In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df=pd.read_csv(r'C:\Users\user\Desktop\rainfall\EAST UTTAR PRADESH.csv')
df
```

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	1012	EAST UTTAR PRADESH	1901	62.6	31.3	8.2	1.1	13.6	21.8	226.5	285.6	215.4	4.9
1	1013	EAST UTTAR PRADESH	1902	6.1	2.3	2.4	2.0	21.4	32.5	411.5	155.4	257.2	13.2
2	1014	EAST UTTAR PRADESH	1903	8.2	0.4	1.3	0.7	15.3	71.6	115.3	420.2	258.7	324.7
3	1015	EAST UTTAR PRADESH	1904	7.3	1.5	8.3	0.4	28.7	148.0	359.4	328.8	95.0	50.6
4	1016	EAST UTTAR PRADESH	1905	16.8	23.6	20.0	5.4	15.4	17.3	302.4	316.2	169.5	3.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	1122	EAST UTTAR PRADESH	2011	1.0	2.7	1.6	2.9	32.2	163.8	197.9	232.1	146.4	0.6
111	1123	EAST UTTAR PRADESH	2012	20.3	1.2	3.4	2.8	0.2	18.5	234.2	156.0	164.4	0.7
112	1124	EAST UTTAR PRADESH	2013	6.1	59.6	2.7	8.7	1.1	309.7	230.0	246.1	78.2	97.4
113	1125	EAST UTTAR PRADESH	2014	47.4	25.8	15.4	1.7	10.7	47.8	224.5	138.1	106.7	74.7
114	1126	EAST UTTAR PRADESH	2015	30.0	4.1	48.2	23.2	8.6	95.3	179.0	175.8	21.9	11.8

115 rows × 20 columns

In [4]:
```
df=df.dropna()
df
```

Out[4]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	1012	EAST UTTAR PRADESH	1901	62.6	31.3	8.2	1.1	13.6	21.8	226.5	285.6	215.4	4.9
1	1013	EAST UTTAR PRADESH	1902	6.1	2.3	2.4	2.0	21.4	32.5	411.5	155.4	257.2	13.2
2	1014	EAST UTTAR PRADESH	1903	8.2	0.4	1.3	0.7	15.3	71.6	115.3	420.2	258.7	324.7
3	1015	EAST UTTAR PRADESH	1904	7.3	1.5	8.3	0.4	28.7	148.0	359.4	328.8	95.0	50.6
4	1016	EAST UTTAR PRADESH	1905	16.8	23.6	20.0	5.4	15.4	17.3	302.4	316.2	169.5	3.3
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	1122	EAST UTTAR PRADESH	2011	1.0	2.7	1.6	2.9	32.2	163.8	197.9	232.1	146.4	0.6
111	1123	EAST UTTAR PRADESH	2012	20.3	1.2	3.4	2.8	0.2	18.5	234.2	156.0	164.4	0.7
112	1124	EAST UTTAR PRADESH	2013	6.1	59.6	2.7	8.7	1.1	309.7	230.0	246.1	78.2	97.4
113	1125	EAST UTTAR PRADESH	2014	47.4	25.8	15.4	1.7	10.7	47.8	224.5	138.1	106.7	74.7
114	1126	EAST UTTAR PRADESH	2015	30.0	4.1	48.2	23.2	8.6	95.3	179.0	175.8	21.9	11.8

115 rows × 20 columns

In [5]:
```
df.columns
```

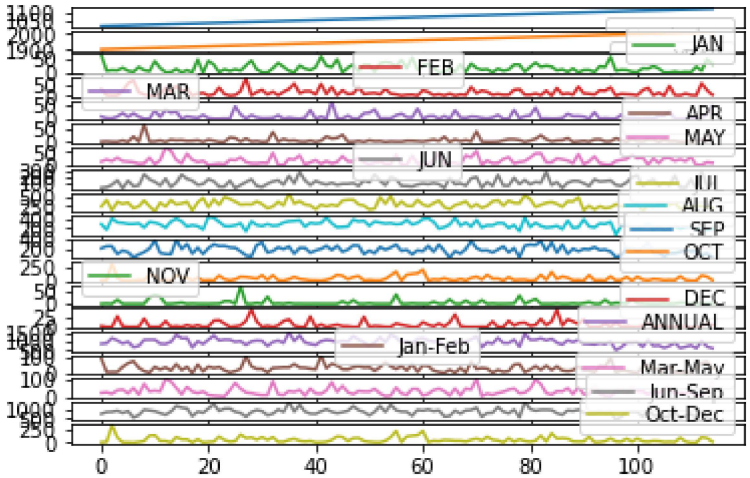
Out[5]:
```
Index(['index', 'SUBDIVISION', 'YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY',
       'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC', 'ANNUAL', 'Jan-Feb',
       'Mar-May', 'Jun-Sep', 'Oct-Dec'],
      dtype='object')
```

In [6]:
```
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
```

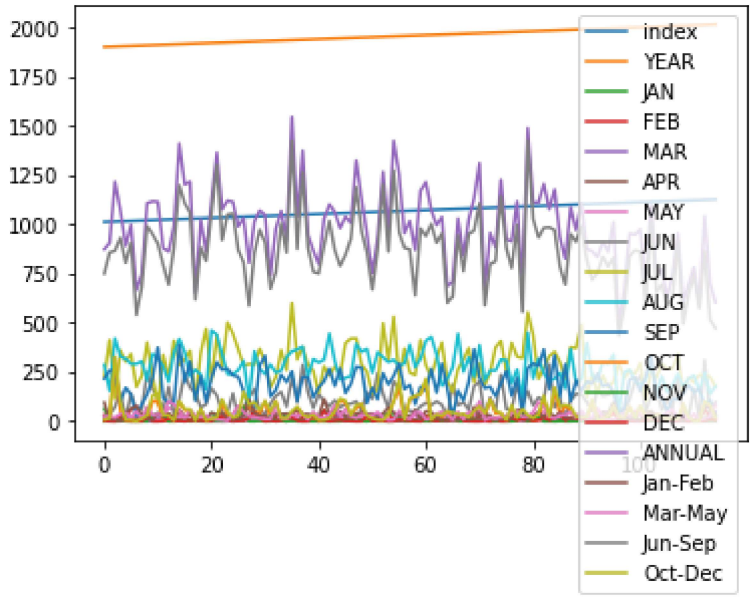
In [7]:
```
df.plot.line(subplots=True)
```

Out[7]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>],
      dtype=object)
```



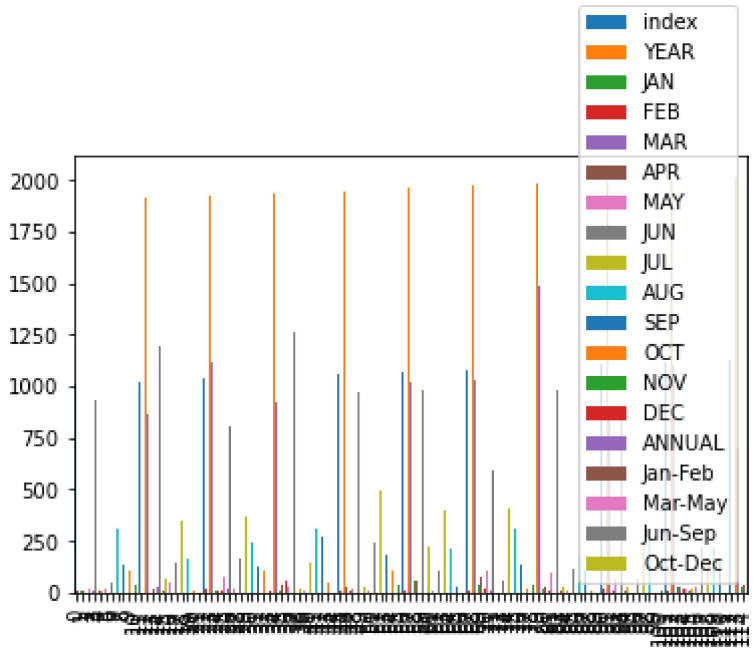
In [8]:
```
df.plot.line()
```

Out[8]: <AxesSubplot:>



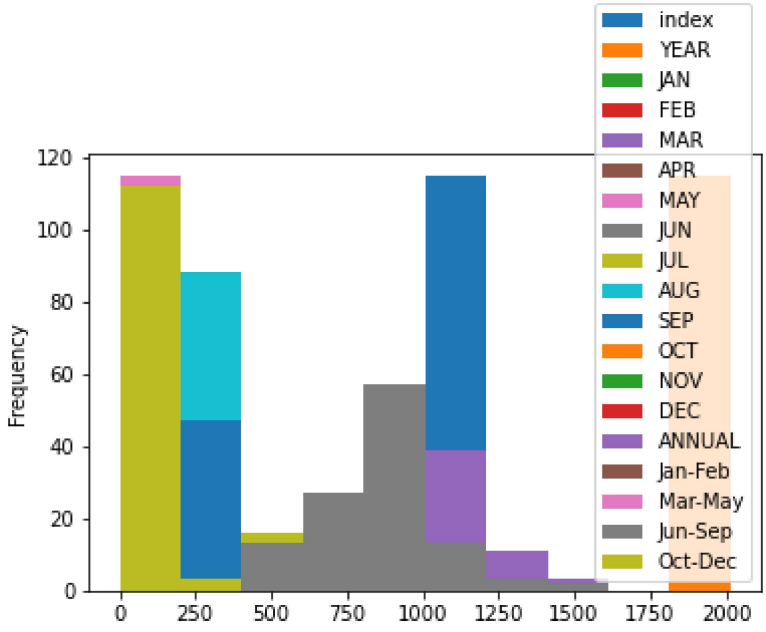
In [9]:
```
df.plot.bar()
```

Out[9]: <AxesSubplot:>



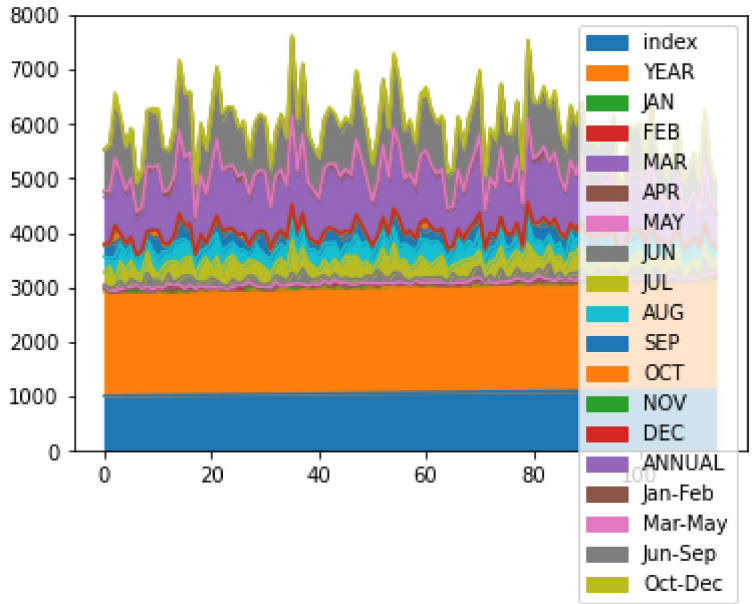
In [10]:
```
df.plot.hist()
```

Out[10]: <AxesSubplot:ylabel='Frequency'>



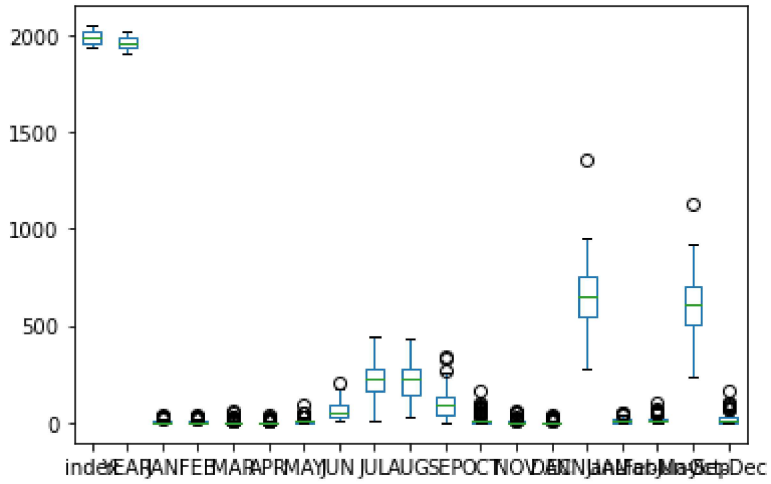
In [11]:
```
df.plot.area()
```

Out[11]: <AxesSubplot:>



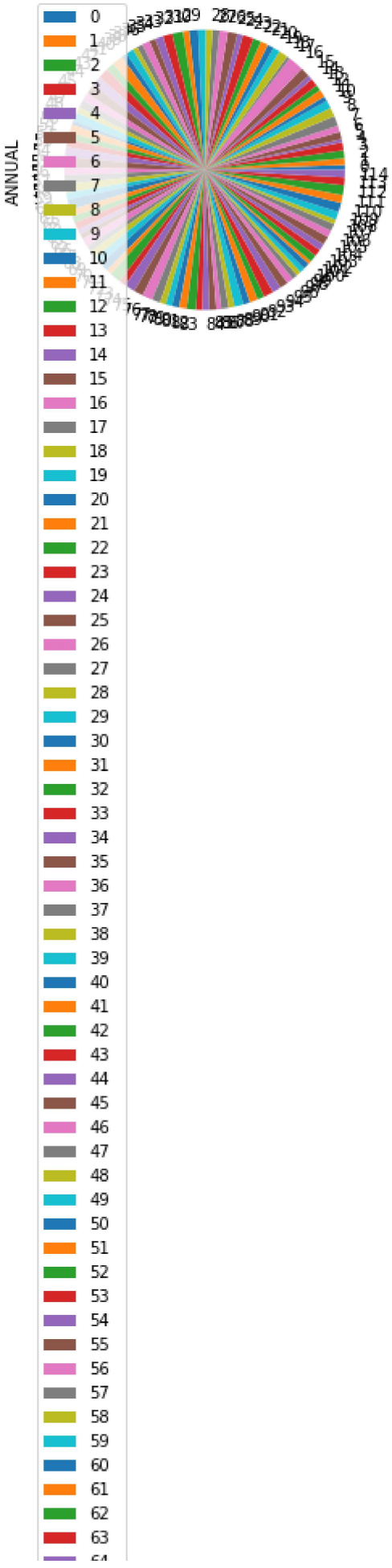
In [11]:
```
df.plot.box()
```

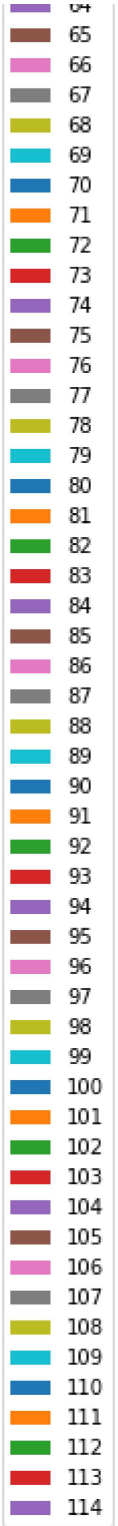
Out[11]: <AxesSubplot:>



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

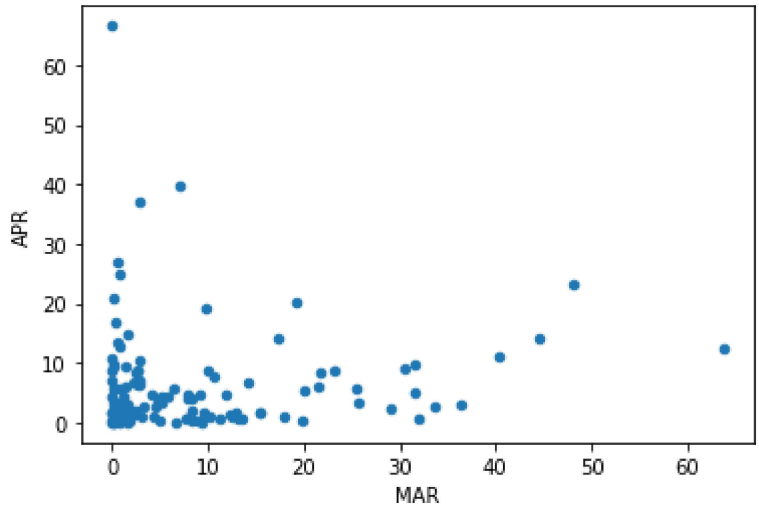
Out[12]:
```
<AxesSubplot:ylabel='ANNUAL'>
```


ANNUAL
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63


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 [12]:
```
df.plot.scatter(x='MAR',y='APR')
```

Out[12]: <AxesSubplot:xlabel='MAR', ylabel='APR'>



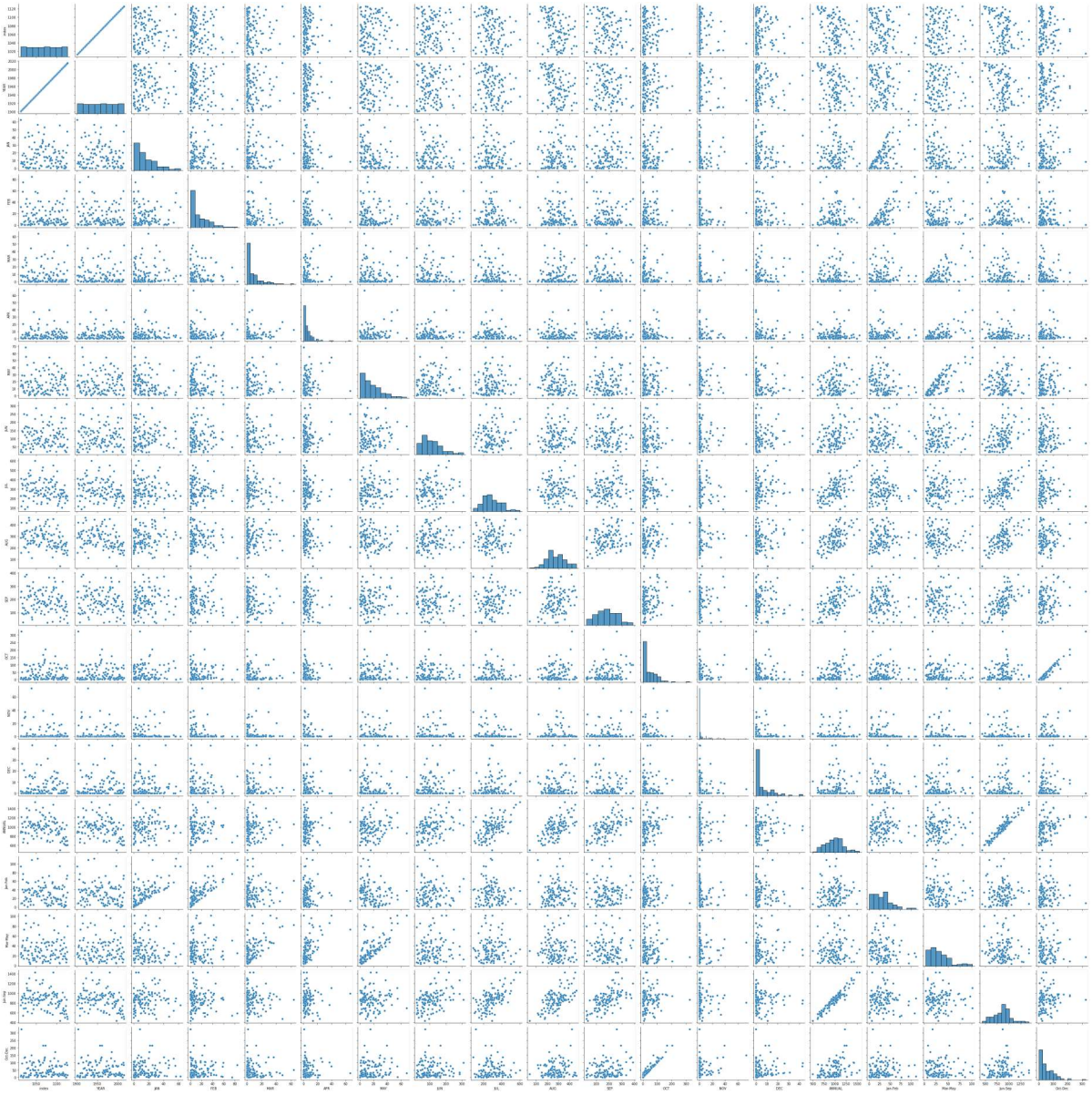
In [13]:
```
df.describe()
```

Out[13]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	11
mean	1069.000000	1958.000000	16.012174	15.873913	8.907826	6.430435	17.211304	11
std	33.341666	33.341666	14.659481	17.263607	11.906840	9.082731	14.095232	6
min	1012.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.200000	1
25%	1040.500000	1929.500000	3.450000	3.200000	0.700000	1.150000	6.550000	6
50%	1069.000000	1958.000000	12.500000	8.800000	3.400000	3.800000	14.000000	9
75%	1097.500000	1986.500000	24.400000	24.150000	12.050000	8.550000	25.900000	14
max	1126.000000	2015.000000	62.600000	84.300000	63.700000	66.600000	68.700000	30

In [14]:
```
sns.pairplot(df)
```

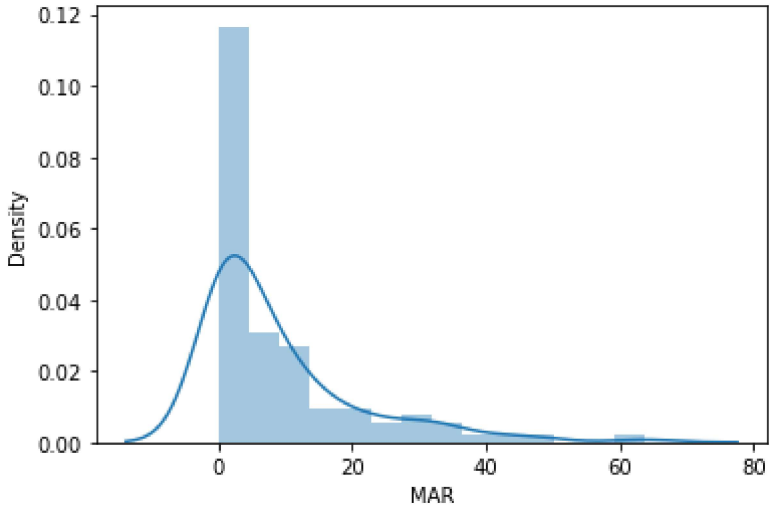
Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x1f925908bb0>
```



In [15]:
```
sns.distplot(df['MAR'])
```

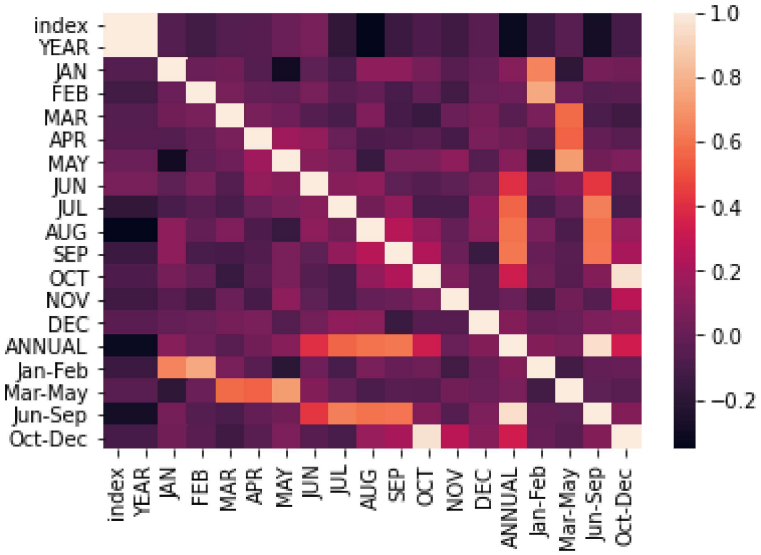
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function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

Out[15]: <AxesSubplot:xlabel='MAR', ylabel='Density'>



In [16]:
```
sns.heatmap(df.corr())
```

Out[16]: <AxesSubplot:>



In [ ]: