In [1]:

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

In [2]:

```
df1=pd.read_csv(r'C:\Users\user\Downloads\7_uber.csv')
df1
```

Out[2]:

	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 [3]:

```
df=df1.head(50)
df
```

Out[3]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_la
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.7
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.7
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.7
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.7
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.7
5	44470845	2011-02-12 02:27:09.0000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.7
6	48725865	2014-10-12 07:04:00.0000002	24.5	2014-10-12 07:04:00 UTC	-73.961447	40.6
7	44195482	2012-12-11 13:52:00.00000029	2.5	2012-12-11 13:52:00 UTC	0.000000	0.0
8	15822268	2012-02-17 09:32:00.00000043	9.7	2012-02-17 09:32:00 UTC	-73.975187	40.7
9	50611056	2012-03-29 19:06:00.000000273	12.5	2012-03-29 19:06:00 UTC	-74.001065	40.7
10	2205147	2015-05-22 17:32:27.0000004	6.5	2015-05-22 17:32:27 UTC	-73.974388	40.7
11	6379048	2011-05-23 22:15:00.00000086	8.5	2011-05-23 22:15:00 UTC	0.000000	0.0
12	31892535	2011-05-17 14:03:00.000000158	3.3	2011-05-17 14:03:00 UTC	-73.966378	40.8
13	13012786	2011-06-25 11:19:00.000000102	10.9	2011-06-25 11:19:00 UTC	-73.953352	40.7
14	48411337	2010-04-06 22:20:27.0000004	6.9	2010-04-06 22:20:27 UTC	-73.973370	40.7
15	46272151	2012-02-21 09:33:00.00000028	9.7	2012-02-21 09:33:00 UTC	-73.990718	40.7
16	11875730	2011-09-01 09:21:40.0000002	4.9	2011-09-01 09:21:40 UTC	-73.988908	40.7
17	1728270	2011-03-19 23:58:27.0000003	10.5	2011-03-19 23:58:27 UTC	-74.005665	40.7
18	49173512	2015-03-25 08:58:35.0000001	12.0	2015-03-25 08:58:35 UTC	-73.962532	40.7
19	33157445	2009-08-08 00:20:00.000000183	4.9	2009-08-08 00:20:00 UTC	-73.992075	40.7
20	55085966	2014-02-18 14:26:00.00000070	10.5	2014-02-18 14:26:00 UTC	-73.980022	40.7
21	9843493	2015-03-03 23:15:03.0000003	5.0	2015-03-03 23:15:03 UTC	-73.989189	40.7
22	47537124	2009-11-26 02:58:00.0000005	4.1	2009-11-26 02:58:00 UTC	-74.010798	40.7

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_la
23	25121708	2010-09-04 16:12:00.000000152	7.7	2010-09-04 16:12:00 UTC	-73.994300	40.7
24	37339061	2010-05-12 22:32:00.000000200	12.9	2010-05-12 22:32:00 UTC	-73.972987	40.7
25	49393874	2009-02-12 17:52:18.0000001	9.5	2009-02-12 17:52:18 UTC	-73.986059	40.7
26	38755863	2014-01-21 06:55:00.00000094	5.0	2014-01-21 06:55:00 UTC	-73.957802	40.7
27	41229643	2012-11-21 17:37:19.0000002	12.0	2012-11-21 17:37:19 UTC	-73.993909	40.7
28	46387690	2009-05-06 20:06:23.0000003	4.9	2009-05-06 20:06:23 UTC	-73.977780	40.7
29	45740211	2011-12-24 02:52:00.00000056	7.3	2011-12-24 02:52:00 UTC	-73.971075	40.7
30	31945670	2011-05-21 09:00:00.00000031	25.7	2011-05-21 09:00:00 UTC	-73.944815	40.8
31	11844693	2009-02-28 15:54:57.0000002	7.7	2009-02-28 15:54:57 UTC	-74.004184	40.7
32	33836728	2013-02-11 19:09:00.000000252	10.5	2013-02-11 19:09:00 UTC	-73.982085	40.7
33	17967628	2013-09-10 20:50:25.0000001	11.0	2013-09-10 20:50:25 UTC	-73.991186	40.7
34	19277743	2014-06-04 06:49:00.000000102	39.5	2014-06-04 06:49:00 UTC	-73.788080	40.6
35	45314451	2009-06-05 05:35:00.00000011	8.1	2009-06-05 05:35:00 UTC	-73.988690	40.7
36	18779733	2011-02-19 16:31:00.000000126	5.7	2011-02-19 16:31:00 UTC	-74.010863	40.7
37	28150703	2011-08-31 19:47:00.000000254	6.9	2011-08-31 19:47:00 UTC	-73.968697	40.7
38	14631620	2010-05-18 21:28:00.000000232	7.7	2010-05-18 21:28:00 UTC	-73.968370	40.7
39	38703737	2014-02-13 17:57:00.000000102	29.0	2014-02-13 17:57:00 UTC	-73.992600	40.7
40	51671648	2010-04-01 14:42:00.000000160	15.7	2010-04-01 14:42:00 UTC	-73.973360	40.7
41	16649523	2014-04-02 14:58:32.0000002	9.0	2014-04-02 14:58:32 UTC	-73.970164	40.7
42	5218851	2011-02-01 15:25:03.0000002	4.9	2011-02-01 15:25:03 UTC	-73.987139	40.7
43	4147756	2009-01-10 22:43:36.0000007	5.4	2009-01-10 22:43:36 UTC	-73.994222	40.7
44	15145716	2012-07-12 00:59:02.0000002	3.3	2012-07-12 00:59:02 UTC	-73.982371	40.7
45	41369172	2009-02-19 08:28:42.0000001	8.9	2009-02-19 08:28:42 UTC	-73.977137	40.7
46	37192633	2014-01-16 14:58:09.0000006	17.0	2014-01-16 14:58:09 UTC	-73.993900	40.7
47	21695280	2015-01-04 09:17:47.0000001	12.0	2015-01-04 09:17:47 UTC	-73.979523	40.7

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_la
48	22405517	2013-01-03 22:24:41.0000002	56.8	2013-01-03 22:24:41 UTC	-73.993498	40.7
49	54858310	2013-05-23 10:57:00.000000198	13.5	2013-05-23 10:57:00 UTC	-73.962043	40.8

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         50 non-null     int64  
 1   key                50 non-null     object 
 2   fare_amount        50 non-null     float64
 3   pickup_datetime    50 non-null     object 
 4   pickup_longitude   50 non-null     float64
 5   pickup_latitude    50 non-null     float64
 6   dropoff_longitude  50 non-null     float64
 7   dropoff_latitude   50 non-null     float64
 8   passenger_count    50 non-null     int64  
dtypes: float64(5), int64(2), object(2)
memory usage: 3.6+ KB
```

In [5]:

```
df.describe()
```

Out[5]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropc
count	5.000000e+01	50.000000	50.000000	50.000000	50.000000	
mean	3.031476e+07	11.176000	-71.018026	39.122071	-71.015808	
std	1.592279e+07	9.555158	14.643705	8.066889	14.643240	
min	1.728270e+06	2.500000	-74.010863	0.000000	-74.009767	
25%	1.688968e+07	5.475000	-73.993274	40.739826	-73.988552	
50%	3.191910e+07	8.700000	-73.979772	40.751817	-73.978048	
75%	4.523193e+07	12.000000	-73.968777	40.764933	-73.963609	
max	5.508597e+07	56.800000	0.000000	40.834367	0.000000	

In [6]:

```
df.columns
```

Out[6]:

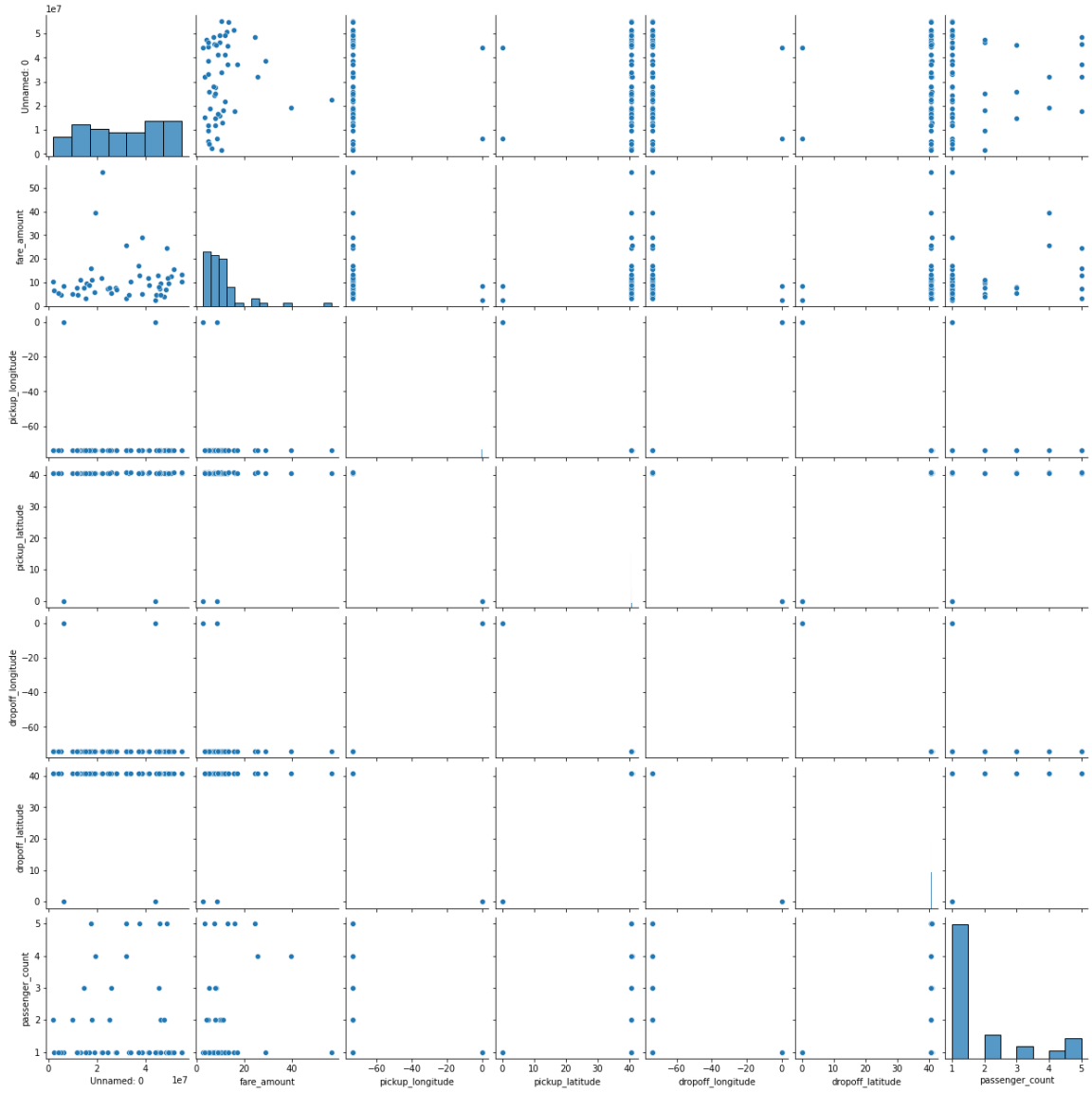
```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

In [7]:

```
sns.pairplot(df)
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x1d815209dc0>
```



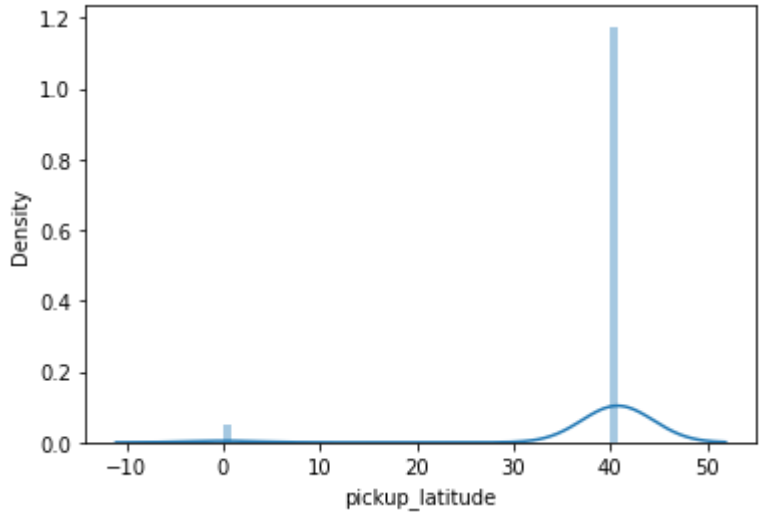
In [8]:

```
sns.distplot(df['pickup_latitude'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[8]:

```
<AxesSubplot:xlabel='pickup_latitude', ylabel='Density'>
```

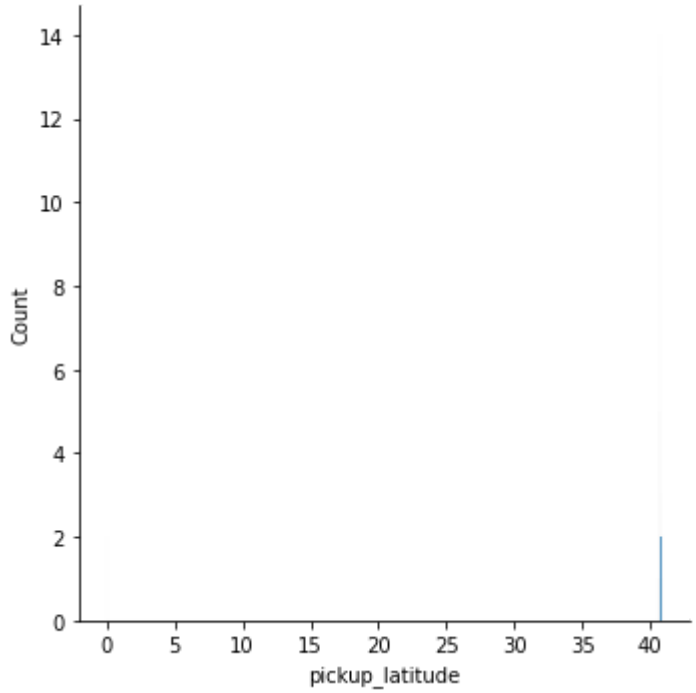


In [9]:

```
sns.displot(df["pickup_latitude"])
```

Out[9]:

```
<seaborn.axisgrid.FacetGrid at 0x1d8141e5f70>
```



In [10]:

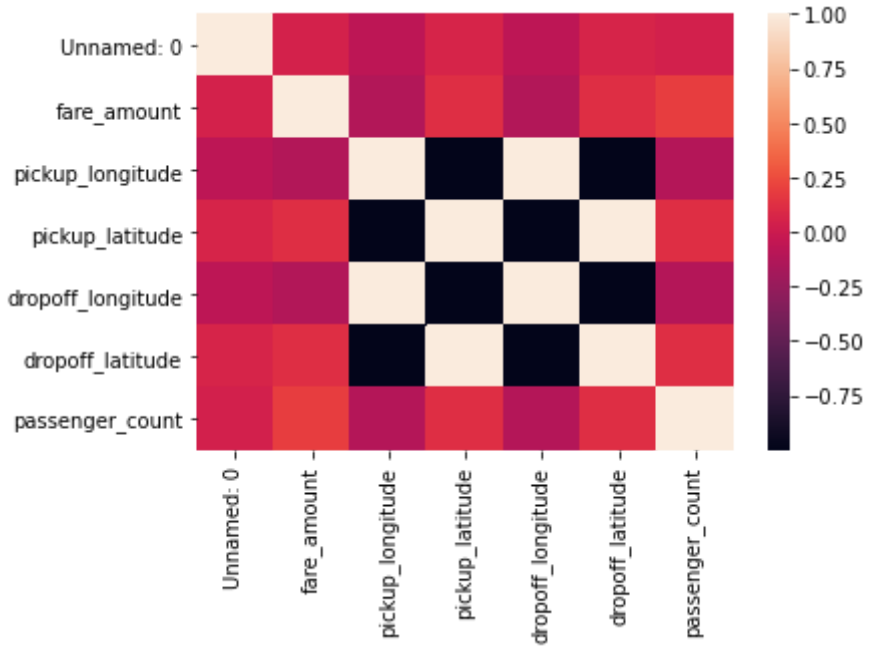
```
df1=df[['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count']]
```

In [11]:

```
sns.heatmap(df1.corr())
```

Out[11]:

<AxesSubplot:>



In [12]:

```
df2=df.dropna(axis=1)
df2
```

Out[12]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_la
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.7
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.7
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.7
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.7
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.7
5	44470845	2011-02-12 02:27:09.0000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.7
6	48725865	2014-10-12 07:04:00.0000002	24.5	2014-10-12 07:04:00 UTC	-73.961447	40.6
7	44195482	2012-12-11 13:52:00.00000029	2.5	2012-12-11 13:52:00 UTC	0.000000	0.0
8	15822268	2012-02-17 09:32:00.00000043	9.7	2012-02-17 09:32:00 UTC	-73.975187	40.7
9	50611056	2012-03-29 19:06:00.000000273	12.5	2012-03-29 19:06:00 UTC	-74.001065	40.7
10	2205147	2015-05-22 17:32:27.0000004	6.5	2015-05-22 17:32:27 UTC	-73.974388	40.7
11	6379048	2011-05-23 22:15:00.00000086	8.5	2011-05-23 22:15:00 UTC	0.000000	0.0
12	31892535	2011-05-17 14:03:00.000000158	3.3	2011-05-17 14:03:00 UTC	-73.966378	40.8
13	13012786	2011-06-25 11:19:00.000000102	10.9	2011-06-25 11:19:00 UTC	-73.953352	40.7
14	48411337	2010-04-06 22:20:27.0000004	6.9	2010-04-06 22:20:27 UTC	-73.973370	40.7
15	46272151	2012-02-21 09:33:00.00000028	9.7	2012-02-21 09:33:00 UTC	-73.990718	40.7
16	11875730	2011-09-01 09:21:40.0000002	4.9	2011-09-01 09:21:40 UTC	-73.988908	40.7
17	1728270	2011-03-19 23:58:27.0000003	10.5	2011-03-19 23:58:27 UTC	-74.005665	40.7
18	49173512	2015-03-25 08:58:35.0000001	12.0	2015-03-25 08:58:35 UTC	-73.962532	40.7
19	33157445	2009-08-08 00:20:00.000000183	4.9	2009-08-08 00:20:00 UTC	-73.992075	40.7
20	55085966	2014-02-18 14:26:00.00000070	10.5	2014-02-18 14:26:00 UTC	-73.980022	40.7
21	9843493	2015-03-03 23:15:03.0000003	5.0	2015-03-03 23:15:03 UTC	-73.989189	40.7
22	47537124	2009-11-26 02:58:00.0000005	4.1	2009-11-26 02:58:00 UTC	-74.010798	40.7

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_la
23	25121708	2010-09-04 16:12:00.000000152	7.7	2010-09-04 16:12:00 UTC	-73.994300	40.7
24	37339061	2010-05-12 22:32:00.000000200	12.9	2010-05-12 22:32:00 UTC	-73.972987	40.7
25	49393874	2009-02-12 17:52:18.0000001	9.5	2009-02-12 17:52:18 UTC	-73.986059	40.7
26	38755863	2014-01-21 06:55:00.00000094	5.0	2014-01-21 06:55:00 UTC	-73.957802	40.7
27	41229643	2012-11-21 17:37:19.0000002	12.0	2012-11-21 17:37:19 UTC	-73.993909	40.7
28	46387690	2009-05-06 20:06:23.0000003	4.9	2009-05-06 20:06:23 UTC	-73.977780	40.7
29	45740211	2011-12-24 02:52:00.00000056	7.3	2011-12-24 02:52:00 UTC	-73.971075	40.7
30	31945670	2011-05-21 09:00:00.00000031	25.7	2011-05-21 09:00:00 UTC	-73.944815	40.8
31	11844693	2009-02-28 15:54:57.0000002	7.7	2009-02-28 15:54:57 UTC	-74.004184	40.7
32	33836728	2013-02-11 19:09:00.000000252	10.5	2013-02-11 19:09:00 UTC	-73.982085	40.7
33	17967628	2013-09-10 20:50:25.0000001	11.0	2013-09-10 20:50:25 UTC	-73.991186	40.7
34	19277743	2014-06-04 06:49:00.000000102	39.5	2014-06-04 06:49:00 UTC	-73.788080	40.6
35	45314451	2009-06-05 05:35:00.00000011	8.1	2009-06-05 05:35:00 UTC	-73.988690	40.7
36	18779733	2011-02-19 16:31:00.000000126	5.7	2011-02-19 16:31:00 UTC	-74.010863	40.7
37	28150703	2011-08-31 19:47:00.000000254	6.9	2011-08-31 19:47:00 UTC	-73.968697	40.7
38	14631620	2010-05-18 21:28:00.000000232	7.7	2010-05-18 21:28:00 UTC	-73.968370	40.7
39	38703737	2014-02-13 17:57:00.000000102	29.0	2014-02-13 17:57:00 UTC	-73.992600	40.7
40	51671648	2010-04-01 14:42:00.000000160	15.7	2010-04-01 14:42:00 UTC	-73.973360	40.7
41	16649523	2014-04-02 14:58:32.0000002	9.0	2014-04-02 14:58:32 UTC	-73.970164	40.7
42	5218851	2011-02-01 15:25:03.0000002	4.9	2011-02-01 15:25:03 UTC	-73.987139	40.7
43	4147756	2009-01-10 22:43:36.0000007	5.4	2009-01-10 22:43:36 UTC	-73.994222	40.7
44	15145716	2012-07-12 00:59:02.0000002	3.3	2012-07-12 00:59:02 UTC	-73.982371	40.7
45	41369172	2009-02-19 08:28:42.0000001	8.9	2009-02-19 08:28:42 UTC	-73.977137	40.7
46	37192633	2014-01-16 14:58:09.0000006	17.0	2014-01-16 14:58:09 UTC	-73.993900	40.7
47	21695280	2015-01-04 09:17:47.0000001	12.0	2015-01-04 09:17:47 UTC	-73.979523	40.7

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_la
48	22405517	2013-01-03 22:24:41.0000002	56.8	2013-01-03 22:24:41 UTC	-73.993498	40.7
49	54858310	2013-05-23 10:57:00.0000001	13.5	2013-05-23 10:57:00 UTC	-73.962043	40.8

In [13]:

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

In [14]:

```
from sklearn.model_selection import train_test_split
```

In [15]:

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

In [16]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)#ValueError: Input contains NaN, infinity or a value too large for
```

Out[16]:

```
LinearRegression()
```

In [17]:

```
print(lr.intercept_)
```

```
[-0.0166357]
```

In [18]:

```
coef= pd.DataFrame(lr.coef_)
coef
```

Out[18]:

	0	1	2	3	4	5
0	3.738319e-10	-0.000103	-0.221355	-0.158854	0.309727	0.007723

In [19]:

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

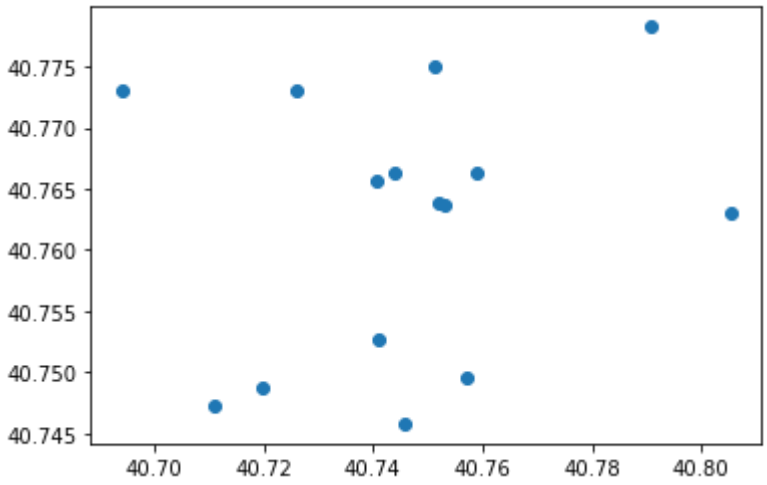
```
-0.3381258647356802
```

In [20]:

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

Out[20]:

```
<matplotlib.collections.PathCollection at 0x1d82ad1c5b0>
```



In [21]:

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

Out[21]:

```
-0.3381258647356802
```

In [22]:

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

Out[22]:

```
0.9999942943569997
```

In [23]:

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

In [24]:

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

Out[24]:

```
Ridge(alpha=10)
```

In [25]:

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

Out[25]:

```
-0.28898631063239355
```

In [26]:

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

Out[26]:

```
Lasso(alpha=10)
```

In [27]:

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

Out[27]:

```
-26.21065815499053
```

Elastic Net

In [28]:

```
from sklearn.linear_model import ElasticNet
en = ElasticNet()
en.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 9.06915014324727
6, tolerance: 0.31321350465365333
  model = cd_fast.enet_coordinate_descent(
```

Out[28]:

```
ElasticNet()
```

In [29]:

```
print(en.coef_)
```

```
[ 7.46191950e-10  0.00000000e+00 -2.83532659e-01 -2.65194373e-01
  0.00000000e+00  0.00000000e+00]
```

In [30]:

```
print(en.intercept_)
```

```
[0.13079592]
```

In [31]:

```
prediction=en.predict(x_test)
print(prediction)
```

```
[40.74208875 40.75627943 40.73733944 40.73995108 40.73361274 40.72133989
 40.76134623 40.75400912 40.75519883 40.76166553 40.76370525 40.73911538
 40.72786572 40.77771982 40.73300489]
```

In [32]:

```
print(en.score(x_test,y_test))
```

```
-0.23908136329998042
```

Evaluation Metrics

In [33]:

```
from sklearn import metrics
```

In [34]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 0.02572221872820156
```

In [35]:

```
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error: 0.0009113279771627271
```

In [36]:

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

```
Root Mean Squared Error: 0.03018820924074045
```