

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('JAMMU_KASHMIR.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	1702		JAMMU & KASHMIR	1901	66.4	69.3	69.6	132.2	105.8	53.4	171.7	181.3	101.8	24.1	0.0	4.4	979.9	135.6	307.6
1	1703		JAMMU & KASHMIR	1902	6.5	9.7	91.3	100.5	70.7	113.3	108.4	136.9	62.2	15.1	44.0	1.8	760.5	16.2	262.5
2	1704		JAMMU & KASHMIR	1903	96.2	21.5	238.6	58.7	57.3	18.9	332.5	218.6	176.9	10.7	15.0	41.8	1286.6	117.7	354.6
3	1705		JAMMU & KASHMIR	1904	110.6	17.3	145.2	64.5	67.8	25.9	182.3	132.2	62.3	50.0	24.8	99.2	982.2	128.0	277.5
4	1706		JAMMU & KASHMIR	1905	146.7	76.3	161.4	71.7	65.2	43.3	145.2	111.5	239.7	5.8	0.6	90.2	1157.7	223.0	298.4
...	
110	1812		JAMMU & KASHMIR	2011	43.4	211.6	97.8	89.0	32.4	72.5	81.6	131.2	72.0	19.4	12.9	23.8	887.6	254.9	219.2
111	1813		JAMMU & KASHMIR	2012	150.9	95.8	45.2	86.6	48.9	32.6	118.8	264.9	106.7	15.7	10.8	57.8	1034.7	246.7	180.7
112	1814		JAMMU & KASHMIR	2013	52.2	136.4	41.9	47.4	47.4	80.5	125.1	219.1	41.2	34.4	13.4	20.3	859.3	188.6	136.7
113	1815		JAMMU & KASHMIR	2014	75.8	64.0	153.1	76.1	52.7	25.3	100.5	134.6	362.8	32.2	14.1	2.3	1093.4	139.8	281.9
114	1816		JAMMU & KASHMIR	2015	27.9	187.2	341.4	173.3	64.6	121.4	233.2	129.2	130.2	87.1	38.1	39.3	1572.8	215.1	579.3

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	1702		JAMMU & KASHMIR	1901	66.4	69.3	69.6	132.2	105.8	53.4	171.7	181.3	101.8	24.1	0.0	4.4	979.9	135.6	307.6
1	1703		JAMMU & KASHMIR	1902	6.5	9.7	91.3	100.5	70.7	113.3	108.4	136.9	62.2	15.1	44.0	1.8	760.5	16.2	262.5
2	1704		JAMMU & KASHMIR	1903	96.2	21.5	238.6	58.7	57.3	18.9	332.5	218.6	176.9	10.7	15.0	41.8	1286.6	117.7	354.6
3	1705		JAMMU & KASHMIR	1904	110.6	17.3	145.2	64.5	67.8	25.9	182.3	132.2	62.3	50.0	24.8	99.2	982.2	128.0	277.5
4	1706		JAMMU & KASHMIR	1905	146.7	76.3	161.4	71.7	65.2	43.3	145.2	111.5	239.7	5.8	0.6	90.2	1157.7	223.0	298.4
...	
110	1812		JAMMU & KASHMIR	2011	43.4	211.6	97.8	89.0	32.4	72.5	81.6	131.2	72.0	19.4	12.9	23.8	887.6	254.9	219.2
111	1813		JAMMU & KASHMIR	2012	150.9	95.8	45.2	86.6	48.9	32.6	118.8	264.9	106.7	15.7	10.8	57.8	1034.7	246.7	180.7
112	1814		JAMMU & KASHMIR	2013	52.2	136.4	41.9	47.4	47.4	80.5	125.1	219.1	41.2	34.4	13.4	20.3	859.3	188.6	136.7
113	1815		JAMMU & KASHMIR	2014	75.8	64.0	153.1	76.1	52.7	25.3	100.5	134.6	362.8	32.2	14.1	2.3	1093.4	139.8	281.9
114	1816		JAMMU & KASHMIR	2015	27.9	187.2	341.4	173.3	64.6	121.4	233.2	129.2	130.2	87.1	38.1	39.3	1572.8	215.1	579.3

114 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')
```

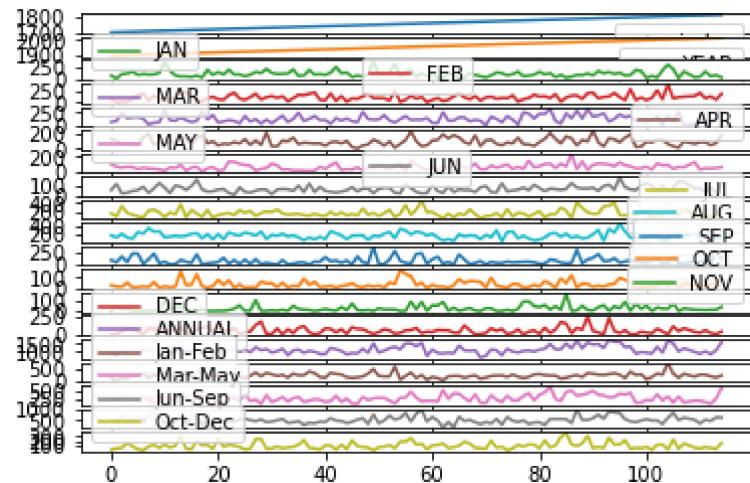
In [5]: 1 df.info()

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

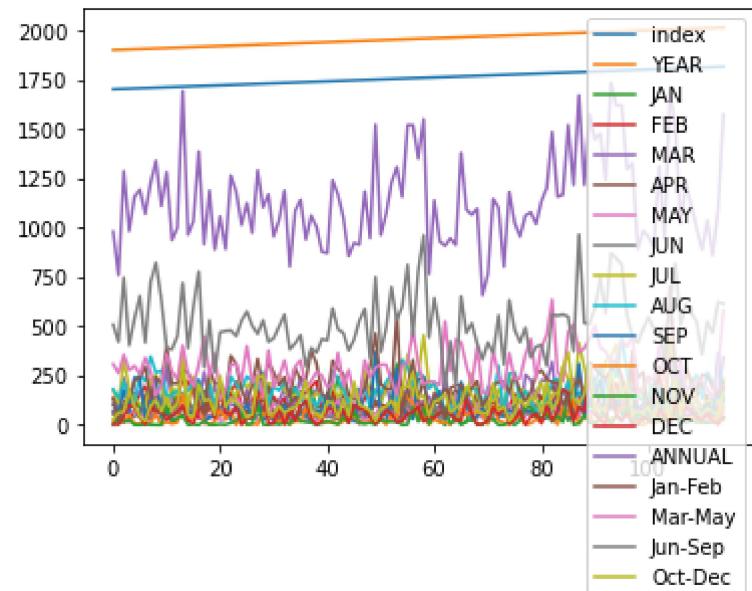
```
In [6]: 1 df.plot.line(subplots=True)
```

```
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()
```

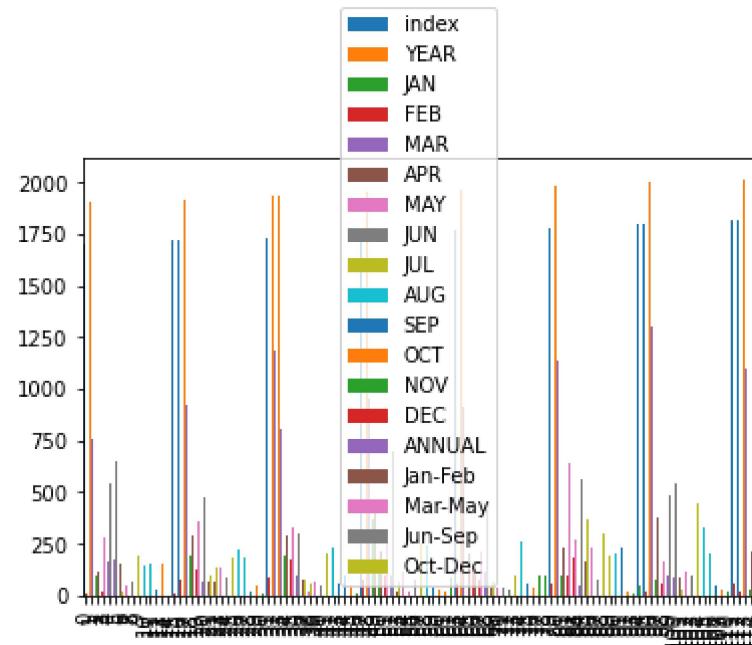
```
Out[7]: <AxesSubplot:>
```



Bar Chart

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

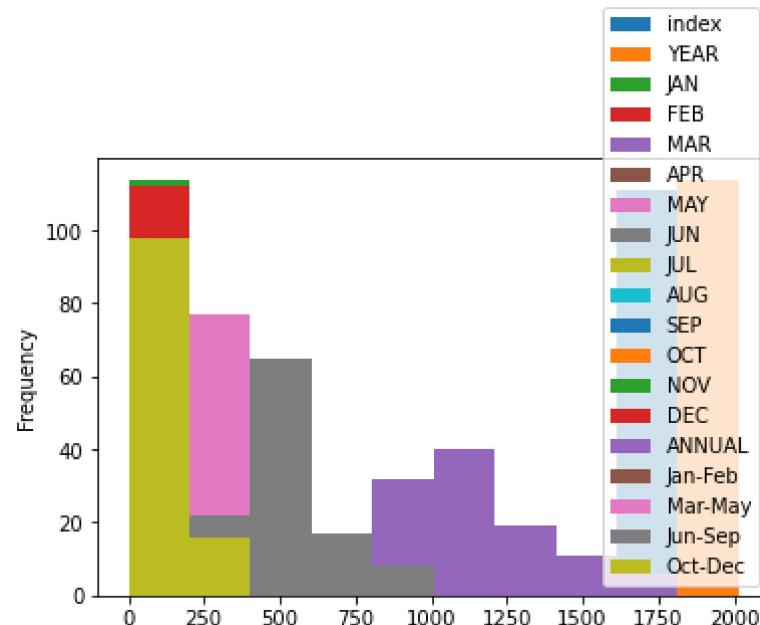
Out[8]: <AxesSubplot:>



Histogram

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

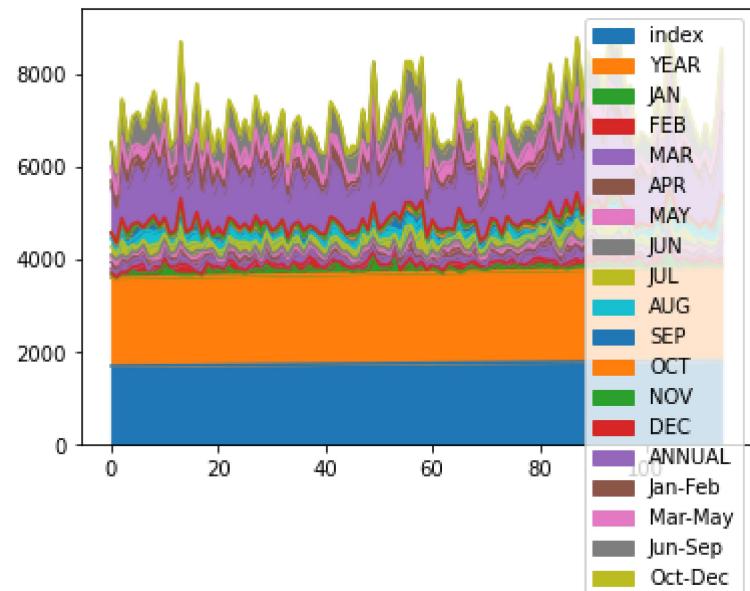
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area Chart

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

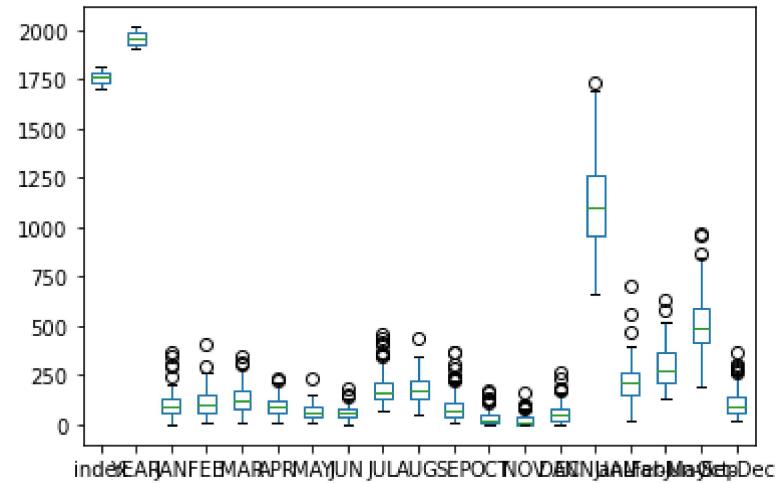
Out[10]: <AxesSubplot:>



Box Chart

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

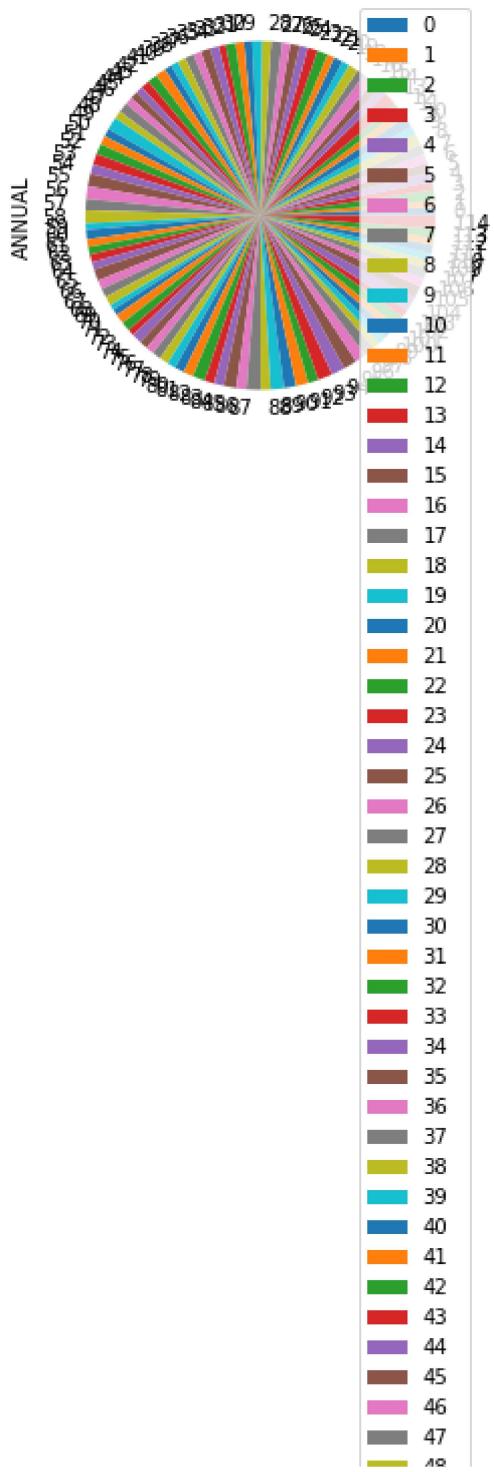
```
Out[11]: <AxesSubplot:>
```



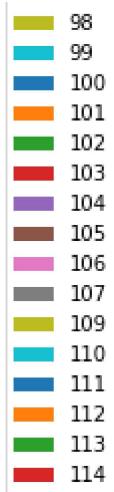
Pie Chart

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

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

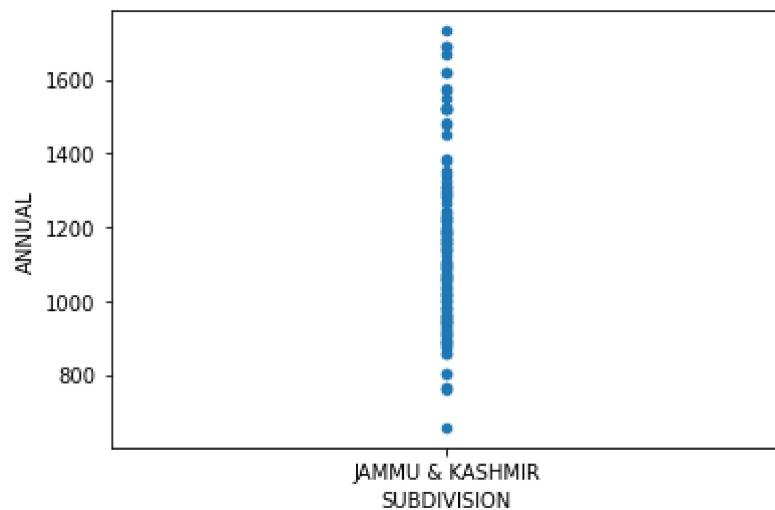
40
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97



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: 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
```

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

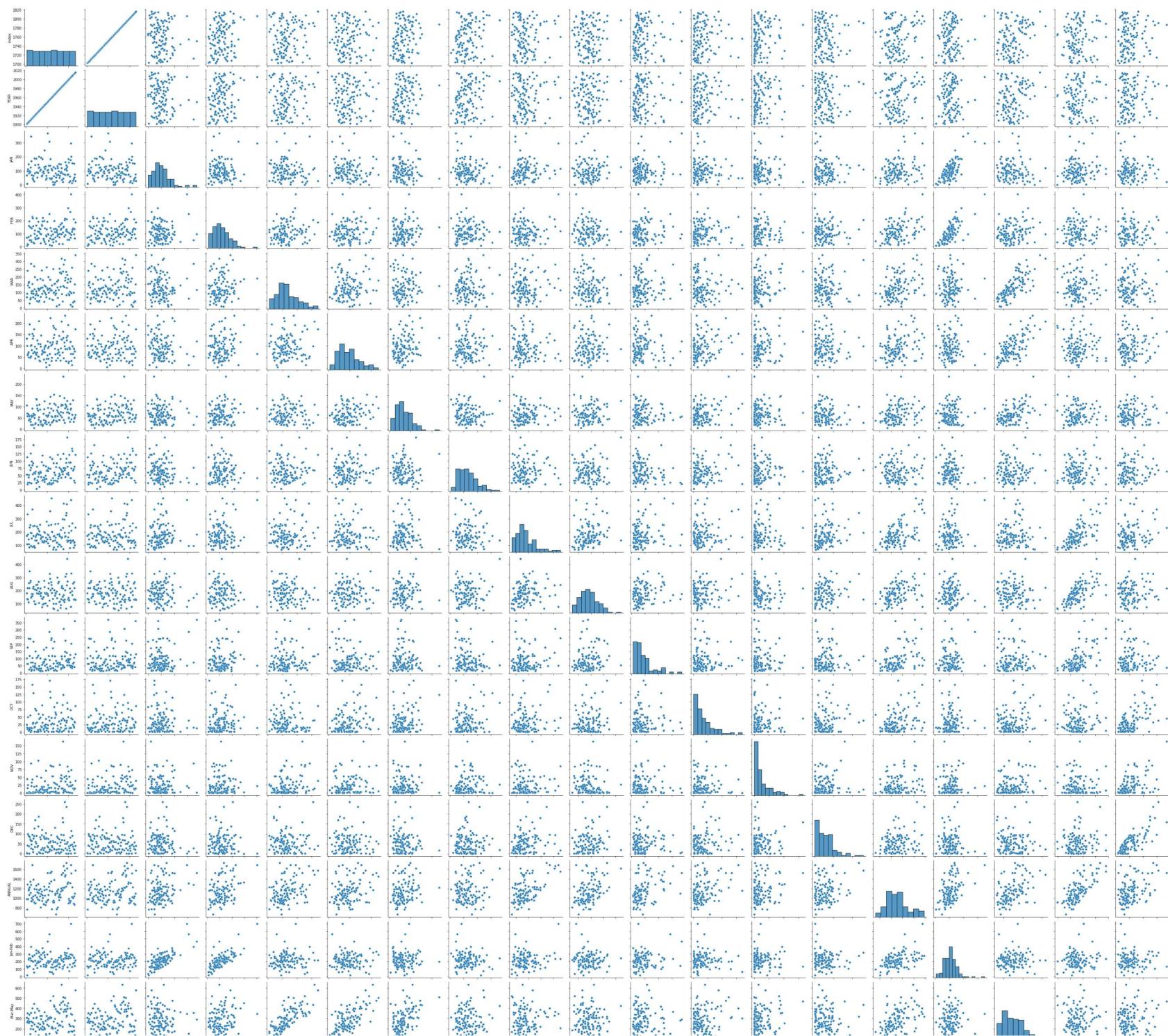
Out[15]:

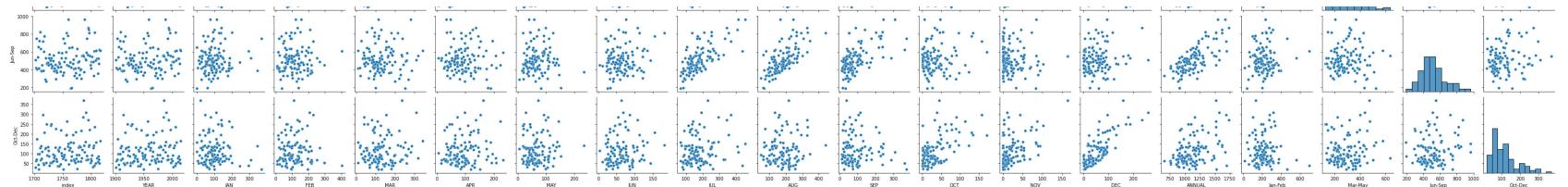
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
count	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000
mean	1758.552632	1957.552632	101.870175	115.428947	131.971930	93.528947	67.504386	64.235965	179.837719	181.714035	89.600000
std	33.140380	33.140380	66.496789	67.229113	71.742153	48.559036	37.603315	33.293551	80.605044	71.137088	74.400000
min	1702.000000	1901.000000	0.000000	9.700000	9.900000	5.700000	4.700000	4.000000	67.200000	49.100000	11.200000
25%	1730.250000	1929.250000	59.000000	64.600000	85.675000	59.475000	40.025000	39.300000	126.800000	131.125000	39.300000
50%	1758.500000	1957.500000	89.800000	101.950000	116.400000	86.700000	61.200000	59.750000	163.200000	176.750000	66.600000
75%	1786.750000	1985.750000	130.700000	149.900000	174.800000	119.450000	91.825000	81.075000	210.500000	220.300000	113.600000
max	1816.000000	2015.000000	367.800000	403.500000	341.400000	233.200000	234.400000	182.000000	454.600000	441.500000	371.100000

EDA And Visualization

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

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x2889cceee50>
```

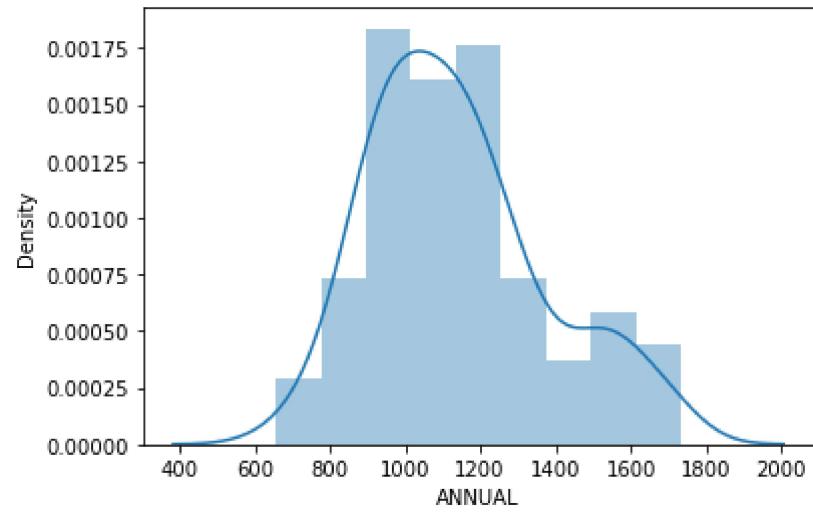





```
In [17]: 1 sns.distplot(df['ANNUAL'])
```

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:>
```

