

# Importing Libraries

```
In [1]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns
```

# Importing Datasets

```
In [2]: df=pd.read_csv(r'C:\Users\user\Downloads\Rainfall\JHARKHAND.csv')  
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	782	JHARKHAND	1901	92.7	66.6	11.1	18.4	33.5	70.9	269.4	415.1	248.0	37.3
1	783	JHARKHAND	1902	4.2	7.7	13.2	28.5	59.8	89.9	456.1	204.9	306.6	17.6
2	784	JHARKHAND	1903	25.1	19.5	10.7	32.8	56.4	142.1	206.1	280.8	190.2	210.1
3	785	JHARKHAND	1904	2.5	17.0	38.1	9.1	116.1	308.9	494.1	336.1	125.6	30.6
4	786	JHARKHAND	1905	38.4	53.3	61.6	32.9	66.2	41.5	420.3	293.7	322.8	21.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	892	JHARKHAND	2011	3.3	2.5	6.4	25.4	55.0	349.0	181.8	403.2	324.6	23.3
111	893	JHARKHAND	2012	34.6	10.3	1.5	9.6	6.6	121.1	287.2	282.4	217.6	37.8
112	894	JHARKHAND	2013	1.1	17.9	1.6	22.3	85.0	181.5	211.1	278.1	173.8	281.1
113	895	JHARKHAND	2014	9.9	47.5	22.9	1.9	98.2	139.7	321.3	290.9	178.2	44.9
114	896	JHARKHAND	2015	12.2	2.6	21.6	55.5	25.5	183.3	429.7	240.7	85.1	22.7

115 rows × 20 columns

# Data Cleaning and Data Preprocessing

In [3]: df=df.dropna()  
df

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	782	JHARKHAND	1901	92.7	66.6	11.1	18.4	33.5	70.9	269.4	415.1	248.0	37.3
1	783	JHARKHAND	1902	4.2	7.7	13.2	28.5	59.8	89.9	456.1	204.9	306.6	17.6
2	784	JHARKHAND	1903	25.1	19.5	10.7	32.8	56.4	142.1	206.1	280.8	190.2	210.1
3	785	JHARKHAND	1904	2.5	17.0	38.1	9.1	116.1	308.9	494.1	336.1	125.6	30.6
4	786	JHARKHAND	1905	38.4	53.3	61.6	32.9	66.2	41.5	420.3	293.7	322.8	21.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	892	JHARKHAND	2011	3.3	2.5	6.4	25.4	55.0	349.0	181.8	403.2	324.6	23.3
111	893	JHARKHAND	2012	34.6	10.3	1.5	9.6	6.6	121.1	287.2	282.4	217.6	37.8
112	894	JHARKHAND	2013	1.1	17.9	1.6	22.3	85.0	181.5	211.1	278.1	173.8	281.1
113	895	JHARKHAND	2014	9.9	47.5	22.9	1.9	98.2	139.7	321.3	290.9	178.2	44.9
114	896	JHARKHAND	2015	12.2	2.6	21.6	55.5	25.5	183.3	429.7	240.7	85.1	22.7

115 rows × 20 columns



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')

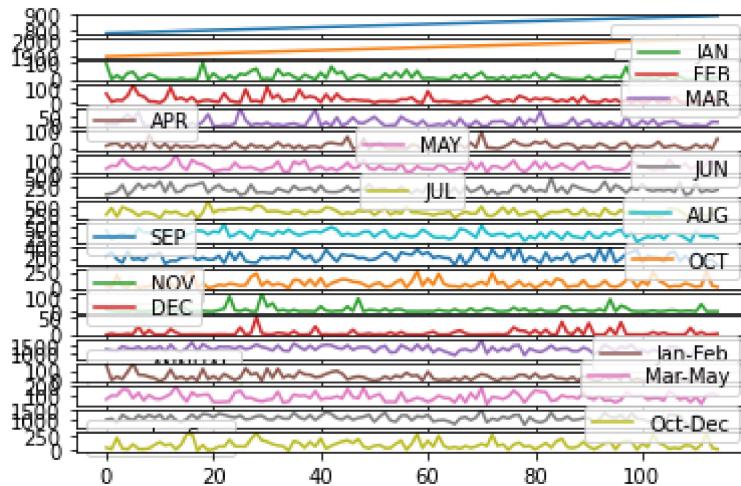
In [5]: df.info()

```
<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

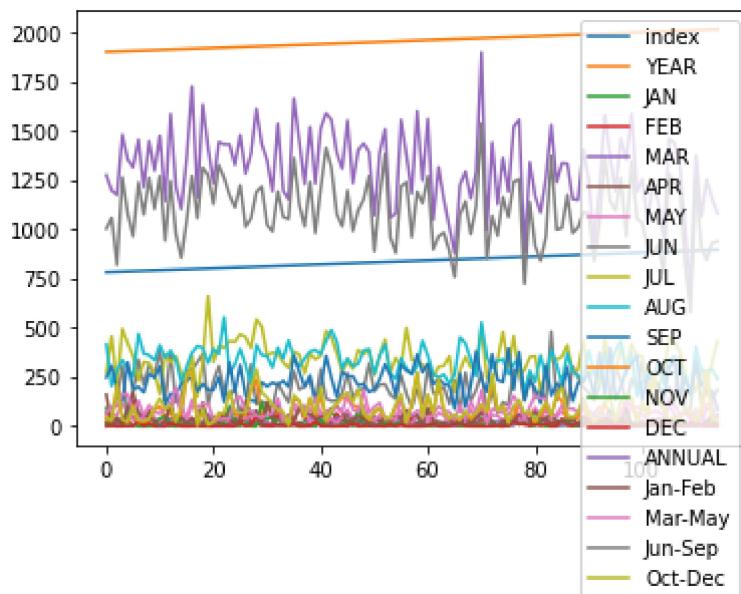
```
In [6]: df.plot.line(subplots=True)
```

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



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

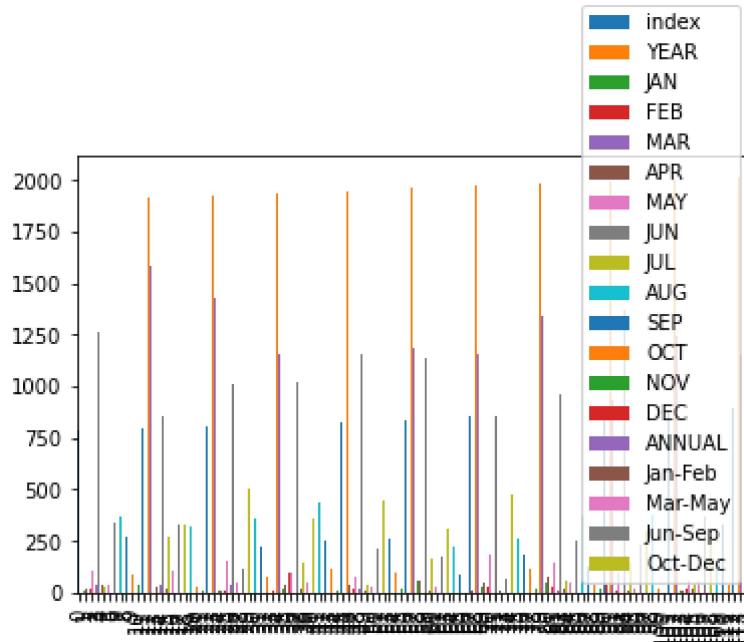
```
Out[7]: <AxesSubplot:>
```



## Bar Chart

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

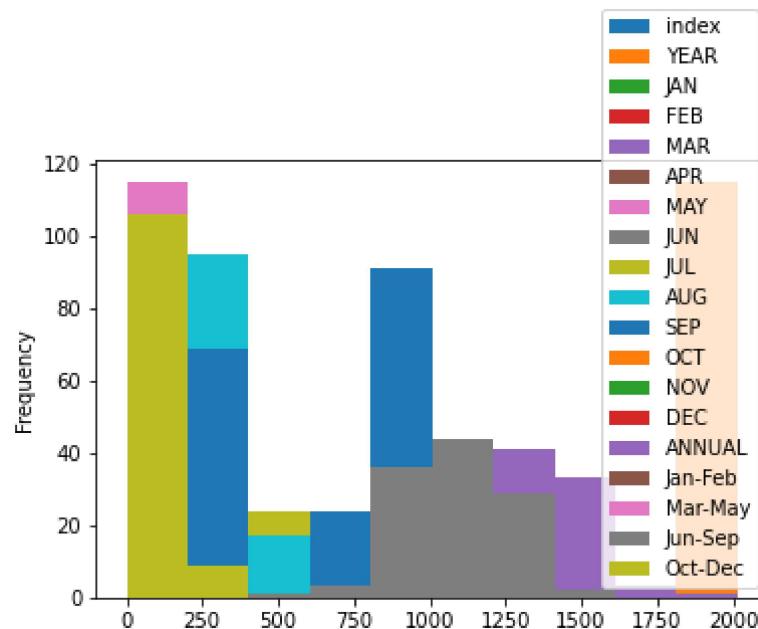
Out[8]: <AxesSubplot:>



## Histogram

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

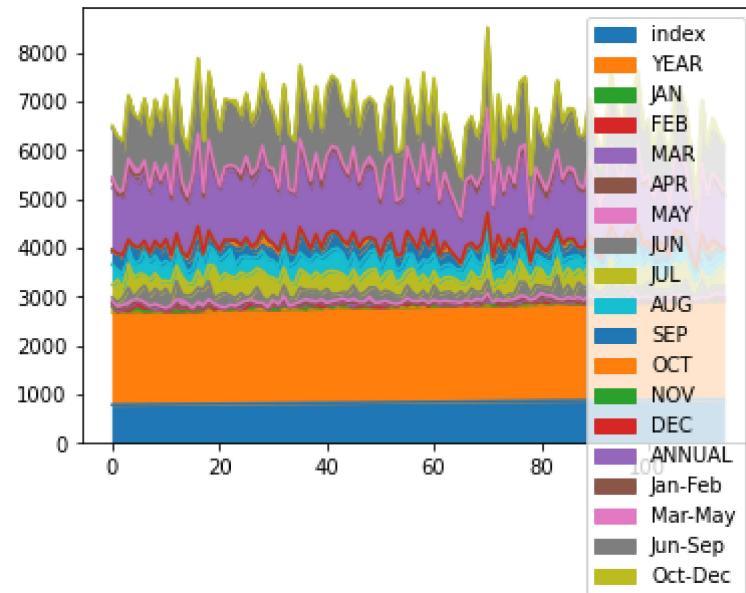
Out[9]: <AxesSubplot:ylabel='Frequency'>



## Area Chart

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

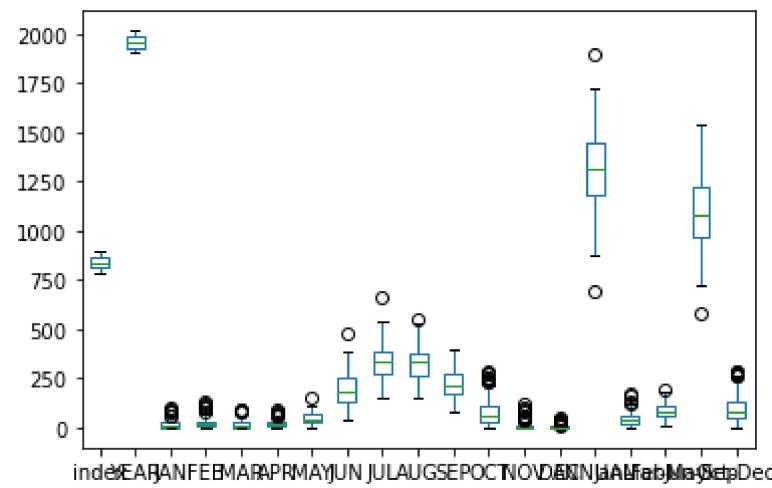
```
Out[10]: <AxesSubplot:>
```



## Box Chart

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

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

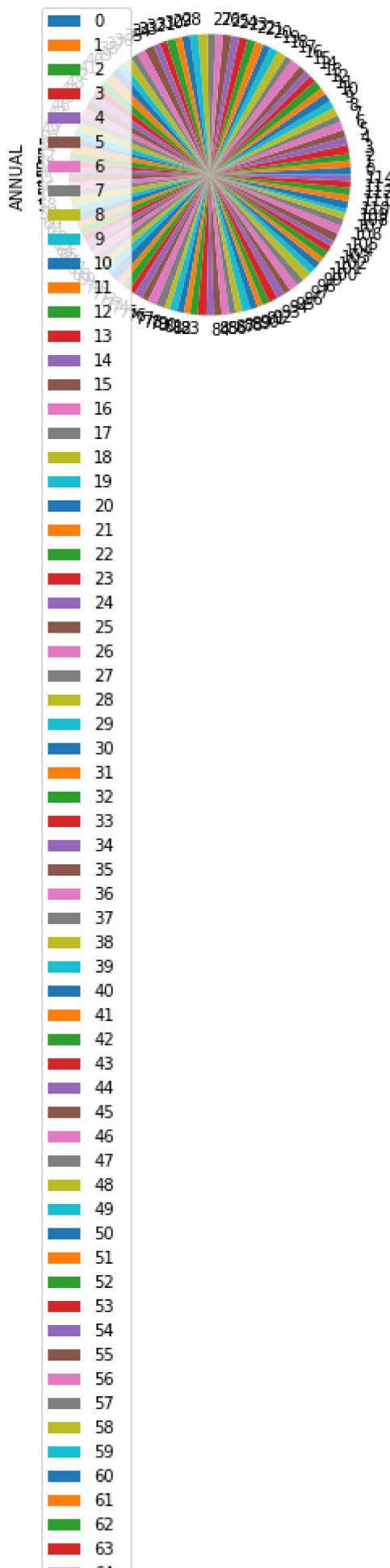


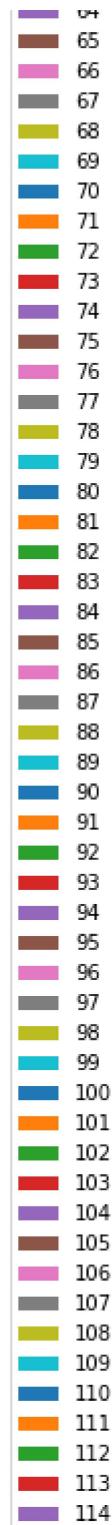
## Pie Chart

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

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



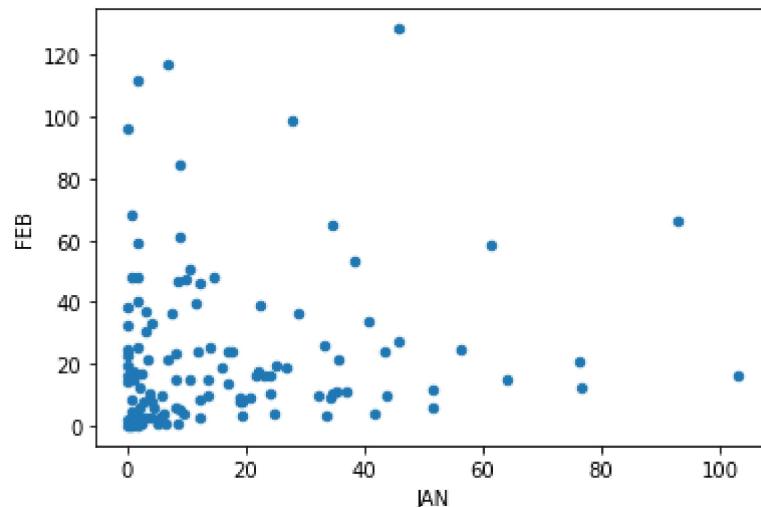




# Scatter Plot

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

```
Out[13]: <AxesSubplot:xlabel='JAN', ylabel='FEB'>
```



```
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()

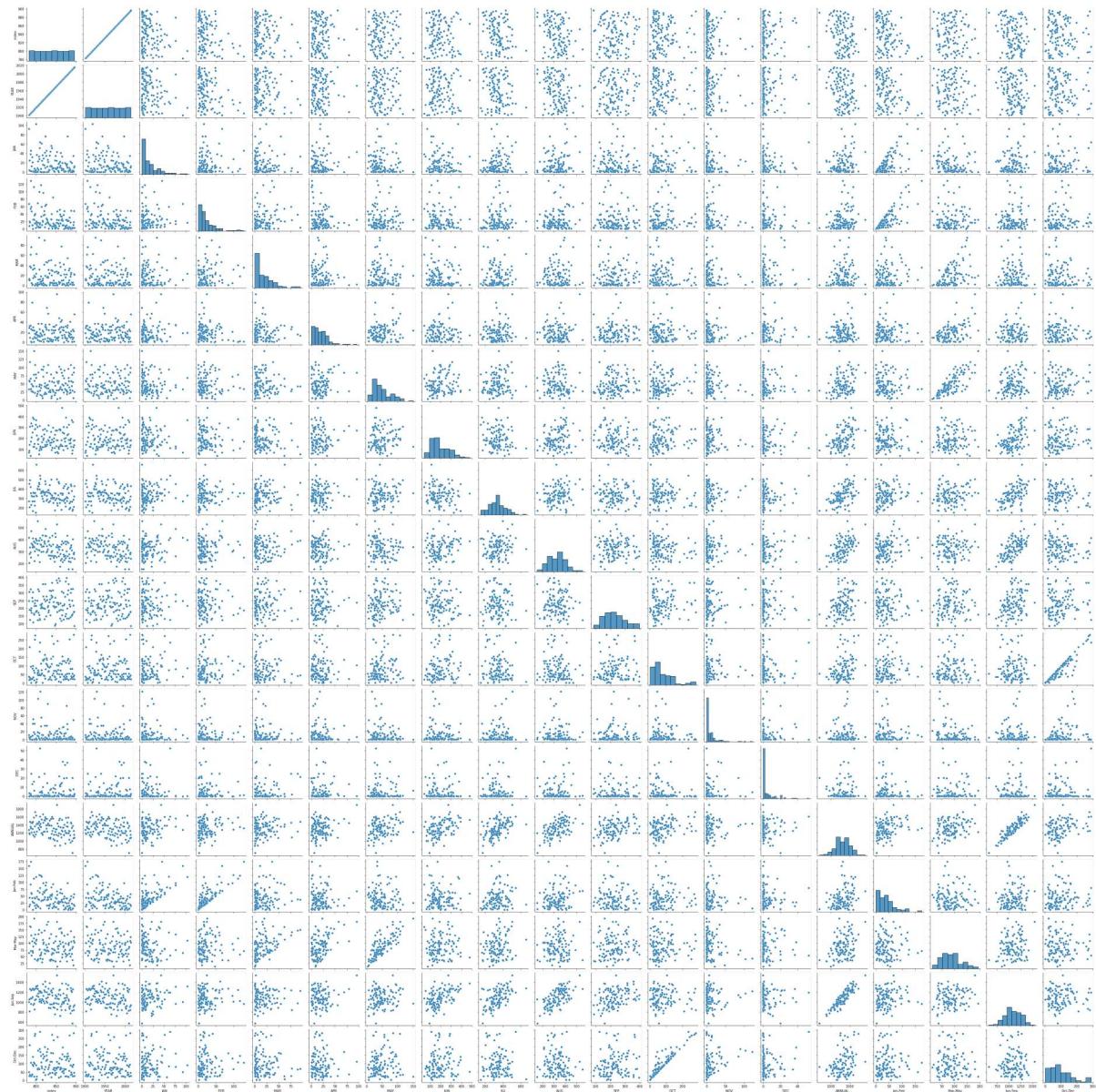
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115
<b>mean</b>	839.000000	1958.000000	17.621739	24.186087	18.423478	19.366957	48.317391	194
<b>std</b>	33.341666	33.341666	20.491016	25.779639	19.867899	16.149628	28.652462	87
<b>min</b>	782.000000	1901.000000	0.000000	0.100000	0.000000	0.100000	3.200000	41
<b>25%</b>	810.500000	1929.500000	2.400000	7.750000	3.350000	8.200000	26.950000	128
<b>50%</b>	839.000000	1958.000000	9.500000	16.200000	11.100000	15.700000	42.300000	183
<b>75%</b>	867.500000	1986.500000	24.900000	32.950000	27.300000	26.350000	66.300000	257
<b>max</b>	896.000000	2015.000000	102.900000	128.500000	95.000000	95.600000	150.100000	479

## EDA And Visualization

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

```
Out[16]: <seaborn.axisgrid.PairGrid at 0x24bd1124d90>
```

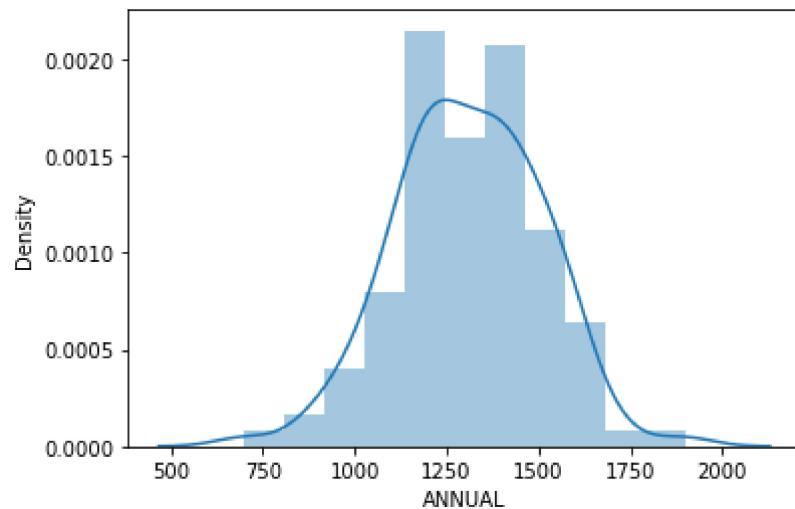


In [17]: `sns.distplot(df['ANNUAL'])`

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:>

