In [76]:
```
# import libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [177]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\\\Fitness.csv")
```

Out[177]:

	SALESMAN	JAN	FEB	MAR	APR	MAY	JUN	TOTAL SALES	Unnamed: 8	Unnamed: 9	Unname
0	ANU	70.0	80.0	75.0	60.0	72.0	55.0	412.0	NaN	NaN	N
1	BABU	30.0	48.0	35.0	45.0	25.0	37.0	220.0	NaN	NaN	N
2	CHANDRU	65.0	54.0	49.0	54.0	35.0	65.0	322.0	NaN	NaN	N
3	DAVID	85.0	71.0	68.0	77.0	88.0	73.0	462.0	NaN	NaN	N
4	EINSTEIN	55.0	25.0	45.0	50.0	53.0	30.0	258.0	NaN	NaN	N
5	FAROOK	35.0	45.0	15.0	45.0	45.0	25.0	210.0	NaN	NaN	N
6	GOWTHAM	75.0	66.0	59.0	65.0	56.0	30.0	351.0	NaN	NaN	N
7	HARSHITH	29.0	35.0	49.0	48.0	35.0	55.0	247.0	NaN	NaN	N
8	INIYAN	35.0	35.0	50.0	59.0	67.0	73.0	319.0	NaN	NaN	N
9	JOHN	77.0	85.0	77.0	68.0	56.0	25.0	388.0	NaN	NaN	N
10	MONTHLY SALES	556.0	544.0	522.0	571.0	532.0	468.0	NaN	3193.0	NaN	N
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3189.0	NaN	NaN	N

In [178]:
```
x=x.head(10)
```

Out[178]:

	SALESMAN	JAN	FEB	MAR	APR	MAY	JUN	TOTAL SALES	Unnamed: 8	Unnamed: 9	Unnamed: 10	U
0	ANU	70.0	80.0	75.0	60.0	72.0	55.0	412.0	NaN	NaN	NaN	
1	BABU	30.0	48.0	35.0	45.0	25.0	37.0	220.0	NaN	NaN	NaN	
2	CHANDRU	65.0	54.0	49.0	54.0	35.0	65.0	322.0	NaN	NaN	NaN	2 tr c
3	DAVID	85.0	71.0	68.0	77.0	88.0	73.0	462.0	NaN	NaN	NaN	3 us pe
4	EINSTEIN	55.0	25.0	45.0	50.0	53.0	30.0	258.0	NaN	NaN	NaN	
5	FAROOK	35.0	45.0	15.0	45.0	45.0	25.0	210.0	NaN	NaN	NaN	5 re
6	GOWTHAM	75.0	66.0	59.0	65.0	56.0	30.0	351.0	NaN	NaN	NaN	
7	HARSHITH	29.0	35.0	49.0	48.0	35.0	55.0	247.0	NaN	NaN	NaN	
8	INIYAN	35.0	35.0	50.0	59.0	67.0	73.0	319.0	NaN	NaN	NaN	
9	JOHN	77.0	85.0	77.0	68.0	56.0	25.0	388.0	NaN	NaN	NaN	

In [179]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   SALESMAN     10 non-null     object 
 1   JAN          10 non-null     float64
 2   FEB          10 non-null     float64
 3   MAR          10 non-null     float64
 4   APR          10 non-null     float64
 5   MAY          10 non-null     float64
 6   JUN          10 non-null     float64
 7   TOTAL SALES  10 non-null     float64
 8   Unnamed: 8   0 non-null      float64
 9   Unnamed: 9   0 non-null      float64
 10  Unnamed: 10  0 non-null      float64
 11  Unnamed: 11  6 non-null      object 
dtypes: float64(10), object(2)
memory usage: 1.1+ KB
```

In [180]:

Out[180]:

```
Index(['SALESMAN', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'TOTAL SALES',
       'Unnamed: 8', 'Unnamed: 9', 'Unnamed: 10', 'Unnamed: 11'],
      dtype='object')
```

In [181]:

```
d=x[['SALESMAN', 'JAN', 'FEB', 'MAR', 'APR', 'MAY']]
```

Out[181]:

	SALESMAN	JAN	FEB	MAR	APR	MAY
0	ANU	70.0	80.0	75.0	60.0	72.0
1	BABU	30.0	48.0	35.0	45.0	25.0
2	CHANDRU	65.0	54.0	49.0	54.0	35.0
3	DAVID	85.0	71.0	68.0	77.0	88.0
4	EINSTEIN	55.0	25.0	45.0	50.0	53.0
5	FAROOK	35.0	45.0	15.0	45.0	45.0
6	GOWTHAM	75.0	66.0	59.0	65.0	56.0
7	HARSHITH	29.0	35.0	49.0	48.0	35.0
8	INIYAN	35.0	35.0	50.0	59.0	67.0
9	JOHN	77.0	85.0	77.0	68.0	56.0

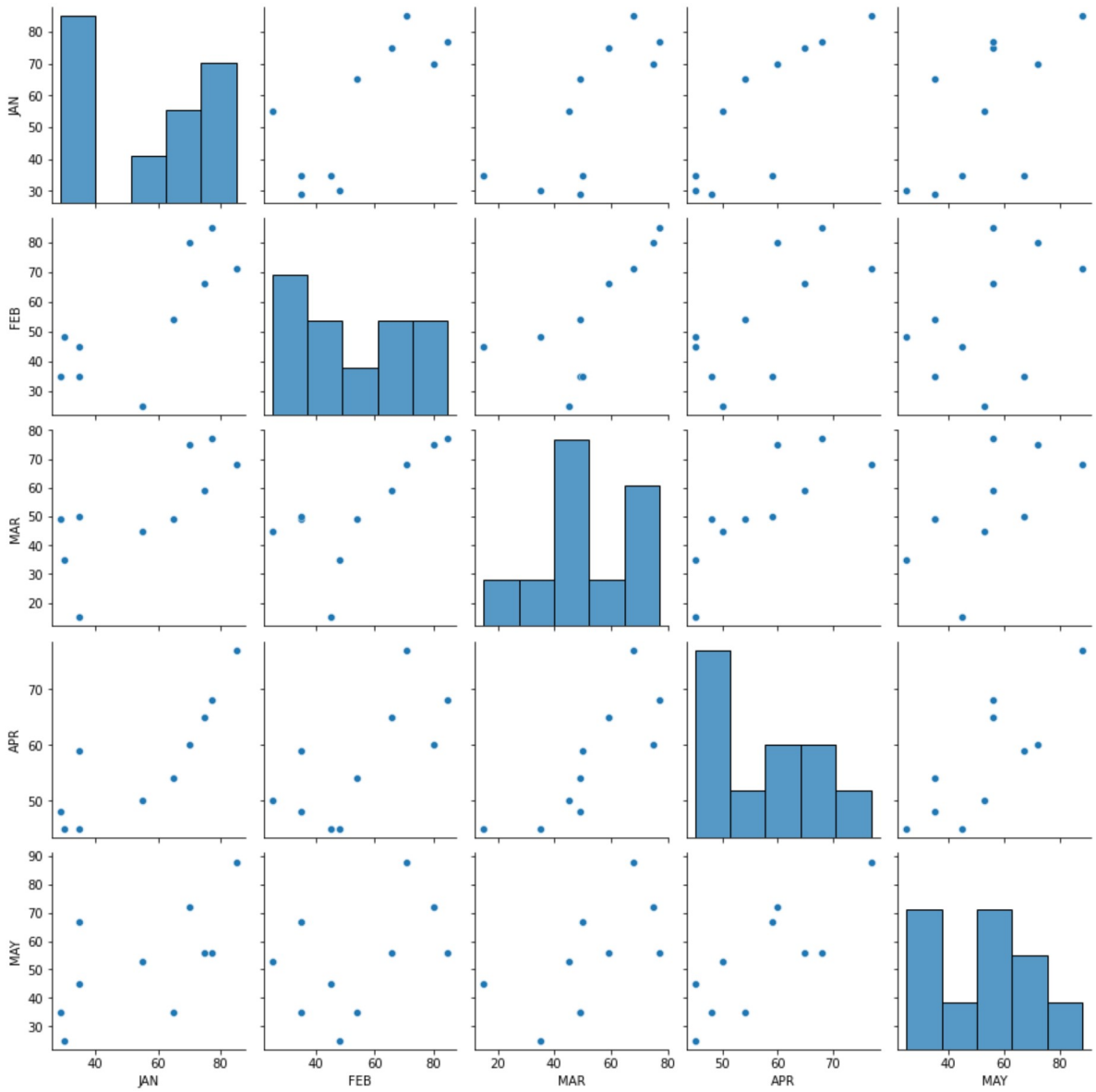
In [182]:

Out[182]:

	JAN	FEB	MAR	APR	MAY	JUN	TOTAL SALES	Unnamed: 8
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	0.0
mean	55.600000	54.400000	52.200000	57.100000	53.200000	46.800000	318.900000	NaN
std	21.618922	20.408059	18.819612	10.671353	19.135772	19.577765	85.296151	NaN
min	29.000000	25.000000	15.000000	45.000000	25.000000	25.000000	210.000000	NaN
25%	35.000000	37.500000	46.000000	48.500000	37.500000	30.000000	249.750000	NaN
50%	60.000000	51.000000	49.500000	56.500000	54.500000	46.000000	320.500000	NaN
75%	73.750000	69.750000	65.750000	63.750000	64.250000	62.500000	378.750000	NaN
max	85.000000	85.000000	77.000000	77.000000	88.000000	73.000000	462.000000	NaN

In [183]:

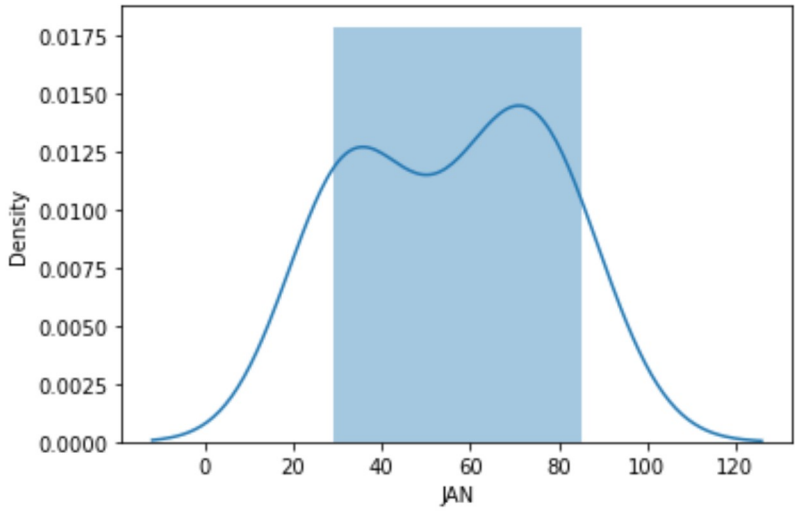
Out[183]: <seaborn.axisgrid.PairGrid at 0x190c77b4c40>



In [184]:

```
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)
```

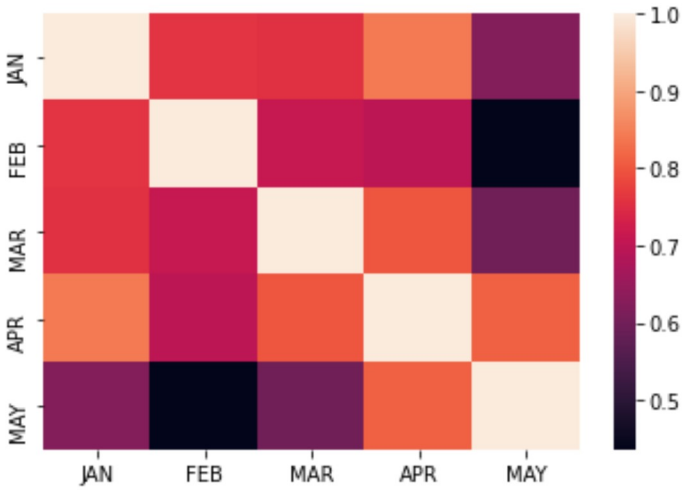
Out[184]: <AxesSubplot:xlabel='JAN', ylabel='Density'>



In [185]:

In [186]:

Out[186]: <AxesSubplot:>



In [188]:

```
x=x1[['JAN']]
```

In [189]:
```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [190]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[190]:
```
LinearRegression()
```

In [191]:

```
0.0
```

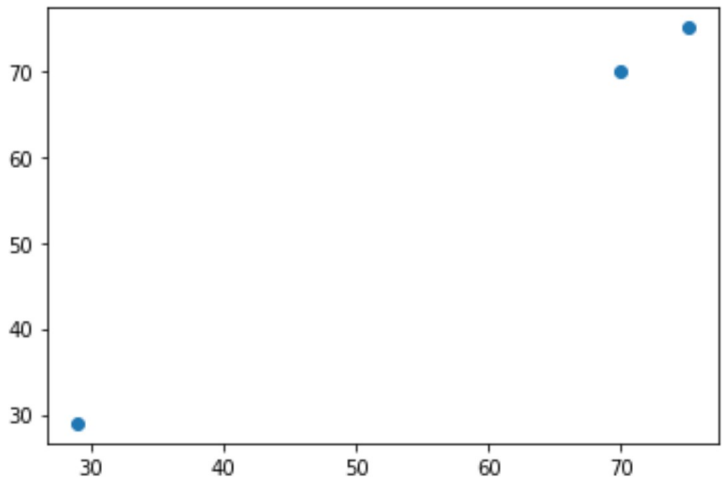
In [192]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[192]:

	Co-efficient
JAN	1.0

In [193]:
```
prediction=lr.predict(x_test)
```

Out[193]:
```
<matplotlib.collections.PathCollection at 0x190c87e5bb0>
```



In [194]:

Out[194]:
```
1.0
```

In [195]:

Out[195]:
```
1.0
```

In [196]:

In [197]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[197]: 0.9999879281811936

In [198]:
```
la=Lasso(alpha=10)
```

Out[198]: Lasso(alpha=10)

In [199]:

Out[199]: 0.9994044052629513

In [200]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
```

Out[200]: ElasticNet()

In [201]:

Out[201]: array([0.99759551])

In [202]:

Out[202]: array([29.06148637, 69.96290208, 74.9508796])

In [203]:

Out[203]: 0.13121672350762736

In [204]:

Out[204]: 0.999994058365066

In [205]:

In [206]:

Mean Absolute Error 0.0

In [207]:

Mean Squared Error 0.0

In [208]:

Root Mean Squared Error 0.0

In [ ]: