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_konkan _ goa.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	O
0	2507	KONKAN & GOA	1901	5.6	0.1	0.4	35.7	19.9	746.1	1075.5	748.0	117.4	3
1	2508	KONKAN & GOA	1902	0.3	0.0	0.0	0.4	7.6	428.2	943.6	515.1	613.8	7
2	2509	KONKAN & GOA	1903	0.0	0.0	0.1	0.0	201.1	470.5	1298.6	673.9	285.1	14
3	2510	KONKAN & GOA	1904	0.0	0.1	6.6	6.3	4.6	975.8	771.7	321.3	217.0	9
4	2511	KONKAN & GOA	1905	0.1	0.1	0.0	0.4	8.6	293.7	770.6	305.5	208.3	8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
110	2617	KONKAN & GOA	2011	0.0	0.0	0.0	3.4	1.1	857.0	1384.1	987.9	468.3	12
111	2618	KONKAN & GOA	2012	0.0	0.0	0.0	0.6	1.1	633.0	928.5	762.5	515.3	17
112	2619	KONKAN & GOA	2013	1.8	5.4	0.1	0.1	18.5	1028.3	1478.5	497.6	340.7	14
113	2620	KONKAN & GOA	2014	1.3	5.3	1.8	0.7	21.3	238.2	1293.2	658.0	419.5	9
114	2621	KONKAN & GOA	2015	2.7	0.0	36.8	3.6	11.3	764.0	526.5	377.3	240.9	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')
```

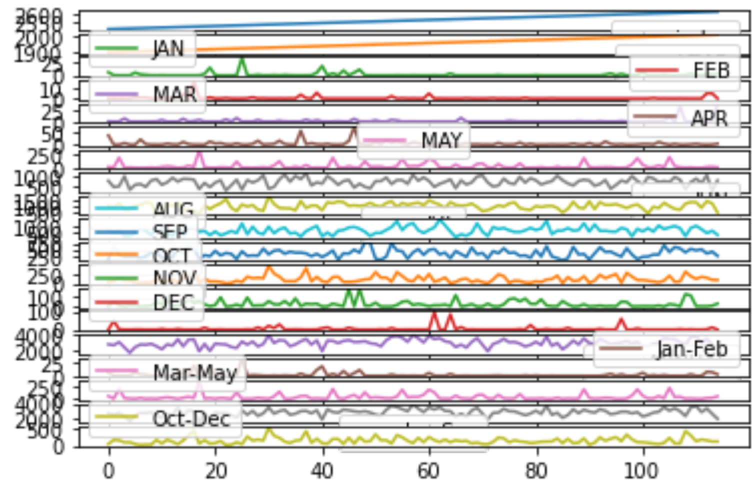
In [5]:
```
df.info()
```

```
<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)
```

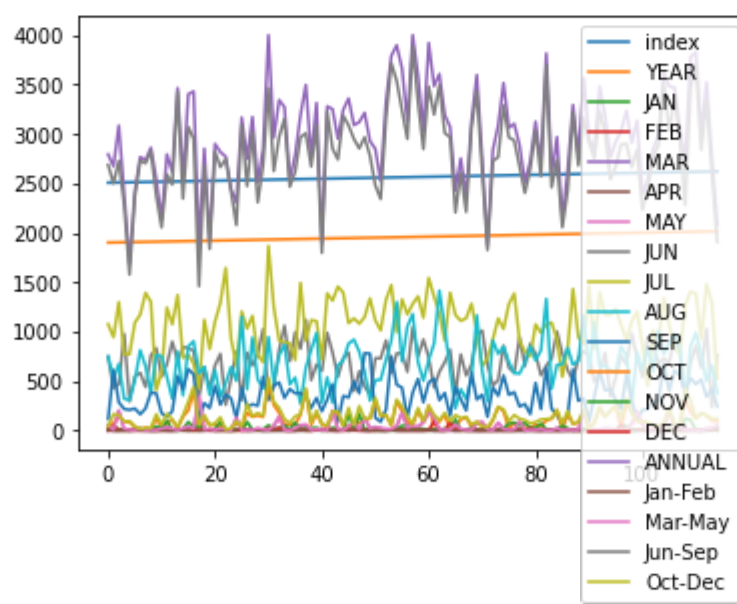
Out[6]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



Line chart

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

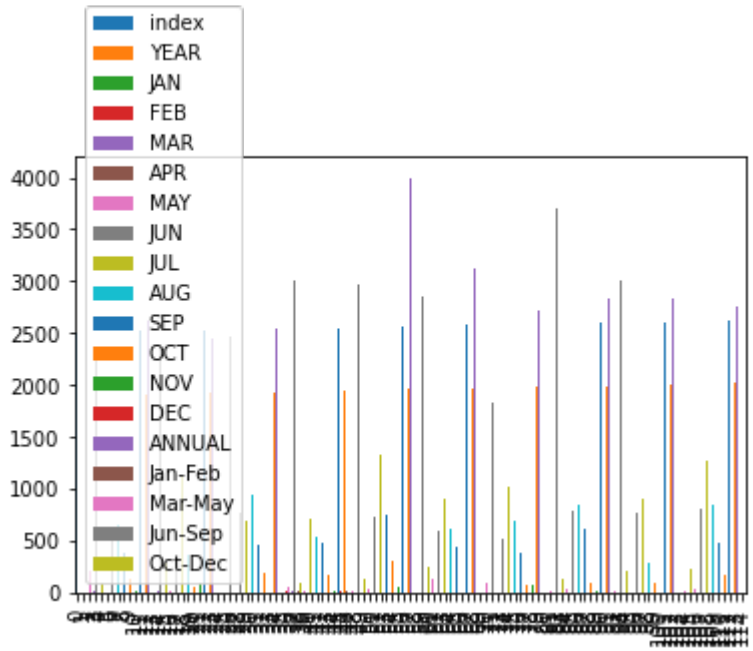
Out[7]: <AxesSubplot:>



Bar chart

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

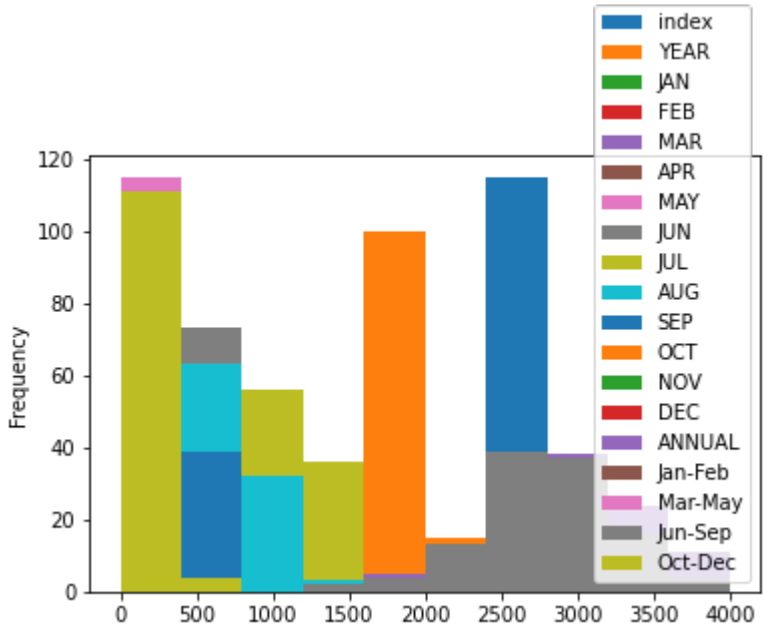
Out[8]: <AxesSubplot:>



Histogram

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

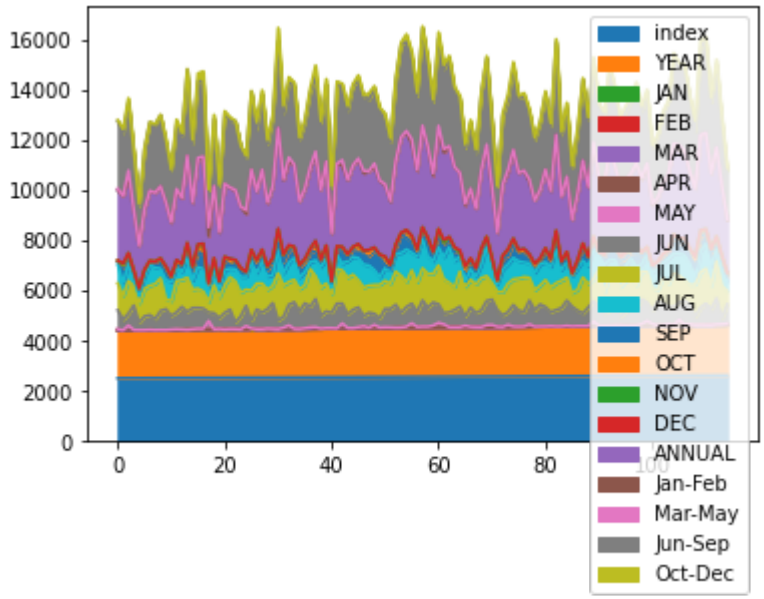
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area chart

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

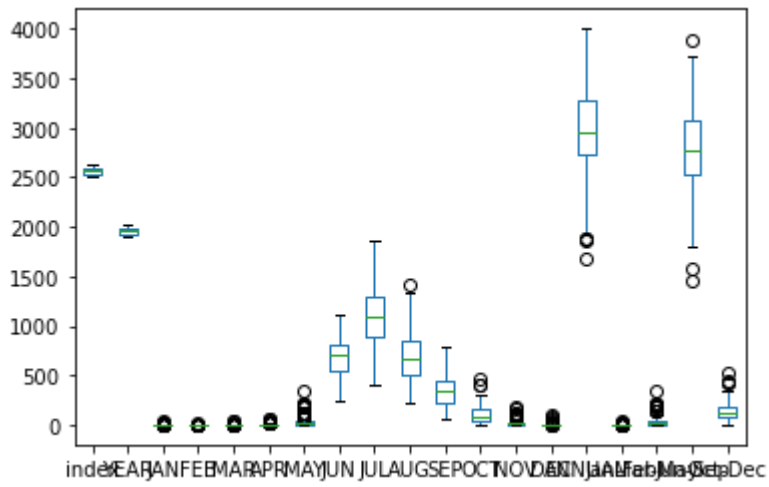
Out[10]: <AxesSubplot:>



Box chart

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

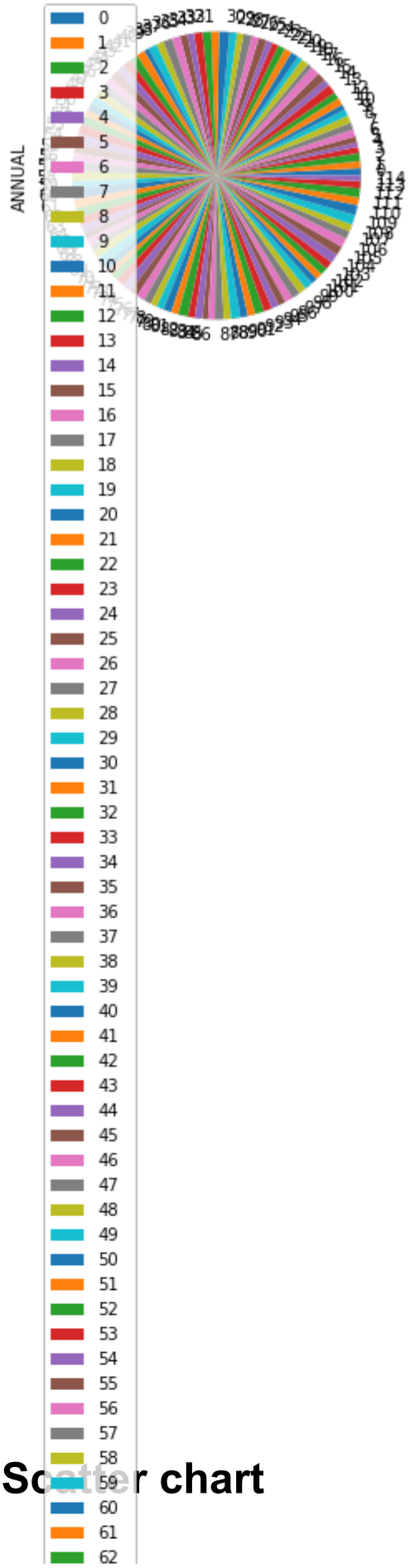
Out[11]: <AxesSubplot:>



Pie chart

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

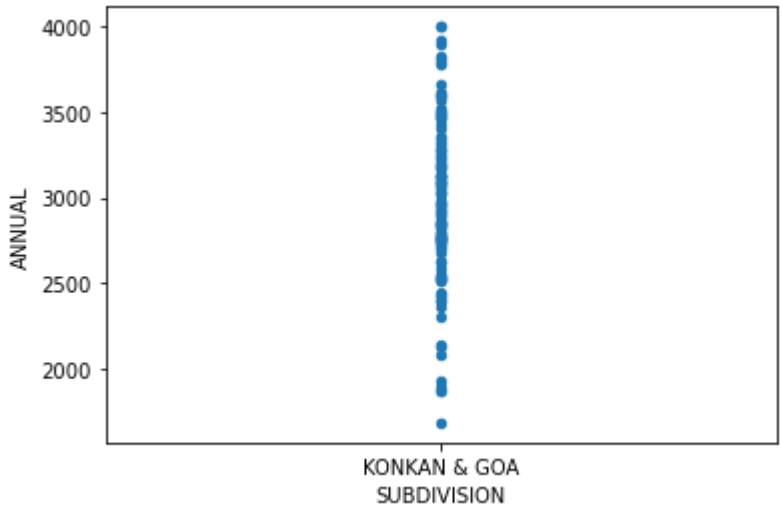
Out[12]:
```
<AxesSubplot:ylabel='ANNUAL'>
```



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()
```

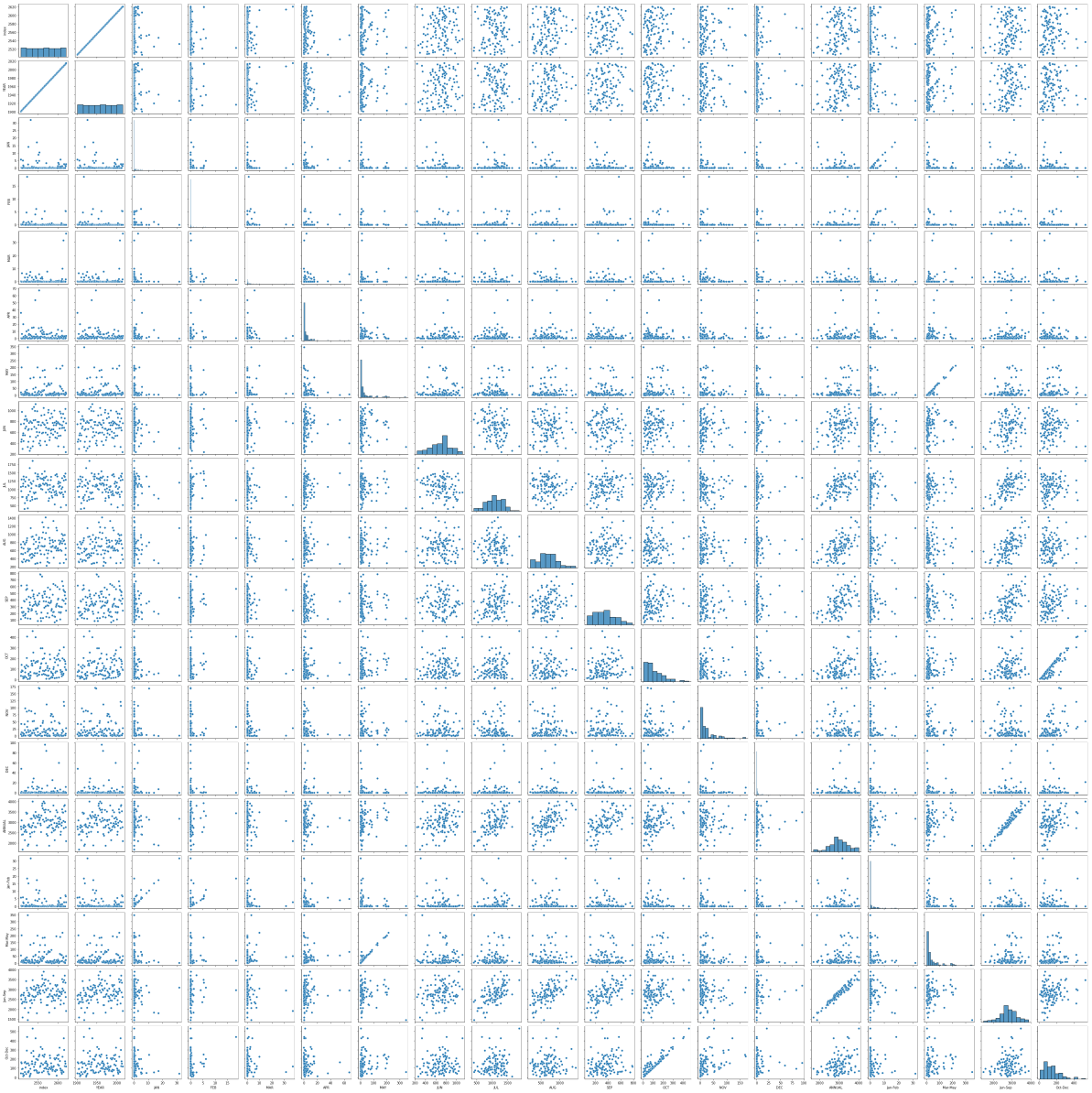
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	1
mean	2564.000000	1958.000000	1.262609	0.546957	1.374783	4.266087	33.515652	6
std	33.341666	33.341666	3.884234	2.048034	4.749309	9.103141	58.327263	1
min	2507.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2
25%	2535.500000	1929.500000	0.000000	0.000000	0.000000	0.300000	2.900000	5
50%	2564.000000	1958.000000	0.000000	0.000000	0.100000	1.300000	9.700000	7
75%	2592.500000	1986.500000	0.500000	0.100000	0.400000	4.200000	30.300000	8
max	2621.000000	2015.000000	31.800000	18.400000	36.800000	67.300000	345.400000	11

EDA AND VISUALIZATION

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

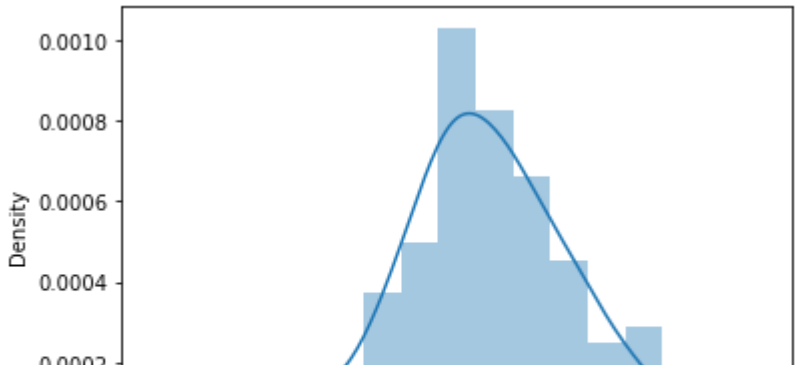
Out[16]: <seaborn.axisgrid.PairGrid at 0x1a3c0e42a60>



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:>