

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_chhattisgarh.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DECEMBER
0	2967	CHHATTISGARH	1901	48.9	116.5	27.8	5.5	18.4	101.6	381.0	476.7	182.8	27.3	0.	10.2
1	2968	CHHATTISGARH	1902	0.6	6.5	0.4	13.9	10.3	37.2	403.8	236.6	198.1	4.7	8.	10.2
2	2969	CHHATTISGARH	1903	6.2	13.9	0.4	6.8	51.1	110.7	365.9	396.0	212.0	168.0	0.	10.2
3	2970	CHHATTISGARH	1904	0.0	8.6	32.3	0.2	77.5	369.5	303.6	483.6	86.8	129.3	1.	10.2
4	2971	CHHATTISGARH	1905	50.3	22.6	19.0	24.6	31.8	40.4	443.7	270.8	338.8	8.9	0.	10.2
...
110	3077	CHHATTISGARH	2011	0.3	11.5	2.6	35.0	16.8	183.5	272.6	379.8	382.2	15.5	0.	10.2
111	3078	CHHATTISGARH	2012	36.6	4.8	1.1	14.9	9.4	147.3	430.6	442.2	245.3	19.8	20.	10.2
112	3079	CHHATTISGARH	2013	2.8	19.7	4.9	45.8	5.7	263.6	418.8	336.6	140.9	180.9	0.	10.2
113	3080	CHHATTISGARH	2014	2.3	29.0	21.4	17.3	25.0	104.9	416.7	327.7	252.7	77.9	2.	10.2
114	3081	CHHATTISGARH	2015	15.8	1.2	21.2	37.0	13.0	257.6	248.6	286.6	216.9	17.7	0.	10.2

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')
```

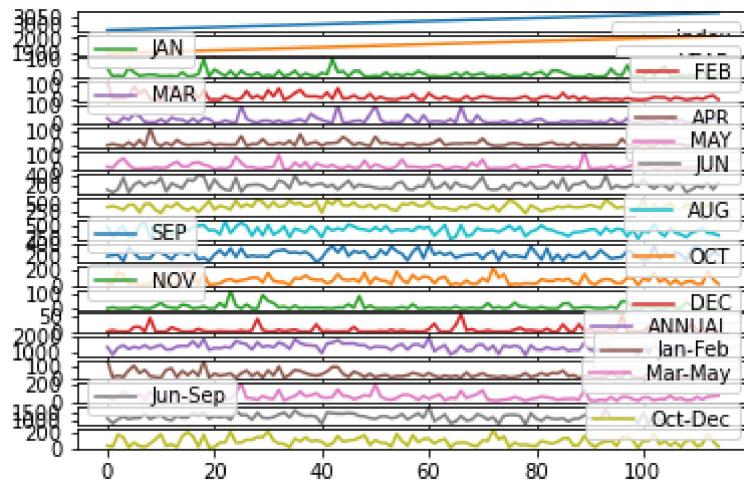
```
In [5]: df.info()
```

```
<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)
```

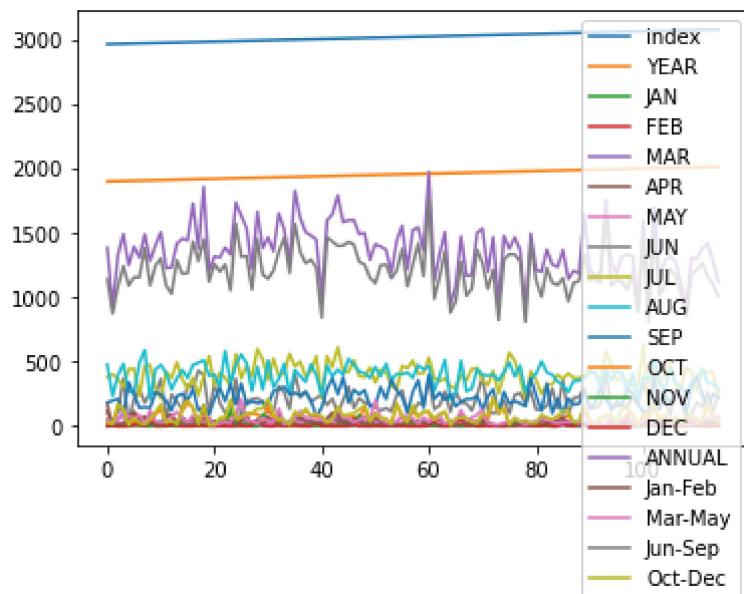
```
Out[6]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



Line chart

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

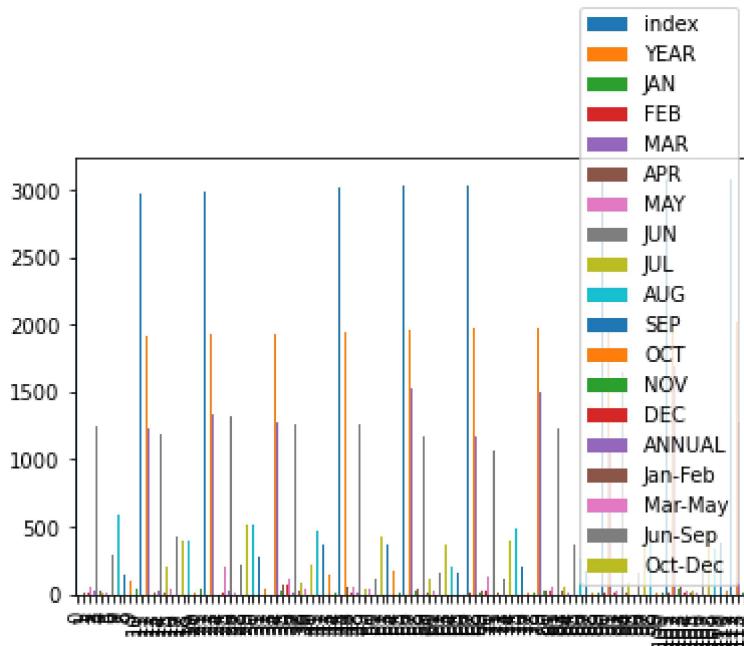
Out[7]: <AxesSubplot:>



Bar chart

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

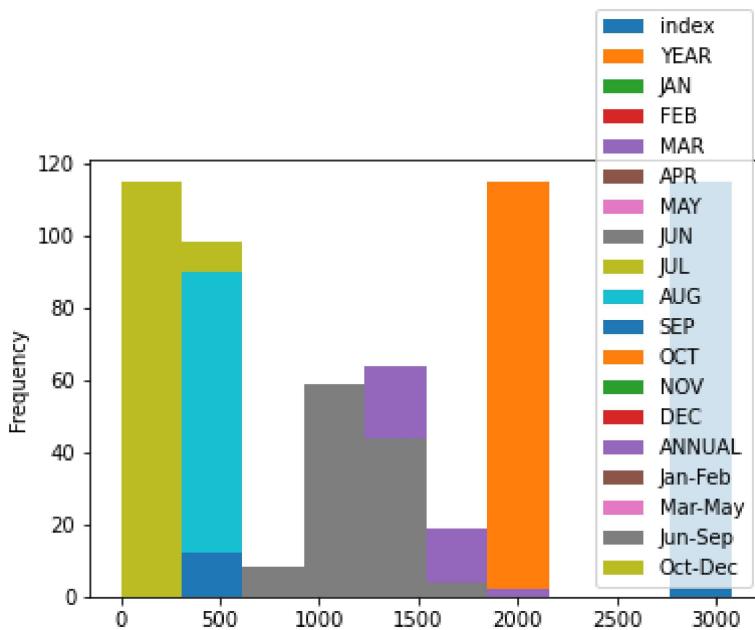
Out[8]: <AxesSubplot:>



Histogram

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

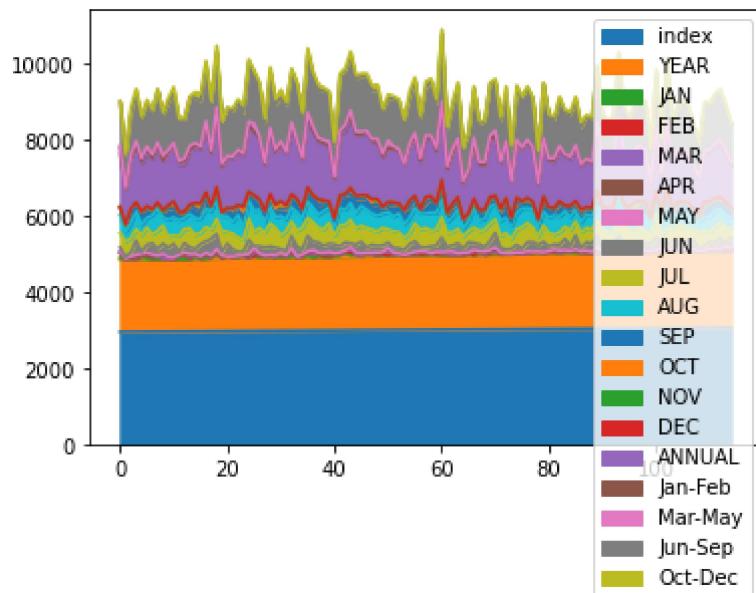
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area chart

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

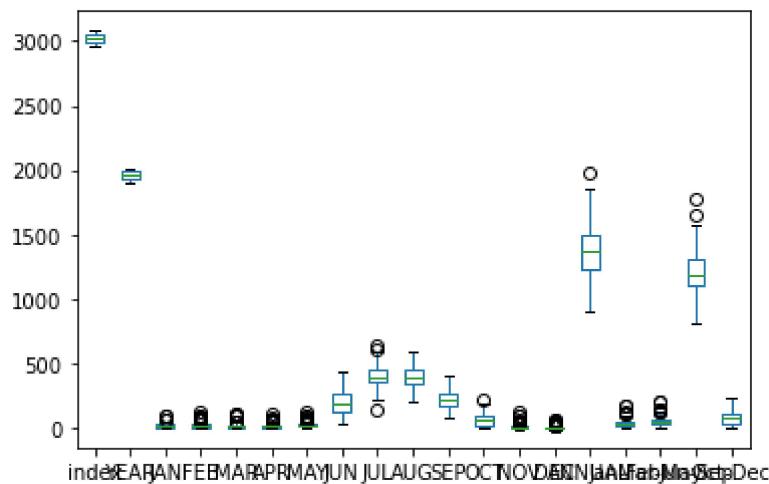
Out[10]: <AxesSubplot:>



Box chart

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

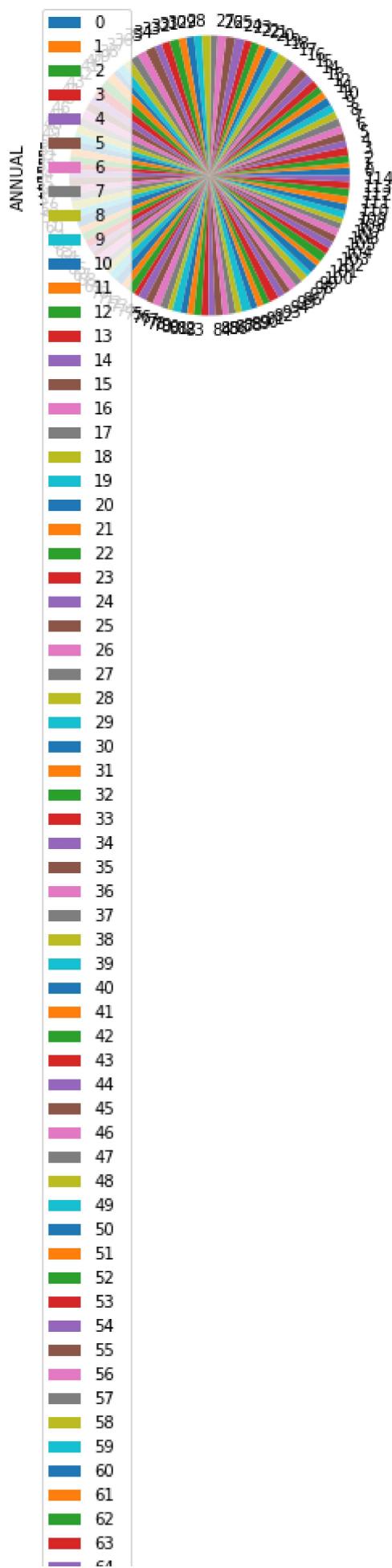
```
Out[11]: <AxesSubplot:>
```

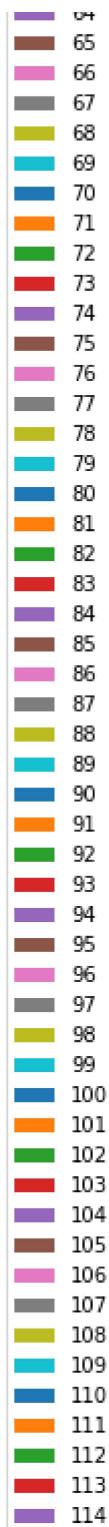


Pie chart

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

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

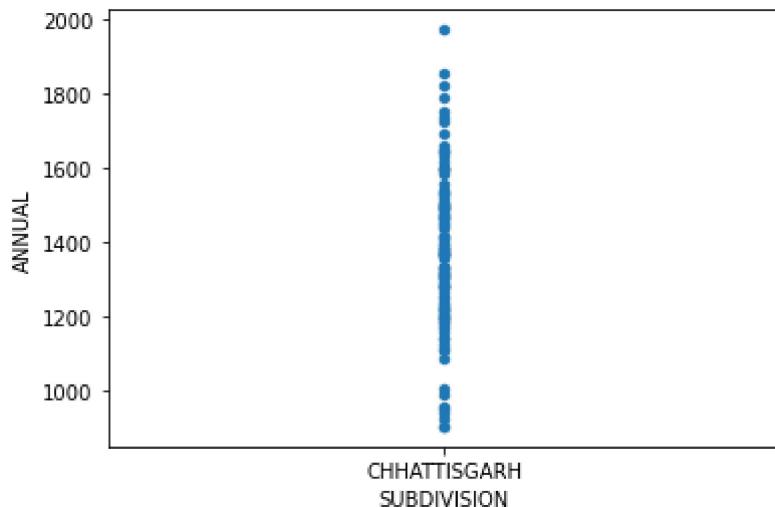




Scatter chart

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]:

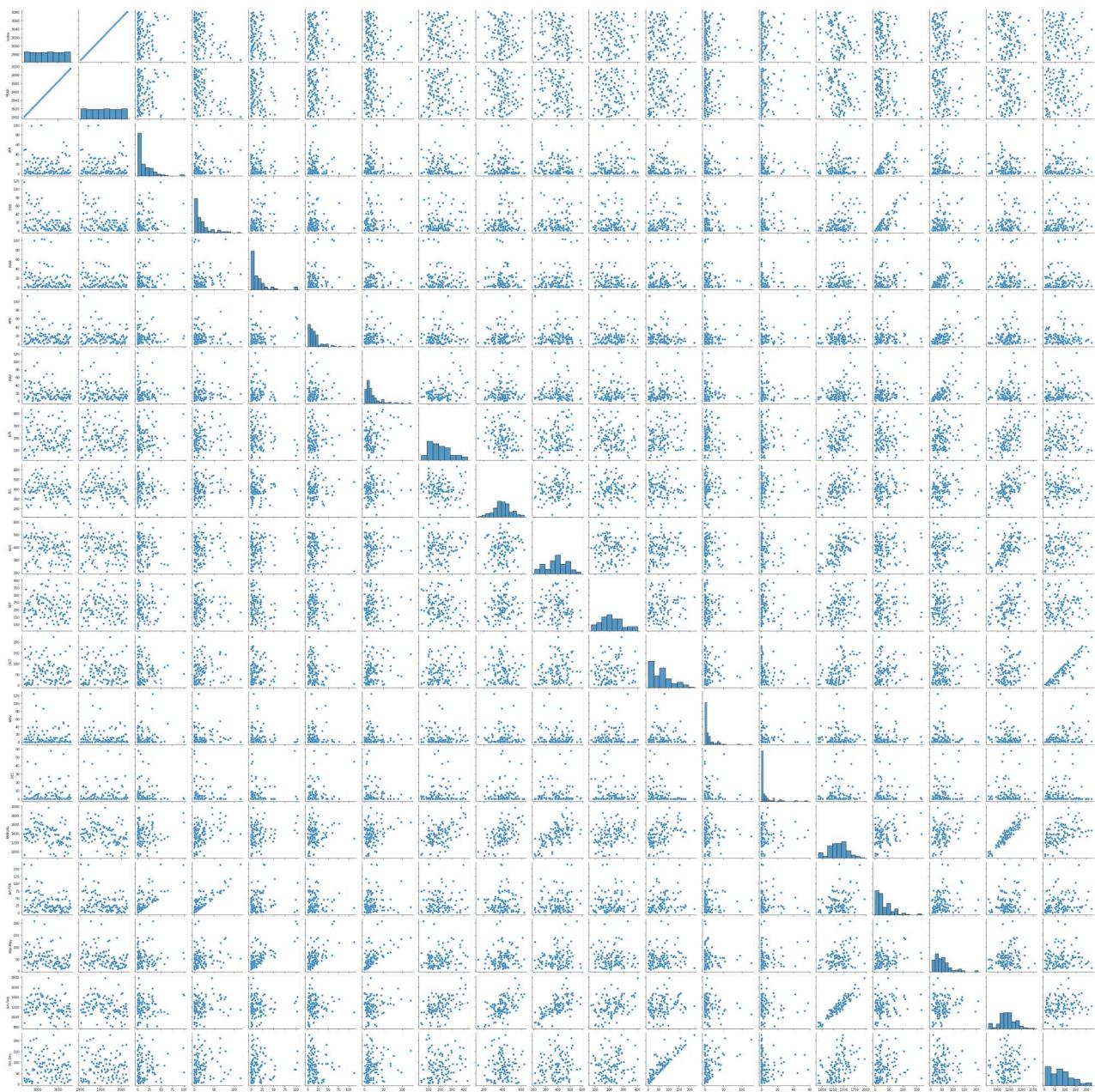
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	3024.000000	1958.000000	14.206957	19.25913	15.266957	16.773043	21.048696	198.266087
std	33.341666	33.341666	18.315772	23.03757	20.653922	17.125078	20.630349	90.108430
min	2967.000000	1901.000000	0.000000	0.00000	0.000000	0.000000	0.000000	37.200000
25%	2995.500000	1929.500000	1.450000	4.05000	2.200000	5.350000	8.700000	128.100000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
50%	3024.000000	1958.000000	6.500000	11.000000	7.900000	11.900000	15.500000	184.300000
75%	3052.500000	1986.500000	22.100000	24.850000	21.300000	20.800000	25.150000	257.300000
max	3081.000000	2015.000000	99.500000	116.500000	102.900000	112.800000	122.300000	431.200000

EDA AND VISUALIZATION

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

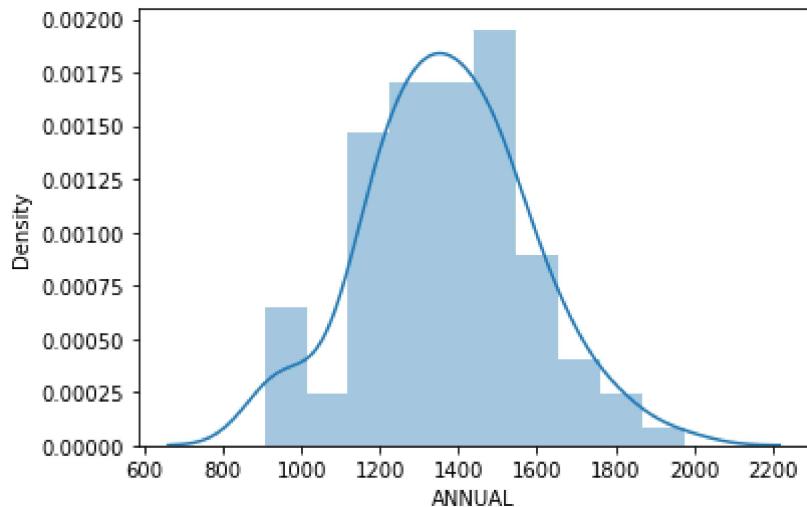
Out[16]: <seaborn.axisgrid.PairGrid at 0x261dd9c0f70>



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:>

