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_gujarat region.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	2277	GUJARAT REGION	1901	4.2	0.0	0.6	1.6	7.0	60.3	240.2	205.4	18.1	16.6
1	2278	GUJARAT REGION	1902	3.9	0.0	0.0	0.6	1.0	32.8	229.8	299.0	281.2	2.3
2	2279	GUJARAT REGION	1903	0.3	0.1	1.4	0.0	12.3	30.1	452.9	202.0	183.2	5.4
3	2280	GUJARAT REGION	1904	0.8	10.6	16.8	0.2	3.9	48.3	194.8	71.8	138.0	6.1
4	2281	GUJARAT REGION	1905	0.1	0.7	1.1	0.3	0.0	20.1	668.3	37.9	81.3	1.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	2387	GUJARAT REGION	2011	0.0	0.2	0.0	0.0	0.0	16.3	259.2	451.7	162.5	0.4
111	2388	GUJARAT REGION	2012	0.1	0.0	0.0	0.0	0.0	34.4	178.2	230.3	263.8	7.1
112	2389	GUJARAT REGION	2013	0.0	0.9	0.1	4.6	0.0	155.7	405.4	211.1	287.3	53.2
113	2390	GUJARAT REGION	2014	5.7	0.1	0.2	1.0	1.3	11.6	307.5	138.6	235.1	3.3
114	2391	GUJARAT REGION	2015	1.8	0.0	6.1	5.5	0.9	120.7	354.7	37.4	93.4	2.2

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')
```

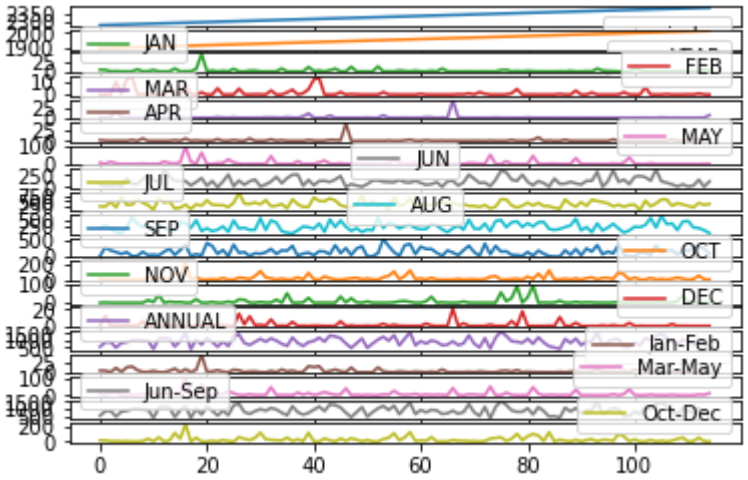
In [5]:
```
df.info()
```

```
<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)
```

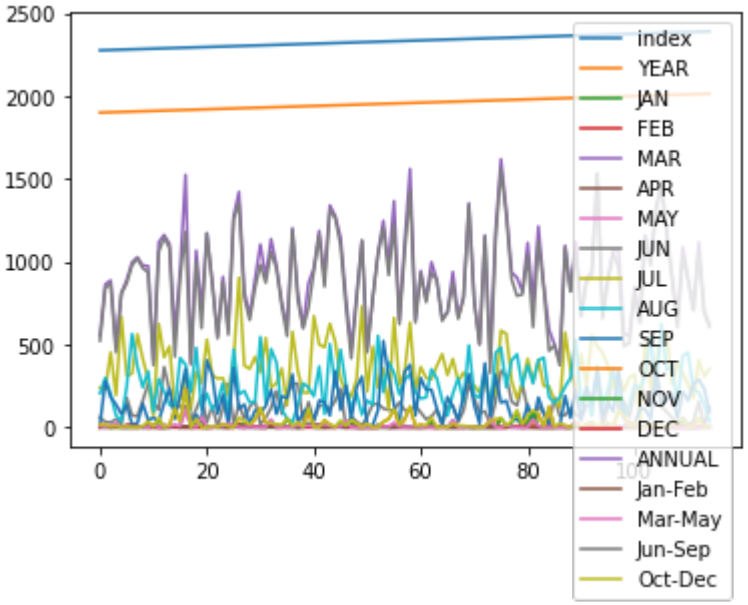
Out[6]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)
```



Line chart

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

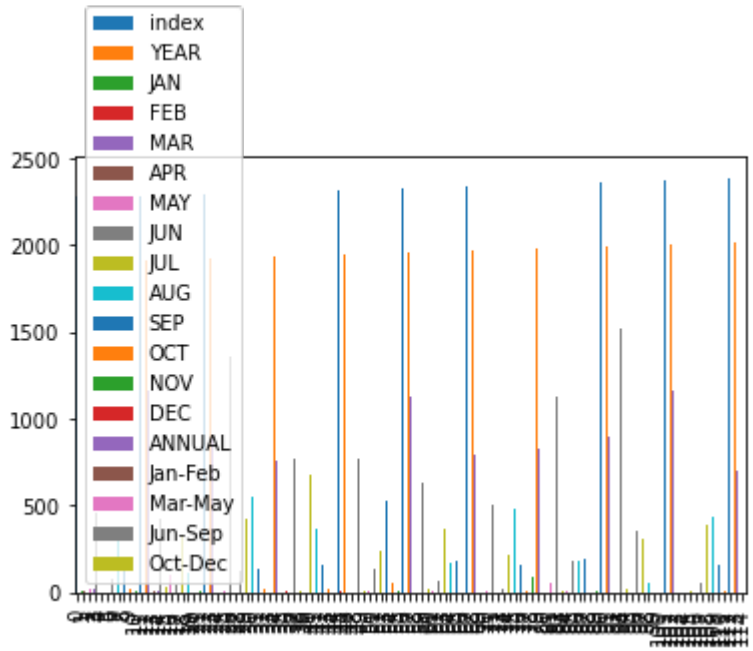
Out[7]:
```
<AxesSubplot:>
```



Bar chart

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

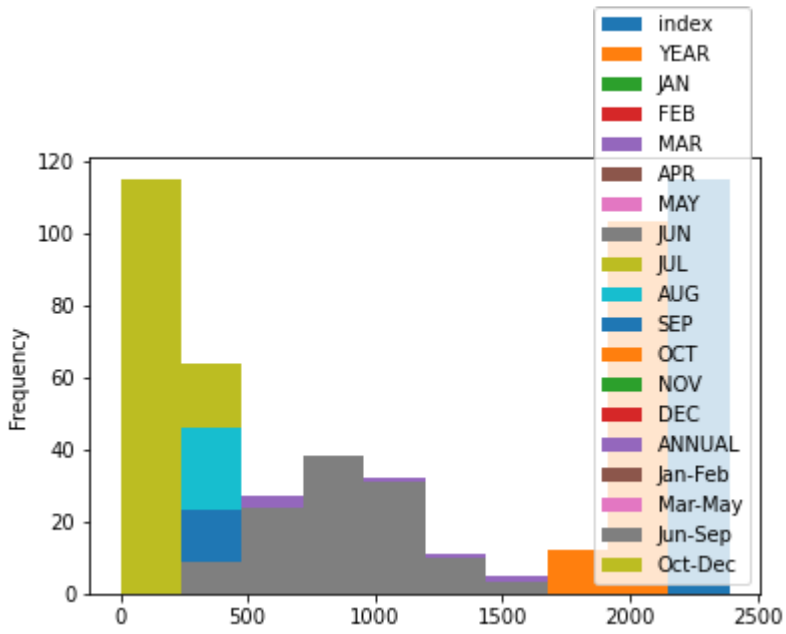
Out[8]: <AxesSubplot:>



Histogram

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

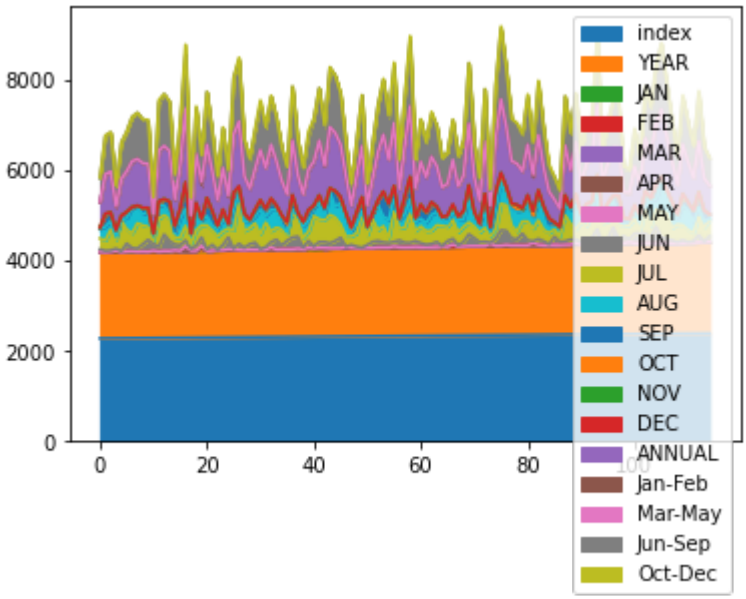
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area chart

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

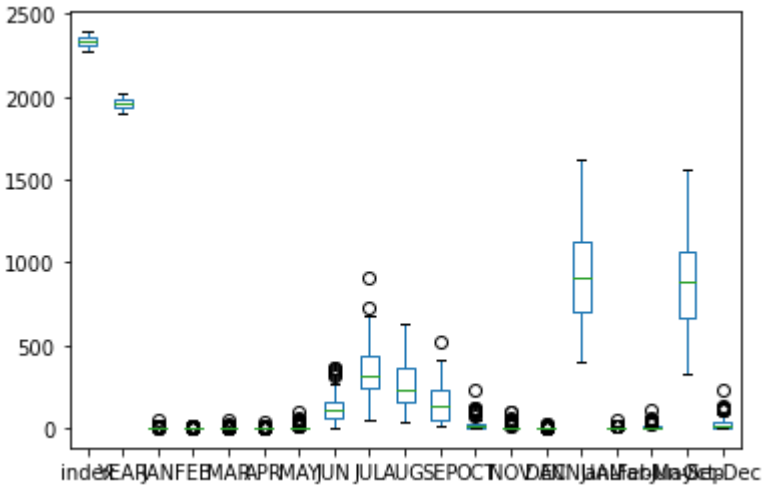
Out[10]: <AxesSubplot:>



Box chart

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

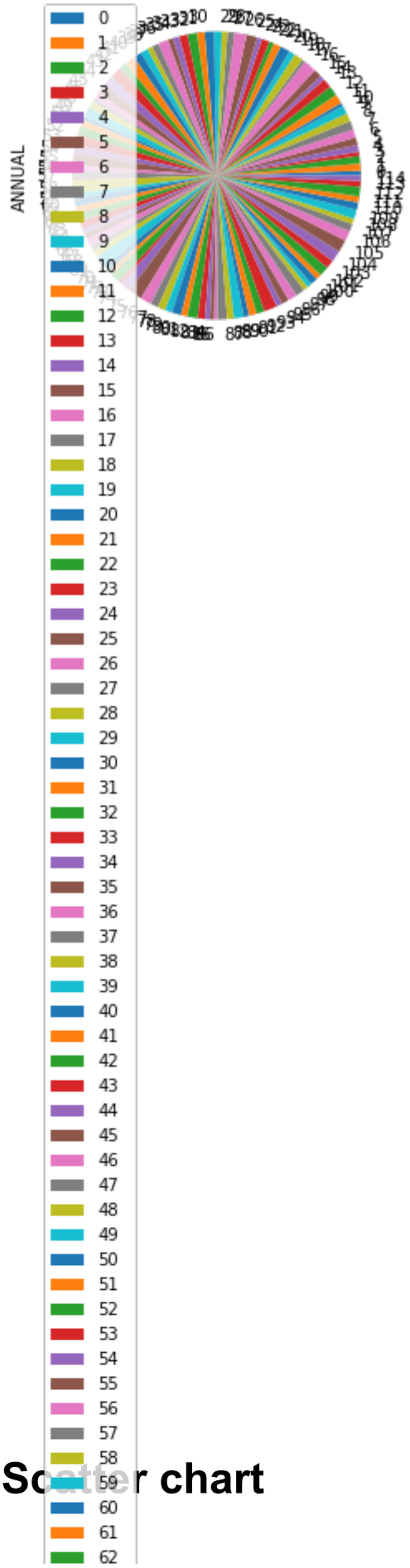
Out[11]: <AxesSubplot:>



Pie chart

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

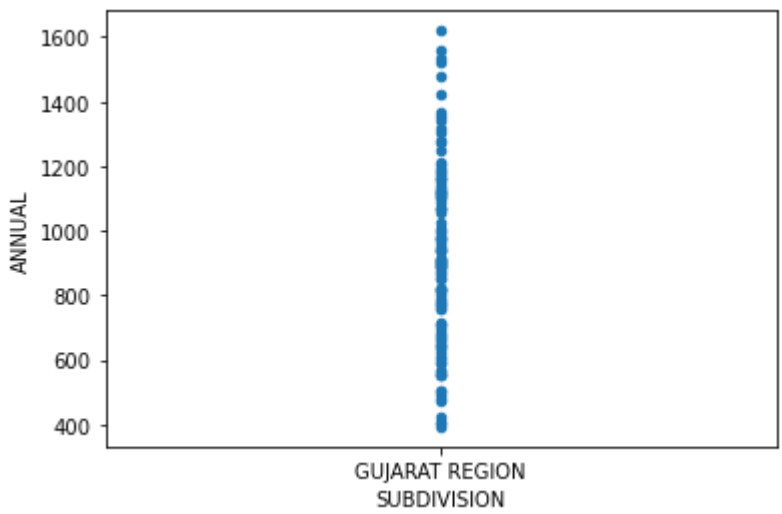
Out[12]:
```
<AxesSubplot:ylabel='ANNUAL'>
```



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()
```

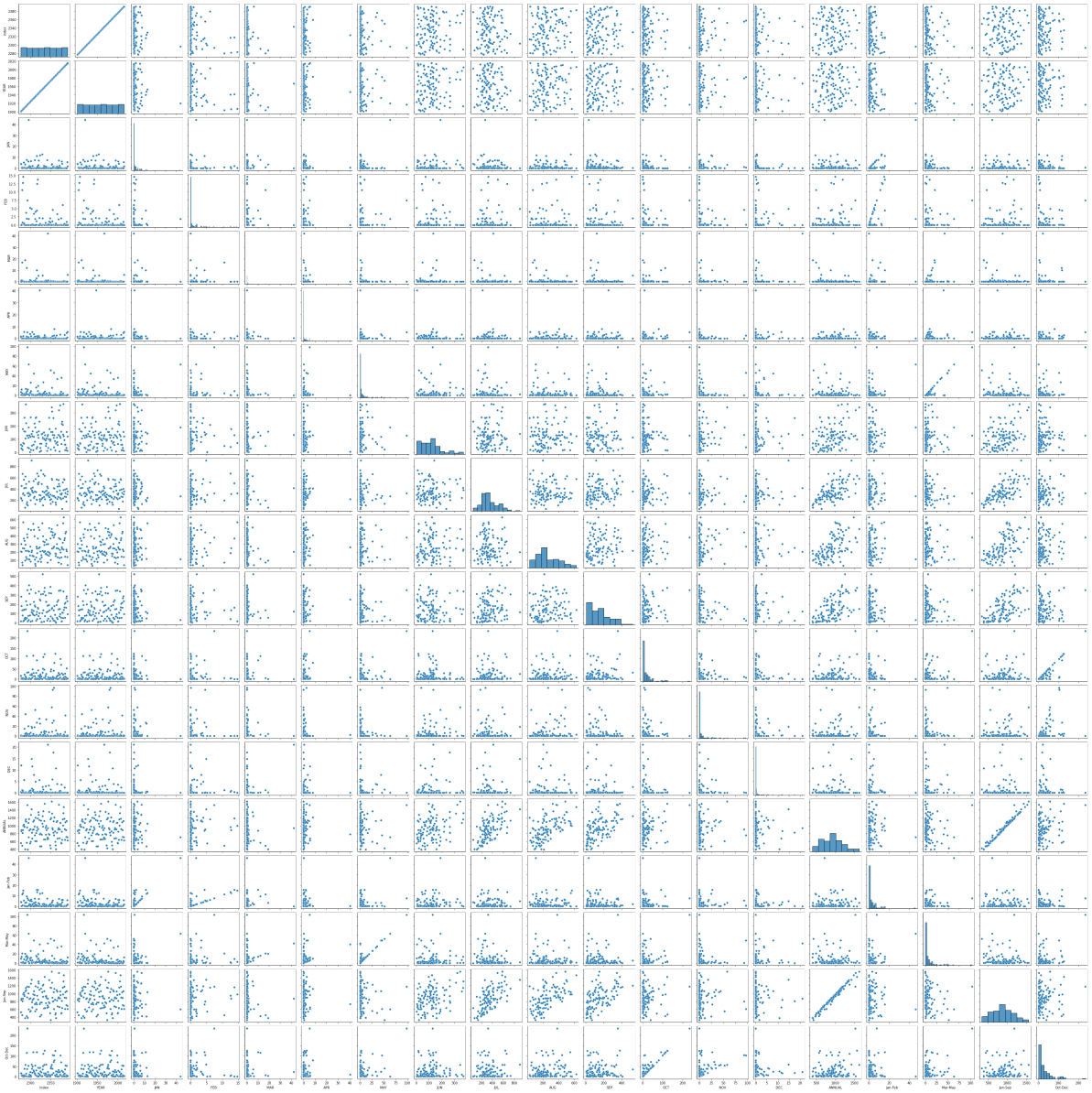
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	11
mean	2334.000000	1958.000000	1.786087	1.191304	1.220870	1.116522	5.809565	12
std	33.341666	33.341666	4.762590	2.870710	4.784102	3.980389	13.981353	8
min	2277.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	2305.500000	1929.500000	0.000000	0.000000	0.000000	0.000000	0.100000	5
50%	2334.000000	1958.000000	0.100000	0.000000	0.000000	0.100000	0.900000	11
75%	2362.500000	1986.500000	1.500000	0.650000	0.250000	0.750000	4.100000	15
max	2391.000000	2015.000000	44.100000	14.600000	42.100000	40.400000	98.300000	36

EDA AND VISUALIZATION

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

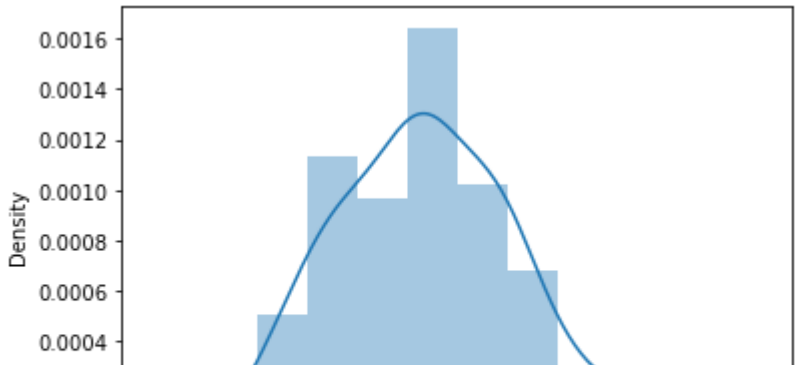
Out[16]: <seaborn.axisgrid.PairGrid at 0x1935347af40>



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:>