In [1]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv(r"C:\Users\NARESH SANA\Downloads\uber.csv")
```

In [3]:
```
data
```

Out[3]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropo
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06 UTC	-73.999817	40.738354	-73.999512	
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56 UTC	-73.994355	40.728225	-73.994710	
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00 UTC	-74.005043	40.740770	-73.962565	
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21 UTC	-73.976124	40.790844	-73.965316	
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00 UTC	-73.925023	40.744085	-73.973082	
...	...	...	...	...	...	...	...	
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00 UTC	-73.987042	40.739367	-73.986525	
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00 UTC	-73.984722	40.736837	-74.006672	
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00 UTC	-73.986017	40.756487	-73.858957	
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25 UTC	-73.997124	40.725452	-73.983215	
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00 UTC	-73.984395	40.720077	-73.985508	

200000 rows × 9 columns

In [4]:
```
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 [5]:
```
data.isnull().sum()
```

Out[5]:
```
Unnamed: 0           0
key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    1
dropoff_latitude     1
passenger_count      0
dtype: int64
```

In [6]:
```
data['dropoff_longitude'].mean()
```

Out[6]:
```
-72.52529162747415
```

In [7]:
```
data['dropoff_longitude'].median()
```

Out[7]:
```
-73.98009300000001
```

In [8]:
```
data['dropoff_latitude'].mean()
```

Out[8]:
```
39.92389040183263
```

In [9]:
```
data['dropoff_latitude'].median()
```

Out[9]:
```
40.753042
```

In [10]:
```
data['pickup_datetime'] =  pd.to_datetime(data['pickup_datetime'], format='%Y-%m-%d %H:%M:%S UTC')
```

In [11]:
```
data.columns
```

Out[11]:
```
Index(['Unnamed: 0', 'key', 'fare_amount', 'pickup_datetime',
       'pickup_longitude', 'pickup_latitude', 'dropoff_longitude',
       'dropoff_latitude', 'passenger_count'],
      dtype='object')
```

In [12]:
```
data['year'] = data['pickup_datetime'].dt.year
data['date'] = data['pickup_datetime'].dt.date
data['time'] = data['pickup_datetime'].dt.time
```

In [13]:
```
print(data[['pickup_datetime', 'year', 'date', 'time']].head())
```

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

In [14]:
```
data1=data.drop(['Unnamed: 0','key','pickup_longitude','dropoff_longitude','pickup_latitude','dropoff_la
```

In [15]:
```
data1
```

Out[15]:

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

200000 rows × 6 columns

In [16]:
```
data['year'] = pd.to_datetime(data['date']).dt.year
result = data.groupby('year')['passenger_count'].sum().reset_index()
result
```

Out[16]:

	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 [17]:
```
data['month'] = pd.to_datetime(data['date']).dt.month
result = data.groupby('month')['passenger_count'].sum().reset_index()
result
```

Out[17]:

	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 [18]:
```
data['dropoff_longitude'].fillna(data['dropoff_longitude'].mean(),inplace=True)
data['dropoff_latitude'].fillna(data['dropoff_latitude'].median(),inplace=True)
```

In [19]:
```
data
```

Out[19]:

	Unnamed: 0	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropo
0	24238194	2015-05-07 19:52:06.0000003	7.5	2015-05-07 19:52:06	-73.999817	40.738354	-73.999512	
1	27835199	2009-07-17 20:04:56.0000002	7.7	2009-07-17 20:04:56	-73.994355	40.728225	-73.994710	
2	44984355	2009-08-24 21:45:00.00000061	12.9	2009-08-24 21:45:00	-74.005043	40.740770	-73.962565	
3	25894730	2009-06-26 08:22:21.0000001	5.3	2009-06-26 08:22:21	-73.976124	40.790844	-73.965316	
4	17610152	2014-08-28 17:47:00.000000188	16.0	2014-08-28 17:47:00	-73.925023	40.744085	-73.973082	
...	...	...	...	...	...	...	...	
199995	42598914	2012-10-28 10:49:00.00000053	3.0	2012-10-28 10:49:00	-73.987042	40.739367	-73.986525	
199996	16382965	2014-03-14 01:09:00.0000008	7.5	2014-03-14 01:09:00	-73.984722	40.736837	-74.006672	
199997	27804658	2009-06-29 00:42:00.00000078	30.9	2009-06-29 00:42:00	-73.986017	40.756487	-73.858957	
199998	20259894	2015-05-20 14:56:25.0000004	14.5	2015-05-20 14:56:25	-73.997124	40.725452	-73.983215	
199999	11951496	2010-05-15 04:08:00.00000076	14.1	2010-05-15 04:08:00	-73.984395	40.720077	-73.985508	

200000 rows × 13 columns

In [20]:
```
data.isnull().sum()
```

Out[20]:
```
Unnamed: 0           0
key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    0
dropoff_latitude     0
passenger_count      0
year                 0
date                 0
time                 0
month                0
dtype: int64
```

In [21]:
```
data_numeric = data.select_dtypes(include='number')
cor_mat = data_numeric.corr()
```

In [22]:
```
data['key'] = pd.to_numeric(data['key'], errors='coerce')
```

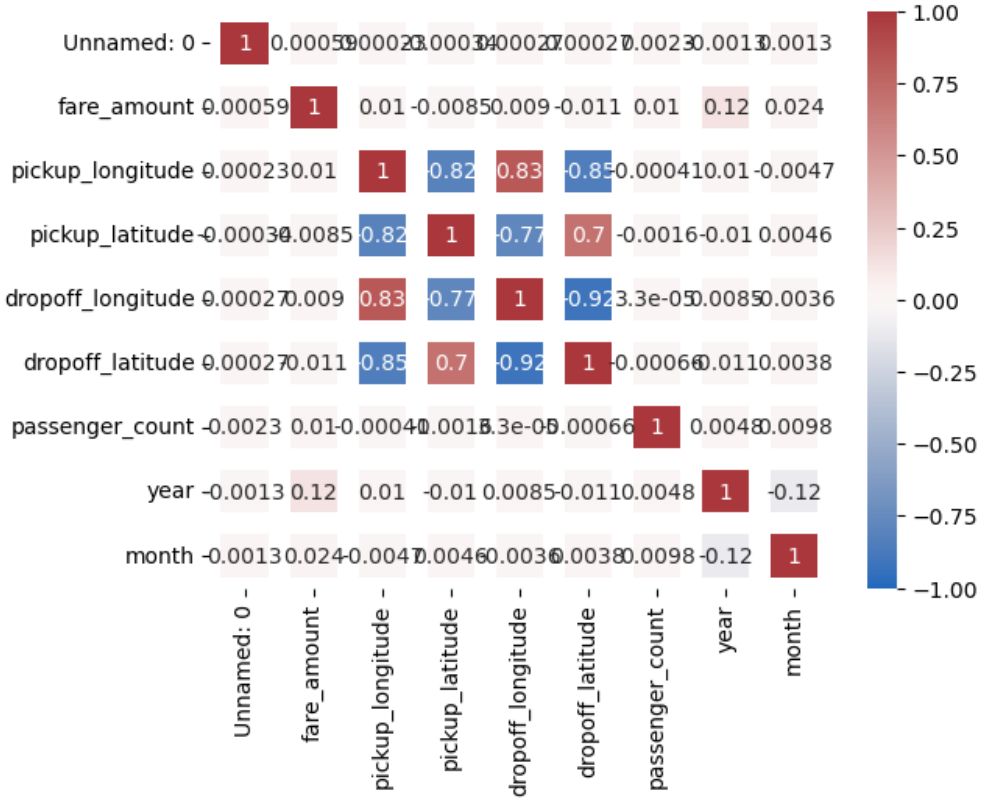
In [23]:
```
cor_mat
```

Out[23]:

	Unnamed: 0	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
Unnamed: 0	1.000000	0.000589	0.000230	-0.000341	0.000270	0.000272	0.002257
fare_amount	0.000589	1.000000	0.010457	-0.008481	0.008986	-0.011013	0.010150
pickup_longitude	0.000230	0.010457	1.000000	-0.816461	0.833026	-0.846324	-0.000414
pickup_latitude	-0.000341	-0.008481	-0.816461	1.000000	-0.774787	0.702367	-0.001560
dropoff_longitude	0.000270	0.008986	0.833026	-0.774787	1.000000	-0.917010	0.000033
dropoff_latitude	0.000272	-0.011013	-0.846324	0.702367	-0.917010	1.000000	-0.000660
passenger_count	0.002257	0.010150	-0.000414	-0.001560	0.000033	-0.000660	1.000000
year	-0.001324	0.118335	0.009966	-0.010233	0.008467	-0.011239	0.004798
month	0.001299	0.023814	-0.004665	0.004625	-0.003605	0.003818	0.009773

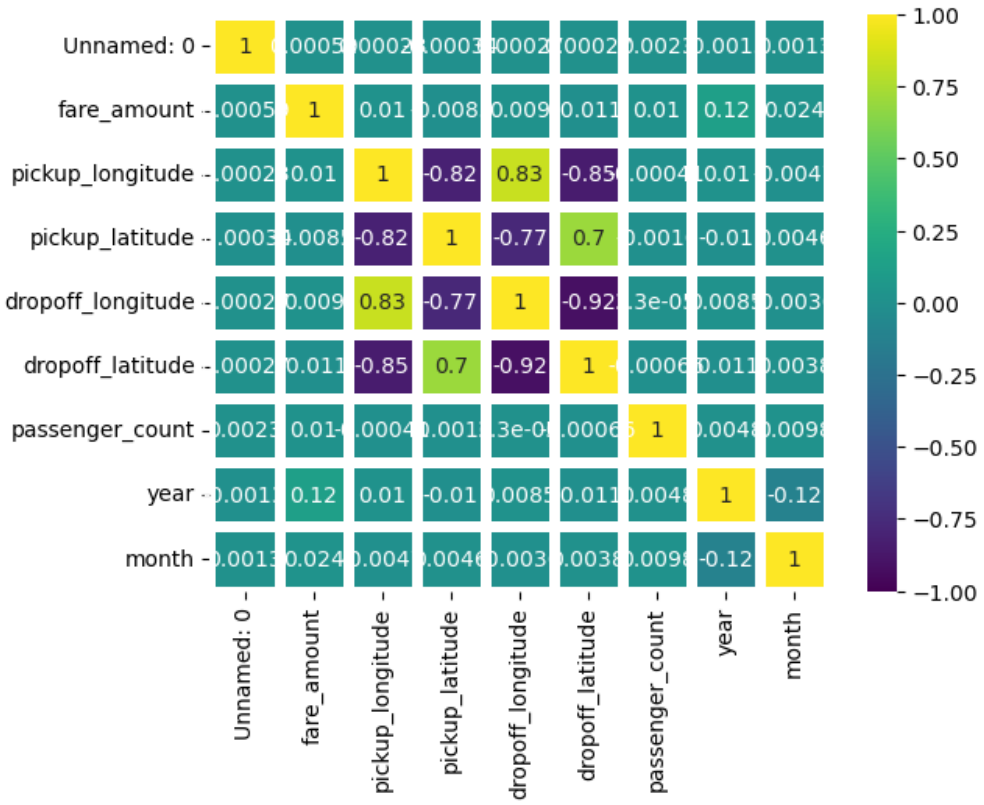
In [24]:
```
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=10,cmap='vlag')#vlag,icefire,coolwarm,bwr,seismi
```

Out[24]: <Axes: >

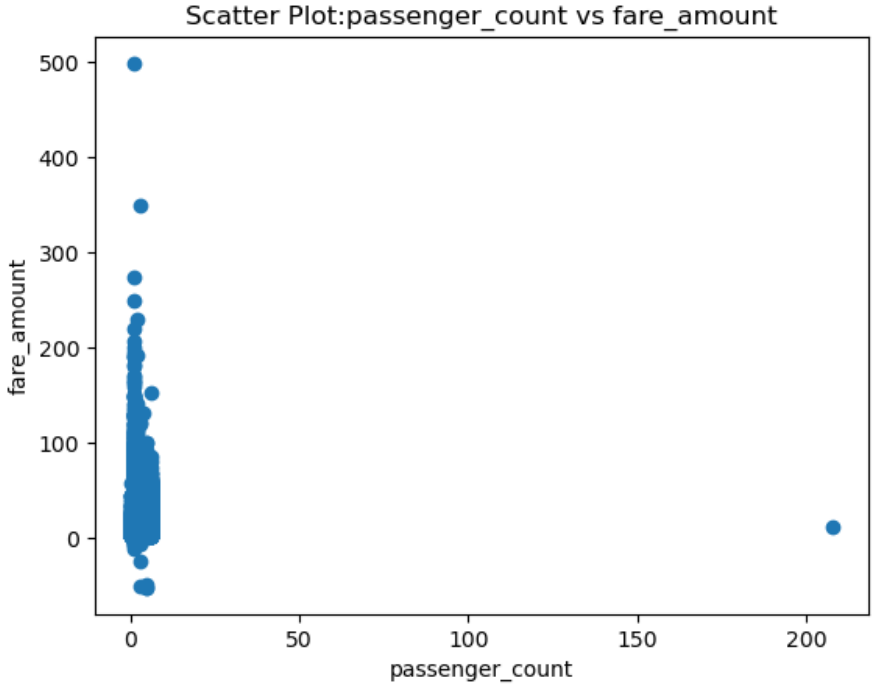


In [33]:
```
import seaborn as sns
sns.heatmap(cor_mat,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='viridis')
```

Out[33]: <Axes: >

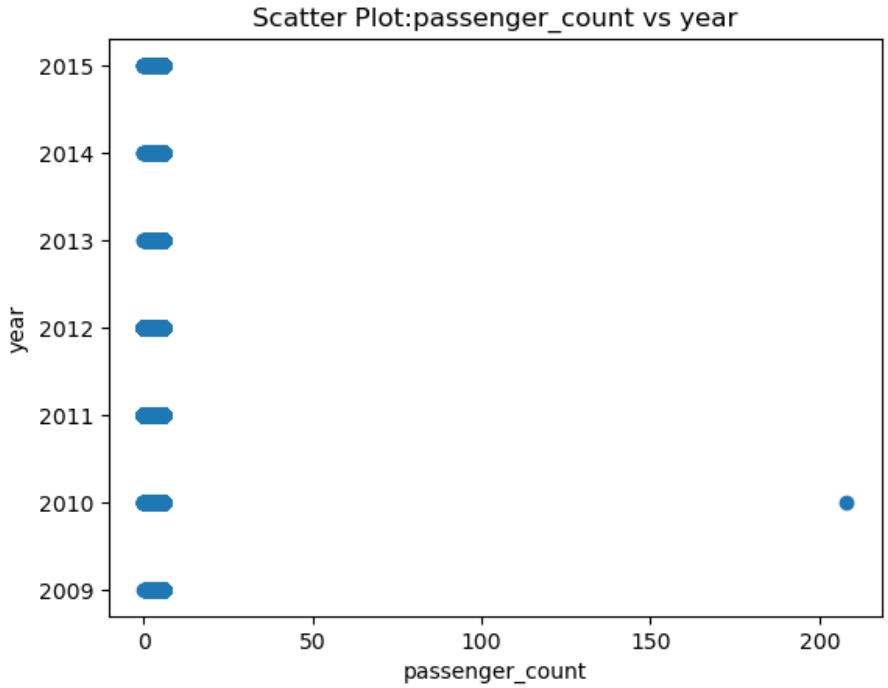


In [29]:
```
plt.scatter(data['passenger_count'],data['fare_amount'])
plt.xlabel('passenger_count')
plt.ylabel('fare_amount')
plt.title('Scatter Plot:passenger_count vs fare_amount')
plt.show()
```



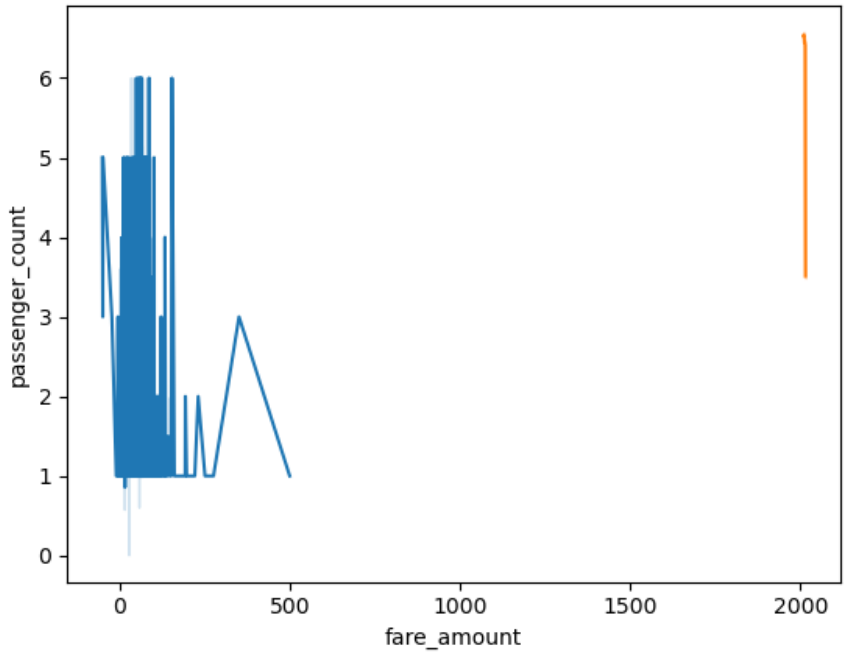
In [ ]:
```
import seaborn as sns
sns.heatmap(cor_mat1,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='viridis')
```

In [30]:
```
plt.scatter(data['passenger_count'],data['year'])
plt.xlabel('passenger_count')
plt.ylabel('year')
plt.title('Scatter Plot:passenger_count vs year')
plt.show()
```



In [27]:
```
sns.lineplot(x='fare_amount',y='passenger_count',data=data)
sns.lineplot(x='year',y='month',data=data)
```

Out[27]: <Axes: xlabel='fare_amount', ylabel='passenger_count'>



In [28]:
```
data1.to_csv('new_uber.csv')
```

In [ ]: