

Import Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\FP2_RainFall\rain.csv")[784:897]
d
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
784	784	JHARKHAND	1903	25.1	19.5	10.7	32.8	56.4	142.1	206.1	280.8	190.2
785	785	JHARKHAND	1904	2.5	17.0	38.1	9.1	116.1	308.9	494.1	336.1	125.6
786	786	JHARKHAND	1905	38.4	53.3	61.6	32.9	66.2	41.5	420.3	293.7	322.8
787	787	JHARKHAND	1906	45.8	128.5	35.9	1.7	33.9	170.9	355.6	251.6	185.4
788	788	JHARKHAND	1907	2.0	59.3	76.0	26.3	27.4	338.1	230.4	469.4	200.2
...
892	892	JHARKHAND	2011	3.3	2.5	6.4	25.4	55.0	349.0	181.8	403.2	324.6
893	893	JHARKHAND	2012	34.6	10.3	1.5	9.6	6.6	121.1	287.2	282.4	217.6
894	894	JHARKHAND	2013	1.1	17.9	1.6	22.3	85.0	181.5	211.1	278.1	173.8
895	895	JHARKHAND	2014	9.9	47.5	22.9	1.9	98.2	139.7	321.3	290.9	178.2
896	896	JHARKHAND	2015	12.2	2.6	21.6	55.5	25.5	183.3	429.7	240.7	85.1

113 rows × 20 columns



Data Cleaning and preprocessing

In [3]:

```
d.dropna()
```

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP
784	784	JHARKHAND	1903	25.1	19.5	10.7	32.8	56.4	142.1	206.1	280.8	190.2
785	785	JHARKHAND	1904	2.5	17.0	38.1	9.1	116.1	308.9	494.1	336.1	125.6
786	786	JHARKHAND	1905	38.4	53.3	61.6	32.9	66.2	41.5	420.3	293.7	322.8
787	787	JHARKHAND	1906	45.8	128.5	35.9	1.7	33.9	170.9	355.6	251.6	185.4
788	788	JHARKHAND	1907	2.0	59.3	76.0	26.3	27.4	338.1	230.4	469.4	200.2
...
892	892	JHARKHAND	2011	3.3	2.5	6.4	25.4	55.0	349.0	181.8	403.2	324.6
893	893	JHARKHAND	2012	34.6	10.3	1.5	9.6	6.6	121.1	287.2	282.4	217.6
894	894	JHARKHAND	2013	1.1	17.9	1.6	22.3	85.0	181.5	211.1	278.1	173.8
895	895	JHARKHAND	2014	9.9	47.5	22.9	1.9	98.2	139.7	321.3	290.9	178.2
896	896	JHARKHAND	2015	12.2	2.6	21.6	55.5	25.5	183.3	429.7	240.7	85.1

113 rows × 20 columns

In [4]:

```
d.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]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113 entries, 784 to 896
Data columns (total 20 columns):
#   Column          Non-Null Count  Dtype
---  -
0   index           113 non-null    int64
1   SUBDIVISION     113 non-null    object
2   YEAR            113 non-null    int64
3   JAN             113 non-null    float64
4   FEB             113 non-null    float64
5   MAR             113 non-null    float64
6   APR             113 non-null    float64
7   MAY             113 non-null    float64
8   JUN             113 non-null    float64
9   JUL             113 non-null    float64
10  AUG             113 non-null    float64
11  SEP             113 non-null    float64
12  OCT             113 non-null    float64
13  NOV             113 non-null    float64
14  DEC             113 non-null    float64
15  ANNUAL          113 non-null    float64
16  Jan-Feb         113 non-null    float64
17  Mar-May         113 non-null    float64
18  Jun-Sep         113 non-null    float64
19  Oct-Dec         113 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 17.8+ KB
```

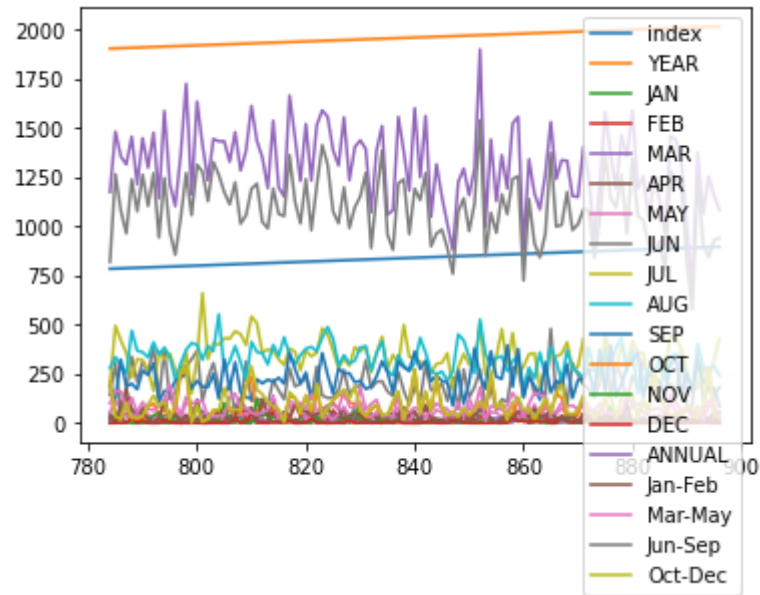
Line Chart

In [6]:

```
d.plot.line()
```

Out[6]:

<AxesSubplot:>



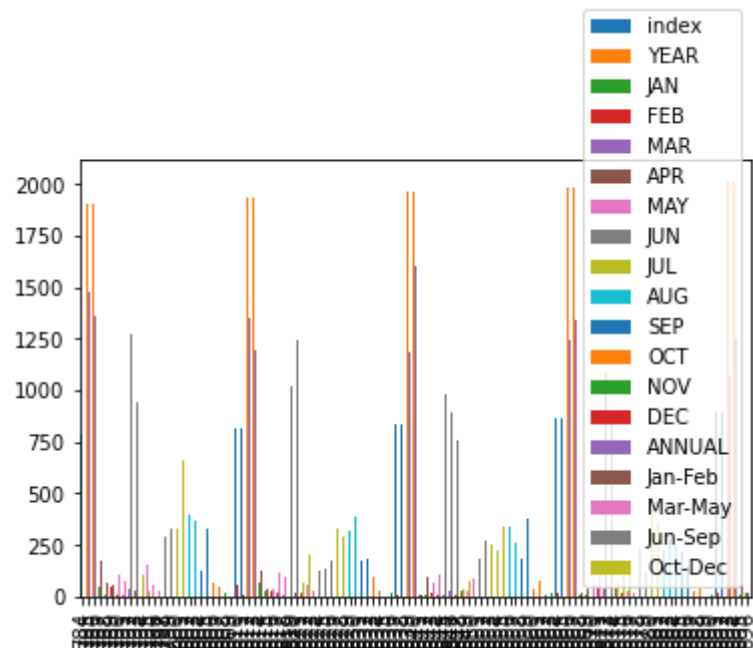
Bar Chart

In [7]:

```
d.plot.bar()
```

Out[7]:

<AxesSubplot:>



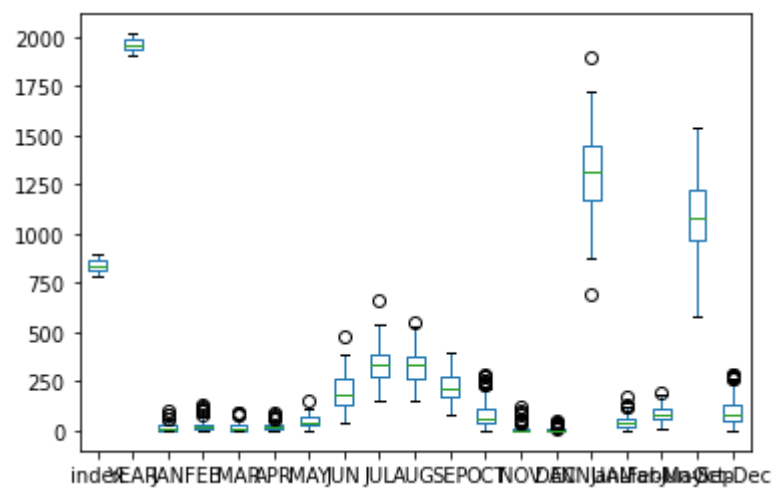
Box Chart

In [8]:

```
d.plot.box()
```

Out[8]:

<AxesSubplot:>



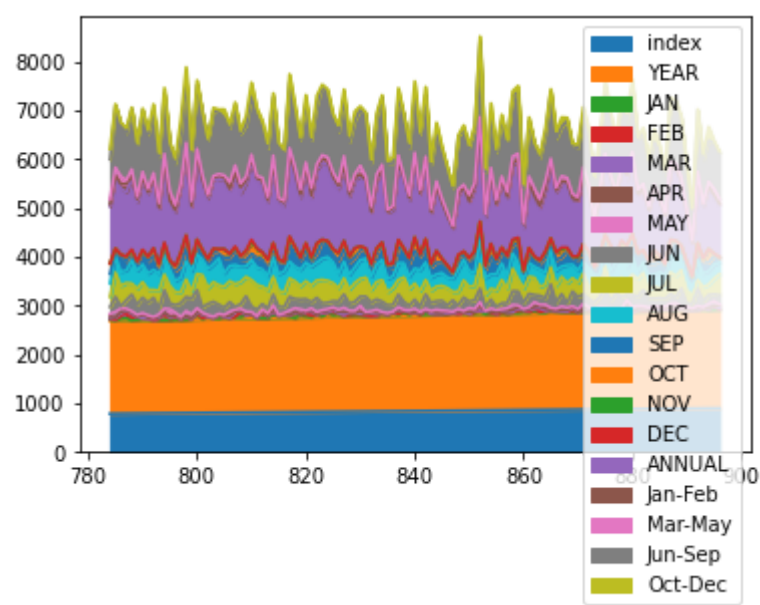
Area Chart

In [9]:

```
d.plot.area()
```

Out[9]:

<AxesSubplot:>



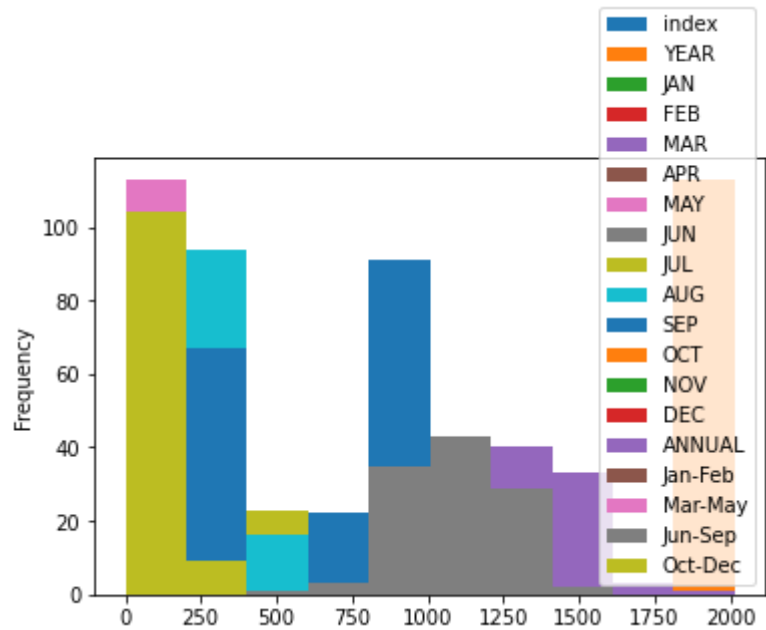
Histogram

In [10]:

```
d.plot.hist()
```

Out[10]:

<AxesSubplot:ylabel='Frequency'>



Pie Chart

In [11]:

```
d.plot.pie(y='ANNUAL')
```

Out[11]:

<AxesSubplot:ylabel='ANNUAL'>

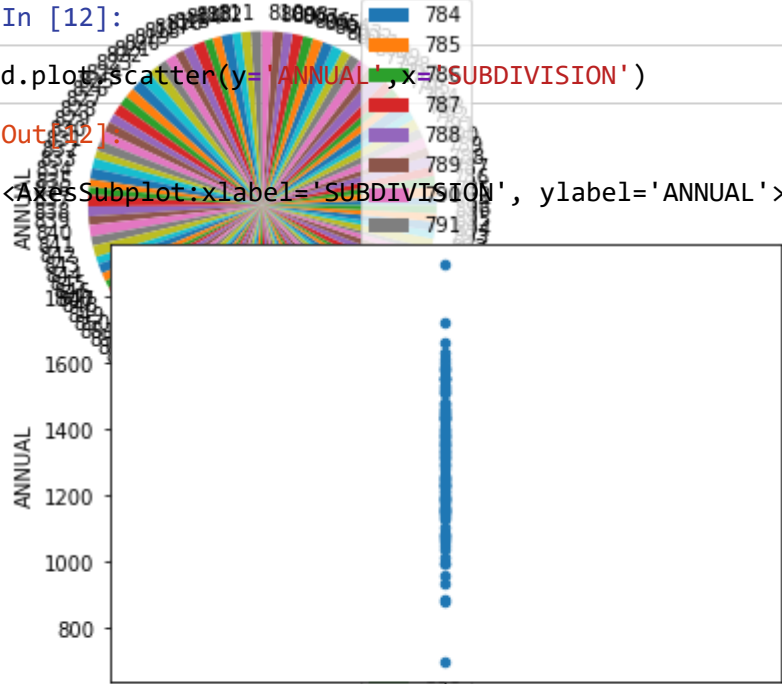
Scatter Chart

In [12]:

```
d.plot.scatter(y='ANNUAL',x='SUBDIVISION')
```

Out[12]:

```
<AxesSubplot:xlabel='SUBDIVISION', ylabel='ANNUAL'>
```



In [13]:

```
d.describe()
```

Out[13]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	113.00000	113.00000	113.00000	113.00000	113.00000	113.00000	113.00000	11
mean	840.00000	1959.00000	17.076106	23.956637	18.534513	19.294690	48.346903	19
std	32.76431	32.76431	19.368602	25.649867	20.026165	16.269894	28.852819	8
min	784.00000	1903.00000	0.000000	0.100000	0.000000	0.100000	3.200000	4
25%	812.00000	1931.00000	2.300000	7.800000	3.100000	8.000000	26.700000	12
50%	840.00000	1959.00000	9.500000	16.200000	10.900000	15.600000	42.300000	18
75%	868.00000	1987.00000	24.700000	32.800000	28.000000	26.300000	66.400000	25
max	896.00000	2015.00000	102.900000	128.500000	95.000000	95.600000	150.100000	47

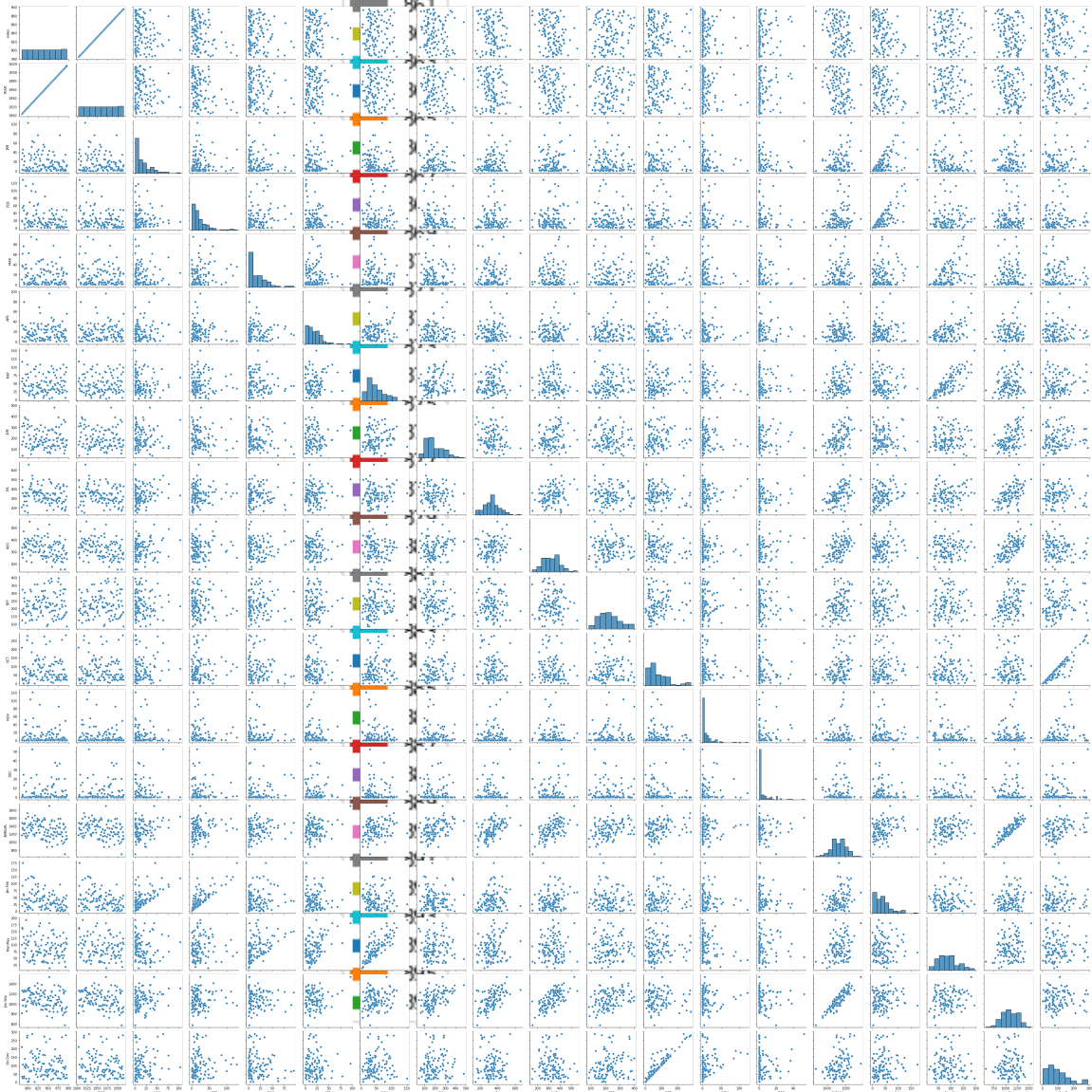
EDA AND VISUALIZATION

In [14]:

```
sns.pairplot(d)
```

Out[14]:

<seaborn.axisgrid.PairGrid at 0x1b844522dc0>

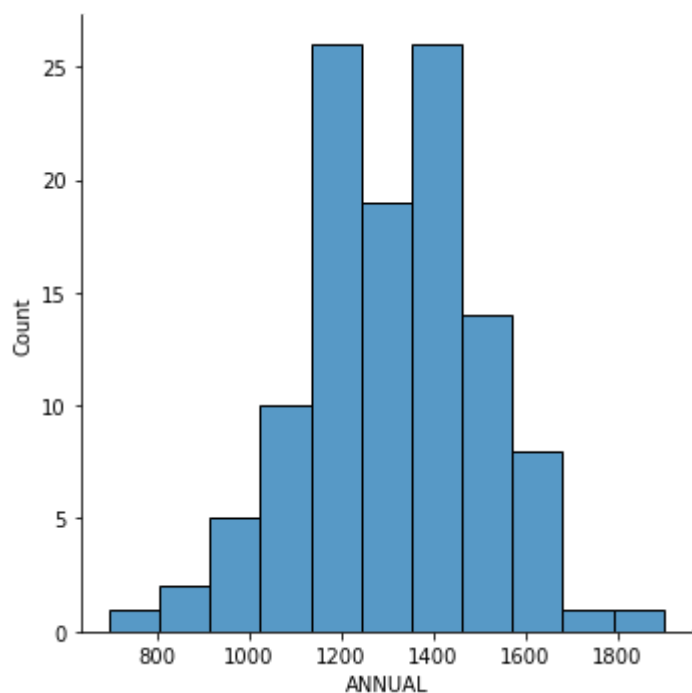


In [15]:

```
sns.displot(d['ANNUAL'])
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x1b850a32b50>
```

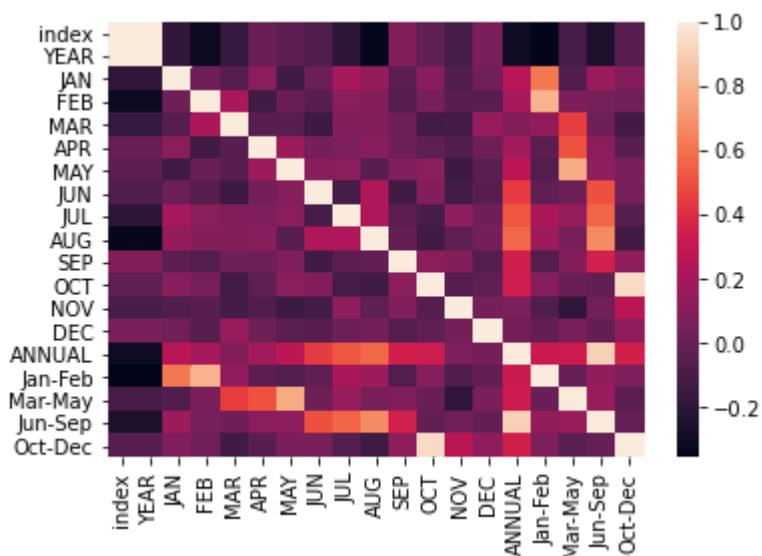


In [16]:

```
sns.heatmap(d.corr())
```

Out[16]:

```
<AxesSubplot:>
```



In []:

