Importing libraries

In [53]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
import matplotlib.pyplot as plt
```

Exploratory Data Analysis

In [54]:
```
data=pd.read_csv(r"C:\Users\Kishore\OneDrive\Desktop\CSV Files\uber.csv")
```

In [55]:
```
data
```

Out[55]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude
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

In [56]:
```
list(data)
```

Out[56]:
```
['Unnamed: 0',
 'key',
 'fare_amount',
 'pickup_datetime',
 'pickup_longitude',
 'pickup_latitude',
 'dropoff_longitude',
 'dropoff_latitude',
 'passenger_count']
```

In [57]:
```
data.shape
```

Out[57]:
```
(200000, 9)
```

In [58]:
```
data.head(5)
```

Out[58]:

	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

In [59]:
```
data.tail(5)
```

Out[59]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	picku
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	

In [60]:
```
data.describe()
```

Out[60]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dr
count	2.000000e+05	200000.000000	200000.000000	200000.000000	199999.000000	1
mean	2.771250e+07	11.359955	-72.527638	39.935885	-72.525292	
std	1.601382e+07	9.901776	11.437787	7.720539	13.117408	
min	1.000000e+00	-52.000000	-1340.648410	-74.015515	-3356.666300	
25%	1.382535e+07	6.000000	-73.992065	40.734796	-73.991407	
50%	2.774550e+07	8.500000	-73.981823	40.752592	-73.980093	
75%	4.155530e+07	12.500000	-73.967154	40.767158	-73.963658	
max	5.542357e+07	499.000000	57.418457	1644.421482	1153.572603	

In [61]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 9 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   Unnamed: 0         200000 non-null  int64  
 1   key                200000 non-null  object 
 2   fare_amount        200000 non-null  float64
 3   pickup_datetime    200000 non-null  object 
 4   pickup_longitude   200000 non-null  float64
 5   pickup_latitude    200000 non-null  float64
 6   dropoff_longitude  199999 non-null  float64
 7   dropoff_latitude   199999 non-null  float64
 8   passenger_count    200000 non-null  int64  
dtypes: float64(5), int64(2), object(2)
memory usage: 13.7+ MB
```

In [62]:
```
data.min()
```

Out[62]:
```
Unnamed: 0                                  1
key                2009-01-01 01:15:22.0000006
fare_amount                              -52.0
pickup_datetime          2009-01-01 01:15:22 UTC
pickup_longitude                    -1340.64841
pickup_latitude                      -74.015515
dropoff_longitude                    -3356.6663
dropoff_latitude                     -881.985513
passenger_count                              0
dtype: object
```

In [63]:
```
data.max()
```

Out[63]:
```
Unnamed: 0                                 55423567
key                2015-06-30 23:40:39.0000001
fare_amount                              499.0
pickup_datetime          2015-06-30 23:40:39 UTC
pickup_longitude                      57.418457
pickup_latitude                     1644.421482
dropoff_longitude                   1153.572603
dropoff_latitude                     872.697628
passenger_count                             208
dtype: object
```

In [64]:
```
data.groupby('passenger_count').count()
```

Out[64]:

passenger_count	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	picku
0	709	709	709	709	709	
1	138425	138425	138425	138425	138425	
2	29428	29428	29428	29428	29428	
3	8881	8881	8881	8881	8881	
4	4276	4276	4276	4276	4276	
5	14009	14009	14009	14009	14009	
6	4271	4271	4271	4271	4271	
208	1	1	1	1	1	

Data Cleaning

In [65]:
```
data['pickup_datetime'] = pd.to_datetime(data['pickup_datetime'])
```

In [66]:
```
data['year'] = data['pickup_datetime'].dt.year
data['date'] = data['pickup_datetime'].dt.date
data['time'] = data['pickup_datetime'].dt.time
data
```

Out[66]:

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

200000 rows × 12 columns

In [67]:
```
print(data[['pickup_datetime', 'year', 'date', 'time']].head().reset_index(
```

```
   index           pickup_datetime  year        date      time
0      0 2015-05-07 19:52:06+00:00  2015  2015-05-07  19:52:06
1      1 2009-07-17 20:04:56+00:00  2009  2009-07-17  20:04:56
2      2 2009-08-24 21:45:00+00:00  2009  2009-08-24  21:45:00
3      3 2009-06-26 08:22:21+00:00  2009  2009-06-26  08:22:21
4      4 2014-08-28 17:47:00+00:00  2014  2014-08-28  17:47:00
```

Grouping the data

In [68]:
```
data['year'] = pd.to_datetime(data['date']).dt.year
result = data.groupby('year')['passenger_count'].sum().reset_index()
result
```

Out[68]:

	year	passenger_count
0	2009	51398
1	2010	50849
2	2011	53079
3	2012	54156
4	2013	53343
5	2014	50923
6	2015	23159

In [69]:
```
data['month'] = pd.to_datetime(data['date']).dt.month
result = data.groupby('month')['passenger_count'].sum().reset_index()
result
```

Out[69]:

	month	passenger_count
0	1	29432
1	2	28028
2	3	31032
3	4	31061
4	5	31847
5	6	29959
6	7	25693
7	8	24314
8	9	25349
9	10	27492
10	11	25944
11	12	26756

In [70]:
```
data['date'] = pd.to_datetime(data['date']).dt.date
result = data.groupby('date')['passenger_count'].sum().reset_index()
result
```

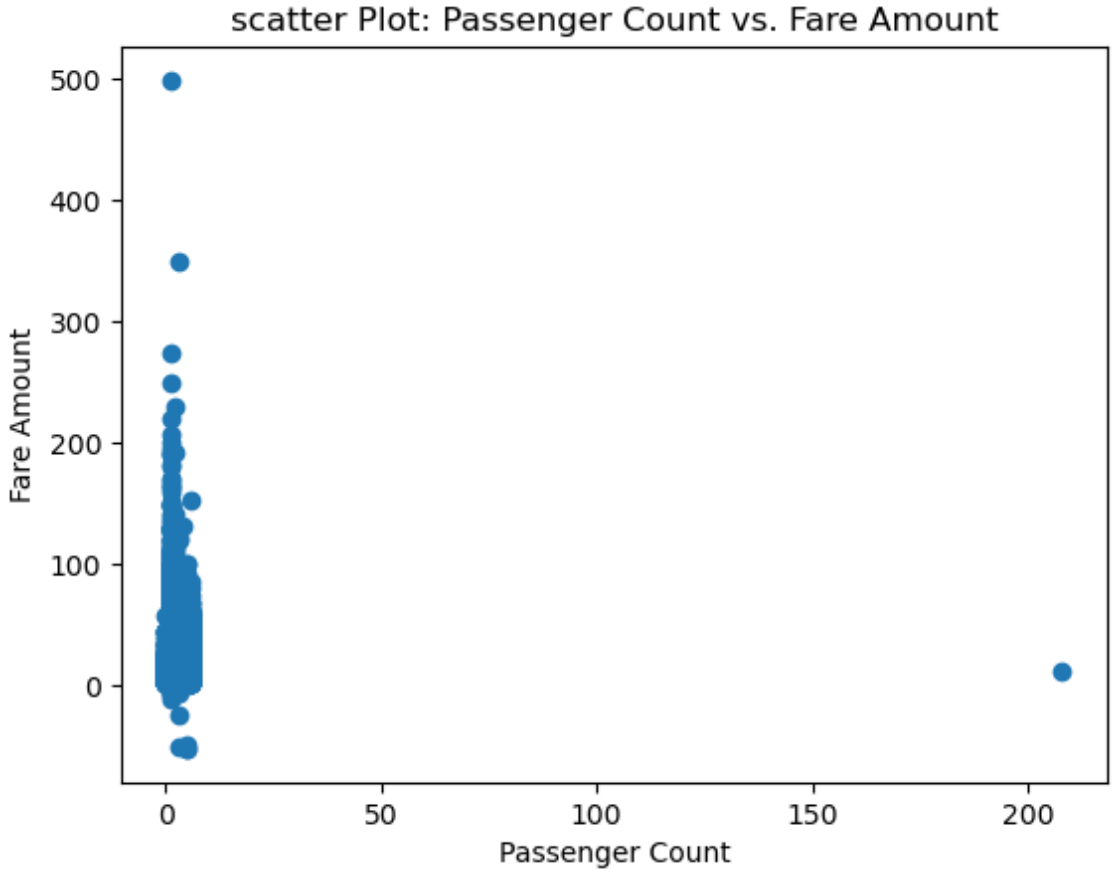
Out[70]:

	date	passenger_count
0	2009-01-01	113
1	2009-01-02	113
2	2009-01-03	147
3	2009-01-04	132
4	2009-01-05	109
...	...	...
2367	2015-06-26	145
2368	2015-06-27	133
2369	2015-06-28	123
2370	2015-06-29	99
2371	2015-06-30	103

2372 rows × 2 columns

Graphical Representation Using Scatter plot

In [71]:
```
plt.scatter(data['passenger_count'], data['fare_amount'])
plt.xlabel('Passenger Count')
plt.ylabel('Fare Amount')
plt.title('scatter Plot: Passenger Count vs. Fare Amount')
plt.show()
```



Dropping Unwanted Columns

In [75]:
```
data1=data.drop(['Unnamed: 0','key','pickup_datetime','pickup_longitude','p
data1
```

Out[75]:

	fare_amount	passenger_count	year	date	time	month
0	7.5	1	2015	2015-05-07	19:52:06	5
1	7.7	1	2009	2009-07-17	20:04:56	7
2	12.9	1	2009	2009-08-24	21:45:00	8
3	5.3	3	2009	2009-06-26	08:22:21	6
4	16.0	5	2014	2014-08-28	17:47:00	8
...	...	...	...	...	...	...
199995	3.0	1	2012	2012-10-28	10:49:00	10
199996	7.5	1	2014	2014-03-14	01:09:00	3
199997	30.9	2	2009	2009-06-29	00:42:00	6
199998	14.5	1	2015	2015-05-20	14:56:25	5
199999	14.1	1	2010	2010-05-15	04:08:00	5

200000 rows × 6 columns

Correlation Matrix for Data Set

In [81]:
```
data_numeric = data.select_dtypes(include='number')
cor_mat = data_numeric.corr()
```

In [88]:
```
cor_mat
```

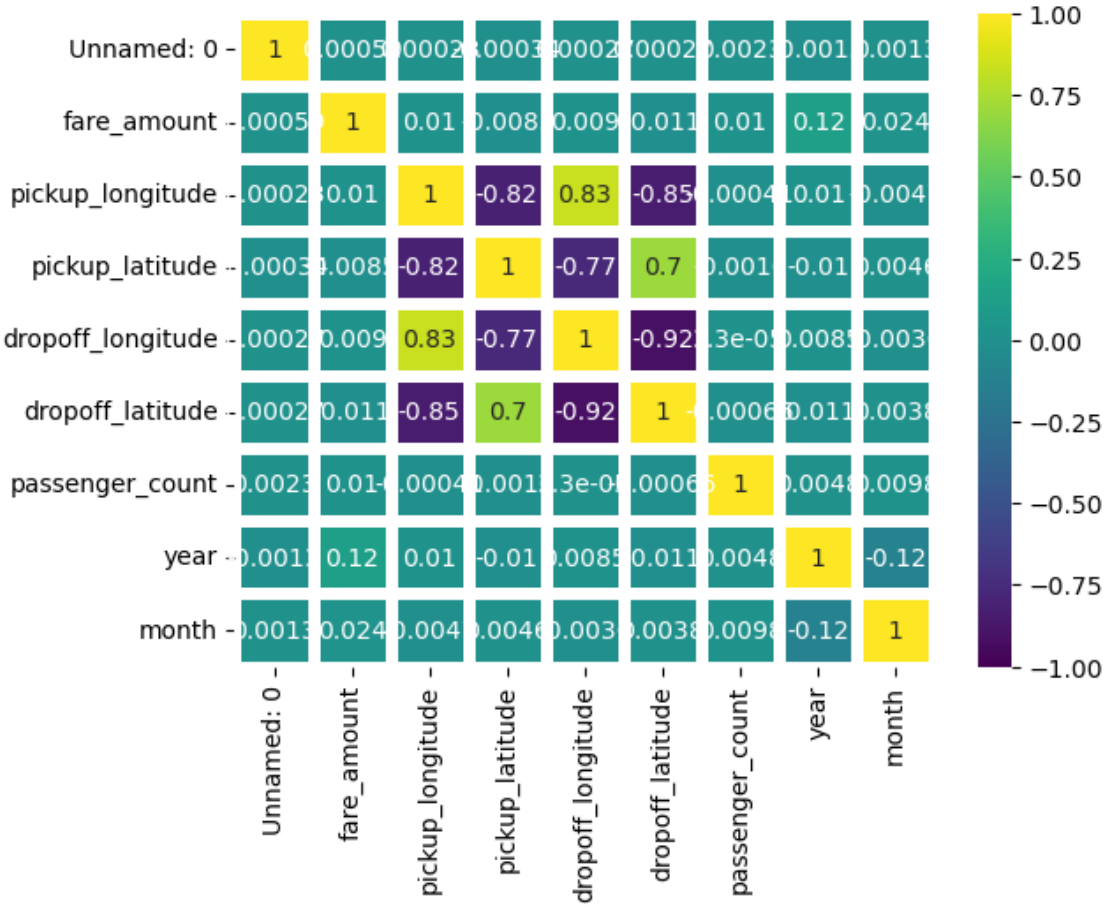
Out[88]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitu
Unnamed: 0	1.000000	0.000589	0.000230	-0.000341	0.0002
fare_amount	0.000589	1.000000	0.010457	-0.008481	0.0089
pickup_longitude	0.000230	0.010457	1.000000	-0.816461	0.8330
pickup_latitude	-0.000341	-0.008481	-0.816461	1.000000	-0.7747
dropoff_longitude	0.000270	0.008986	0.833026	-0.774787	1.0000
dropoff_latitude	0.000271	-0.011014	-0.846324	0.702367	-0.9170
passenger_count	0.002257	0.010150	-0.000414	-0.001560	0.0000
year	-0.001324	0.118335	0.009966	-0.010233	0.0084
month	0.001299	0.023814	-0.004665	0.004625	-0.0036

HeatMap for Data

In [84]:
```
import seaborn as sns
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='viridis')
```

Out[84]: <Axes: >



Correlation Matrix for Data1

In [83]:
```
data_numeric = data1.select_dtypes(include='number')
cor_mat1 = data_numeric.corr()
cor_mat1
```

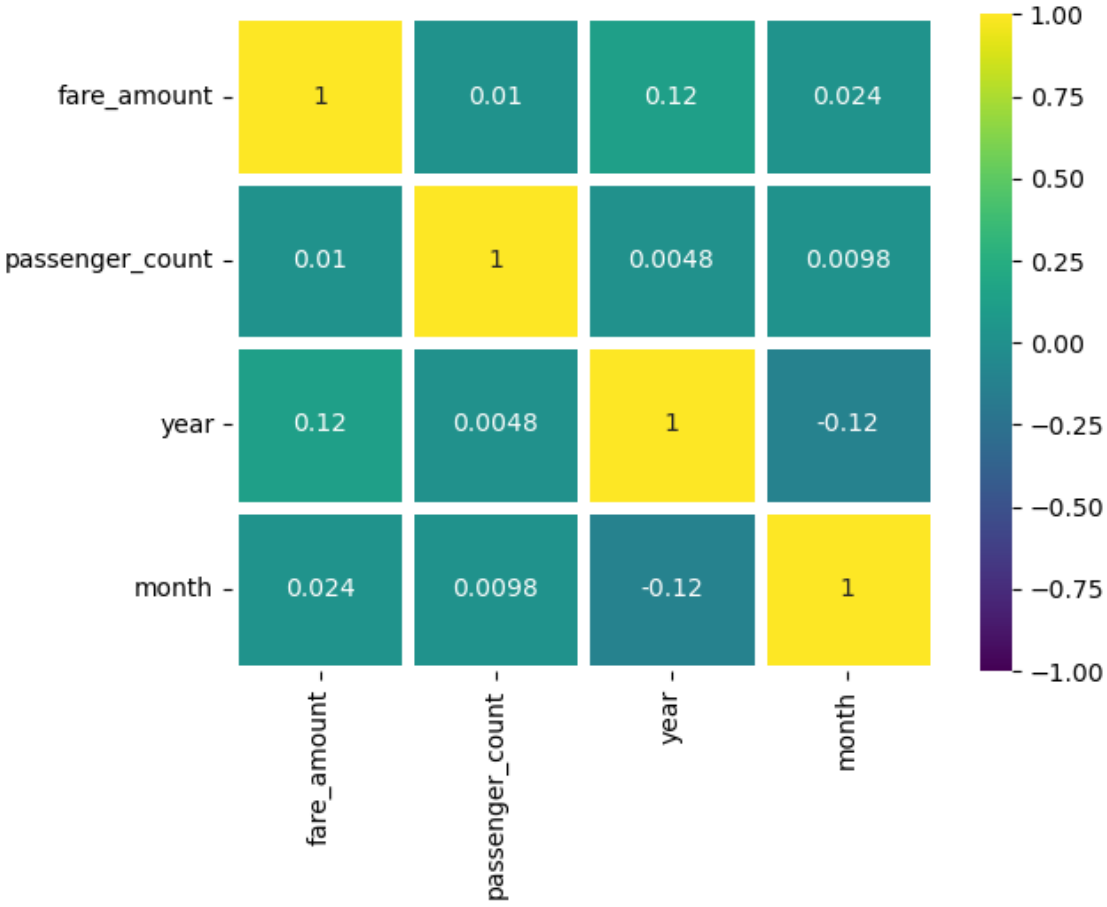
Out[83]:

	fare_amount	passenger_count	year	month
fare_amount	1.000000	0.010150	0.118335	0.023814
passenger_count	0.010150	1.000000	0.004798	0.009773
year	0.118335	0.004798	1.000000	-0.115859
month	0.023814	0.009773	-0.115859	1.000000

Heat Map for Data1

In [85]:
```
import seaborn as sns
sns.heatmap(cor_mat1,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='viridis')
```

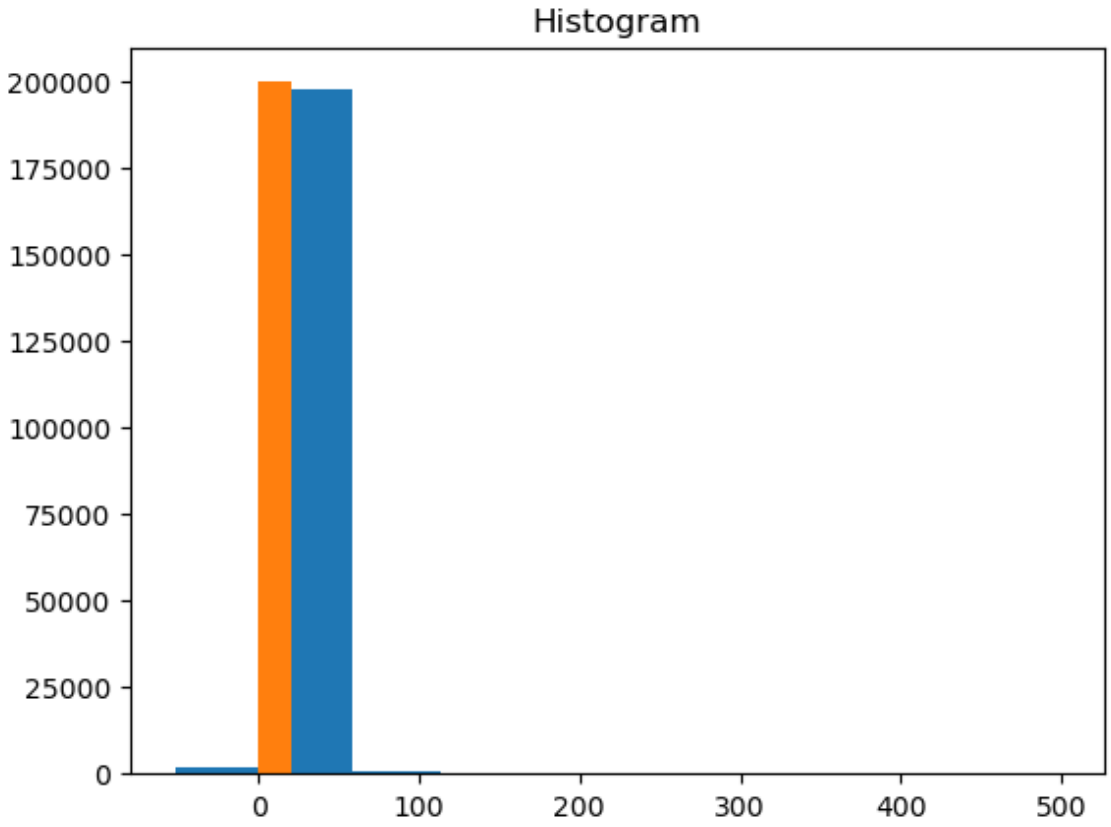
Out[85]: <Axes: >



In [89]:
```
## A Sample Histogram Representation
```

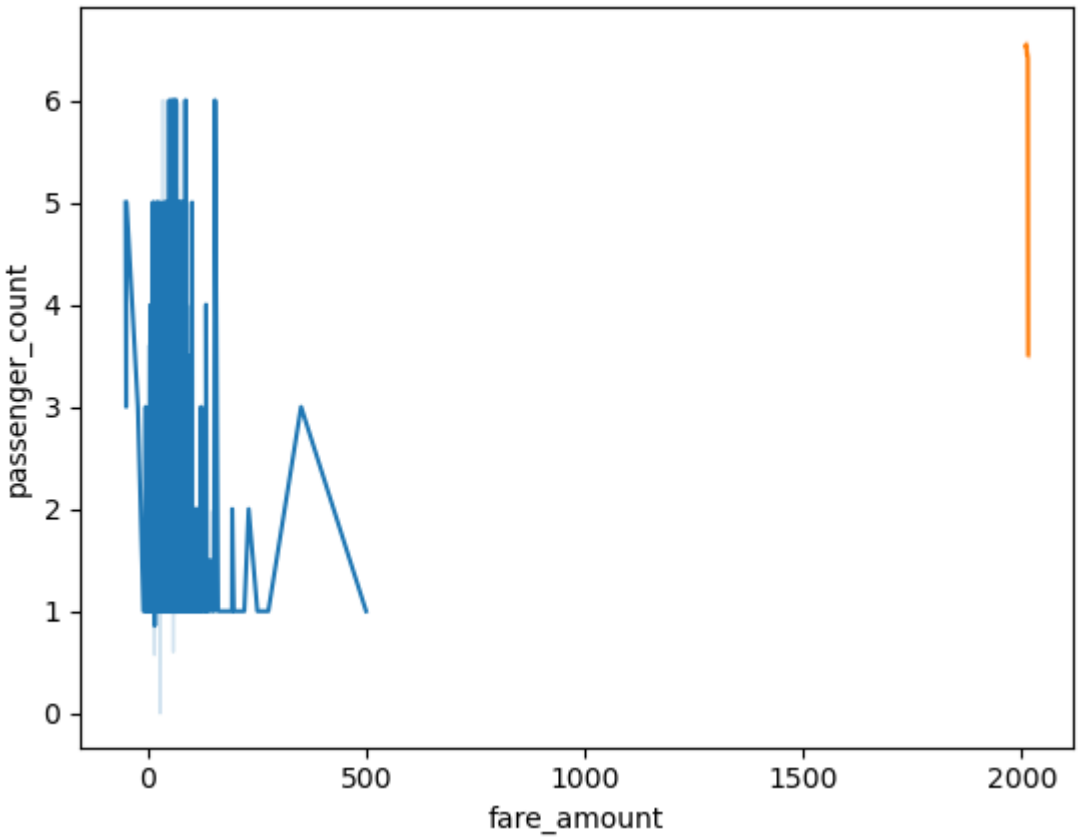
In [86]:
```
plt.hist(data1['fare_amount'])
plt.hist(data1['passenger_count'])

plt.title('Histogram')
plt.show()
```



In [87]:
```
sns.lineplot(x='fare_amount',y='passenger_count',data=data)
sns.lineplot(x='year',y='month',data=data)
```

Out[87]:
```
<Axes: xlabel='fare_amount', ylabel='passenger_count'>
```



In [ ]:
```

```