

Type *Markdown* and LaTeX: α^2

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(r"C:\Users\user\Downloads\drive-download-20230804T043023Z-001\rainfall_andaman _ nicobar islands.csv")
df
```

Out[2]:

| | index | SUBDIVISION | YEAR | JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC | ANNUAL | Jan-Feb | Mar-May | Jun-Sep |
|-----|-------|---------------------------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|---------|---------|---------|
| 0 | 0 | ANDAMAN & NICOBAR ISLANDS | 1901 | 49.2 | 87.1 | 29.2 | 2.3 | 528.8 | 517.5 | 365.1 | 481.1 | 332.6 | 388.5 | 558.2 | 33.6 | 3373.2 | 136.3 | 560.3 | 169 |
| 1 | 1 | ANDAMAN & NICOBAR ISLANDS | 1902 | 0.0 | 159.8 | 12.2 | 0.0 | 446.1 | 537.1 | 228.9 | 753.7 | 666.2 | 197.2 | 359.0 | 160.5 | 3520.7 | 159.8 | 458.3 | 218 |
| 2 | 2 | ANDAMAN & NICOBAR ISLANDS | 1903 | 12.7 | 144.0 | 0.0 | 1.0 | 235.1 | 479.9 | 728.4 | 326.7 | 339.0 | 181.2 | 284.4 | 225.0 | 2957.4 | 156.7 | 236.1 | 187 |
| 3 | 3 | ANDAMAN & NICOBAR ISLANDS | 1904 | 9.4 | 14.7 | 0.0 | 202.4 | 304.5 | 495.1 | 502.0 | 160.1 | 820.4 | 222.2 | 308.7 | 40.1 | 3079.6 | 24.1 | 506.9 | 197 |
| 4 | 4 | ANDAMAN & NICOBAR ISLANDS | 1905 | 1.3 | 0.0 | 3.3 | 26.9 | 279.5 | 628.7 | 368.7 | 330.5 | 297.0 | 260.7 | 25.4 | 344.7 | 2566.7 | 1.3 | 309.7 | 162 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 105 | 105 | ANDAMAN & NICOBAR ISLANDS | 2011 | 265.9 | 84.8 | 272.8 | 111.4 | 326.5 | 383.2 | 583.2 | 441.5 | 757.1 | 212.3 | 150.8 | 238.5 | 3828.0 | 350.7 | 710.7 | 216 |
| 106 | 106 | ANDAMAN & NICOBAR ISLANDS | 2012 | 119.9 | 45.6 | 30.9 | 55.8 | 533.9 | 458.2 | 317.3 | 369.6 | 868.9 | 209.7 | 300.5 | 187.3 | 3497.6 | 165.6 | 620.5 | 201 |
| 107 | 107 | ANDAMAN & NICOBAR ISLANDS | 2013 | 67.1 | 37.6 | 43.0 | 46.3 | 509.3 | 777.0 | 564.8 | 336.7 | 473.6 | 455.8 | 354.2 | 92.3 | 3757.8 | 104.7 | 598.6 | 215 |
| 108 | 108 | ANDAMAN & NICOBAR ISLANDS | 2014 | 41.9 | 8.6 | 0.0 | 11.1 | 238.0 | 416.6 | 467.6 | 321.6 | 412.9 | 402.6 | 201.2 | 100.4 | 2622.4 | 50.5 | 249.1 | 161 |
| 109 | 109 | ANDAMAN & NICOBAR ISLANDS | 2015 | 126.8 | 7.6 | 3.1 | 138.2 | 331.9 | 346.4 | 328.9 | 480.0 | 523.3 | 252.1 | 236.3 | 129.9 | 2904.6 | 134.4 | 473.2 | 167 |

110 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')
```

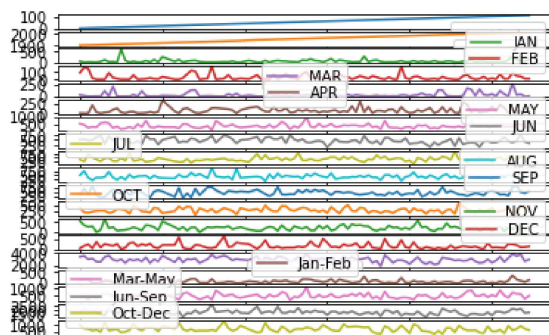
In [5]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 104 entries, 0 to 109
Data columns (total 20 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    index           104 non-null    int64
1    SUBDIVISION     104 non-null    object
2    YEAR            104 non-null    int64
3    JAN             104 non-null    float64
4    FEB             104 non-null    float64
5    MAR             104 non-null    float64
6    APR             104 non-null    float64
7    MAY             104 non-null    float64
8    JUN             104 non-null    float64
9    JUL             104 non-null    float64
10   AUG             104 non-null    float64
11   SEP             104 non-null    float64
12   OCT             104 non-null    float64
13   NOV             104 non-null    float64
14   DEC             104 non-null    float64
15   ANNUAL          104 non-null    float64
16   Jan-Feb         104 non-null    float64
17   Mar-May         104 non-null    float64
18   Jun-Sep         104 non-null    float64
19   Oct-Dec         104 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 17.1+ KB
```

Line chart

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

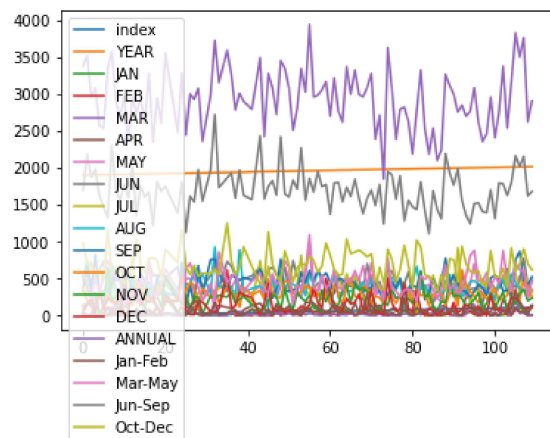
Out[6]: array([<AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>, <AxesSubplot:~>], dtype=object)



Line chart

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

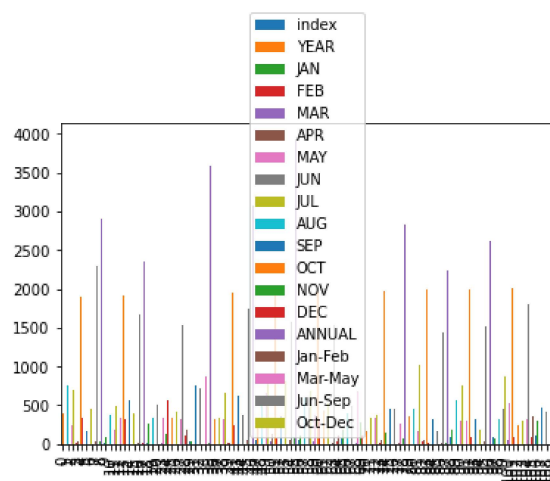
```
Out[7]: <AxesSubplot:>
```



Bar chart

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

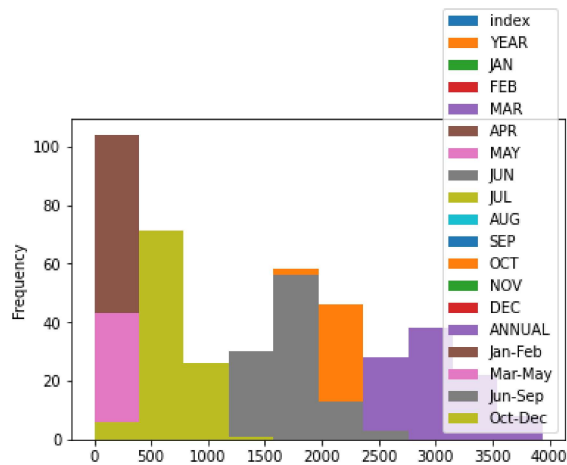
```
Out[8]: <AxesSubplot:>
```



Histogram

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

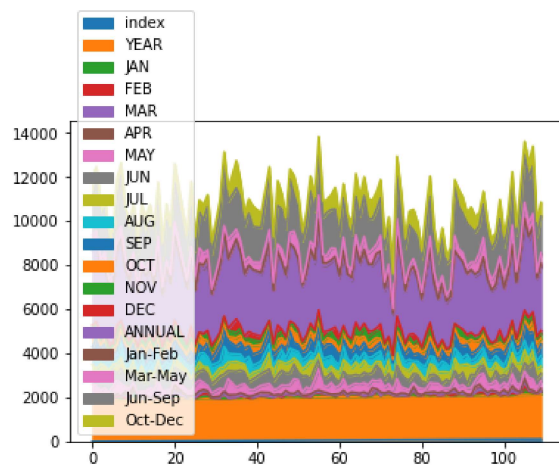
```
Out[9]: <AxesSubplot:ylabel='Frequency'>
```



Area chart

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

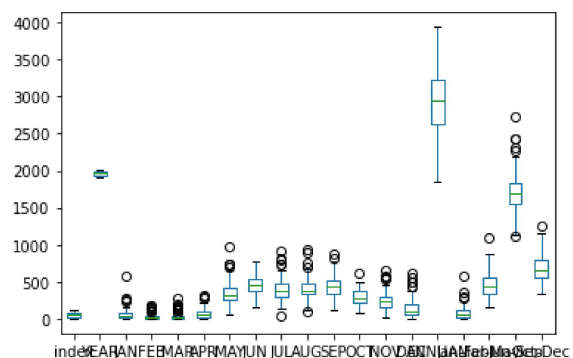
```
Out[10]: <AxesSubplot:>
```



Box chart

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

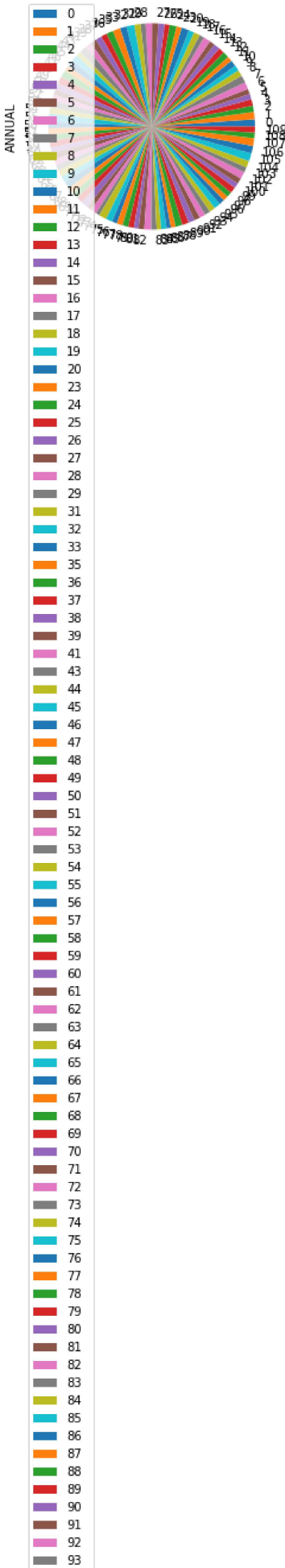
```
Out[11]: <AxesSubplot:>
```

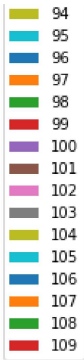


Pie chart

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

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

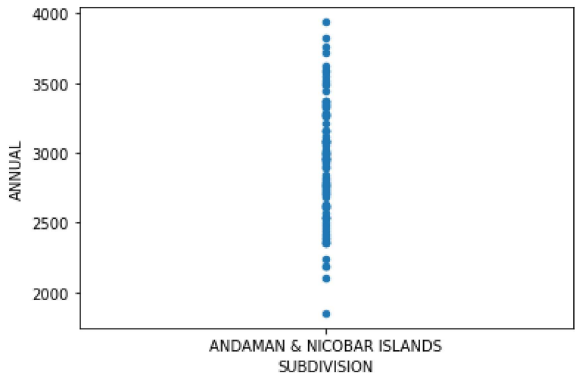





Scatter chart

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

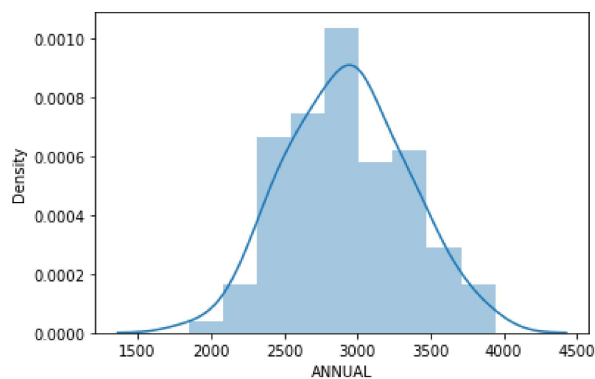
Out[13]: <AxesSubplot:xlabel='SUBDIVISION', ylabel='ANNUAL'>



Seaborn

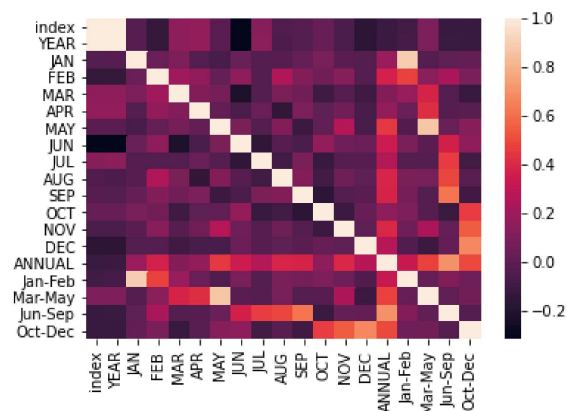

```
In [15]: sns.distplot(df['ANNUAL'])
# distplot (an axes-level function for histograms) or histplot (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
```

```
Out[15]: <AxesSubplot:xlabel='ANNUAL', ylabel='Density'>
```



```
In [16]: sns.heatmap(df.corr())
```

```
Out[16]: <AxesSubplot:>
```



```
In [ ]:
```