In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [176]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\12_mobile_prices_2023.csv")
a
```

Out[176]:

	Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR	S
0	POCO C50 (Royal Blue, 32 GB)	4.2	33,561	2 GB RAM	32 GB ROM	8MP Dual Camera	5MP Front Camera	5000 mAh	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	₹5,649	
1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77,128	4 GB RAM	64 GB ROM	50MP + 2MP	8MP Front Camera	5000 mAh	Mediatek Dimensity 700 Processor	₹11,999	
2	POCO C51 (Royal Blue, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999	

In [177]:
```
a=a.head(50)
a
```

Out[177]:

	Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR
0	POCO C50 (Royal Blue, 32 GB)	4.2	33,561	2 GB RAM	32 GB ROM	8MP Dual Camera	5MP Front Camera	5000 mAh	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	₹5,649
1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77,128	4 GB RAM	64 GB ROM	50MP + 2MP	8MP Front Camera	5000 mAh	Mediatek Dimensity 700 Processor	₹11,999
2	POCO C51 (Royal Blue, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999
3	POCO C55 (Cool Blue	4.2	22,621	4 GB	64 GB ROM	50MP Dual Rear	5MP Front	5000	Mediatek Helio G85	₹7,749

In [178]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Phone Name         50 non-null     object
 1   Rating ?/5         50 non-null     float64
 2   Number of Ratings  50 non-null     object
 3   RAM                50 non-null     object
 4   ROM/Storage        50 non-null     object
 5   Back/Rare Camera   50 non-null     object
 6   Front Camera       50 non-null     object
 7   Battery            50 non-null     object
 8   Processor          50 non-null     object
 9   Price in INR       50 non-null     object
 10  Date of Scraping   50 non-null     object
dtypes: float64(1), object(10)
memory usage: 4.4+ KB
```

In [179]:
```
a.columns
```

Out[179]:
```
Index(['Phone Name', 'Rating ?/5', 'Number of Ratings', 'RAM', 'ROM/Storage',
       'Back/Rare Camera', 'Front Camera', 'Battery', 'Processor',
       'Price in INR', 'Date of Scraping'],
      dtype='object')
```

In [180]:
```
a.head()
```

Out[180]:

	Phone Name	Rating ?/5	Number of Ratings	RAM	ROM/Storage	Back/Rare Camera	Front Camera	Battery	Processor	Price in INR	Date of Scraping
0	POCO C50 (Royal Blue, 32 GB)	4.2	33,561	2 GB RAM	32 GB ROM	8MP Dual Camera	5MP Front Camera	5000 mAh	Mediatek Helio A22 Processor, Upto 2.0 GHz Pro...	₹5,649	2023-06-17
1	POCO M4 5G (Cool Blue, 64 GB)	4.2	77,128	4 GB RAM	64 GB ROM	50MP + 2MP	8MP Front Camera	5000 mAh	Mediatek Dimensity 700 Processor	₹11,999	2023-06-17
2	POCO C51 (Royal Blue, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999	2023-06-17
3	POCO C55 (Cool Blue, 64 GB)	4.2	22,621	4 GB RAM	64 GB ROM	50MP Dual Rear Camera	5MP Front Camera	5000 mAh	Mediatek Helio G85 Processor	₹7,749	2023-06-17
4	POCO C51 (Power Black, 64 GB)	4.3	15,175	4 GB RAM	64 GB ROM	8MP Dual Rear Camera	5MP Front Camera	5000 mAh	Helio G36 Processor	₹6,999	2023-06-17

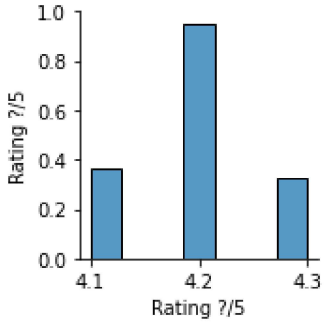
In [181]:
```
a.describe()
```

Out[181]:

	Rating ?/5
count	50.000000
mean	4.198000
std	0.065434
min	4.100000
25%	4.200000
50%	4.200000
75%	4.200000
max	4.300000

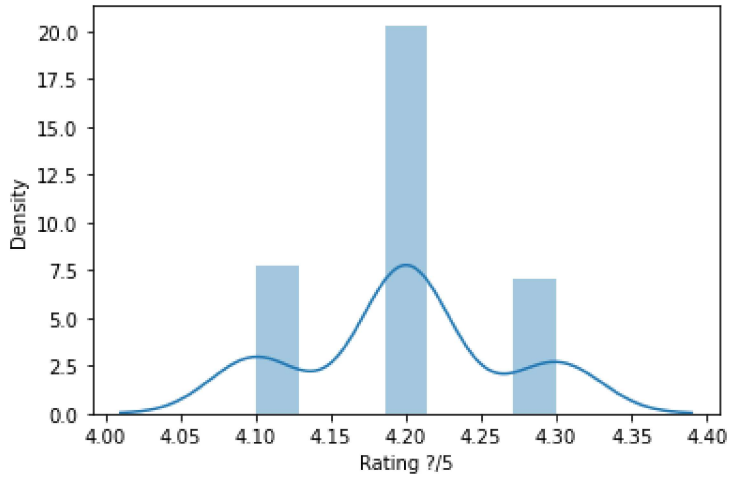
In [182]:
```
sns.pairplot(a)
```

Out[182]:
```
<seaborn.axisgrid.PairGrid at 0x22eb3bbda00>
```



In [186]:
```
sns.distplot(a['Rating ?/5'])
```

Out[186]:
```
<AxesSubplot:xlabel='Rating ?/5', ylabel='Density'>
```



In [188]:
```
x1=a[['Rating ?/5']]
```

In [189]:
```
sns.heatmap(x1.corr())
```

Out[189]: <AxesSubplot:>

In [191]:
```
x=a[['Rating ?/5']]
y=a['Rating ?/5']
```

In [192]:
```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [193]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[193]: LinearRegression()

In [194]:
```
print(lr.intercept_)
```

```
1.7763568394002505e-15
```

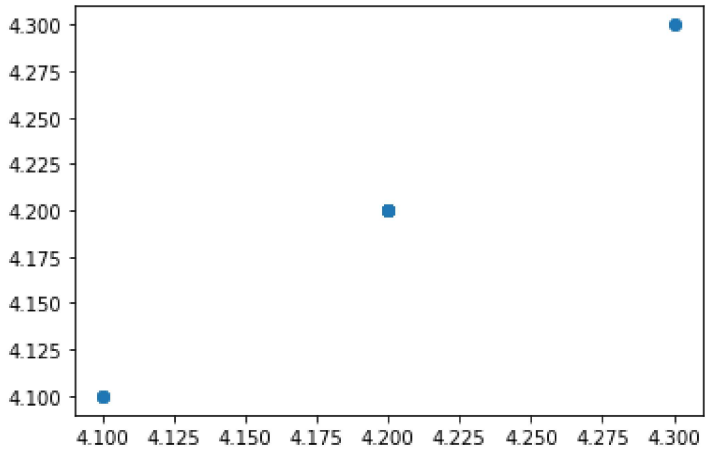
In [195]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[195]:

	Co-efficient
Rating ?/5	1.0

In [196]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[196]:
```
<matplotlib.collections.PathCollection at 0x22eb3f30fa0>
```



In [197]:
```
print(lr.score(x_test,y_test))
```

```
1.0
```

In [198]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [199]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[199]:
```
Ridge(alpha=10)
```

In [200]:
```
rr.score(x_test,y_test)
```

Out[200]:
```
-0.03728902196791495
```

In [201]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[201]:
```
Lasso(alpha=10)
```

In [202]:
```
la.score(x_test,y_test)
```

Out[202]:
```
-0.0685809207403889
```

In [203]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[203]:
```
ElasticNet()
```

In [204]:
```
print(en.coef_)
```

```
[0.]
```

In [205]:
```
print(en.intercept_)
```

```
4.202857142857143
```

In [206]:
```
print(en.predict(x_test))
```

```
[4.20285714 4.20285714 4.20285714 4.20285714 4.20285714 4.20285714
 4.20285714 4.20285714 4.20285714 4.20285714 4.20285714 4.20285714
 4.20285714 4.20285714 4.20285714]
```

In [207]:
```
en.score(x_test,y_test)
```

Out[207]:
```
-0.0685809207403889
```

In [208]:
```
from sklearn import metrics
```

In [212]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 0.0
```

In [213]:
```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 0.0
```

In [214]:
```
print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
 Root Mean Squared Error 0.0
```