

# 20104028

## Heamnath N

### 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_east madhya pradesh.csv")
df
```

Out[2]:

		index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	2162		EAST MADHYA PRADESH	1901	48.5	38.1	15.7	10.7	6.2	61.0	367.5	589.2	189.9	5.9	0.0
1	2163		EAST MADHYA PRADESH	1902	14.9	8.9	0.0	3.6	2.7	28.0	411.9	227.0	236.6	17.0	27.6
2	2164		EAST MADHYA PRADESH	1903	5.6	2.9	0.3	0.9	37.5	67.5	261.4	366.7	257.4	177.9	0.0
3	2165		EAST MADHYA PRADESH	1904	2.0	15.3	48.2	0.0	8.6	109.9	443.2	316.6	135.6	44.8	3.2
4	2166		EAST MADHYA PRADESH	1905	15.9	8.0	14.3	12.3	10.2	34.4	292.4	243.3	250.9	2.9	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	2272		EAST MADHYA PRADESH	2011	0.6	1.9	0.3	7.1	4.7	332.5	323.6	326.9	276.5	1.1	0.0
111	2273		EAST MADHYA PRADESH	2012	39.4	0.7	0.6	1.1	1.2	67.8	398.9	351.7	172.6	12.7	3.8
112	2274		EAST MADHYA PRADESH	2013	2.0	43.4	14.1	9.5	0.3	311.9	456.2	480.8	78.0	124.2	0.5

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
113	2275	EAST MADHYA PRADESH	2014	32.1	49.7	17.8	5.1	2.5	91.8	283.4	231.8	139.6	56.4	1.9
114	2276	EAST MADHYA PRADESH	2015	37.3	11.0	73.4	25.8	6.3	139.2	262.2	272.1	71.6	38.2	1.2

115 rows × 20 columns

## Data Cleaning and Data Preprocessing

In [3]: `df=df.dropna()`

In [4]: `df.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')`

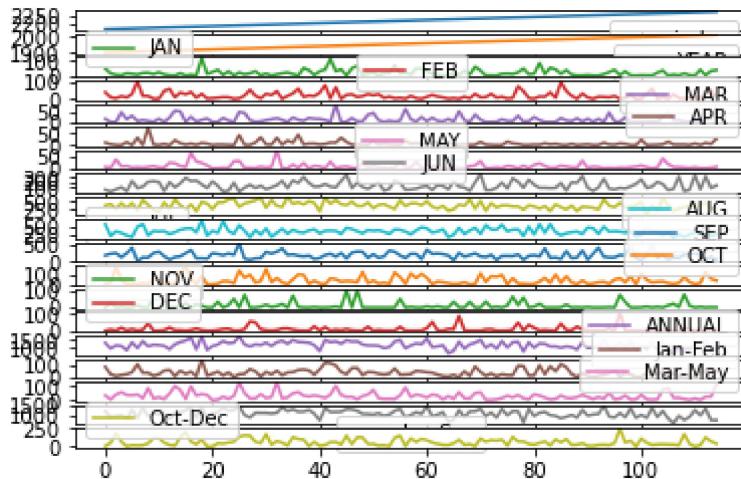
In [5]: `df.info()`

```
<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 [6]: `df.plot.line(subplots=True)`

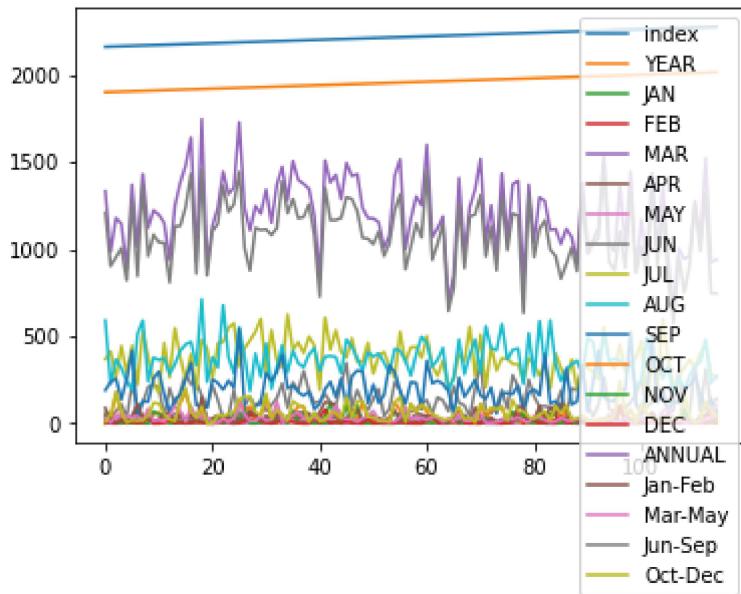
Out[6]: `array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)`



## Line chart

In [7]: `df.plot.line()`

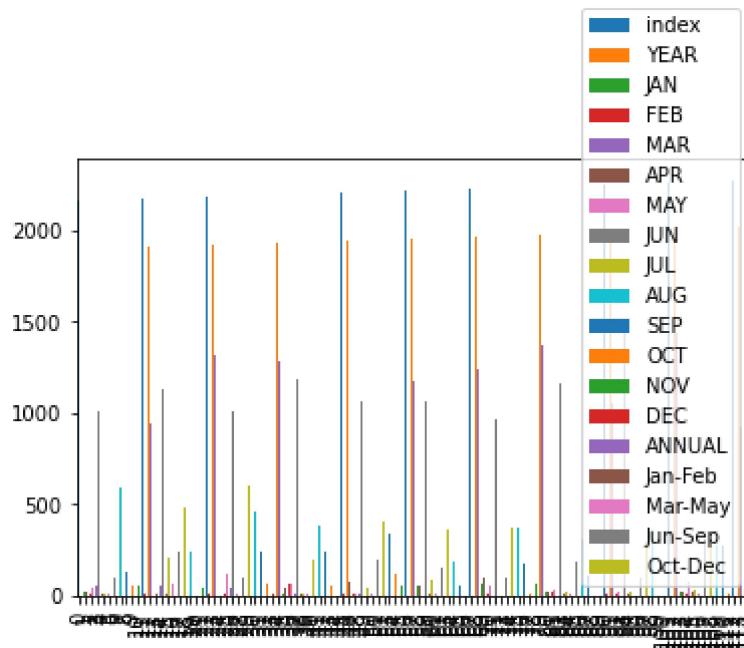
Out[7]: `<AxesSubplot:>`



## Bar chart

In [8]: `df.plot.bar()`

Out[8]: &lt;AxesSubplot:&gt;

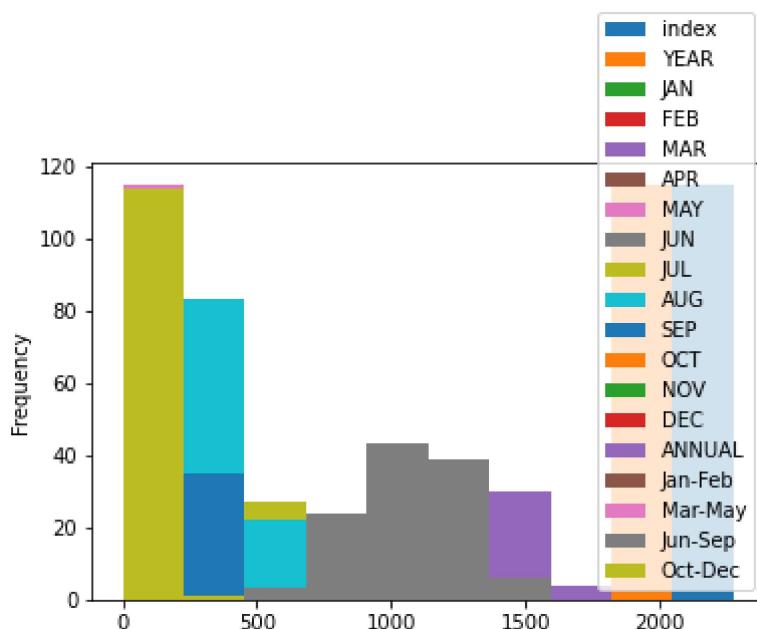


## Histogram

In [9]:

df.plot.hist()

Out[9]: &lt;AxesSubplot:ylabel='Frequency'&gt;

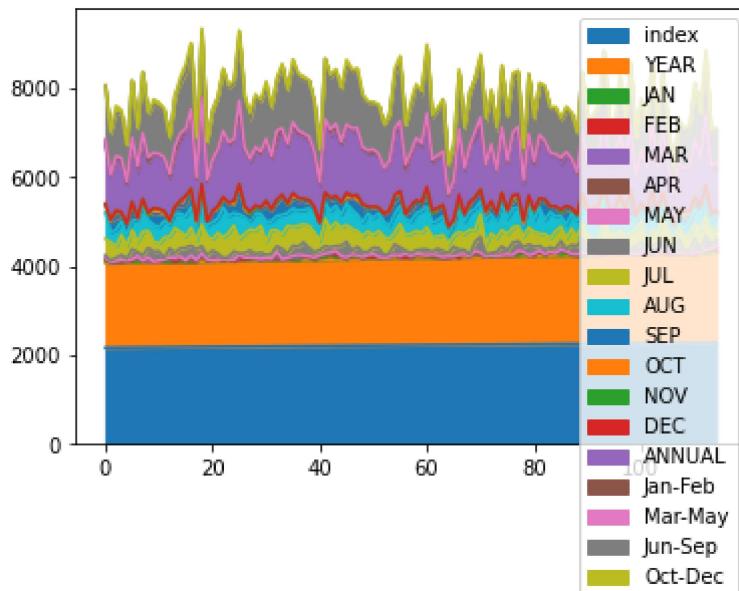


## Area chart

In [10]:

df.plot.area()

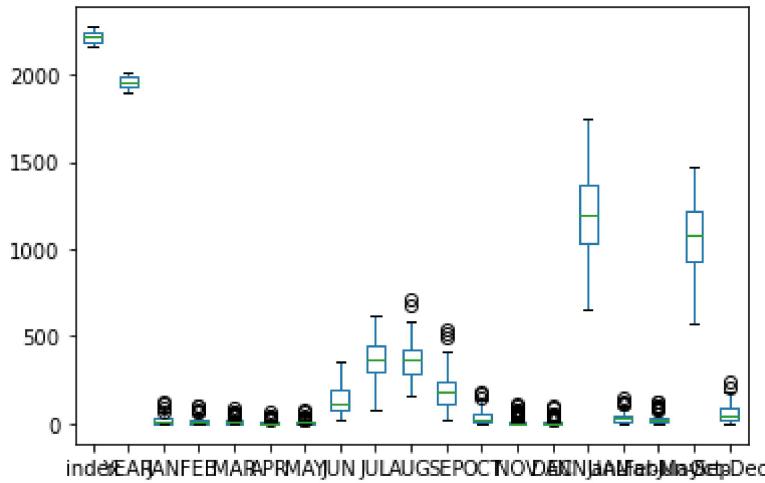
Out[10]: &lt;AxesSubplot:&gt;



## Box chart

In [11]: `df.plot.box()`

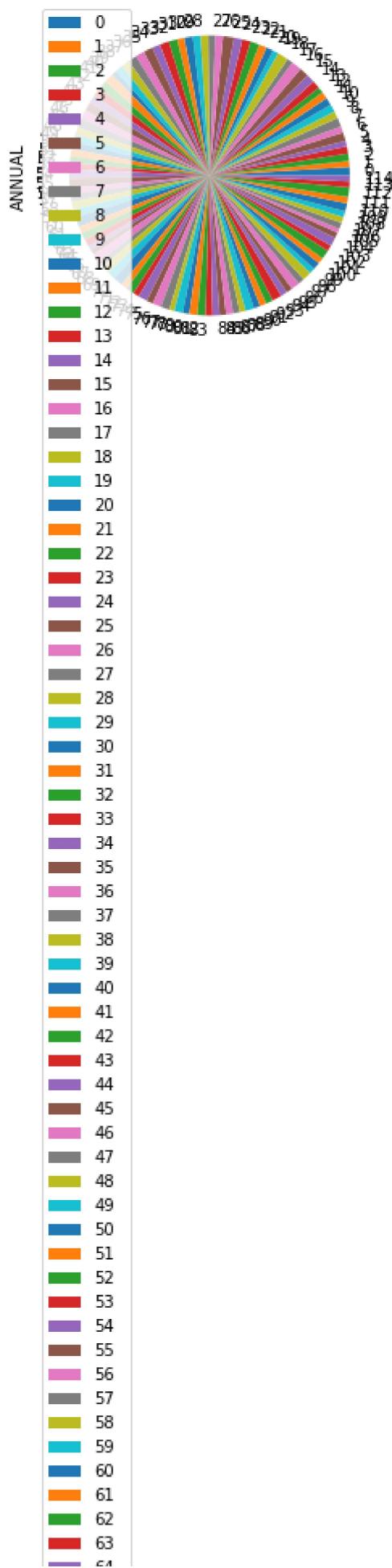
Out[11]: &lt;AxesSubplot:&gt;

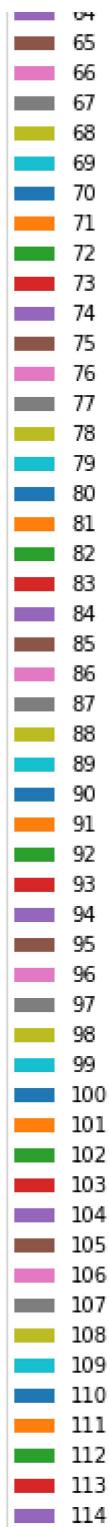


## Pie chart

In [12]: `df.plot.pie(y='ANNUAL')`

Out[12]: &lt;AxesSubplot:ylabel='ANNUAL'&gt;





## Scatter chart

In [13]: `df.plot.scatter(x='SUBDIVISION' ,y='ANNUAL')`

Out[13]: <AxesSubplot:xlabel='SUBDIVISION', ylabel='ANNUAL'>



In [14]:

`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 [15]:

`df.describe()`

Out[15]:

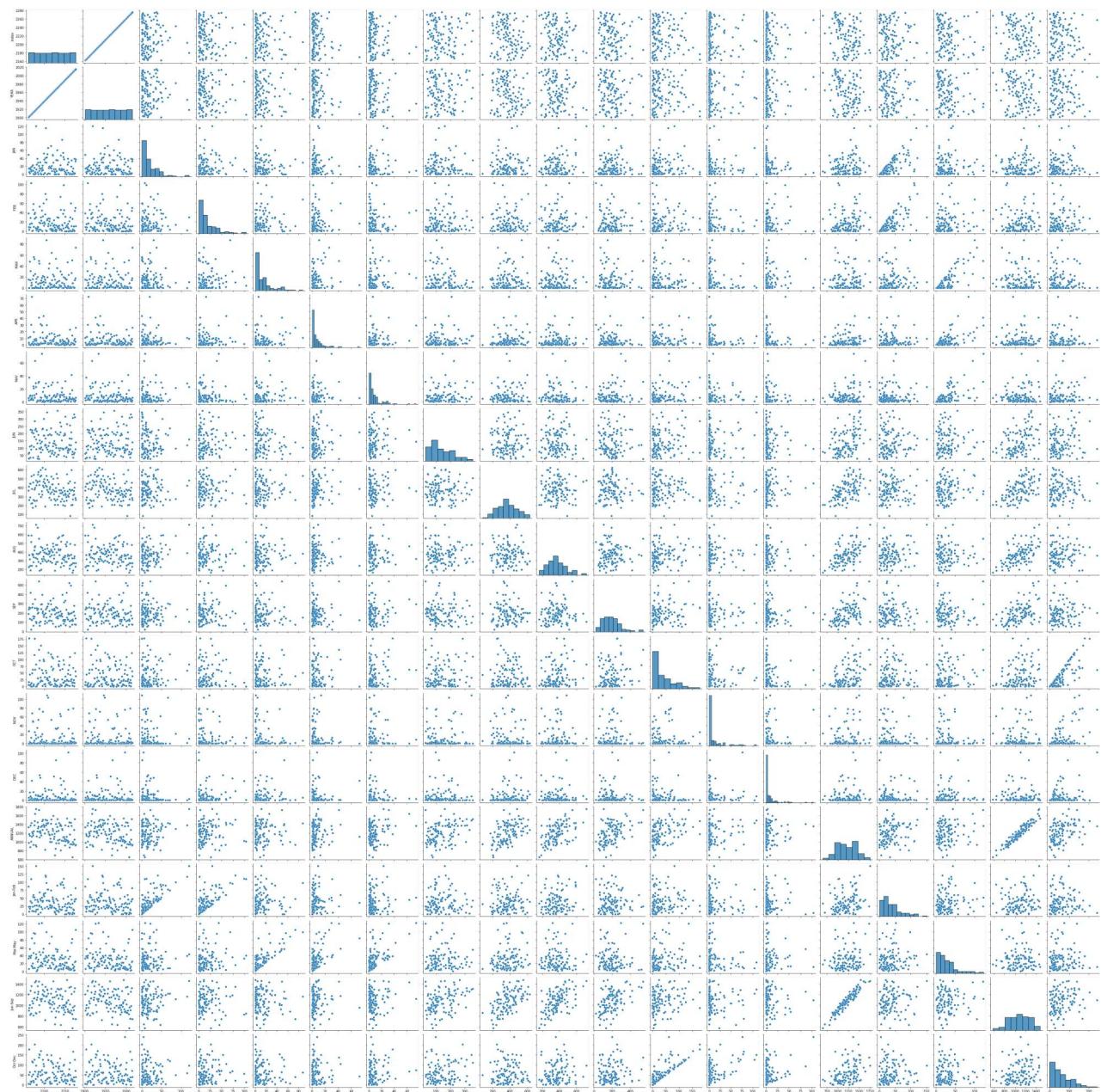
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	2219.000000	1958.000000	19.401739	18.693913	13.637391	7.188696	9.273043	141.029565
<b>std</b>	33.341666	33.341666	22.318347	20.795522	17.354996	10.473272	12.145379	79.359922
<b>min</b>	2162.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	26.300000
<b>25%</b>	2190.500000	1929.500000	2.200000	3.650000	1.150000	1.350000	2.100000	82.850000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>50%</b>	2219.000000	1958.000000	12.800000	11.300000	8.000000	3.200000	5.100000	118.500000
<b>75%</b>	2247.500000	1986.500000	29.650000	27.400000	18.650000	8.750000	10.500000	197.300000
<b>max</b>	2276.000000	2015.000000	120.700000	103.100000	87.300000	72.400000	74.200000	356.600000

## EDA AND VISUALIZATION

In [16]: `sns.pairplot(df)`

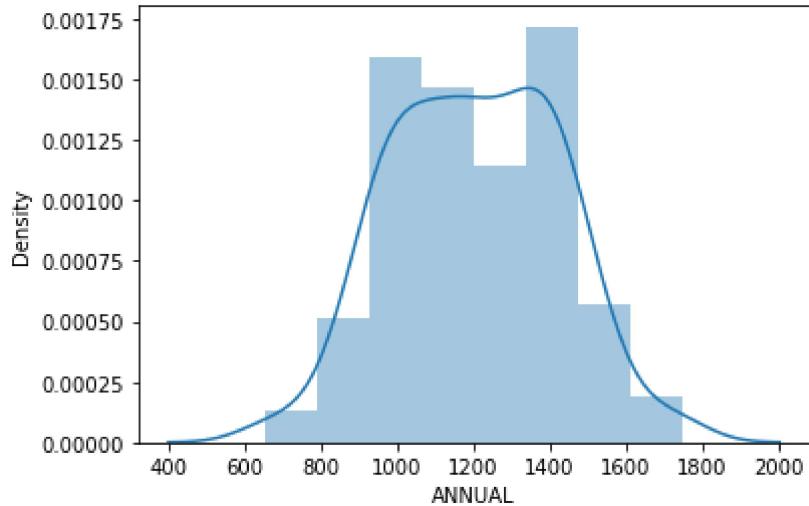
Out[16]: <seaborn.axisgrid.PairGrid at 0x230e76b1df0>



In [17]: `sns.distplot(df['ANNUAL'])`

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:  
`distplot` is a deprecated function and will be removed in a future version. Please adapt  
your code to use either `displot` (a figure-level function with similar flexibility) o  
r `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)
```

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



```
In [18]: sns.heatmap(df.corr())
```

```
Out[18]: <AxesSubplot:>
```

