D5

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

In [2]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\7_uber.csv")
df
```

Out[2]:
```
        Unnamed:                    key  fare_amount  pickup_datetime  pickup_longitude  pickup_l
               0
     0  24238194             2015-05-07          7.5       2015-05-07        -73.999817       40.
                     19:52:06.0000003                    19:52:06 UTC
     1  27835199             2009-07-17          7.7       2009-07-17        -73.994355       40.
                     20:04:56.0000002                    20:04:56 UTC
     2  44984355             2009-08-24         12.9       2009-08-24        -74.005043       40.
                    21:45:00.00000061                    21:45:00 UTC
     3  25894730             2009-06-26          5.3       2009-06-26        -73.976124       40.
                     08:22:21.0000001                    08:22:21 UTC
     4  17610152             2014-08-28         16.0       2014-08-28        -73.925023       40.
                   17:47:00.000000188                    17:47:00 UTC
   ...       ...                    ...          ...              ...               ...
199995  42598914             2012-10-28          3.0       2012-10-28        -73.987042       40.
                    10:49:00.00000053                    10:49:00 UTC
199996  16382965             2014-03-14          7.5       2014-03-14        -73.984722       40.
                     01:09:00.0000008                    01:09:00 UTC
199997  27804658             2009-06-29         30.9       2009-06-29        -73.986017       40.
                    00:42:00.00000078                    00:42:00 UTC
199998  20259894             2015-05-20         14.5       2015-05-20        -73.997124       40.
                     14:56:25.0000004                    14:56:25 UTC
199999  11951496             2010-05-15         14.1       2010-05-15        -73.984395       40.
                    04:08:00.00000076                    04:08:00 UTC

200000 rows × 9 columns
```

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

Out[3]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitud
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.73835
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.72822
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.74077
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.79084
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.74408

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 9 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   Unnamed: 0         200000 non-null  int64  
 1   key                200000 non-null  object 
 2   fare_amount        200000 non-null  float64
 3   pickup_datetime    200000 non-null  object 
 4   pickup_longitude   200000 non-null  float64
 5   pickup_latitude    200000 non-null  float64
 6   dropoff_longitude  199999 non-null  float64
 7   dropoff_latitude   199999 non-null  float64
 8   passenger_count    200000 non-null  int64  
dtypes: float64(5), int64(2), object(2)
memory usage: 13.7+ MB
```

In [5]:
```
dff=df.dropna()
```

In [21]:
```
dff.describe()
```

Out[21]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff
count	1.999990e+05	199999.000000	199999.000000	199999.000000	199999.000000	19999
mean	2.771248e+07	11.359892	-72.527631	39.935881	-72.525292	3
std	1.601386e+07	9.901760	11.437815	7.720558	13.117408	
min	1.000000e+00	-52.000000	-1340.648410	-74.015515	-3356.666300	-88
25%	1.382534e+07	6.000000	-73.992065	40.734796	-73.991407	4
50%	2.774524e+07	8.500000	-73.981823	40.752592	-73.980093	4
75%	4.155535e+07	12.500000	-73.967154	40.767158	-73.963658	4
max	5.542357e+07	499.000000	57.418457	1644.421482	1153.572603	87

In [22]:
```
dff.columns
```

Out[22]:
```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

In [23]:
```
dft=dff[['Unnamed: 0', 'fare_amount',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count']]
```

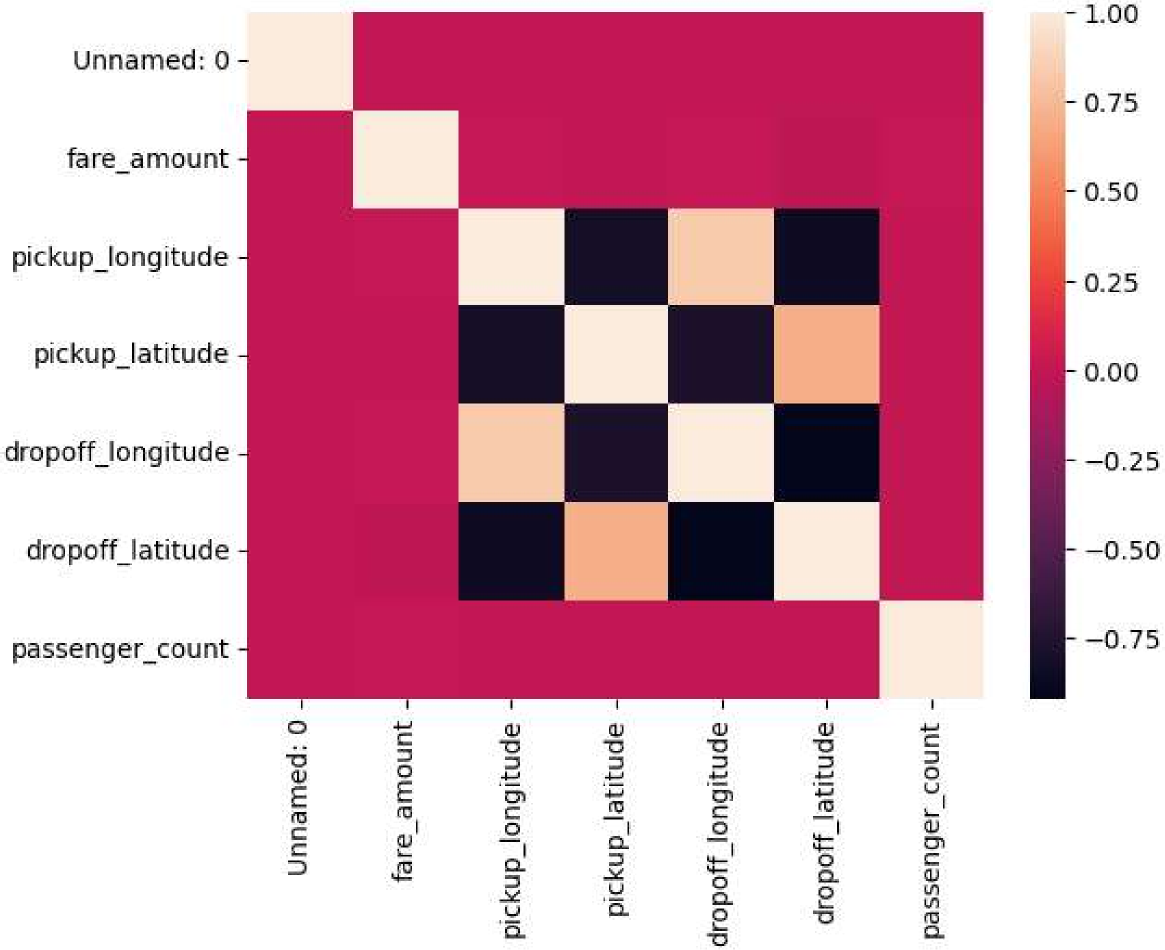
In [ ]:
```
sns.pairplot(dft)
```

In [ ]:
```
sns.distplot(dft["fare_amount"])
```

In [24]:
```
df1=dft[['Unnamed: 0', 'fare_amount',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count']]
```

In [25]:
```
sns.heatmap(df1.corr())
```

Out[25]:
```
<Axes: >
```



In [39]:
```
x=df1[['Unnamed: 0','pickup_longitude', 'pickup_latitude', 'dropoff_longitude'
       'dropoff_latitude', 'passenger_count']]
y=df1['fare_amount']
```

In [40]:
```
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 [41]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[41]:
```
▾ LinearRegression
LinearRegression()
```

In [42]:
```
print(lr.intercept_)
```

```
12.157257820553914
```

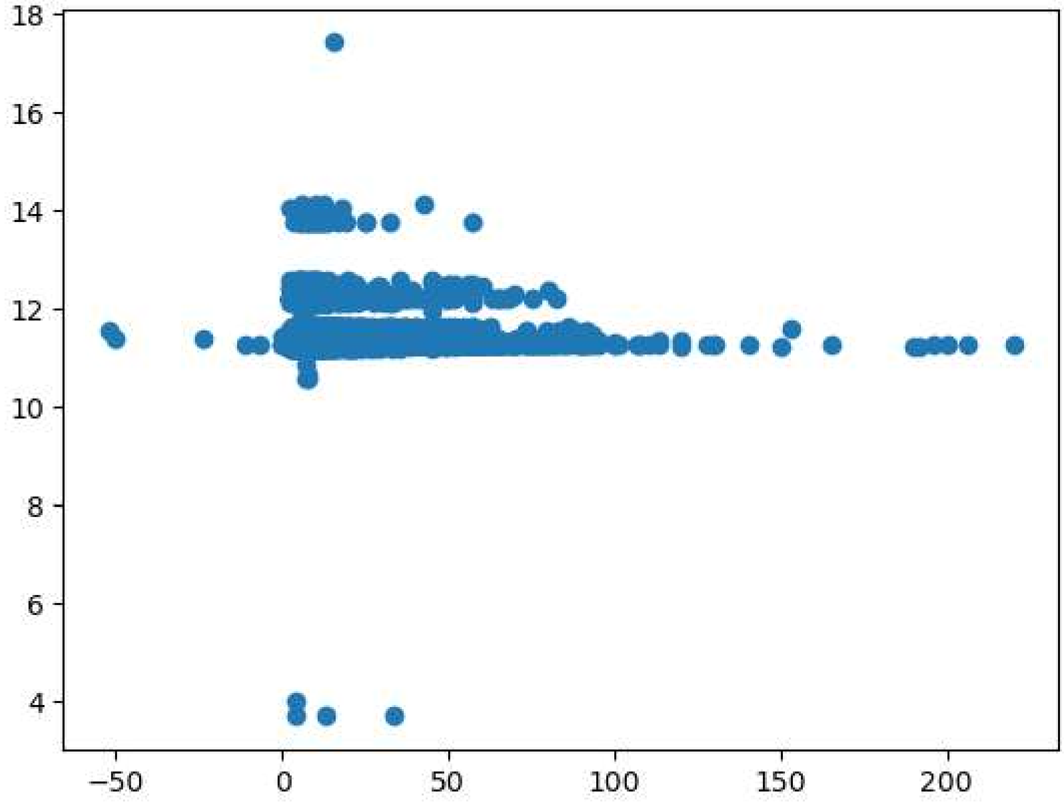
In [43]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[43]:

	Co-efficient
Unnamed: 0	1.778644e-10
pickup_longitude	1.141615e-02
pickup_latitude	-2.969902e-04
dropoff_longitude	-9.176792e-03
dropoff_latitude	-1.937769e-02
passenger_count	7.150869e-02

In [44]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[44]:
```
<matplotlib.collections.PathCollection at 0x21f516dea50>
```



In [45]:
```
print(lr.score(x_test,y_test))
```

```
-0.00010459028551013105
```

In [46]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [47]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[47]:

```
▾      Ridge
Ridge(alpha=10)
```

In [48]:

```
rr.score(x_test,y_test)
```

Out[48]:

```
-0.0001045904980259138
```

In [49]:

```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[49]:

```
▾      Lasso
Lasso(alpha=10)
```

In [50]:

```
la.score(x_test,y_test)
```

Out[50]:

```
-7.285937013001842e-05
```

In [ ]:

```

```