

# FINAL ASSESSMENT 2

In [1]:

```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#importing dataset
data=pd.read_csv(r"C:\Users\user\Downloads\rainfall in india 1901-2015.csv")
data
```

Out[2]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	
0	0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	:
1	1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	:
2	2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	:
3	3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	:
4	4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	:
...	...	...	...	...	...	...	...	...	...	...	...	...	
4111	4111	LAKSHADWEEP	2011	5.1	2.8	3.1	85.9	107.2	153.6	350.2	254.0	255.2	
4112	4112	LAKSHADWEEP	2012	19.2	0.1	1.6	76.8	21.2	327.0	231.5	381.2	179.8	:
4113	4113	LAKSHADWEEP	2013	26.2	34.4	37.5	5.3	88.3	426.2	296.4	154.4	180.0	
4114	4114	LAKSHADWEEP	2014	53.2	16.1	4.4	14.9	57.4	244.1	116.1	466.1	132.2	:
4115	4115	LAKSHADWEEP	2015	2.2	0.5	3.7	87.1	133.1	296.6	257.5	146.4	160.4	:

4116 rows × 20 columns

# BIHAR

```
In [7]: df=data.iloc[897:1012]
df
```

Out[7]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
897	897	BIHAR	1901	51.8	19.6	11.9	1.1	65.6	66.3	245.9	319.4	155.1	8.3
898	898	BIHAR	1902	4.6	0.7	24.3	17.3	66.3	118.2	361.0	225.5	358.7	28.5
899	899	BIHAR	1903	5.3	4.7	2.0	4.7	28.2	192.9	115.0	342.6	173.9	147.0
900	900	BIHAR	1904	6.3	1.7	3.5	5.3	118.7	191.6	394.4	351.3	84.4	98.1
901	901	BIHAR	1905	16.0	30.1	32.6	21.4	77.5	50.5	409.1	495.3	353.9	11.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1007	1007	BIHAR	2011	4.2	7.7	9.2	23.9	74.5	211.0	241.1	278.7	234.1	10.0
1008	1008	BIHAR	2012	18.1	2.7	7.3	20.4	18.8	96.2	354.0	240.4	233.8	34.3
1009	1009	BIHAR	2013	5.1	22.6	0.6	32.3	89.5	183.3	182.0	213.6	143.3	197.1
1010	1010	BIHAR	2014	17.0	33.5	8.4	0.7	103.9	115.2	265.4	307.6	160.3	47.8
1011	1011	BIHAR	2015	12.8	1.8	27.2	38.7	39.5	122.1	231.5	287.0	101.7	10.4

115 rows × 20 columns

# Data Cleaning and Preprocessing

```
In [23]: df.head()
```

Out[23]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
897	897	BIHAR	1901	51.8	19.6	11.9	1.1	65.6	66.3	245.9	319.4	155.1	8.3
898	898	BIHAR	1902	4.6	0.7	24.3	17.3	66.3	118.2	361.0	225.5	358.7	28.5
899	899	BIHAR	1903	5.3	4.7	2.0	4.7	28.2	192.9	115.0	342.6	173.9	147.0
900	900	BIHAR	1904	6.3	1.7	3.5	5.3	118.7	191.6	394.4	351.3	84.4	98.1
901	901	BIHAR	1905	16.0	30.1	32.6	21.4	77.5	50.5	409.1	495.3	353.9	11.6

In [24]: `df.tail()`

Out[24]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
<b>1007</b>	1007	BIHAR	2011	4.2	7.7	9.2	23.9	74.5	211.0	241.1	278.7	234.1	10.0
<b>1008</b>	1008	BIHAR	2012	18.1	2.7	7.3	20.4	18.8	96.2	354.0	240.4	233.8	34.5
<b>1009</b>	1009	BIHAR	2013	5.1	22.6	0.6	32.3	89.5	183.3	182.0	213.6	143.3	197.1
<b>1010</b>	1010	BIHAR	2014	17.0	33.5	8.4	0.7	103.9	115.2	265.4	307.6	160.3	47.8
<b>1011</b>	1011	BIHAR	2015	12.8	1.8	27.2	38.7	39.5	122.1	231.5	287.0	101.7	10.4

In [25]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 897 to 1011
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: 18.1+ KB
```

```
In [26]: #filling null values
df1=df.fillna(0)
df1
```

Out[26]:

	index	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT
897	897	BIHAR	1901	51.8	19.6	11.9	1.1	65.6	66.3	245.9	319.4	155.1	8.5
898	898	BIHAR	1902	4.6	0.7	24.3	17.3	66.3	118.2	361.0	225.5	358.7	28.5
899	899	BIHAR	1903	5.3	4.7	2.0	4.7	28.2	192.9	115.0	342.6	173.9	147.0
900	900	BIHAR	1904	6.3	1.7	3.5	5.3	118.7	191.6	394.4	351.3	84.4	98.1
901	901	BIHAR	1905	16.0	30.1	32.6	21.4	77.5	50.5	409.1	495.3	353.9	11.6
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1007	1007	BIHAR	2011	4.2	7.7	9.2	23.9	74.5	211.0	241.1	278.7	234.1	10.0
1008	1008	BIHAR	2012	18.1	2.7	7.3	20.4	18.8	96.2	354.0	240.4	233.8	34.5
1009	1009	BIHAR	2013	5.1	22.6	0.6	32.3	89.5	183.3	182.0	213.6	143.3	197.1
1010	1010	BIHAR	2014	17.0	33.5	8.4	0.7	103.9	115.2	265.4	307.6	160.3	47.8
1011	1011	BIHAR	2015	12.8	1.8	27.2	38.7	39.5	122.1	231.5	287.0	101.7	10.4

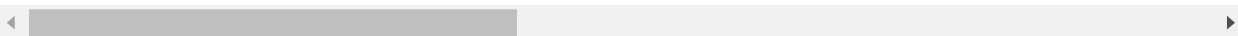
115 rows × 20 columns



```
In [27]: df1.describe()
```

Out[27]:

	index	YEAR	JAN	FEB	MAR	APR	MAY	
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.0
mean	954.000000	1958.000000	13.386087	14.393913	10.124348	16.918261	53.081739	174.3
std	33.341666	33.341666	14.791960	15.075036	11.695340	15.978278	27.941714	76.1
min	897.000000	1901.000000	0.000000	0.000000	0.000000	0.100000	1.300000	48.1
25%	925.500000	1929.500000	2.350000	2.750000	1.800000	5.250000	31.550000	117.1
50%	954.000000	1958.000000	9.400000	8.400000	6.500000	12.600000	46.200000	165.5
75%	982.500000	1986.500000	18.700000	21.400000	12.850000	24.500000	76.200000	211.0
max	1011.000000	2015.000000	81.200000	66.300000	65.500000	91.400000	118.700000	446.0



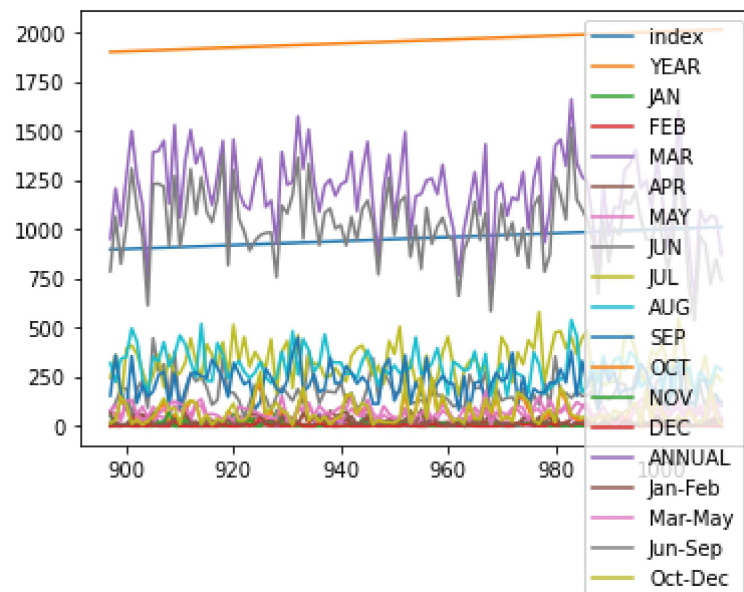
```
In [28]: df1.columns
```

```
Out[28]: 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')
```

# Data Visulaization

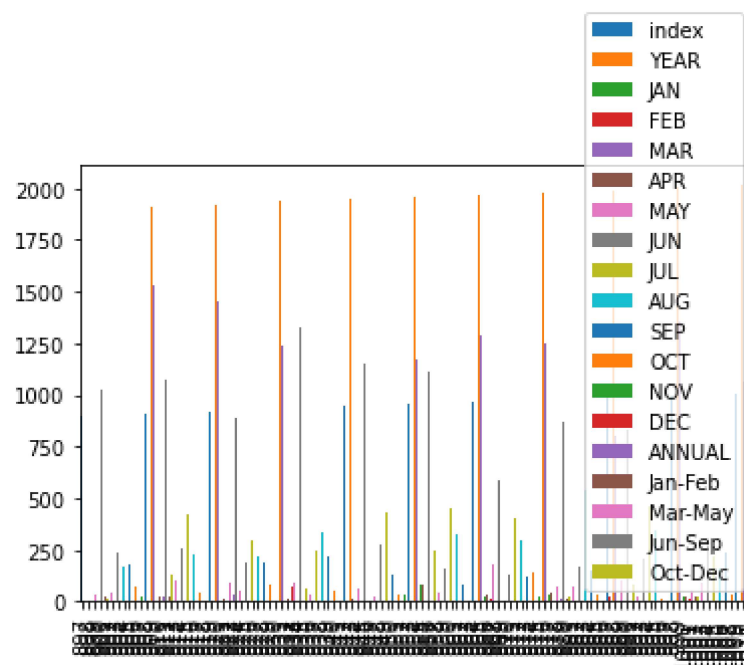
In [29]: `df1.plot.line()`

Out[29]: `<AxesSubplot:>`



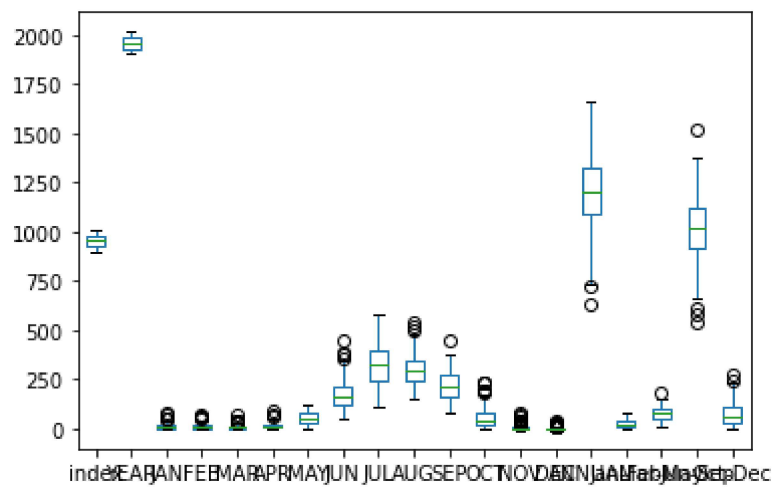
In [30]: `df1.plot.bar()`

Out[30]: `<AxesSubplot:>`



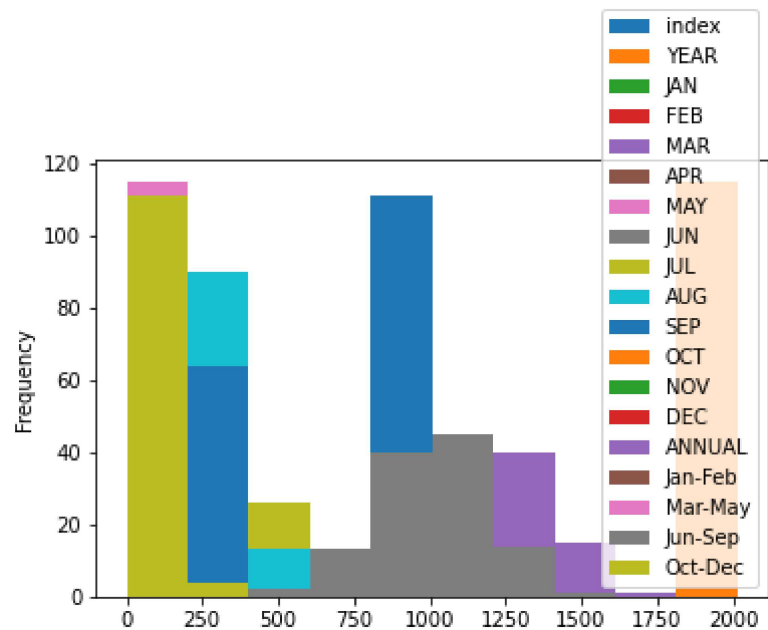
```
In [31]: df1.plot.box()
```

```
Out[31]: <AxesSubplot:>
```



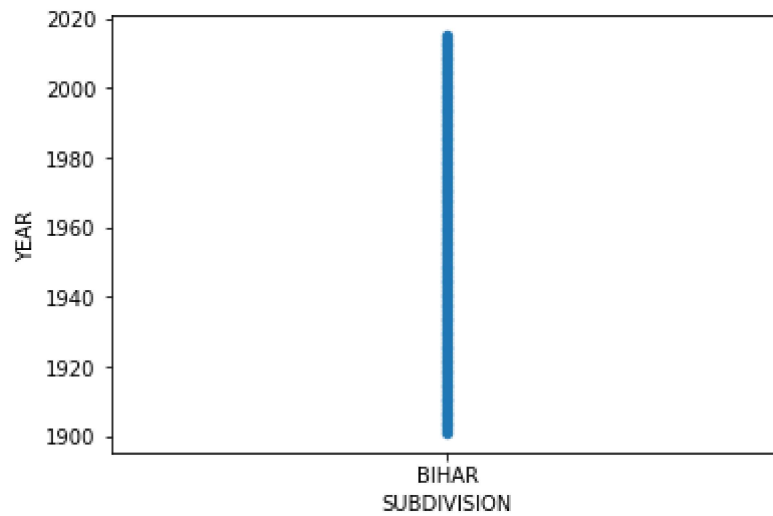
```
In [32]: df1.plot.hist()
```

```
Out[32]: <AxesSubplot:ylabel='Frequency'>
```



```
In [33]: df1.plot.scatter(x="SUBDIVISION",y="YEAR")
```

```
Out[33]: <AxesSubplot:xlabel='SUBDIVISION', ylabel='YEAR'>
```



```
In [34]: df2=df1[[ 'Jun-Sep']]
df2.plot.pie(subplots=True)
```

```
Out[34]: array([<AxesSubplot:ylabel='Jun-Sep'>], dtype=object)
```

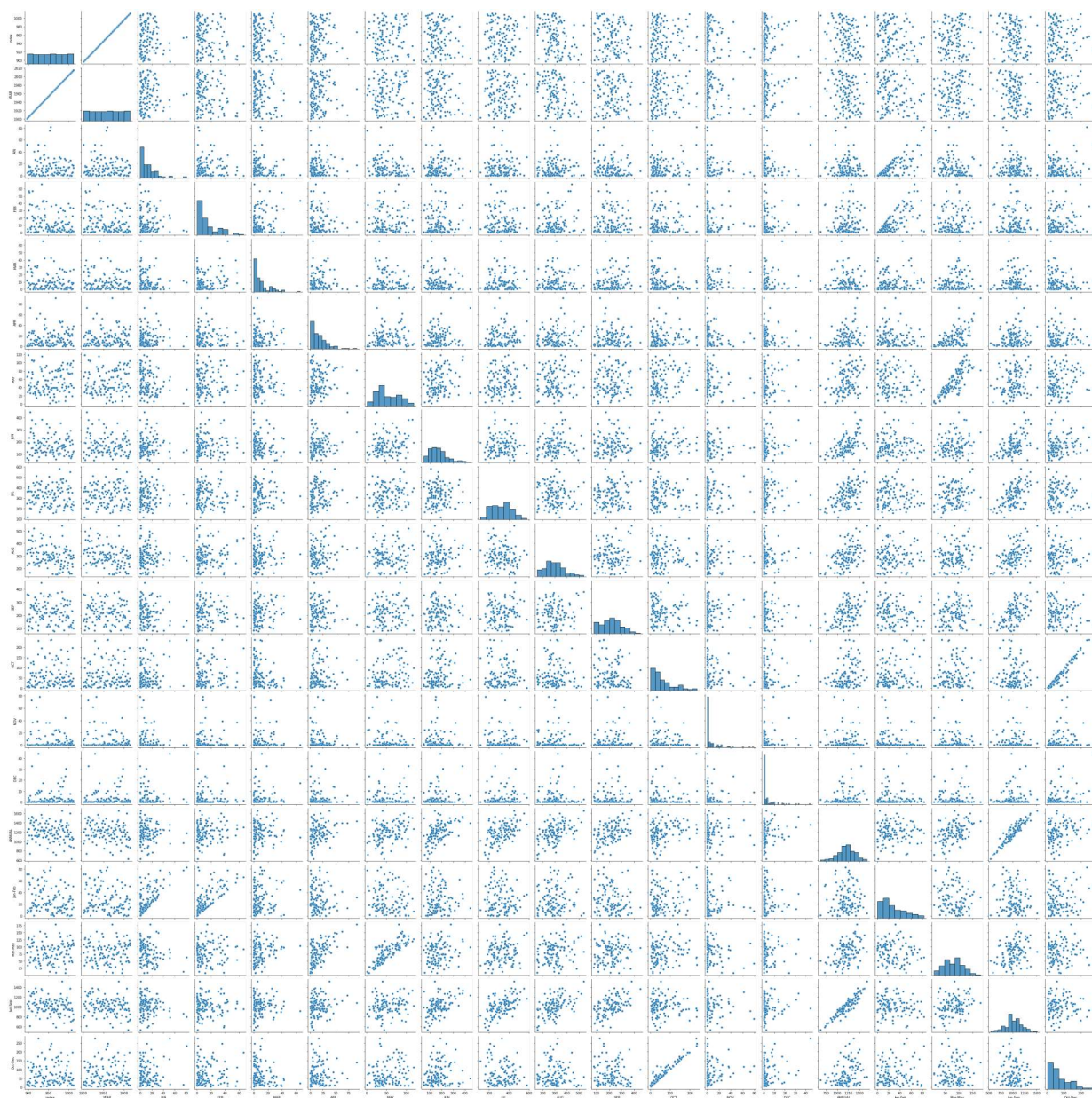




	955
	956
	957
	958
	959
	960
	961
	962
	963
	964
	965
	966
	967
	968
	969
	970
	971
	972
	973
	974
	975
	976
	977
	978
	979
	980
	981
	982
	983
	984
	985
	986
	987
	988
	989
	990
	991
	992
	993
	994
	995
	996
	997
	998
	999
	1000
	1001
	1002
	1003
	1004
	1005
	1006
	1007
	1008
	1009
	1010
	1011

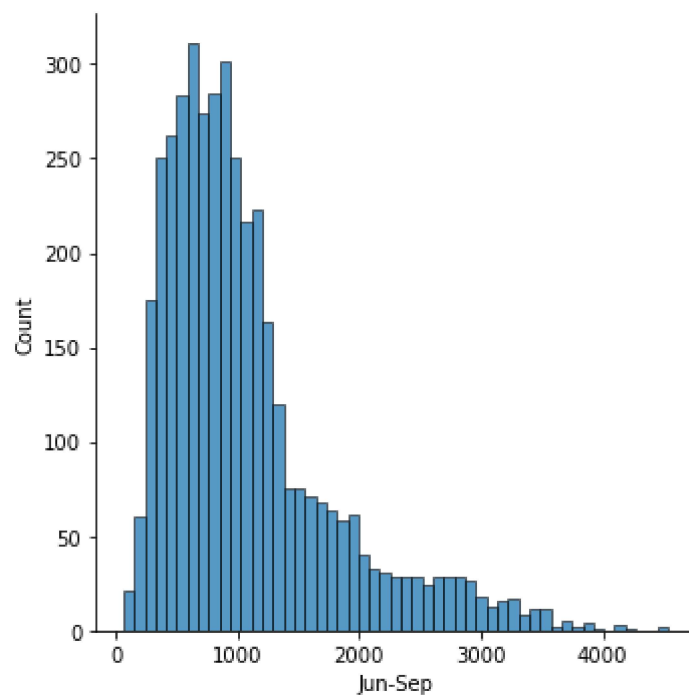
```
In [35]: sns.pairplot(df1)
```

```
Out[35]: <seaborn.axisgrid.PairGrid at 0x229f5e4bb20>
```



```
In [36]: sns.displot(data["Jun-Sep"])
```

```
Out[36]: <seaborn.axisgrid.FacetGrid at 0x22984c82be0>
```



```
In [22]: sns.heatmap(df1.corr())
```

```
Out[22]: <AxesSubplot:>
```

