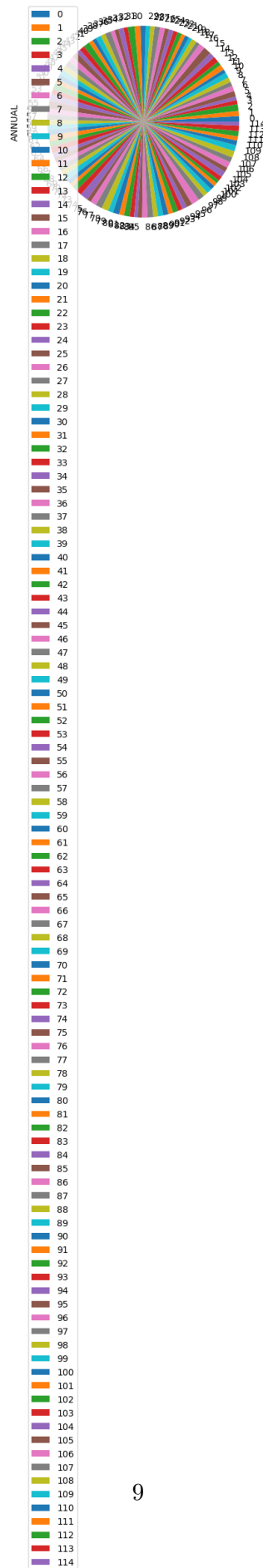


## 10 Pie chart

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

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

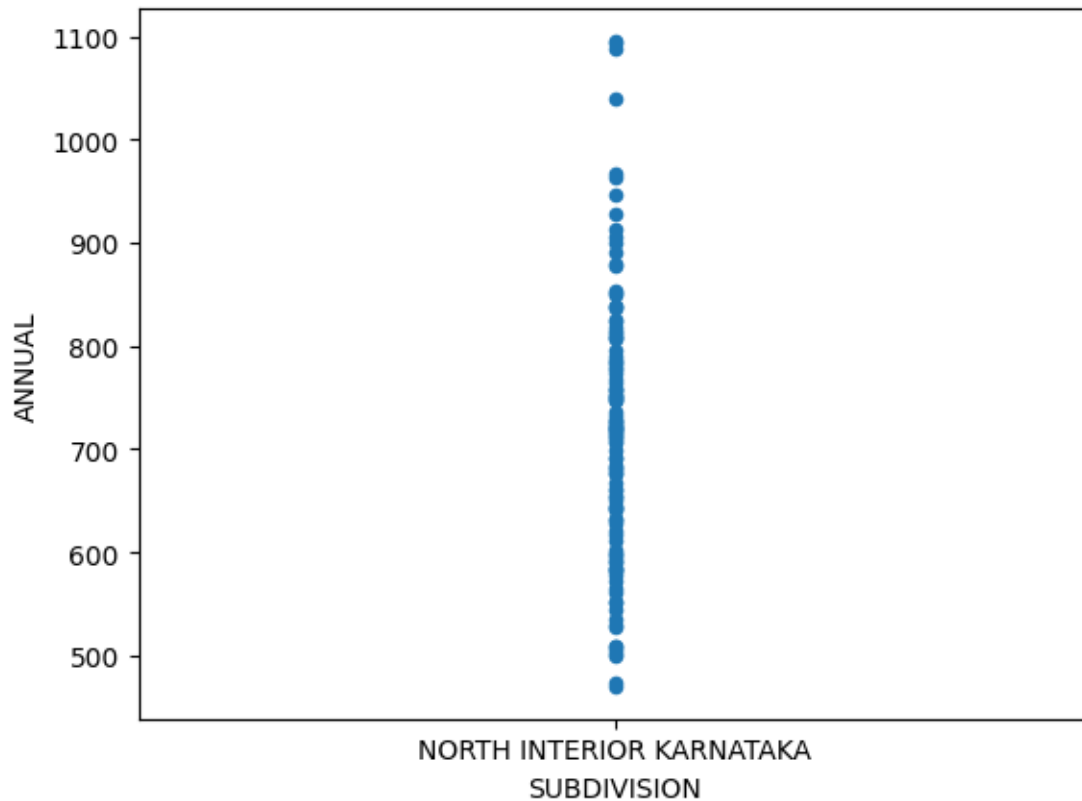




## 11 Scatter chart

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

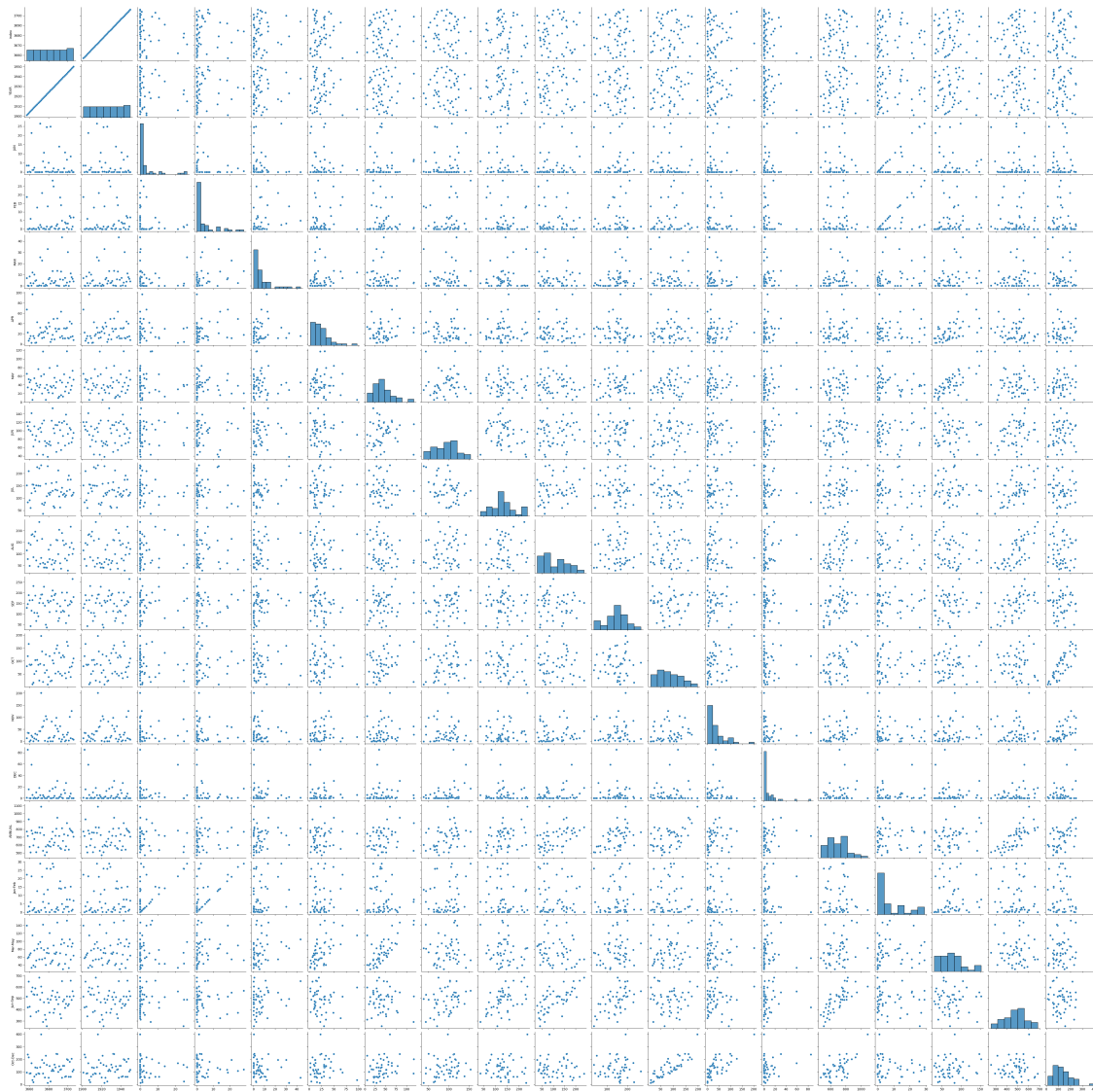
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



## 12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7bb84d647f70>
```



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

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

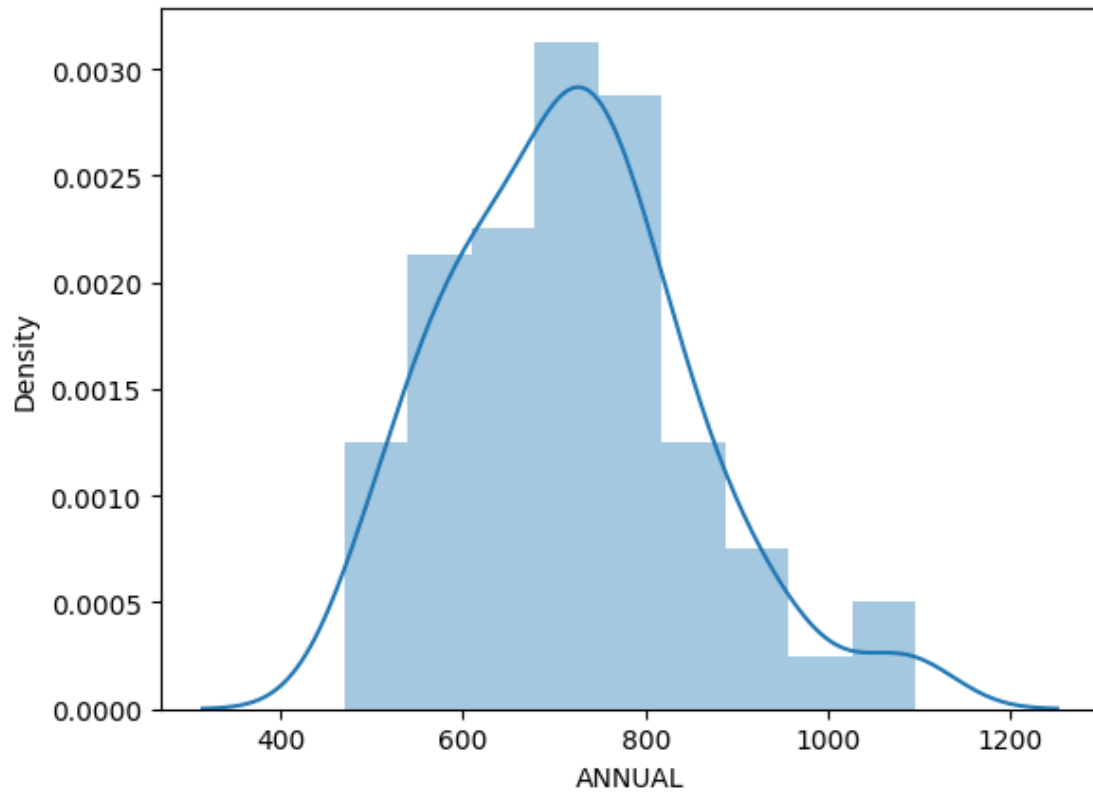
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

