

```
In [45]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [46]: data=pd.read_csv(r"C:\Users\user\Downloads\7_uber.csv")
data
```

Out[46]:

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

200000 rows × 9 columns



```
In [47]: df=data.head(100)
df
```

Out[47]:

	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	40.7383
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.7282
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.7407
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.7906
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.7440
...
95	25431833	2015-04-11 08:47:47.0000001	9.5	2015-04-11 08:47:47 UTC	-73.978432	40.7523
96	44792012	2011-10-03 20:29:00.000000179	4.5	2011-10-03 20:29:00 UTC	-73.990055	40.7564
97	18571020	2010-04-26 03:12:44.0000001	3.3	2010-04-26 03:12:44 UTC	-73.982326	40.7313
98	37942404	2011-11-18 09:51:00.000000166	30.9	2011-11-18 09:51:00 UTC	-73.995888	40.7590
99	29024472	2009-08-30 14:03:55.0000002	26.9	2009-08-30 14:03:55 UTC	-73.990137	40.7560

100 rows × 9 columns




```
In [48]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0            100 non-null   int64
1   key                   100 non-null   object
2   fare_amount           100 non-null   float64
3   pickup_datetime       100 non-null   object
4   pickup_longitude      100 non-null   float64
5   pickup_latitude       100 non-null   float64
6   dropoff_longitude     100 non-null   float64
7   dropoff_latitude      100 non-null   float64
8   passenger_count       100 non-null   int64
dtypes: float64(5), int64(2), object(2)
memory usage: 7.2+ KB
```

```
In [49]: df.describe()
```

```
Out[49]:
```

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_
count	1.000000e+02	100.000000	100.000000	100.000000	100.000000	100
mean	2.810554e+07	11.065700	-71.019759	39.123621	-71.015479	39
std	1.635033e+07	9.029756	14.569902	8.026358	14.569028	8
min	2.268700e+05	2.500000	-74.013173	0.000000	-74.016152	0
25%	1.422691e+07	5.475000	-73.992601	40.733982	-73.989142	40
50%	2.710896e+07	8.100000	-73.982002	40.752764	-73.979396	40
75%	4.480811e+07	12.600000	-73.968615	40.765572	-73.960980	40
max	5.508597e+07	56.800000	0.000000	40.850558	0.000000	40

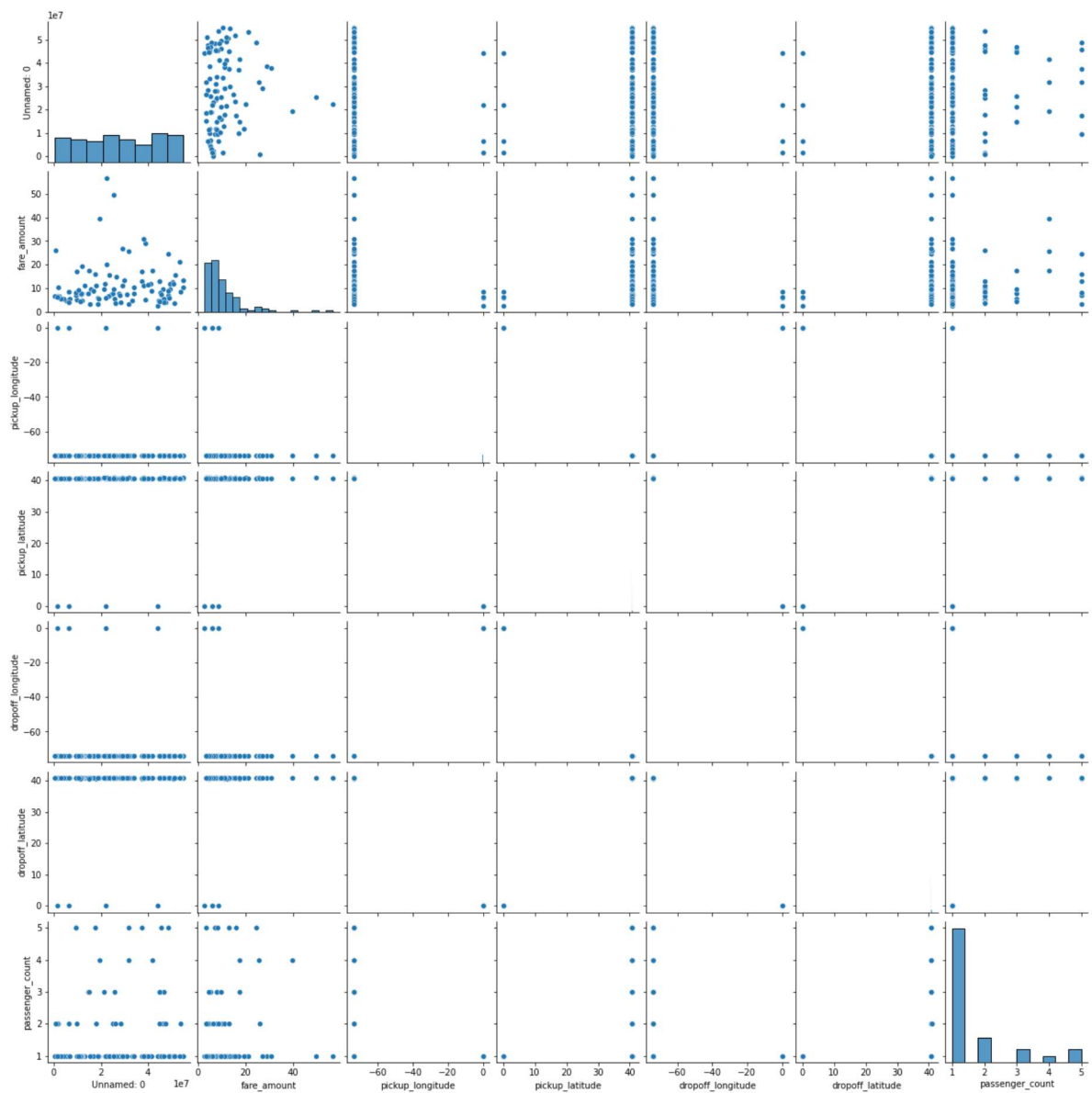


```
In [50]: df.columns
```

```
Out[50]: Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',  
               'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',  
               'dropoff_latitude', 'passenger_count'],  
              dtype='object')
```

```
In [51]: sns.pairplot(df)
```

```
Out[51]: <seaborn.axisgrid.PairGrid at 0x1ed5a384ee0>
```



```
In [52]: da=df[['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
                'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
                'dropoff_latitude', 'passenger_count']]
da
```

Out[52]:

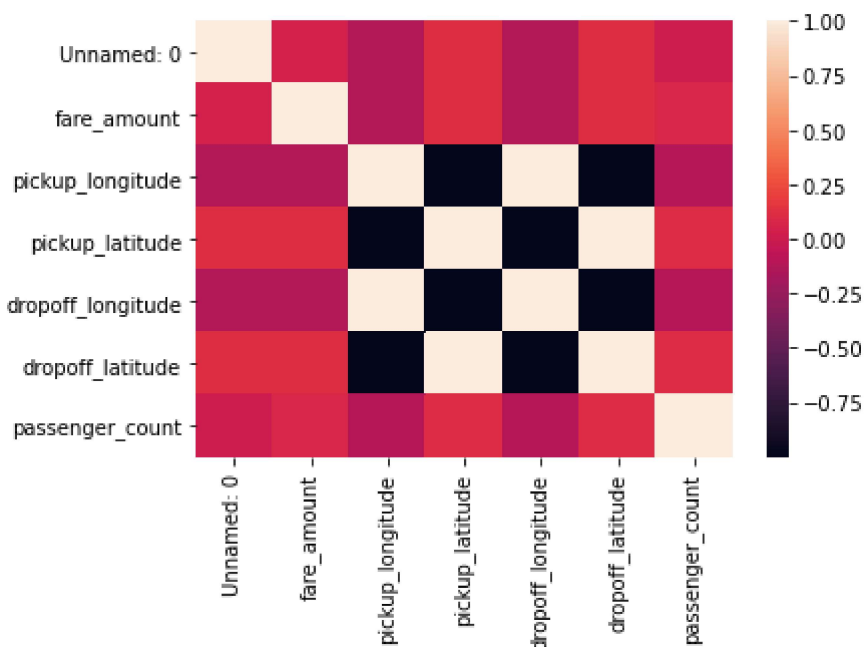
	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.7383
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.7282
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.7407
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.7908
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.7440
...
95	25431833	2015-04-11 08:47:47.0000001	9.5	2015-04-11 08:47:47 UTC	-73.978432	40.7523
96	44792012	2011-10-03 20:29:00.000000179	4.5	2011-10-03 20:29:00 UTC	-73.990055	40.7564
97	18571020	2010-04-26 03:12:44.0000001	3.3	2010-04-26 03:12:44 UTC	-73.982326	40.7313
98	37942404	2011-11-18 09:51:00.000000166	30.9	2011-11-18 09:51:00 UTC	-73.995888	40.7590
99	29024472	2009-08-30 14:03:55.0000002	26.9	2009-08-30 14:03:55 UTC	-73.990137	40.7560

100 rows × 9 columns



```
In [53]: sns.heatmap(da.corr())
```

```
Out[53]: <AxesSubplot:>
```



```
In [55]: x=df[['fare_amount']]
y=df['passenger_count']
```

```
In [56]: 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 [57]: from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

```
Out[57]: LinearRegression()
```

```
In [58]: print(lr.intercept_)
```

```
1.56284497031874
```

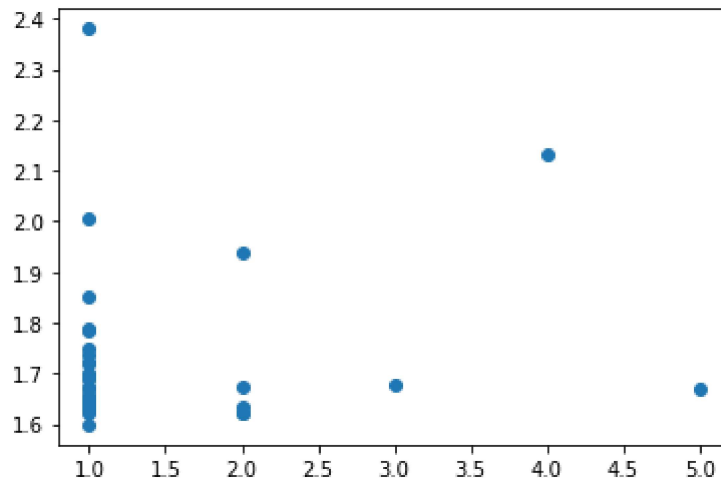
```
In [59]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

```
Out[59]:
```

	Co-efficient
fare_amount	0.01439

```
In [60]: prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[60]: <matplotlib.collections.PathCollection at 0x1ed6b8d6d30>



```
In [61]: print(lr.score(x_test,y_test))
```

-0.07265446995602343

```
In [62]: print(lr.score(x_train,y_train))
```

0.007307889101744958

```
In [63]: from sklearn.linear_model import Ridge,Lasso
```

```
In [64]: rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[64]: Ridge(alpha=10)

```
In [65]: rr.score(x_test,y_test)
```

Out[65]: -0.0725587790137805

```
In [66]: la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[66]: Lasso(alpha=10)

```
In [67]: la.score(x_test,y_test)
```

Out[67]: -0.06697047751139307

In []: