SUMESH R -20104169

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
df = pd.read_csv("7_uber.csv")[0:600].dropna(axis=1)
df
```

Out[2]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dr
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	
...	...	...	...	...	...	...	...
595	3268252	2012-06-12 11:41:16.0000001	6.1	2012-06-12 11:41:16 UTC	-73.952088	40.786637	
596	5992726	2011-09-20 22:04:00.00000089	9.7	2011-09-20 22:04:00 UTC	-73.956445	40.775568	
597	42806767	2011-09-07 14:15:00.00000041	14.9	2011-09-07 14:15:00 UTC	-74.009533	40.705928	
598	8308940	2011-02-17 04:27:00.0000008	6.9	2011-02-17 04:27:00 UTC	-74.005672	40.725620	
599	41718495	2011-05-29 22:07:00.000000102	7.7	2011-05-29 22:07:00 UTC	-73.956430	40.813242	

600 rows × 9 columns

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

Out[3]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	drop
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	

Data cleaning and pre processing

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         600 non-null    int64  
 1   key                600 non-null    object 
 2   fare_amount        600 non-null    float64
 3   pickup_datetime    600 non-null    object 
 4   pickup_longitude   600 non-null    float64
 5   pickup_latitude    600 non-null    float64
 6   dropoff_longitude  600 non-null    float64
 7   dropoff_latitude   600 non-null    float64
 8   passenger_count    600 non-null    int64  
dtypes: float64(5), int64(2), object(2)
memory usage: 42.3+ KB
```

In [5]:
```
df.describe()
```

Out[5]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
count	6.000000e+02	600.000000	600.000000	600.000000	600.000000	600.000000
mean	2.754724e+07	10.797317	-72.128589	39.733052	-72.249515	39.800268
std	1.603314e+07	8.299398	11.559512	6.367668	11.176725	6.156939
min	1.862090e+05	2.500000	-74.030417	0.000000	-74.027813	0.000000
25%	1.294860e+07	6.000000	-73.992810	40.735292	-73.991901	40.731075
50%	2.791547e+07	8.100000	-73.982352	40.752495	-73.980722	40.750670
75%	4.171866e+07	12.500000	-73.968882	40.766560	-73.965445	40.767777

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude
max	5.519870e+07	57.330000	0.001782	40.850558	0.000875	40.901391

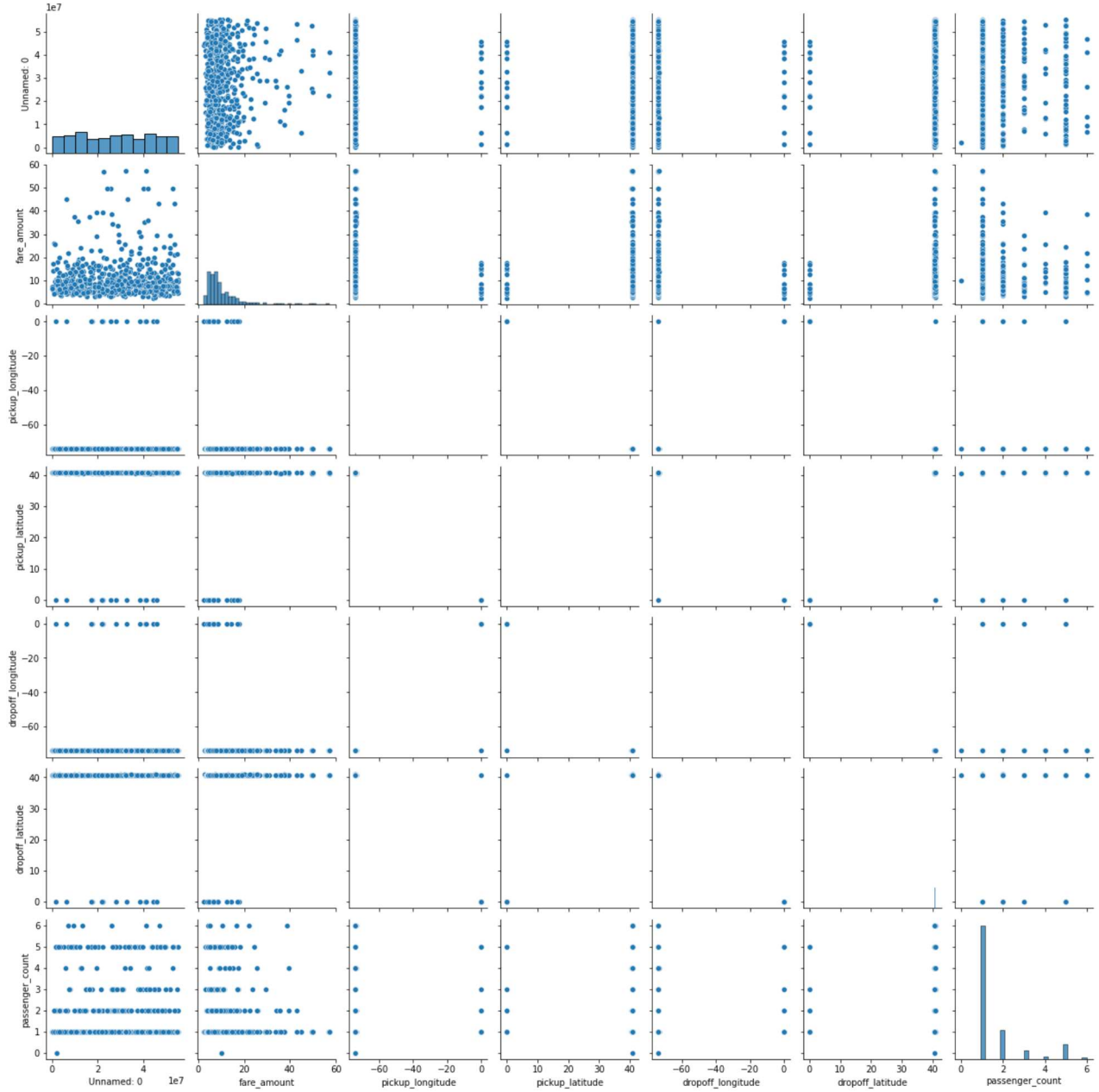
In [6]:
```
df.columns
```

Out[6]:
```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

EDA and VISUALIZATION

In [7]:
```
sns.pairplot(df)
```

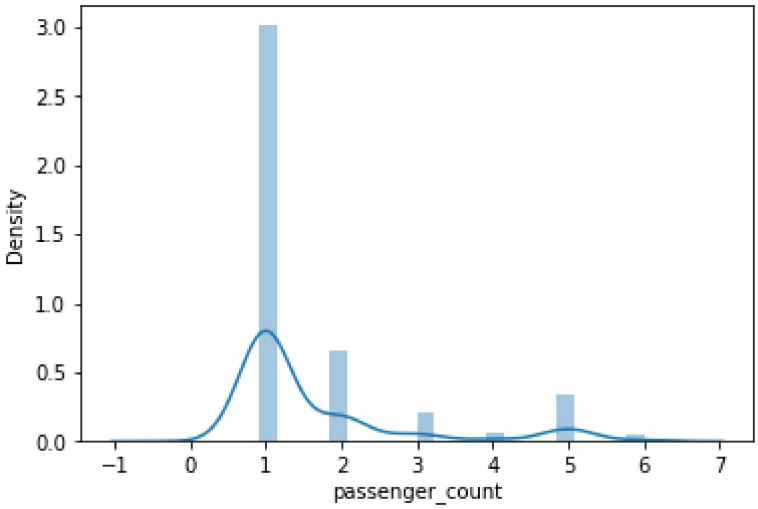
Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x19d447ee3d0>
```



In [8]:
```
sns.distplot(df["passenger_count"])
```

```
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 adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

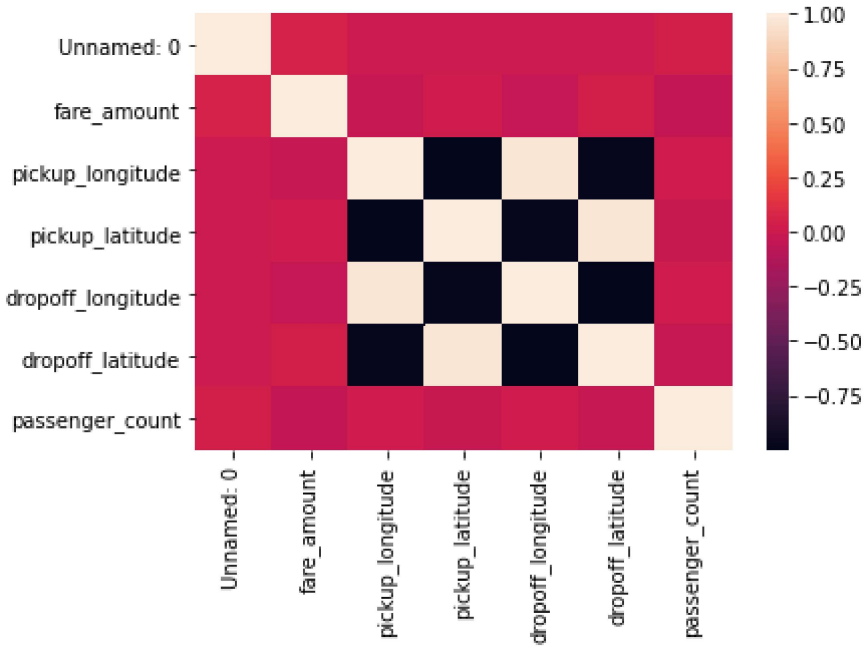
Out[8]: <AxesSubplot:xlabel='passenger_count', ylabel='Density'>



In [9]:
```
df1 = df[['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count']]
```

In [10]:
```
sns.heatmap(df1.corr())
```

Out[10]: <AxesSubplot:>



In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Unnamed: 0         600 non-null    int64  
 1   key                600 non-null    object 
 2   fare_amount        600 non-null    float64
 3   pickup_datetime    600 non-null    object 
```

```
 4   pickup_longitude   600 non-null    float64
 5   pickup_latitude    600 non-null    float64
 6   dropoff_longitude  600 non-null    float64
 7   dropoff_latitude   600 non-null    float64
 8   passenger_count    600 non-null    int64
dtypes: float64(5), int64(2), object(2)
memory usage: 42.3+ KB
```

In [12]:
```
x = df1[['Unnamed: 0',  'fare_amount',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude']]
y = df1['passenger_count']
```

split the data into training and test data

In [13]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

In [14]:
```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

Out[14]:
```
LinearRegression()
```

In [15]:
```
lr.intercept_
```

Out[15]:
```
1.995428756841116
```

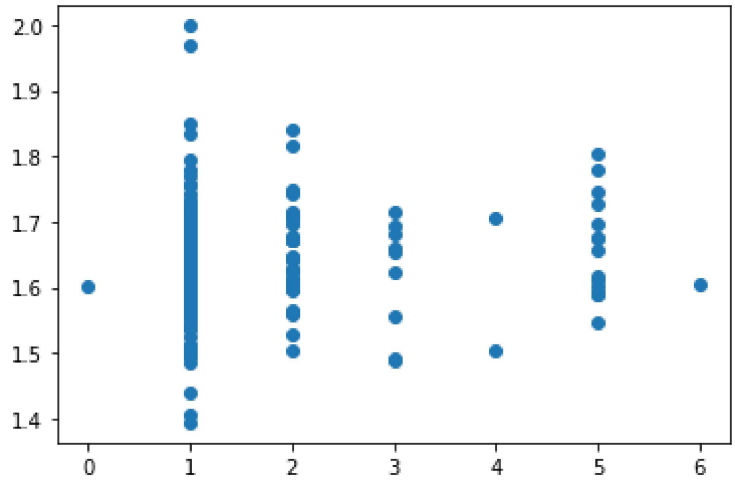
In [16]:
```
coeff = pd.DataFrame(lr.coef_, x.columns, columns =['Co-efficient'])
coeff
```

Out[16]:

	Co-efficient
Unnamed: 0	1.362330e-09
fare_amount	-4.030148e-03
pickup_longitude	-1.045435e-01
pickup_latitude	-1.731362e-01
dropoff_longitude	1.001974e+00
dropoff_latitude	1.793730e+00

In [17]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test, prediction)
```

Out[17]:
```
<matplotlib.collections.PathCollection at 0x19d59fd4910>
```



In [18]:
```
lr.score(x_test,y_test)
```

Out[18]: -0.0009036009247667121