

# 20104016

## DEENA

### Importing Libraries

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

### Importing Datasets

```
In [2]: df=pd.read_csv("rainfall_assam _ meghalaya.csv")
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	207	ASSAM & MEGHALAYA	1901	27.1	19.5	30.6	223.0	207.0	524.9	430.6	464.1	291.4	163
1	208	ASSAM & MEGHALAYA	1902	9.3	10.2	105.6	350.0	262.1	620.7	510.8	536.0	441.3	97
2	209	ASSAM & MEGHALAYA	1903	19.9	25.4	103.6	140.6	206.6	607.4	362.7	551.9	306.4	159
3	210	ASSAM & MEGHALAYA	1904	11.1	56.1	51.9	457.1	375.2	385.7	477.6	438.8	245.9	115
4	211	ASSAM & MEGHALAYA	1905	19.9	16.9	137.9	213.0	275.5	521.7	439.1	649.1	276.0	200
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	317	ASSAM & MEGHALAYA	2011	11.1	11.4	109.0	92.1	238.3	316.0	395.8	302.6	221.6	30
111	318	ASSAM & MEGHALAYA	2012	15.2	6.9	28.8	279.1	185.8	729.7	444.3	289.2	411.6	199
112	319	ASSAM & MEGHALAYA	2013	1.1	9.6	44.0	112.8	346.7	286.2	367.8	289.7	229.3	126
113	320	ASSAM & MEGHALAYA	2014	2.0	28.3	29.3	51.5	351.1	426.4	374.4	484.6	420.2	35
114	321	ASSAM & MEGHALAYA	2015	13.4	15.5	37.5	250.9	332.5	558.5	300.1	590.9	279.9	62

115 rows × 20 columns

# Data Cleaning and Data Preprocessing

In [3]: `df = df.dropna()`

In [4]: `df.columns`

Out[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')

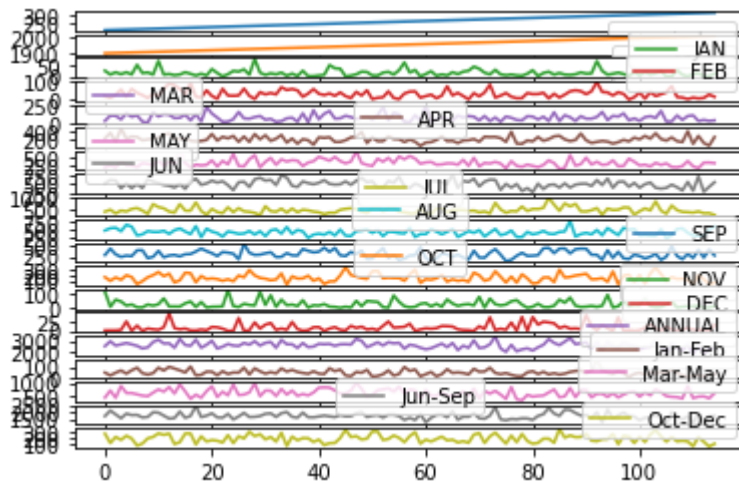
In [5]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 18.9+ KB
```

## Line chart

In [6]: `df.plot.line(subplots=True)`

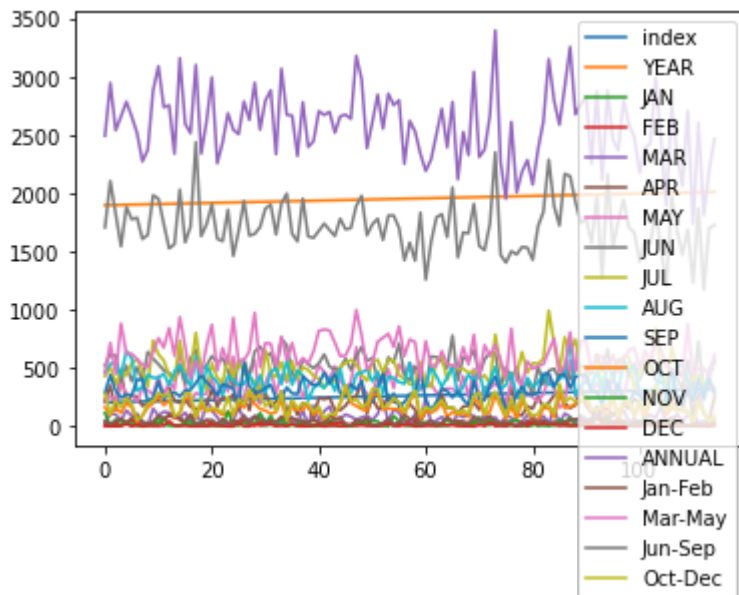
Out[6]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)



## Line chart

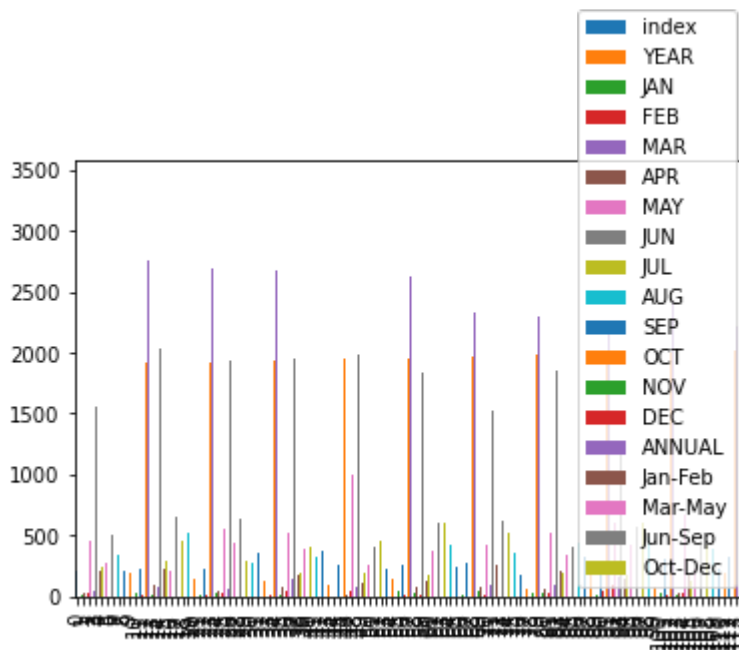
In [7]: `df.plot.line()`

Out[7]: <AxesSubplot:>



## Bar chart

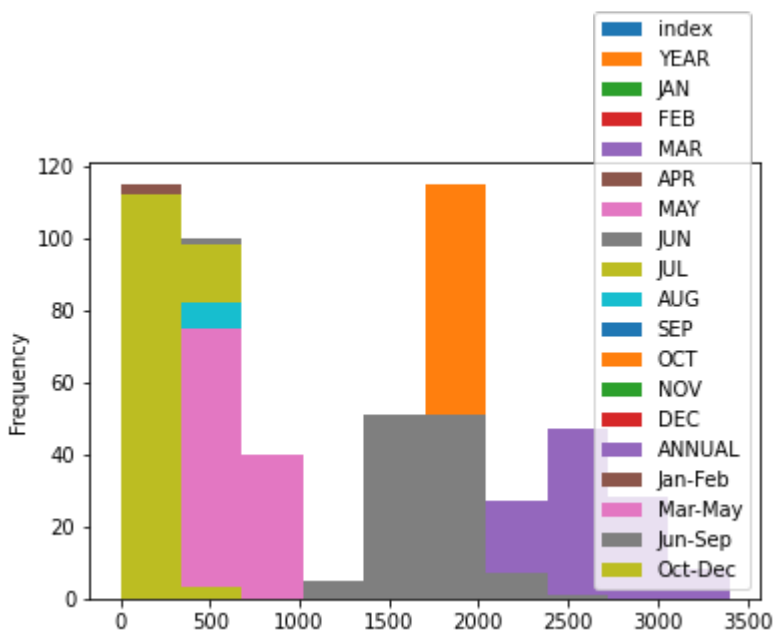
Out[8]: <AxesSubplot:>



# Histogram

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

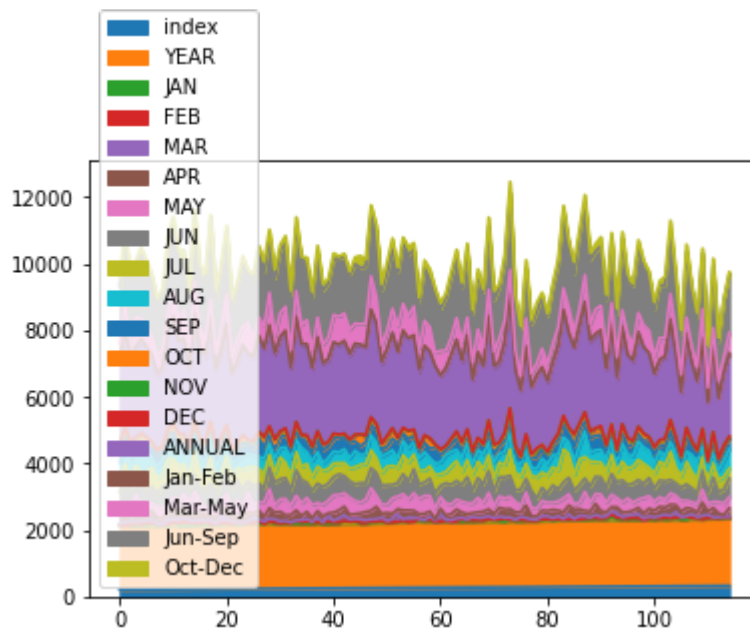
```
Out[9]: <AxesSubplot:ylabel='Frequency'>
```



## Area chart

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

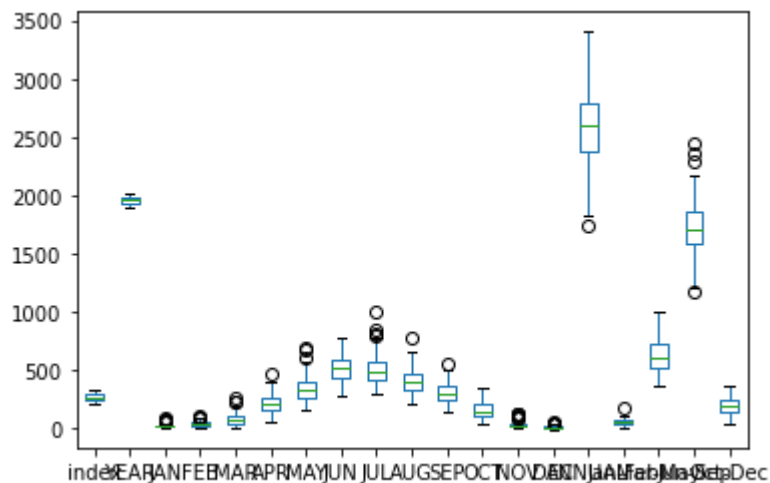
Out[10]: `<AxesSubplot:>`



## Box chart

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

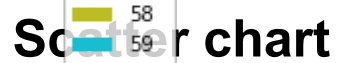
Out[11]: `<AxesSubplot:>`



## Pie chart

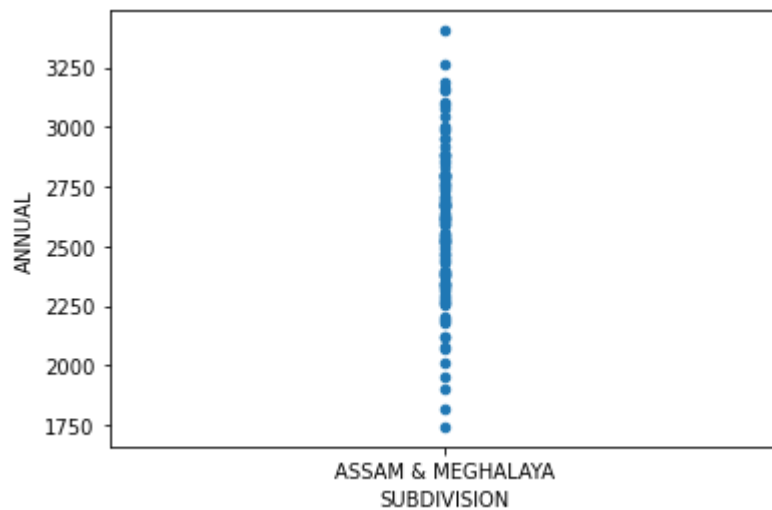
In [12]: `df.plot(figsize=(10, 5), color='r', label='ANNUAL')`

Out[12]: `<AxesSubplot:ylabel='ANNUAL'>`



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

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



In [14]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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
```



In [15]: `df.describe()`

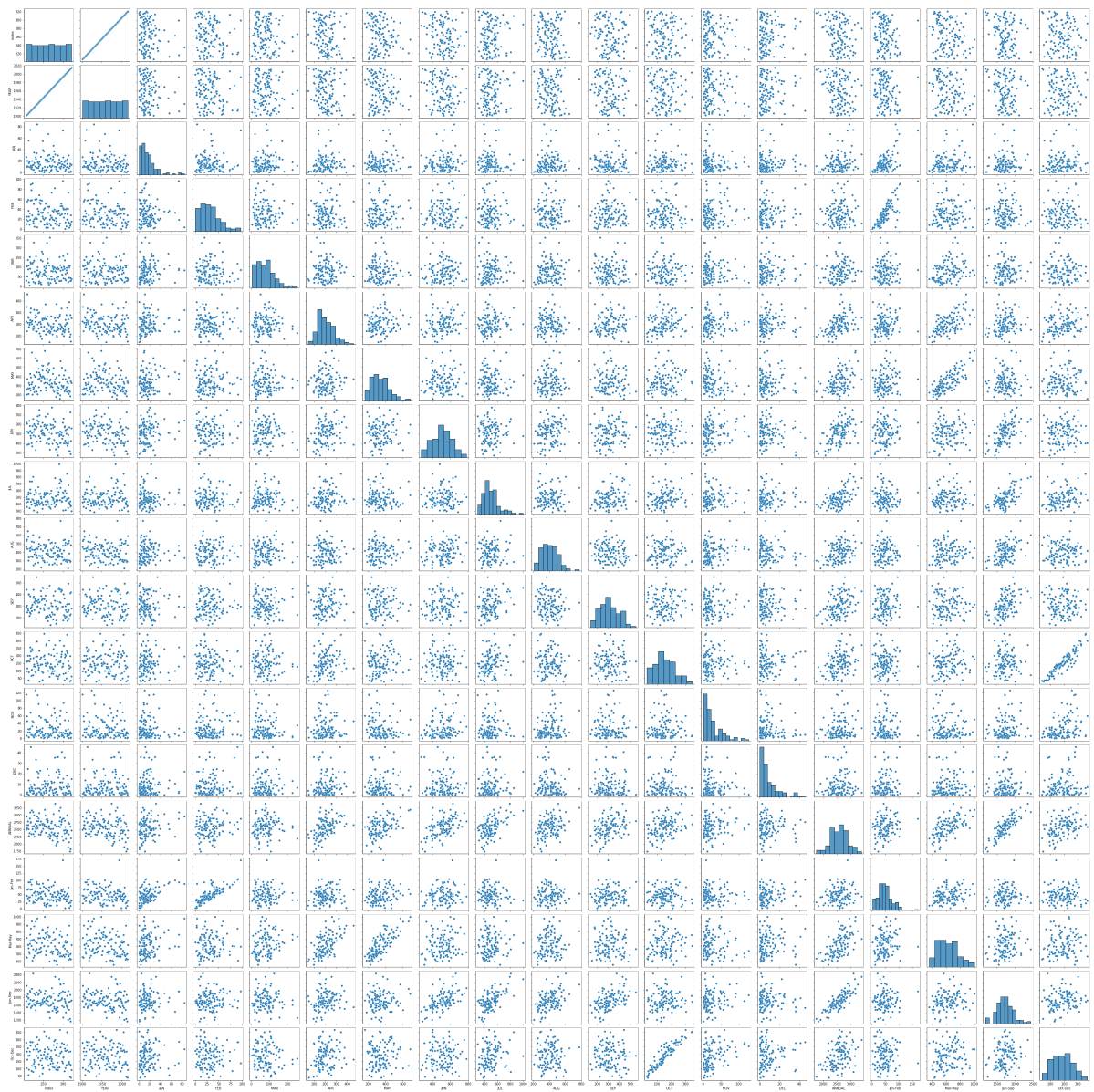
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115
<b>mean</b>	264.000000	1958.000000	16.974783	31.441739	79.026957	203.115652	341.539130	510
<b>std</b>	33.341666	33.341666	15.668601	20.963452	48.679075	74.338367	105.771005	106
<b>min</b>	207.000000	1901.000000	0.100000	0.500000	4.400000	45.900000	159.700000	273
<b>25%</b>	235.500000	1929.500000	7.500000	15.800000	38.700000	151.350000	261.100000	421
<b>50%</b>	264.000000	1958.000000	12.800000	28.300000	75.700000	197.600000	325.700000	513
<b>75%</b>	292.500000	1986.500000	21.600000	41.950000	104.600000	251.500000	397.150000	583
<b>max</b>	321.000000	2015.000000	83.900000	96.900000	253.800000	457.100000	681.200000	780

## EDA AND VISUALIZATION

In [16]: `sns.pairplot(df)`

Out[16]: `<seaborn.axisgrid.PairGrid at 0x267f9dadca0>`

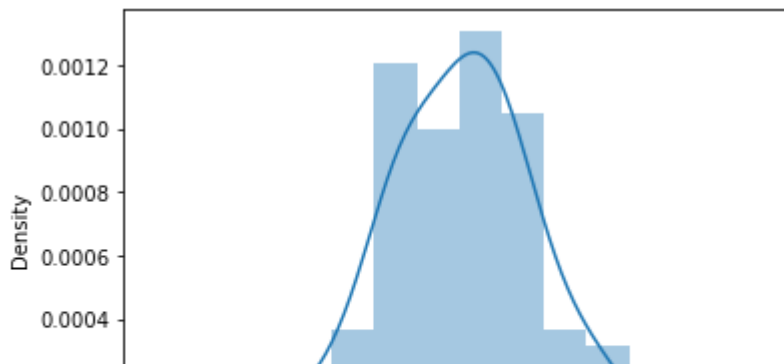


In [17]: `sns.distplot(df['ANNUAL'])`

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

warnings.warn(msg, FutureWarning)

Out[17]: `<AxesSubplot:xlabel='ANNUAL', ylabel='Density'>`



In [18]: `sns.heatmap(df.corr())`

Out[18]: `<AxesSubplot:>`

