

# Importing Libraries

In [1]:

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

# Importing Datasets

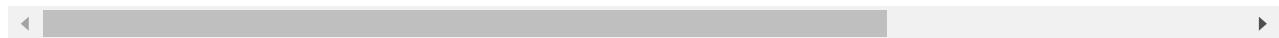
In [2]:

```
df=pd.read_csv("rainfall_jammu_kashmir.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	1702	JAMMU & KASHMIR	1901	66.4	69.3	69.6	132.2	105.8	53.4	171.7	181.3	101.8	24.1	0.
1	1703	JAMMU & KASHMIR	1902	6.5	9.7	91.3	100.5	70.7	113.3	108.4	136.9	62.2	15.1	44.1
2	1704	JAMMU & KASHMIR	1903	96.2	21.5	238.6	58.7	57.3	18.9	332.5	218.6	176.9	10.7	15.1
3	1705	JAMMU & KASHMIR	1904	110.6	17.3	145.2	64.5	67.8	25.9	182.3	132.2	62.3	50.0	24.1
4	1706	JAMMU & KASHMIR	1905	146.7	76.3	161.4	71.7	65.2	43.3	145.2	111.5	239.7	5.8	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	1812	JAMMU & KASHMIR	2011	43.4	211.6	97.8	89.0	32.4	72.5	81.6	131.2	72.0	19.4	12.1
111	1813	JAMMU & KASHMIR	2012	150.9	95.8	45.2	86.6	48.9	32.6	118.8	264.9	106.7	15.7	10.1
112	1814	JAMMU & KASHMIR	2013	52.2	136.4	41.9	47.4	47.4	80.5	125.1	219.1	41.2	34.4	13.1
113	1815	JAMMU & KASHMIR	2014	75.8	64.0	153.1	76.1	52.7	25.3	100.5	134.6	362.8	32.2	14.1
114	1816	JAMMU & KASHMIR	2015	27.9	187.2	341.4	173.3	64.6	121.4	233.2	129.2	130.2	87.1	38.1

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	DECEMBER
0	1702	JAMMU & KASHMIR	1901	66.4	69.3	69.6	132.2	105.8	53.4	171.7	181.3	101.8	24.1	0.	1.0
1	1703	JAMMU & KASHMIR	1902	6.5	9.7	91.3	100.5	70.7	113.3	108.4	136.9	62.2	15.1	44.1	1.0
2	1704	JAMMU & KASHMIR	1903	96.2	21.5	238.6	58.7	57.3	18.9	332.5	218.6	176.9	10.7	15.1	1.0
3	1705	JAMMU & KASHMIR	1904	110.6	17.3	145.2	64.5	67.8	25.9	182.3	132.2	62.3	50.0	24.1	1.0
4	1706	JAMMU & KASHMIR	1905	146.7	76.3	161.4	71.7	65.2	43.3	145.2	111.5	239.7	5.8	0.	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	1812	JAMMU & KASHMIR	2011	43.4	211.6	97.8	89.0	32.4	72.5	81.6	131.2	72.0	19.4	12.1	1.0
111	1813	JAMMU & KASHMIR	2012	150.9	95.8	45.2	86.6	48.9	32.6	118.8	264.9	106.7	15.7	10.1	1.0
112	1814	JAMMU & KASHMIR	2013	52.2	136.4	41.9	47.4	47.4	80.5	125.1	219.1	41.2	34.4	13.1	1.0
113	1815	JAMMU & KASHMIR	2014	75.8	64.0	153.1	76.1	52.7	25.3	100.5	134.6	362.8	32.2	14.1	1.0
114	1816	JAMMU & KASHMIR	2015	27.9	187.2	341.4	173.3	64.6	121.4	233.2	129.2	130.2	87.1	38.1	1.0

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	DECEMBER
0	1702	JAMMU & KASHMIR	1901	66.4	69.3	69.6	132.2	105.8	53.4	171.7	181.3	101.8	24.1	0.	1.0
1	1703	JAMMU & KASHMIR	1902	6.5	9.7	91.3	100.5	70.7	113.3	108.4	136.9	62.2	15.1	44.1	1.0
2	1704	JAMMU & KASHMIR	1903	96.2	21.5	238.6	58.7	57.3	18.9	332.5	218.6	176.9	10.7	15.1	1.0

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DECEMBER
3	1705	JAMMU & KASHMIR	1904	110.6	17.3	145.2	64.5	67.8	25.9	182.3	132.2	62.3	50.0	24.	1.0
4	1706	JAMMU & KASHMIR	1905	146.7	76.3	161.4	71.7	65.2	43.3	145.2	111.5	239.7	5.8	0.	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	1812	JAMMU & KASHMIR	2011	43.4	211.6	97.8	89.0	32.4	72.5	81.6	131.2	72.0	19.4	12.	1.0
111	1813	JAMMU & KASHMIR	2012	150.9	95.8	45.2	86.6	48.9	32.6	118.8	264.9	106.7	15.7	10.	1.0
112	1814	JAMMU & KASHMIR	2013	52.2	136.4	41.9	47.4	47.4	80.5	125.1	219.1	41.2	34.4	13.	1.0
113	1815	JAMMU & KASHMIR	2014	75.8	64.0	153.1	76.1	52.7	25.3	100.5	134.6	362.8	32.2	14.	1.0
114	1816	JAMMU & KASHMIR	2015	27.9	187.2	341.4	173.3	64.6	121.4	233.2	129.2	130.2	87.1	38.	1.0

115 rows × 20 columns

## Data Cleaning and Data Preprocessing

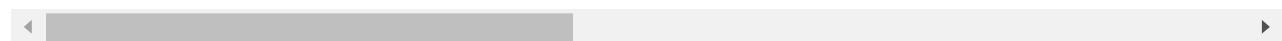
### describe()

In [5]:

```
df.describe()
```

Out[5]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	1759.000000	1958.000000	102.030435	115.450435	131.378261	93.702609	67.476522	64.234783
<b>std</b>	33.341666	33.341666	66.226797	66.933995	71.709964	48.381444	37.439217	33.147208
<b>min</b>	1702.000000	1901.000000	0.000000	9.700000	9.900000	5.700000	4.700000	4.000000
<b>25%</b>	1730.500000	1929.500000	59.300000	65.200000	83.800000	59.850000	40.150000	39.400000
<b>50%</b>	1759.000000	1958.000000	90.400000	102.700000	116.000000	86.800000	61.300000	59.900000
<b>75%</b>	1787.500000	1986.500000	129.700000	149.700000	173.500000	119.400000	91.750000	80.950000
<b>max</b>	1816.000000	2015.000000	367.800000	403.500000	341.400000	233.200000	234.400000	182.000000



### 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()`

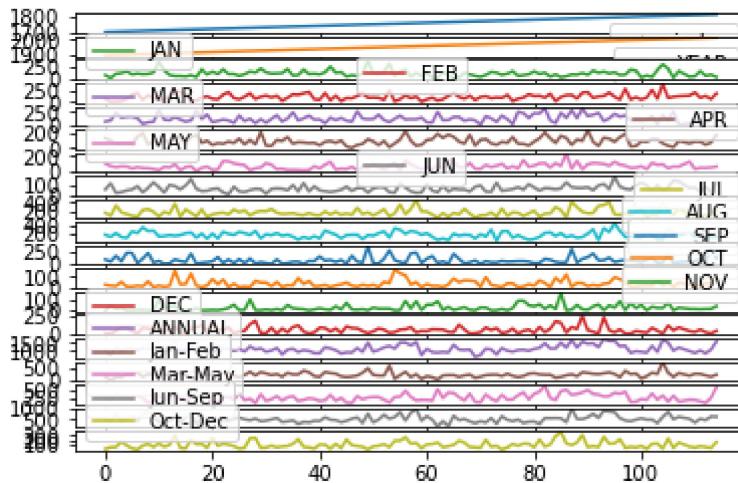
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 114 entries, 0 to 114
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   index       114 non-null    int64  
 1   SUBDIVISION 114 non-null    object  
 2   YEAR        114 non-null    int64  
 3   JAN         114 non-null    float64 
 4   FEB         114 non-null    float64 
 5   MAR         114 non-null    float64 
 6   APR         114 non-null    float64 
 7   MAY         114 non-null    float64 
 8   JUN         114 non-null    float64 
 9   JUL         114 non-null    float64 
 10  AUG         114 non-null    float64 
 11  SEP         114 non-null    float64 
 12  OCT         114 non-null    float64 
 13  NOV         114 non-null    float64 
 14  DEC         114 non-null    float64 
 15  ANNUAL      114 non-null    float64 
 16  Jan-Feb     114 non-null    float64
```

```
17 Mar-May      114 non-null    float64
18 Jun-Sep      114 non-null    float64
19 Oct-Dec      114 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 18.7+ KB
```

## Line chart

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

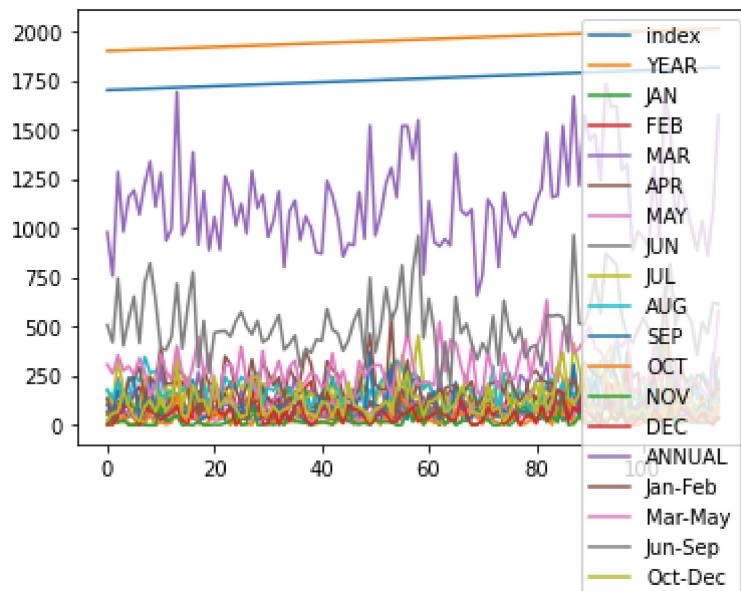
```
Out[11]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)
```



## Line chart

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

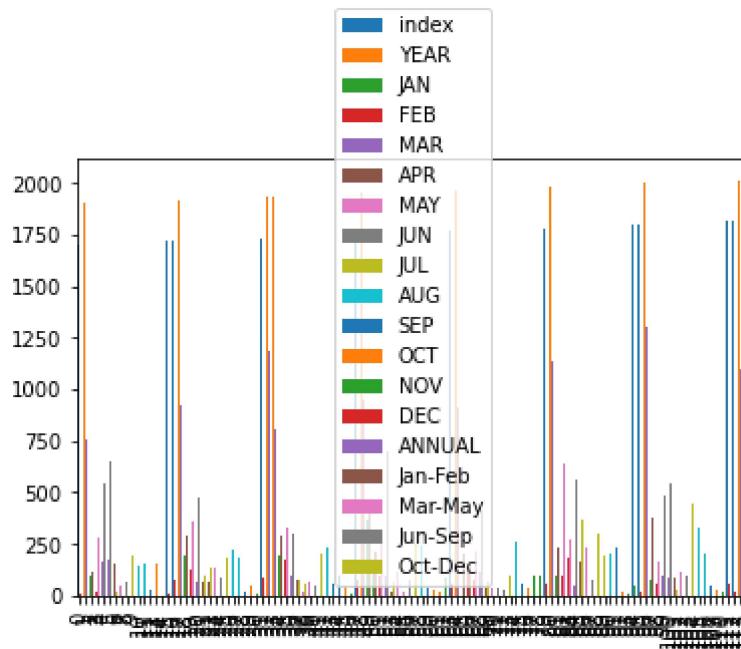
```
Out[12]: <AxesSubplot:>
```



## Bar chart

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

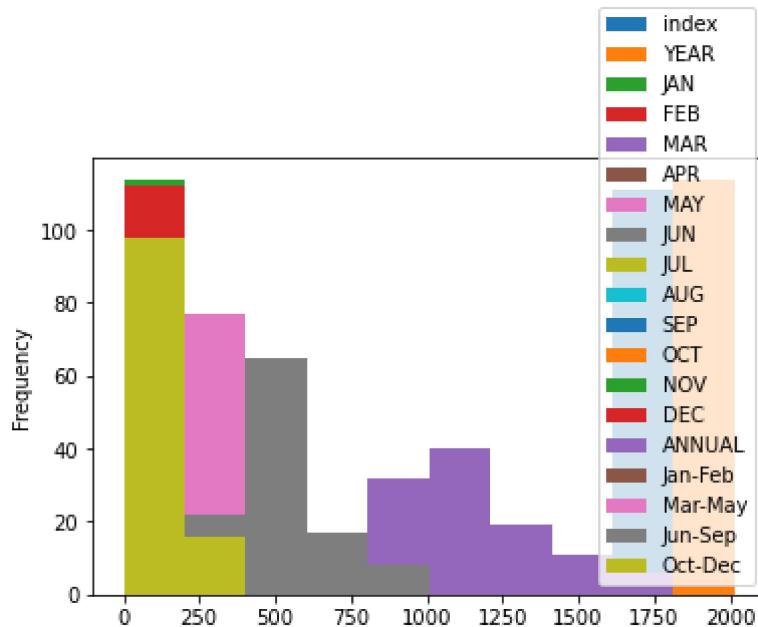
Out[13]: <AxesSubplot:>



## Histogram

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

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

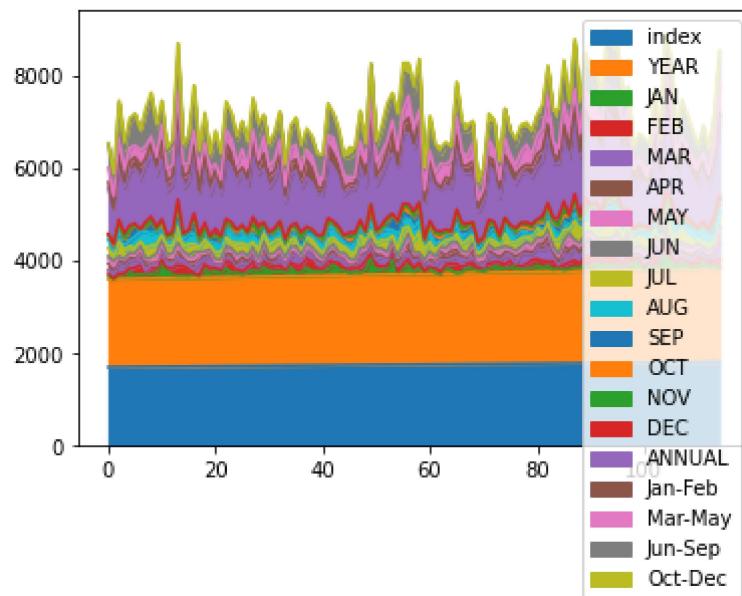


## Area chart

In [15]:

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

Out[15]: &lt;AxesSubplot:&gt;

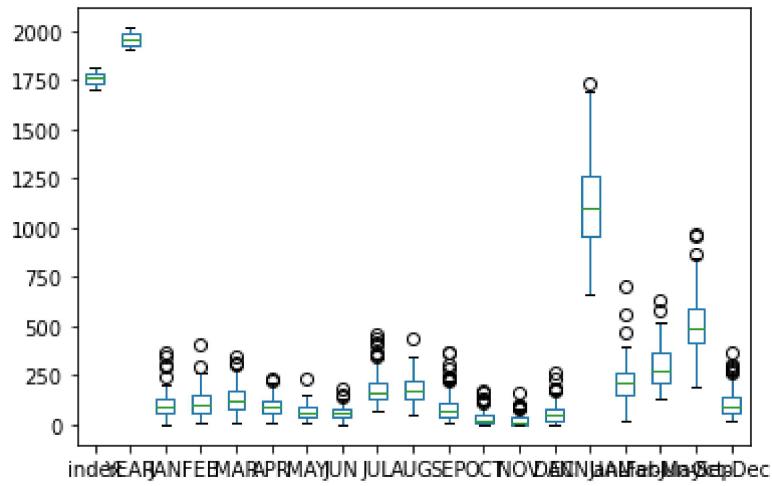


## Box chart

In [16]:

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

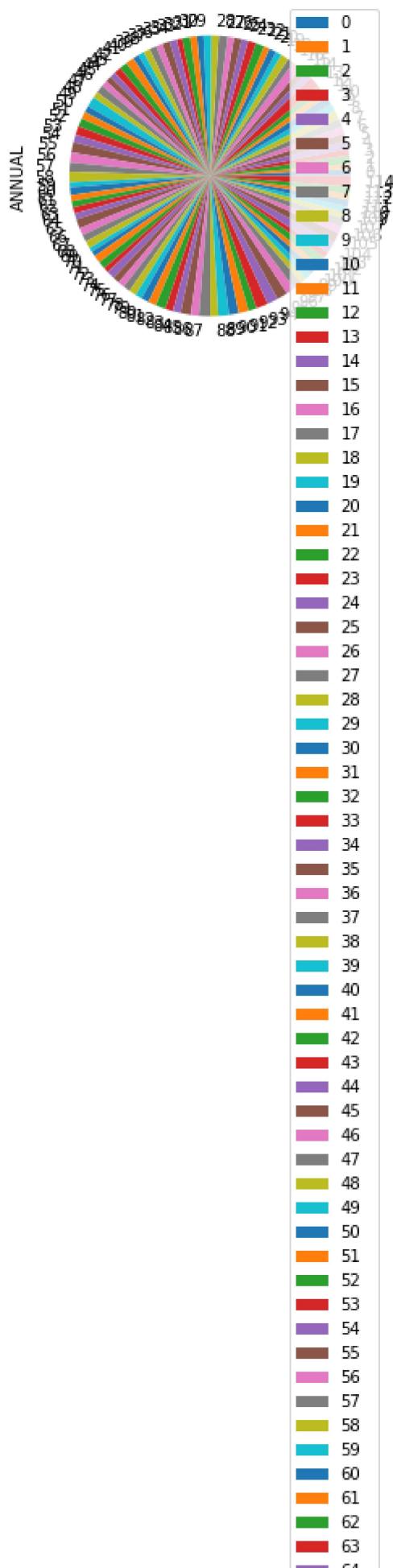
Out[16]: &lt;AxesSubplot:&gt;

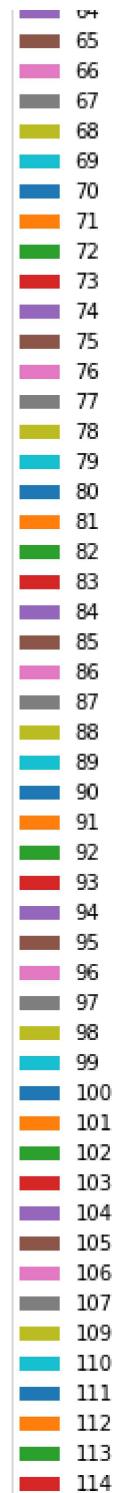


## Pie chart

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

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

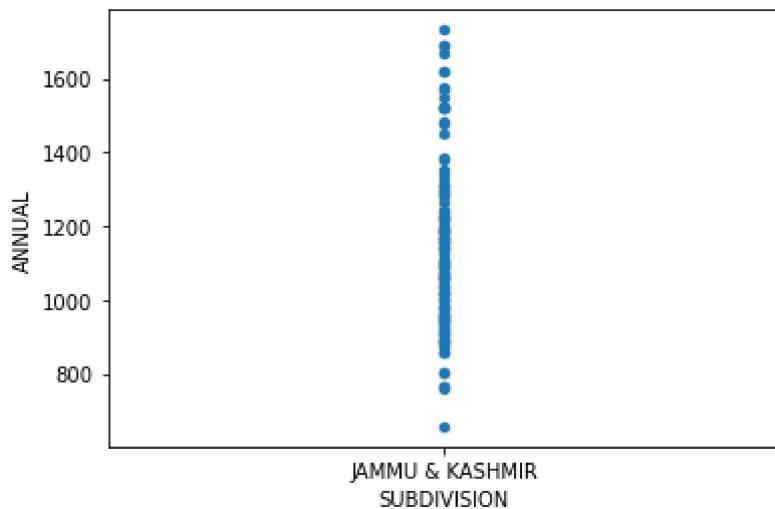




## Scatter chart

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: 114 entries, 0 to 114
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   index       114 non-null    int64  
 1   SUBDIVISION 114 non-null    object  
 2   YEAR        114 non-null    int64  
 3   JAN         114 non-null    float64 
 4   FEB         114 non-null    float64 
 5   MAR         114 non-null    float64 
 6   APR         114 non-null    float64 
 7   MAY         114 non-null    float64 
 8   JUN         114 non-null    float64 
 9   JUL         114 non-null    float64 
 10  AUG         114 non-null    float64 
 11  SEP         114 non-null    float64 
 12  OCT         114 non-null    float64 
 13  NOV         114 non-null    float64 
 14  DEC         114 non-null    float64 
 15  ANNUAL      114 non-null    float64 
 16  Jan-Feb     114 non-null    float64 
 17  Mar-May     114 non-null    float64 
 18  Jun-Sep     114 non-null    float64 
 19  Oct-Dec     114 non-null    float64 
dtypes: float64(17), int64(2), object(1)
memory usage: 18.7+ KB
```

In [20]:

`df.describe()`

Out[20]:

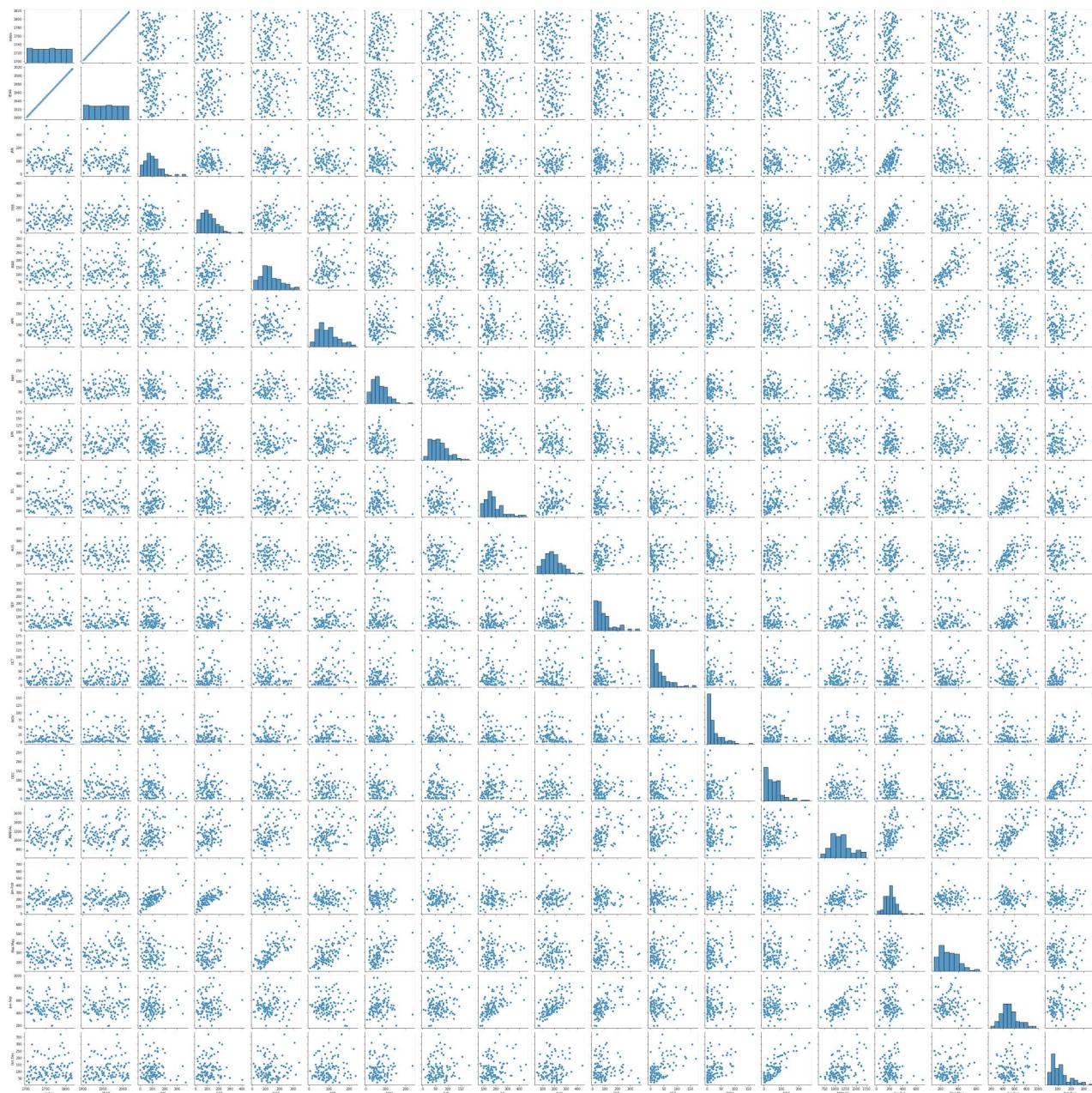
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000	114.000000
<b>mean</b>	1758.552632	1957.552632	101.870175	115.428947	131.971930	93.528947	67.504386	64.235965
<b>std</b>	33.140380	33.140380	66.496789	67.229113	71.742153	48.559036	37.603315	33.293551
<b>min</b>	1702.000000	1901.000000	0.000000	9.700000	9.900000	5.700000	4.700000	4.000000
<b>25%</b>	1730.250000	1929.250000	59.000000	64.600000	85.675000	59.475000	40.025000	39.300000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
50%	1758.500000	1957.500000	89.800000	101.950000	116.400000	86.700000	61.200000	59.750000
75%	1786.750000	1985.750000	130.700000	149.900000	174.800000	119.450000	91.825000	81.075000
max	1816.000000	2015.000000	367.800000	403.500000	341.400000	233.200000	234.400000	182.000000

## EDA AND VISUALIZATION

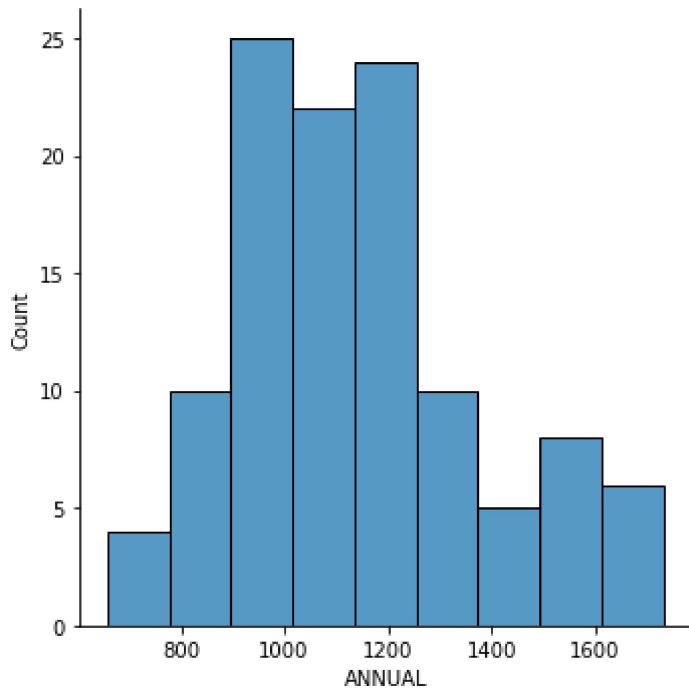
In [21]: `sns.pairplot(df)`

Out[21]: <seaborn.axisgrid.PairGrid at 0x2405da77910>



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

Out[22]: &lt;seaborn.axisgrid.FacetGrid at 0x24068f12070&gt;

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

Out[23]: &lt;AxesSubplot:&gt;

