

# 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_tamil nadu.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	3427	TAMIL NADU	1901	24.5	39.1	21.7	36.0	74.0	41.8	49.3	67.9	191.1	122.3	212.3
1	3428	TAMIL NADU	1902	67.2	9.8	25.1	21.9	84.7	39.3	55.1	113.8	98.6	282.2	174.9
2	3429	TAMIL NADU	1903	19.3	7.8	1.7	18.2	128.5	58.5	72.6	115.0	210.4	128.1	200.5
3	3430	TAMIL NADU	1904	35.2	0.1	0.7	19.5	121.9	34.9	89.0	40.4	85.7	163.2	23.6
4	3431	TAMIL NADU	1905	6.5	7.5	17.2	64.8	83.7	49.8	39.0	101.8	73.5	250.4	123.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	3537	TAMIL NADU	2011	4.3	11.2	8.0	91.5	33.4	56.0	45.5	128.9	76.0	200.4	230.5
111	3538	TAMIL NADU	2012	3.0	0.1	2.5	35.5	41.9	30.1	46.5	98.0	84.9	235.2	44.5
112	3539	TAMIL NADU	2013	3.9	30.9	30.0	20.3	42.0	54.6	42.7	110.7	113.5	127.9	112.3
113	3540	TAMIL NADU	2014	7.4	6.1	8.1	8.3	139.1	47.8	50.6	117.7	98.9	252.2	110.8
114	3541	TAMIL NADU	2015	8.3	2.3	21.7	108.8	112.4	62.4	43.5	81.6	98.4	132.6	379.8

115 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')
```

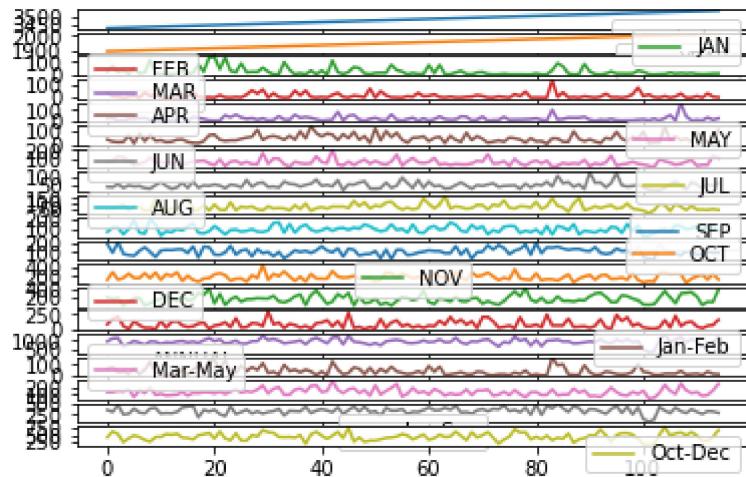
In [5]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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]: df.plot.line(subplots=True)

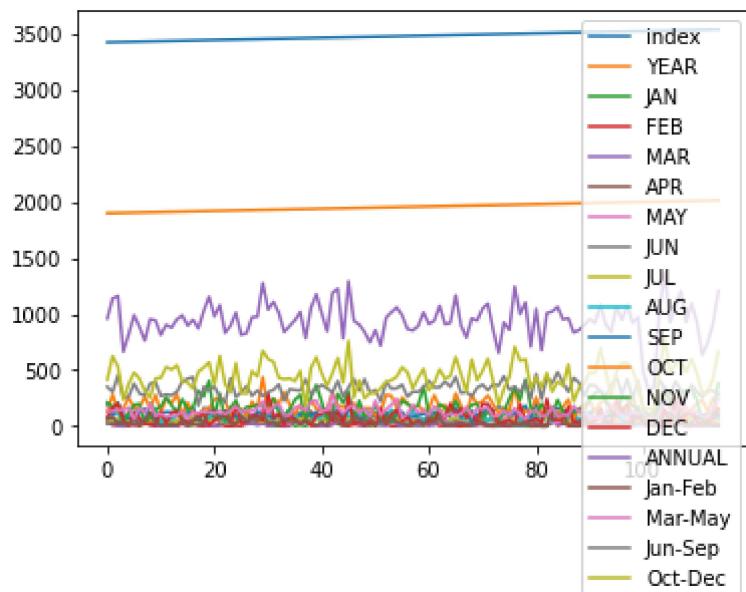
```
Out[6]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



## Line chart

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

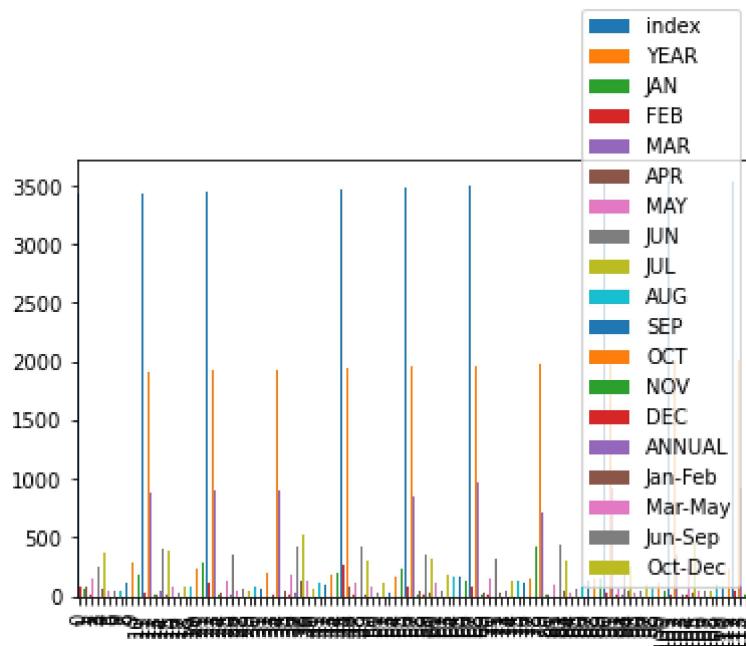
Out[7]: <AxesSubplot:>



## Bar chart

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

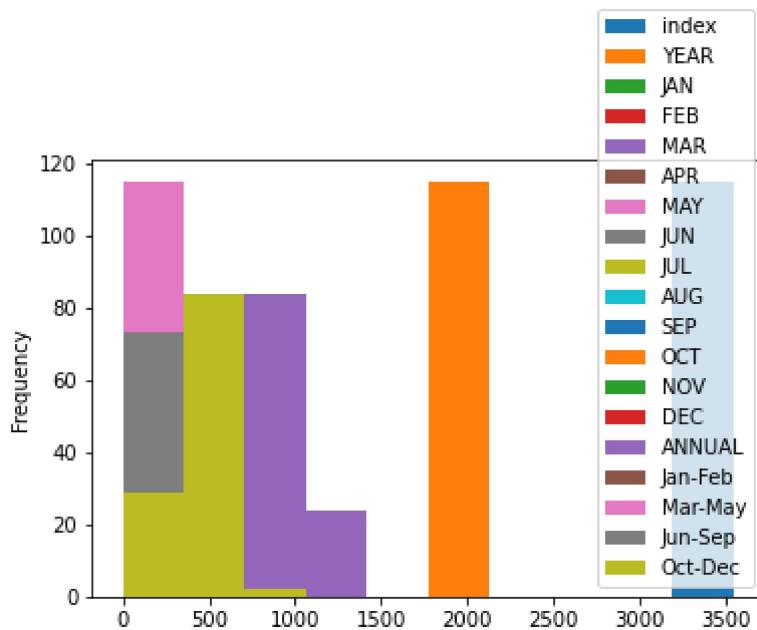
Out[8]: <AxesSubplot:>



## Histogram

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

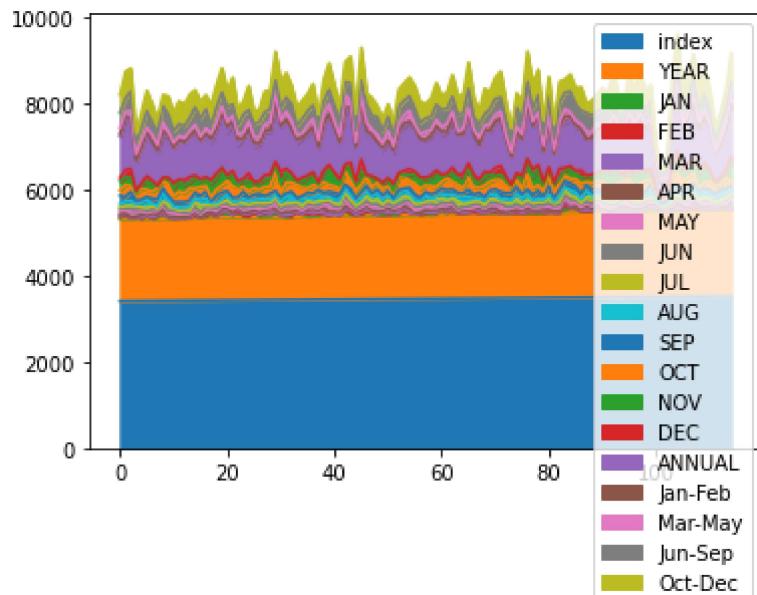
Out[9]: <AxesSubplot:ylabel='Frequency'>



## Area chart

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

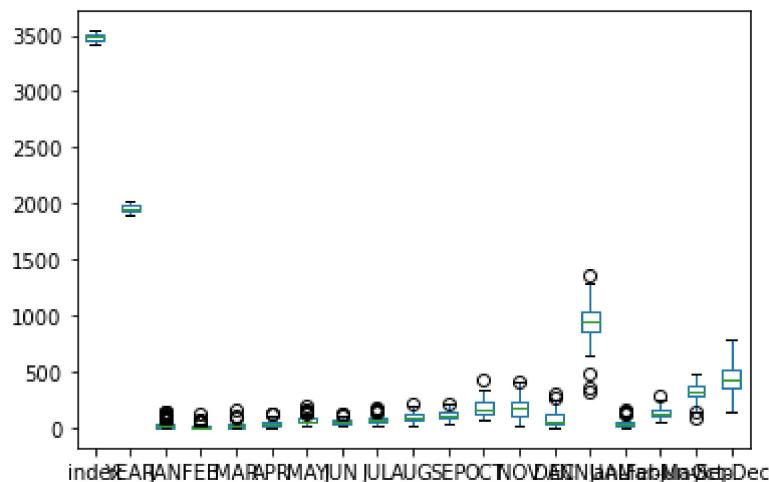
Out[10]: <AxesSubplot:>



## Box chart

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

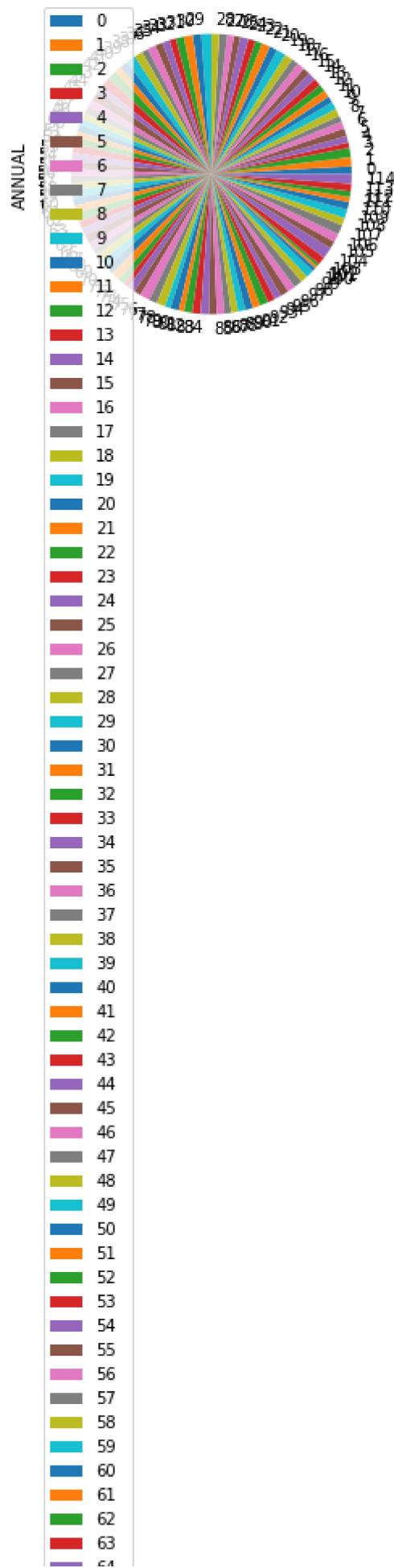
```
Out[11]: <AxesSubplot:>
```

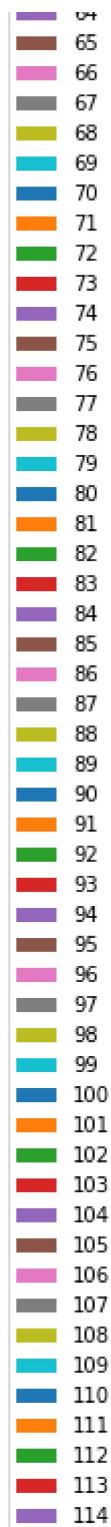


## Pie chart

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

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

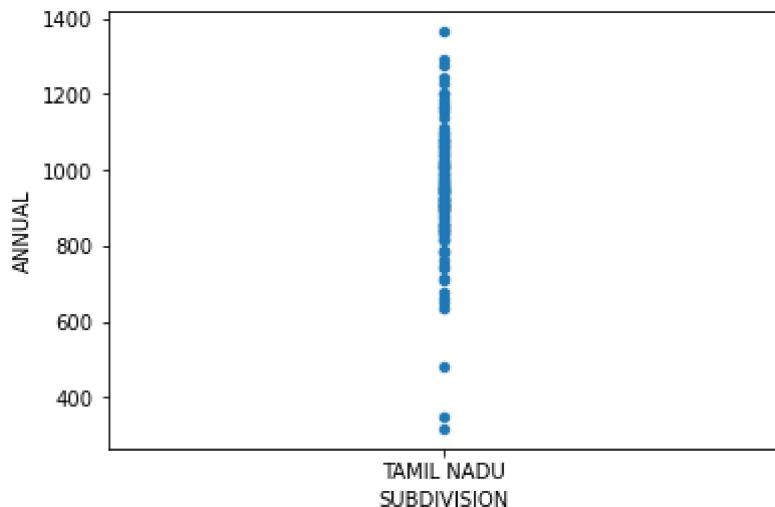




## Scatter chart

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: 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]:

`df.describe()`

Out[15]:

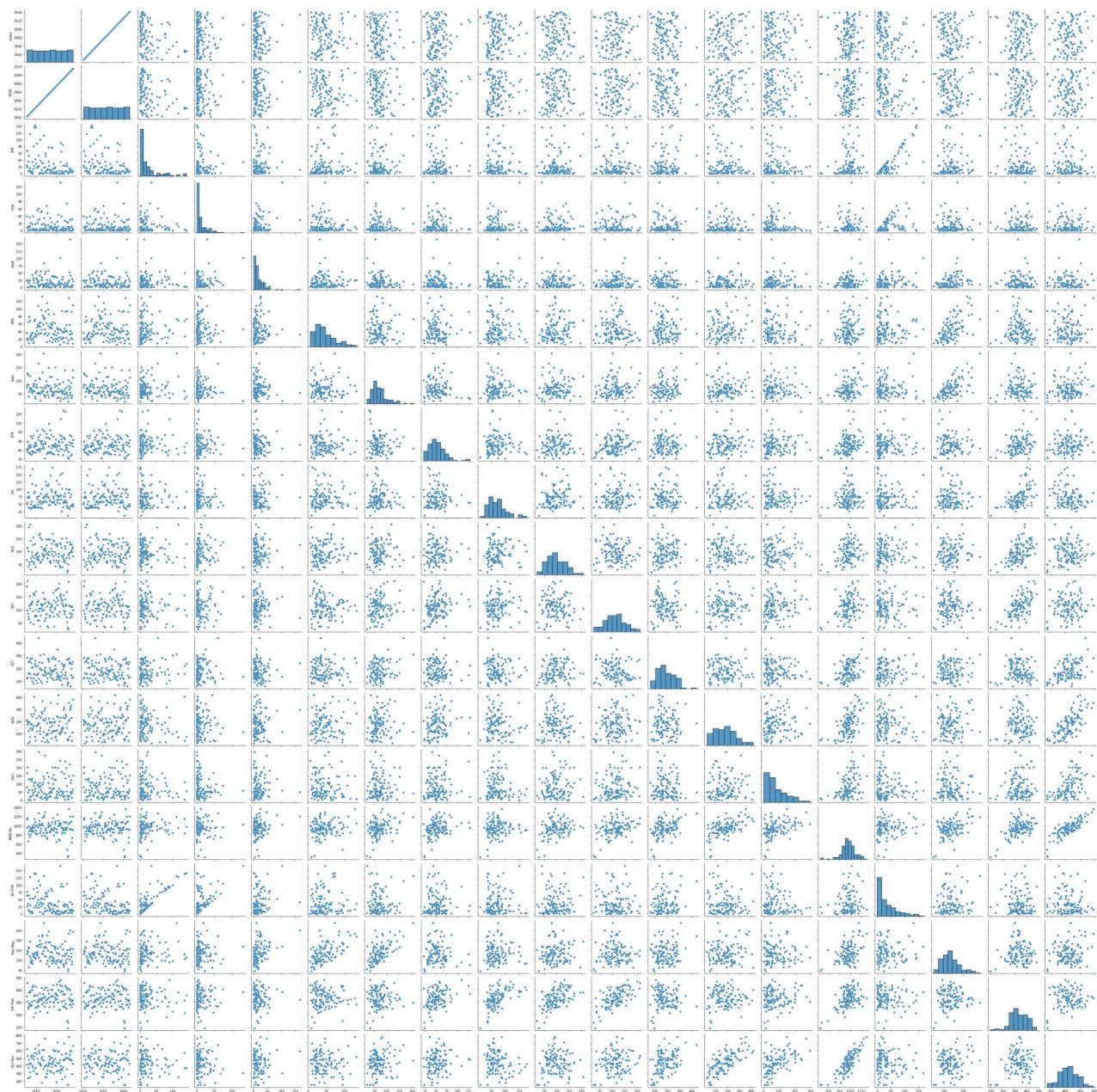
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	3484.000000	1958.000000	23.819130	13.422609	19.475652	44.995652	69.920870	52.056522
<b>std</b>	33.341666	33.341666	32.253728	19.501219	22.389999	27.920223	31.775268	19.136583
<b>min</b>	3427.000000	1901.000000	0.100000	0.000000	0.000000	5.500000	19.800000	21.200000
<b>25%</b>	3455.500000	1929.500000	2.950000	1.200000	5.150000	23.650000	50.050000	38.650000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>50%</b>	3484.000000	1958.000000	10.000000	5.500000	11.900000	37.000000	61.100000	49.800000
<b>75%</b>	3512.500000	1986.500000	29.300000	18.050000	26.700000	59.300000	82.400000	62.800000
<b>max</b>	3541.000000	2015.000000	141.200000	131.300000	164.700000	132.100000	204.400000	128.100000

## EDA AND VISUALIZATION

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

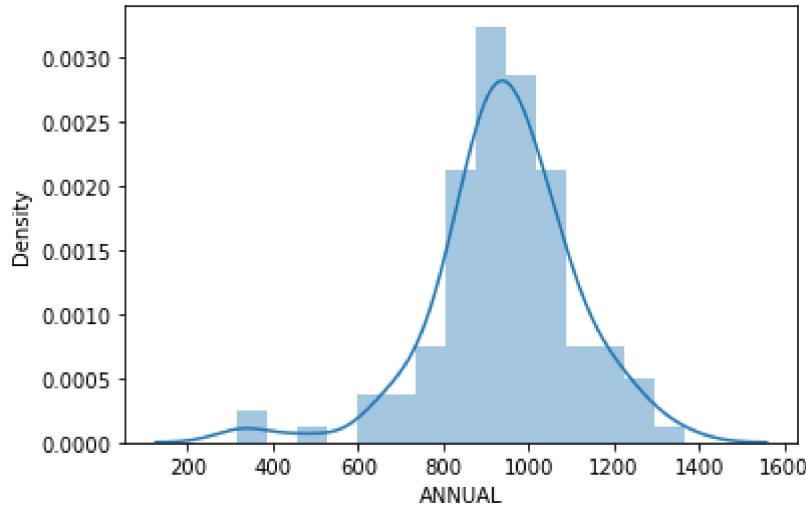
Out[16]: <seaborn.axisgrid.PairGrid at 0x13407644df0>



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) o  
r `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:>
```

