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_rayalseema.csv")
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
0	3312	RAYALSEEMA	1901	7.0	50.2	0.0	12.1	38.9	53.0	73.4	60.3	109.0	81.6
1	3313	RAYALSEEMA	1902	10.0	0.2	1.7	11.0	36.8	73.6	41.3	148.3	181.7	188.5
2	3314	RAYALSEEMA	1903	30.0	0.1	0.0	3.6	80.5	67.5	127.5	140.6	219.7	95.3
3	3315	RAYALSEEMA	1904	14.8	0.0	1.7	7.1	58.8	39.8	75.1	19.4	84.7	111.5
4	3316	RAYALSEEMA	1905	6.5	6.8	17.0	18.3	44.2	66.1	50.9	219.3	36.5	180.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
110	3422	RAYALSEEMA	2011	0.8	12.1	0.0	34.6	33.0	44.5	128.9	163.6	71.2	107.5
111	3423	RAYALSEEMA	2012	2.7	0.0	2.5	32.7	38.8	47.0	139.7	120.0	69.5	113.7
112	3424	RAYALSEEMA	2013	1.3	30.6	11.5	26.8	38.9	73.8	95.7	110.3	163.2	169.3
113	3425	RAYALSEEMA	2014	0.2	0.7	12.5	5.1	46.7	66.3	68.7	115.1	81.4	104.6
114	3426	RAYALSEEMA	2015	1.9	0.0	13.4	73.4	39.7	73.0	43.1	123.6	136.3	106.7

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')
```

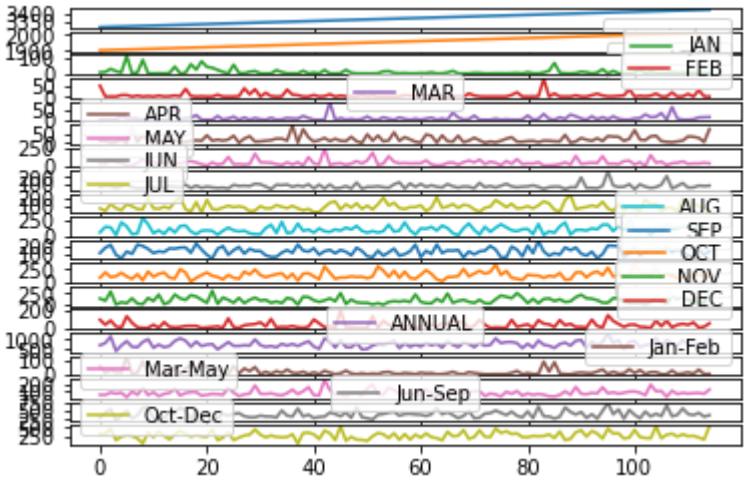
In [5]:
```
df.info()
```

```
<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)
```

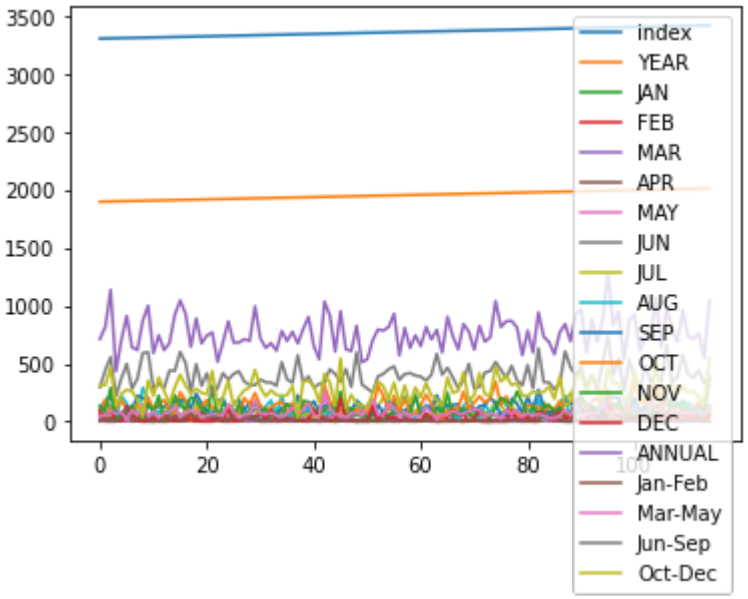
Out[6]:
```
array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
       <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
    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



Line chart

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

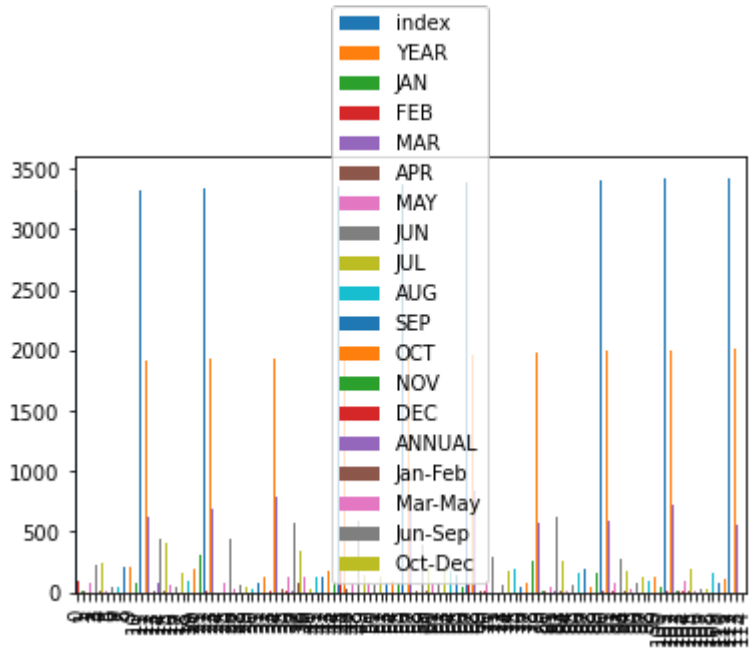
Out[7]:
```
<AxesSubplot:>
```



Bar chart

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

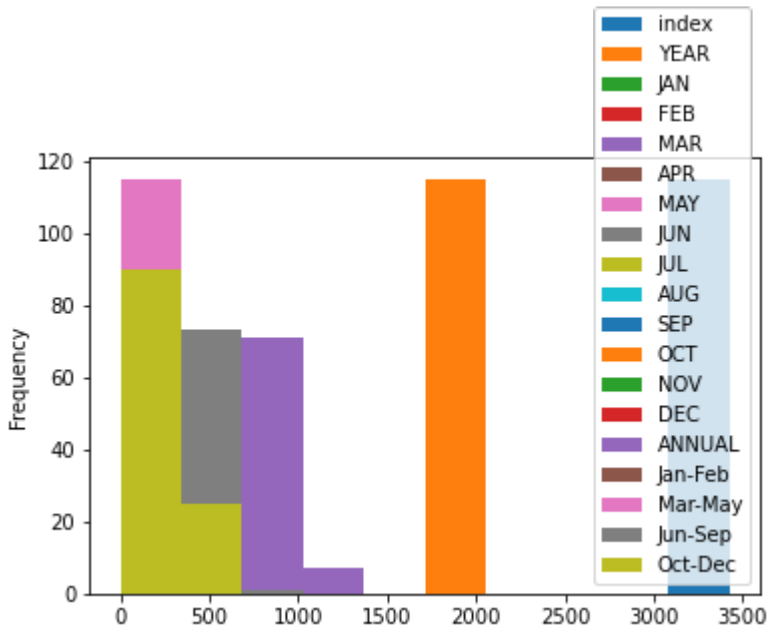
Out[8]: <AxesSubplot:>



Histogram

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

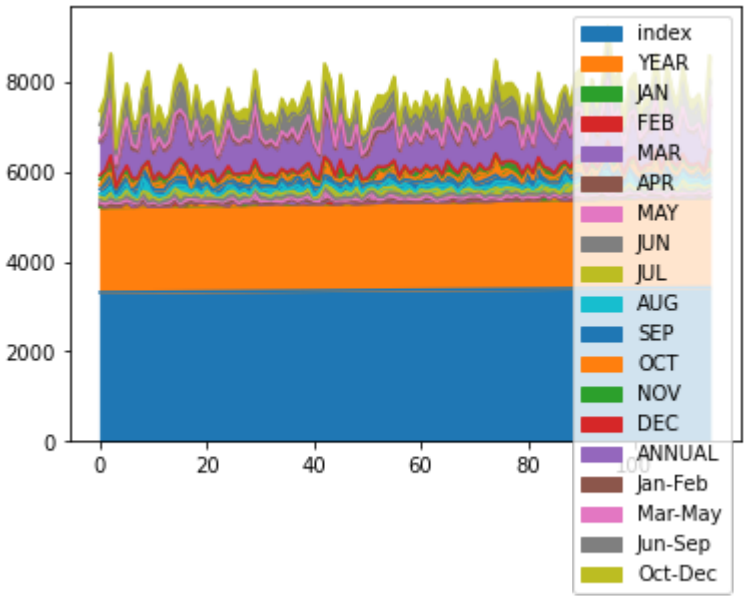
Out[9]: <AxesSubplot:ylabel='Frequency'>



Area chart

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

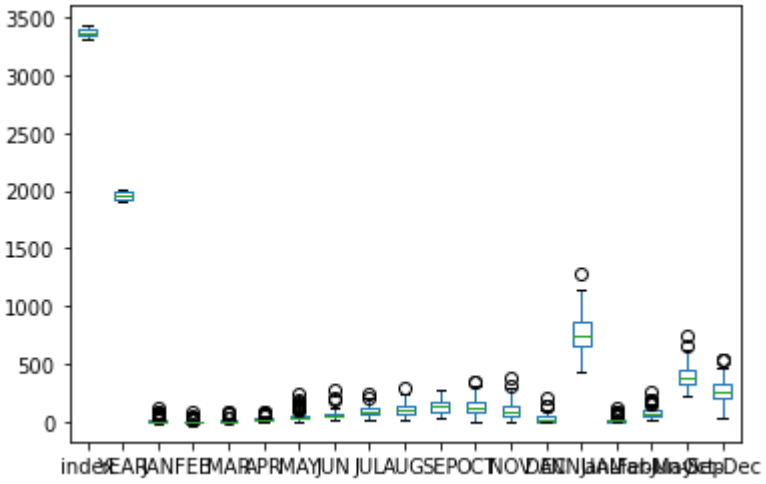
Out[10]: <AxesSubplot:>



Box chart

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

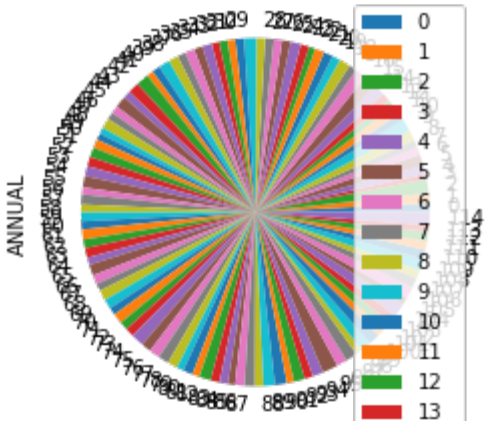
Out[11]: <AxesSubplot:>

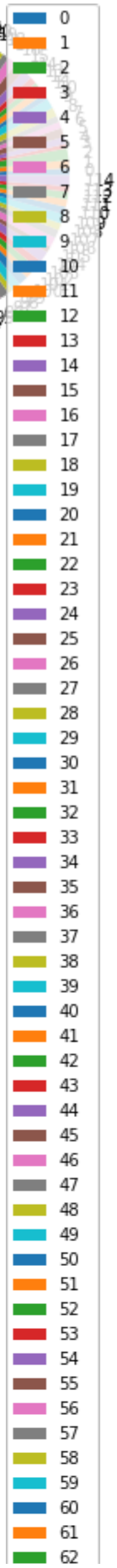


Pie chart

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

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

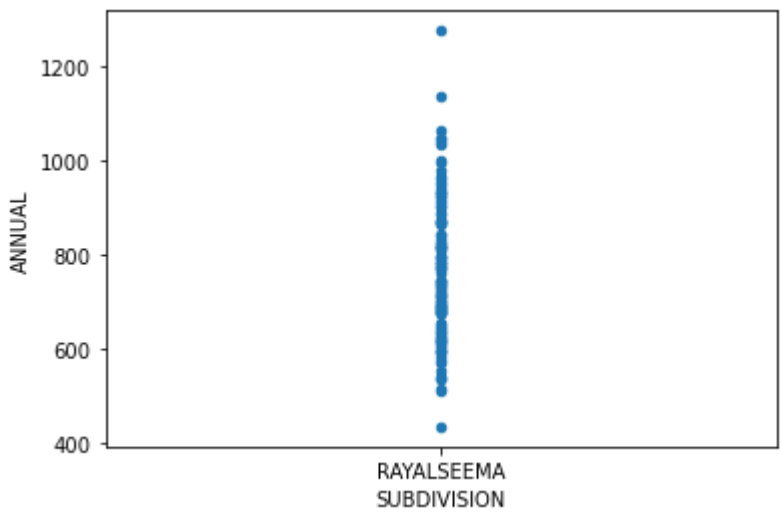




Scatter chart

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()
```

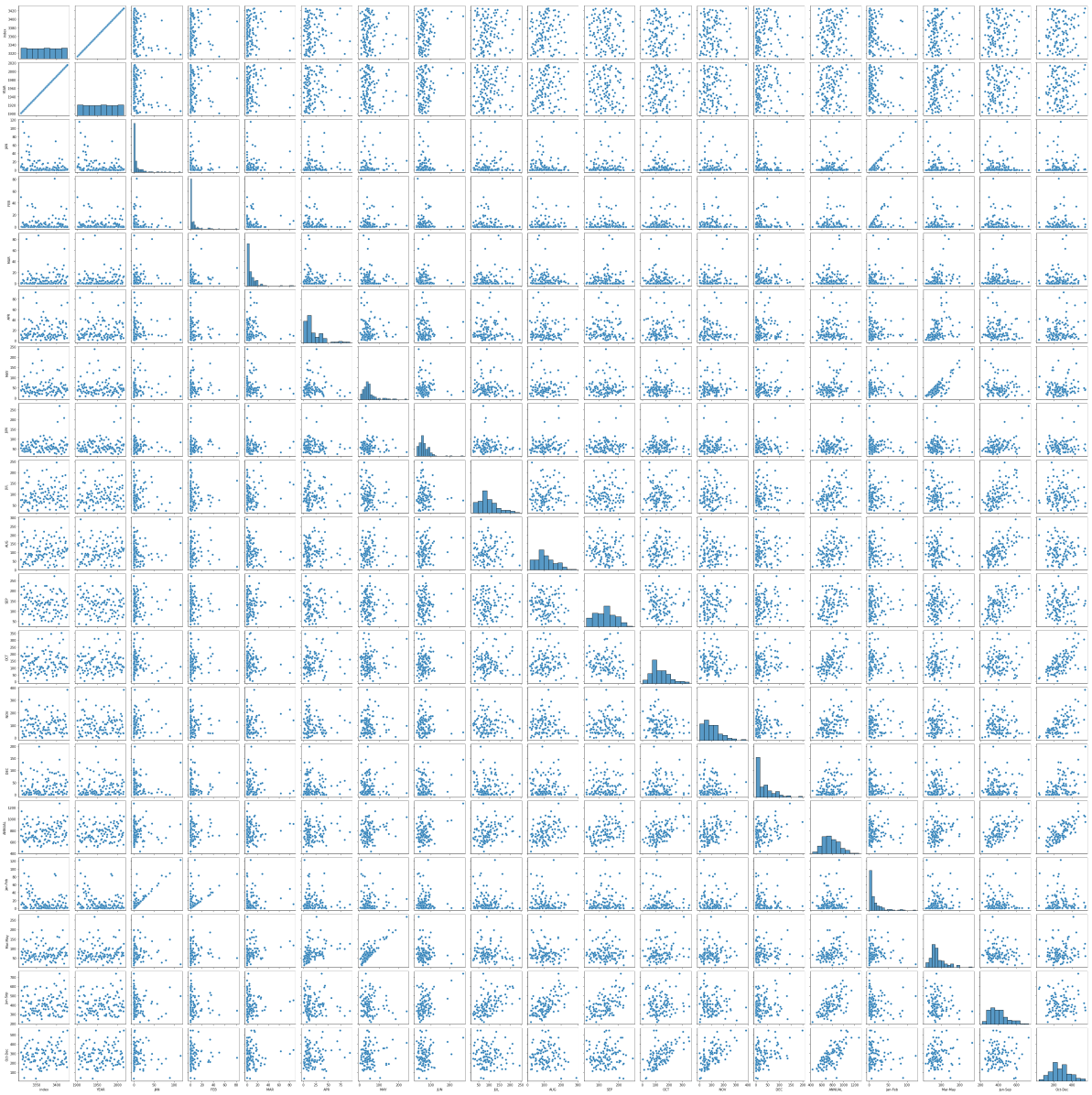
Out[15]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	11
mean	3369.000000	1958.000000	9.867826	5.680000	8.076522	19.808696	50.475652	6
std	33.341666	33.341666	19.121736	11.652063	13.711580	17.575449	37.569280	3
min	3312.000000	1901.000000	0.000000	0.000000	0.000000	0.700000	4.100000	2
25%	3340.500000	1929.500000	0.200000	0.000000	0.300000	8.250000	29.150000	4
50%	3369.000000	1958.000000	1.900000	1.000000	4.000000	12.400000	41.400000	5
75%	3397.500000	1986.500000	9.900000	5.700000	10.850000	27.250000	55.150000	7
max	3426.000000	2015.000000	115.300000	81.000000	86.900000	93.500000	239.800000	27

EDA AND VISUALIZATION

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

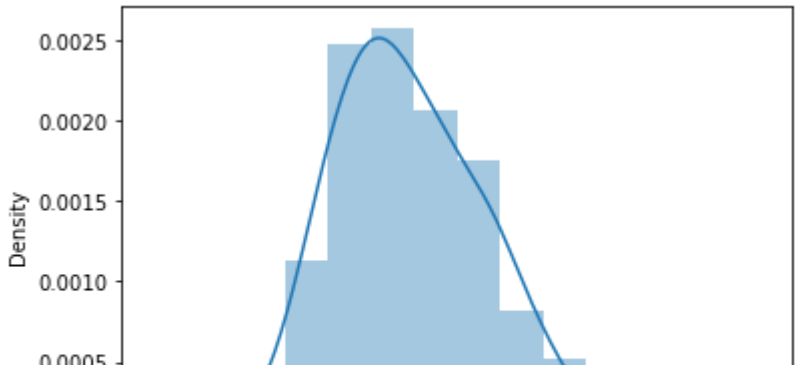
Out[16]: <seaborn.axisgrid.PairGrid at 0x25f0dd7eeb0>



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:>