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

In [242]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\11_winequality-red - 11_winequality-re
```

Out[242]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcc
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	
...	...	...	...	...	...	...	...	...	...	...	
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	

1599 rows × 12 columns

In [243]:
```
x=x.head(10)
```

Out[243]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5
9	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5

In [244]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         10 non-null     float64
 1   volatile acidity      10 non-null     float64
 2   citric acid           10 non-null     float64
 3   residual sugar        10 non-null     float64
 4   chlorides             10 non-null     float64
 5   free sulfur dioxide   10 non-null     float64
 6   total sulfur dioxide  10 non-null     float64
 7   density               10 non-null     float64
 8   pH                    10 non-null     float64
 9   sulphates             10 non-null     float64
 10  alcohol               10 non-null     float64
 11  quality               10 non-null     int64  
dtypes: float64(11), int64(1)
memory usage: 1.1 KB
```

In [245]:

Out[245]:

```
Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol', 'quality'],
      dtype='object')
```

In [246]:

```
d=x[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar']]
```

Out[246]:

	fixed acidity	volatile acidity	citric acid	residual sugar
0	7.4	0.70	0.00	1.9
1	7.8	0.88	0.00	2.6
2	7.8	0.76	0.04	2.3
3	11.2	0.28	0.56	1.9
4	7.4	0.70	0.00	1.9
5	7.4	0.66	0.00	1.8
6	7.9	0.60	0.06	1.6
7	7.3	0.65	0.00	1.2
8	7.8	0.58	0.02	2.0
9	7.5	0.50	0.36	6.1

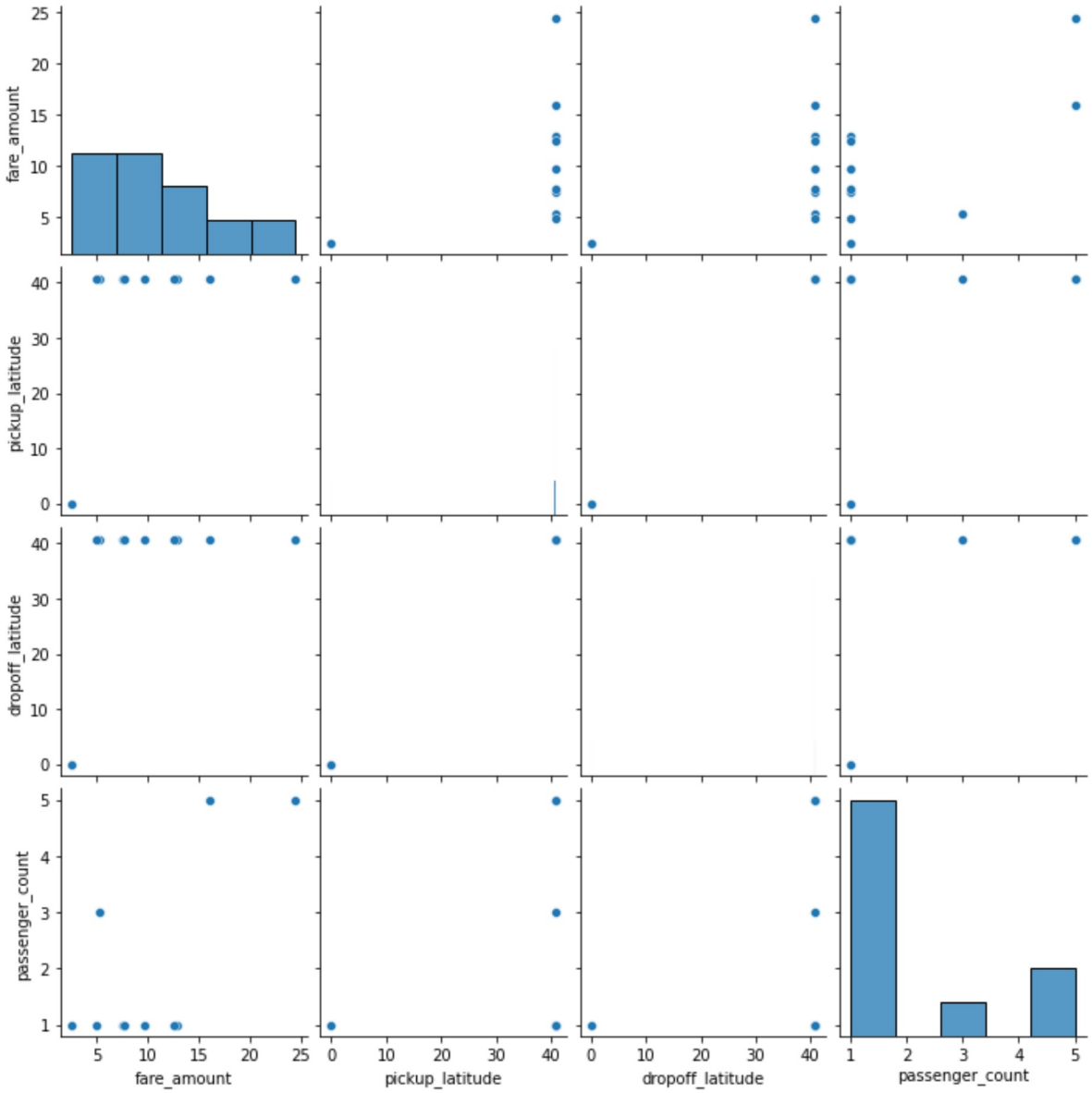
In [214]:

Out[214]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_l
count	1.000000e+01	10.000000	10.000000	10.000000	10.000000	10.
mean	3.443881e+07	10.350000	-66.580708	36.667971	-66.570116	36.
std	1.342943e+07	6.460693	23.394088	12.883834	23.390384	12.
min	1.582227e+07	2.500000	-74.005043	0.000000	-74.002720	0.
25%	2.465233e+07	5.850000	-73.998451	40.730757	-73.989303	40.
50%	3.601534e+07	8.700000	-73.975656	40.741278	-73.967168	40.
75%	4.485598e+07	12.800000	-73.963340	40.745346	-73.962684	40.
max	5.061106e+07	24.500000	0.000000	40.790844	0.000000	40.

In [215]:

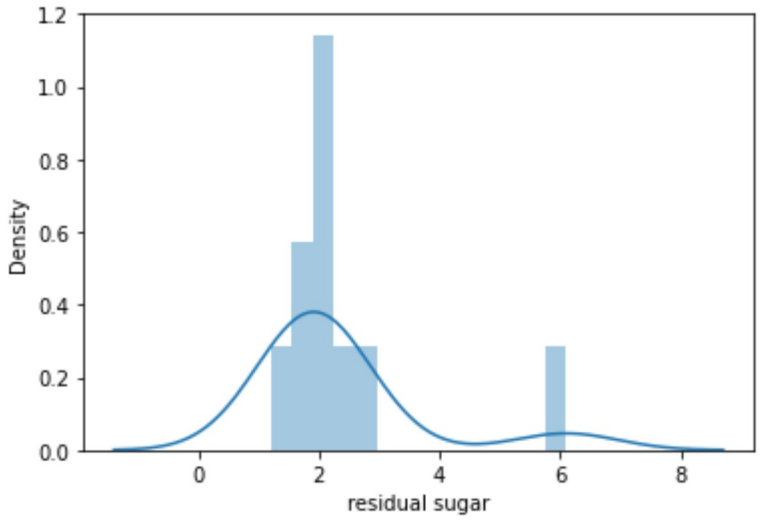
Out[215]: <seaborn.axisgrid.PairGrid at 0x190c880c670>



In [247]:

```
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[247]: <AxesSubplot:xlabel='residual sugar', ylabel='Density'>



In [249]:

In [250]:

Out[250]: <AxesSubplot:>

In [251]:

```
x=x1[['citric acid']]
```

In [252]:
```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [253]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[253]:
```
LinearRegression()
```

In [254]:

```
-5.551115123125783e-17
```

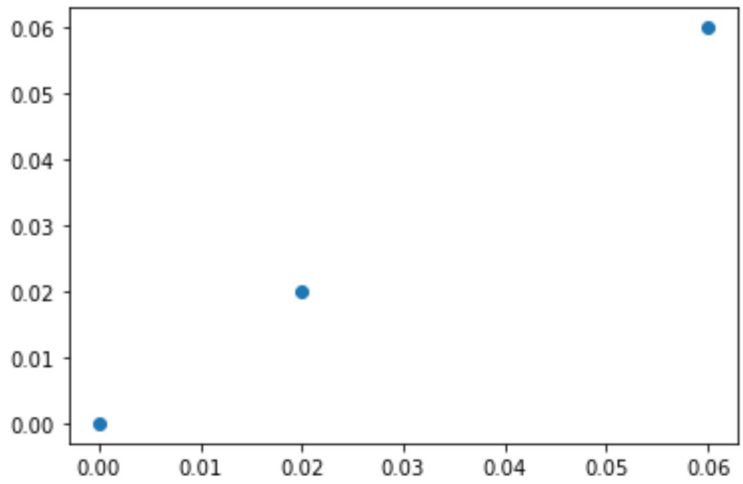
In [255]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[255]:

	Co-efficient
citric acid	1.0

In [256]:
```
prediction=lr.predict(x_test)
```

Out[256]:
```
<matplotlib.collections.PathCollection at 0x190cede38b0>
```



In [257]:

Out[257]:
```
1.0
```

In [258]:

Out[258]:
```
1.0
```

In [259]:

In [260]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[260]: -18.382270484550485

In [261]:
```
la=Lasso(alpha=10)
```

Out[261]: Lasso(alpha=10)

In [262]:

Out[262]: -19.61516034985423

In [263]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
```

Out[263]: ElasticNet()

In [264]:

Out[264]: array([0.])

In [265]:

Out[265]: array([0.13714286, 0.13714286, 0.13714286])

In [266]:

Out[266]: 0.13714285714285715

In [267]:

Out[267]: -19.61516034985423

In [268]:

In [269]:

Mean Absolute Error 4.2789845740761244e-17

In [270]:

Mean Squared Error 1.9620411308444965e-33

In [271]:

Root Mean Squared Error 4.4294933466983516e-17

In [ ]: