

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_coastal karnataka.csv")
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	
0	3542	COASTAL KARNATAKA	1901	1.8	0.6	10.7	52.4	81.6	960.9	991.2	606.4	108.0	1
1	3543	COASTAL KARNATAKA	1902	3.2	0.3	4.9	10.2	54.6	698.4	1401.6	454.2	708.4	1
2	3544	COASTAL KARNATAKA	1903	0.7	0.0	0.0	4.1	202.8	536.5	1405.5	593.8	304.4	1
3	3545	COASTAL KARNATAKA	1904	2.4	0.0	4.8	23.7	93.2	1108.2	1070.0	465.6	245.3	1
4	3546	COASTAL KARNATAKA	1905	0.0	0.2	0.0	6.4	83.1	767.3	777.3	586.9	172.9	2
...
110	3652	COASTAL KARNATAKA	2011	4.8	3.8	8.7	66.1	49.3	1018.4	1080.5	861.3	545.2	1
111	3653	COASTAL KARNATAKA	2012	NaN	11.4	5.1	77.0	22.9	650.9	754.6	1027.6	382.0	1
112	3654	COASTAL KARNATAKA	2013	2.4	19.6	19.0	28.5	100.4	1153.0	1515.3	680.2	379.1	2
113	3655	COASTAL KARNATAKA	2014	0.0	0.3	1.9	40.5	181.9	507.0	1155.4	1121.0	379.3	2
114	3656	COASTAL KARNATAKA	2015	1.4	1.0	32.3	72.2	150.3	735.3	930.9	575.2	260.3	2

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: 114 entries, 0 to 114  
Data columns (total 20 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   index                 114 non-null   int64  
1   SUBDIVISION           114 non-null   object  
2   YEAR                  114 non-null   int64  
3   JAN                   114 non-null   float64  
4   FEB                   114 non-null   float64  
5   MAR                   114 non-null   float64  
6   APR                   114 non-null   float64  
7   MAY                   114 non-null   float64  
8   JUN                   114 non-null   float64  
9   JUL                   114 non-null   float64  
10  AUG                   114 non-null   float64  
11  SEP                   114 non-null   float64  
12  OCT                   114 non-null   float64  
13  NOV                   114 non-null   float64  
14  DEC                   114 non-null   float64  
15  ANNUAL                114 non-null   float64  
16  Jan-Feb               114 non-null   float64  
17  Mar-May               114 non-null   float64  
18  Jun-Sep               114 non-null   float64  
19  Oct-Dec               114 non-null   float64  
dtypes: float64(17), int64(2), object(1)  
memory usage: 18.7+ KB
```

Line chart

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

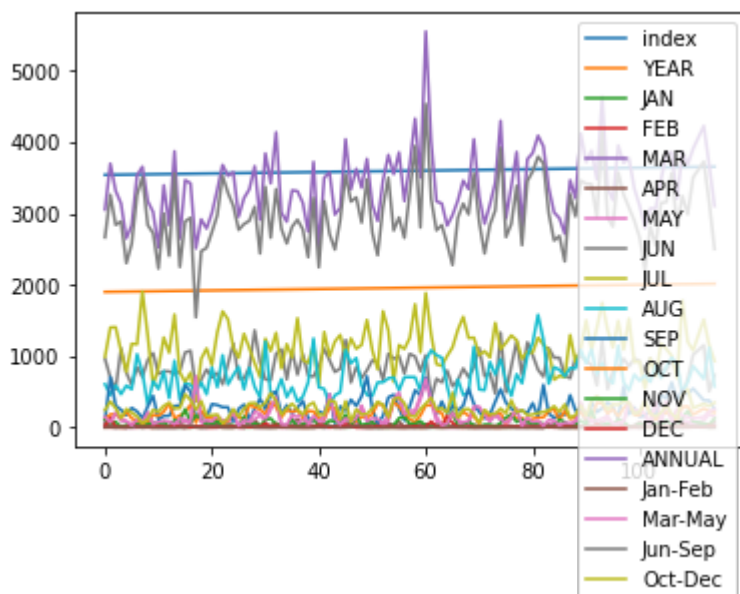
Out[6]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)



Line chart

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

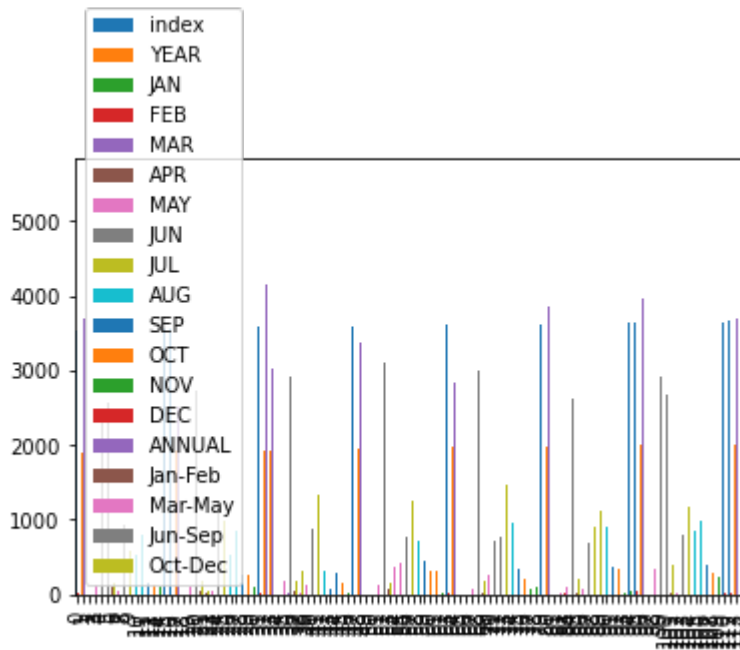
Out[7]: <AxesSubplot:>



Bar chart

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

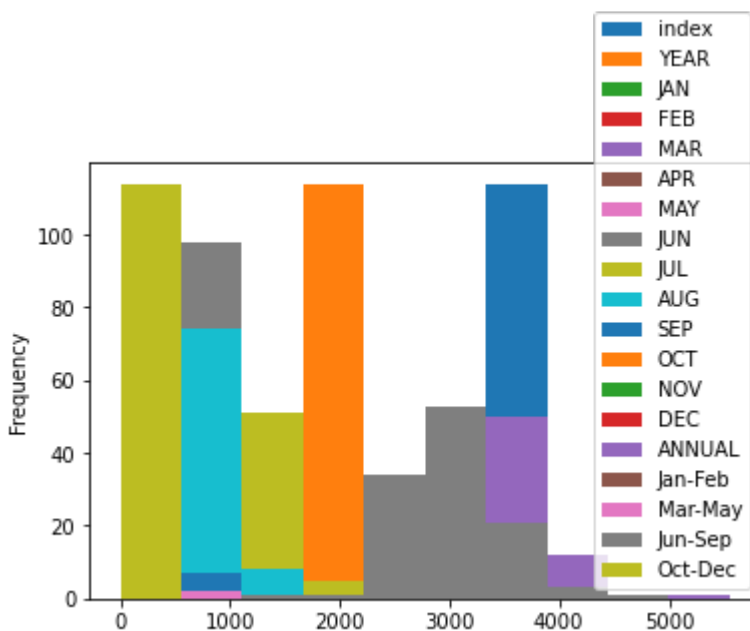
Out[8]: `<AxesSubplot:>`



Histogram

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

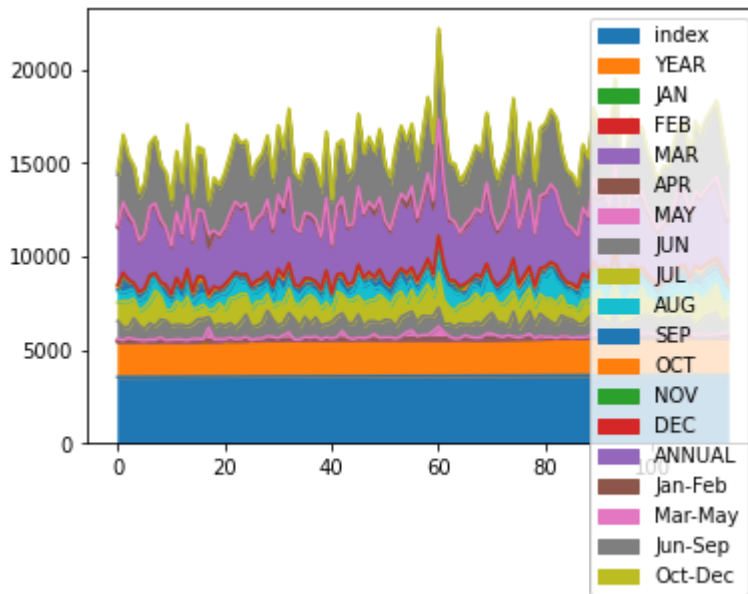
Out[9]: `<AxesSubplot:ylabel='Frequency'>`



Area chart

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

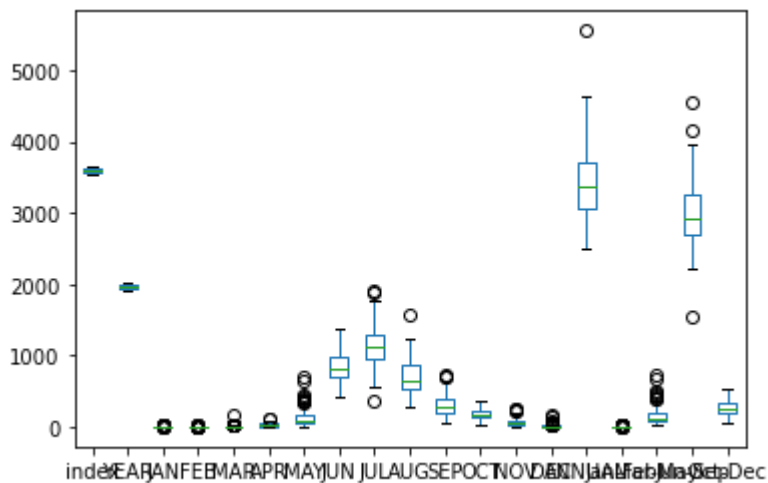
Out[10]: `<AxesSubplot:>`



Box chart

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

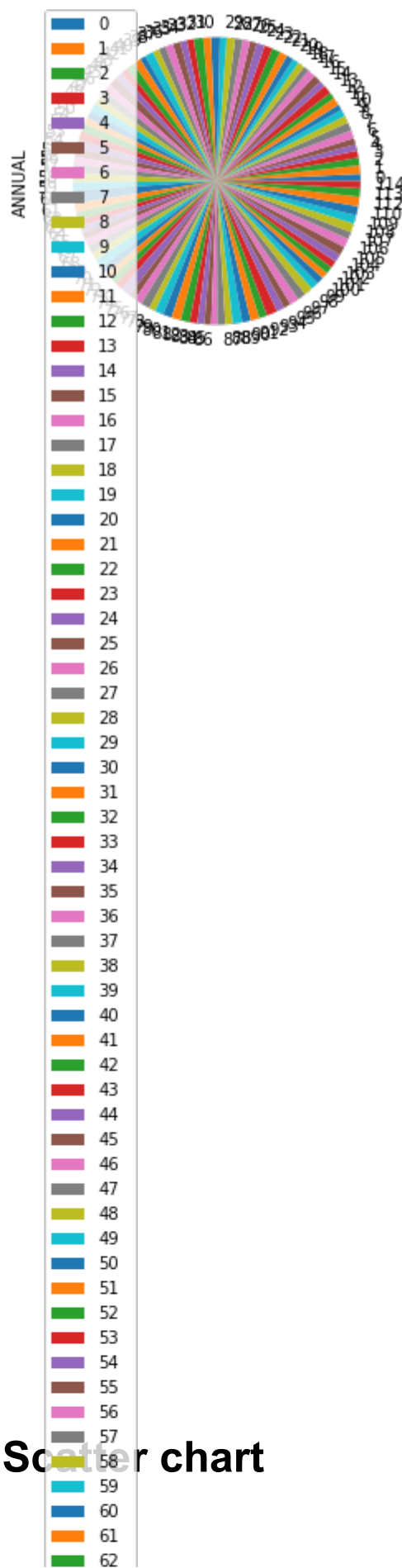
Out[11]: `<AxesSubplot:>`



Pie chart

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

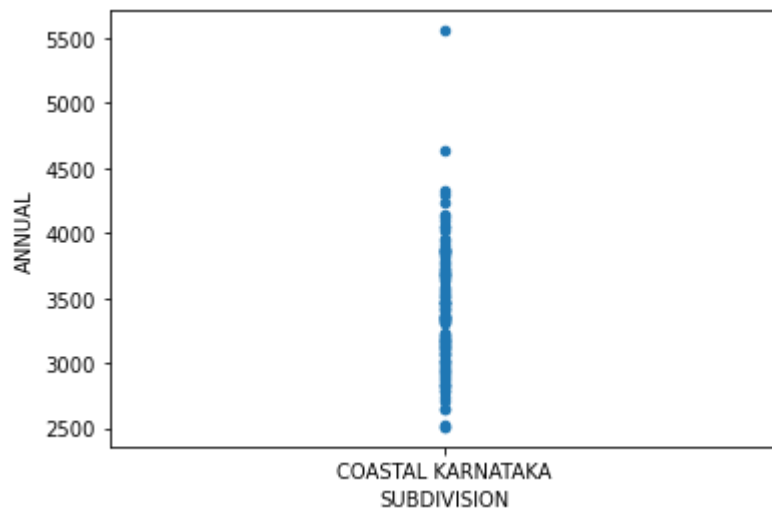
Out[12]: `<AxesSubplot:ylabel='ANNUAL'>`



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: 114 entries, 0 to 114
Data columns (total 20 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   index           114 non-null   int64
1   SUBDIVISION     114 non-null   object
2   YEAR            114 non-null   int64
3   JAN             114 non-null   float64
4   FEB             114 non-null   float64
5   MAR             114 non-null   float64
6   APR             114 non-null   float64
7   MAY             114 non-null   float64
8   JUN             114 non-null   float64
9   JUL             114 non-null   float64
10  AUG             114 non-null   float64
11  SEP             114 non-null   float64
12  OCT             114 non-null   float64
13  NOV             114 non-null   float64
14  DEC             114 non-null   float64
```


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

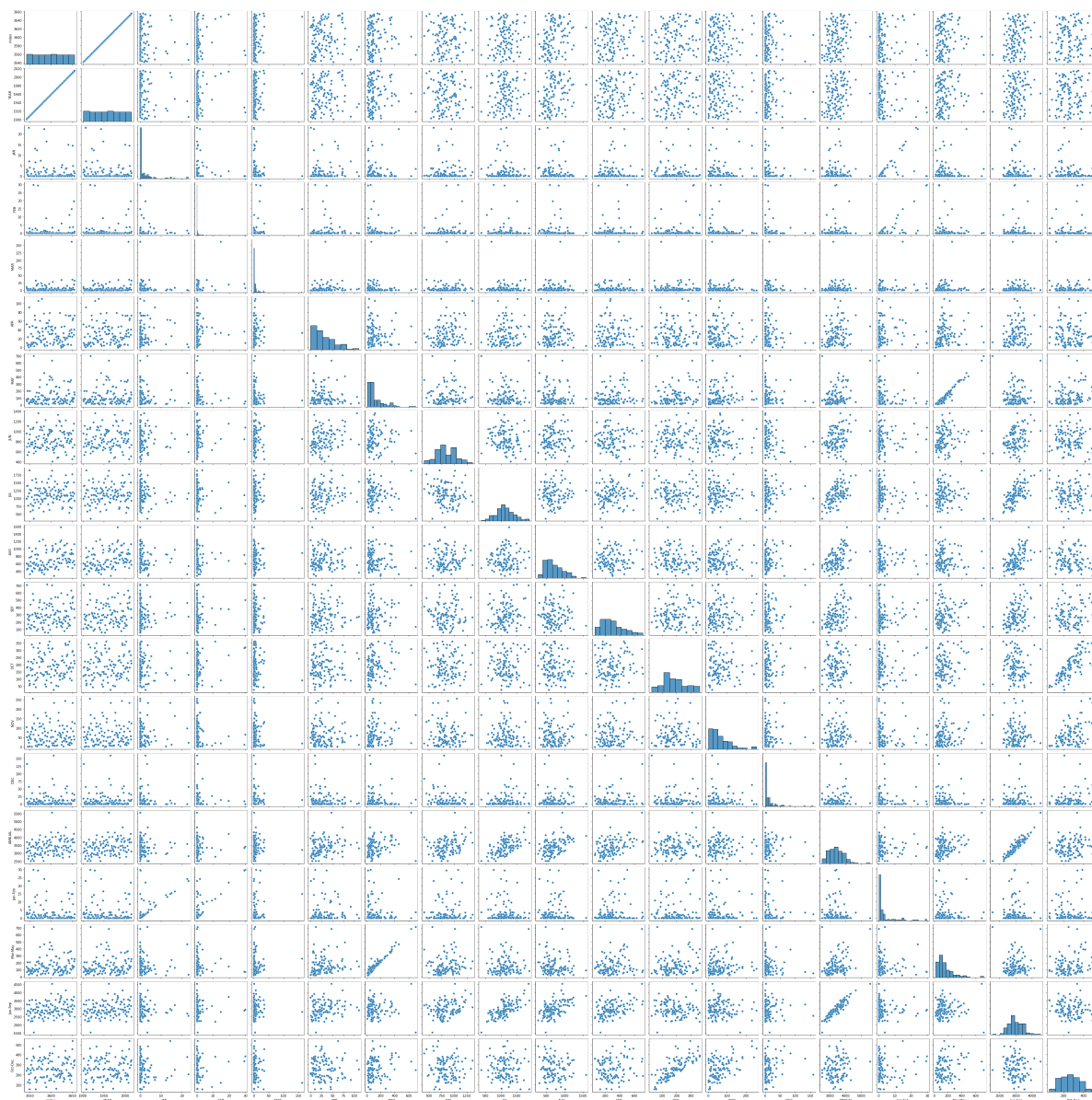
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	1
mean	3598.526316	1957.526316	1.937719	1.431579	6.368421	30.512281	123.664035	8
std	33.097927	33.097927	4.218363	4.657477	16.646083	23.845658	125.592810	1
min	3542.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	8.400000	4
25%	3570.250000	1929.250000	0.000000	0.000000	0.200000	11.325000	44.325000	7
50%	3598.500000	1957.500000	0.100000	0.000000	1.450000	24.700000	80.500000	8
75%	3626.750000	1985.750000	1.975000	0.500000	6.150000	44.800000	162.225000	9
max	3656.000000	2015.000000	23.000000	29.800000	161.400000	110.100000	699.500000	13

EDA AND VISUALIZATION

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

Out[16]: `<seaborn.axisgrid.PairGrid at 0x20ef3dbdaf0>`

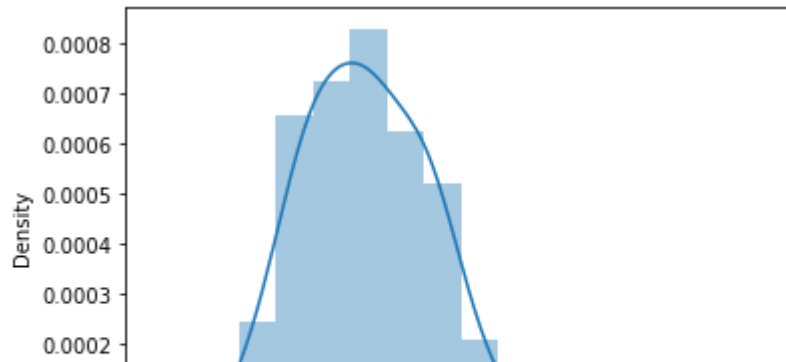


In [17]: `sns.distplot(df['ANNUAL'])`

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:>`

