Import Libraies

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\FP2_RainFall\rainfall in india 1901-2015.csv")[2047:216
df
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL
2047	2047	WEST MADHYA PRADESH	1901	25.8	5.8	5.8	2.8	2.1	41.2	228.9	349.9	47.9	5.6	0.0	2.4	718.2
2048	2048	WEST MADHYA PRADESH	1902	22.1	8.4	0.0	2.0	5.9	35.9	401.9	179.4	194.1	37.9	10.0	14.2	911.7
2049	2049	WEST MADHYA PRADESH	1903	5.3	0.0	0.0	0.0	22.3	50.6	304.9	261.1	250.2	55.1	0.0	0.0	949.6
2050	2050	WEST MADHYA PRADESH	1904	3.2	15.5	14.8	0.0	12.0	96.6	273.0	218.6	125.9	3.3	1.8	9.6	774.4
2051	2051	WEST MADHYA PRADESH	1905	3.5	4.4	1.1	0.8	3.0	36.1	326.3	137.6	183.5	0.3	0.0	0.0	696.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2156	2156	WEST MADHYA PRADESH	2010	2.2	5.4	0.8	0.1	0.4	62.2	258.5	291.5	136.1	13.6	4.6	0.8	776.3
2157	2157	WEST MADHYA PRADESH	2011	0.0	1.7	0.1	1.8	3.6	241.5	306.7	343.3	165.0	0.2	0.0	0.0	1063.9
2158	2158	WEST MADHYA PRADESH	2012	6.2	0.0	0.0	0.9	3.1	48.2	439.2	341.2	194.3	2.1	0.0	0.0	1035.2
2159	2159	WEST MADHYA PRADESH	2013	1.7	31.1	8.5	2.8	0.4	263.7	485.1	432.6	98.9	68.7	0.3	2.4	1396.3
2160	2160	WEST MADHYA PRADESH	2014	25.6	34.4	4.6	1.4	1.4	30.6	337.4	211.0	192.6	7.0	3.0	15.8	864.9

114 rows × 20 columns

Data Cleaning and Preprocessing

In [3]:
```
df.dropna()
```

Out[3]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL
2047	2047	WEST MADHYA PRADESH	1901	25.8	5.8	5.8	2.8	2.1	41.2	228.9	349.9	47.9	5.6	0.0	2.4	718.2
2048	2048	WEST MADHYA PRADESH	1902	22.1	8.4	0.0	2.0	5.9	35.9	401.9	179.4	194.1	37.9	10.0	14.2	911.7
2049	2049	WEST MADHYA PRADESH	1903	5.3	0.0	0.0	0.0	22.3	50.6	304.9	261.1	250.2	55.1	0.0	0.0	949.6
2050	2050	WEST MADHYA PRADESH	1904	3.2	15.5	14.8	0.0	12.0	96.6	273.0	218.6	125.9	3.3	1.8	9.6	774.4
2051	2051	WEST MADHYA PRADESH	1905	3.5	4.4	1.1	0.8	3.0	36.1	326.3	137.6	183.5	0.3	0.0	0.0	696.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2156	2156	WEST MADHYA PRADESH	2010	2.2	5.4	0.8	0.1	0.4	62.2	258.5	291.5	136.1	13.6	4.6	0.8	776.3
2157	2157	WEST MADHYA PRADESH	2011	0.0	1.7	0.1	1.8	3.6	241.5	306.7	343.3	165.0	0.2	0.0	0.0	1063.9
2158	2158	WEST MADHYA PRADESH	2012	6.2	0.0	0.0	0.9	3.1	48.2	439.2	341.2	194.3	2.1	0.0	0.0	1035.2
2159	2159	WEST MADHYA PRADESH	2013	1.7	31.1	8.5	2.8	0.4	263.7	485.1	432.6	98.9	68.7	0.3	2.4	1396.3
2160	2160	WEST MADHYA PRADESH	2014	25.6	34.4	4.6	1.4	1.4	30.6	337.4	211.0	192.6	7.0	3.0	15.8	864.9

113 rows × 20 columns

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')
```

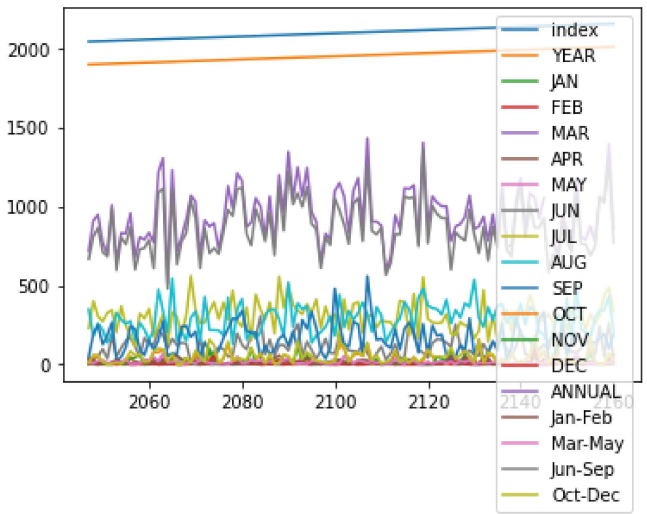
In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 114 entries, 2047 to 2160
Data columns (total 20 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   index        114 non-null    int64  
 1   SUBDIVISION  114 non-null    object 
 2   YEAR         114 non-null    int64  
 3   JAN          114 non-null    float64
 4   FEB          113 non-null    float64
 5   MAR          114 non-null    float64
 6   APR          114 non-null    float64
 7   MAY          114 non-null    float64
 8   JUN          114 non-null    float64
 9   JUL          114 non-null    float64
 10  AUG          114 non-null    float64
 11  SEP          114 non-null    float64
 12  OCT          114 non-null    float64
 13  NOV          114 non-null    float64
 14  DEC          114 non-null    float64
 15  ANNUAL       113 non-null    float64
 16  Jan-Feb      113 non-null    float64
 17  Mar-May      114 non-null    float64
 18  Jun-Sep      114 non-null    float64
 19  Oct-Dec      114 non-null    float64
dtypes: float64(17), int64(2), object(1)
memory usage: 17.9+ KB
```

Line chart

In [6]:
```
df.plot.line()
```

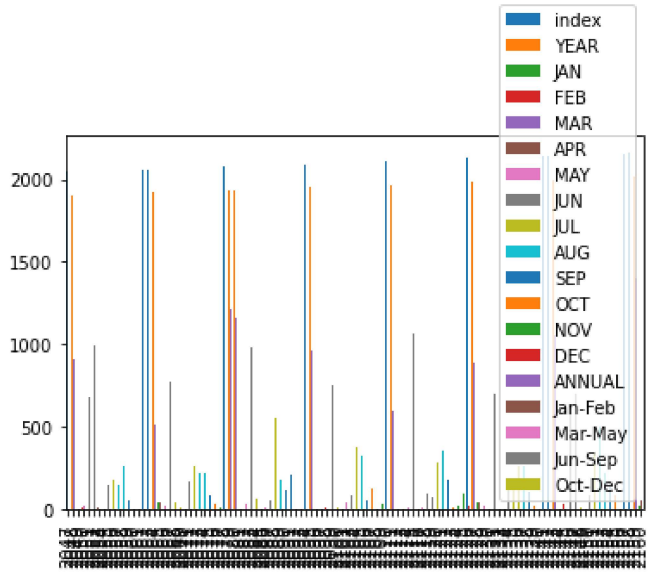
Out[6]: <AxesSubplot:>



Bar chart

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

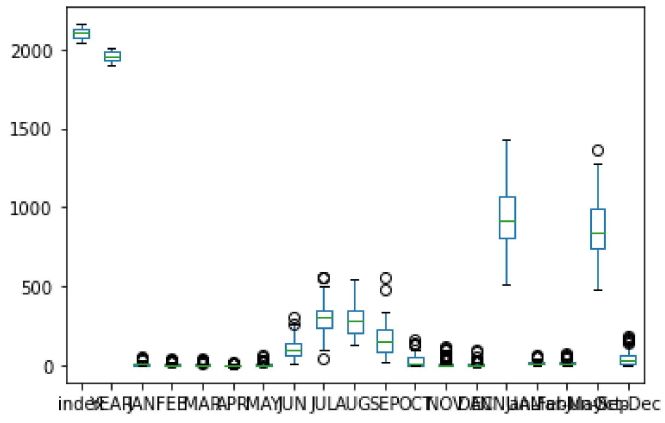
Out[7]: <AxesSubplot:>



Box chart

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

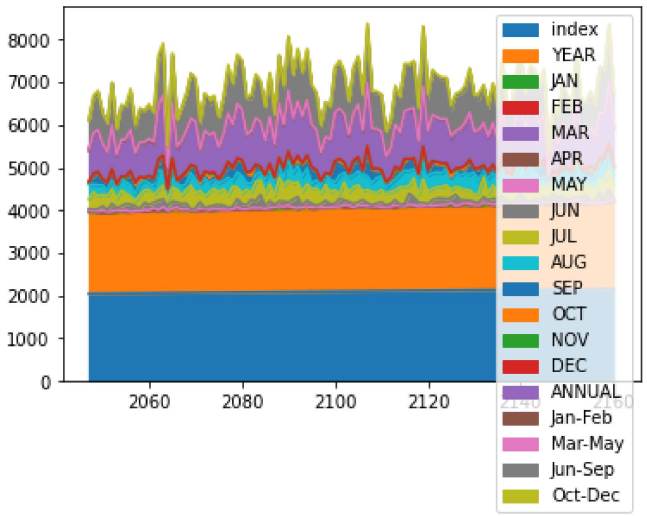
Out[8]: <AxesSubplot:>



Area chart

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

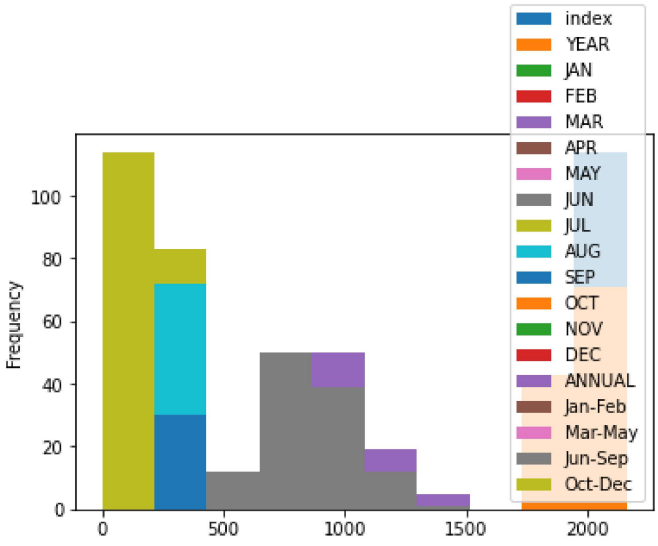
Out[9]: <AxesSubplot:>



Histogram

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

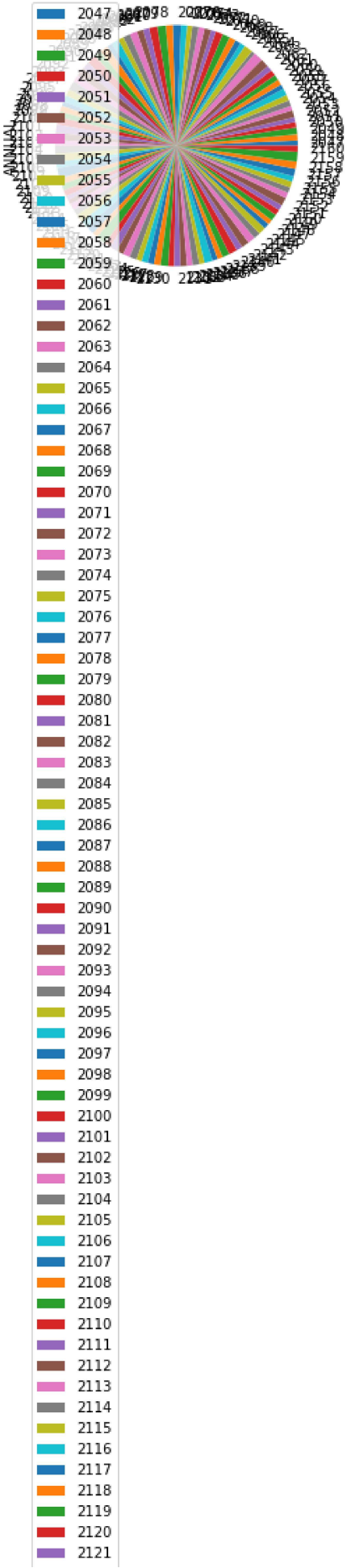
Out[10]: <AxesSubplot:ylabel='Frequency'>



pie chart

In [11]:
```
df.plot.pie(y="ANNUAL")
```

Out[11]:
```
<AxesSubplot:ylabel='ANNUAL'>
```

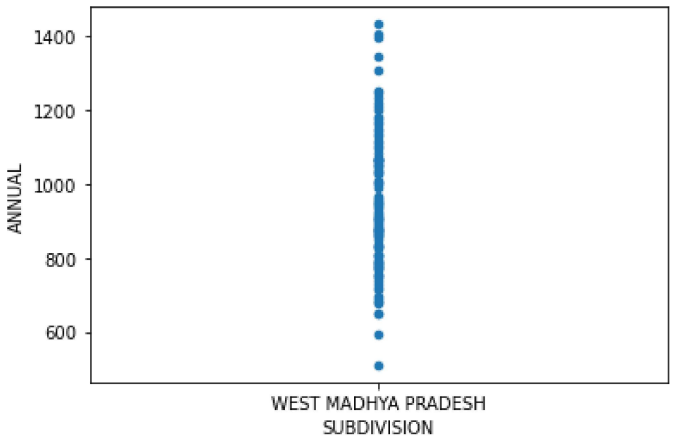

2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121

2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160

Scatter chart

In [12]:
```
df.plot.scatter(y='ANNUAL',x='SUBDIVISION')
```

Out[12]: <AxesSubplot:xlabel='SUBDIVISION', ylabel='ANNUAL'>



In [13]:
```
df.describe()
```

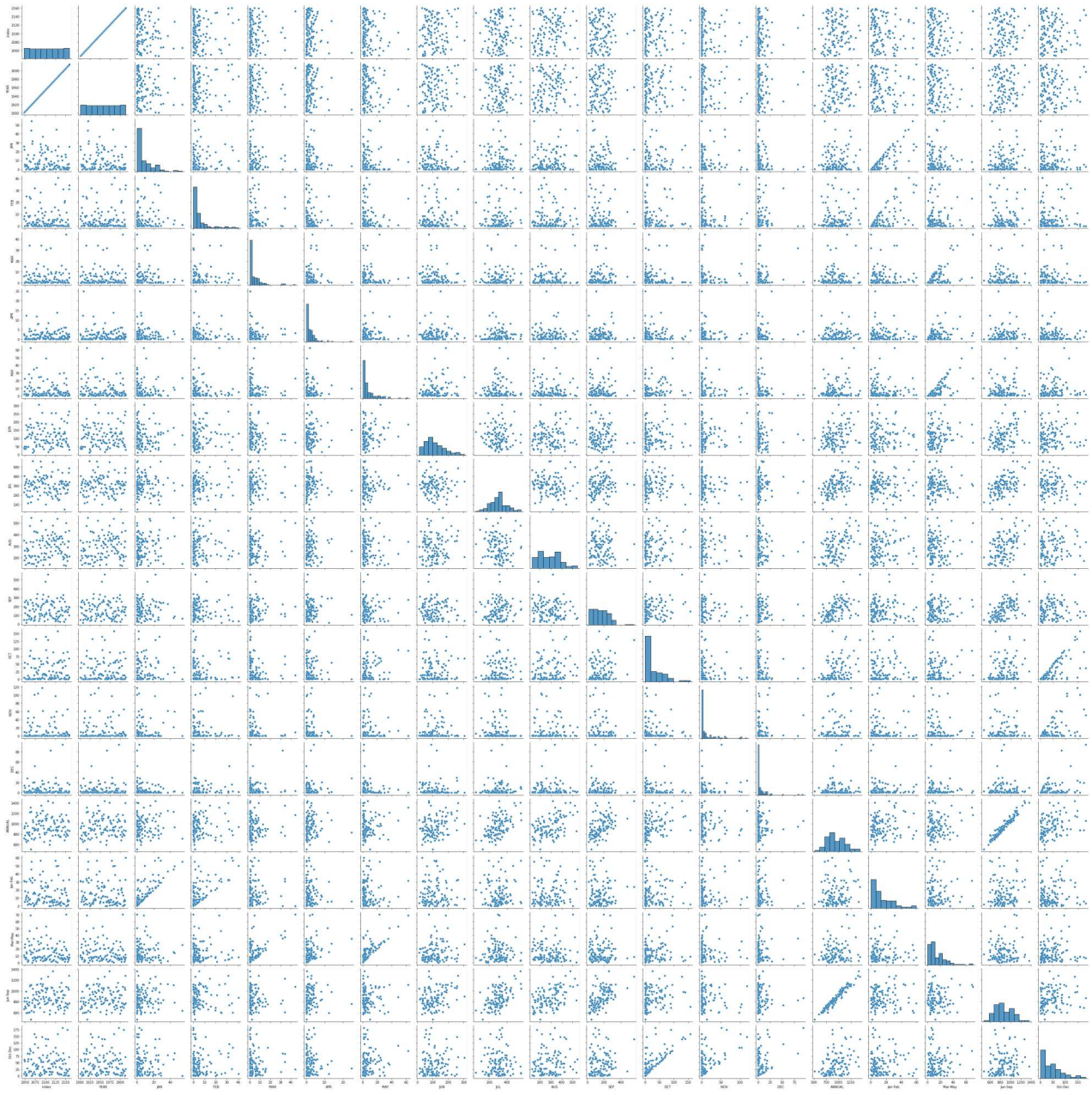
Out[13]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	
count	114.000000	114.000000	114.000000	113.000000	114.000000	114.000000	114.000000	114.000000	114.000000	1
mean	2103.500000	1957.500000	8.970175	6.307080	4.749123	2.279825	7.707018	111.410526	301.884211	2
std	33.052988	33.052988	10.922745	9.033813	7.739317	3.342808	10.433919	60.957935	95.917190	
min	2047.000000	1901.000000	0.000000	0.000000	0.000000	0.000000	0.000000	12.100000	46.900000	1
25%	2075.250000	1929.250000	0.900000	0.500000	0.200000	0.200000	1.325000	64.875000	239.275000	2
50%	2103.500000	1957.500000	4.800000	2.700000	1.850000	1.350000	3.550000	98.250000	305.250000	2
75%	2131.750000	1985.750000	14.450000	8.400000	6.100000	3.000000	10.100000	143.475000	351.075000	3
max	2160.000000	2014.000000	54.100000	40.500000	44.300000	24.800000	62.700000	306.300000	561.600000	5

EDA AND VISUALIZATION

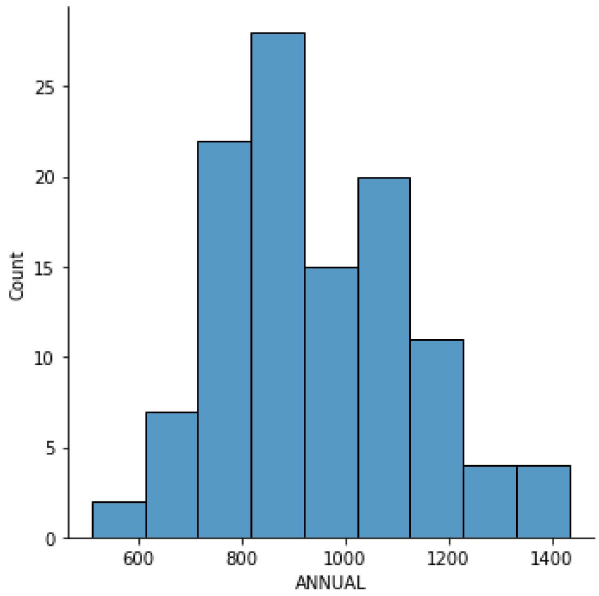
In [14]:
```
sns.pairplot(df)
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x238df8b7fa0>
```



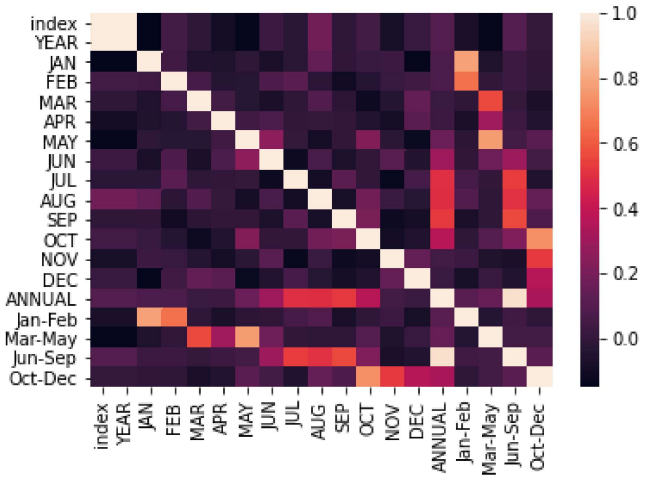
In [15]:
```
sns.displot(df['ANNUAL'])
```

Out[15]: <seaborn.axisgrid.FacetGrid at 0x238ebd6d280>



In [16]:
```
sns.heatmap(df.corr())
```

Out[16]: <AxesSubplot:>



In [ ]:
```

```