

In []:

In [2]:

```
# IMPORT LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\uber.csv")
a
```

Out[3]:

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

200000 rows × 9 columns



In [4]:

```
a=a.head(10)
a
```

Out[4]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude
0	24238194	2015-05-07 19:52:06.00000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.731
1	27835199	2009-07-17 20:04:56.00000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.721
2	44984355	2009-08-24 21:45:00.000000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.741
3	25894730	2009-06-26 08:22:21.00000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.791
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.741
5	44470845	2011-02-12 02:27:09.00000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.751
6	48725865	2014-10-12 07:04:00.00000002	24.5	2014-10-12 07:04:00 UTC	-73.961447	40.691
7	44195482	2012-12-11 13:52:00.000000029	2.5	2012-12-11 13:52:00 UTC	0.000000	0.001
8	15822268	2012-02-17 09:32:00.000000043	9.7	2012-02-17 09:32:00 UTC	-73.975187	40.741
9	50611056	2012-03-29 19:06:00.000000273	12.5	2012-03-29 19:06:00 UTC	-74.001065	40.741

In [5]:

```
# to find
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0            10 non-null    int64
1   key                   10 non-null    object
2   fare_amount           10 non-null    float64
3   pickup_datetime       10 non-null    object
4   pickup_longitude      10 non-null    float64
5   pickup_latitude       10 non-null    float64
6   dropoff_longitude     10 non-null    float64
7   dropoff_latitude      10 non-null    float64
8   passenger_count       10 non-null    int64
dtypes: float64(5), int64(2), object(2)
memory usage: 848.0+ bytes
```

In [6]:

```
# to display summary of statistic  
a.describe()
```

Out[6]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropo
count	1.000000e+01	10.000000	10.000000	10.000000	10.000000	
mean	3.443881e+07	10.350000	-66.580708	36.667971	-66.570116	
std	1.342943e+07	6.460693	23.394088	12.883834	23.390384	
min	1.582227e+07	2.500000	-74.005043	0.000000	-74.002720	
25%	2.465233e+07	5.850000	-73.998451	40.730757	-73.989303	
50%	3.601534e+07	8.700000	-73.975656	40.741278	-73.967167	
75%	4.485598e+07	12.800000	-73.963340	40.745346	-73.962684	
max	5.061106e+07	24.500000	0.000000	40.790844	0.000000	

In [7]:

```
# to display colum heading  
a.columns
```

Out[7]:

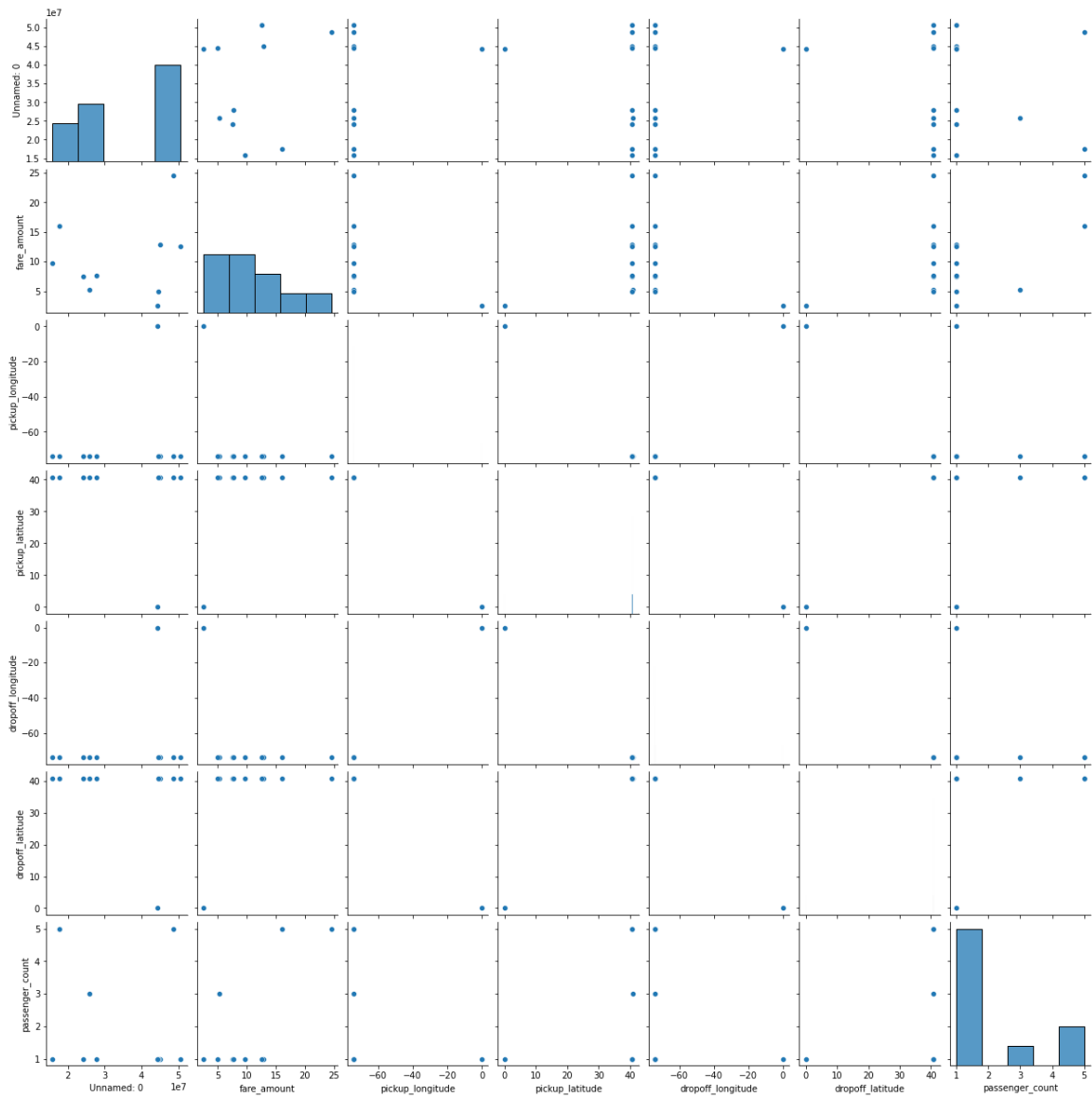
```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',  
      'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',  
      'dropoff_latitude', 'passenger_count'],  
      dtype='object')
```

In [8]:

```
sns.pairplot(a)
```

Out[8]:

<seaborn.axisgrid.PairGrid at 0x20b6c54dfd0>

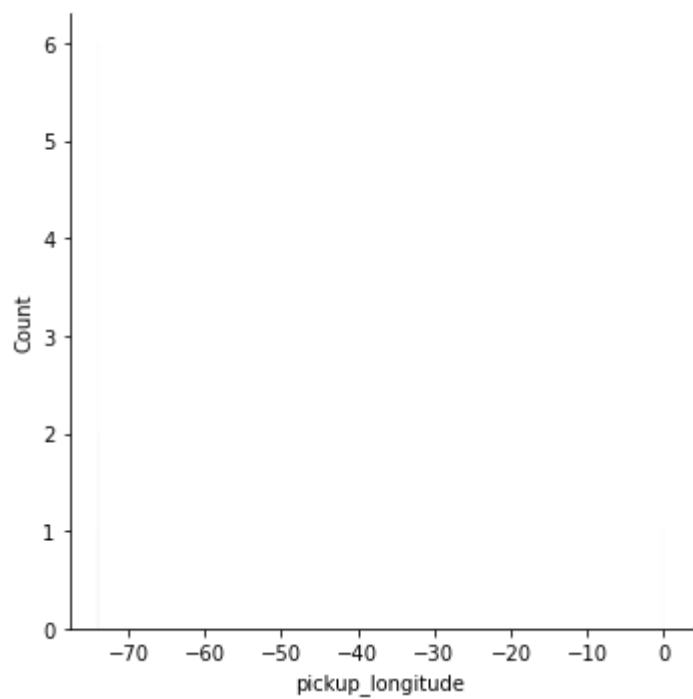


In [9]:

```
sns.displot(a["pickup_longitude"])
```

Out[9]:

<seaborn.axisgrid.FacetGrid at 0x20b6e6fb3a0>



In [10]:

```
b=a[['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',  
      'pickup_longitude', 'pickup_latitude', 'dropoff_longitude']]  
b
```

Out[10]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_lati
0	24238194	2015-05-07 19:52:06.00000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.73i
1	27835199	2009-07-17 20:04:56.00000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.72i
2	44984355	2009-08-24 21:45:00.000000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.74i
3	25894730	2009-06-26 08:22:21.00000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.79i
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.74i
5	44470845	2011-02-12 02:27:09.00000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.75i
6	48725865	2014-10-12 07:04:00.00000002	24.5	2014-10-12 07:04:00 UTC	-73.961447	40.69i
7	44195482	2012-12-11 13:52:00.000000029	2.5	2012-12-11 13:52:00 UTC	0.000000	0.00i
8	15822268	2012-02-17 09:32:00.000000043	9.7	2012-02-17 09:32:00 UTC	-73.975187	40.74i
9	50611056	2012-03-29 19:06:00.000000273	12.5	2012-03-29 19:06:00 UTC	-74.001065	40.74i

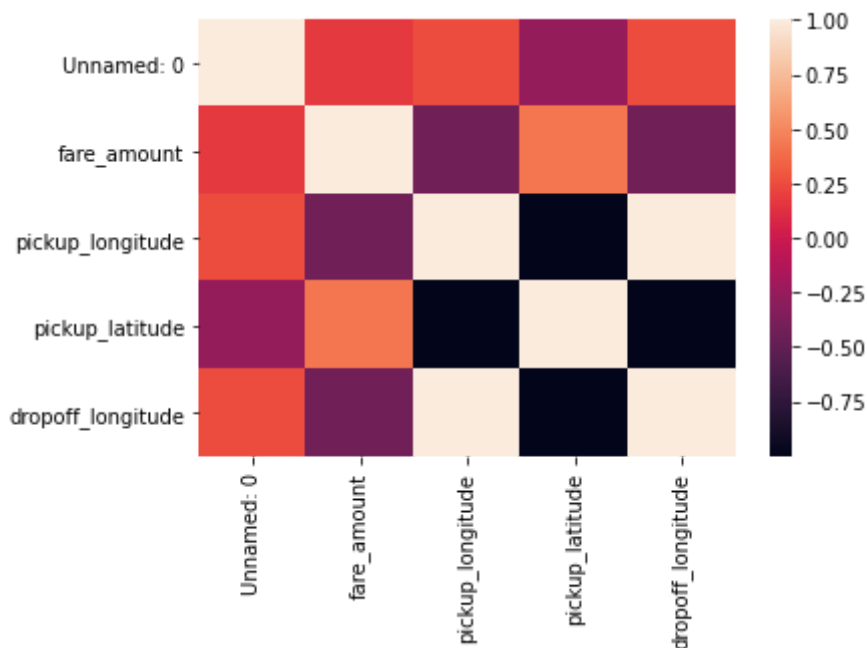


In [11]:

```
sns.heatmap(b.corr())
```

Out[11]:

<AxesSubplot:>



In [12]:

```
x=a[['pickup_latitude', 'dropoff_longitude',  
      'dropoff_latitude', 'passenger_count']]  
y=a['pickup_longitude']
```

In [13]:

```
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 [14]:

```
from sklearn.linear_model import LinearRegression  
lr=LinearRegression()  
lr.fit(x_train,y_train)
```

Out[14]:

LinearRegression()

In [15]:

```
lr.intercept_
```

Out[15]:

-0.019246413402029816

In [16]:

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

Out[16]:

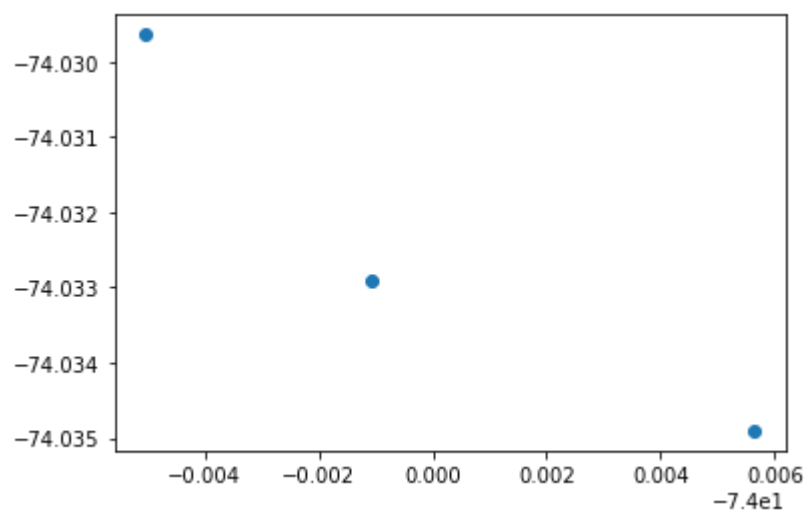
	Co-efficient
pickup_latitude	1.659767
dropoff_longitude	0.853440
dropoff_latitude	-1.925976
passenger_count	0.019239

In [17]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[17]:

<matplotlib.collections.PathCollection at 0x20b7d482c10>



In [18]:

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

Out[18]:

-54.91563115270013

In [19]:

```
lr.score(x_train,y_train)
```

Out[19]:

0.9999996326207706

In [22]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [25]:

```
rr=Ridge(alpha=10)  
rr.fit(x_test,y_test)
```

Out[25]:

Ridge(alpha=10)

In [26]:

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

Out[26]:

0.0001964680256717477

In [27]:

```
la=Lasso(alpha=10)  
la.fit(x_test,y_test)
```

Out[27]:

Lasso(alpha=10)

In [28]:

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

Out[28]:

0.0

In []: