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("an.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5
1	1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2
2	2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2
3	3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2
4	4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...
105	105	ANDAMAN & NICOBAR ISLANDS	2011	265.9	84.8	272.8	111.4	326.5	383.2	583.2	441.5	757.1	212.3
106	106	ANDAMAN & NICOBAR ISLANDS	2012	119.9	45.6	30.9	55.8	533.9	458.2	317.3	369.6	868.9	209.7
107	107	ANDAMAN & NICOBAR ISLANDS	2013	67.1	37.6	43.0	46.3	509.3	777.0	564.8	336.7	473.6	455.8
108	108	ANDAMAN & NICOBAR ISLANDS	2014	41.9	8.6	0.0	11.1	238.0	416.6	467.6	321.6	412.9	402.6
109	109	ANDAMAN & NICOBAR ISLANDS	2015	126.8	7.6	3.1	138.2	331.9	346.4	328.9	480.0	523.3	252.1

110 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')
```

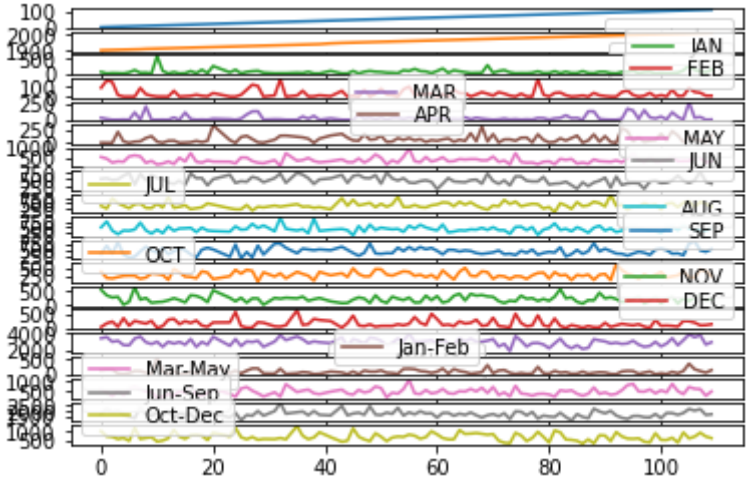
In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 104 entries, 0 to 109
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   index        104 non-null    int64  
 1   SUBDIVISION  104 non-null    object 
 2   YEAR         104 non-null    int64  
 3   JAN          104 non-null    float64
 4   FEB          104 non-null    float64
 5   MAR          104 non-null    float64
 6   APR          104 non-null    float64
 7   MAY          104 non-null    float64
 8   JUN          104 non-null    float64
 9   JUL          104 non-null    float64
 10  AUG          104 non-null    float64
 11  SEP          104 non-null    float64
 12  OCT          104 non-null    float64
 13  NOV          104 non-null    float64
 14  DEC          104 non-null    float64
 15  ANNUAL       104 non-null    float64
 16  Jan-Feb      104 non-null    float64
 17  Mar-May      104 non-null    float64
 18  Jun-Sep      104 non-null    float64
 19  Oct-Dec      104 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 17.1+ KB
```

Line chart

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

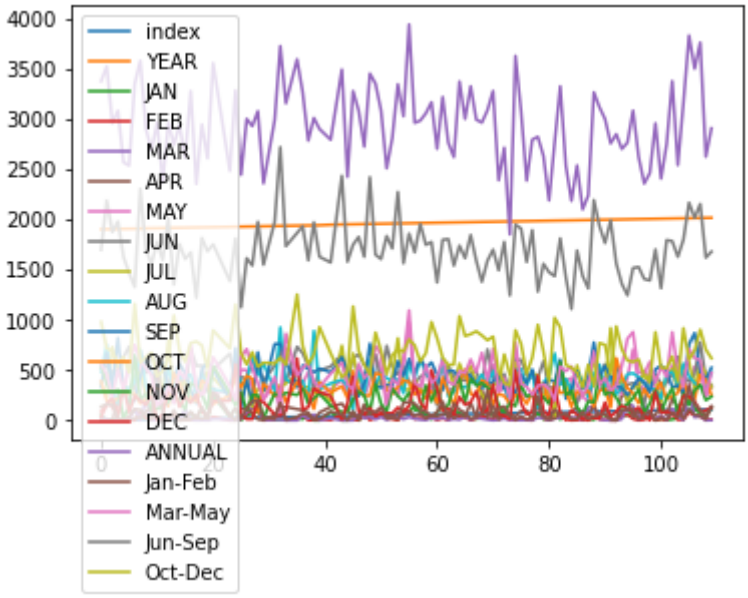
Out[6]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



Line chart

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

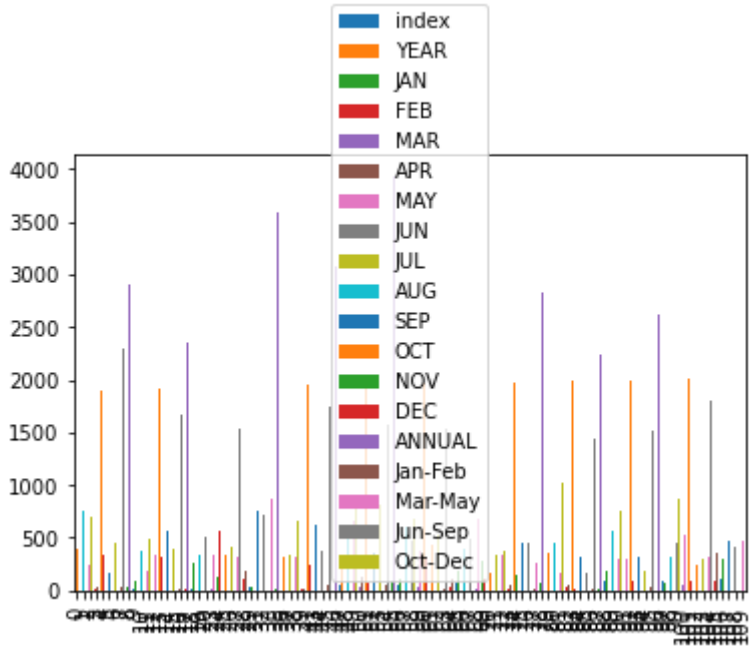
Out[7]: <AxesSubplot:>



Bar chart

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

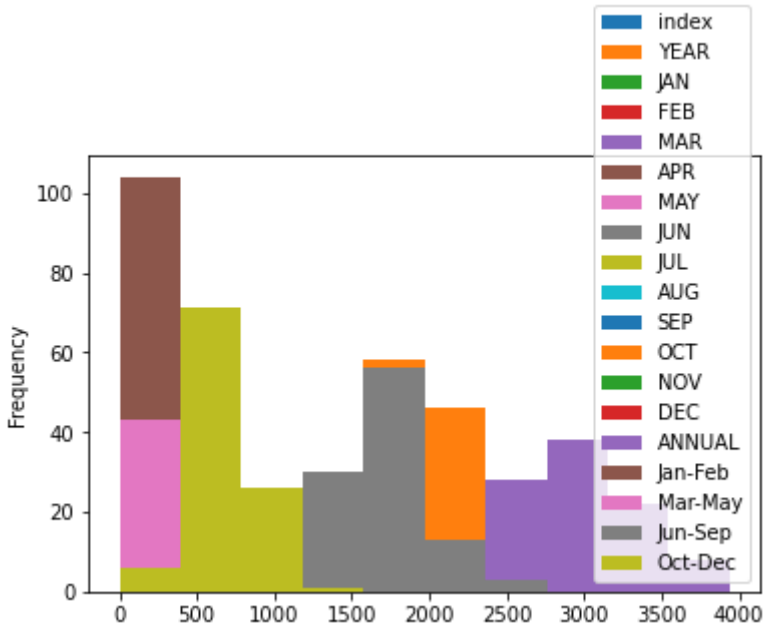
Out[8]: <AxesSubplot:>



Histogram

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

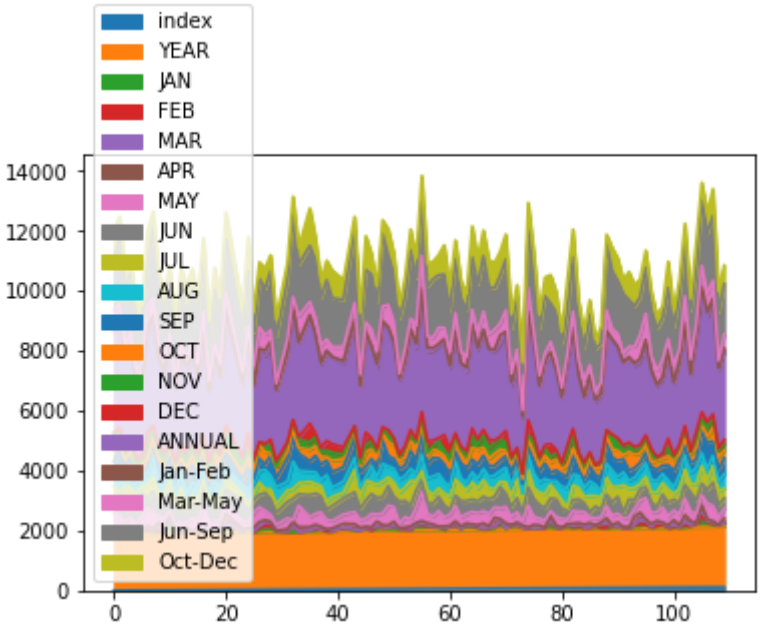
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area chart

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

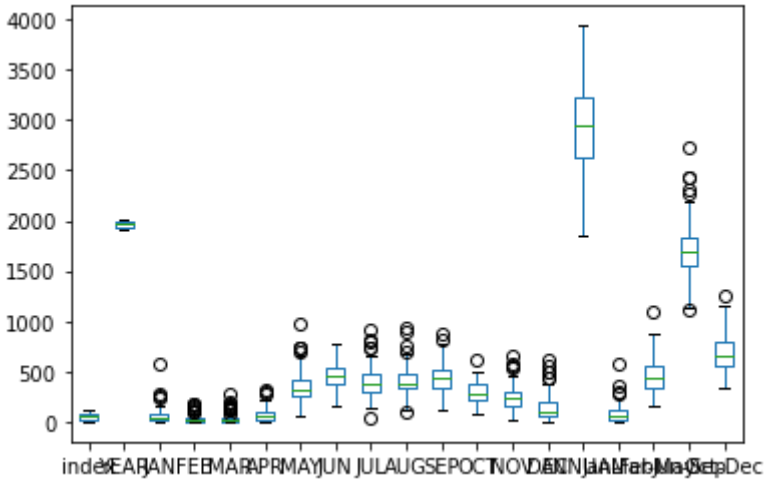
Out[10]: <AxesSubplot:>



Box chart

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

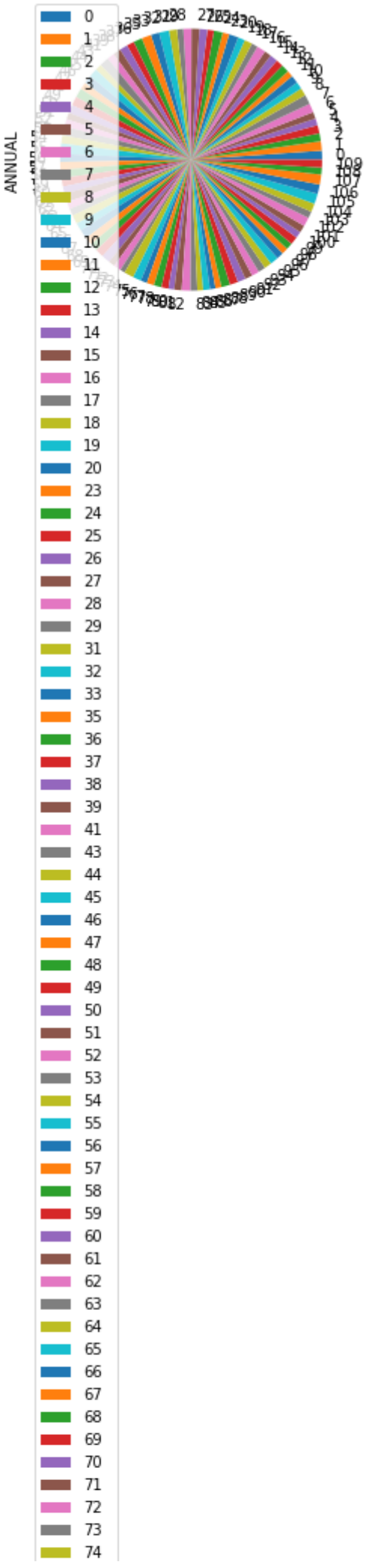
Out[11]: <AxesSubplot:>

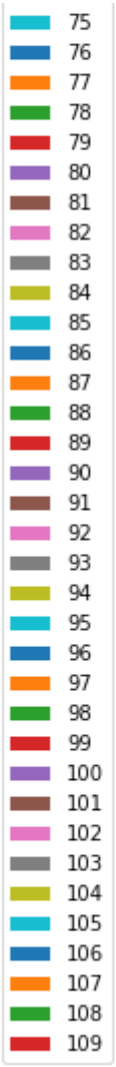


Pie chart

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

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

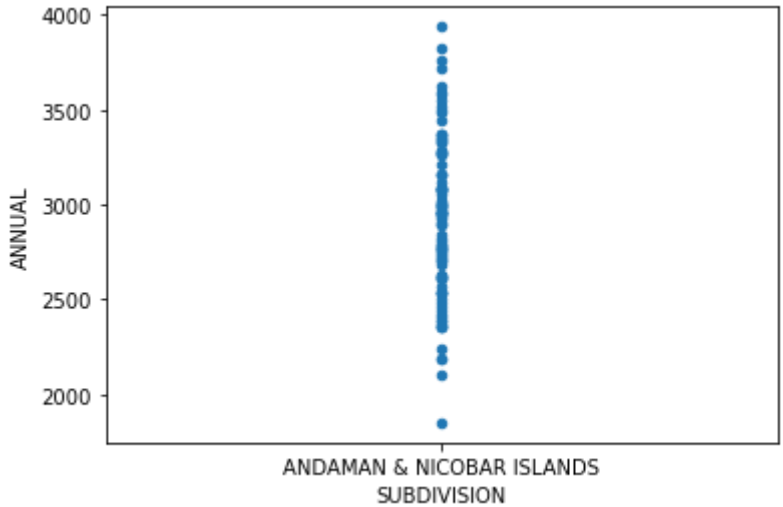

ANNUAL
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
23
24
25
26
27
28
29
31
32
33
35
36
37
38
39
41
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74



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: 104 entries, 0 to 109
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   index        104 non-null    int64  
 1   SUBDIVISION  104 non-null    object 
 2   YEAR         104 non-null    int64  
 3   JAN          104 non-null    float64
 4   FEB          104 non-null    float64
 5   MAR          104 non-null    float64
 6   APR          104 non-null    float64
 7   MAY          104 non-null    float64
 8   JUN          104 non-null    float64
 9   JUL          104 non-null    float64
 10  AUG          104 non-null    float64
 11  SEP          104 non-null    float64
 12  OCT          104 non-null    float64
 13  NOV          104 non-null    float64
 14  DEC          104 non-null    float64
 15  ANNUAL       104 non-null    float64
 16  Jan-Feb      104 non-null    float64
 17  Mar-May      104 non-null    float64
 18  Jun-Sep      104 non-null    float64
 19  Oct-Dec      104 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 17.1+ KB
```

In [15]:
```
df.describe()
```

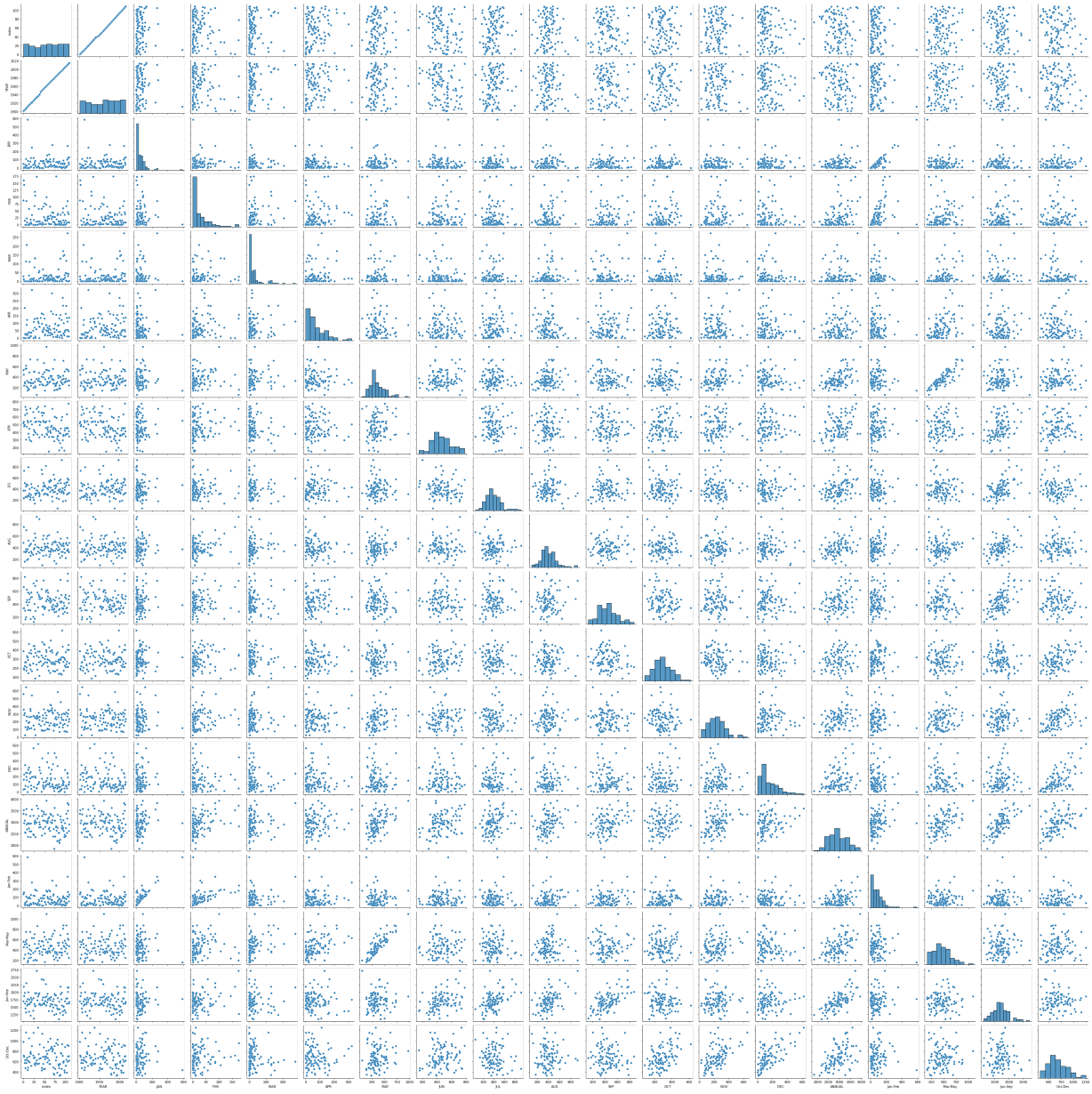
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	JU
count	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000	104.000000	104.000(
mean	55.826923	1960.355769	53.829808	28.299038	31.080769	71.473077	361.098077	465.357(
std	32.254884	34.010826	75.012392	38.286466	48.842153	66.908670	150.341139	136.471(
min	0.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	62.000000	148.800(
25%	27.750000	1929.750000	10.200000	1.775000	2.300000	21.025000	263.125000	369.975(
50%	57.500000	1963.500000	31.750000	12.800000	12.100000	52.300000	321.050000	450.250(
75%	83.250000	1989.250000	76.275000	36.325000	31.775000	103.350000	425.325000	545.625(
max	109.000000	2015.000000	583.700000	173.800000	272.800000	323.100000	973.100000	777.000(

EDA AND VISUALIZATION

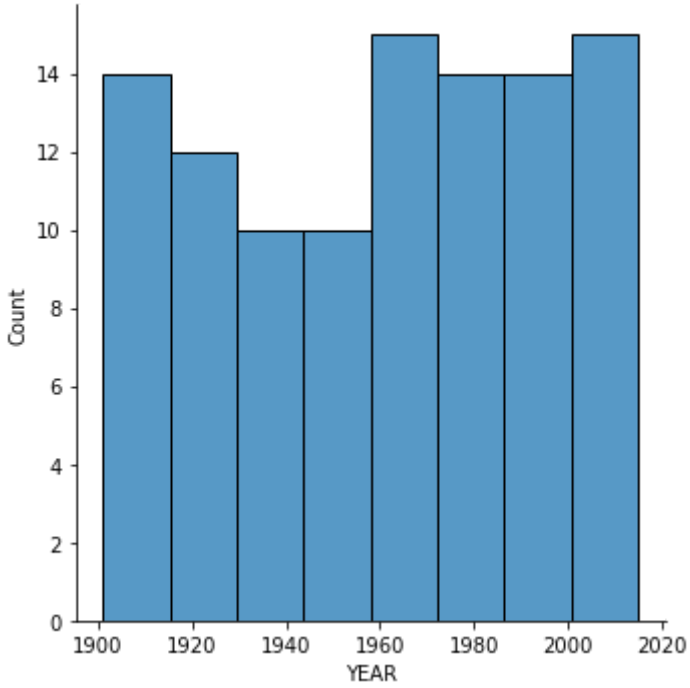
In [16]:
```
sns.pairplot(df)
```

Out[16]:
```
<seaborn.axisgrid.PairGrid at 0x1d6198d9b80>
```



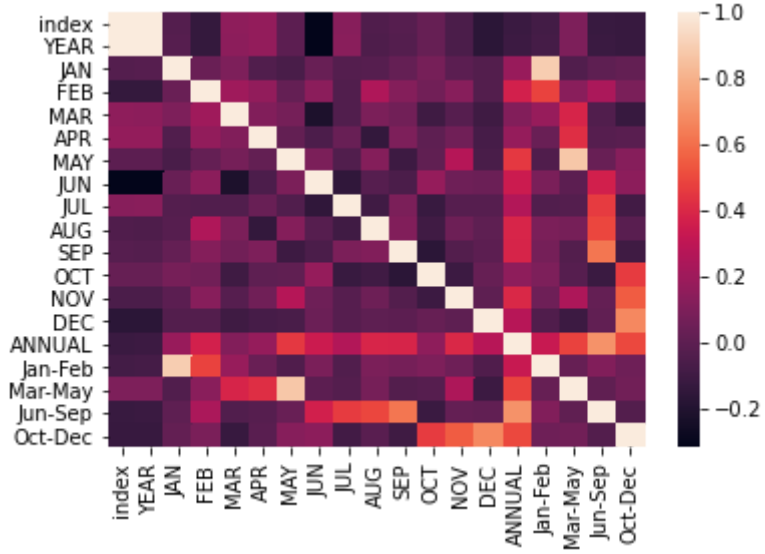
In [19]:
```
sns.displot(df['YEAR'])
```

Out[19]:
```
<seaborn.axisgrid.FacetGrid at 0x1d6187400d0>
```



In [20]:
```
sns.heatmap(df.corr())
```

Out[20]: <AxesSubplot:>



In [ ]: