

## 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('ASSAM MEGHALAYA.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	207	ASSAM & MEGHALAYA	1901	27.1	19.5	30.6	223.0	207.0	524.9	430.6	464.1	291.4	163.7	115.6	1.2	2498.6	46.6	460.5
1	208	ASSAM & MEGHALAYA	1902	9.3	10.2	105.6	350.0	262.1	620.7	510.8	536.0	441.3	97.0	7.8	1.3	2952.1	19.4	717.6
2	209	ASSAM & MEGHALAYA	1903	19.9	25.4	103.6	140.6	206.6	607.4	362.7	551.9	306.4	159.5	59.3	1.3	2544.7	45.4	450.8
3	210	ASSAM & MEGHALAYA	1904	11.1	56.1	51.9	457.1	375.2	385.7	477.6	438.8	245.9	115.9	46.4	2.5	2664.1	67.1	884.2
4	211	ASSAM & MEGHALAYA	1905	19.9	16.9	137.9	213.0	275.5	521.7	439.1	649.1	276.0	200.0	16.8	24.8	2790.6	36.8	626.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
110	317	ASSAM & MEGHALAYA	2011	11.1	11.4	109.0	92.1	238.3	316.0	395.8	302.6	221.6	30.2	11.9	3.5	1743.4	22.5	439.4
111	318	ASSAM & MEGHALAYA	2012	15.2	6.9	28.8	279.1	185.8	729.7	444.3	289.2	411.6	199.4	17.1	2.3	2609.4	22.1	493.7
112	319	ASSAM & MEGHALAYA	2013	1.1	9.6	44.0	112.8	346.7	286.2	367.8	289.7	229.3	126.3	1.0	2.0	1816.4	10.7	503.5
113	320	ASSAM & MEGHALAYA	2014	2.0	28.3	29.3	51.5	351.1	426.4	374.4	484.6	420.2	35.0	3.0	0.4	2206.1	30.3	431.9
114	321	ASSAM & MEGHALAYA	2015	13.4	15.5	37.5	250.9	332.5	558.5	300.1	590.9	279.9	62.6	14.0	15.2	2470.9	28.9	620.9

115 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	207		ASSAM & MEGHALAYA	1901	27.1	19.5	30.6	223.0	207.0	524.9	430.6	464.1	291.4	163.7	115.6	1.2	2498.6	46.6	460.5
1	208		ASSAM & MEGHALAYA	1902	9.3	10.2	105.6	350.0	262.1	620.7	510.8	536.0	441.3	97.0	7.8	1.3	2952.1	19.4	717.6
2	209		ASSAM & MEGHALAYA	1903	19.9	25.4	103.6	140.6	206.6	607.4	362.7	551.9	306.4	159.5	59.3	1.3	2544.7	45.4	450.8
3	210		ASSAM & MEGHALAYA	1904	11.1	56.1	51.9	457.1	375.2	385.7	477.6	438.8	245.9	115.9	46.4	2.5	2664.1	67.1	884.2
4	211		ASSAM & MEGHALAYA	1905	19.9	16.9	137.9	213.0	275.5	521.7	439.1	649.1	276.0	200.0	16.8	24.8	2790.6	36.8	626.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
110	317		ASSAM & MEGHALAYA	2011	11.1	11.4	109.0	92.1	238.3	316.0	395.8	302.6	221.6	30.2	11.9	3.5	1743.4	22.5	439.4
111	318		ASSAM & MEGHALAYA	2012	15.2	6.9	28.8	279.1	185.8	729.7	444.3	289.2	411.6	199.4	17.1	2.3	2609.4	22.1	493.7
112	319		ASSAM & MEGHALAYA	2013	1.1	9.6	44.0	112.8	346.7	286.2	367.8	289.7	229.3	126.3	1.0	2.0	1816.4	10.7	503.5
113	320		ASSAM & MEGHALAYA	2014	2.0	28.3	29.3	51.5	351.1	426.4	374.4	484.6	420.2	35.0	3.0	0.4	2206.1	30.3	431.9
114	321		ASSAM & MEGHALAYA	2015	13.4	15.5	37.5	250.9	332.5	558.5	300.1	590.9	279.9	62.6	14.0	15.2	2470.9	28.9	620.9

115 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')
```

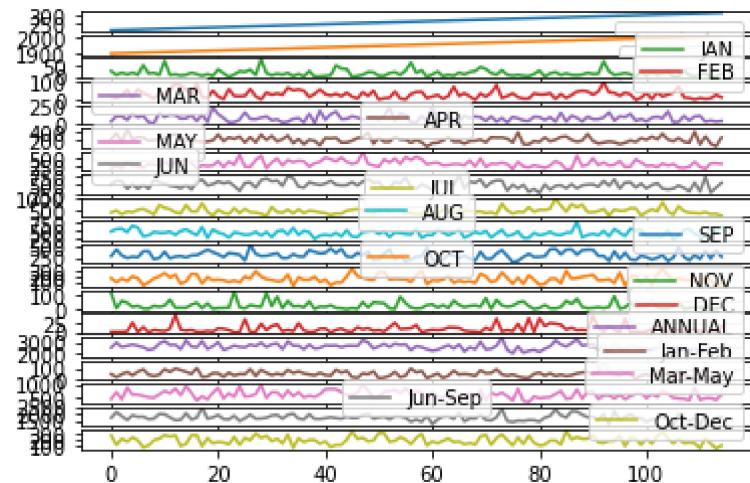
In [5]: 1 df.info()

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

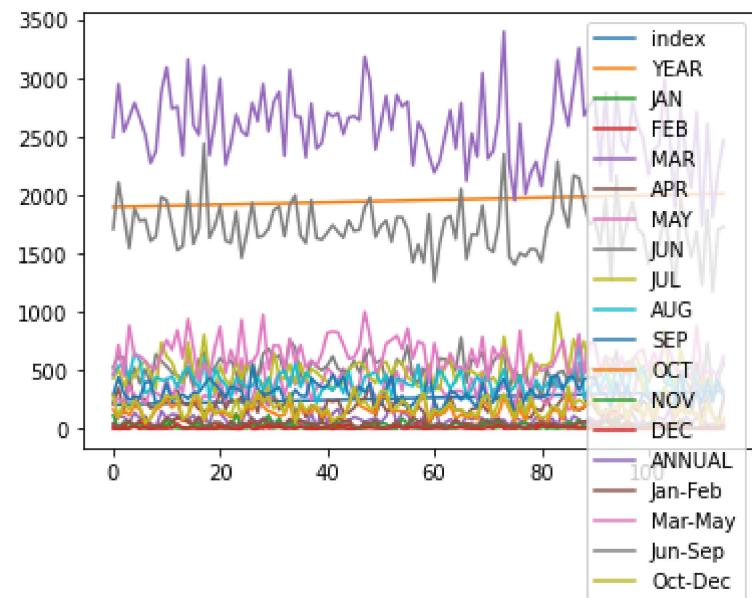
```
In [6]: 1 df.plot.line(subplots=True)
```

```
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()

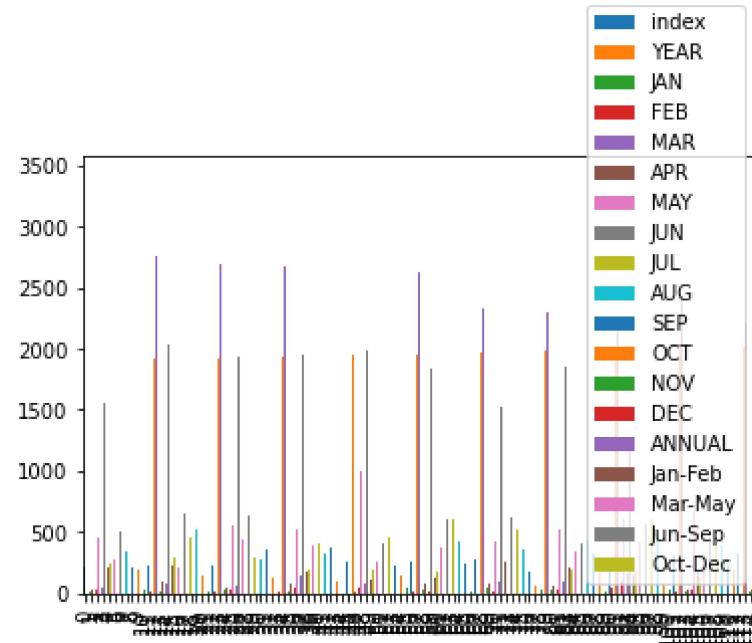
Out[7]: <AxesSubplot:>



## Bar Chart

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

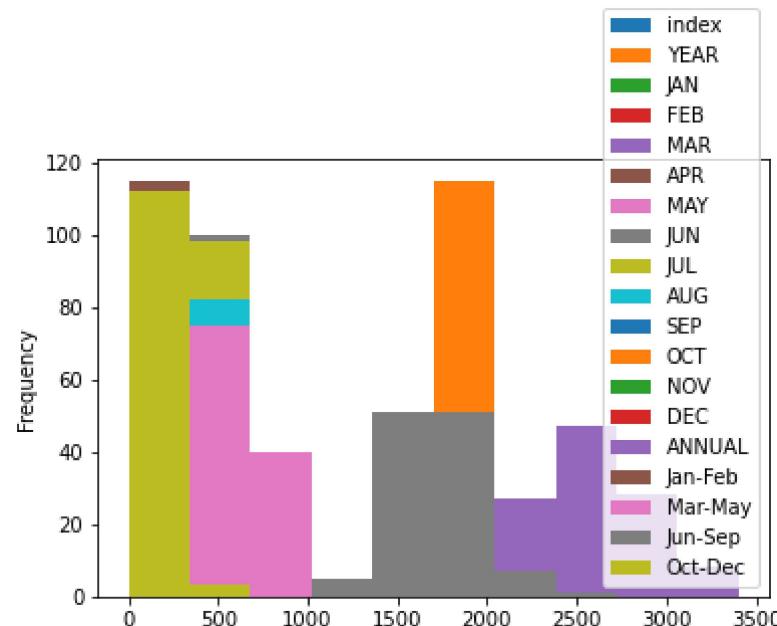
Out[8]: <AxesSubplot:>



## Histogram

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

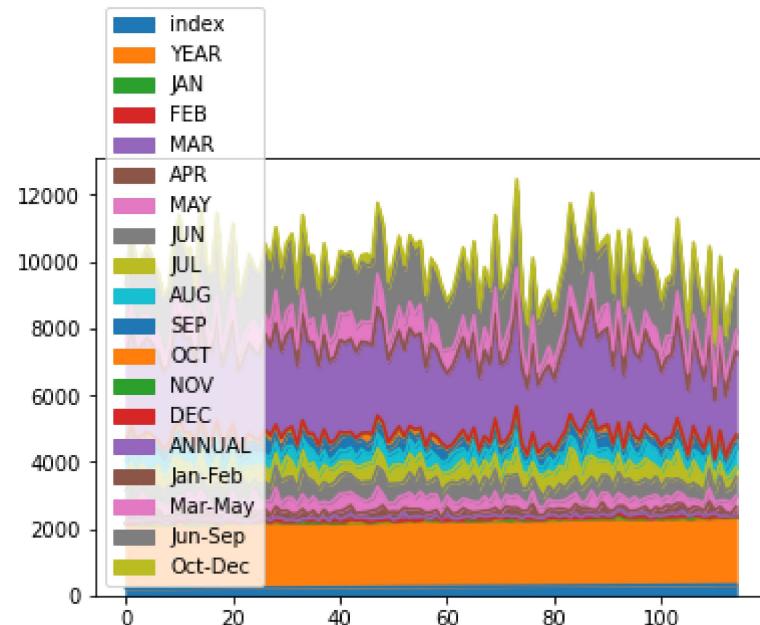
Out[9]: <AxesSubplot:ylabel='Frequency'>



## Area Chart

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

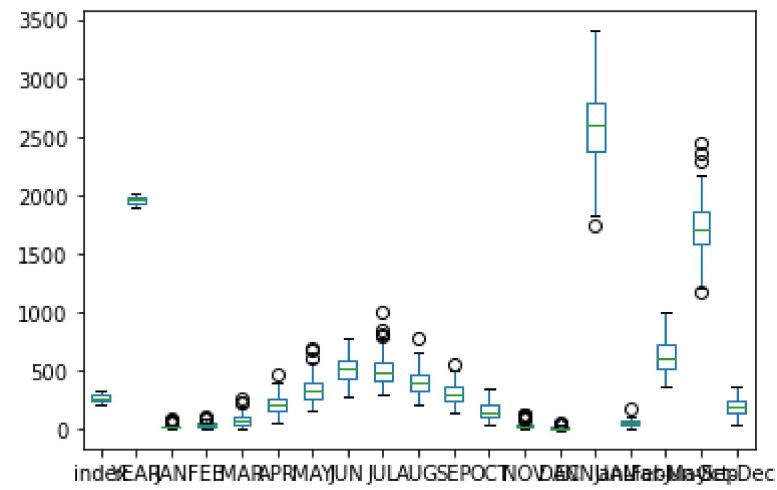
Out[10]: <AxesSubplot:>



## Box Chart

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

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

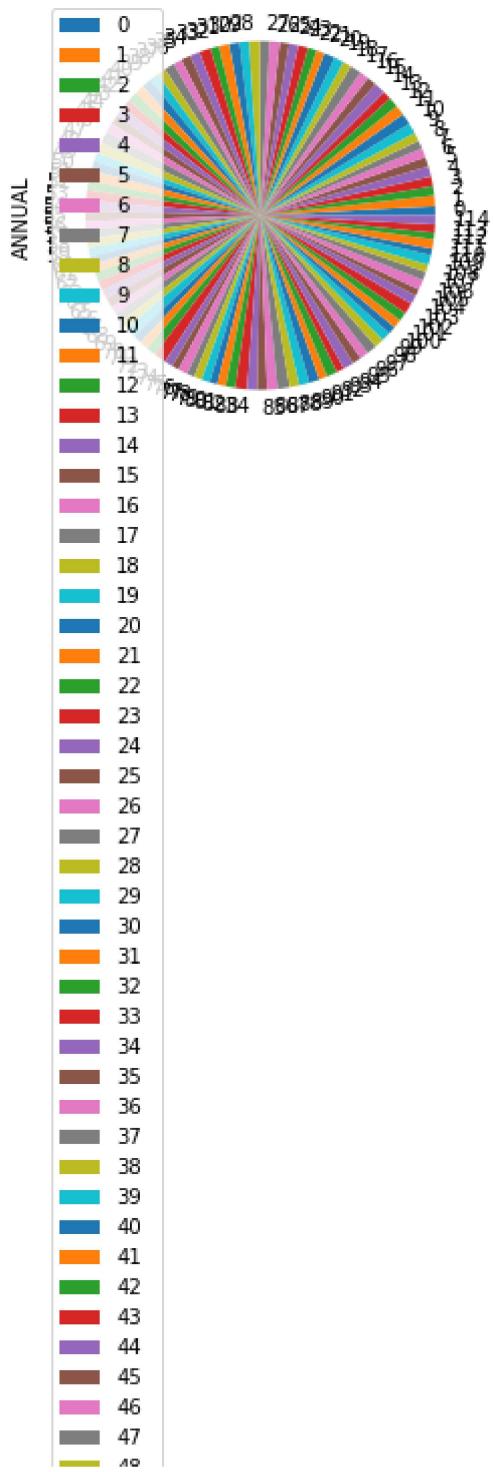


Pie Chart

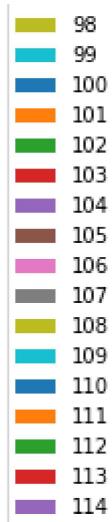
```
In [12]: 1 df.plot.pie(y='ANNUAL')
```

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





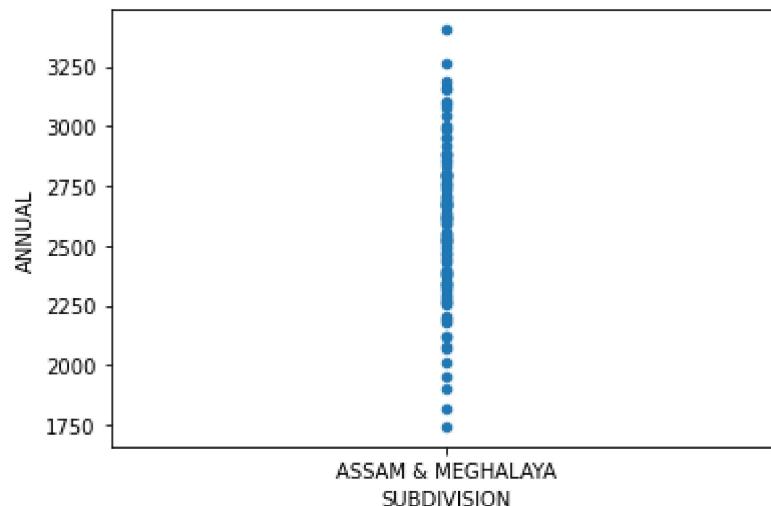
40
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97



## 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: 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
```

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

Out[15]:

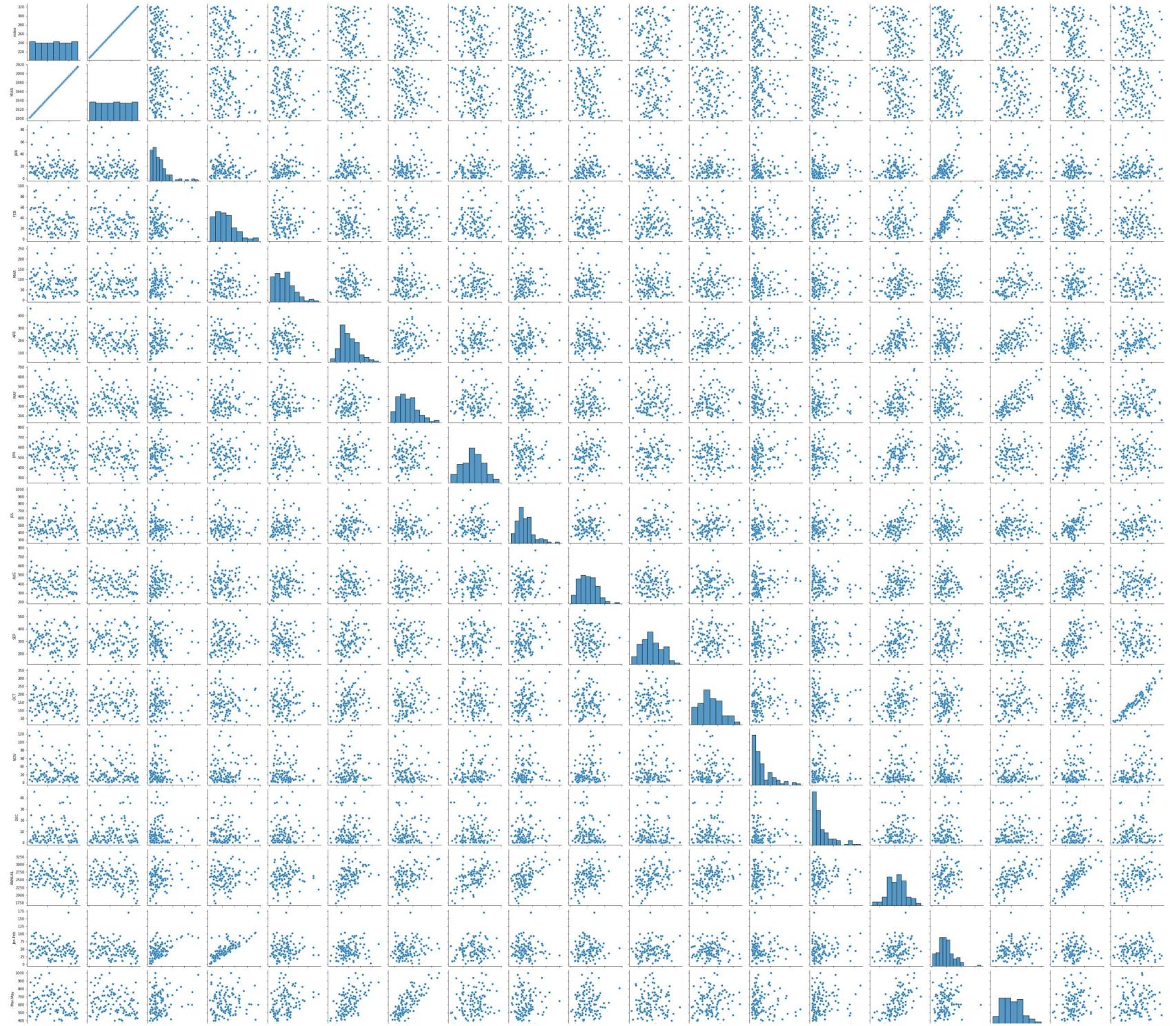
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	264.000000	1958.000000	16.974783	31.441739	79.026957	203.115652	341.539130	510.161739	495.102609	404.593043
std	33.341666	33.341666	15.668601	20.963452	48.679075	74.338367	105.771005	106.248805	123.681262	100.365062
min	207.000000	1901.000000	0.100000	0.500000	4.400000	45.900000	159.700000	273.100000	287.800000	209.700000
25%	235.500000	1929.500000	7.500000	15.800000	38.700000	151.350000	261.100000	421.550000	409.300000	332.000000
50%	264.000000	1958.000000	12.800000	28.300000	75.700000	197.600000	325.700000	513.300000	474.500000	400.700000
75%	292.500000	1986.500000	21.600000	41.950000	104.600000	251.500000	397.150000	583.650000	559.150000	470.250000
max	321.000000	2015.000000	83.900000	96.900000	253.800000	457.100000	681.200000	780.500000	995.200000	773.400000

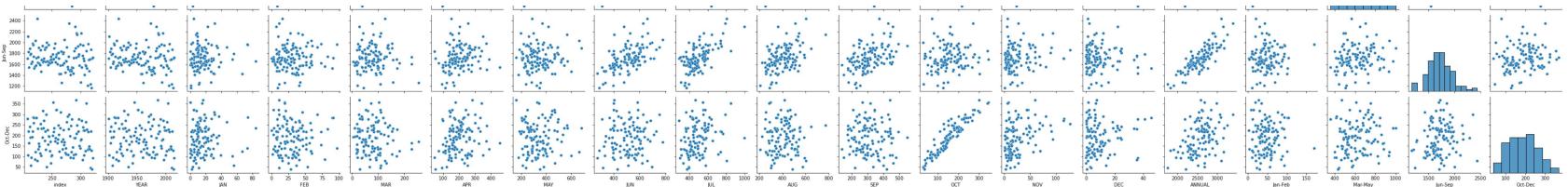
## EDA And Visualization

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

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x21454380df0>
```



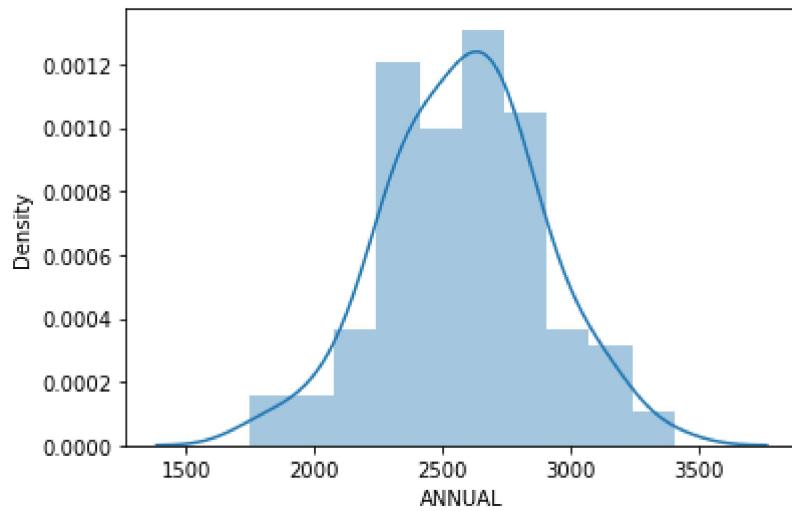




```
In [17]: 1 sns.distplot(df['ANNUAL'])
```

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:>

