

# 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_telangana.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	3197	TELANGANA	1901	6.9	41.8	7.8	45.2	22.0	123.6	237.8	177.2	77.7	75.5	12.2
1	3198	TELANGANA	1902	0.0	0.0	0.2	10.7	7.3	52.4	146.3	142.8	190.5	41.7	31.2
2	3199	TELANGANA	1903	12.9	4.6	0.0	9.9	40.7	99.2	505.2	246.7	191.9	155.8	15.5
3	3200	TELANGANA	1904	0.0	0.0	10.8	0.8	14.7	104.2	139.5	50.0	162.3	44.4	0.0
4	3201	TELANGANA	1905	0.0	4.3	12.8	27.6	32.2	129.5	82.4	237.3	179.1	19.6	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	3307	TELANGANA	2011	0.0	11.9	2.6	25.6	9.3	83.9	268.2	225.9	107.6	13.9	4.2
111	3308	TELANGANA	2012	6.7	0.0	0.2	14.0	8.4	124.4	300.3	229.9	202.4	83.6	38.7
112	3309	TELANGANA	2013	2.4	29.0	0.2	24.4	8.5	213.4	453.8	230.6	161.4	205.9	16.4
113	3310	TELANGANA	2014	0.2	2.9	58.3	10.3	73.3	62.3	146.0	205.2	146.8	29.6	10.8
114	3311	TELANGANA	2015	17.5	0.0	43.0	65.7	23.3	266.9	104.4	160.5	158.3	15.6	0.3

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	3197	TELANGANA	1901	6.9	41.8	7.8	45.2	22.0	123.6	237.8	177.2	77.7	75.5	12.2
1	3198	TELANGANA	1902	0.0	0.0	0.2	10.7	7.3	52.4	146.3	142.8	190.5	41.7	31.2

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
2	3199	TELANGANA	1903	12.9	4.6	0.0	9.9	40.7	99.2	505.2	246.7	191.9	155.8	15.5
3	3200	TELANGANA	1904	0.0	0.0	10.8	0.8	14.7	104.2	139.5	50.0	162.3	44.4	0.0
4	3201	TELANGANA	1905	0.0	4.3	12.8	27.6	32.2	129.5	82.4	237.3	179.1	19.6	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	3307	TELANGANA	2011	0.0	11.9	2.6	25.6	9.3	83.9	268.2	225.9	107.6	13.9	4.2
111	3308	TELANGANA	2012	6.7	0.0	0.2	14.0	8.4	124.4	300.3	229.9	202.4	83.6	38.7
112	3309	TELANGANA	2013	2.4	29.0	0.2	24.4	8.5	213.4	453.8	230.6	161.4	205.9	16.4
113	3310	TELANGANA	2014	0.2	2.9	58.3	10.3	73.3	62.3	146.0	205.2	146.8	29.6	10.8
114	3311	TELANGANA	2015	17.5	0.0	43.0	65.7	23.3	266.9	104.4	160.5	158.3	15.6	0.3

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	3197	TELANGANA	1901	6.9	41.8	7.8	45.2	22.0	123.6	237.8	177.2	77.7	75.5	12.2
1	3198	TELANGANA	1902	0.0	0.0	0.2	10.7	7.3	52.4	146.3	142.8	190.5	41.7	31.2
2	3199	TELANGANA	1903	12.9	4.6	0.0	9.9	40.7	99.2	505.2	246.7	191.9	155.8	15.5
3	3200	TELANGANA	1904	0.0	0.0	10.8	0.8	14.7	104.2	139.5	50.0	162.3	44.4	0.0
4	3201	TELANGANA	1905	0.0	4.3	12.8	27.6	32.2	129.5	82.4	237.3	179.1	19.6	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	3307	TELANGANA	2011	0.0	11.9	2.6	25.6	9.3	83.9	268.2	225.9	107.6	13.9	4.2
111	3308	TELANGANA	2012	6.7	0.0	0.2	14.0	8.4	124.4	300.3	229.9	202.4	83.6	38.7
112	3309	TELANGANA	2013	2.4	29.0	0.2	24.4	8.5	213.4	453.8	230.6	161.4	205.9	16.4
113	3310	TELANGANA	2014	0.2	2.9	58.3	10.3	73.3	62.3	146.0	205.2	146.8	29.6	10.8
114	3311	TELANGANA	2015	17.5	0.0	43.0	65.7	23.3	266.9	104.4	160.5	158.3	15.6	0.3

115 rows × 20 columns

## Data Cleaning and Data Preprocessing

## describe()

In [5]:

```
df.describe()
```

Out[5]:

	<b>index</b>	<b>YEAR</b>	<b>JAN</b>	<b>FEB</b>	<b>MAR</b>	<b>APR</b>	<b>MAY</b>	<b>JUN</b>
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	3254.000000	1958.000000	7.702609	9.688696	12.614783	18.185217	25.373913	142.126087
<b>std</b>	33.341666	33.341666	13.881222	15.223808	18.781819	15.931307	23.642777	57.725527
<b>min</b>	3197.000000	1901.000000	0.000000	0.000000	0.000000	0.100000	0.200000	28.500000
<b>25%</b>	3225.500000	1929.500000	0.000000	0.000000	1.400000	6.950000	8.450000	103.700000
<b>50%</b>	3254.000000	1958.000000	1.000000	3.400000	4.700000	14.000000	20.600000	133.800000
<b>75%</b>	3282.500000	1986.500000	9.700000	13.900000	15.250000	24.850000	34.500000	180.950000
<b>max</b>	3311.000000	2015.000000	98.700000	79.100000	108.600000	105.600000	159.800000	332.000000

## 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()
```

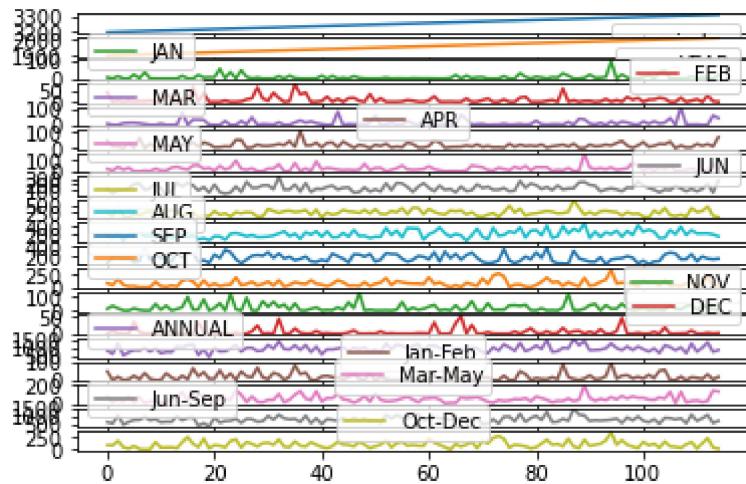
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 115 entries, 0 to 114
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   index       115 non-null    int64  
 1   SUBDIVISION 115 non-null    object  
 2   YEAR        115 non-null    int64  
 3   JAN         115 non-null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

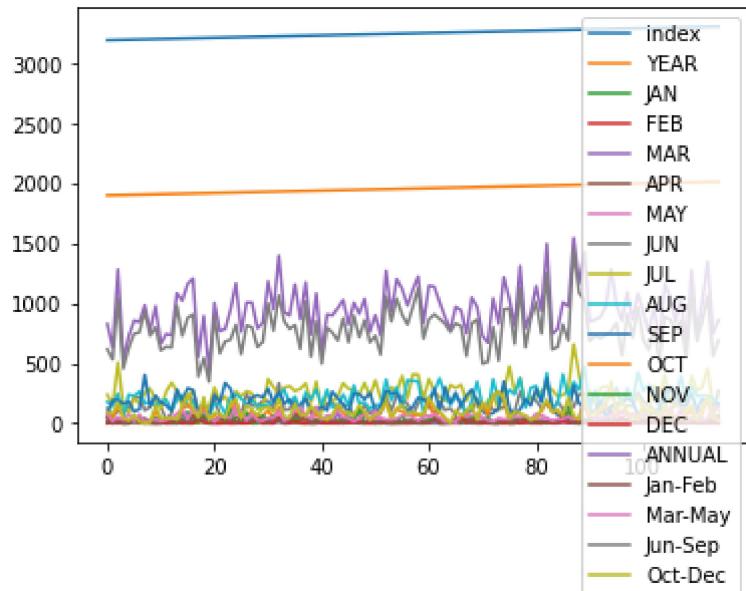
```
Out[11]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
   <AxesSubplot:>,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



## Line chart

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

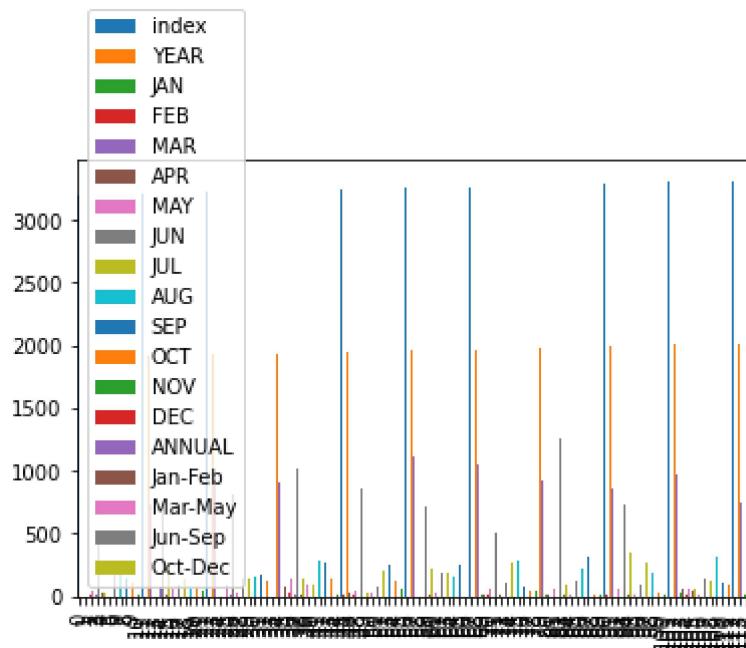
Out[12]: <AxesSubplot:>



## Bar chart

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

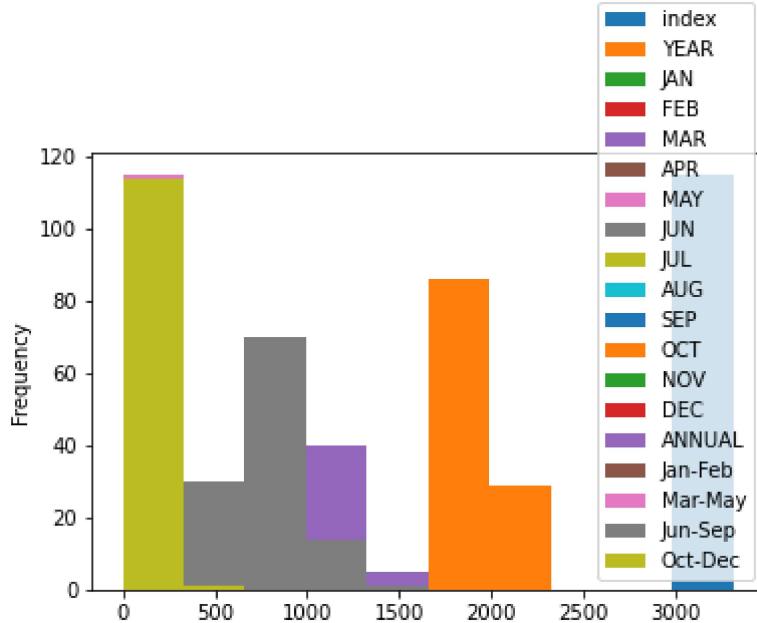
Out[13]: <AxesSubplot:>



## Histogram

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

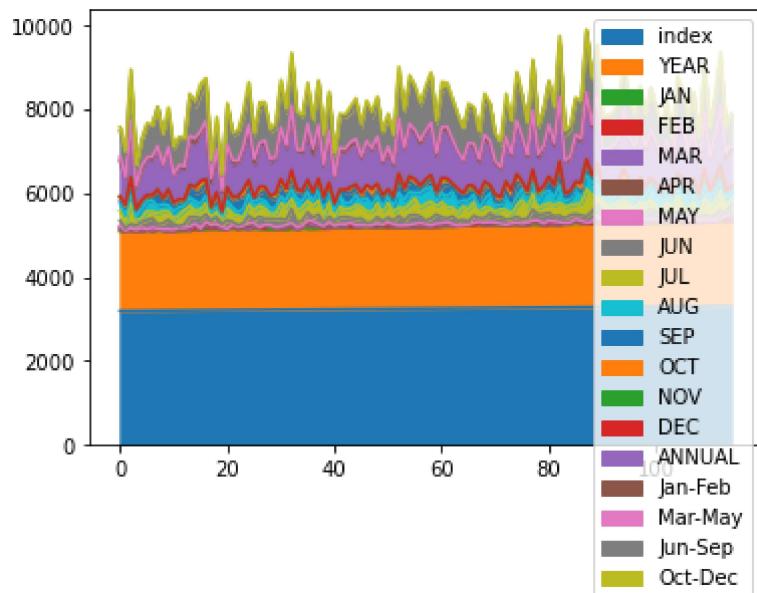
Out[14]: <AxesSubplot:ylabel='Frequency'>



## Area chart

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

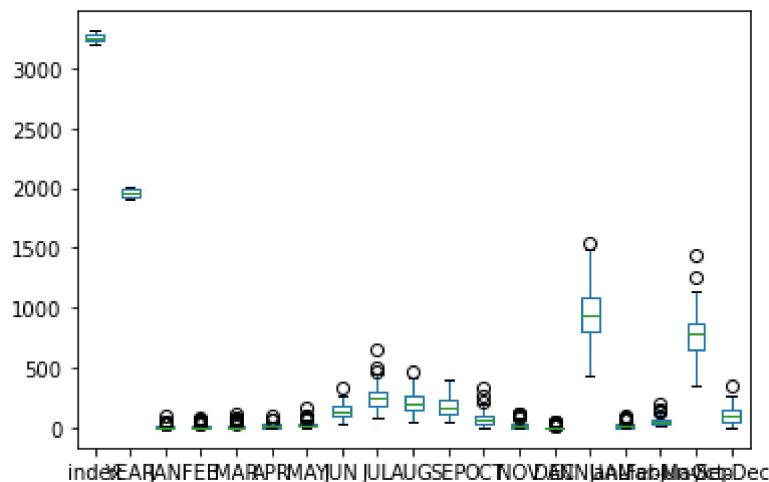
Out[15]: <AxesSubplot:>



## Box chart

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

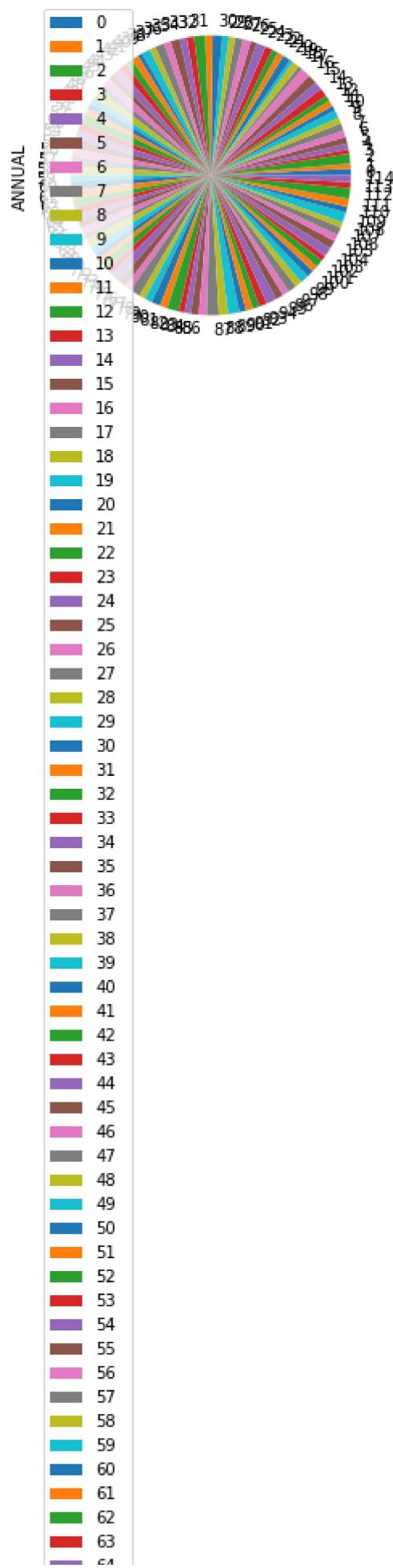
```
Out[16]: <AxesSubplot:>
```

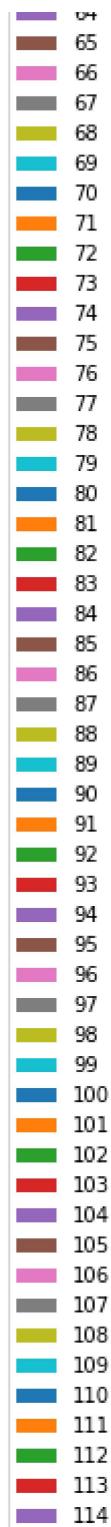


## Pie chart

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

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

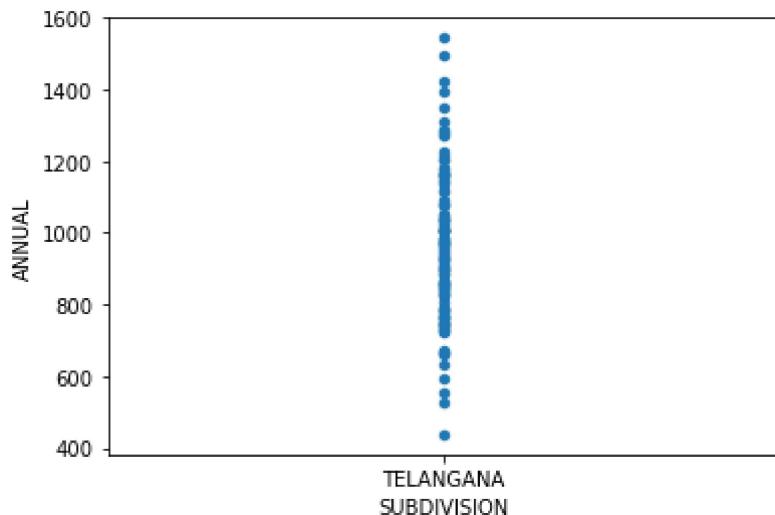




## Scatter chart

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]:

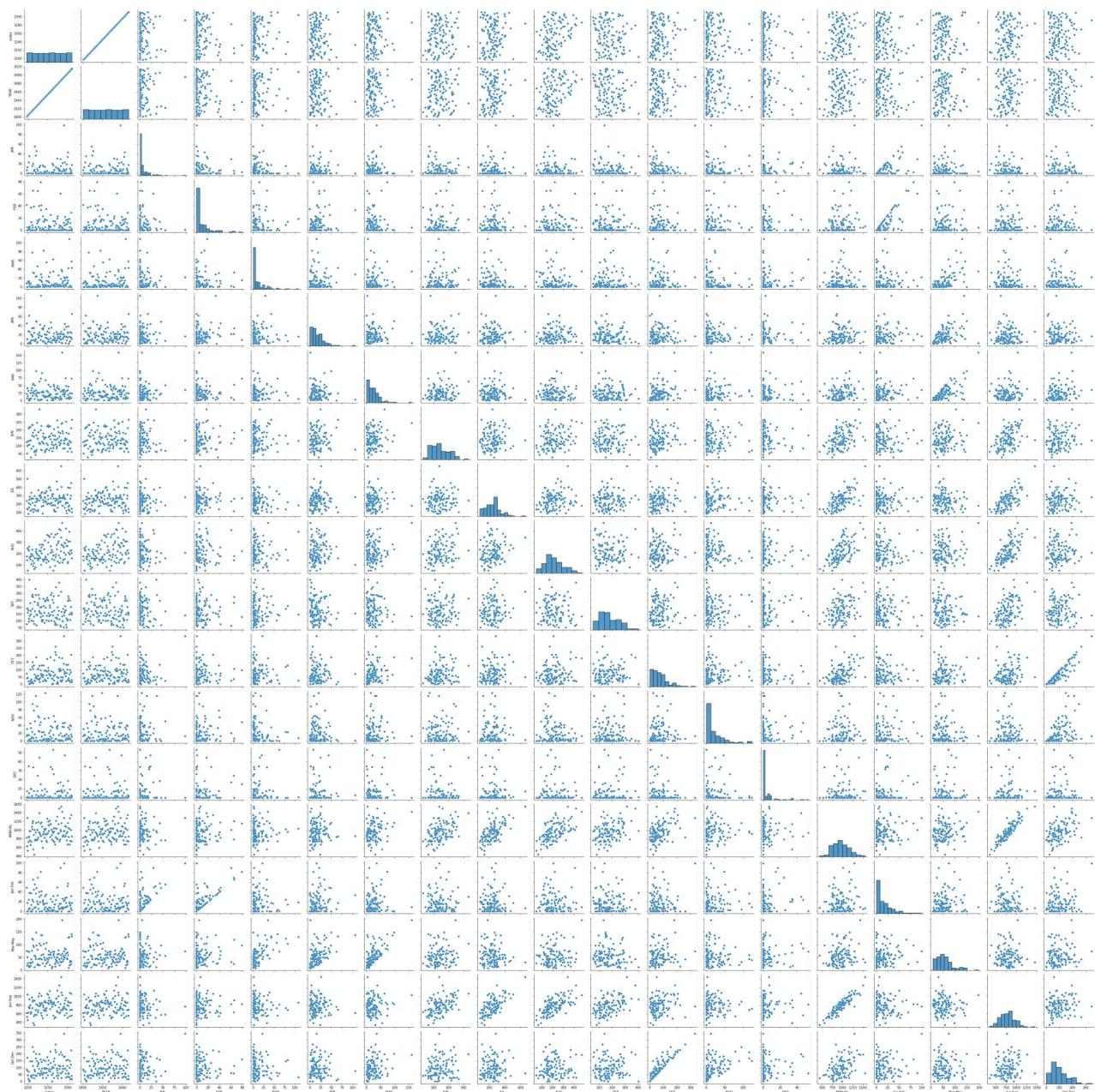
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	3254.000000	1958.000000	7.702609	9.688696	12.614783	18.185217	25.373913	142.126087
<b>std</b>	33.341666	33.341666	13.881222	15.223808	18.781819	15.931307	23.642777	57.725527
<b>min</b>	3197.000000	1901.000000	0.000000	0.000000	0.000000	0.100000	0.200000	28.500000
<b>25%</b>	3225.500000	1929.500000	0.000000	0.000000	1.400000	6.950000	8.450000	103.700000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>50%</b>	3254.000000	1958.000000	1.000000	3.400000	4.700000	14.000000	20.600000	133.800000
<b>75%</b>	3282.500000	1986.500000	9.700000	13.900000	15.250000	24.850000	34.500000	180.950000
<b>max</b>	3311.000000	2015.000000	98.700000	79.100000	108.600000	105.600000	159.800000	332.000000

## EDA AND VISUALIZATION

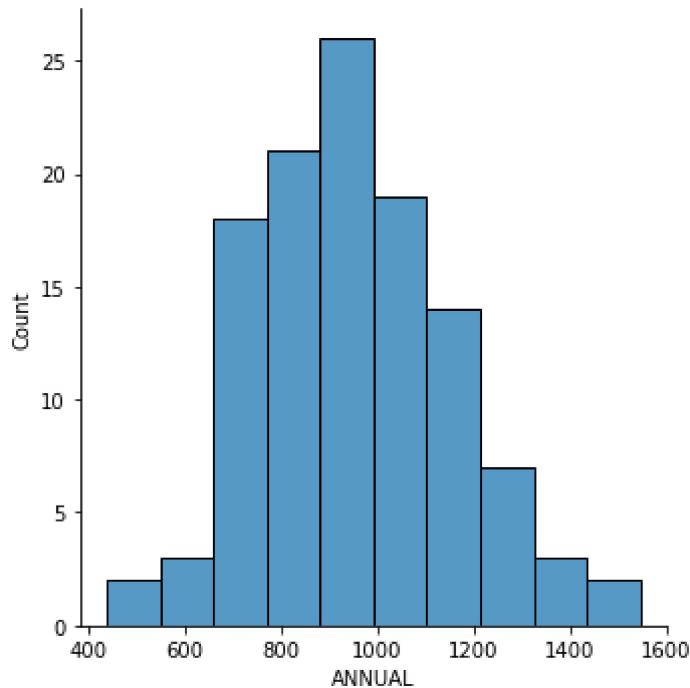
In [21]: `sns.pairplot(df)`

Out[21]: <seaborn.axisgrid.PairGrid at 0x2145fc328b0>



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

Out[22]: &lt;seaborn.axisgrid.FacetGrid at 0x2146b1cbf10&gt;

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

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

