

Importing Libraries

In [1]:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

Importing Datasets

```
In [2]: 1 df=pd.read_csv('Andaman_Nicobar.csv')
2 df
```

Out[2]:

		index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May
0	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
1	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
2	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
3	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
4	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
...	
105	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
106	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
107	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
108	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
109	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

110 rows × 20 columns

Data Cleaning and Data Preprocessing

In [3]:

```

1 df=df.dropna()
2 df

```

Out[3]:

		index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May
0	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
1	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
2	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
3	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
4	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
...	
105	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
106	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
107	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
108	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
109	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

104 rows × 20 columns

```
In [4]: 1 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')
```

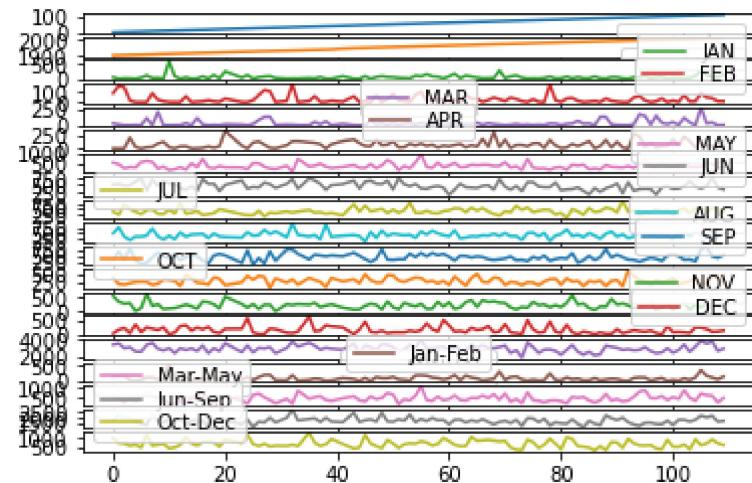
```
In [5]: 1 df.info()
```

```
<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

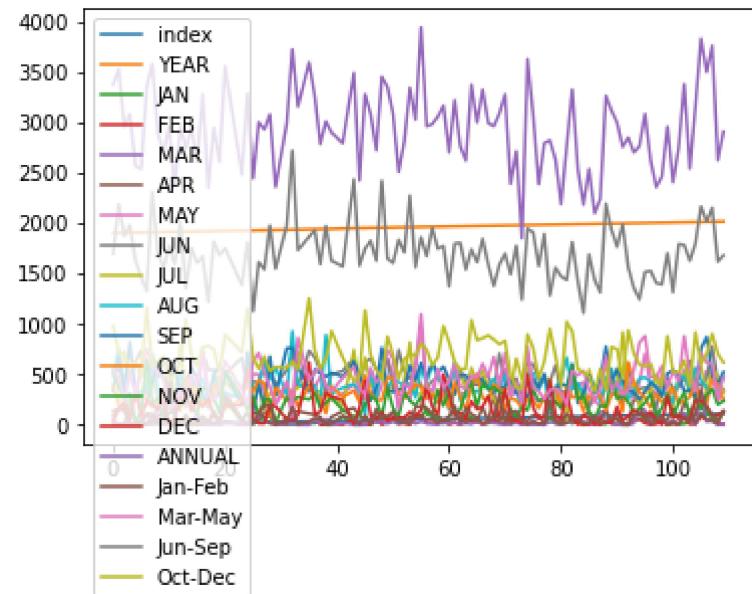
```
In [6]: 1 df.plot.line(subplots=True)
```

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



In [7]: 1 df.plot.line()

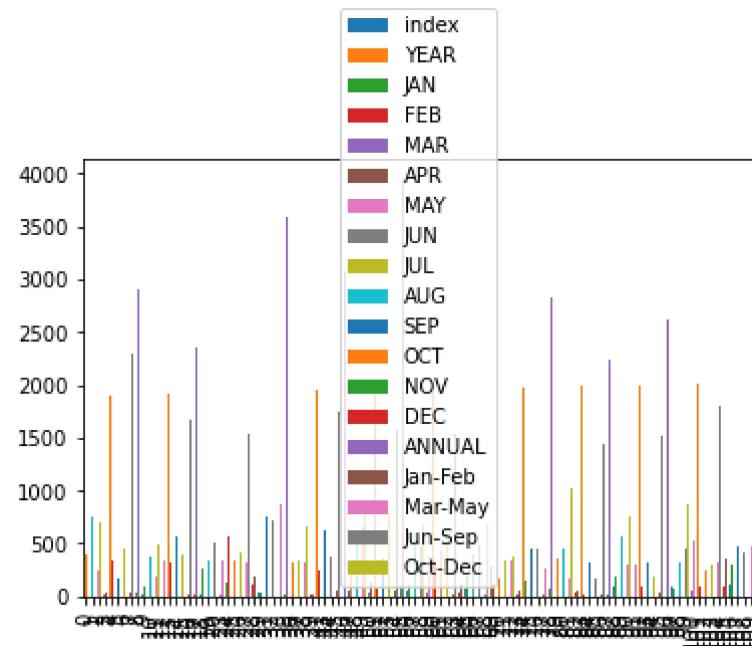
Out[7]: <AxesSubplot:>



Bar Chart

In [8]: 1 df.plot.bar()

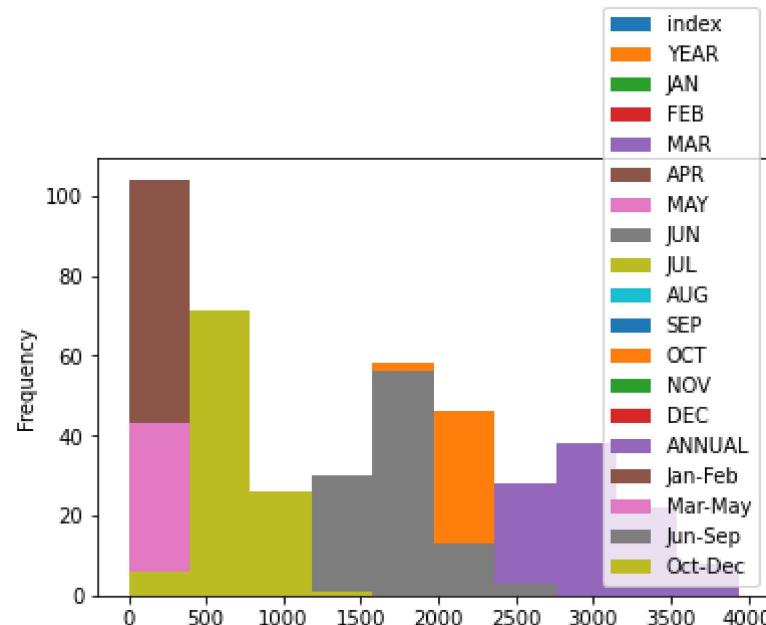
Out[8]: <AxesSubplot:>



Histogram

In [9]: 1 df.plot.hist()

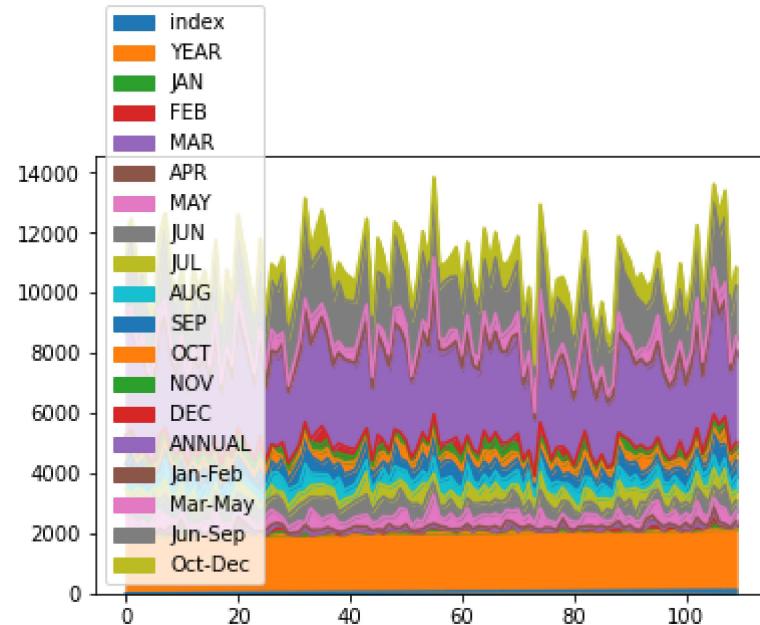
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area Chart

In [10]: 1 df.plot.area()

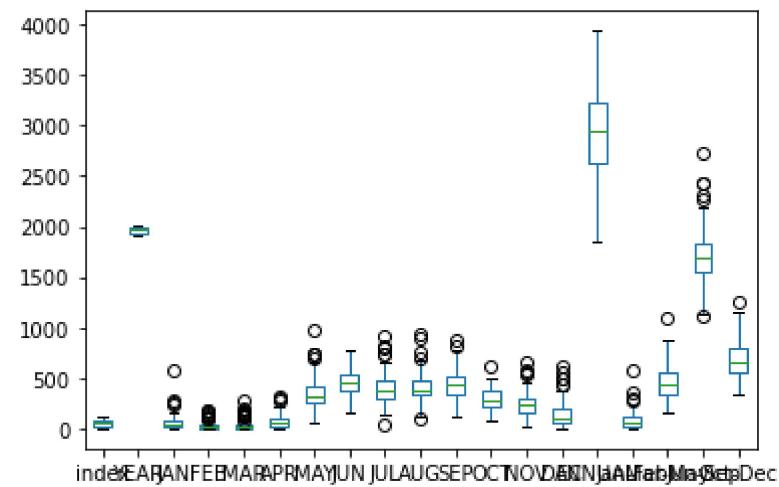
Out[10]: <AxesSubplot:>



Box Chart

```
In [11]: 1 df.plot.box()
```

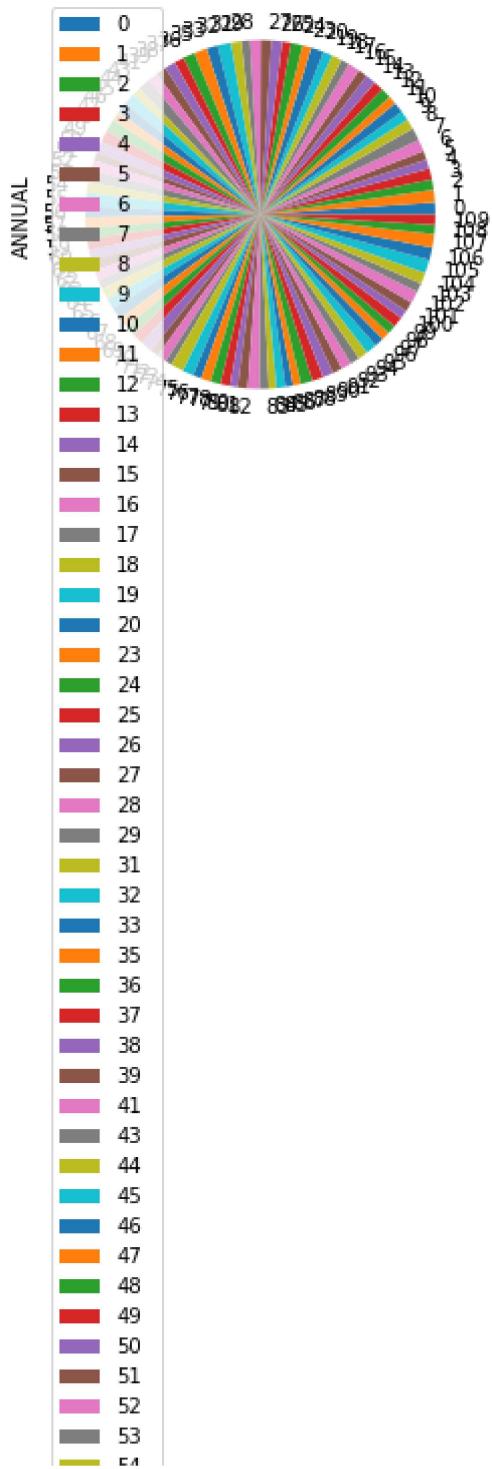
```
Out[11]: <AxesSubplot:>
```



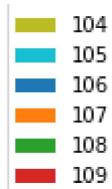
Pie Chart

```
In [12]: 1 df.plot.pie(y='ANNUAL')
```

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

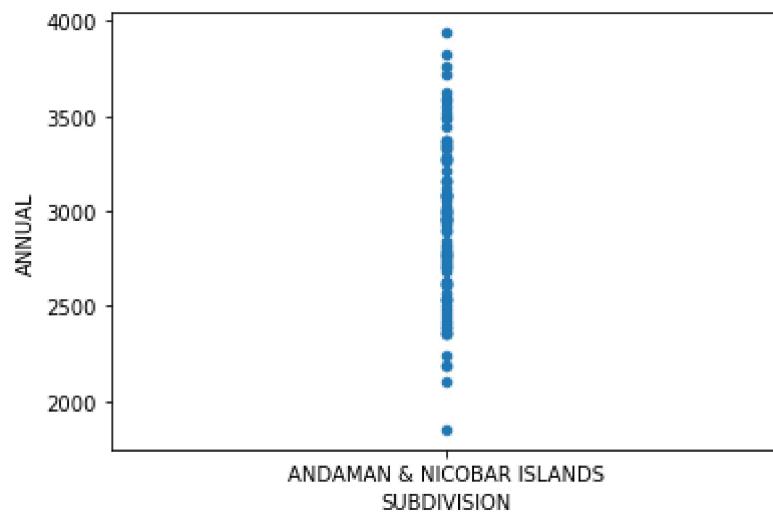
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103



Scatter Plot

```
In [13]: 1 df.plot.scatter(x='SUBDIVISION',y='ANNUAL')
```

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



In [14]: 1 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
```

```
In [15]: 1 df.describe()
```

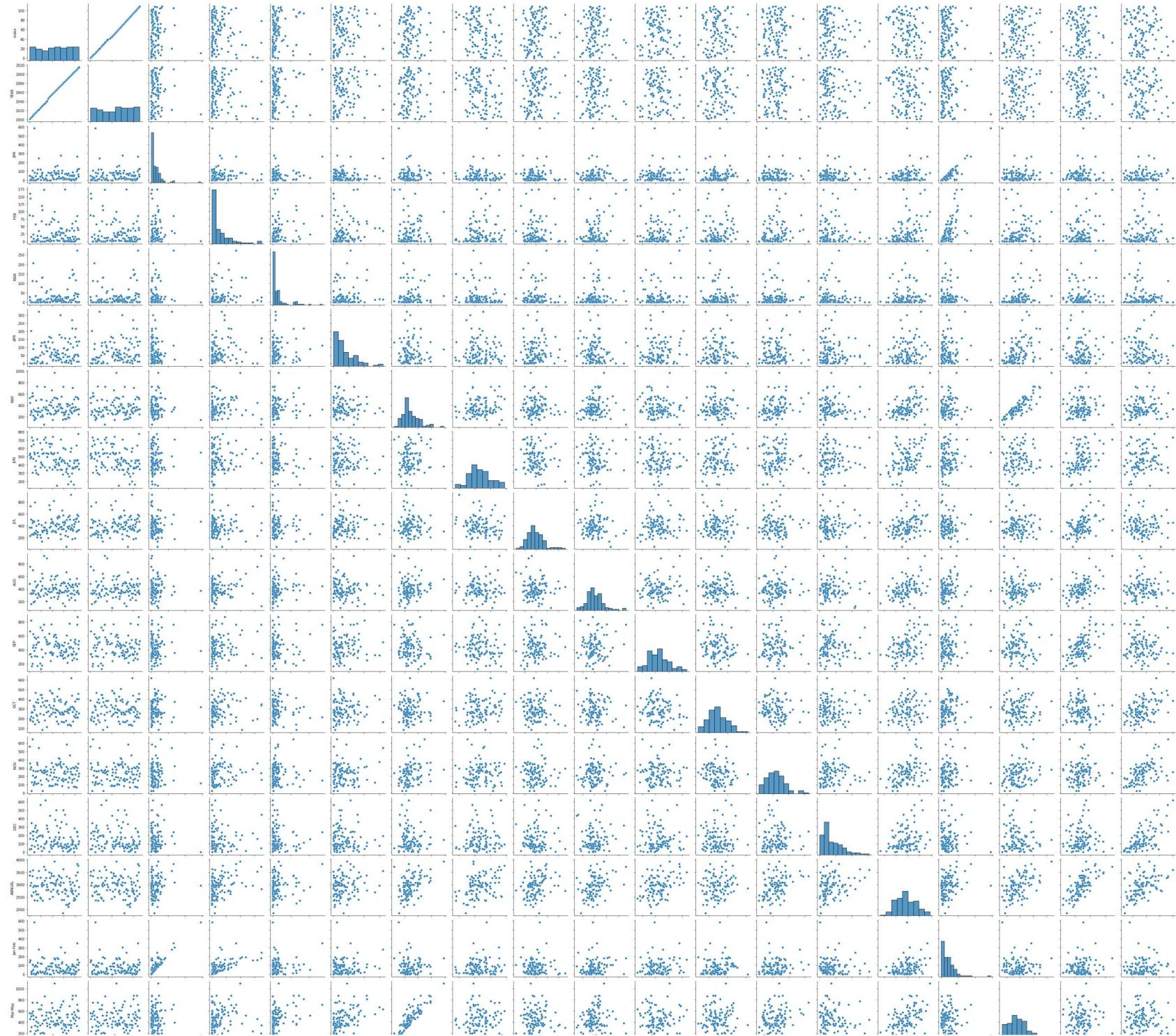
Out[15]:

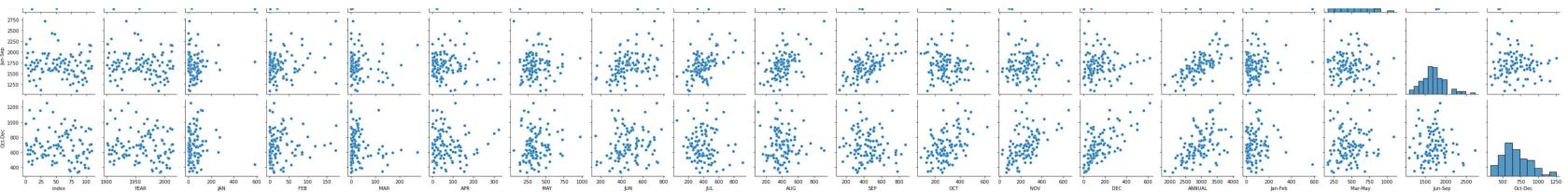
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG
count	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000
mean	55.826923	1960.355769	53.829808	28.299038	31.080769	71.473077	361.098077	465.357692	393.457692	402.648077
std	32.254884	34.010826	75.012392	38.286466	48.842153	66.908670	150.341139	136.471692	148.868238	139.164658
min	0.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	62.000000	148.800000	45.300000	101.300000
25%	27.750000	1929.750000	10.200000	1.775000	2.300000	21.025000	263.125000	369.975000	301.475000	327.600000
50%	57.500000	1963.500000	31.750000	12.800000	12.100000	52.300000	321.050000	450.250000	374.000000	378.500000
75%	83.250000	1989.250000	76.275000	36.325000	31.775000	103.350000	425.325000	545.625000	468.900000	468.025000
max	109.000000	2015.000000	583.700000	173.800000	272.800000	323.100000	973.100000	777.000000	918.500000	924.900000

EDA And Visualization

```
In [16]: 1 sns.pairplot(df)
```

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x2653cbc9370>
```

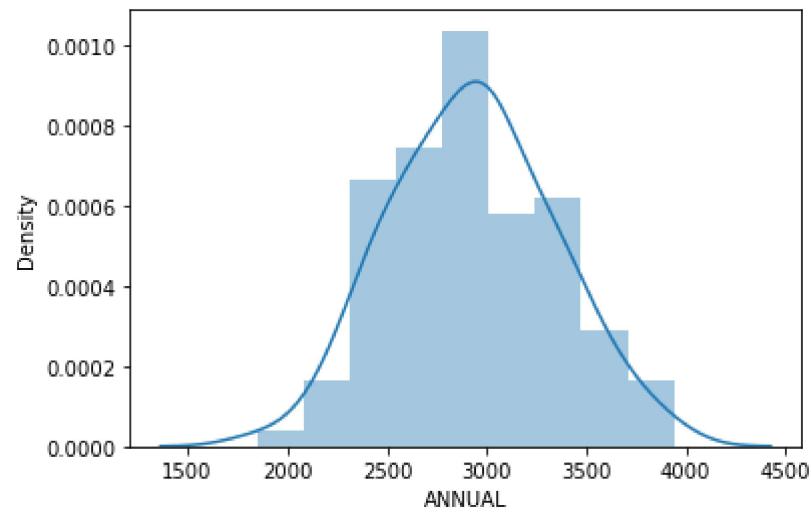





In [17]: 1 sns.distplot(df['ANNUAL'])

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]: 1 sns.heatmap(df.corr())
```

```
Out[18]: <AxesSubplot:>
```

