

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_gangetic west bengal.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	552	GANGETIC WEST BENGAL	1901	37.1	58.4	3.9	64.1	121.7	198.0	280.8	275.7	313.5	51.1	83.4
1	553	GANGETIC WEST BENGAL	1902	0.0	1.2	44.2	103.8	161.6	140.9	347.8	264.8	230.5	32.5	10.4
2	554	GANGETIC WEST BENGAL	1903	17.5	24.6	37.3	30.6	78.5	201.7	179.6	277.6	300.7	198.0	8.2
3	555	GANGETIC WEST BENGAL	1904	0.1	23.9	35.6	17.5	160.2	286.7	435.3	241.7	142.8	35.1	4.1
4	556	GANGETIC WEST BENGAL	1905	30.9	49.6	84.7	84.9	156.8	70.9	525.5	263.6	287.6	107.3	0.0
...
110	662	GANGETIC WEST BENGAL	2011	2.5	2.7	40.5	75.0	132.6	434.5	219.9	443.2	295.9	36.9	1.3
111	663	GANGETIC WEST BENGAL	2012	40.7	15.3	4.4	57.7	44.2	146.6	315.0	261.4	246.9	64.2	47.0
112	664	GANGETIC WEST BENGAL	2013	2.5	10.0	4.8	45.6	195.9	233.4	263.2	401.4	254.0	353.2	0.0
113	665	GANGETIC WEST BENGAL	2014	0.9	42.2	19.9	1.9	124.4	193.6	298.7	292.6	229.5	56.9	0.1
114	666	GANGETIC WEST BENGAL	2015	12.9	5.5	19.3	88.7	57.6	247.2	633.1	260.6	164.0	32.7	2.3

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
0	552		GANGETIC WEST BENGAL	1901	37.1	58.4	3.9	64.1	121.7	198.0	280.8	275.7	313.5	51.1	83.4
1	553		GANGETIC WEST BENGAL	1902	0.0	1.2	44.2	103.8	161.6	140.9	347.8	264.8	230.5	32.5	10.4
2	554		GANGETIC WEST BENGAL	1903	17.5	24.6	37.3	30.6	78.5	201.7	179.6	277.6	300.7	198.0	8.2
3	555		GANGETIC WEST BENGAL	1904	0.1	23.9	35.6	17.5	160.2	286.7	435.3	241.7	142.8	35.1	4.1
4	556		GANGETIC WEST BENGAL	1905	30.9	49.6	84.7	84.9	156.8	70.9	525.5	263.6	287.6	107.3	0.0
...
110	662		GANGETIC WEST BENGAL	2011	2.5	2.7	40.5	75.0	132.6	434.5	219.9	443.2	295.9	36.9	1.3
111	663		GANGETIC WEST BENGAL	2012	40.7	15.3	4.4	57.7	44.2	146.6	315.0	261.4	246.9	64.2	47.0
112	664		GANGETIC WEST BENGAL	2013	2.5	10.0	4.8	45.6	195.9	233.4	263.2	401.4	254.0	353.2	0.0
113	665		GANGETIC WEST BENGAL	2014	0.9	42.2	19.9	1.9	124.4	193.6	298.7	292.6	229.5	56.9	0.1
114	666		GANGETIC WEST BENGAL	2015	12.9	5.5	19.3	88.7	57.6	247.2	633.1	260.6	164.0	32.7	2.3

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
0	552		GANGETIC WEST BENGAL	1901	37.1	58.4	3.9	64.1	121.7	198.0	280.8	275.7	313.5	51.1	83.4
1	553		GANGETIC WEST BENGAL	1902	0.0	1.2	44.2	103.8	161.6	140.9	347.8	264.8	230.5	32.5	10.4
2	554		GANGETIC WEST BENGAL	1903	17.5	24.6	37.3	30.6	78.5	201.7	179.6	277.6	300.7	198.0	8.2
3	555		GANGETIC WEST BENGAL	1904	0.1	23.9	35.6	17.5	160.2	286.7	435.3	241.7	142.8	35.1	4.1
4	556		GANGETIC WEST BENGAL	1905	30.9	49.6	84.7	84.9	156.8	70.9	525.5	263.6	287.6	107.3	0.0
...
110	662		GANGETIC WEST BENGAL	2011	2.5	2.7	40.5	75.0	132.6	434.5	219.9	443.2	295.9	36.9	1.3
111	663		GANGETIC WEST BENGAL	2012	40.7	15.3	4.4	57.7	44.2	146.6	315.0	261.4	246.9	64.2	47.0
112	664		GANGETIC WEST BENGAL	2013	2.5	10.0	4.8	45.6	195.9	233.4	263.2	401.4	254.0	353.2	0.0
113	665		GANGETIC WEST BENGAL	2014	0.9	42.2	19.9	1.9	124.4	193.6	298.7	292.6	229.5	56.9	0.1
114	666		GANGETIC WEST BENGAL	2015	12.9	5.5	19.3	88.7	57.6	247.2	633.1	260.6	164.0	32.7	2.3

115 rows × 20 columns



Data Cleaning and Data Preprocessing

describe()

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

	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	609.000000	1958.000000	12.595652	22.452174	29.090435	44.885217	107.787826	247.196522
std	33.341666	33.341666	14.741821	24.165919	30.754415	31.812974	51.001443	99.676225
min	552.000000	1901.000000	0.000000	0.000000	0.100000	0.900000	16.400000	69.700000
25%	580.500000	1929.500000	1.250000	5.200000	7.200000	21.000000	71.900000	180.950000
50%	609.000000	1958.000000	6.800000	13.600000	18.900000	39.200000	98.900000	227.600000
75%	637.500000	1986.500000	18.350000	30.650000	42.250000	61.050000	131.400000	304.800000
max	666.000000	2015.000000	60.000000	123.600000	152.500000	174.200000	250.900000	597.100000



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()`

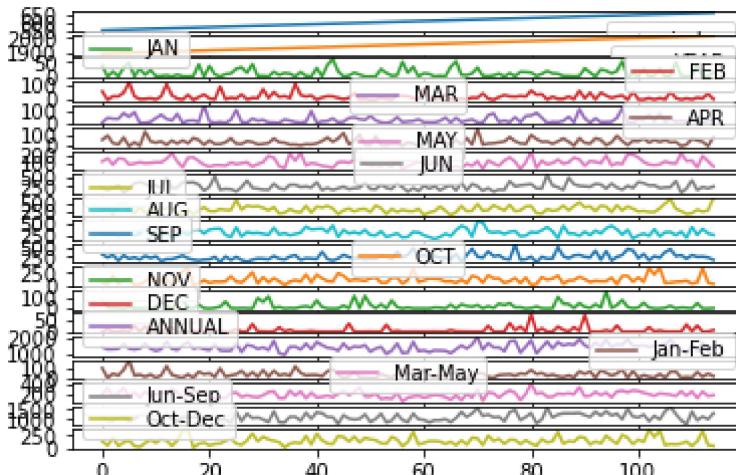
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 115 entries, 0 to 114
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   index       115 non-null    int64  
 1   SUBDIVISION 115 non-null    object  
 2   YEAR        115 non-null    int64  
 3   JAN         115 non-null    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)`

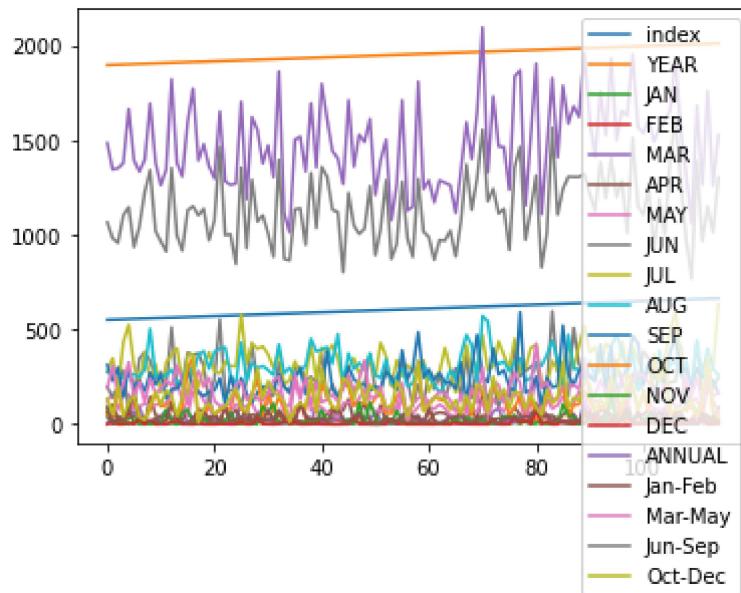
```
Out[11]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>,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



Line chart

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

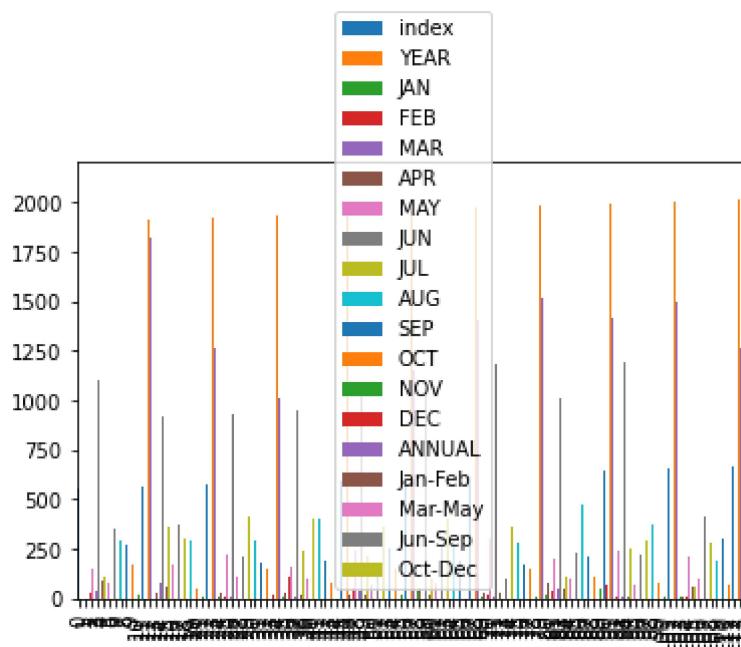
```
Out[12]: <AxesSubplot:>
```



Bar chart

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

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

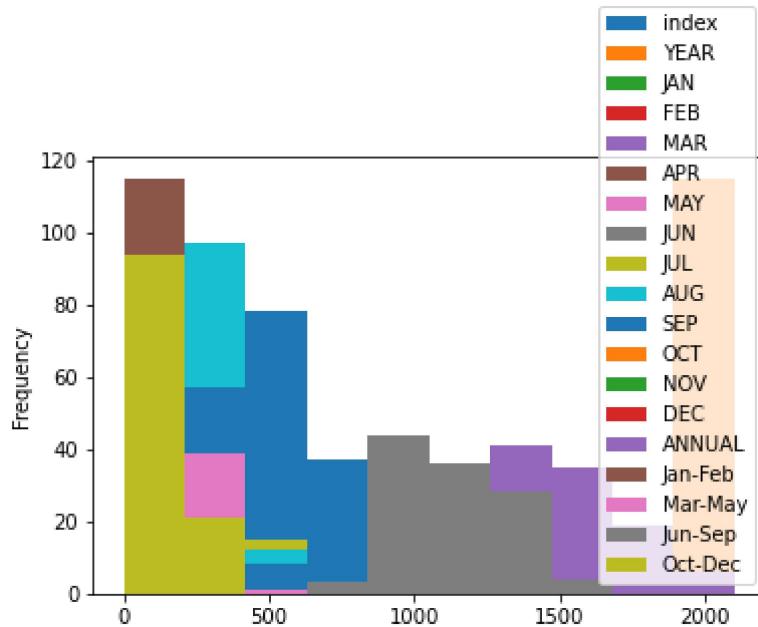


Histogram

In [14]:

```
df.plot.hist()
```

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

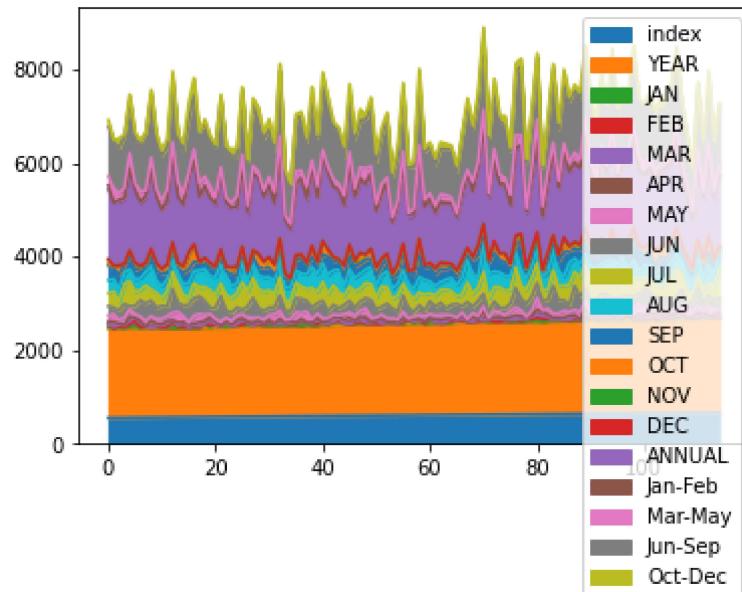


Area chart

In [15]:

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

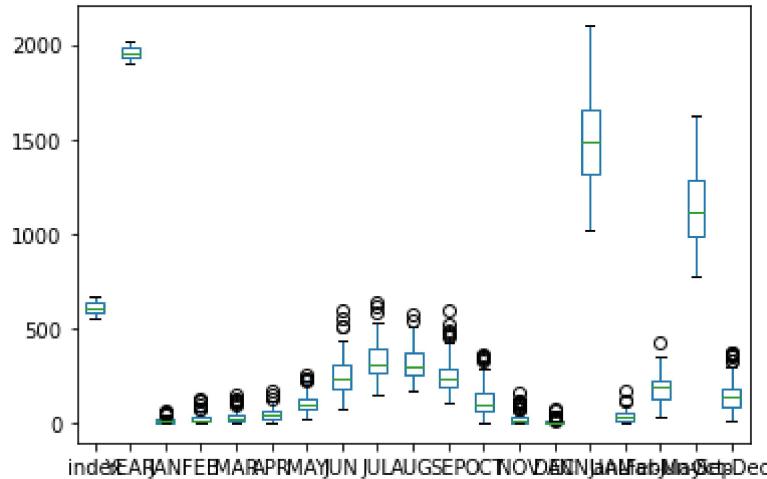
Out[15]: <AxesSubplot:>



Box chart

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

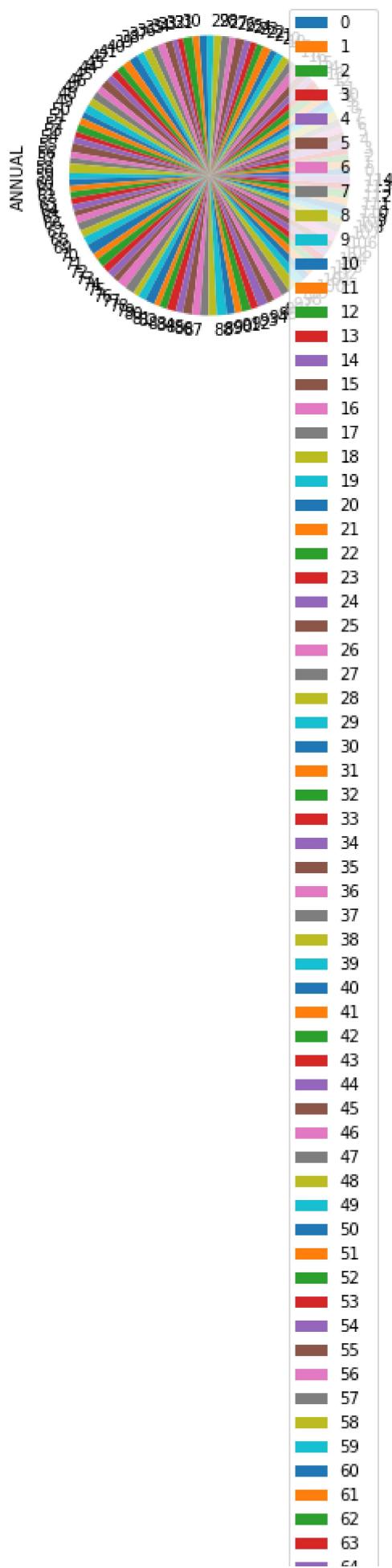
```
Out[16]: <AxesSubplot:
```

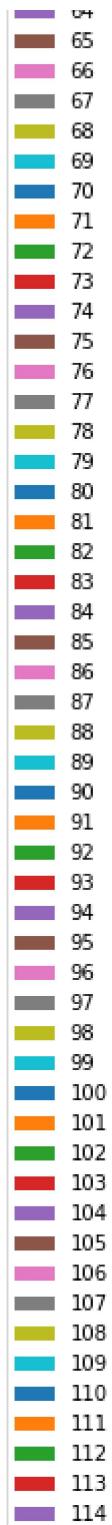


Pie chart

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

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

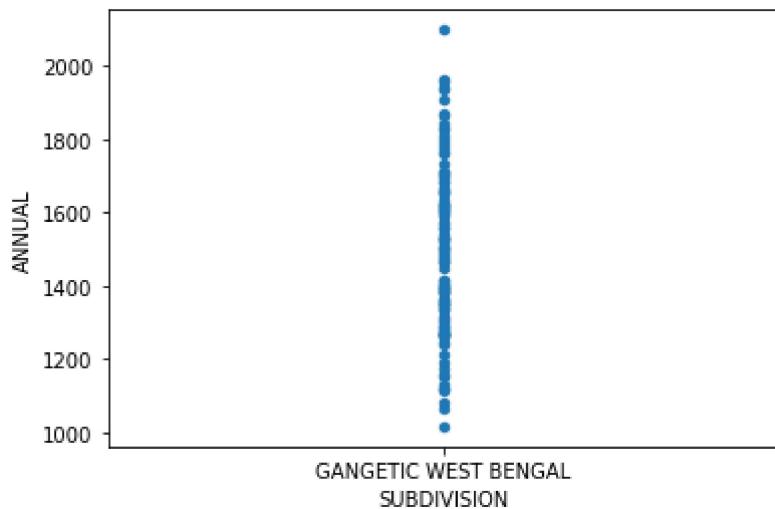




Scatter chart

```
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]:

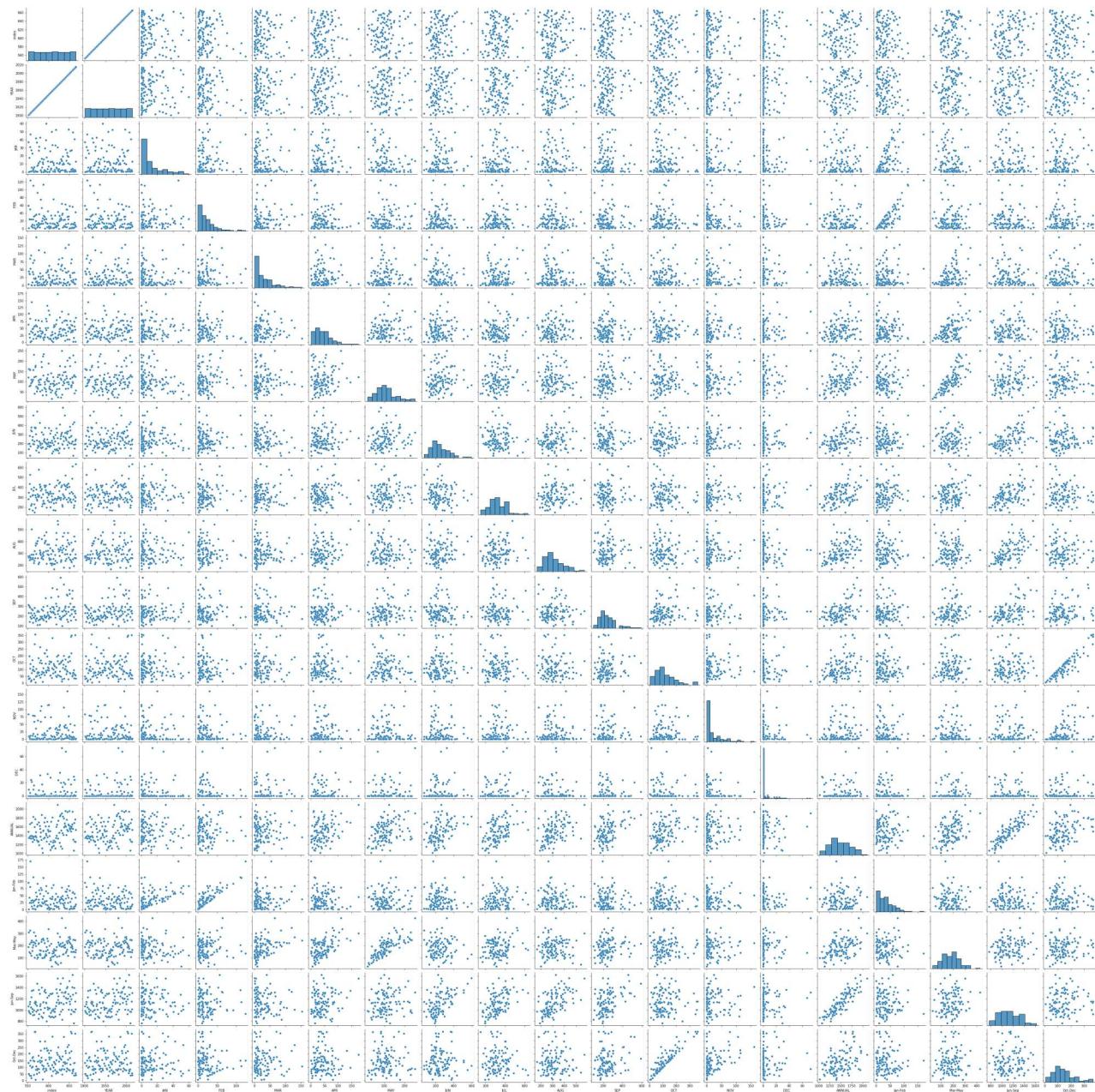
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	609.000000	1958.000000	12.595652	22.452174	29.090435	44.885217	107.787826	247.196522
std	33.341666	33.341666	14.741821	24.165919	30.754415	31.812974	51.001443	99.676225
min	552.000000	1901.000000	0.000000	0.000000	0.100000	0.900000	16.400000	69.700000
25%	580.500000	1929.500000	1.250000	5.200000	7.200000	21.000000	71.900000	180.950000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
50%	609.000000	1958.000000	6.800000	13.600000	18.900000	39.200000	98.900000	227.600000
75%	637.500000	1986.500000	18.350000	30.650000	42.250000	61.050000	131.400000	304.800000
max	666.000000	2015.000000	60.000000	123.600000	152.500000	174.200000	250.900000	597.100000

EDA AND VISUALIZATION

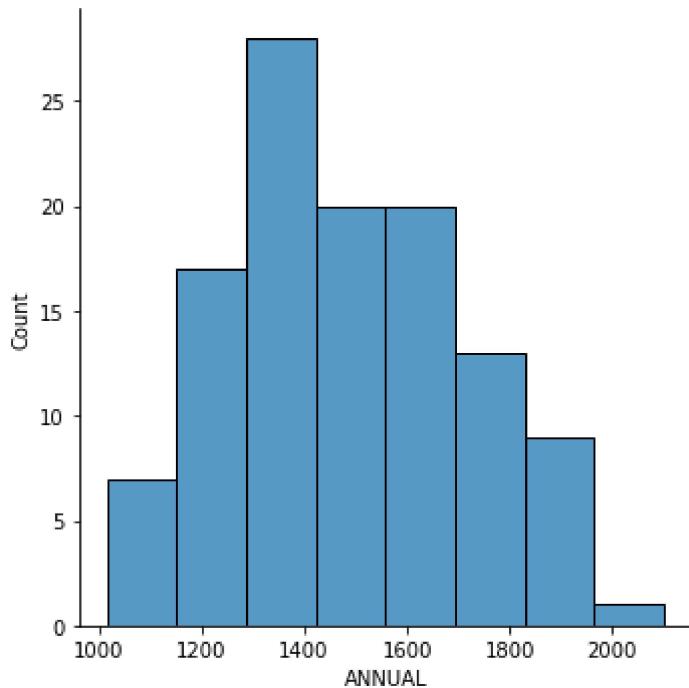
In [21]: `sns.pairplot(df)`

Out[21]: <seaborn.axisgrid.PairGrid at 0x1eff2220b80>



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

Out[22]: <seaborn.axisgrid.FacetGrid at 0x1effd7a82e0>

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

Out[23]: <AxesSubplot:>

