

# 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_arunachal pradesh.csv")
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	
0	110	ARUNACHAL PRADESH	1916	48.1	69.8	71.1	316.1	424.6	1124.9	NaN	629.7	333.9	
1	111	ARUNACHAL PRADESH	1917	21.4	164.5	NaN	269.6	107.9	823.8	909.1	628.4	411.5	1
2	112	ARUNACHAL PRADESH	1918	10.4	11.0	191.2	144.6	861.1	1609.9	1303.0	692.6	515.8	1
3	113	ARUNACHAL PRADESH	1919	34.5	67.8	28.5	256.9	420.6	973.6	999.0	286.7	628.7	9
4	114	ARUNACHAL PRADESH	1920	14.0	196.3	605.6	364.7	173.6	840.6	535.4	896.5	376.7	1
...	...	...	...	...	...	...	...	...	...	...	...	...	
92	202	ARUNACHAL PRADESH	2011	40.0	51.3	174.5	240.8	219.6	288.4	531.4	277.6	286.7	
93	203	ARUNACHAL PRADESH	2012	57.8	35.8	134.2	403.4	187.4	645.8	638.9	316.0	724.9	2
94	204	ARUNACHAL PRADESH	2013	18.5	40.5	115.1	175.1	335.8	290.0	329.6	230.2	316.1	1
95	205	ARUNACHAL PRADESH	2014	19.0	101.9	80.3	86.7	299.0	415.8	392.4	599.6	343.0	
96	206	ARUNACHAL PRADESH	2015	30.8	47.5	97.5	287.1	238.9	637.9	329.3	595.5	374.2	

97 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')
```

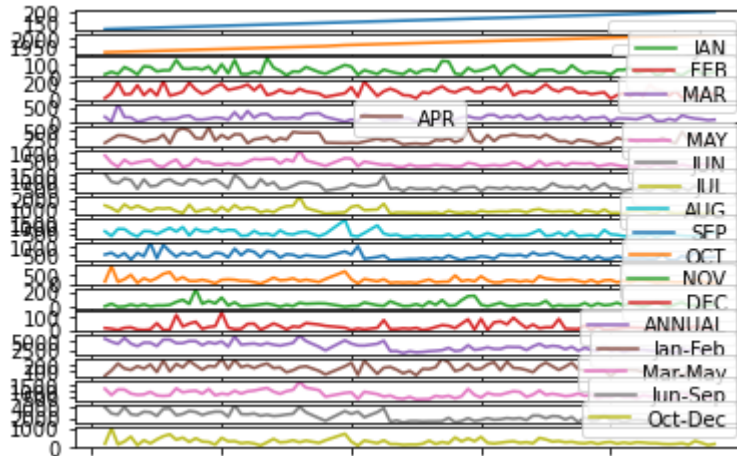
```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 91 entries, 2 to 96  
Data columns (total 20 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   index                 91 non-null    int64  
1   SUBDIVISION           91 non-null    object  
2   YEAR                  91 non-null    int64  
3   JAN                   91 non-null    float64  
4   FEB                   91 non-null    float64  
5   MAR                   91 non-null    float64  
6   APR                   91 non-null    float64  
7   MAY                   91 non-null    float64  
8   JUN                   91 non-null    float64  
9   JUL                   91 non-null    float64  
10  AUG                   91 non-null    float64  
11  SEP                   91 non-null    float64  
12  OCT                   91 non-null    float64  
13  NOV                   91 non-null    float64  
14  DEC                   91 non-null    float64  
15  ANNUAL                91 non-null    float64  
16  Jan-Feb               91 non-null    float64  
17  Mar-May               91 non-null    float64  
18  Jun-Sep               91 non-null    float64  
19  Oct-Dec               91 non-null    float64  
dtypes: float64(17), int64(2), object(1)  
memory usage: 14.9+ KB
```

## Line chart

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

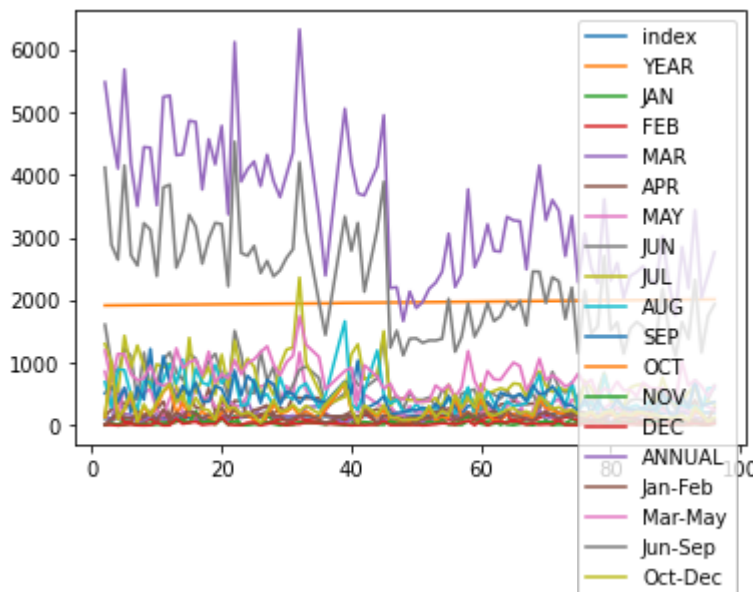
Out[6]: array([<AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>], dtype=object)



## Line chart

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

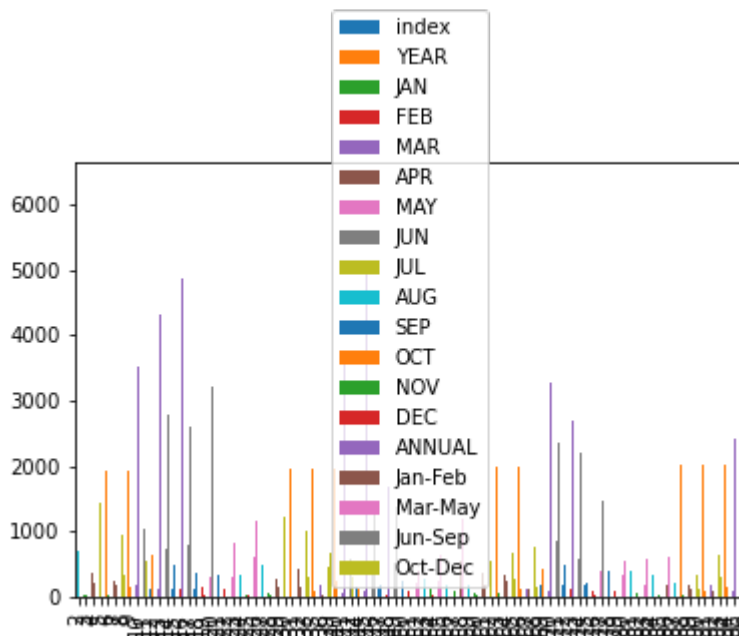
Out[7]: <AxesSubplot:~>



## Bar chart

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

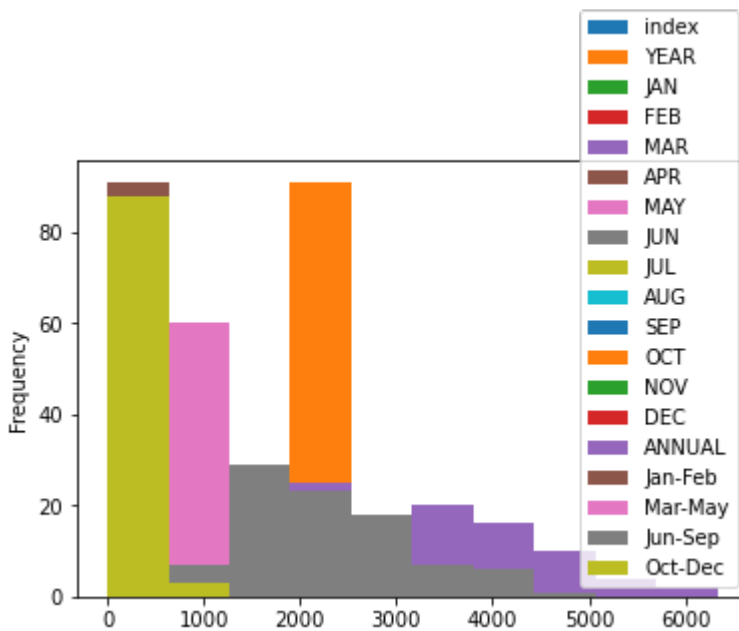
```
Out[8]: <AxesSubplot:>
```



## Histogram

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

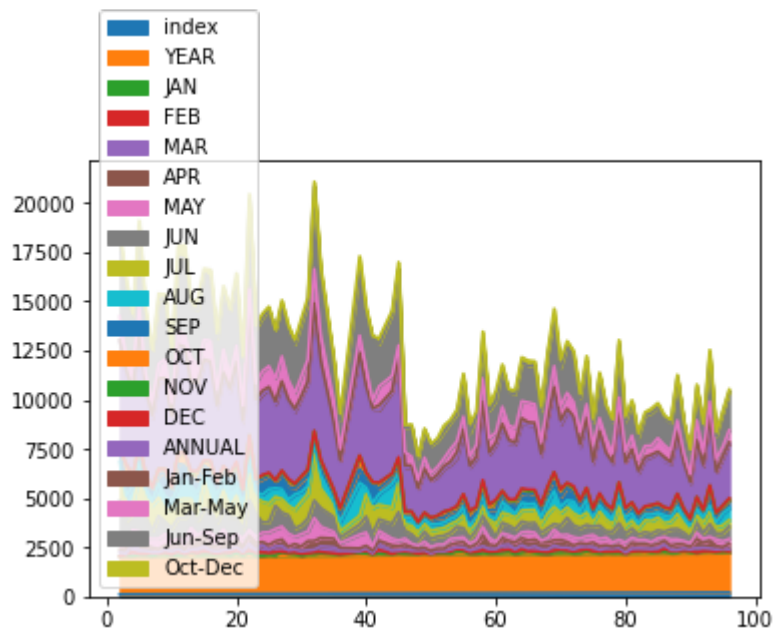
```
Out[9]: <AxesSubplot:ylabel='Frequency'>
```



## Area chart

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

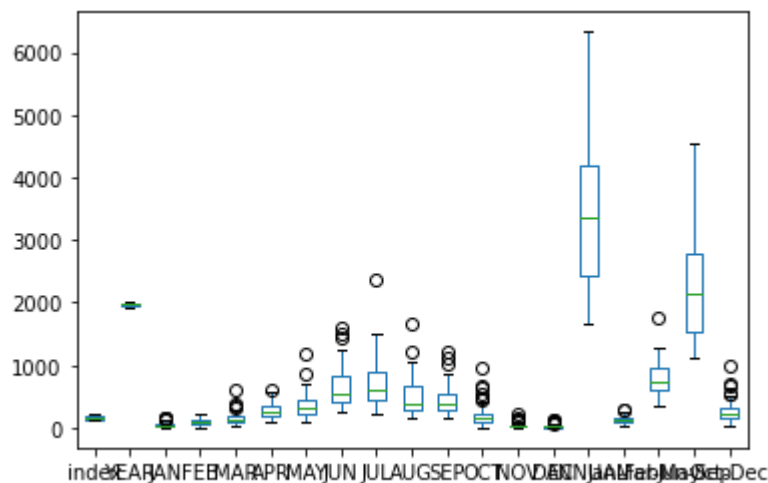
Out[10]: `<AxesSubplot:>`



## Box chart

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

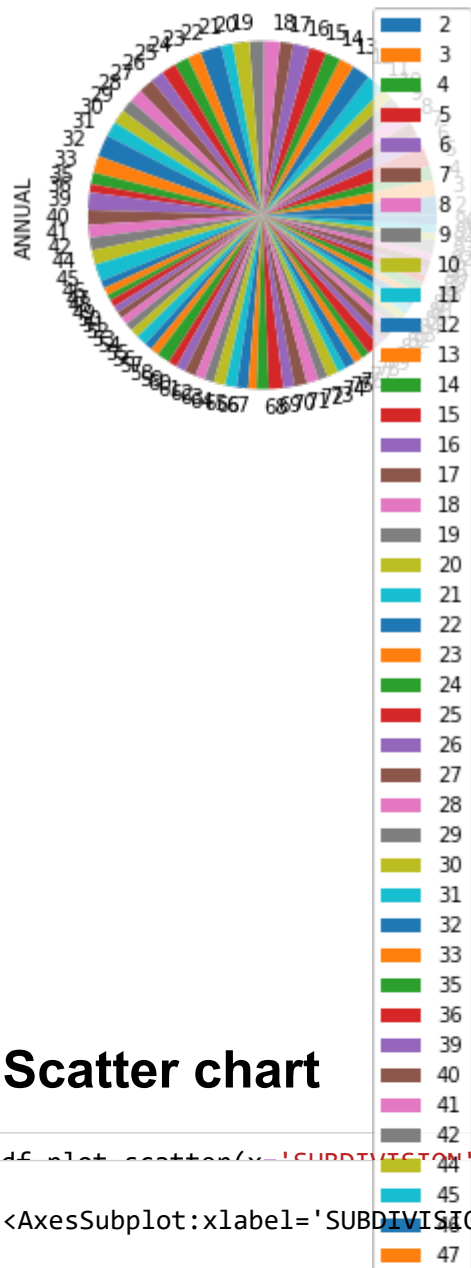
Out[11]: `<AxesSubplot:>`



## Pie chart

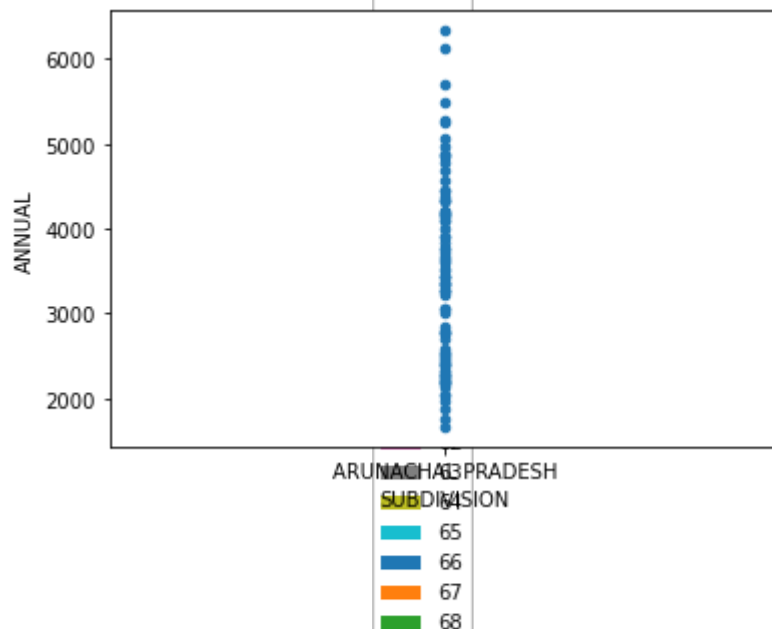
In [12]: `df.plot(figsize=(10, 5), color='r', label='ANNUAL')`

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



## Scatter chart

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: 91 entries, 2 to 96
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype  
---  --
 0   index           91 non-null    int64  
 1   SUBDIVISION     91 non-null    object  
 2   YEAR            91 non-null    int64  
 3   JAN             91 non-null    float64 
 4   FEB             91 non-null    float64 
 5   MAR             91 non-null    float64 
 6   APR             91 non-null    float64 
 7   MAY             91 non-null    float64 
 8   JUN             91 non-null    float64 
 9   JUL             91 non-null    float64 
10  AUG             91 non-null    float64 
11  SEP             91 non-null    float64 
12  OCT             91 non-null    float64 
13  NOV             91 non-null    float64 
14  DEC             91 non-null    float64
```

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

Out[15]:

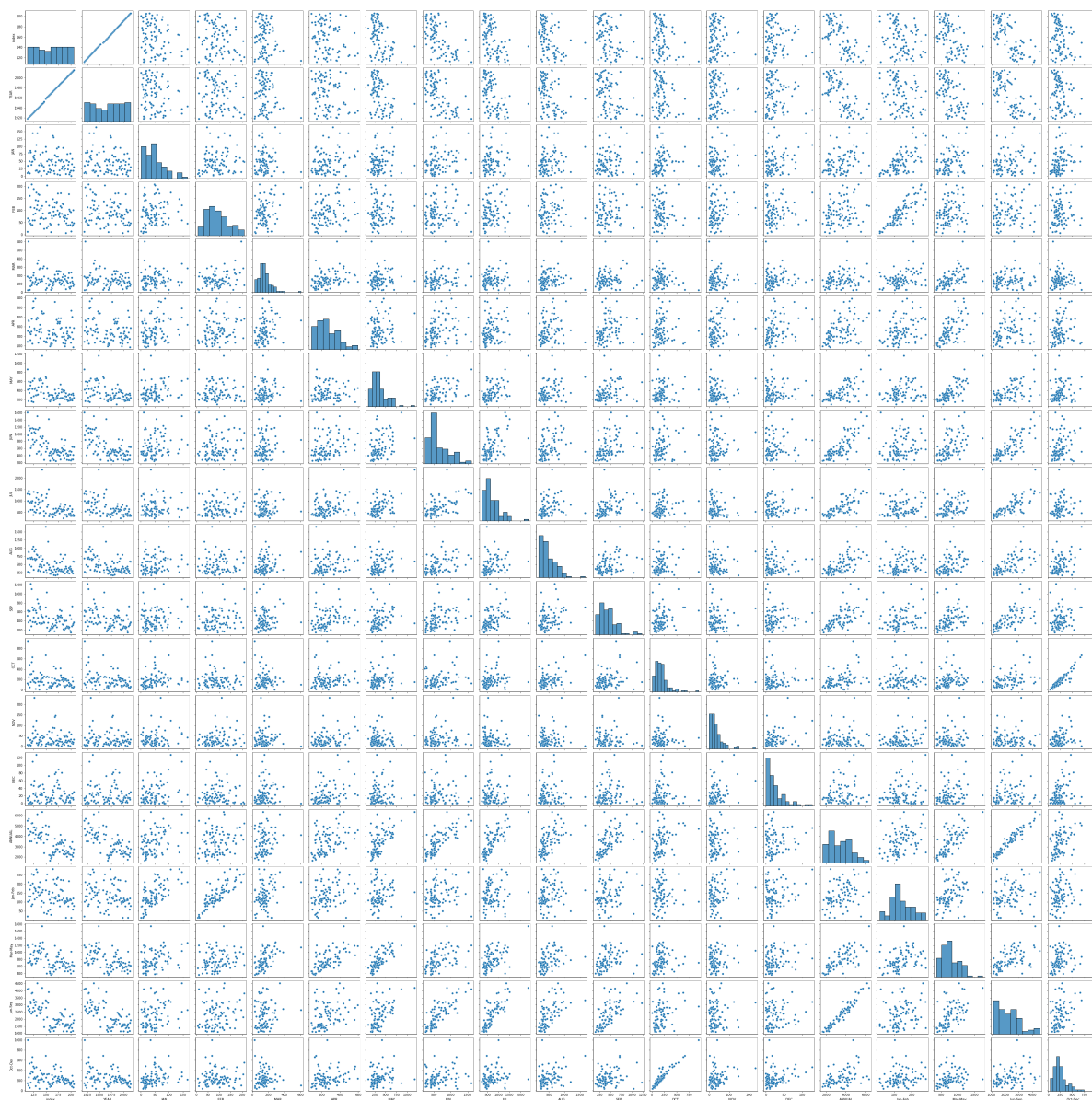
	index	YEAR	JAN	FEB	MAR	APR	MAY
count	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000
mean	159.483516	1967.362637	47.680220	90.396703	154.143956	262.297802	358.289011
std	28.065939	29.324437	35.045676	47.178011	86.284987	116.737705	178.900132
min	112.000000	1918.000000	0.600000	6.100000	28.500000	86.700000	101.800000
25%	134.500000	1940.500000	19.100000	55.250000	102.700000	177.500000	232.950000
50%	161.000000	1970.000000	40.000000	83.200000	139.900000	240.800000	306.900000
75%	183.500000	1992.500000	64.900000	118.900000	182.450000	341.200000	433.600000
max	206.000000	2015.000000	164.500000	208.500000	605.600000	595.100000	1168.600000

## EDA AND VISUALIZATION



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

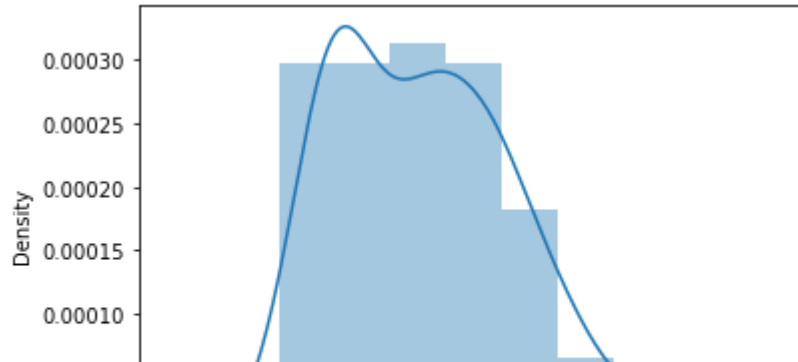
Out[16]: `<seaborn.axisgrid.PairGrid at 0x207760bee80>`



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:>`

