

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

```
In [2]: df=pd.read_csv(r'C:\Users\user\Desktop\rainfall\SUB HIMALAYAN WEST BENGAL SIKI')
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	437	SUB HIMALAYAN WEST BENGAL & SIKKIM	1901	26.5	14.8	14.1	29.2	195.5	488.4	524.8	501.1	242.7	55.5
1	438	SUB HIMALAYAN WEST BENGAL & SIKKIM	1902	1.2	0.7	87.1	126.1	271.3	539.2	671.0	603.8	799.9	74.4
2	439	SUB HIMALAYAN WEST BENGAL & SIKKIM	1903	5.5	8.7	19.6	18.6	163.6	541.2	431.5	708.8	365.2	14.1
3	440	SUB HIMALAYAN WEST BENGAL & SIKKIM	1904	3.4	29.2	0.9	124.3	333.6	274.2	500.4	468.5	260.6	16.4
4	441	SUB HIMALAYAN WEST BENGAL & SIKKIM	1905	12.0	31.2	51.9	104.4	290.6	524.8	523.1	1036.6	321.1	87.1
...
110	547	SUB HIMALAYAN WEST BENGAL & SIKKIM	2011	8.5	19.9	71.2	135.0	247.8	419.8	612.3	470.3	356.3	46.6
111	548	SUB HIMALAYAN WEST BENGAL & SIKKIM	2012	15.3	13.9	45.5	159.8	202.4	604.2	684.5	332.7	434.7	119.5
112	549	SUB HIMALAYAN WEST BENGAL & SIKKIM	2013	3.0	23.6	32.1	114.7	296.5	404.9	588.4	416.3	308.0	19.9
113	550	SUB HIMALAYAN WEST BENGAL & SIKKIM	2014	0.2	26.6	37.7	47.9	308.6	543.2	384.6	563.3	371.5	31.1
114	551	SUB HIMALAYAN WEST BENGAL & SIKKIM	2015	15.7	15.0	64.8	149.0	304.6	508.2	393.3	626.6	354.9	55.5

115 rows × 20 columns



In [3]:

```
df=df.dropna()
df
```

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	437	SUB HIMALAYAN WEST BENGAL & SIKKIM	1901	26.5	14.8	14.1	29.2	195.5	488.4	524.8	501.1	242.7	55.5
1	438	SUB HIMALAYAN WEST BENGAL & SIKKIM	1902	1.2	0.7	87.1	126.1	271.3	539.2	671.0	603.8	799.9	74.4
2	439	SUB HIMALAYAN WEST BENGAL & SIKKIM	1903	5.5	8.7	19.6	18.6	163.6	541.2	431.5	708.8	365.2	14.1
3	440	SUB HIMALAYAN WEST BENGAL & SIKKIM	1904	3.4	29.2	0.9	124.3	333.6	274.2	500.4	468.5	260.6	16.4
4	441	SUB HIMALAYAN WEST BENGAL & SIKKIM	1905	12.0	31.2	51.9	104.4	290.6	524.8	523.1	1036.6	321.1	87.1
...
110	547	SUB HIMALAYAN WEST BENGAL & SIKKIM	2011	8.5	19.9	71.2	135.0	247.8	419.8	612.3	470.3	356.3	46.6
111	548	SUB HIMALAYAN WEST BENGAL & SIKKIM	2012	15.3	13.9	45.5	159.8	202.4	604.2	684.5	332.7	434.7	119.5
112	549	SUB HIMALAYAN WEST BENGAL & SIKKIM	2013	3.0	23.6	32.1	114.7	296.5	404.9	588.4	416.3	308.0	19.9
113	550	SUB HIMALAYAN WEST BENGAL & SIKKIM	2014	0.2	26.6	37.7	47.9	308.6	543.2	384.6	563.3	371.5	31.1
114	551	SUB HIMALAYAN WEST BENGAL & SIKKIM	2015	15.7	15.0	64.8	149.0	304.6	508.2	393.3	626.6	354.9	55.5

115 rows × 20 columns



```
In [4]: df.columns
```

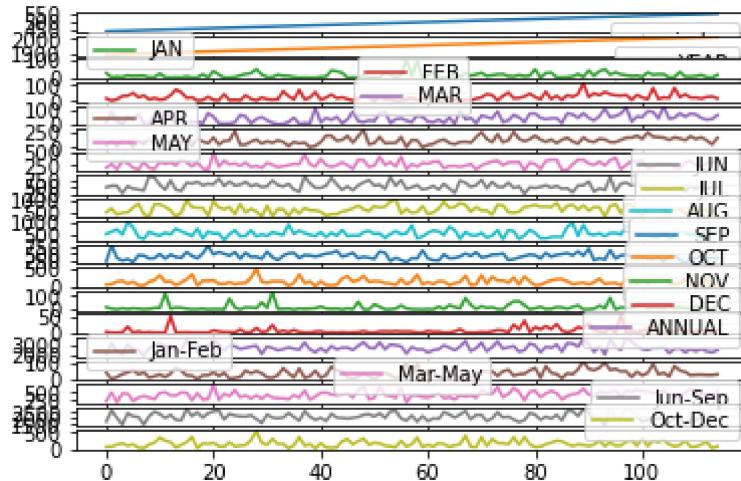
```
Out[4]: Index(['index', 'SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY',  
       'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL', 'Jan-Feb',  
       'Mar-May', 'Jun-Sep', 'Oct-Dec'],  
      dtype='object')
```

```
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
```

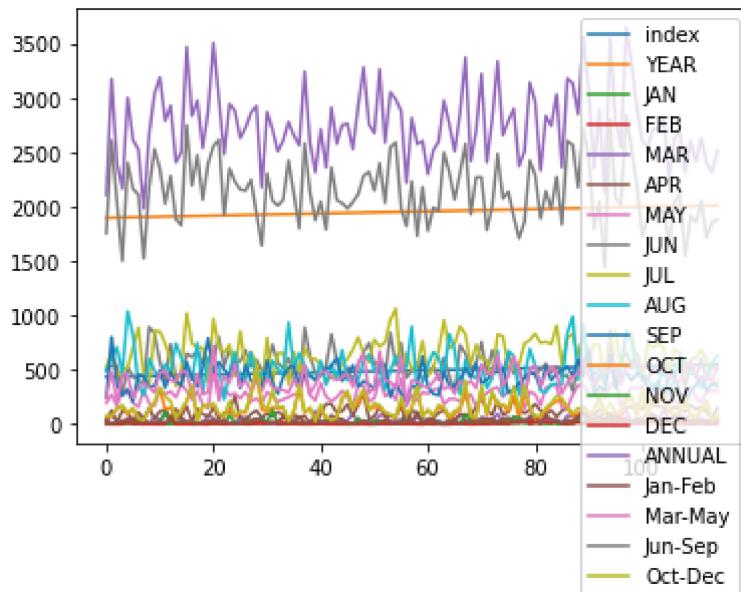
```
In [6]: df.plot.line(subplots=True)
```

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



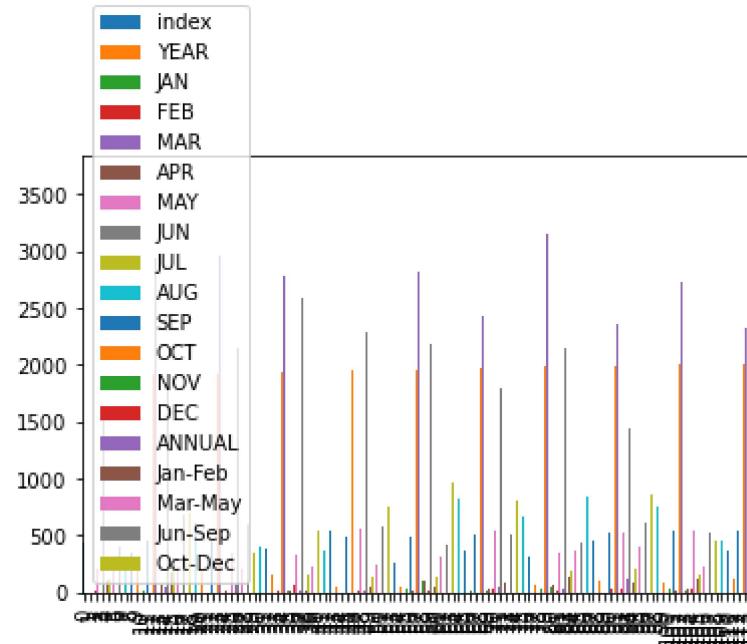
```
In [7]: df.plot.line()
```

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



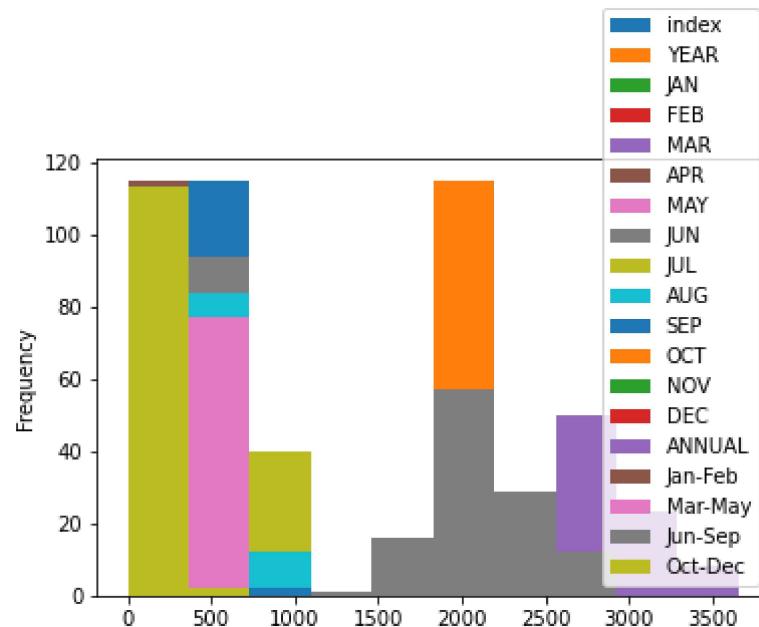
In [8]: `df.plot.bar()`

Out[8]: <AxesSubplot:>



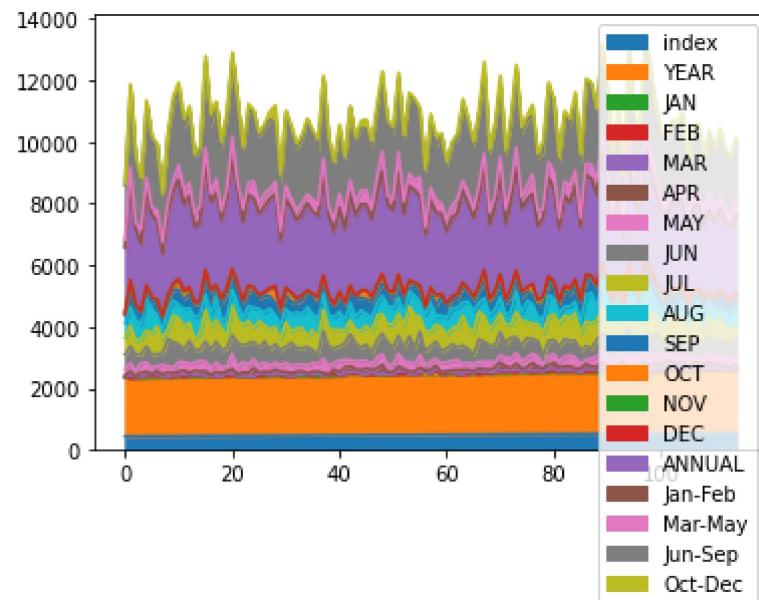
In [9]: `df.plot.hist()`

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



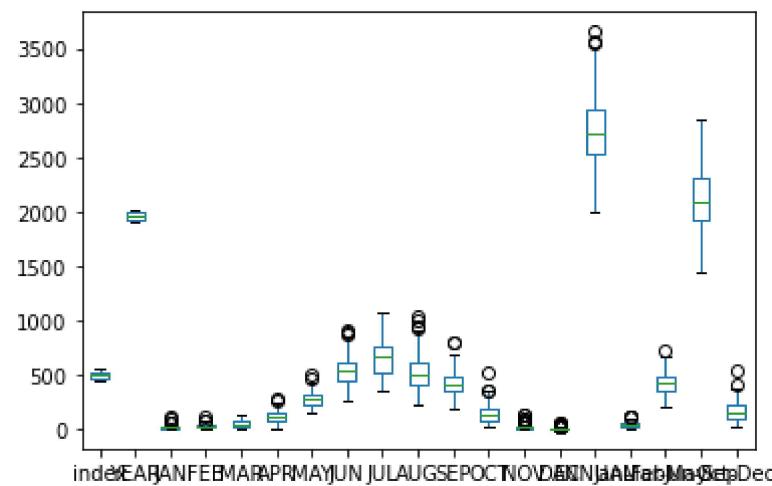
```
In [10]: df.plot.area()
```

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



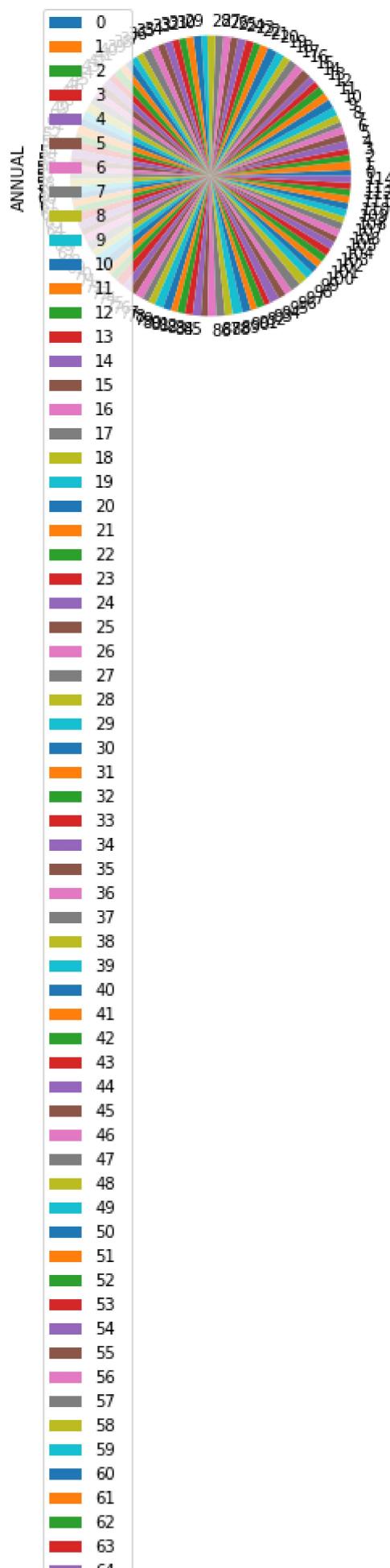
```
In [11]: df.plot.box()
```

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



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

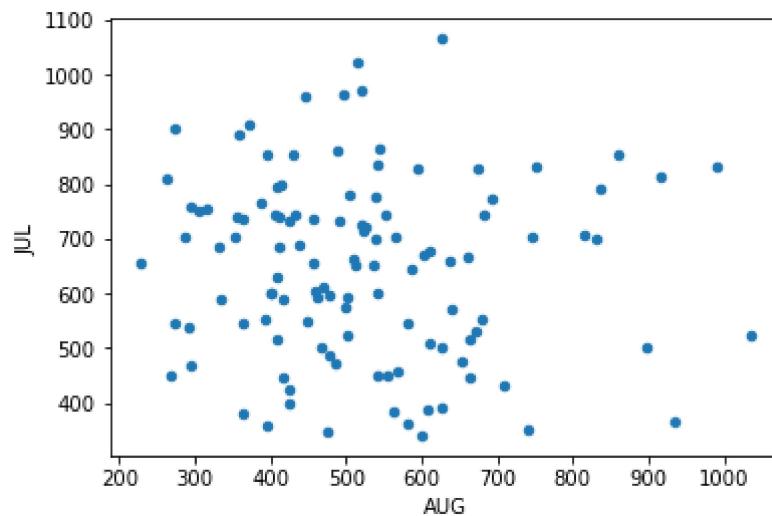
```
Out[12]: <AxesSubplot:ylabel='ANNUAL'>
```

64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

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

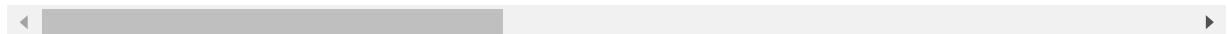
```
Out[13]: <AxesSubplot:xlabel='AUG', ylabel='JUL'>
```



```
In [14]: df.describe()
```

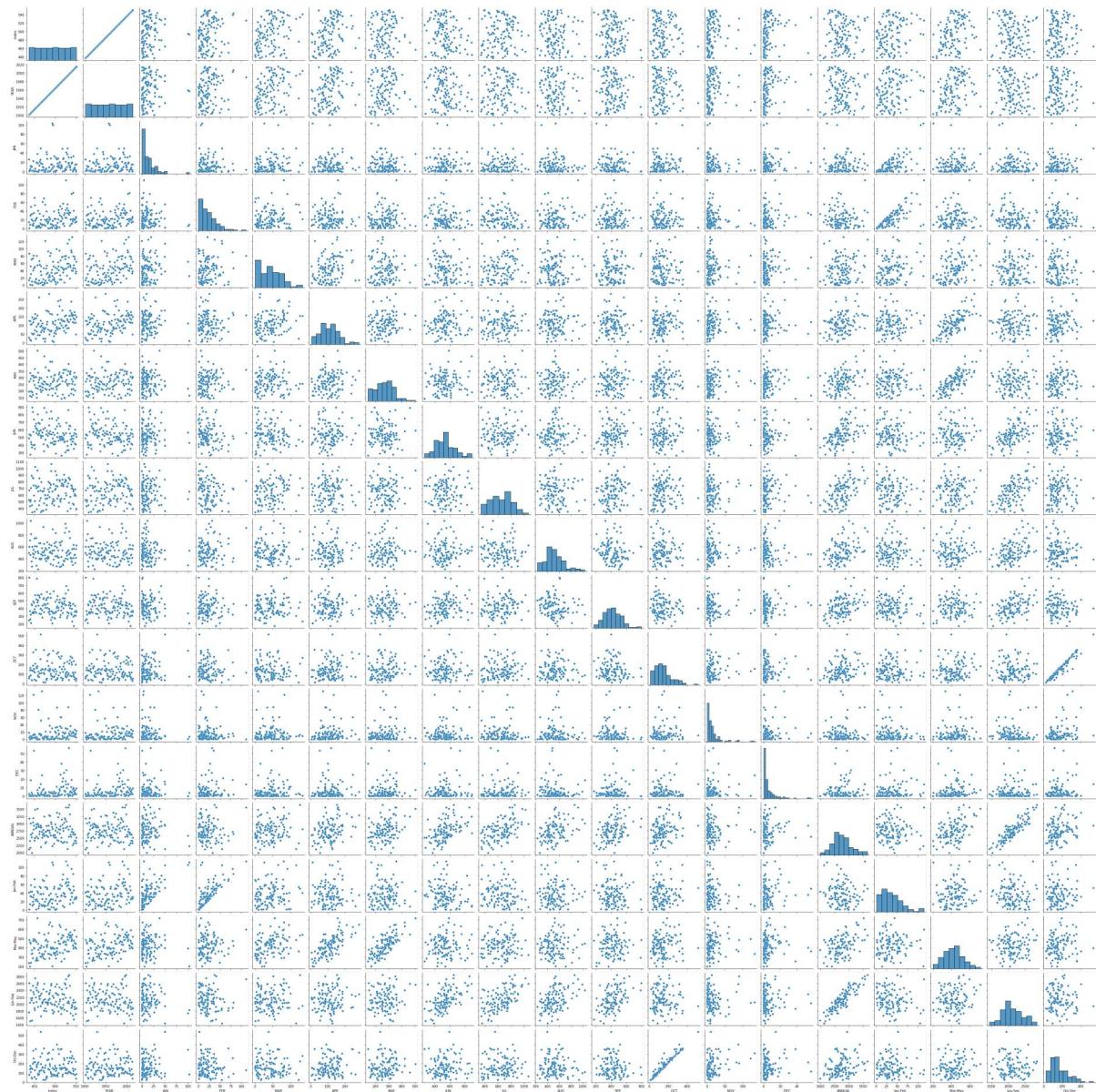
```
Out[14]:
```

	index	YEAR	JAN	FEB	MAR	APR	MAY	6
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	494.000000	1958.000000	14.083478	22.974783	43.135652	110.681739	269.143478	537
std	33.341666	33.341666	17.066089	19.583787	30.851319	55.688697	69.790921	134
min	437.000000	1901.000000	0.000000	0.100000	0.000000	4.800000	142.000000	261
25%	465.500000	1929.500000	2.250000	8.650000	15.100000	71.300000	217.100000	447
50%	494.000000	1958.000000	9.400000	19.600000	42.600000	110.900000	269.400000	527
75%	522.500000	1986.500000	19.550000	33.400000	63.650000	144.850000	311.100000	611
max	551.000000	2015.000000	103.000000	109.900000	132.100000	281.800000	503.100000	896



```
In [15]: sns.pairplot(df)
```

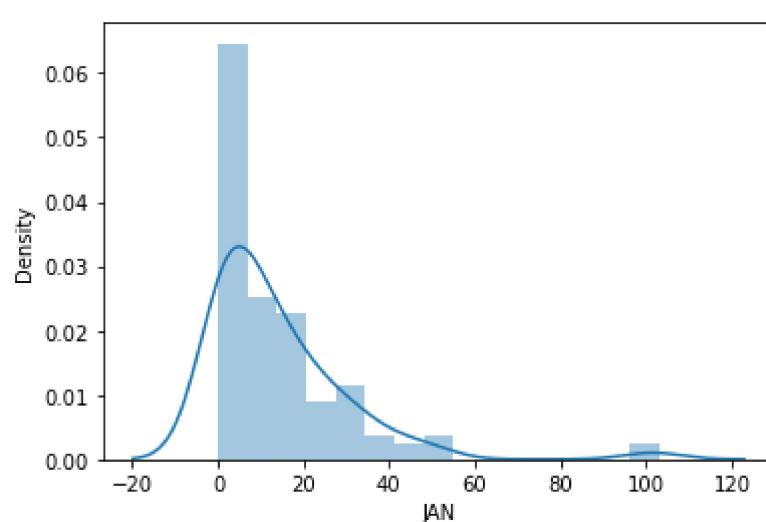
```
Out[15]: <seaborn.axisgrid.PairGrid at 0x15386087b80>
```



In [16]: `sns.distplot(df['JAN'])`

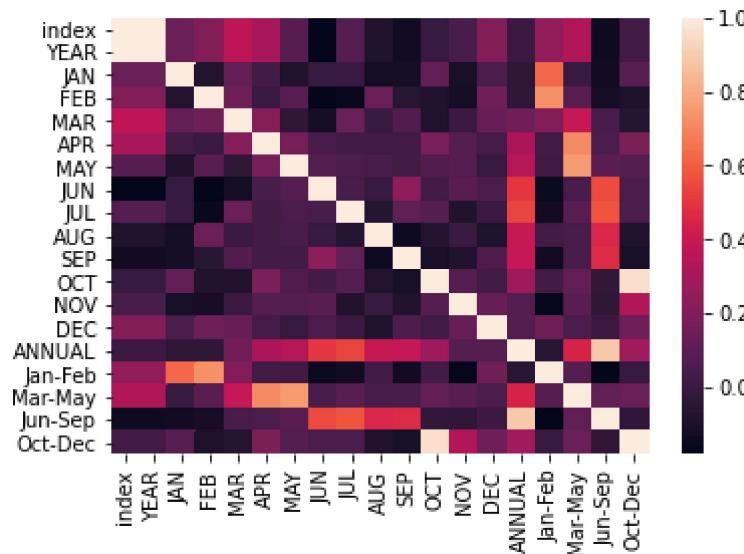
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function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

Out[16]: <AxesSubplot:xlabel='JAN', ylabel='Density'>



In [17]: `sns.heatmap(df.corr())`

Out[17]: <AxesSubplot:>



In []:

