

20104028

Heamnath N

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_lakshadweep.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	4002	LAKSHADWEEP	1901	22.6	86.4	114.8	263.8	37.3	459.0	0.0	0.0	46.7	183.7	229.1
1	4003	LAKSHADWEEP	1902	99.3	9.6	32.6	40.4	179.1	374.2	413.3	170.0	214.3	384.2	192.1
2	4004	LAKSHADWEEP	1903	63.5	95.0	0.0	29.5	144.1	212.4	261.8	202.0	292.1	79.1	NaN
3	4005	LAKSHADWEEP	1904	0.0	0.0	13.5	13.2	143.3	261.3	256.0	38.9	219.9	153.6	8.1
4	4006	LAKSHADWEEP	1905	62.4	0.0	0.0	0.0	166.7	400.7	68.7	377.5	107.5	232.1	159.1
...
109	4111	LAKSHADWEEP	2011	5.1	2.8	3.1	85.9	107.2	153.6	350.2	254.0	255.2	117.4	184.1
110	4112	LAKSHADWEEP	2012	19.2	0.1	1.6	76.8	21.2	327.0	231.5	381.2	179.8	145.9	12.1
111	4113	LAKSHADWEEP	2013	26.2	34.4	37.5	5.3	88.3	426.2	296.4	154.4	180.0	72.8	78.1
112	4114	LAKSHADWEEP	2014	53.2	16.1	4.4	14.9	57.4	244.1	116.1	466.1	132.2	169.2	59.1
113	4115	LAKSHADWEEP	2015	2.2	0.5	3.7	87.1	133.1	296.6	257.5	146.4	160.4	165.4	231.1

114 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')
```

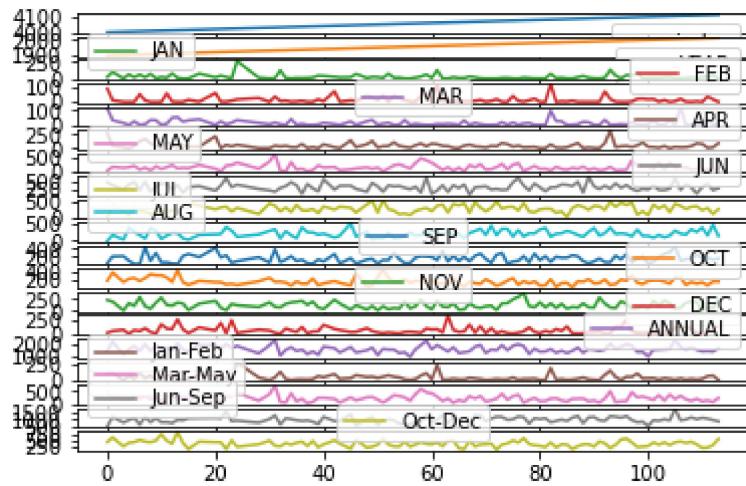
```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103 entries, 0 to 113
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   index       103 non-null    int64  
 1   SUBDIVISION 103 non-null    object 
 2   YEAR        103 non-null    int64  
 3   JAN         103 non-null    float64 
 4   FEB         103 non-null    float64 
 5   MAR         103 non-null    float64 
 6   APR         103 non-null    float64 
 7   MAY         103 non-null    float64 
 8   JUN         103 non-null    float64 
 9   JUL         103 non-null    float64 
 10  AUG         103 non-null    float64 
 11  SEP         103 non-null    float64 
 12  OCT         103 non-null    float64 
 13  NOV         103 non-null    float64 
 14  DEC         103 non-null    float64 
 15  ANNUAL      103 non-null    float64 
 16  Jan-Feb     103 non-null    float64 
 17  Mar-May     103 non-null    float64 
 18  Jun-Sep     103 non-null    float64 
 19  Oct-Dec     103 non-null    float64 
dtypes: float64(17), int64(2), object(1)
memory usage: 16.9+ KB
```

Line chart

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

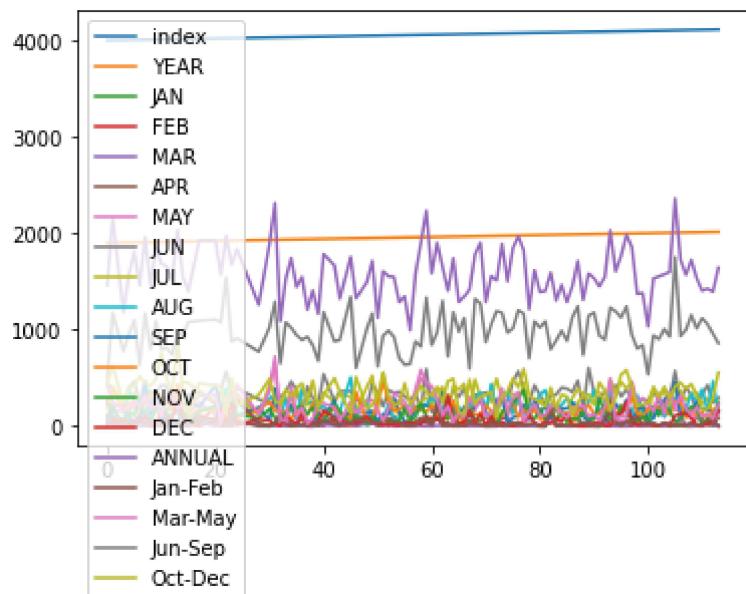
```
Out[6]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



Line chart

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

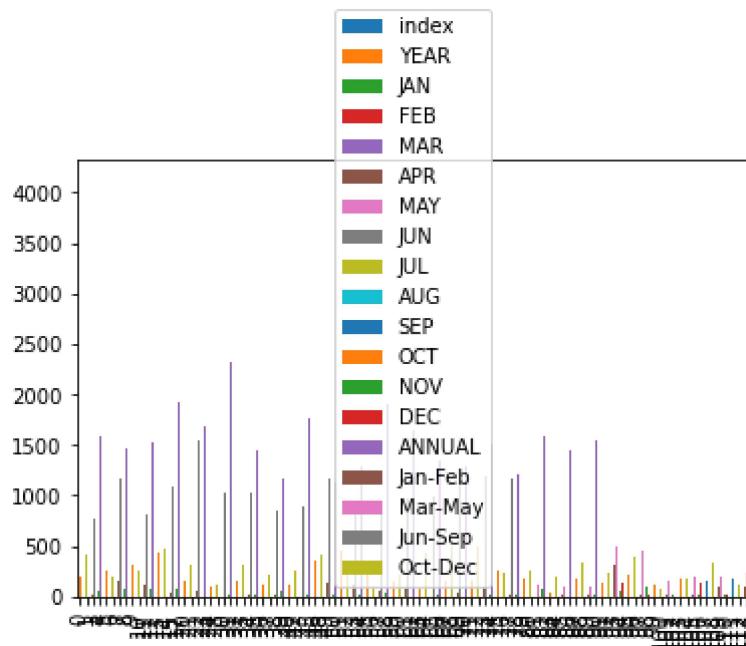
```
Out[7]: <AxesSubplot:>
```



Bar chart

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

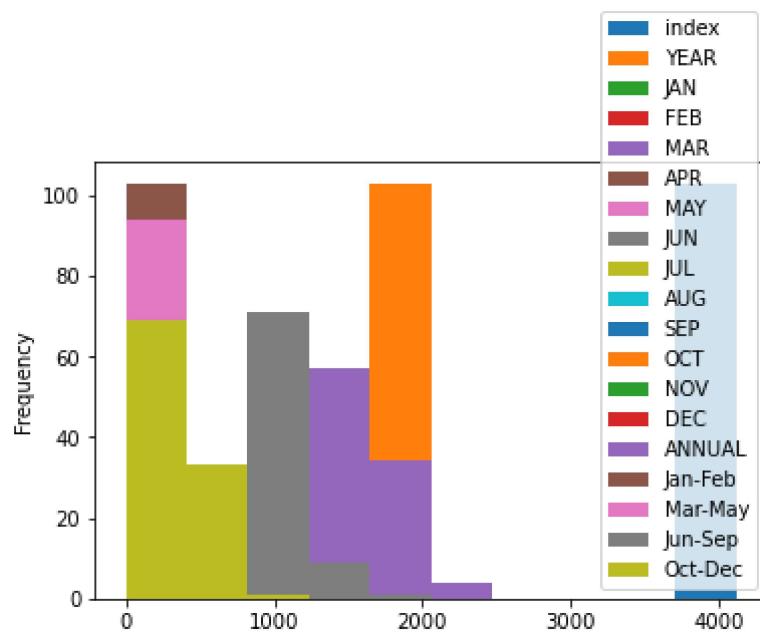
```
Out[8]: <AxesSubplot:>
```



Histogram

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

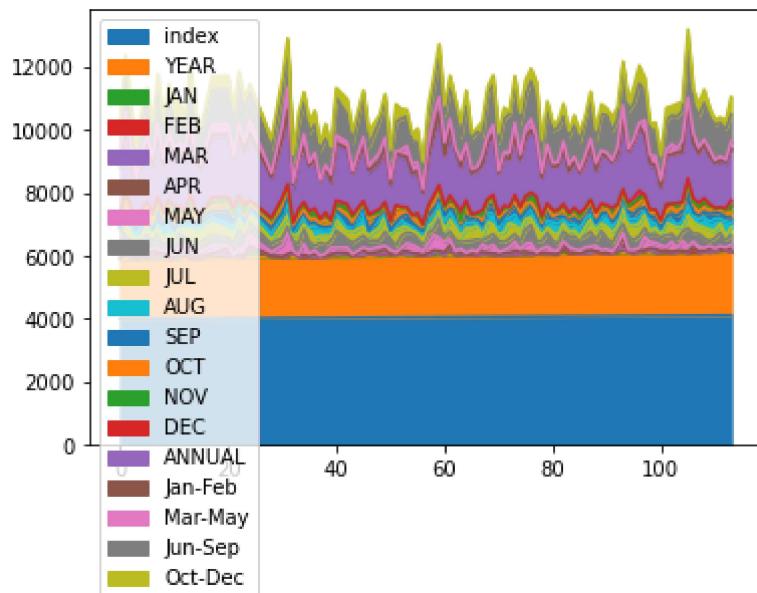
```
Out[9]: <AxesSubplot:ylabel='Frequency'>
```



Area chart

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

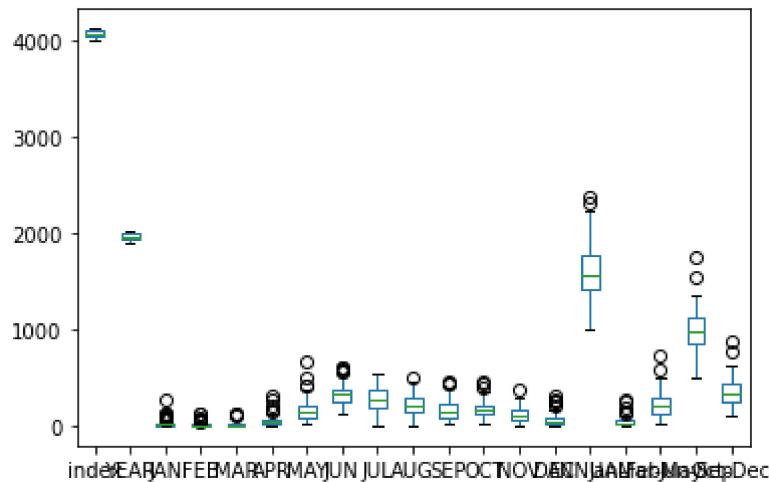
```
Out[10]: <AxesSubplot:>
```



Box chart

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

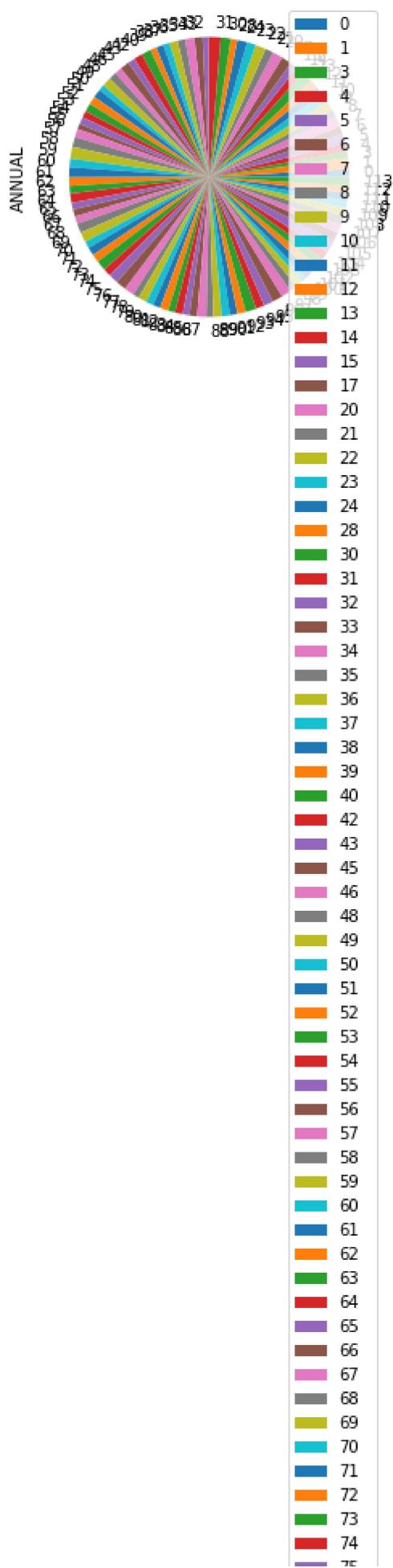
```
Out[11]: <AxesSubplot: >
```

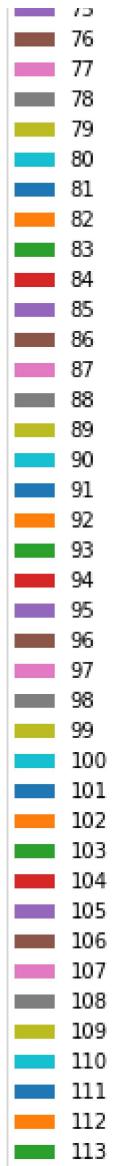


Pie chart

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

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

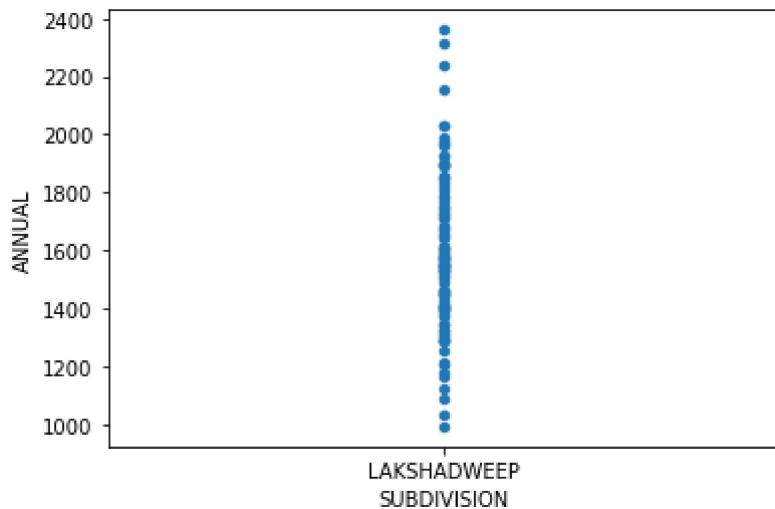




Scatter chart

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

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



In [14]:

`df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103 entries, 0 to 113
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   index       103 non-null    int64  
 1   SUBDIVISION 103 non-null    object  
 2   YEAR        103 non-null    int64  
 3   JAN         103 non-null    float64 
 4   FEB         103 non-null    float64 
 5   MAR         103 non-null    float64 
 6   APR         103 non-null    float64 
 7   MAY         103 non-null    float64 
 8   JUN         103 non-null    float64 
 9   JUL         103 non-null    float64 
 10  AUG         103 non-null    float64 
 11  SEP         103 non-null    float64 
 12  OCT         103 non-null    float64 
 13  NOV         103 non-null    float64 
 14  DEC         103 non-null    float64 
 15  ANNUAL      103 non-null    float64 
 16  Jan-Feb     103 non-null    float64 
 17  Mar-May     103 non-null    float64 
 18  Jun-Sep     103 non-null    float64 
 19  Oct-Dec     103 non-null    float64 
dtypes: float64(17), int64(2), object(1)
memory usage: 16.9+ KB
```

In [15]:

`df.describe()`

Out[15]:

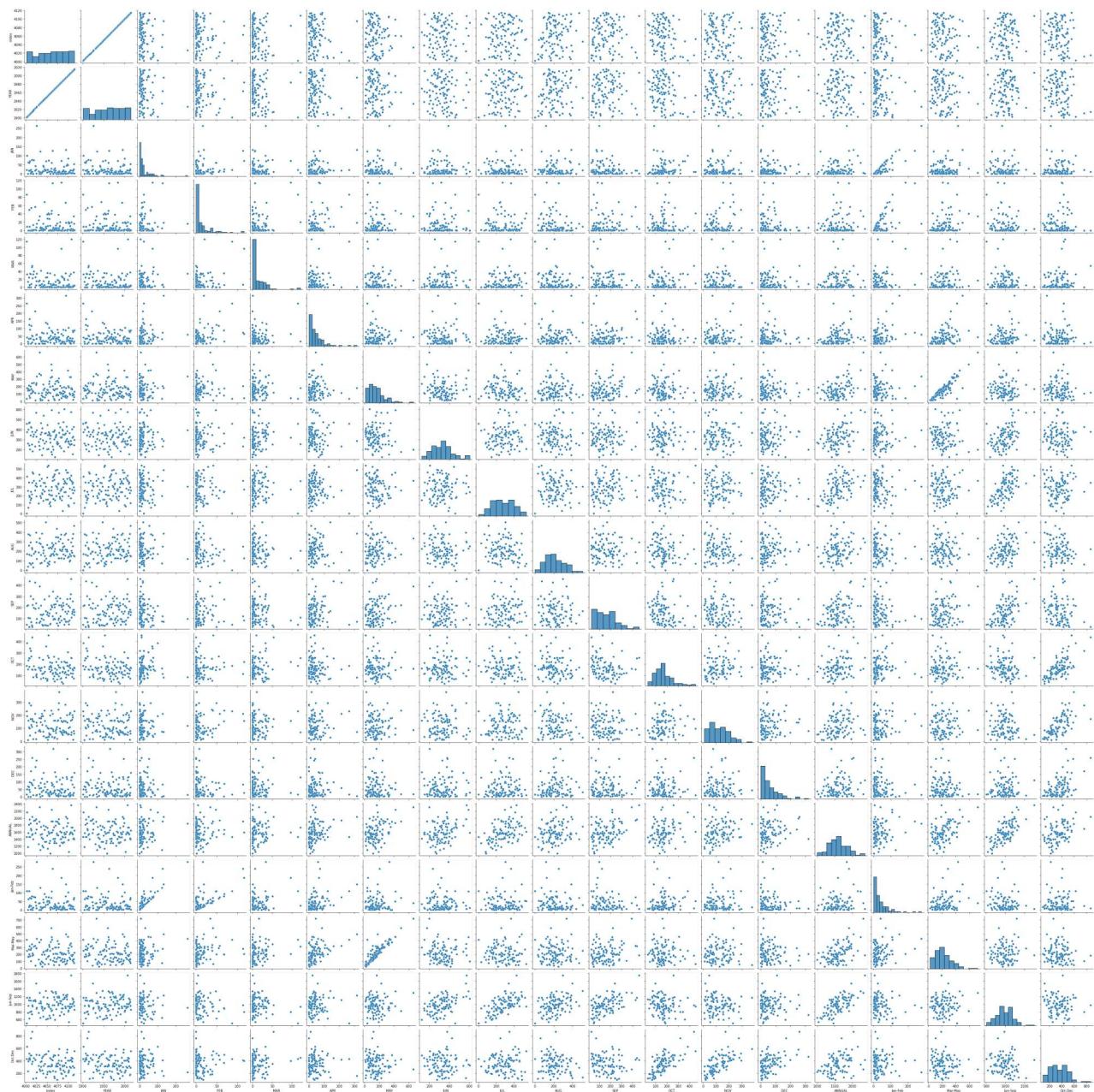
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
count	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000	103.000000
mean	4061.679612	1961.533981	25.324272	13.766019	14.128155	45.643689	159.710680	329.257282
std	32.970044	33.202237	37.228830	22.446431	21.518731	52.277828	111.277485	101.356376
min	4002.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	13.500000	125.600000
25%	4035.500000	1935.500000	3.900000	0.400000	0.450000	14.050000	80.500000	257.550000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
50%	4064.000000	1964.000000	12.300000	3.800000	5.200000	32.600000	142.800000	327.000000
75%	4089.500000	1989.500000	25.800000	16.800000	22.150000	61.750000	204.600000	379.550000
max	4115.000000	2015.000000	262.800000	114.900000	120.700000	315.400000	660.800000	604.300000

EDA AND VISUALIZATION

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

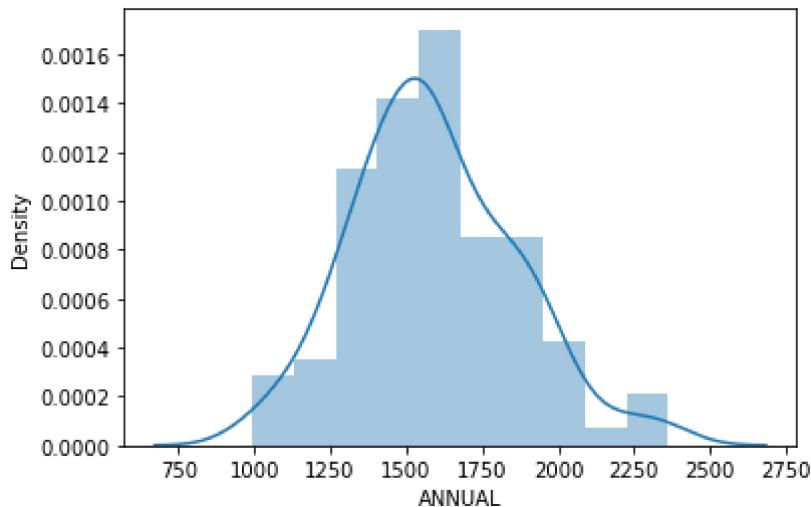
Out[16]: <seaborn.axisgrid.PairGrid at 0x132ce7864c0>



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:>
```

