Data Collection

In [1]:

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

In [2]:

```
df=pd.read_csv(r"C:\Users\user\Downloads\uber.csv")[0:50]
df
```

Out[2]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_lati
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.73
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.72
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.74
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.79
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.74
5	44470845	2011-02-12 02:27:09.0000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.75
6	48725865	2014-10-12	24.5	2014-10-12	-73.961447	40.69

In [3]:
```
df.head(10)
```

Out[3]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude
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
5	44470845	2011-02-12 02:27:09.0000006	4.9	2011-02-12 02:27:09 UTC	-73.969019	40.755910
6	48725865	2014-10-12 07:04:00.0000002	24.5	2014-10-12 07:04:00 UTC	-73.961447	40.693965
7	44195482	2012-12-11 13:52:00.00000029	2.5	2012-12-11 13:52:00 UTC	0.000000	0.000000
8	15822268	2012-02-17 09:32:00.00000043	9.7	2012-02-17 09:32:00 UTC	-73.975187	40.745767
9	50611056	2012-03-29 19:06:00.000000273	12.5	2012-03-29 19:06:00 UTC	-74.001065	40.741787

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

Out[4]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_la
count	5.000000e+01	50.000000	50.000000	50.000000	50.000000	50.0
mean	3.031476e+07	11.176000	-71.018026	39.122071	-71.015808	39.1
std	1.592279e+07	9.555158	14.643705	8.066889	14.643240	8.0
min	1.728270e+06	2.500000	-74.010863	0.000000	-74.009767	0.0
25%	1.688968e+07	5.475000	-73.993274	40.739826	-73.988552	40.7
50%	3.191910e+07	8.700000	-73.979772	40.751817	-73.978048	40.7
75%	4.523193e+07	12.000000	-73.968777	40.764933	-73.963609	40.7
max	5.508597e+07	56.800000	0.000000	40.834367	0.000000	40.8

In [5]:
```
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 [6]:
```
df.columns
```

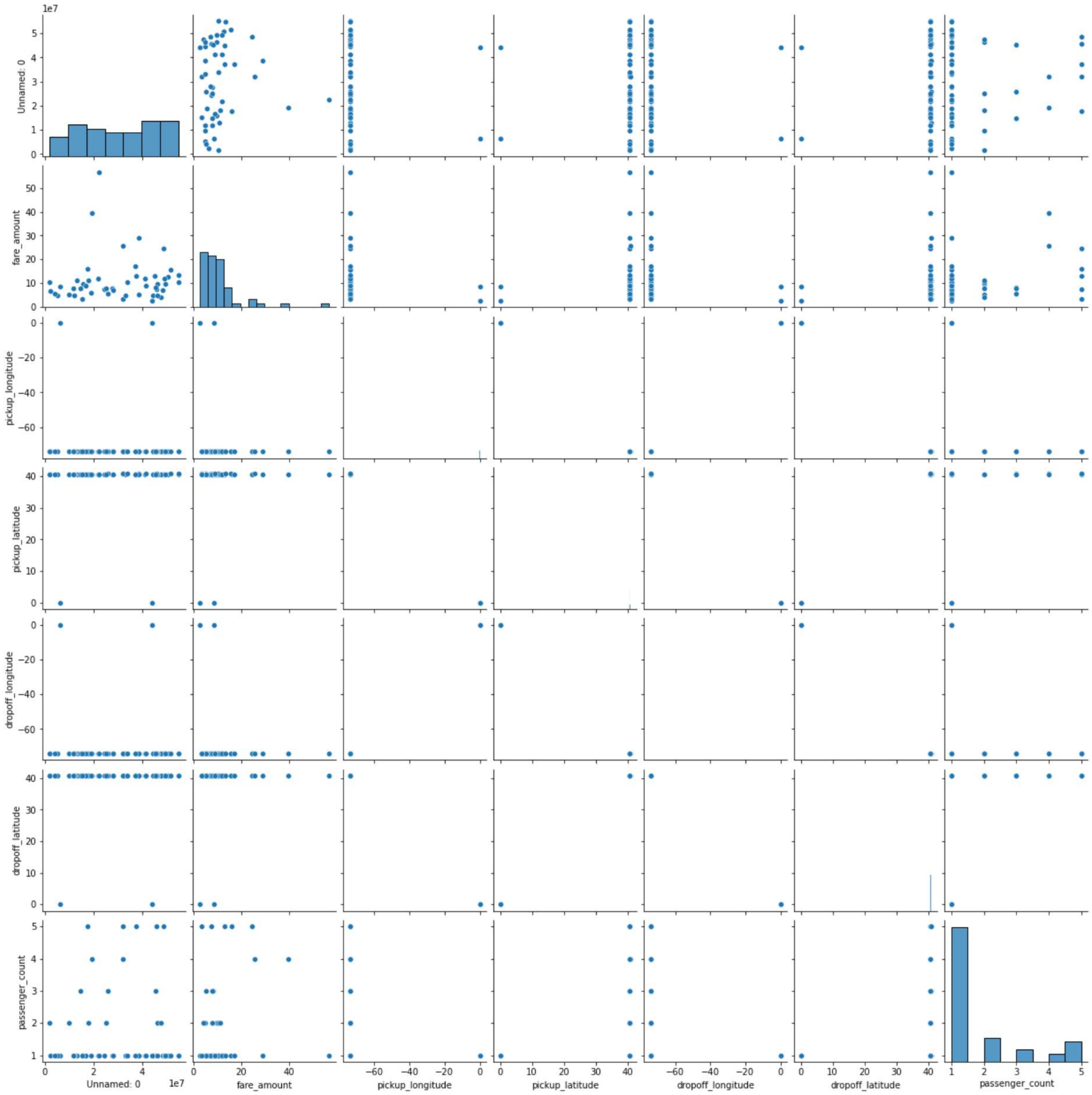
Out[6]:
```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

In [7]:

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

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x1ac7888e130>
```



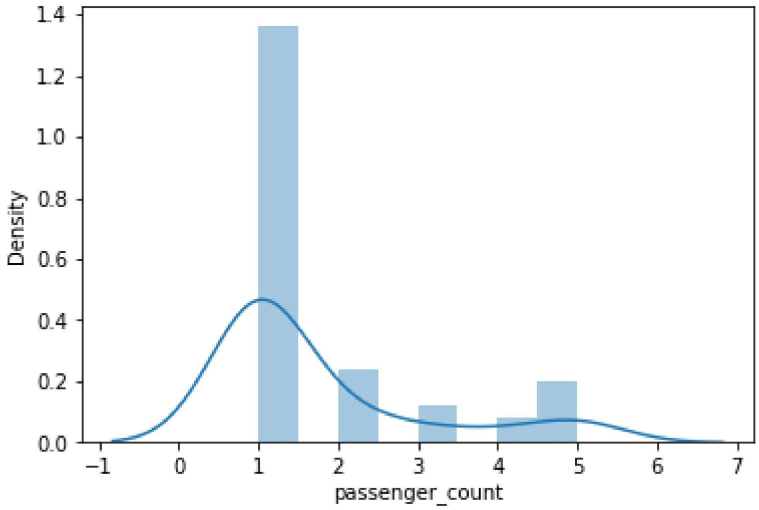
In [8]:

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

```
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[8]:

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



In [10]:

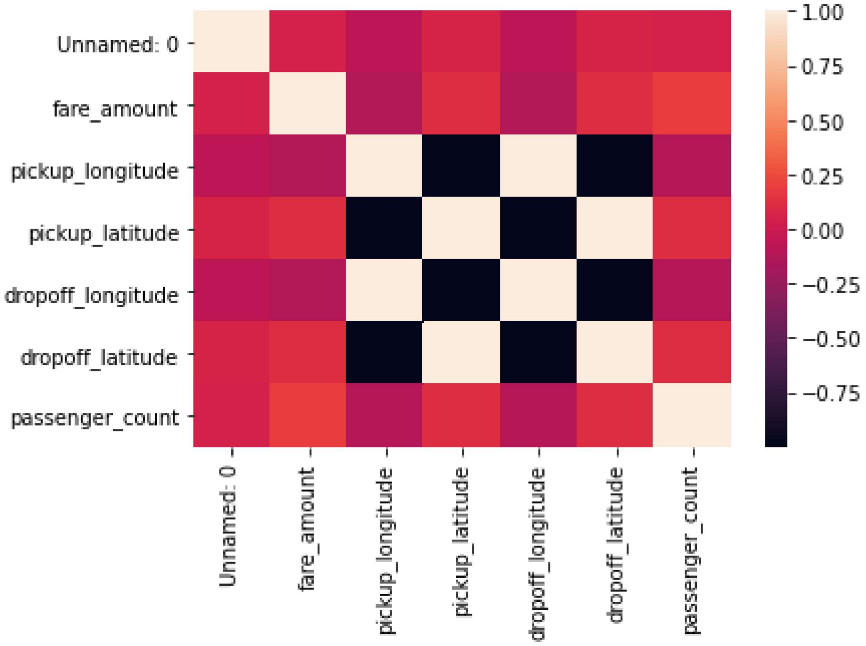
```
d=df[['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count']]
```

In [11]:

```
sns.heatmap(d.corr())
```

Out[11]:

```
<AxesSubplot:>
```



To TRAIN THE MODEL=MODEL BUILDING

WE ARE GOING TO TRAIN LINEAR REGRESSION MODEL;WE NEED TO SPLIT OUT DATA INTO TWO VARIABLES X AND Y IS INDEPENDENT VARIABLE (INPUT) AND Y IS DEPENDENT ON X (OUTPUT) WE COULD IGNORE ADDRESS COLUMN AS IT IS NOT REQUIRED FOR OUR MODEL

In [15]:

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

In [16]:

```
#to split my dataset into traning and test data

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 [17]:

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(x_train,y_train)
```

Out[17]:

```
LinearRegression()
```

In [18]:

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

```
0.8894301442348738
```

In [19]:

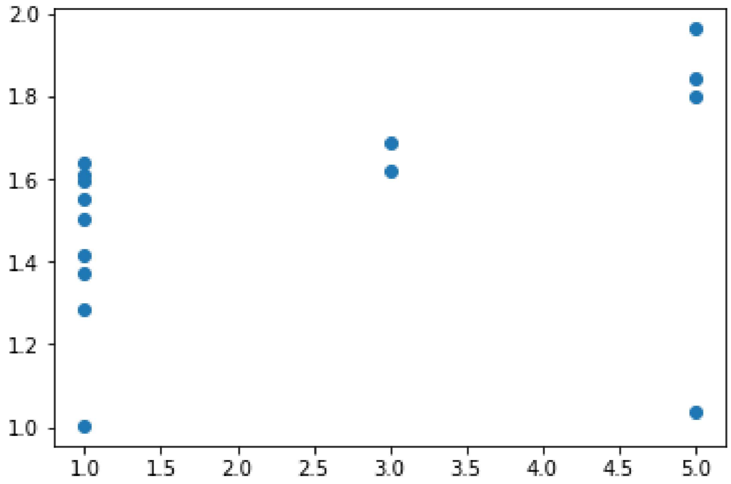
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['co-effecient'])
coeff
```

Out[19]:

	co-effecient
Unnamed: 0	1.900822e-09
fare_amount	1.205181e-02
pickup_longitude	8.533600e+00
pickup_latitude	5.875177e+00
dropoff_longitude	-4.833191e+00
dropoff_latitude	8.535022e-01

```
In [20]: prediction=lr.predict(x_test)
         plt.scatter(y_test,prediction)

Out[20]: <matplotlib.collections.PathCollection at 0x1ac0cc05550>
```



```
In [21]: print(lr.score(x_test,y_test))

         -0.12117367047318783

In [ ]:
```