

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_west rajasthan.csv")
df
```

Out[2]:

		index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	D
0	1817		WEST RAJASTHAN	1901	6.7	0.0	1.1	0.0	6.1	3.0	79.0	59.2	1.0	2.1	0.0	
1	1818		WEST RAJASTHAN	1902	0.0	0.0	0.0	0.5	4.0	49.1	27.0	71.3	41.8	1.8	0.0	
2	1819		WEST RAJASTHAN	1903	1.7	1.3	5.5	0.0	4.2	2.7	154.8	87.1	49.3	0.1	0.0	
3	1820		WEST RAJASTHAN	1904	3.8	2.9	16.3	0.7	11.4	14.6	39.8	45.6	21.4	1.4	2.9	
4	1821		WEST RAJASTHAN	1905	6.3	4.8	0.7	1.3	0.3	4.9	30.1	0.6	64.5	0.0	0.0	
...
110	1927		WEST RAJASTHAN	2011	0.0	11.8	1.5	1.5	7.8	24.4	88.5	166.8	116.3	0.1	0.0	
111	1928		WEST RAJASTHAN	2012	0.5	0.0	0.0	9.5	10.4	5.3	40.4	166.7	92.0	1.9	0.0	
112	1929		WEST RAJASTHAN	2013	8.6	21.8	4.2	3.1	1.7	37.6	104.5	138.2	58.7	10.1	1.0	
113	1930		WEST RAJASTHAN	2014	0.8	2.2	4.7	8.4	23.0	13.8	94.3	69.6	84.9	0.5	0.2	
114	1931		WEST RAJASTHAN	2015	1.4	0.9	30.3	25.2	15.5	53.2	234.6	60.5	35.7	1.1	0.1	

115 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	NOV	D
0	1817		WEST RAJASTHAN	1901	6.7	0.0	1.1	0.0	6.1	3.0	79.0	59.2	1.0	2.1	0.0	
1	1818		WEST RAJASTHAN	1902	0.0	0.0	0.0	0.5	4.0	49.1	27.0	71.3	41.8	1.8	0.0	
2	1819		WEST RAJASTHAN	1903	1.7	1.3	5.5	0.0	4.2	2.7	154.8	87.1	49.3	0.1	0.0	
3	1820		WEST RAJASTHAN	1904	3.8	2.9	16.3	0.7	11.4	14.6	39.8	45.6	21.4	1.4	2.9	
4	1821		WEST RAJASTHAN	1905	6.3	4.8	0.7	1.3	0.3	4.9	30.1	0.6	64.5	0.0	0.0	
...
110	1927		WEST RAJASTHAN	2011	0.0	11.8	1.5	1.5	7.8	24.4	88.5	166.8	116.3	0.1	0.0	
111	1928		WEST RAJASTHAN	2012	0.5	0.0	0.0	9.5	10.4	5.3	40.4	166.7	92.0	1.9	0.0	
112	1929		WEST RAJASTHAN	2013	8.6	21.8	4.2	3.1	1.7	37.6	104.5	138.2	58.7	10.1	1.0	
113	1930		WEST RAJASTHAN	2014	0.8	2.2	4.7	8.4	23.0	13.8	94.3	69.6	84.9	0.5	0.2	
114	1931		WEST RAJASTHAN	2015	1.4	0.9	30.3	25.2	15.5	53.2	234.6	60.5	35.7	1.1	0.1	

115 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	NOV	D
0	1817		WEST RAJASTHAN	1901	6.7	0.0	1.1	0.0	6.1	3.0	79.0	59.2	1.0	2.1	0.0	
1	1818		WEST RAJASTHAN	1902	0.0	0.0	0.0	0.5	4.0	49.1	27.0	71.3	41.8	1.8	0.0	
2	1819		WEST RAJASTHAN	1903	1.7	1.3	5.5	0.0	4.2	2.7	154.8	87.1	49.3	0.1	0.0	

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	D
3	1820	WEST RAJASTHAN	1904	3.8	2.9	16.3	0.7	11.4	14.6	39.8	45.6	21.4	1.4	2.9	
4	1821	WEST RAJASTHAN	1905	6.3	4.8	0.7	1.3	0.3	4.9	30.1	0.6	64.5	0.0	0.0	
...
110	1927	WEST RAJASTHAN	2011	0.0	11.8	1.5	1.5	7.8	24.4	88.5	166.8	116.3	0.1	0.0	
111	1928	WEST RAJASTHAN	2012	0.5	0.0	0.0	9.5	10.4	5.3	40.4	166.7	92.0	1.9	0.0	
112	1929	WEST RAJASTHAN	2013	8.6	21.8	4.2	3.1	1.7	37.6	104.5	138.2	58.7	10.1	1.0	
113	1930	WEST RAJASTHAN	2014	0.8	2.2	4.7	8.4	23.0	13.8	94.3	69.6	84.9	0.5	0.2	
114	1931	WEST RAJASTHAN	2015	1.4	0.9	30.3	25.2	15.5	53.2	234.6	60.5	35.7	1.1	0.1	

115 rows × 20 columns

Data Cleaning and Data Preprocessing

describe()

In [5]:

```
df.describe()
```

Out[5]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	1874.000000	1958.000000	3.327826	4.930435	3.986087	3.571304	9.443478	28.637391
std	33.341666	33.341666	4.551914	7.858800	7.813965	5.916803	10.853168	22.809360
min	1817.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.400000
25%	1845.500000	1929.500000	0.350000	0.200000	0.200000	0.400000	1.950000	13.300000
50%	1874.000000	1958.000000	1.600000	1.300000	1.100000	1.400000	6.100000	21.900000
75%	1902.500000	1986.500000	4.000000	5.950000	5.200000	3.750000	12.150000	39.650000
max	1931.000000	2015.000000	21.400000	39.100000	59.000000	36.100000	56.800000	143.200000

shape

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

```
Out[6]: (115, 20)
```

size

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

```
Out[7]: 2300
```

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: 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
```

```

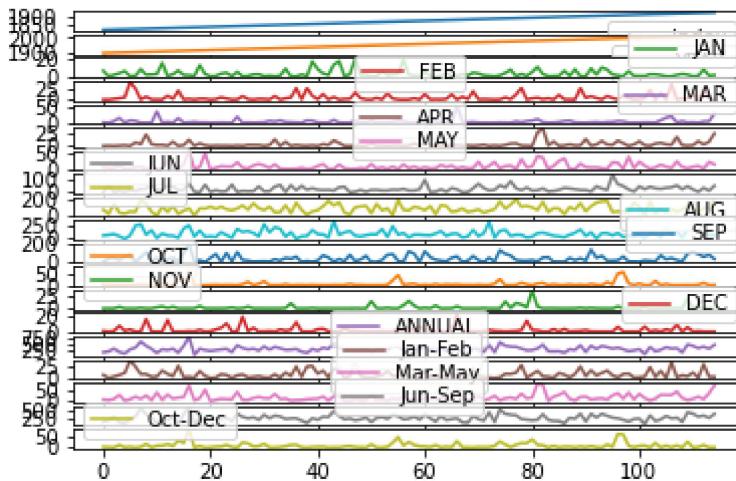
17 Mar-May      115 non-null    float64
18 Jun-Sep      115 non-null    float64
19 Oct-Dec      115 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 18.9+ KB

```

Line chart

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

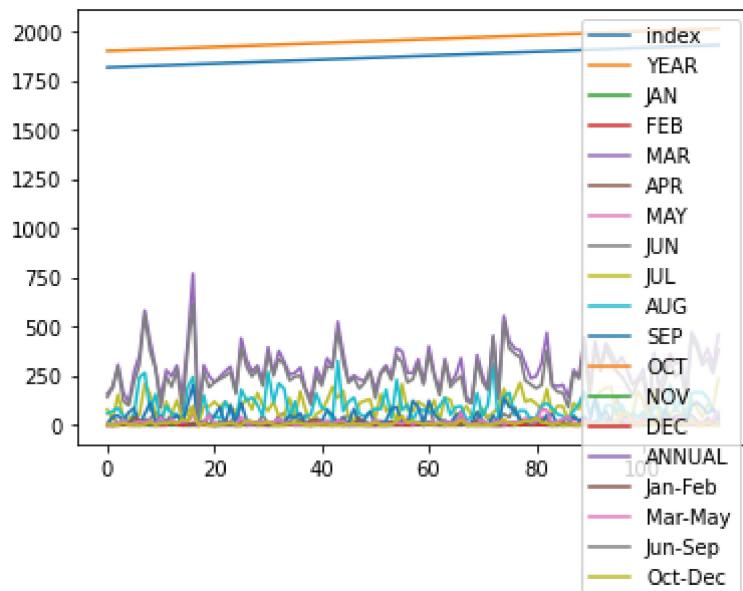
```
Out[11]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)
```



Line chart

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

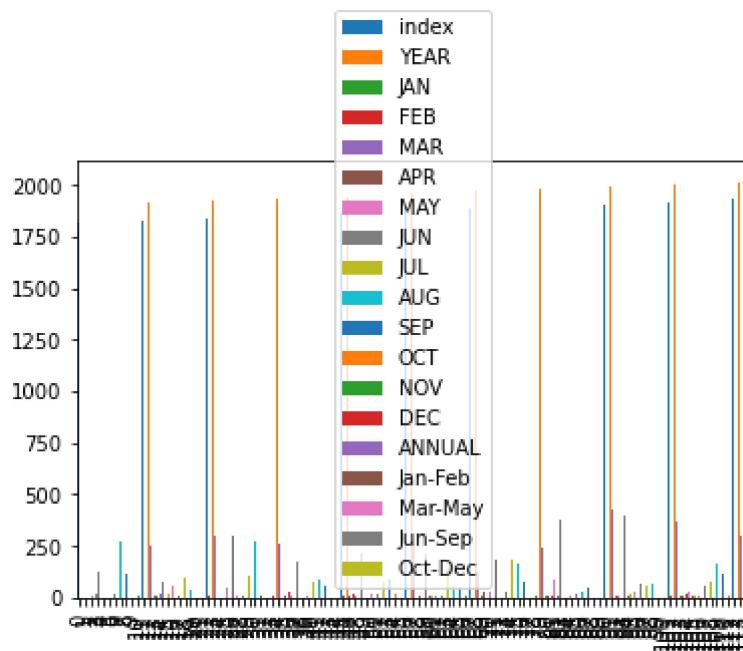
```
Out[12]: <AxesSubplot:>
```



Bar chart

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

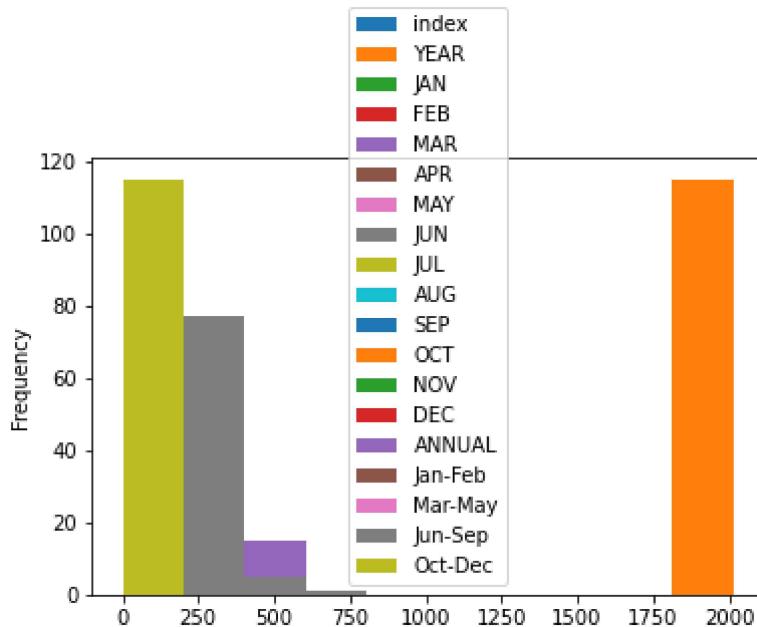
```
Out[13]: <AxesSubplot:>
```



Histogram

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

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

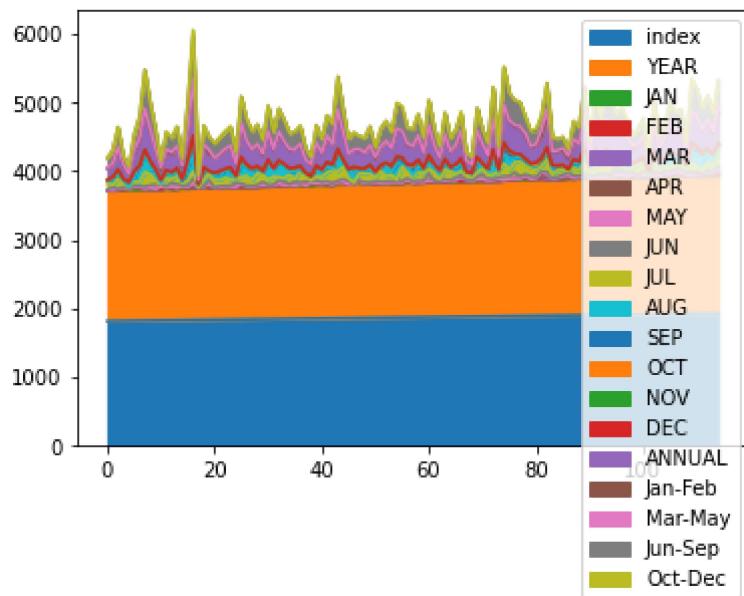


Area chart

In [15]:

```
df.plot.area()
```

Out[15]: <AxesSubplot:>

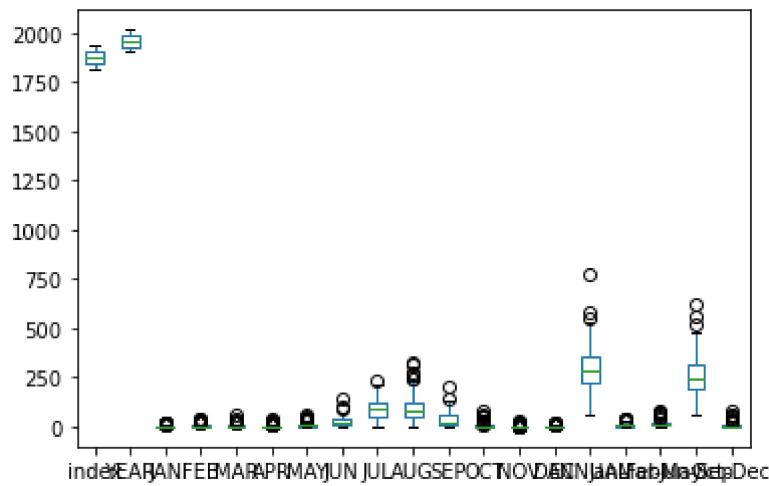


Box chart

In [16]:

```
df.plot.box()
```

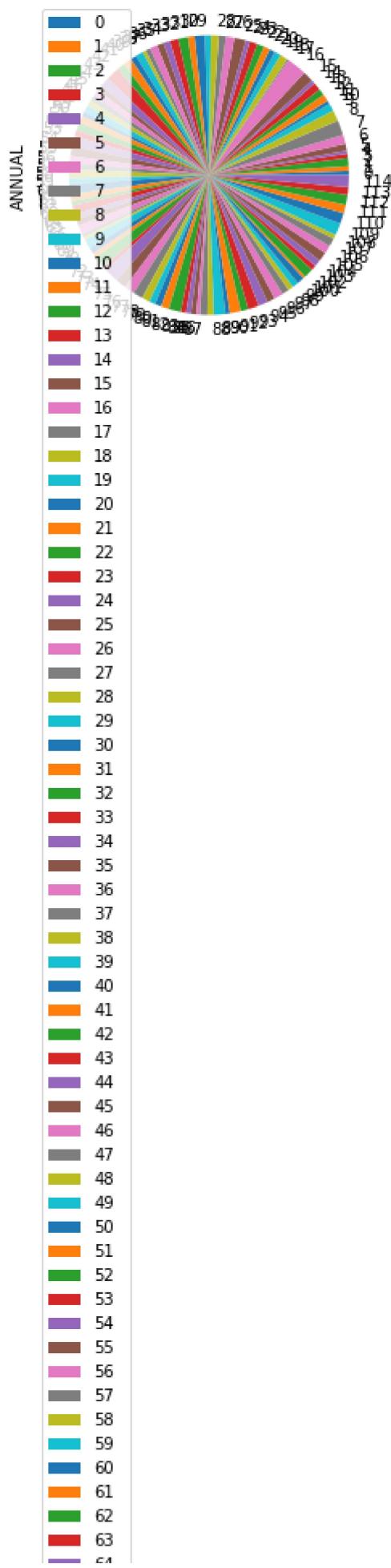
Out[16]: <AxesSubplot:>

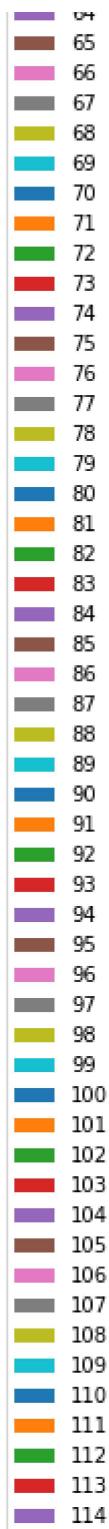


Pie chart

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

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

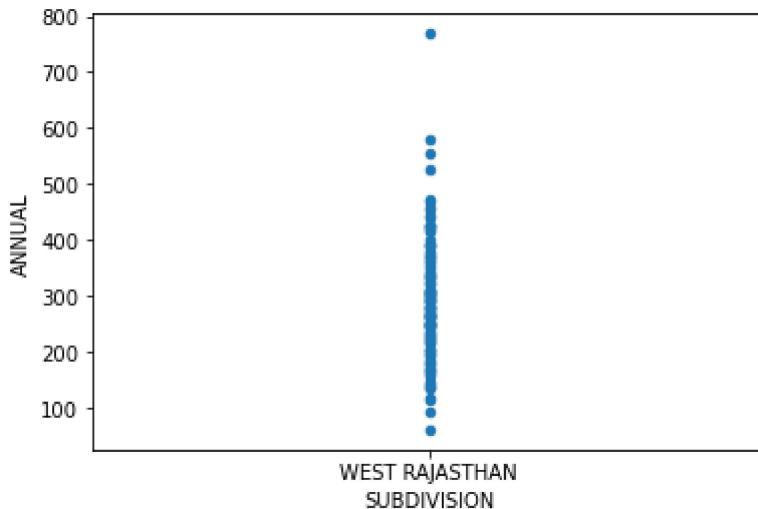




Scatter chart

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: 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 [20]:

```
df.describe()
```

Out[20]:

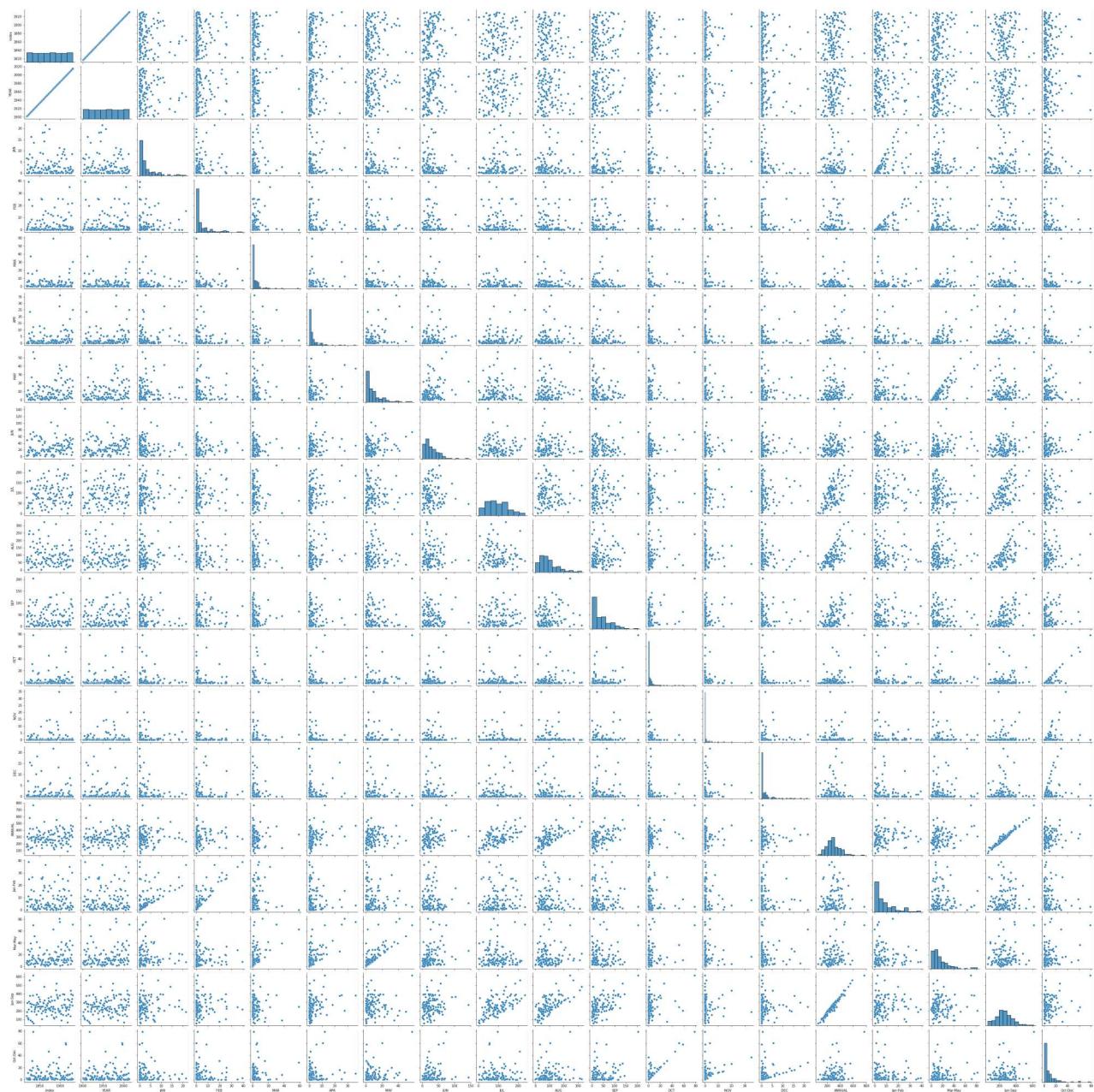
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	1874.000000	1958.000000	3.327826	4.930435	3.986087	3.571304	9.443478	28.637391
std	33.341666	33.341666	4.551914	7.858800	7.813965	5.916803	10.853168	22.809360
min	1817.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.400000
25%	1845.500000	1929.500000	0.350000	0.200000	0.200000	0.400000	1.950000	13.300000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
50%	1874.000000	1958.000000	1.600000	1.300000	1.100000	1.400000	6.100000	21.900000
75%	1902.500000	1986.500000	4.000000	5.950000	5.200000	3.750000	12.150000	39.650000
max	1931.000000	2015.000000	21.400000	39.100000	59.000000	36.100000	56.800000	143.200000

EDA AND VISUALIZATION

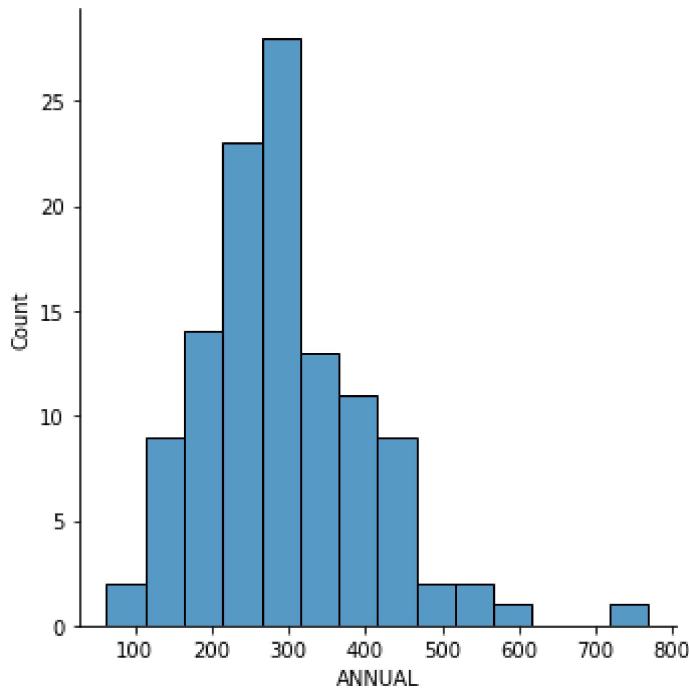
In [21]: `sns.pairplot(df)`

Out[21]: <seaborn.axisgrid.PairGrid at 0x1d2f7c63910>



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

Out[22]: <seaborn.axisgrid.FacetGrid at 0x1d2fe6466640>

In [23]:

```
sns.heatmap(df.corr())
```

Out[23]: <AxesSubplot:>

