

xtbzl9glf

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_gujarat_
↪region.csv")
df
```

Mounted at /content/drive

```
[2]:
```

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	\
0	2277	GUJARAT REGION	1901	4.2	0.0	0.6	1.6	7.0	60.3	240.2	
1	2278	GUJARAT REGION	1902	3.9	0.0	0.0	0.6	1.0	32.8	229.8	
2	2279	GUJARAT REGION	1903	0.3	0.1	1.4	0.0	12.3	30.1	452.9	
3	2280	GUJARAT REGION	1904	0.8	10.6	16.8	0.2	3.9	48.3	194.8	
4	2281	GUJARAT REGION	1905	0.1	0.7	1.1	0.3	0.0	20.1	668.3	
..	...	...	...	...	...	...	...	...	...	...	
110	2387	GUJARAT REGION	2011	0.0	0.2	0.0	0.0	0.0	16.3	259.2	
111	2388	GUJARAT REGION	2012	0.1	0.0	0.0	0.0	0.0	34.4	178.2	
112	2389	GUJARAT REGION	2013	0.0	0.9	0.1	4.6	0.0	155.7	405.4	
113	2390	GUJARAT REGION	2014	5.7	0.1	0.2	1.0	1.3	11.6	307.5	
114	2391	GUJARAT REGION	2015	1.8	0.0	6.1	5.5	0.9	120.7	354.7	
		AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	205.4	18.1	16.6	0.0	0.3	554.3	4.2	9.2	524.0	17.0	
1	299.0	281.2	2.3	1.5	11.9	863.9	3.9	1.5	842.8	15.7	
2	202.0	183.2	5.4	0.0	0.0	887.6	0.3	13.6	868.2	5.4	
3	71.8	138.0	6.1	0.1	1.2	492.6	11.3	20.9	453.0	7.4	
4	37.9	81.3	1.4	0.2	0.1	811.5	0.8	1.3	807.6	1.7	
..	...	...	...	...	...	...	...	...	...	...	

110	451.7	162.5	0.4	0.0	0.0	890.5	0.2	0.0	889.8	0.4
111	230.3	263.8	7.1	0.0	0.0	714.0	0.1	0.0	706.7	7.1
112	211.1	287.3	53.2	0.1	0.0	1118.6	0.9	4.7	1059.6	53.3
113	138.6	235.1	3.3	1.3	0.0	705.7	5.8	2.5	692.8	4.6
114	37.4	93.4	2.2	0.3	0.0	622.9	1.8	12.5	606.1	2.6

[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()
```

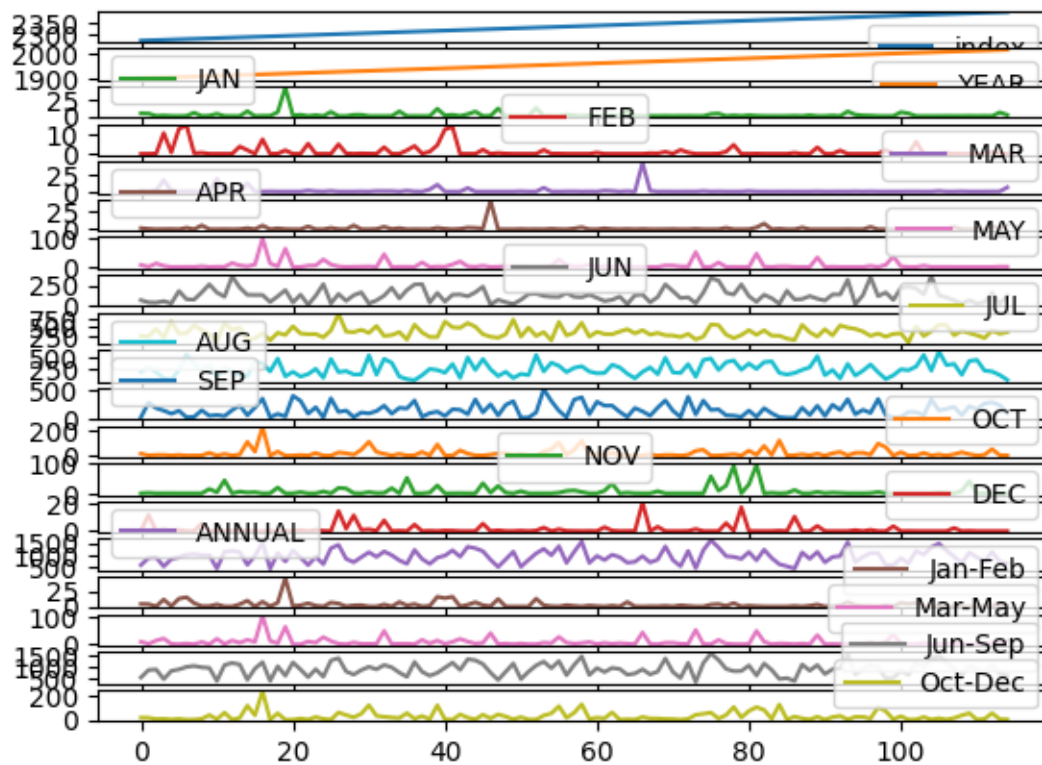
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 20 columns):
#   Column          Non-Null Count  Dtype
---  -
0   index           115 non-null   int64
1   SUBDIVISION     115 non-null   object
2   YEAR            115 non-null   int64
3   JAN           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

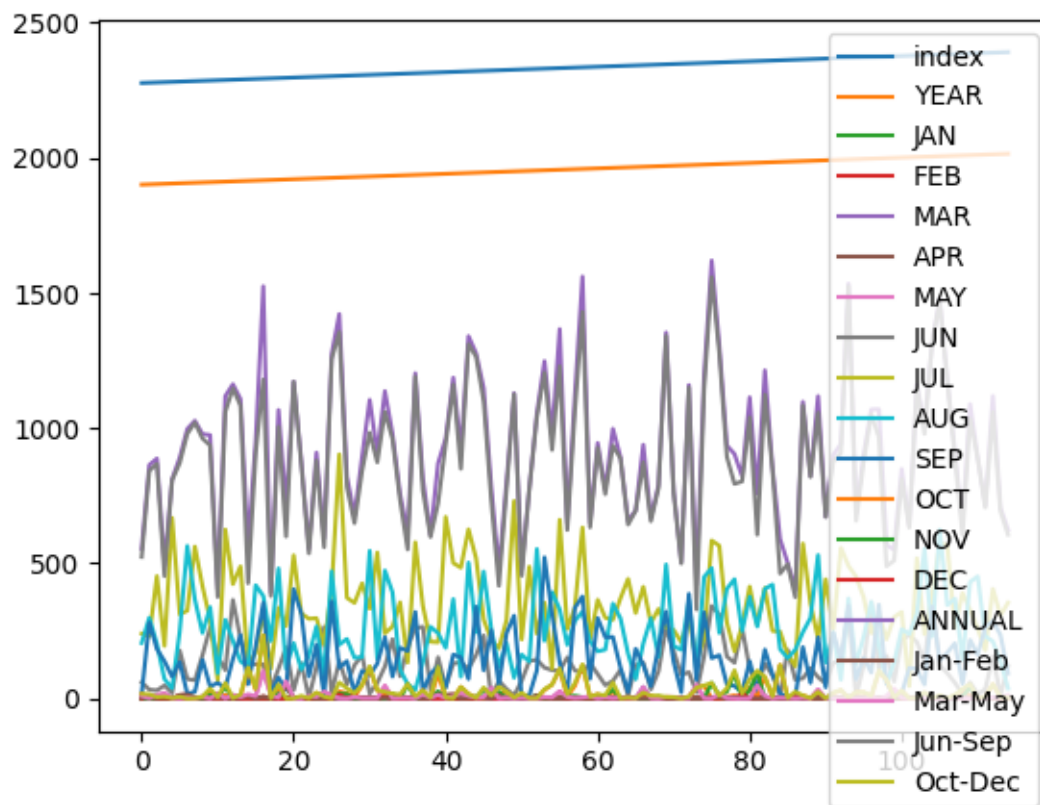
```
[6]: array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >,
<Axes: >], dtype=object)
```



## 5 Line chart

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

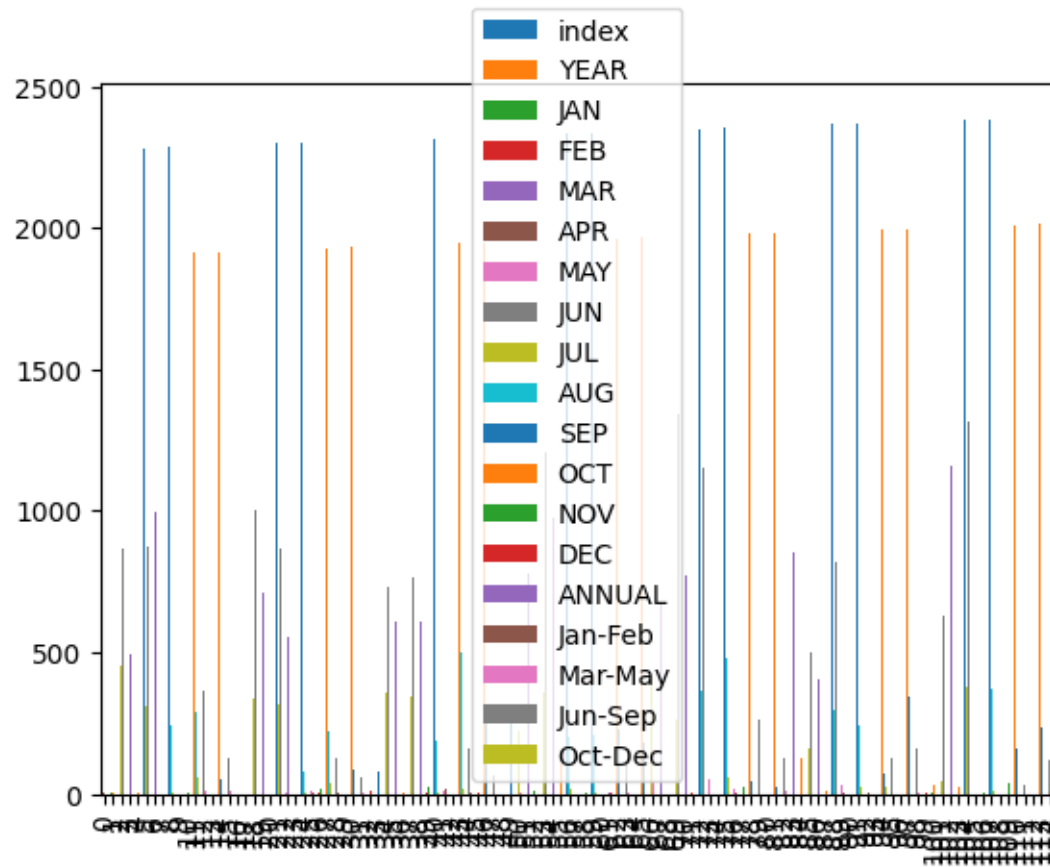
```
[7]: <Axes: >
```



## 6 Bar chart

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

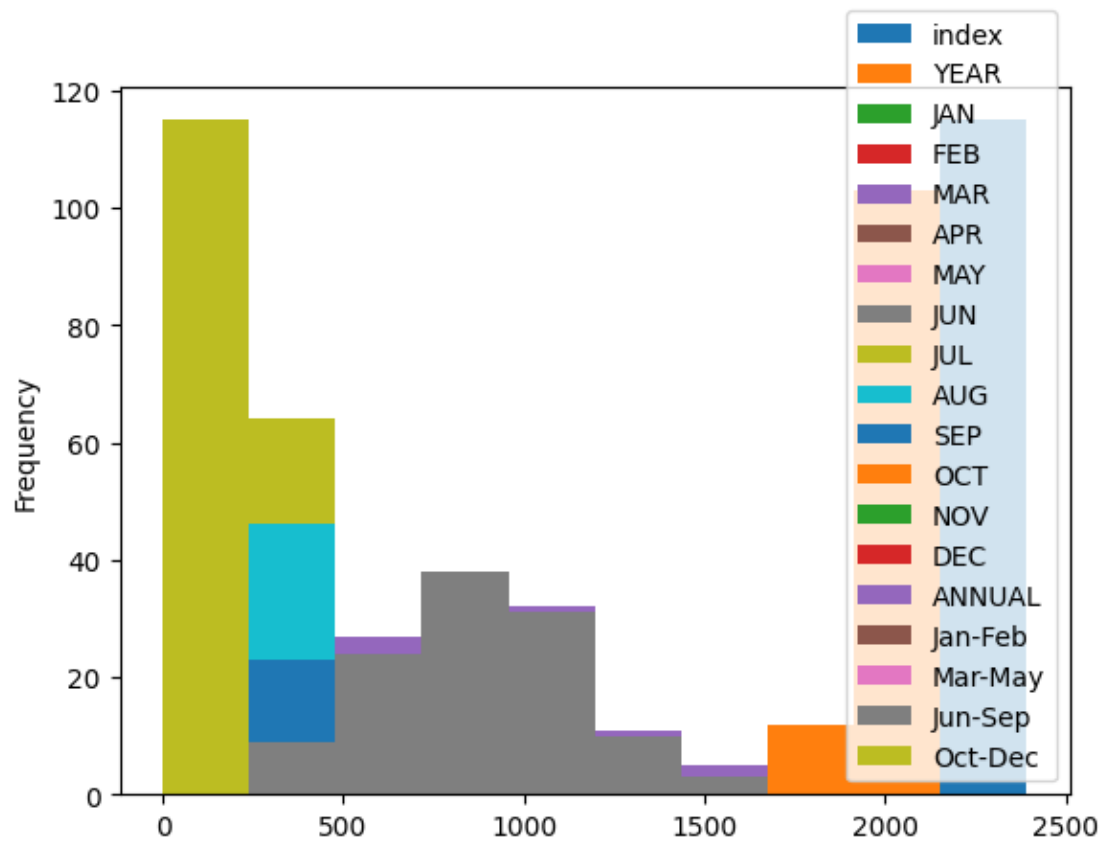
```
[8]: <Axes: >
```



## 7 Histogram

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

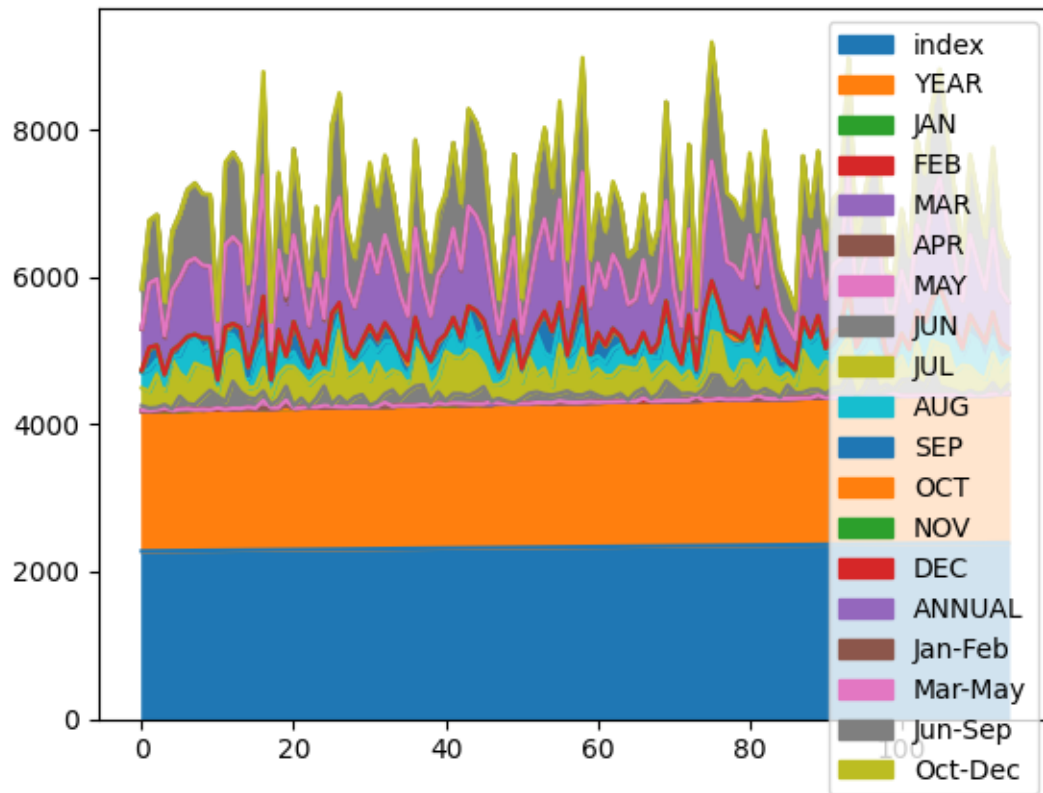
```
[9]: <Axes: ylabel='Frequency'>
```



## 8 Area chart

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

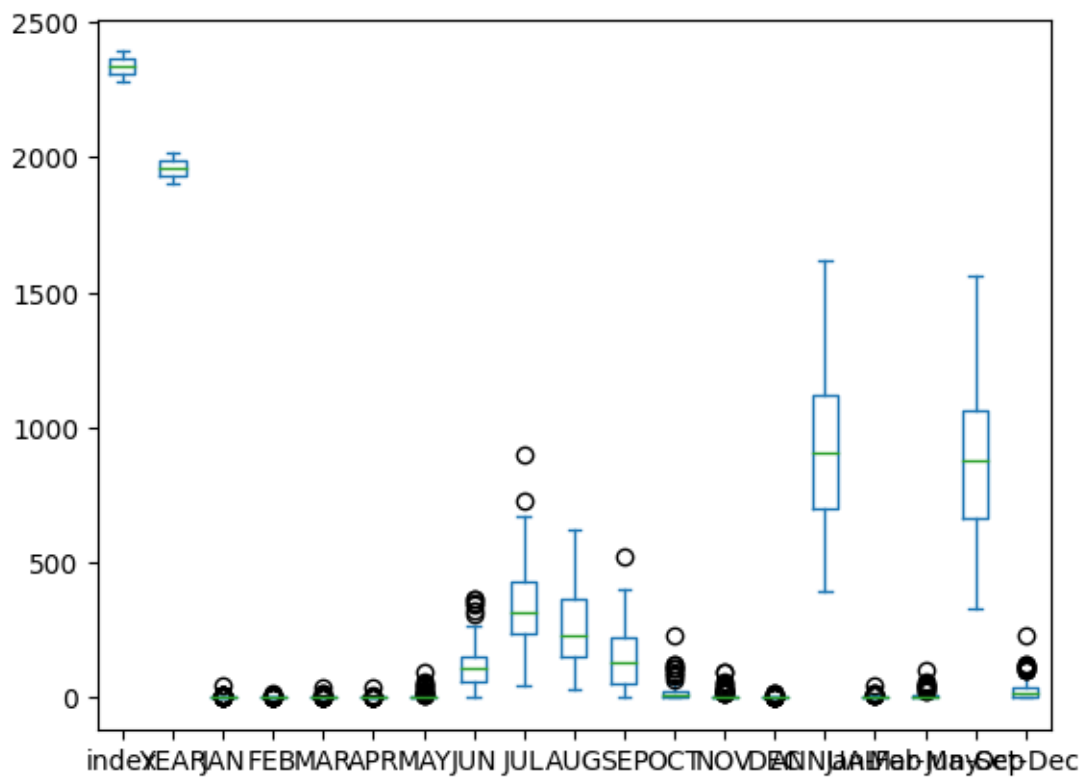
```
[10]: <Axes: >
```



## 9 Box chart

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

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

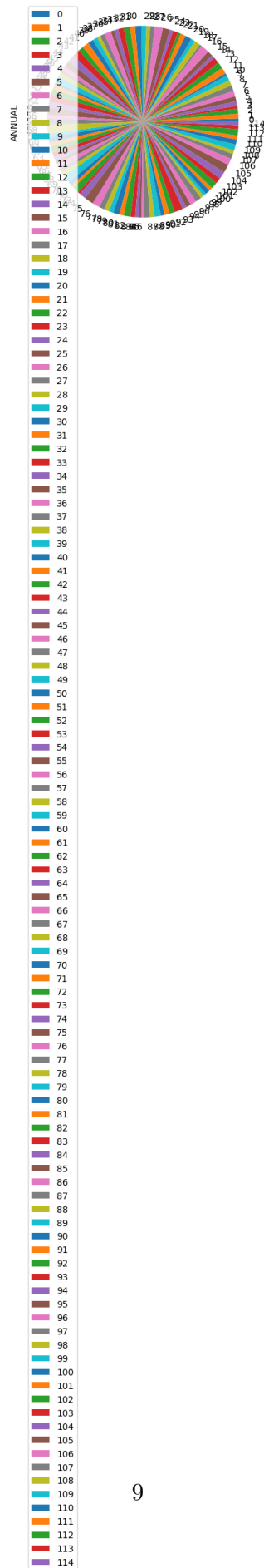


## 10 Pie chart

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

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

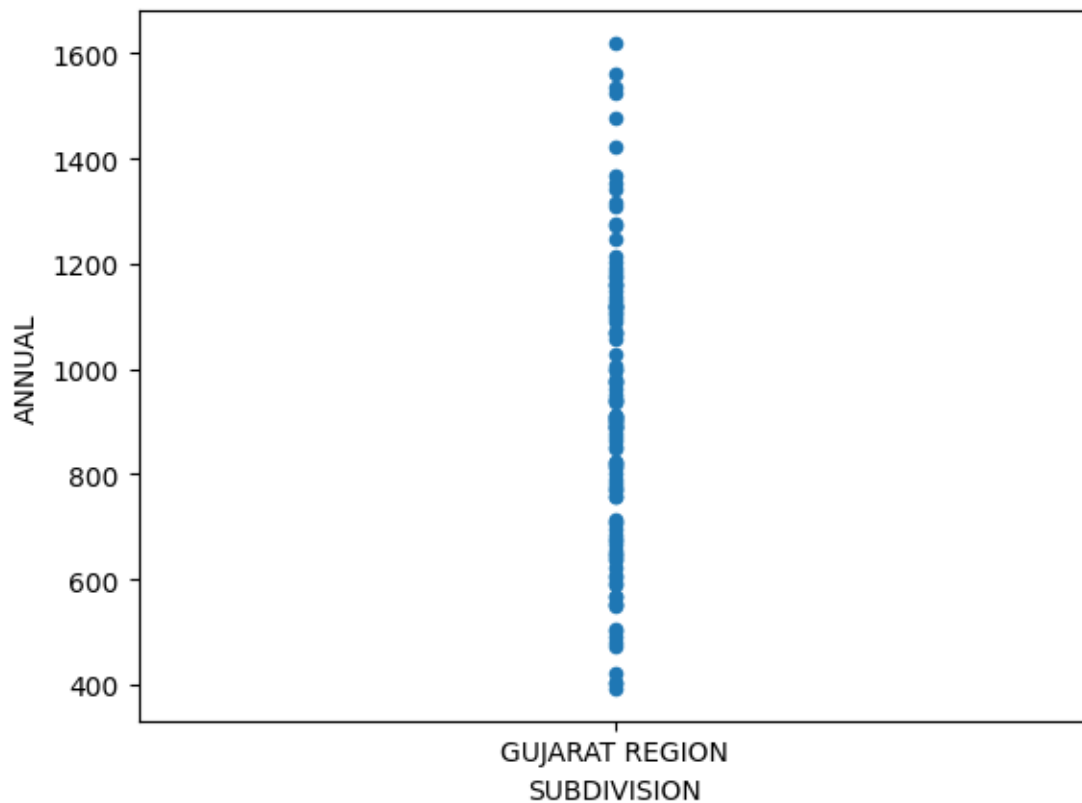




## 11 Scatter chart

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

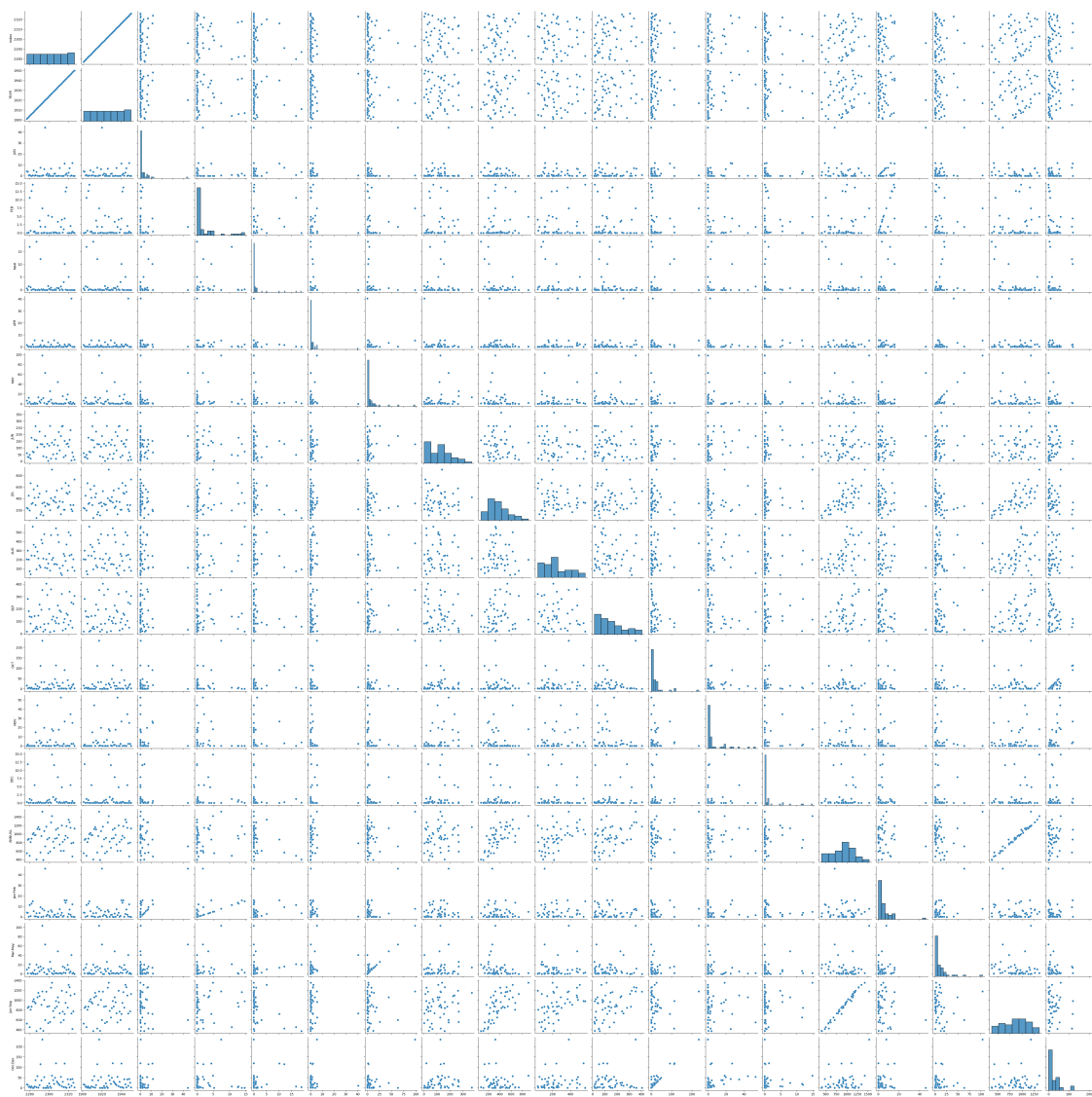
```
[13]: <Axes: xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



## 12 Seaborn

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

```
[14]: <seaborn.axisgrid.PairGrid at 0x7b2469febfa0>
```



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

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

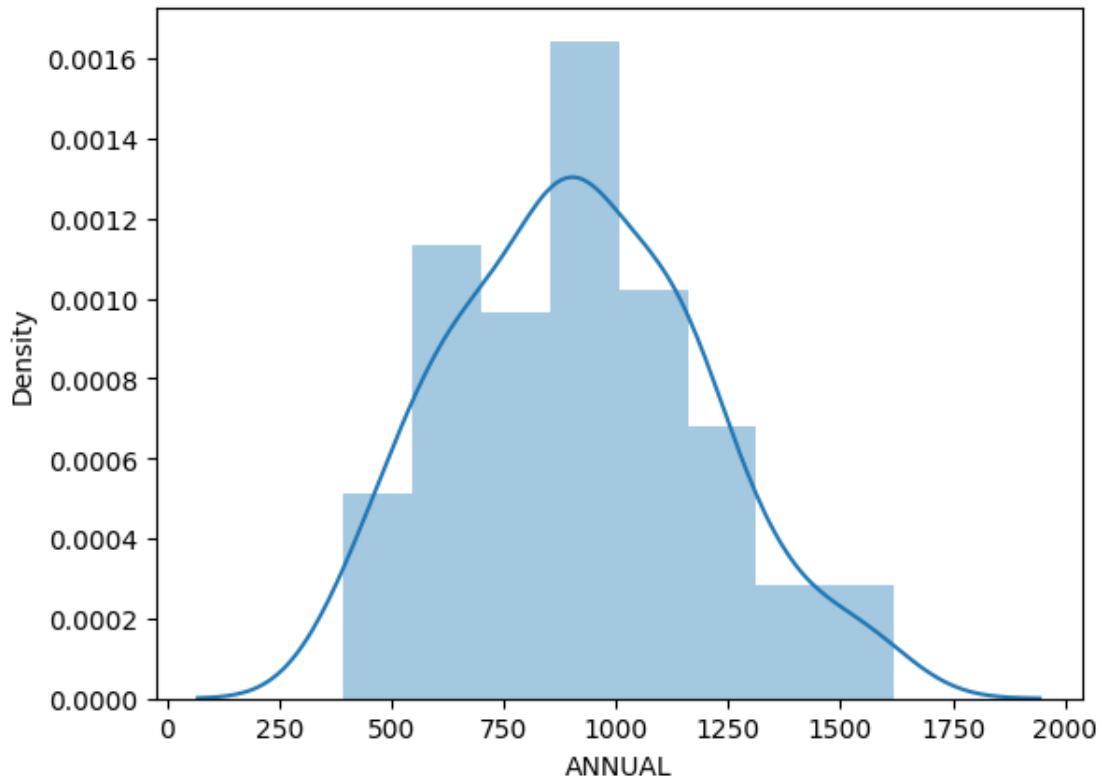
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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: >
```

