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("rainfall_arunachal pradesh.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NC
0	110	ARUNACHAL PRADESH	1916	48.1	69.8	71.1	316.1	424.6	1124.9	NaN	629.7	333.9	NaN	Na
1	111	ARUNACHAL PRADESH	1917	21.4	164.5	NaN	269.6	107.9	823.8	909.1	628.4	411.5	199.3	63
2	112	ARUNACHAL PRADESH	1918	10.4	11.0	191.2	144.6	861.1	1609.9	1303.0	692.6	515.8	125.2	7
3	113	ARUNACHAL PRADESH	1919	34.5	67.8	28.5	256.9	420.6	973.6	999.0	286.7	628.7	948.3	40
4	114	ARUNACHAL PRADESH	1920	14.0	196.3	605.6	364.7	173.6	840.6	535.4	896.5	376.7	103.3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
92	202	ARUNACHAL PRADESH	2011	40.0	51.3	174.5	240.8	219.6	288.4	531.4	277.6	286.7	51.9	16
93	203	ARUNACHAL PRADESH	2012	57.8	35.8	134.2	403.4	187.4	645.8	638.9	316.0	724.9	248.1	22
94	204	ARUNACHAL PRADESH	2013	18.5	40.5	115.1	175.1	335.8	290.0	329.6	230.2	316.1	164.1	13
95	205	ARUNACHAL PRADESH	2014	19.0	101.9	80.3	86.7	299.0	415.8	392.4	599.6	343.0	35.1	20
96	206	ARUNACHAL PRADESH	2015	30.8	47.5	97.5	287.1	238.9	637.9	329.3	595.5	374.2	65.2	33

97 rows × 20 columns

head

In [3]:
```
df.head(5)
df
```

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NC
0	110	ARUNACHAL PRADESH	1916	48.1	69.8	71.1	316.1	424.6	1124.9	NaN	629.7	333.9	NaN	Na
1	111	ARUNACHAL PRADESH	1917	21.4	164.5	NaN	269.6	107.9	823.8	909.1	628.4	411.5	199.3	63
2	112	ARUNACHAL PRADESH	1918	10.4	11.0	191.2	144.6	861.1	1609.9	1303.0	692.6	515.8	125.2	7
3	113	ARUNACHAL PRADESH	1919	34.5	67.8	28.5	256.9	420.6	973.6	999.0	286.7	628.7	948.3	40
4	114	ARUNACHAL PRADESH	1920	14.0	196.3	605.6	364.7	173.6	840.6	535.4	896.5	376.7	103.3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
92	202	ARUNACHAL PRADESH	2011	40.0	51.3	174.5	240.8	219.6	288.4	531.4	277.6	286.7	51.9	16
93	203	ARUNACHAL PRADESH	2012	57.8	35.8	134.2	403.4	187.4	645.8	638.9	316.0	724.9	248.1	22
94	204	ARUNACHAL PRADESH	2013	18.5	40.5	115.1	175.1	335.8	290.0	329.6	230.2	316.1	164.1	13
95	205	ARUNACHAL PRADESH	2014	19.0	101.9	80.3	86.7	299.0	415.8	392.4	599.6	343.0	35.1	20
96	206	ARUNACHAL PRADESH	2015	30.8	47.5	97.5	287.1	238.9	637.9	329.3	595.5	374.2	65.2	33

97 rows × 20 columns

tail

In [4]:
```
df.tail(5)
df
```

Out[4]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NC
0	110	ARUNACHAL PRADESH	1916	48.1	69.8	71.1	316.1	424.6	1124.9	NaN	629.7	333.9	NaN	Na
1	111	ARUNACHAL PRADESH	1917	21.4	164.5	NaN	269.6	107.9	823.8	909.1	628.4	411.5	199.3	63
2	112	ARUNACHAL PRADESH	1918	10.4	11.0	191.2	144.6	861.1	1609.9	1303.0	692.6	515.8	125.2	7

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NC
3	113	ARUNACHAL PRADESH	1919	34.5	67.8	28.5	256.9	420.6	973.6	999.0	286.7	628.7	948.3	40
4	114	ARUNACHAL PRADESH	1920	14.0	196.3	605.6	364.7	173.6	840.6	535.4	896.5	376.7	103.3	C
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
92	202	ARUNACHAL PRADESH	2011	40.0	51.3	174.5	240.8	219.6	288.4	531.4	277.6	286.7	51.9	16
93	203	ARUNACHAL PRADESH	2012	57.8	35.8	134.2	403.4	187.4	645.8	638.9	316.0	724.9	248.1	22
94	204	ARUNACHAL PRADESH	2013	18.5	40.5	115.1	175.1	335.8	290.0	329.6	230.2	316.1	164.1	13
95	205	ARUNACHAL PRADESH	2014	19.0	101.9	80.3	86.7	299.0	415.8	392.4	599.6	343.0	35.1	20
96	206	ARUNACHAL PRADESH	2015	30.8	47.5	97.5	287.1	238.9	637.9	329.3	595.5	374.2	65.2	33

97 rows × 20 columns

Data Cleaning and Data Preprocessing

describe()

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

Out[5]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUI
count	97.000000	97.000000	96.000000	96.000000	95.000000	97.000000	97.000000	96.00000
mean	158.000000	1965.824742	47.297917	91.116667	153.527368	263.836082	358.522680	647.37395
std	28.145456	29.400613	34.256472	46.619805	86.182539	114.490513	177.484444	307.55936
min	110.000000	1916.000000	0.600000	6.100000	28.500000	86.700000	101.800000	239.40000
25%	134.000000	1940.000000	19.275000	58.450000	101.650000	180.200000	235.100000	422.75000
50%	158.000000	1967.000000	39.600000	85.150000	139.900000	245.400000	307.200000	537.20000
75%	182.000000	1991.000000	62.075000	118.600000	187.100000	335.300000	435.300000	834.77500
max	206.000000	2015.000000	164.500000	208.500000	605.600000	595.100000	1168.600000	1609.90000

shape

In [6]:
```
np.shape(df)
```

Out[6]:
```
(97, 20)
```

size

In [7]:
```
np.size(df)
```

Out[7]:
```
1940
```

dropna

In [8]:
```
df=df.dropna()
```

columns

In [9]:
```
df.columns
```

Out[9]:
```
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')
```

info()

In [10]:
```
df.info()
```

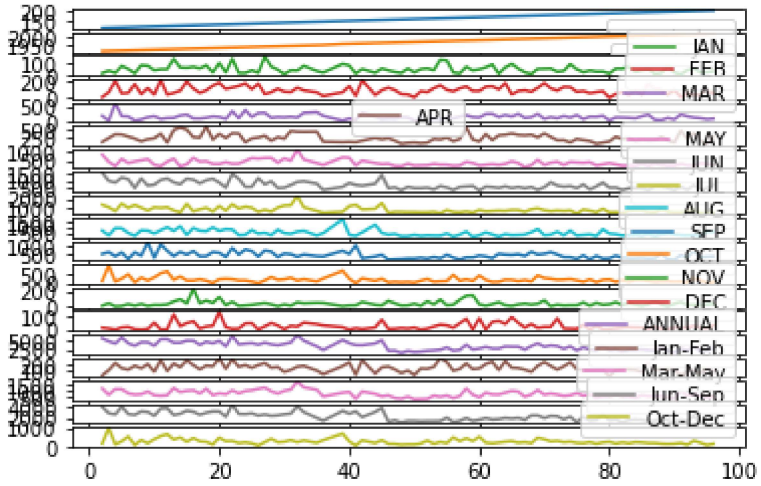
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 91 entries, 2 to 96
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   index        91 non-null     int64  
 1   SUBDIVISION  91 non-null     object 
 2   YEAR         91 non-null     int64  
 3   JAN          91 non-null     float64
 4   FEB          91 non-null     float64
 5   MAR          91 non-null     float64
 6   APR          91 non-null     float64
 7   MAY          91 non-null     float64
 8   JUN          91 non-null     float64
 9   JUL          91 non-null     float64
 10  AUG          91 non-null     float64
 11  SEP          91 non-null     float64
 12  OCT          91 non-null     float64
 13  NOV          91 non-null     float64
 14  DEC          91 non-null     float64
 15  ANNUAL       91 non-null     float64
 16  Jan-Feb      91 non-null     float64
```

```
 17  Mar-May      91 non-null     float64
 18  Jun-Sep      91 non-null     float64
 19  Oct-Dec      91 non-null     float64
dtypes: float64(17), int64(2), object(1)
memory usage: 14.9+ KB
```

Line chart

In [11]:
```
df.plot.line(subplots=True)
```

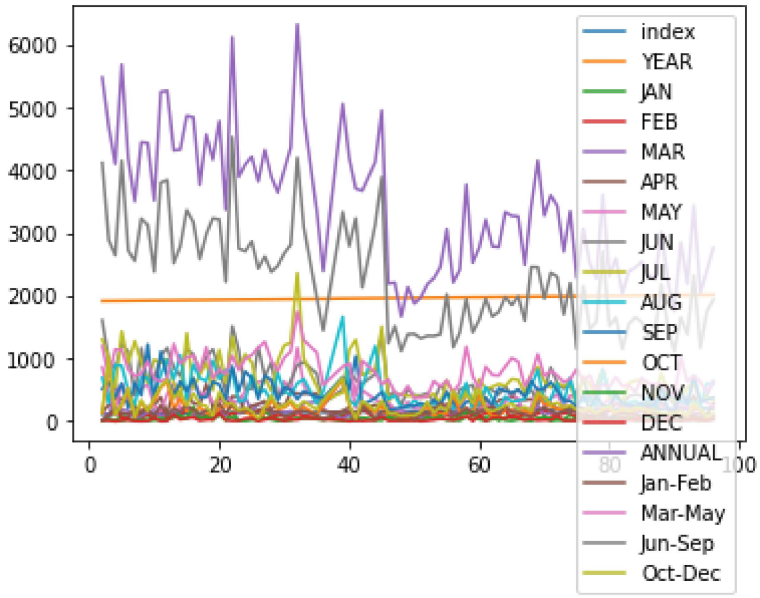
Out[11]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)
```



Line chart

In [12]:
```
df.plot.line()
```

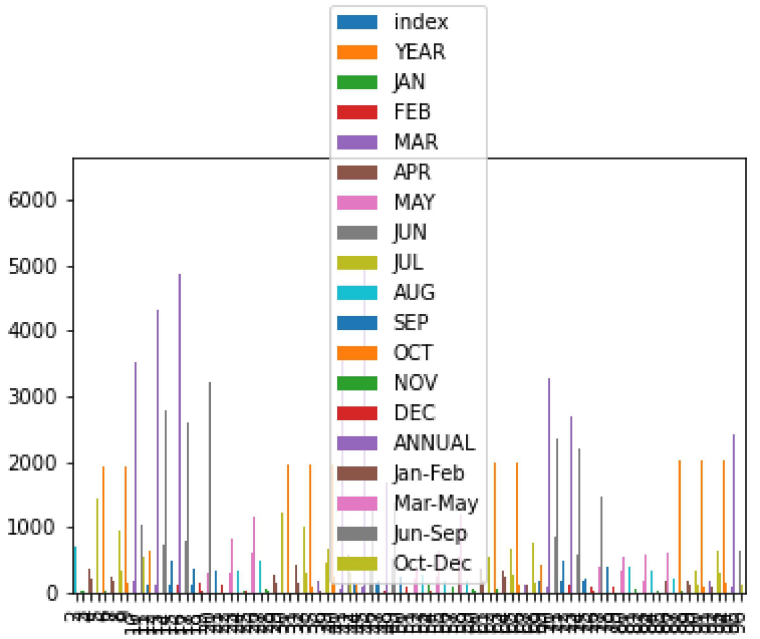
Out[12]:
```
<AxesSubplot:>
```



Bar chart

In [13]:
```
df.plot.bar()
```

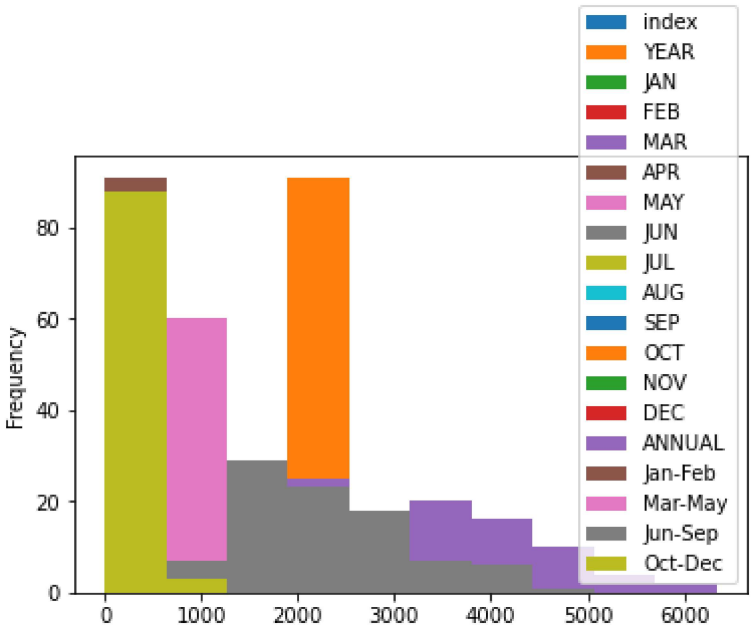
Out[13]: <AxesSubplot:>



Histogram

In [14]:
```
df.plot.hist()
```

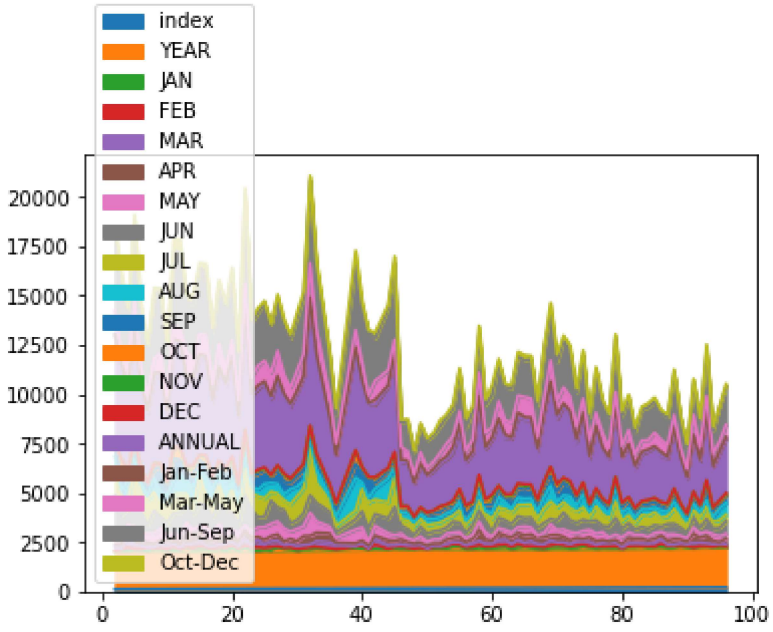
Out[14]: <AxesSubplot:ylabel='Frequency'>



Area chart

In [15]:
```
df.plot.area()
```

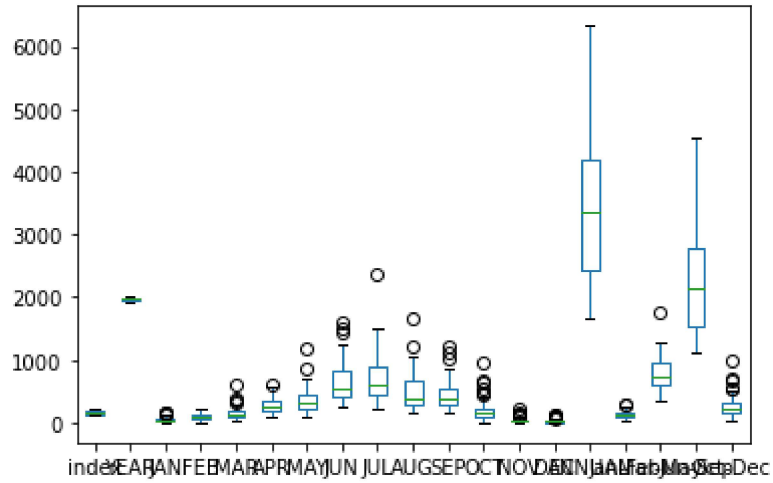
Out[15]: <AxesSubplot:>



Box chart

In [16]:
```
df.plot.box()
```

Out[16]: <AxesSubplot:>

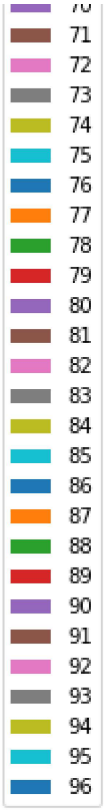


Pie chart

In [17]:
```
df.plot.pie(y='ANNUAL')
```

Out[17]:
```
<AxesSubplot:ylabel='ANNUAL'>
```

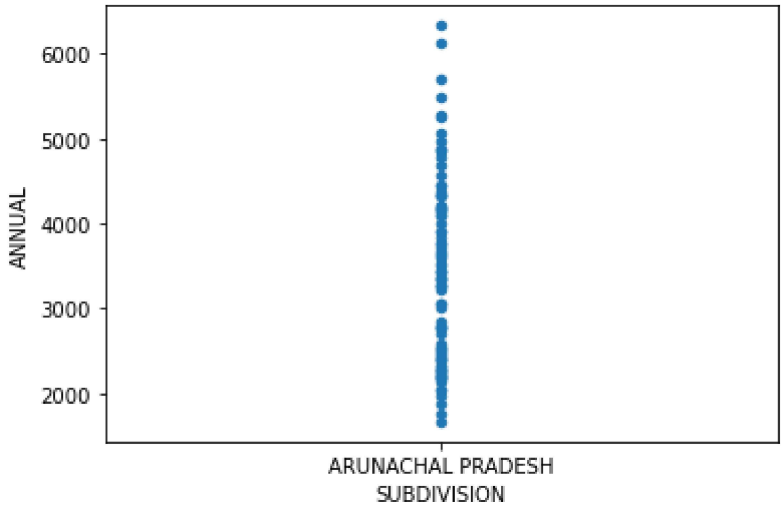

ANNUAL
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
35
36
39
40
41
42
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69



Scatter chart

In [18]:
```
df.plot.scatter(x='SUBDIVISION' ,y='ANNUAL')
```

Out[18]: <AxesSubplot:xlabel='SUBDIVISION', ylabel='ANNUAL'>



In [19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 91 entries, 2 to 96
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   index        91 non-null     int64
```

```
 1   SUBDIVISION  91 non-null     object
 2   YEAR         91 non-null     int64
 3   JAN          91 non-null     float64
 4   FEB          91 non-null     float64
 5   MAR          91 non-null     float64
 6   APR          91 non-null     float64
 7   MAY          91 non-null     float64
 8   JUN          91 non-null     float64
 9   JUL          91 non-null     float64
 10  AUG          91 non-null     float64
 11  SEP          91 non-null     float64
 12  OCT          91 non-null     float64
 13  NOV          91 non-null     float64
 14  DEC          91 non-null     float64
 15  ANNUAL       91 non-null     float64
 16  Jan-Feb      91 non-null     float64
 17  Mar-May      91 non-null     float64
 18  Jun-Sep      91 non-null     float64
 19  Oct-Dec      91 non-null     float64
dtypes: float64(17), int64(2), object(1)
memory usage: 14.9+ KB
```

In [20]:
```
df.describe()
```

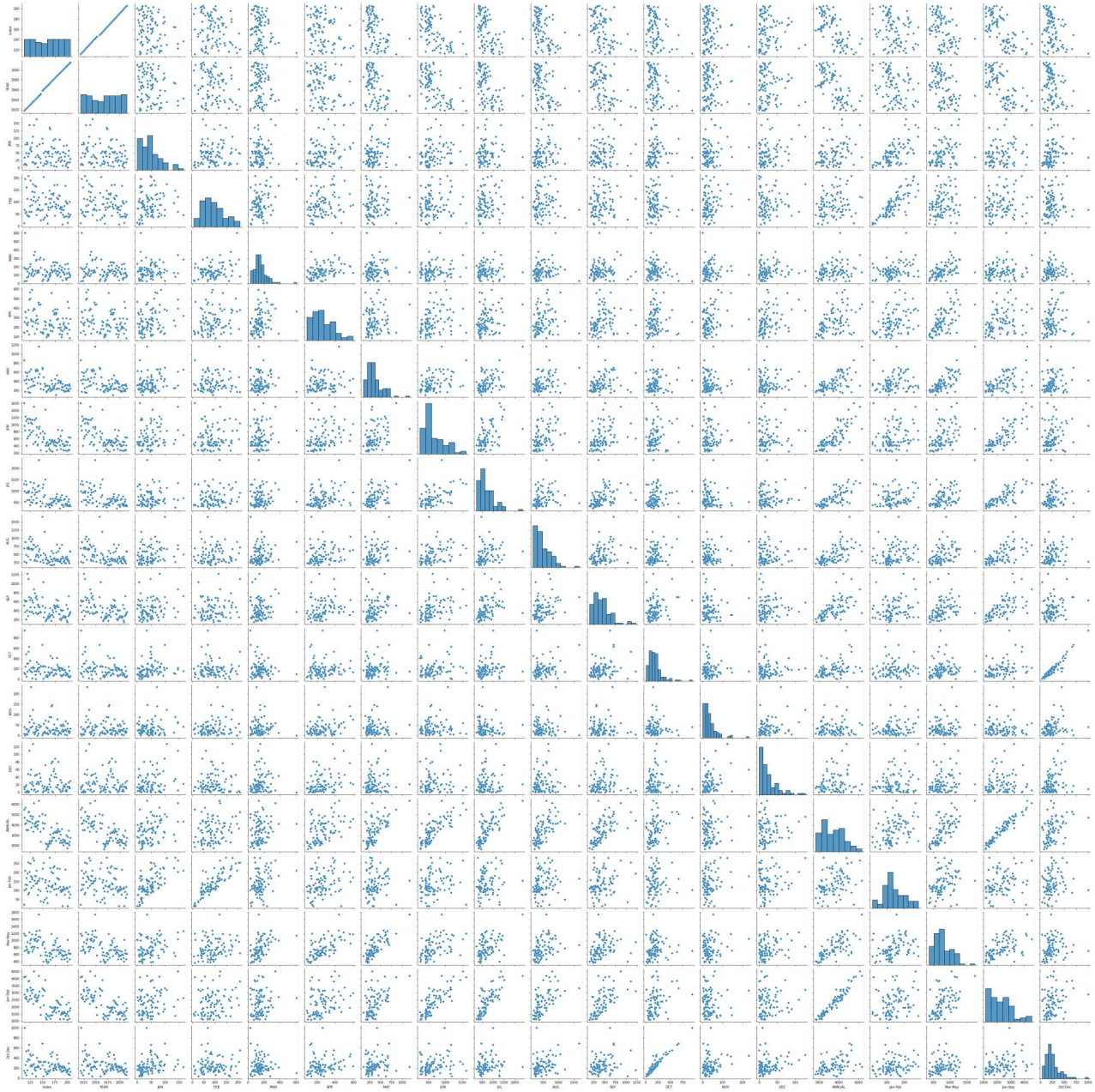
Out[20]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUI
count	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000	91.00000
mean	159.483516	1967.362637	47.680220	90.396703	154.143956	262.297802	358.289011	638.37802
std	28.065939	29.324437	35.045676	47.178011	86.284987	116.737705	178.900132	306.72096
min	112.000000	1918.000000	0.600000	6.100000	28.500000	86.700000	101.800000	239.40000
25%	134.500000	1940.500000	19.100000	55.250000	102.700000	177.500000	232.950000	421.90000
50%	161.000000	1970.000000	40.000000	83.200000	139.900000	240.800000	306.900000	530.20000
75%	183.500000	1992.500000	64.900000	118.900000	182.450000	341.200000	433.600000	823.00000
max	206.000000	2015.000000	164.500000	208.500000	605.600000	595.100000	1168.600000	1609.90000

EDA AND VISUALIZATION

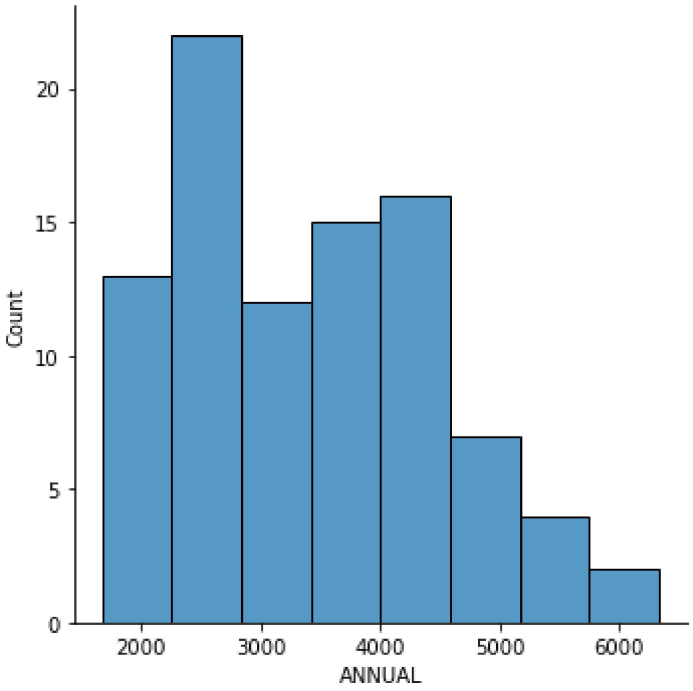
In [21]:
```
sns.pairplot(df)
```

Out[21]:
```
<seaborn.axisgrid.PairGrid at 0x1af5e335790>
```



In [22]:
```
sns.displot(df['ANNUAL'])
```

Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x1af696dbf40>
```



In [23]:
```
sns.heatmap(df.corr())
```

Out[23]: <AxesSubplot:>