20104016

DEENA

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_west uttar pradesh.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	1127	WEST UTTAR PRADESH	1901	51.4	25.6	9.5	0.7	5.6	23.8	201.9	374.3	67.7	7.6
1	1128	WEST UTTAR PRADESH	1902	4.6	4.6	0.6	4.8	7.2	54.5	325.9	180.6	143.1	9.6
2	1129	WEST UTTAR PRADESH	1903	13.4	0.4	1.2	0.0	8.2	32.7	145.4	279.1	150.4	177.3
3	1130	WEST UTTAR PRADESH	1904	6.3	2.0	29.7	0.4	24.8	68.5	358.8	311.1	97.1	2.7
4	1131	WEST UTTAR PRADESH	1905	32.3	26.6	14.8	3.6	7.1	18.9	139.8	95.0	92.2	0.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	1237	WEST UTTAR PRADESH	2011	2.1	10.4	3.9	2.8	29.6	175.9	215.9	232.3	101.7	0.7
111	1238	WEST UTTAR PRADESH	2012	14.5	0.1	1.4	4.7	0.3	4.0	145.1	149.1	67.8	0.5
112	1239	WEST UTTAR PRADESH	2013	20.4	69.5	3.5	1.6	2.1	190.6	233.9	287.1	52.2	61.2
113	1240	WEST UTTAR PRADESH	2014	48.3	29.4	22.6	5.3	11.0	22.0	151.6	81.0	84.7	14.6
114	1241	WEST UTTAR PRADESH	2015	31.6	7.2	66.8	21.0	8.1	72.0	194.2	143.5	26.5	6.9

115 rows × 20 columns

Data Cleaning and Data Preprocessing

In [3]:
```
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')
```

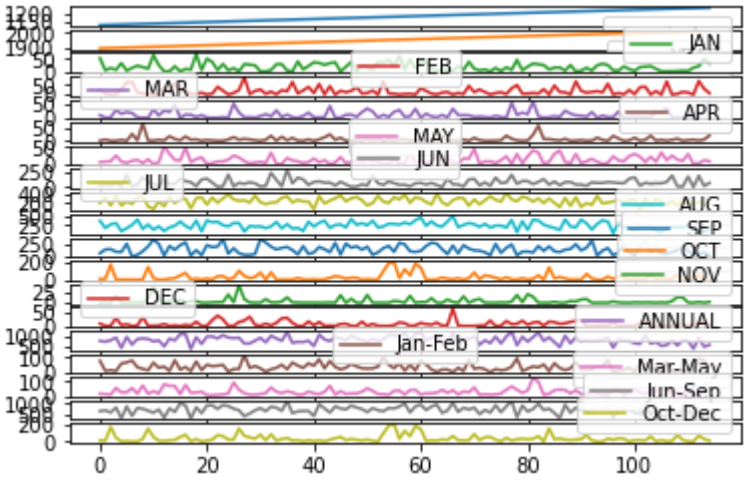
In [5]:
```
df.info()
```

```
<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)
```

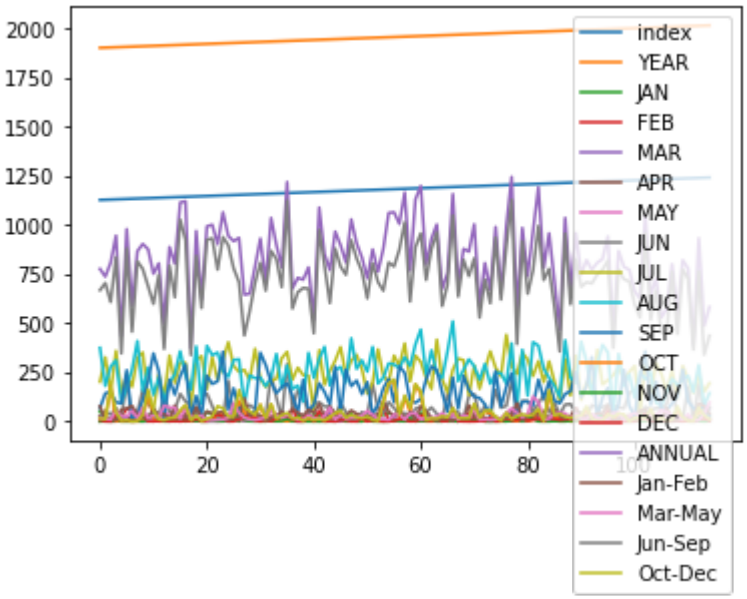
Out[6]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



Line chart

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

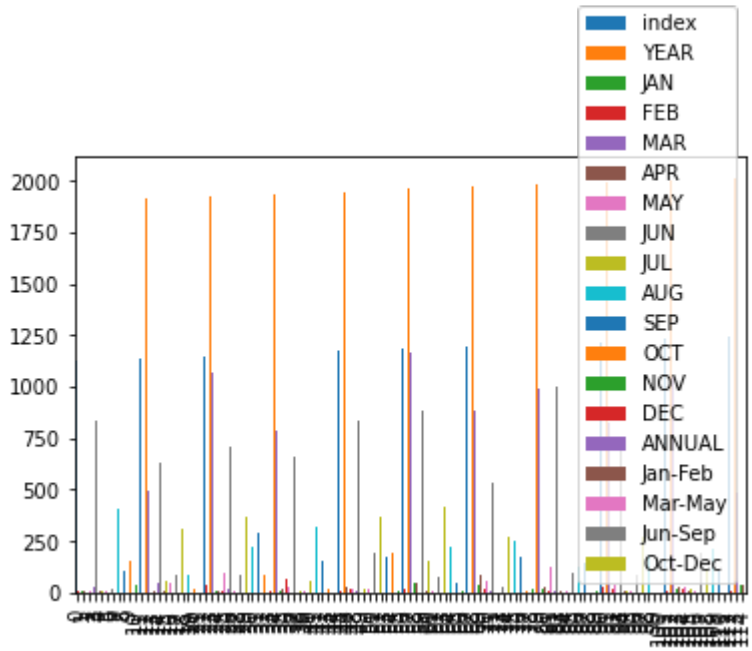
Out[7]:
```
<AxesSubplot:>
```



Bar chart

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

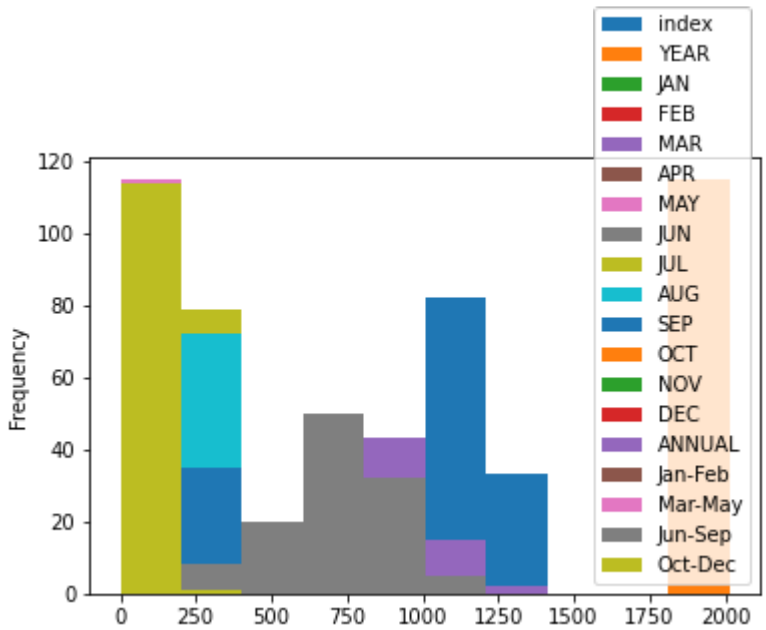
Out[8]: <AxesSubplot:>



Histogram

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

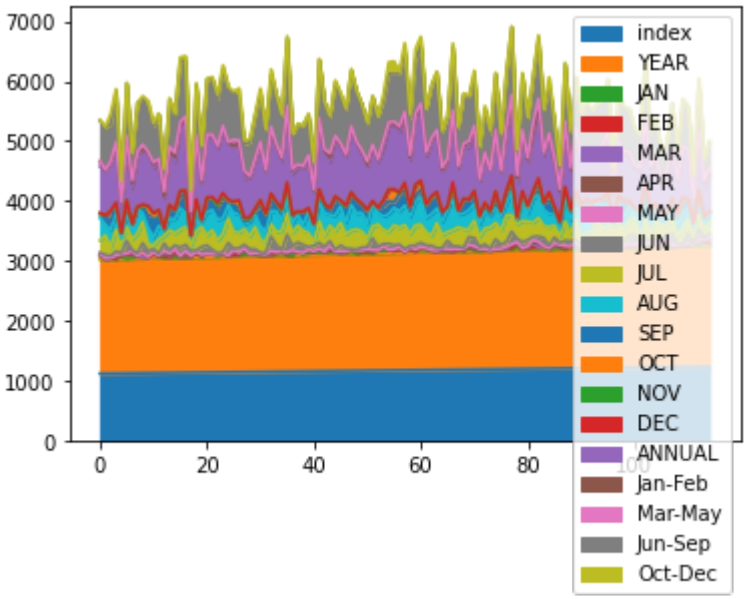
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area chart

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

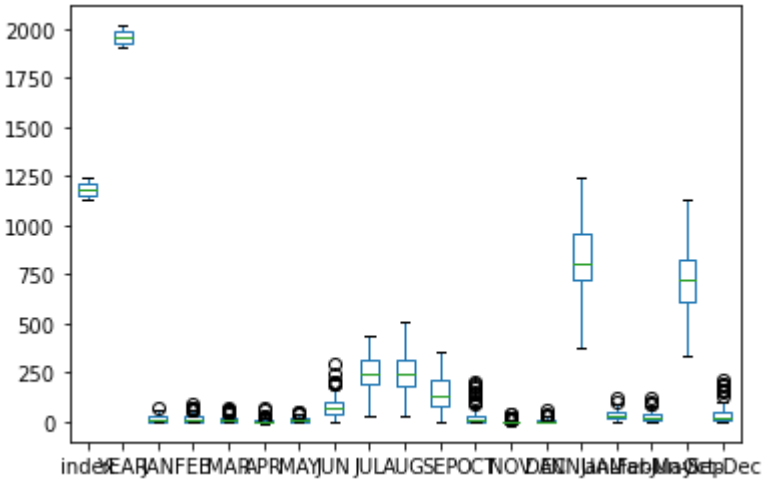
Out[10]: <AxesSubplot:>



Box chart

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

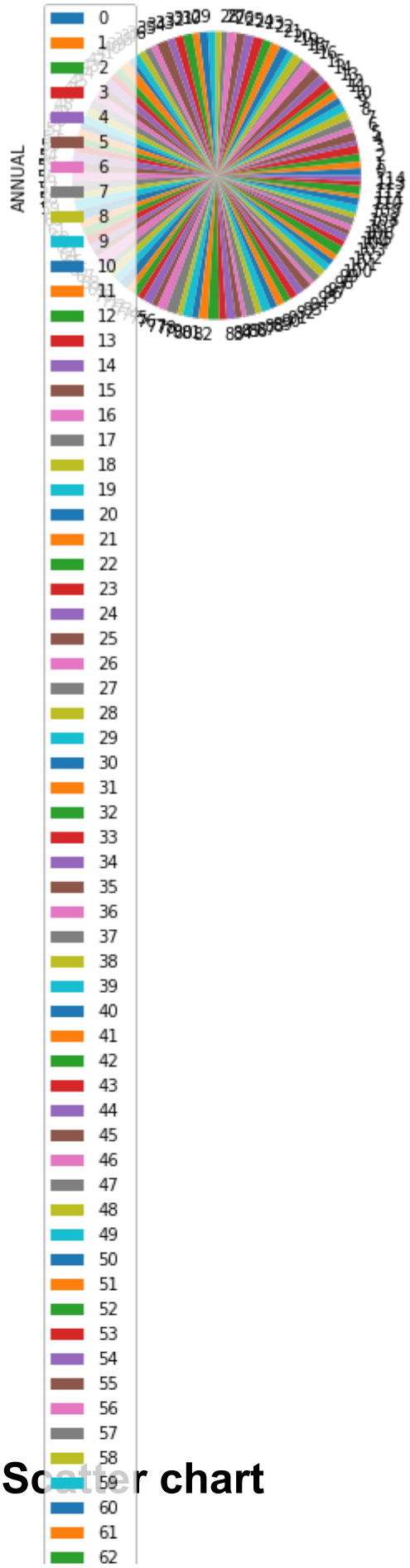
Out[11]: <AxesSubplot:>



Pie chart

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

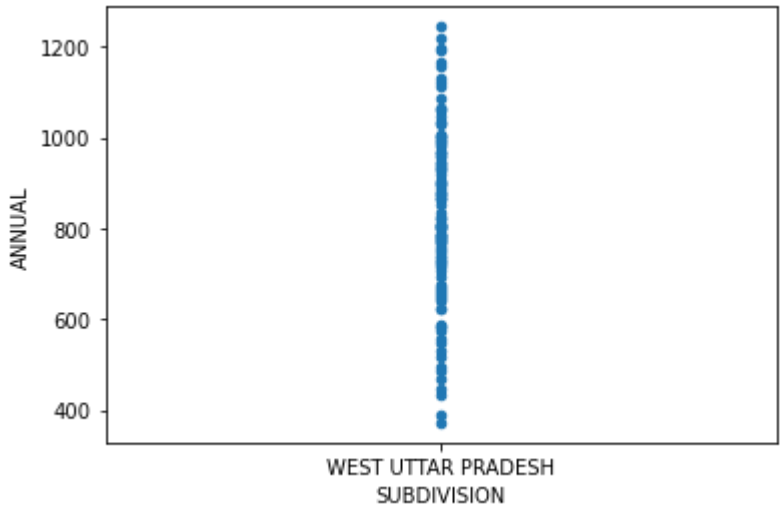
Out[12]:
```
<AxesSubplot:ylabel='ANNUAL'>
```



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
```

In [15]:
```
df.describe()
```

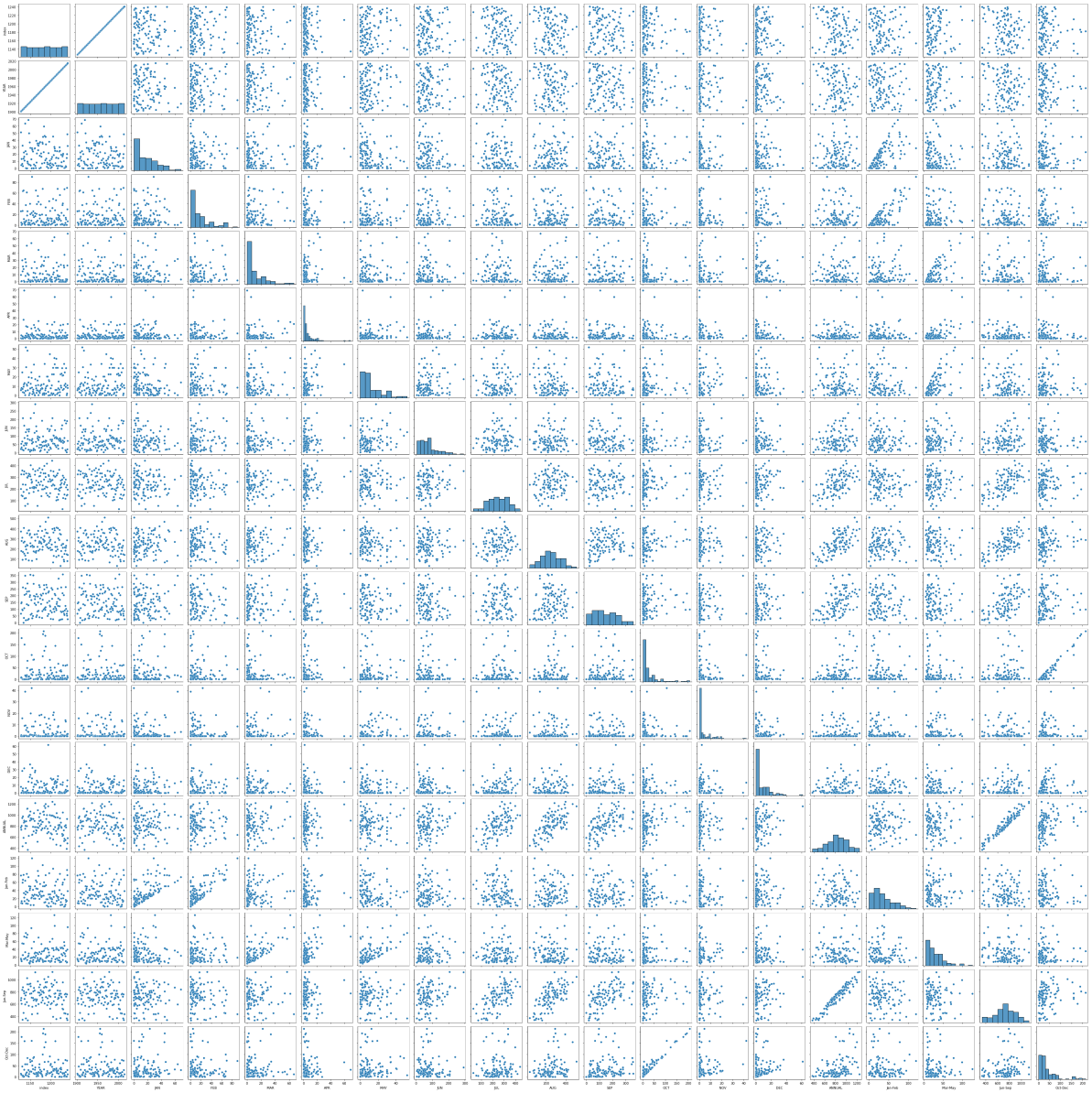
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	11
mean	1184.000000	1958.000000	17.666087	17.893913	11.461739	6.253043	12.306087	7
std	33.341666	33.341666	15.791531	19.972785	14.286434	10.015552	11.528510	5
min	1127.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1155.500000	1929.500000	4.100000	3.450000	1.700000	0.750000	3.750000	3
50%	1184.000000	1958.000000	14.200000	10.400000	5.700000	3.100000	8.100000	6
75%	1212.500000	1986.500000	28.100000	25.600000	18.350000	6.750000	18.300000	9
max	1241.000000	2015.000000	68.600000	89.900000	66.800000	69.000000	52.000000	29

EDA AND VISUALIZATION

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

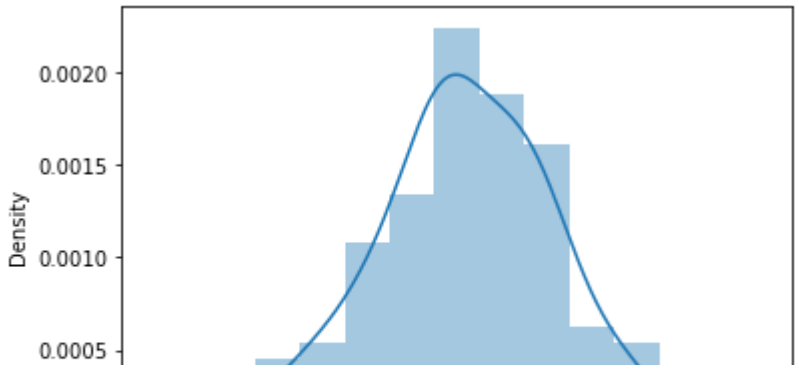
Out[16]:
```
<seaborn.axisgrid.PairGrid at 0x18d65217a60>
```



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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

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



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

Out[18]: <AxesSubplot:>