20104016

DEENA

Importing Libraries

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

Importing Datasets

In [2]:
```python
df=pd.read_csv("rainfall_vidarbha.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	2852	VIDARBHA	1901	36.8	39.9	30.9	26.1	7.3	129.7	295.3	368.8	123.4	35.2
1	2853	VIDARBHA	1902	1.6	0.1	0.0	6.5	4.1	38.0	270.7	204.7	150.9	29.6
2	2854	VIDARBHA	1903	5.2	4.0	0.1	2.5	37.8	121.2	475.5	325.5	154.8	100.8
3	2855	VIDARBHA	1904	4.3	2.4	12.9	0.2	14.8	148.9	158.3	151.8	196.9	61.7
4	2856	VIDARBHA	1905	7.3	12.7	12.4	16.2	14.0	81.0	254.5	216.3	321.3	6.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	2962	VIDARBHA	2011	0.0	1.2	0.1	7.7	0.6	137.9	247.1	302.8	191.0	4.7
111	2963	VIDARBHA	2012	3.1	0.1	0.0	0.6	0.2	125.5	370.5	316.2	249.4	34.9
112	2964	VIDARBHA	2013	6.6	13.0	3.8	2.8	0.5	366.7	535.5	326.1	131.7	133.5
113	2965	VIDARBHA	2014	1.2	18.3	49.6	2.6	4.0	63.3	337.6	191.7	224.9	17.3
114	2966	VIDARBHA	2015	26.3	4.7	66.3	28.1	12.8	254.6	137.2	288.9	167.5	7.0

115 rows × 20 columns

Data Cleaning and Data Preprocessing

In [3]:
```python
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'],
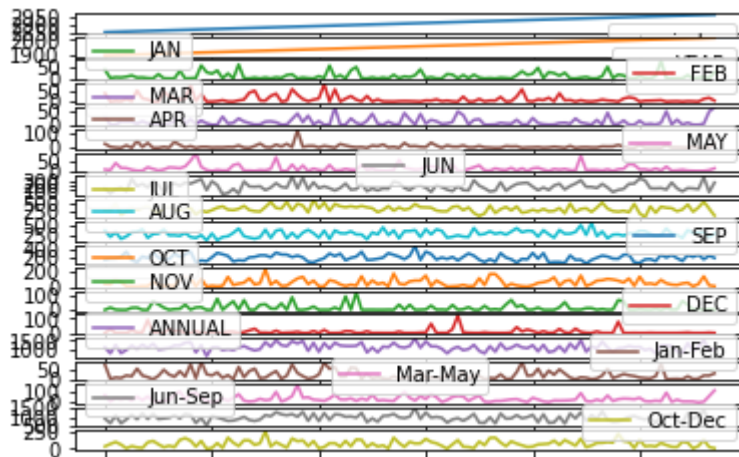   dtype='object')

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:>, <AxesSubplot:>], dtype=object)
```
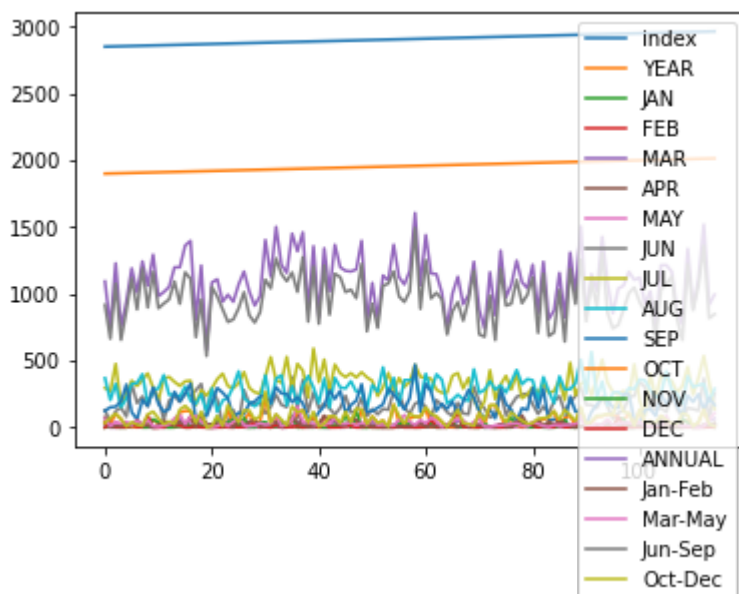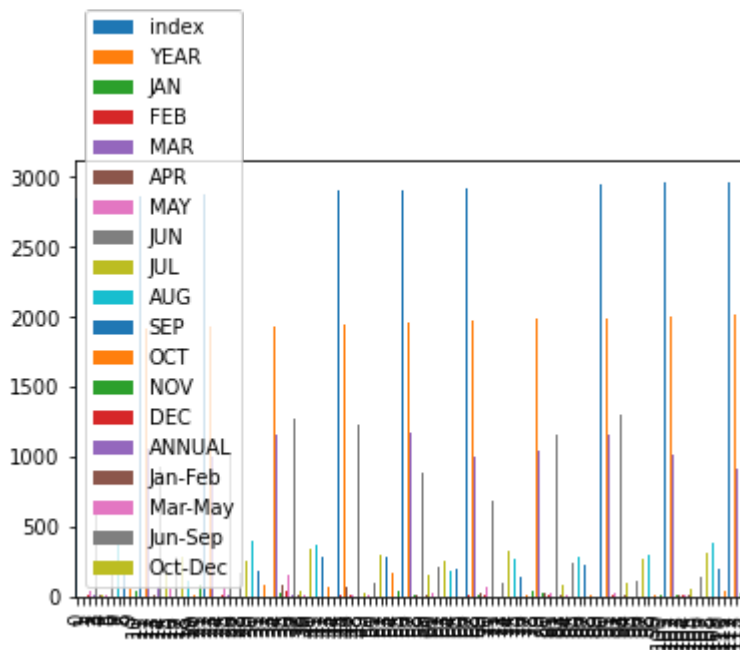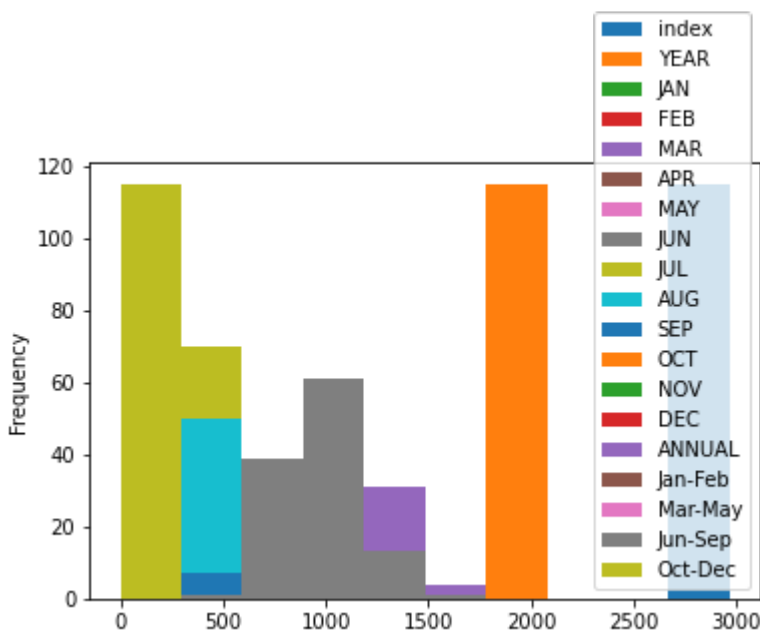
Line chart

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

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

Bar chart

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

Out[8]: <AxesSubplot:>

Histogram

In [9]: df.plot.hist()
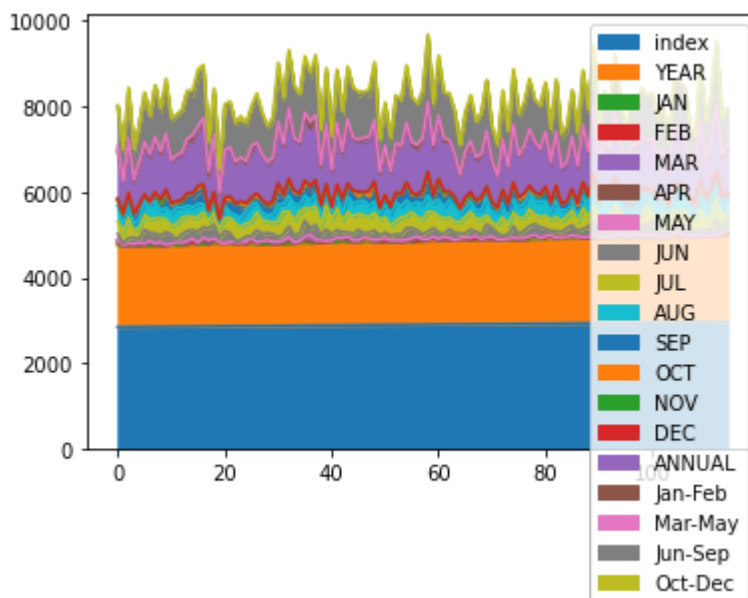
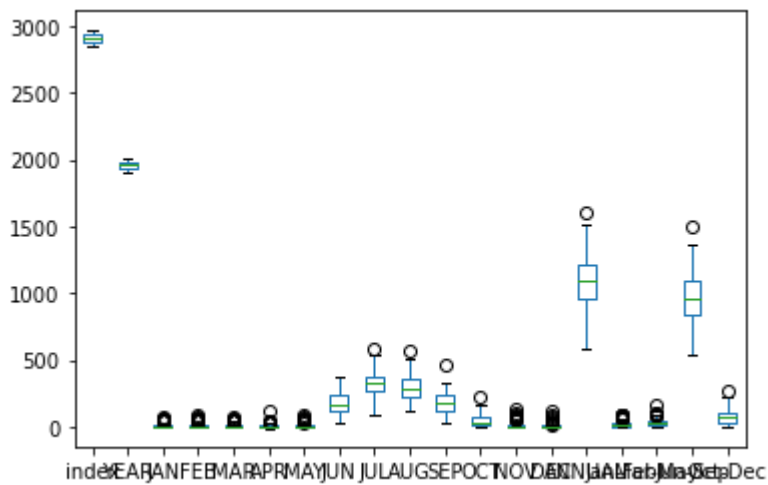Out[9]: <AxesSubplot:ylabel='Frequency'>

Area chart

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

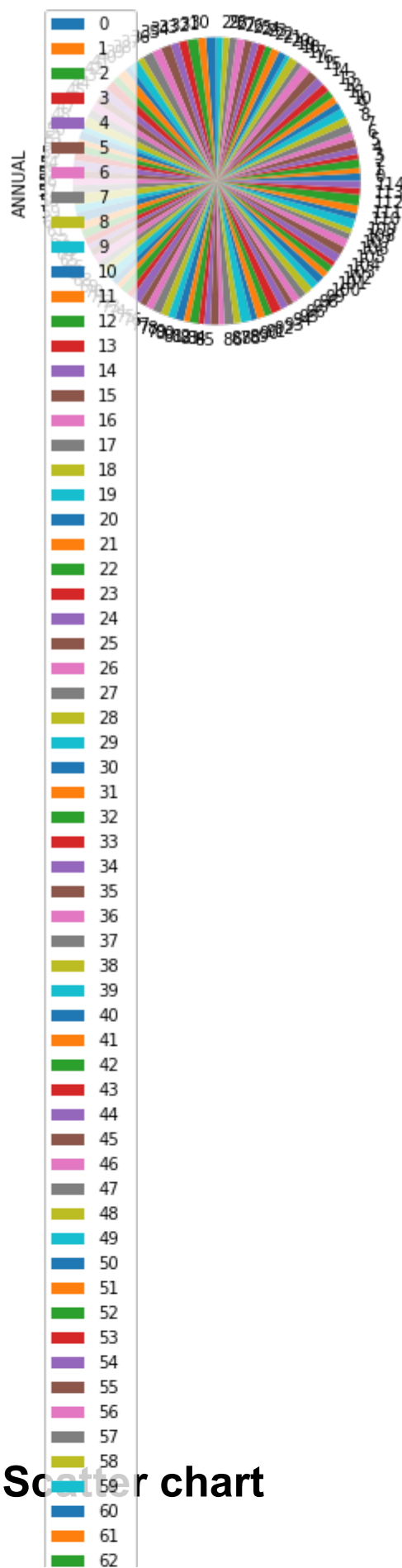Out[10]: <AxesSubplot:>

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

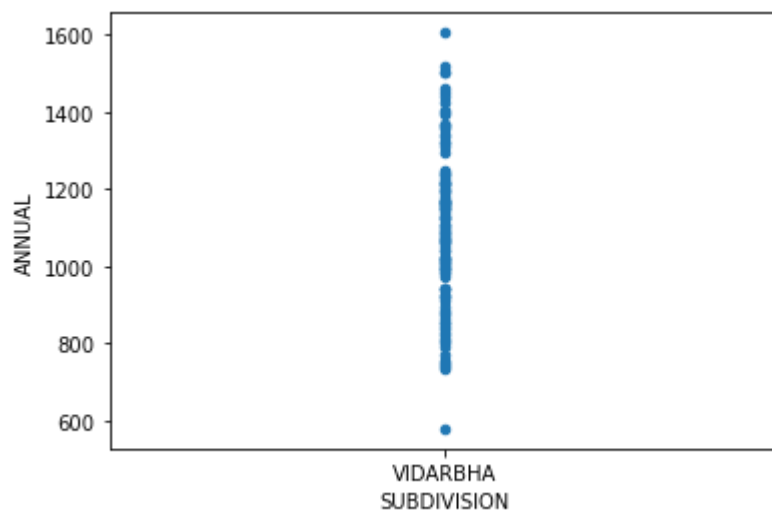In [13]: ```df.plot.scatter(x='SUBDIVISION', y='ANNUAL')```

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
```

In [15]: df.describe()

Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	11
mean	2909.000000	1958.000000	10.563478	11.982609	11.872174	9.435652	11.551304	17
std	33.341666	33.341666	15.105752	17.288888	15.540002	13.177554	14.604363	7
min	2852.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2
25%	2880.500000	1929.500000	0.400000	1.100000	0.950000	2.650000	2.250000	11
50%	2909.000000	1958.000000	3.800000	4.800000	4.900000	5.600000	6.200000	15
75%	2937.500000	1986.500000	14.950000	14.300000	16.050000	12.000000	14.900000	23
max	2966.000000	2015.000000	74.900000	84.900000	66.300000	112.700000	83.100000	37

EDA AND VISUALIZATION

In [16]: sns.pairplot(df)

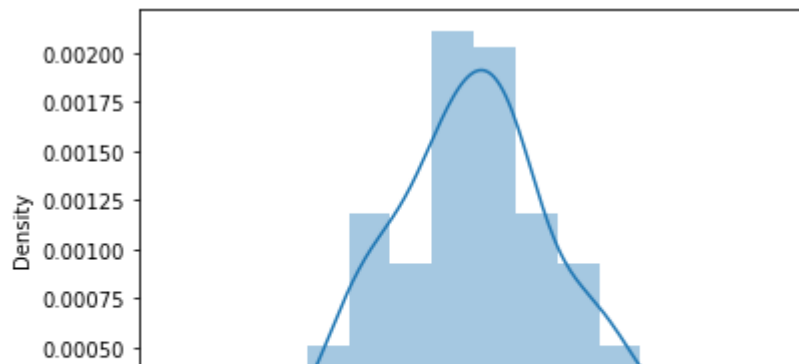Out[16]: <seaborn.axisgrid.PairGrid at 0x28fa6542ee0>
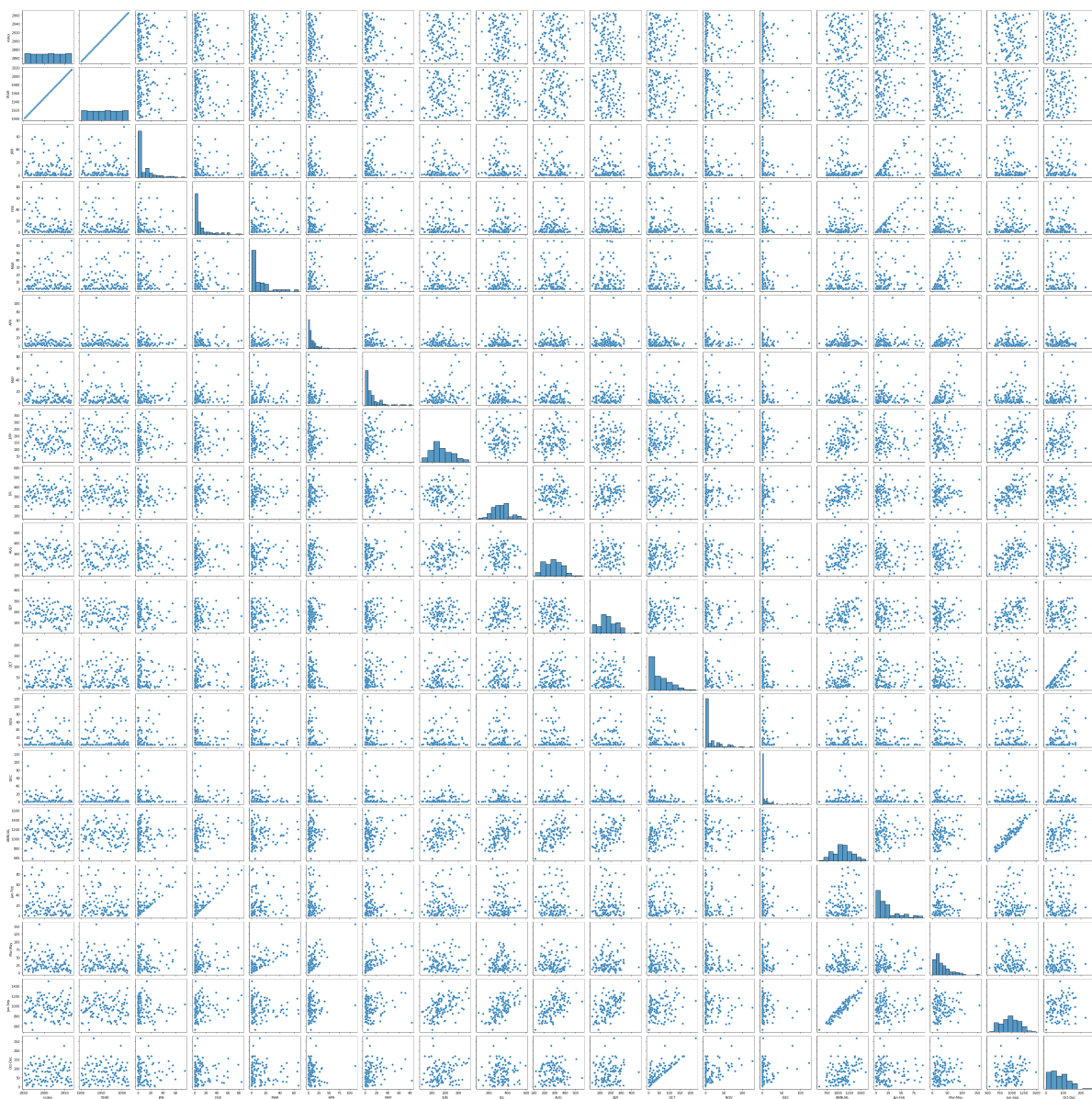
In [17]: `sns.distplot(df['ANNUAL'])`

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

Out[17]: <AxesSubplot:xlabel='ANNUAL', ylabel='Density'>

In [18]: `sns.heatmap(df.corr())`

Out[18]: <AxesSubplot:>