

# 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_saurashtra _ kutch.csv")
df
```

Out[2]:

		index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DECEMBER
0	2392	SAURASHTRA & KUTCH		1901	1.9	0.0	0.1	0.2	3.2	9.1	87.8	62.5	12.0	3.8	0.0	0.0
1	2393	SAURASHTRA & KUTCH		1902	0.1	0.0	0.0	0.5	1.1	14.4	92.9	160.0	123.9	1.5	0.1	0.0
2	2394	SAURASHTRA & KUTCH		1903	0.5	0.0	1.7	0.0	3.1	10.5	337.9	96.1	61.9	11.1	0.0	0.0
3	2395	SAURASHTRA & KUTCH		1904	1.4	5.8	17.5	0.0	0.0	9.5	111.2	9.4	28.9	0.3	1.7	0.0
4	2396	SAURASHTRA & KUTCH		1905	1.5	1.0	0.6	0.4	0.0	6.4	254.5	12.3	12.8	0.4	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	2502	SAURASHTRA & KUTCH		2011	0.0	1.4	0.0	0.0	0.0	26.0	212.7	290.9	210.1	1.2	0.1	0.0
111	2503	SAURASHTRA & KUTCH		2012	0.0	0.0	0.0	0.2	0.1	22.4	34.7	34.5	228.5	2.4	0.0	0.0
112	2504	SAURASHTRA & KUTCH		2013	1.7	0.2	0.1	8.5	0.1	127.7	171.2	83.3	260.2	28.6	0.0	0.0
113	2505	SAURASHTRA & KUTCH		2014	0.3	0.0	0.1	0.5	2.1	17.3	137.7	118.8	99.2	5.2	2.7	0.0
114	2506	SAURASHTRA & KUTCH		2015	0.9	0.0	4.4	2.1	0.8	112.6	226.7	10.6	79.9	3.3	0.3	0.0

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')
```

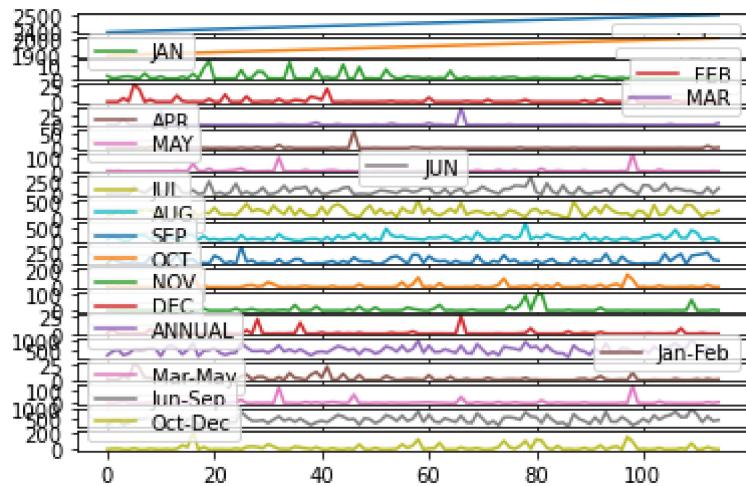
```
In [5]: df.info()
```

```
<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)
```

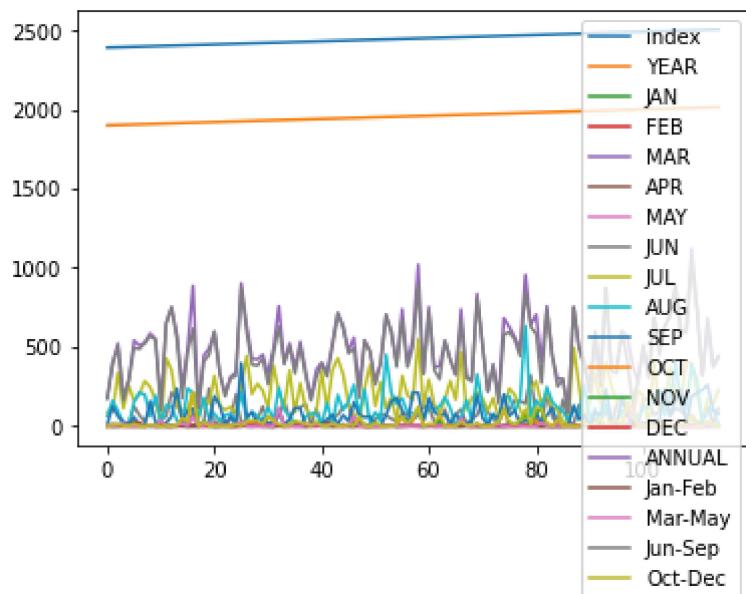
```
Out[6]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



## Line chart

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

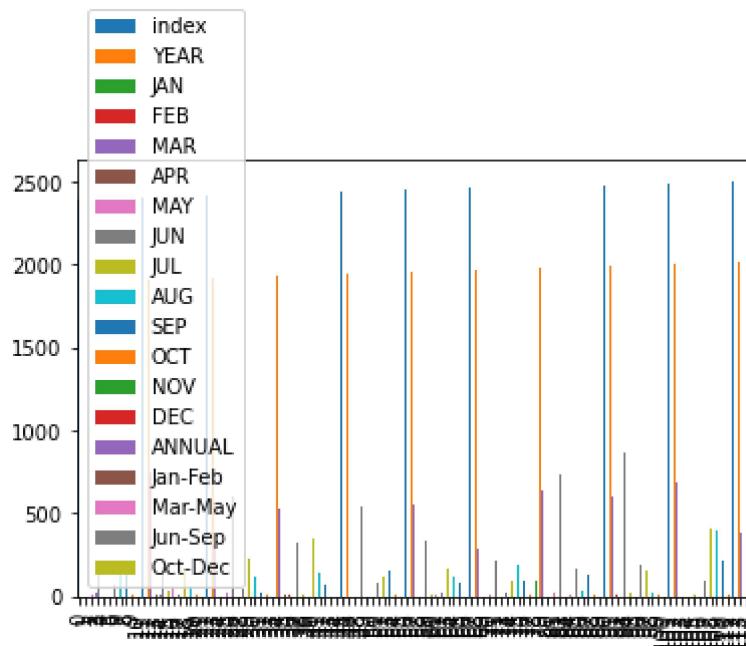
Out[7]: <AxesSubplot:>



## Bar chart

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

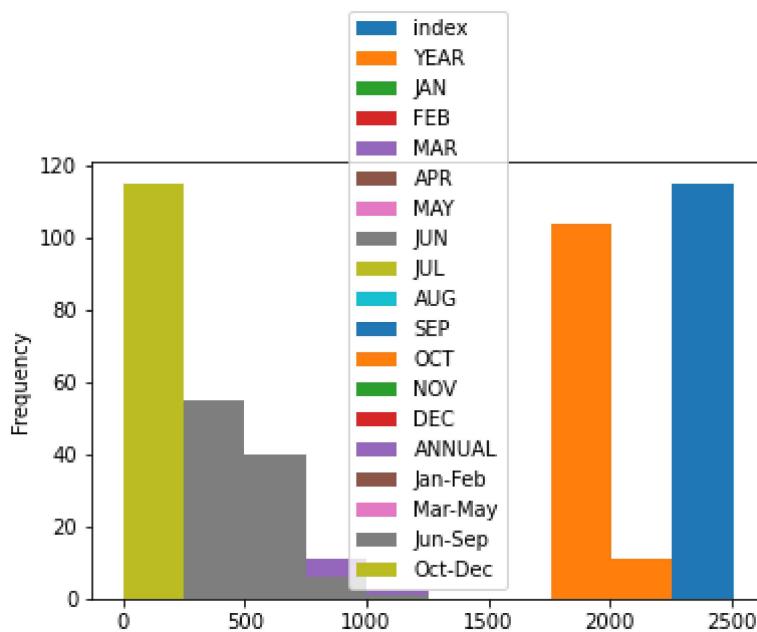
Out[8]: <AxesSubplot:>



## Histogram

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

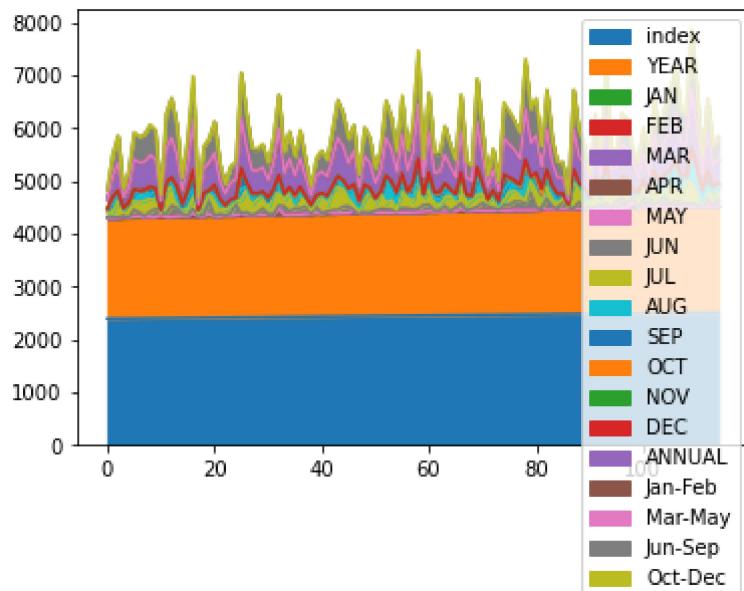
Out[9]: <AxesSubplot:ylabel='Frequency'>



## Area chart

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

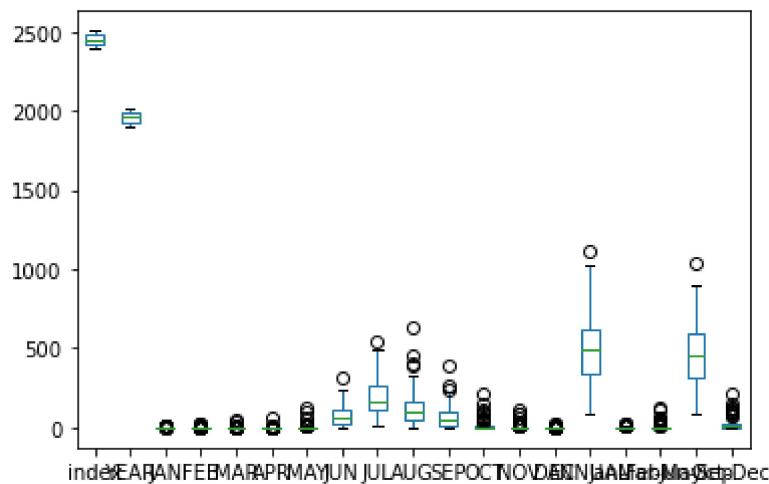
Out[10]: <AxesSubplot:>



## Box chart

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

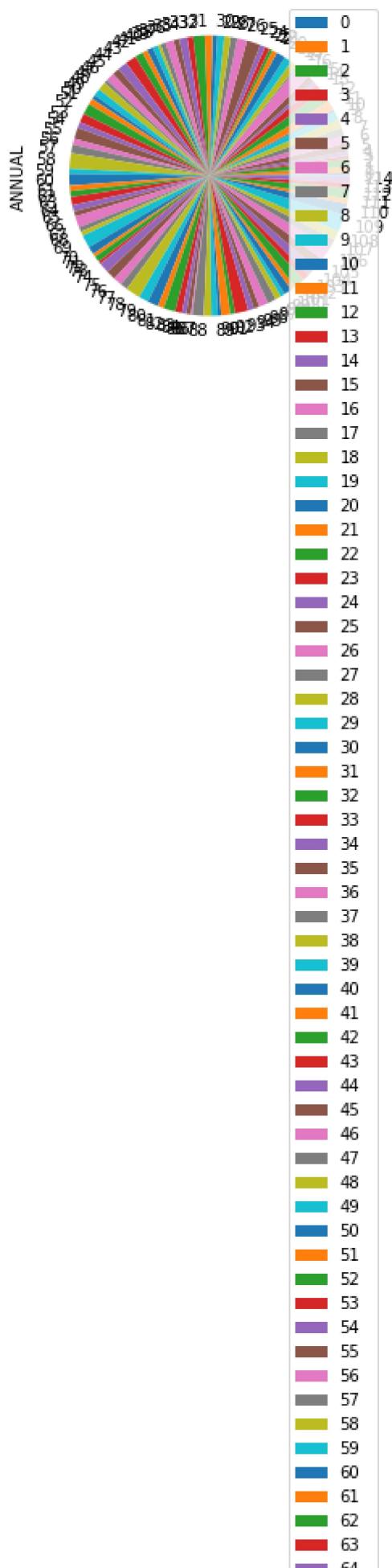
```
Out[11]: <AxesSubplot:>
```

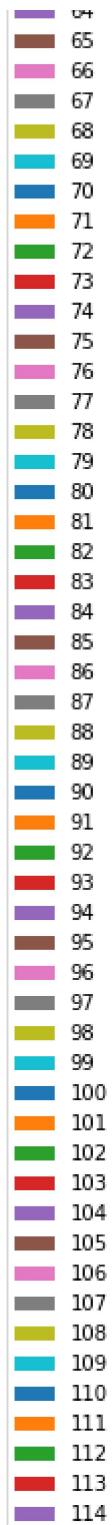


## Pie chart

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

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

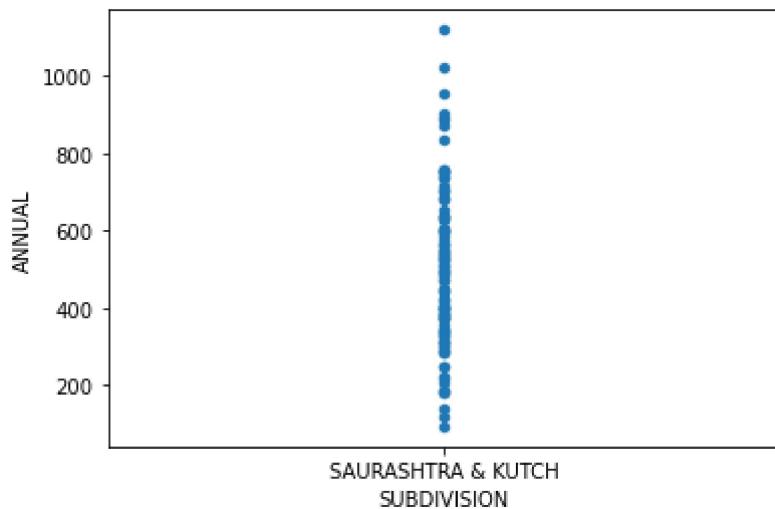




## Scatter chart

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]:

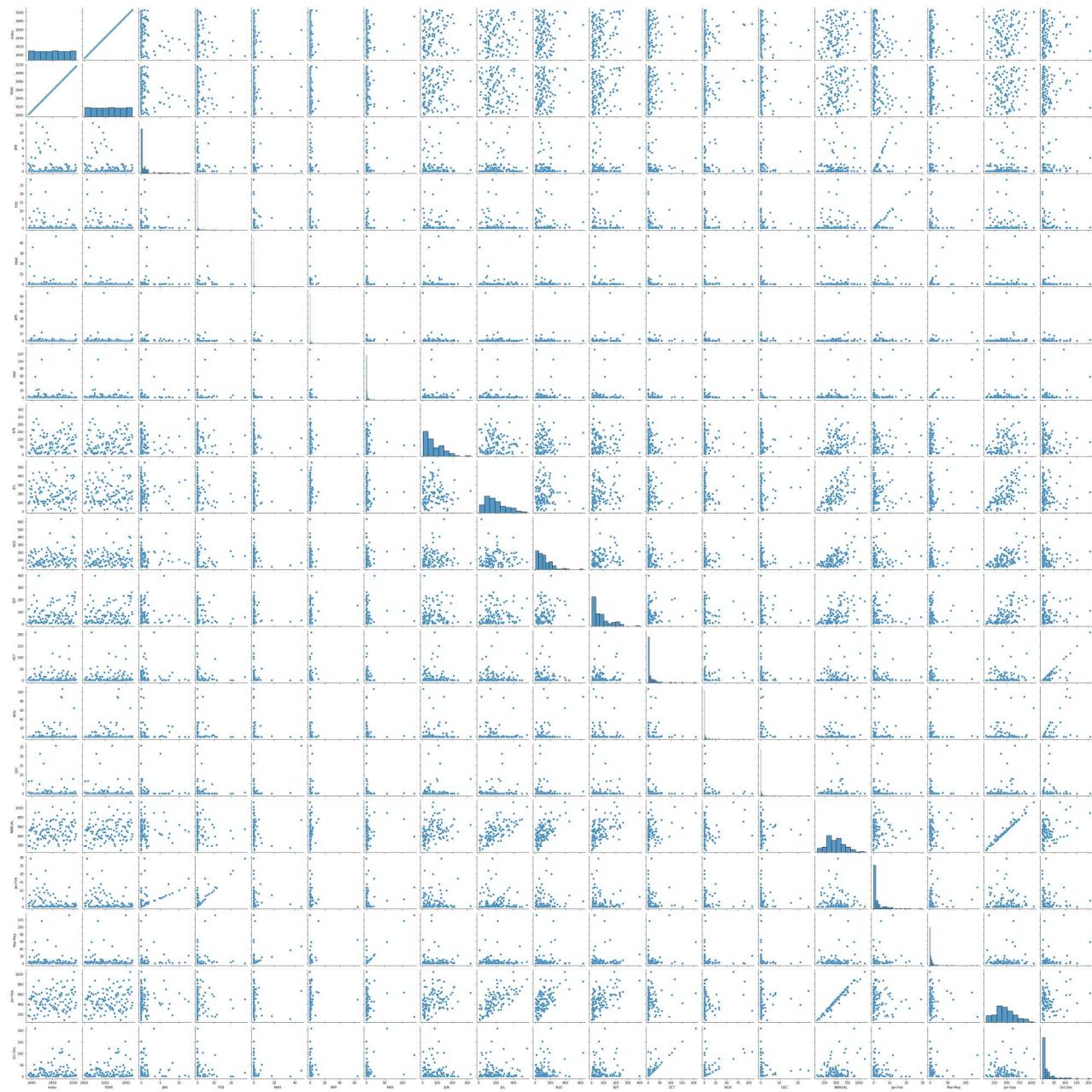
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	2449.000000	1958.000000	1.139130	1.615652	1.296522	1.183478	4.662609	74.371304
<b>std</b>	33.341666	33.341666	2.374709	4.270576	5.691544	6.158847	16.587231	63.062739
<b>min</b>	2392.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.200000
<b>25%</b>	2420.500000	1929.500000	0.000000	0.000000	0.000000	0.000000	0.000000	19.150000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>50%</b>	2449.000000	1958.000000	0.200000	0.000000	0.000000	0.000000	0.500000	62.100000
<b>75%</b>	2477.500000	1986.500000	1.000000	0.550000	0.400000	0.500000	2.700000	114.250000
<b>max</b>	2506.000000	2015.000000	12.500000	28.200000	46.200000	64.400000	131.900000	321.800000

## EDA AND VISUALIZATION

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

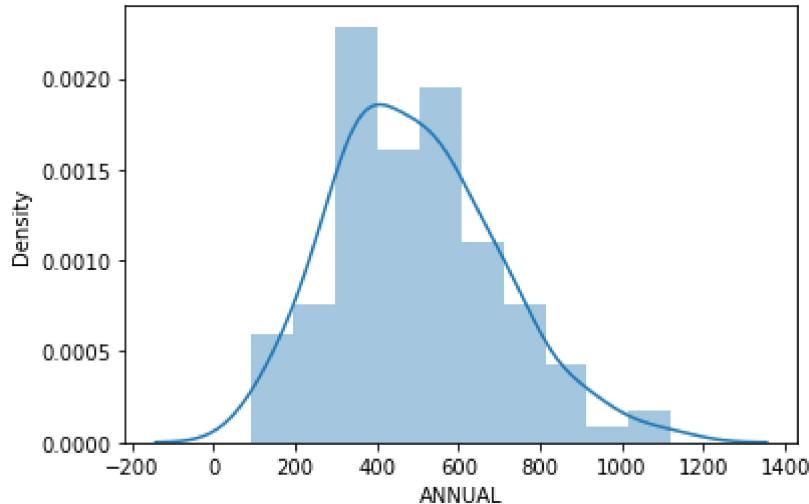
Out[16]: <seaborn.axisgrid.PairGrid at 0x1407f0297c0>



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:>
```

