

# 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_north interior karnataka.csv")
df
```

Out[2]:

		index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
0	3657		NORTH INTERIOR KARNATAKA	1901	3.5	18.8	7.1	67.2	65.5	120.5	151.9	115.1	128.8	80.0	13.6
1	3658		NORTH INTERIOR KARNATAKA	1902	0.0	0.0	0.3	22.5	34.4	111.3	83.2	78.1	146.7	118.8	35.7
2	3659		NORTH INTERIOR KARNATAKA	1903	3.5	0.0	0.1	6.9	53.4	102.8	209.4	146.4	189.3	166.4	34.3
3	3660		NORTH INTERIOR KARNATAKA	1904	0.2	0.3	8.5	11.0	46.3	120.6	91.6	48.5	165.1	86.5	0.0
4	3661		NORTH INTERIOR KARNATAKA	1905	0.0	6.0	2.6	16.0	51.2	99.6	60.1	139.2	42.2	85.0	4.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	3767		NORTH INTERIOR KARNATAKA	2011	0.5	7.2	7.2	41.2	46.8	101.3	150.8	152.0	69.0	73.4	5.7
111	3768		NORTH INTERIOR KARNATAKA	2012	28.5	6.2	0.4	35.4	19.5	60.0	114.5	105.5	79.2	85.2	46.5
112	3769		NORTH INTERIOR KARNATAKA	2013	1.2	6.1	3.0	25.4	47.4	99.4	160.7	73.9	201.0	101.0	4.2

			index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
113	3770			NORTH INTERIOR KARNATAKA	2014	0.0	6.1	29.2	26.4	93.0	50.4	136.8	205.2	90.2	80.3	25.0
114	3771			NORTH INTERIOR KARNATAKA	2015	2.4	0.0	27.5	50.8	45.3	89.6	38.5	78.4	150.8	61.2	5.7

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')`

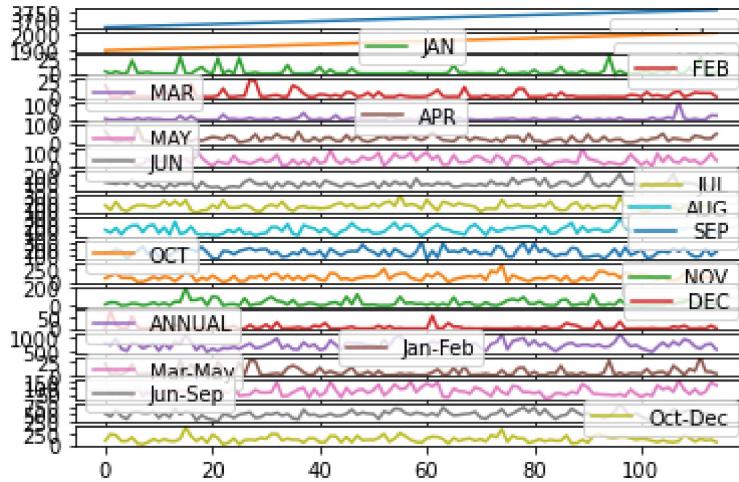
In [5]: `df.info()`

```
<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)`

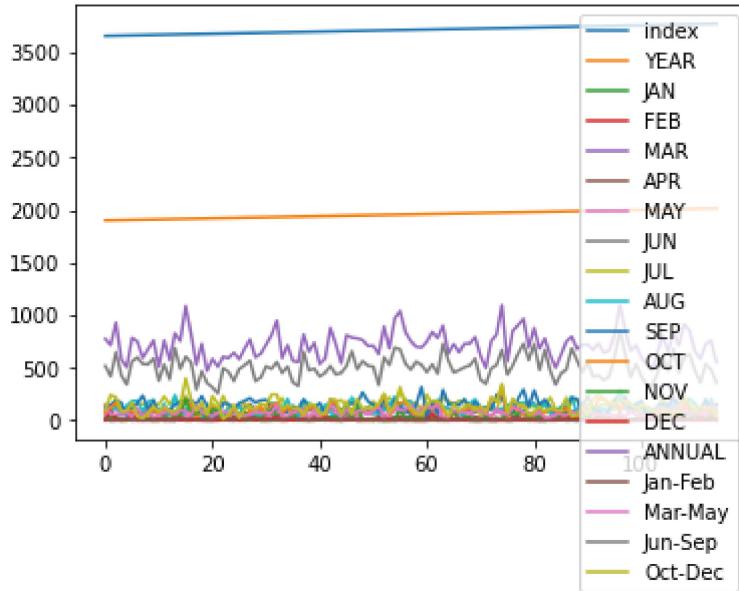
Out[6]: `array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)`



## Line chart

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

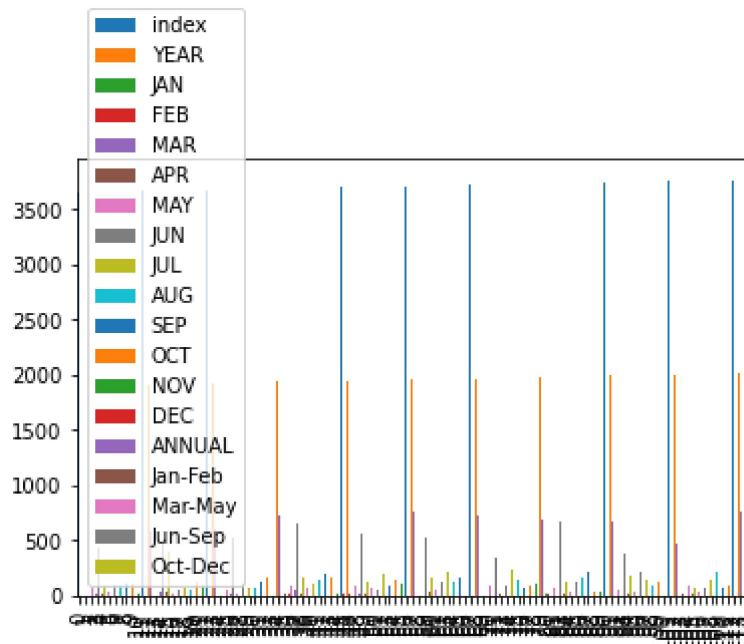
Out[7]: `<AxesSubplot:>`



## Bar chart

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

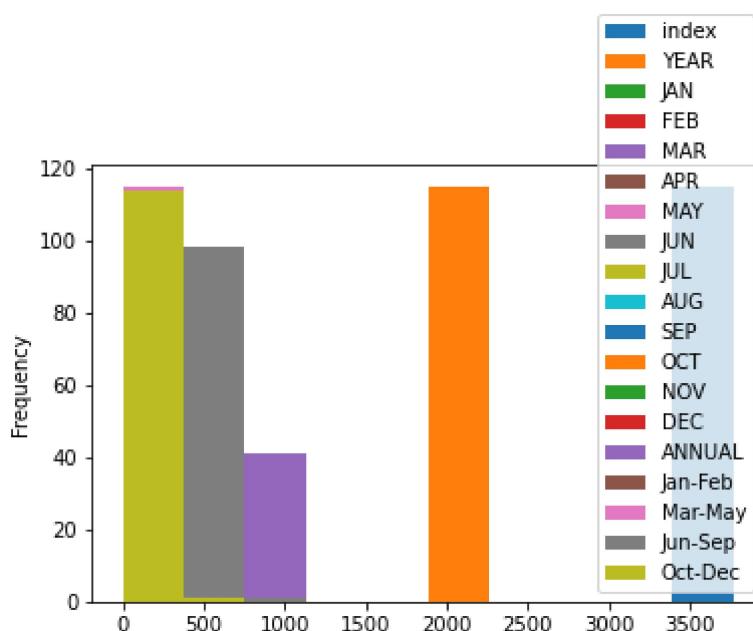
Out[8]: &lt;AxesSubplot:&gt;



## Histogram

In [9]: `df.plot.hist()`

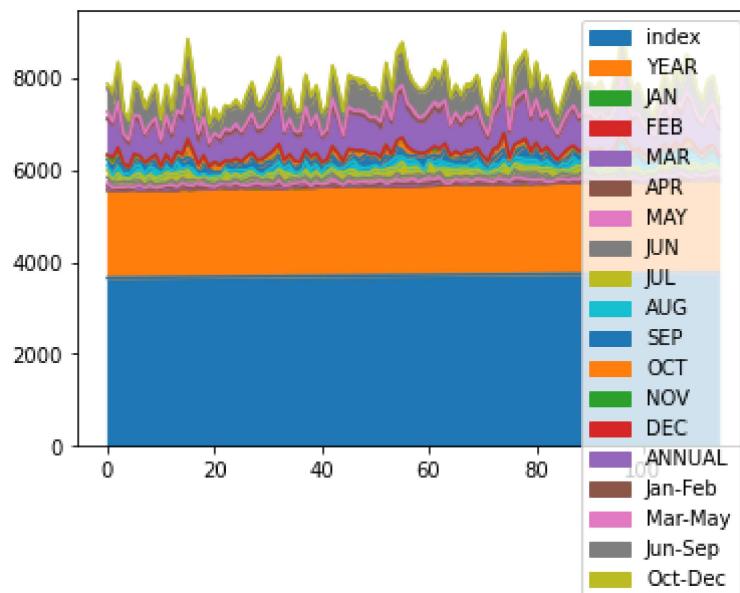
Out[9]: &lt;AxesSubplot:ylabel='Frequency'&gt;



## Area chart

In [10]: `df.plot.area()`

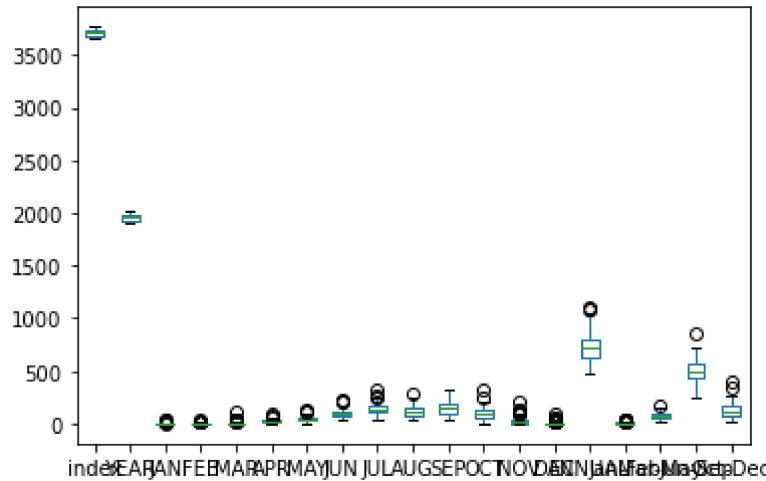
Out[10]: &lt;AxesSubplot:&gt;



## Box chart

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

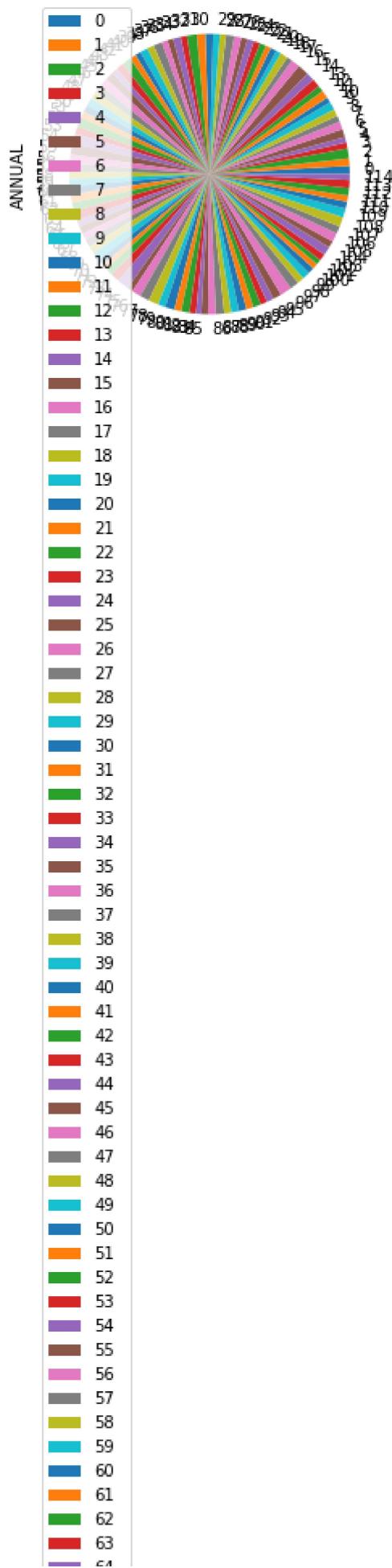
Out[11]: &lt;AxesSubplot:&gt;

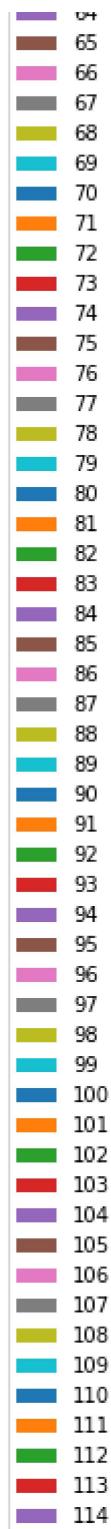


## Pie chart

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

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

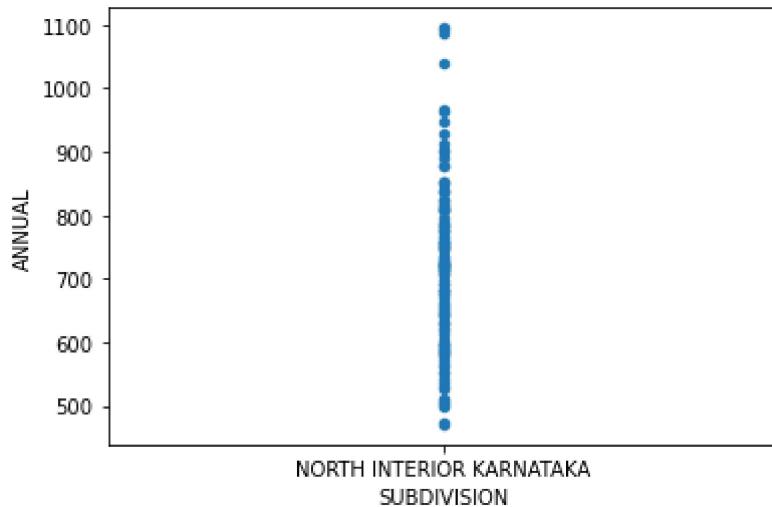




## Scatter chart

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]:

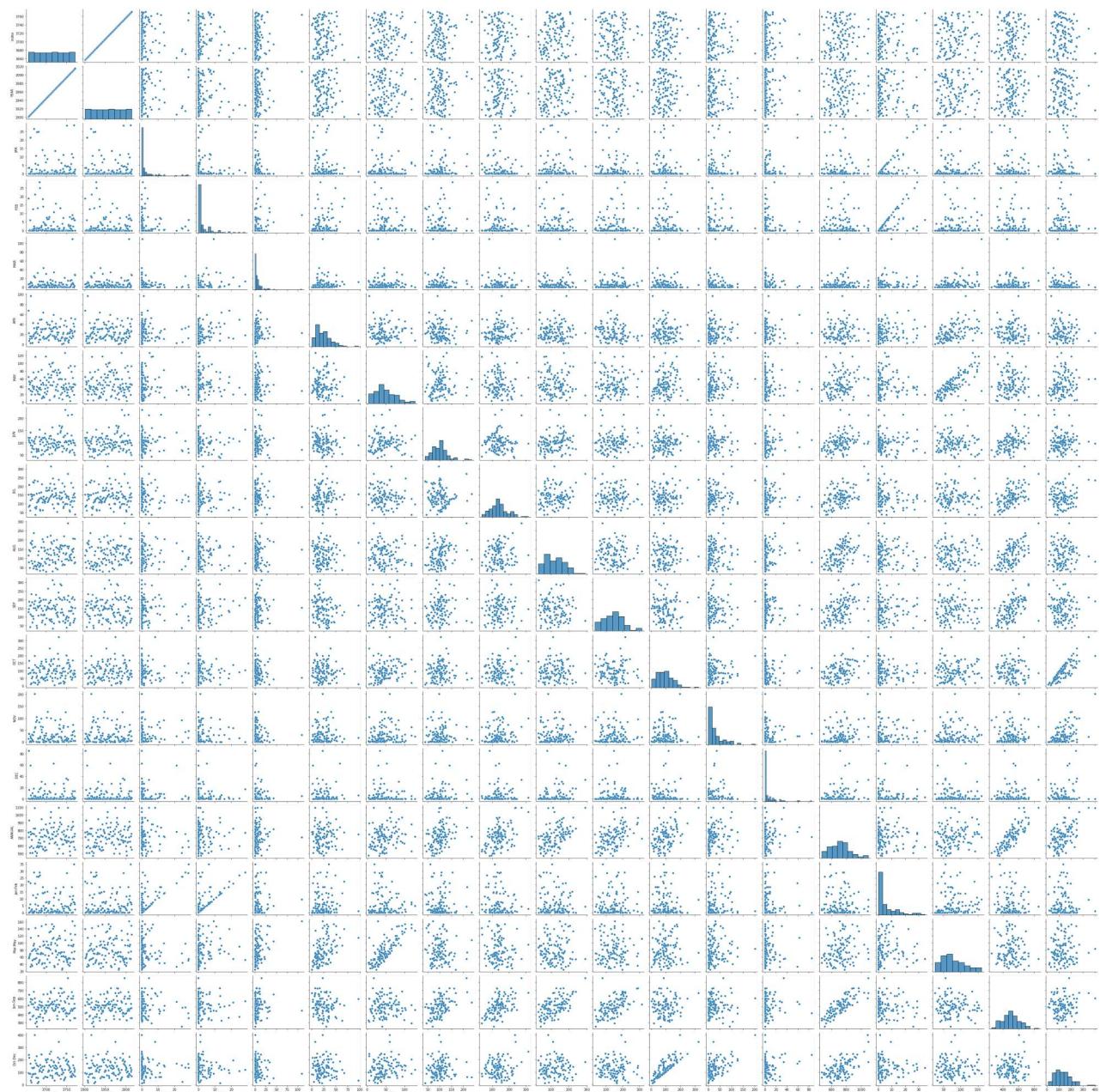
	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>count</b>	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
<b>mean</b>	3714.000000	1958.000000	3.013043	3.172174	7.123478	24.300870	47.035652	100.993043
<b>std</b>	33.341666	33.341666	6.197658	5.535778	12.671354	15.697337	26.576408	34.059413
<b>min</b>	3657.000000	1901.000000	0.000000	0.000000	0.000000	0.200000	3.500000	38.200000
<b>25%</b>	3685.500000	1929.500000	0.000000	0.000000	0.500000	12.200000	29.100000	76.350000

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN
<b>50%</b>	3714.000000	1958.000000	0.200000	0.300000	3.800000	22.500000	40.600000	99.900000
<b>75%</b>	3742.500000	1986.500000	2.500000	3.650000	7.500000	32.250000	63.450000	116.300000
<b>max</b>	3771.000000	2015.000000	28.500000	28.400000	109.200000	96.900000	127.300000	235.700000

## EDA AND VISUALIZATION

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

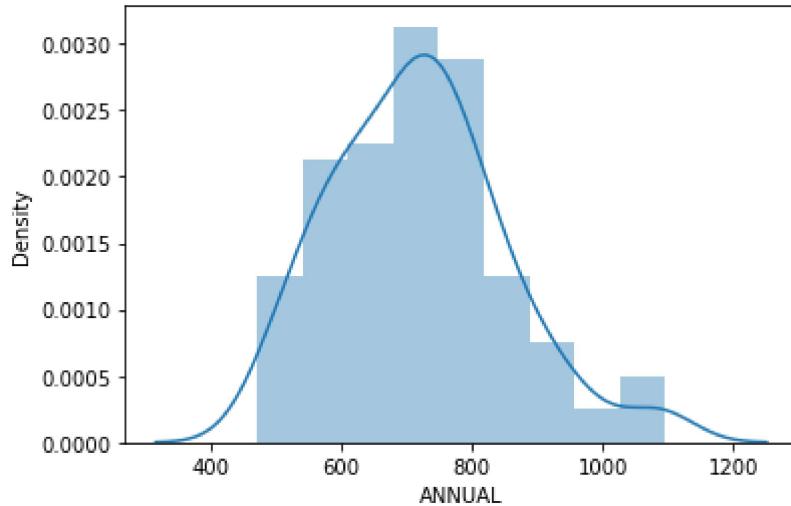
Out[16]: <seaborn.axisgrid.PairGrid at 0x2a93f34bbb0>



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:>
```

