Problem Definition : Predict the total ride duration of taxi trips in New York City.

Hypothesis Generated:

Trip duration will be affected by following -

1. Pickup datetime of trips
2. Pickup and Drop location
3. Number of passengers
4. Traffic condition
5. Weather Condition
6. Mechanical issue

Data Loading After Extraction:

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline
import warnings
warnings.filterwarnings(action = 'ignore')
import datetime as dt
```

In [2]:
```
data = pd.read_csv('nyc_taxi_trip_duration.csv')
```

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

Out[3]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_lati
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01	1	-73.953918	40.778
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57	2	-73.988312	40.731
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48	2	-73.997314	40.721
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32	6	-73.961670	40.759
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31	1	-74.017120	40.708

Variable Identification and Typecasting:

In [4]:
```
data.dtypes
```

Out[4]:
```
id                     object
vendor_id               int64
pickup_datetime        object
dropoff_datetime       object
passenger_count         int64
pickup_longitude      float64
pickup_latitude       float64
dropoff_longitude     float64
dropoff_latitude      float64
store_and_fwd_flag     object
trip_duration           int64
dtype: object
```

Here we have only numerical data type that needs to be analysed and we will convert the pickup_datetime and dropoff_datetime to datetime module.

In [5]:
```
# converting strings to datetime features

data['pickup_datetime'] = pd.to_datetime(data.pickup_datetime)
data['dropoff_datetime'] = pd.to_datetime(data.dropoff_datetime)
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
id                    0
vendor_id             0
pickup_datetime       0
dropoff_datetime      0
passenger_count       0
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
store_and_fwd_flag    0
trip_duration         0
dtype: int64
```

There are no missing values, then we don't need to drop NaN values.

In [7]:
```
# Importing the library which lets us calculate distance from geographical coordinates
from geopy.distance import great_circle
```

In [8]:
```
# Defining a function to take coordinates as inputs and return us distance
def cal_distance(pickup_lat,pickup_long,dropoff_lat,dropoff_long):
    start_coordinates=(pickup_lat,pickup_long)
    stop_coordinates=(dropoff_lat,dropoff_long)
    return great_circle(start_coordinates,stop_coordinates).km
```

In [9]:
```
# Applying the function to our dataset and creating the feature ‘distance’.
data['distance'] = data.apply(lambda x: cal_distance(x['pickup_latitude'],x['pickup_long
                                                     x['dropoff_latitude'],x['dropoff_lo
```

In [10]:
```
data.head()
```
Out[10]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_lati
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01	1	-73.953918	40.778
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57	2	-73.988312	40.731
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48	2	-73.997314	40.721
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32	6	-73.961670	40.759
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31	1	-74.017120	40.708

Target Variable:

Let us analyse our target variable trip_duration.

In [11]:
```
# Trip duration in hours

data['trip_duration'].mean()/3600
```
Out[11]:
```
0.2645080925998545
```

In [12]:
```
data['trip_duration'].std()/3600
```
Out[12]:
```
1.0735072770225538
```

In [13]:
```
data['trip_duration'].min()/3600
```
Out[13]:
```
0.0002777777777777778
```

In [14]:
```
data['trip_duration'].max()/3600
```
Out[14]:
```
538.8155555555555
```

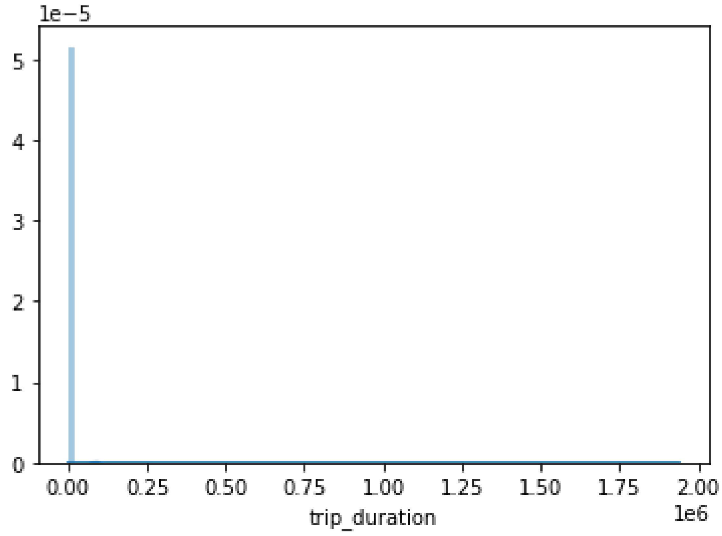
In [15]:
```
data['trip_duration'].median()/3600
```
Out[15]:
```
0.18416666666666667
```

Or we can use .describe() function

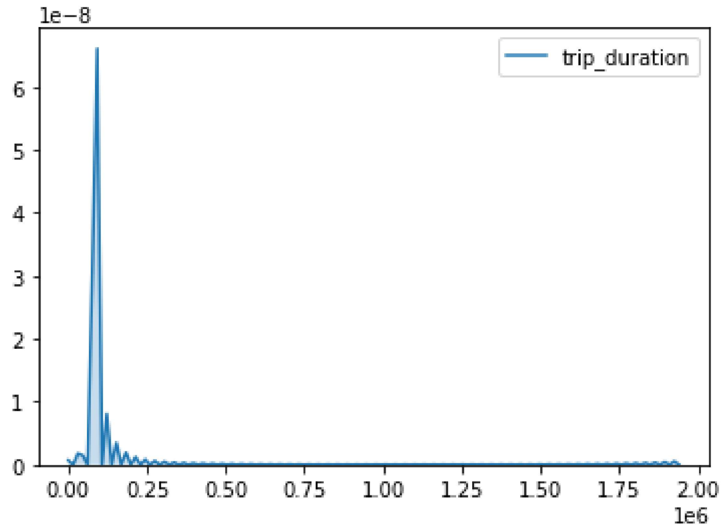
In [16]:
```
data['trip_duration'].describe()/3600
```

Out[16]:
```
count    202.589444
mean       0.264508
std        1.073507
min        0.000278
25%        0.110278
50%        0.184167
75%        0.298611
max      538.815556
Name: trip_duration, dtype: float64
```

In [17]:
```
sns.distplot(data['trip_duration'], bins = 100)
plt.show()
```

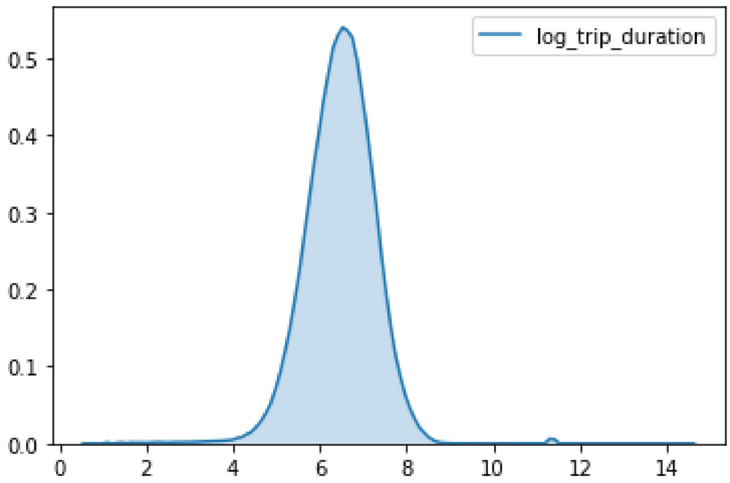


In [18]:
```
sns.kdeplot(data['trip_duration'], shade = True)
plt.show()
```



Since there is a huge outlier, we will take log of the trip_duration for visualising it better.

In [19]:
```
data['log_trip_duration'] = np.log(data['trip_duration'].values + 1)
sns.kdeplot(data['log_trip_duration'], shade = True)
plt.show()
```



1. The trip duration of rides are forming almost normal curve.
2. As noticed earlier, there is an outlier present near 12.
3. Also there are very short rides present which are of less than 10 seconds, which are suspicious.

Univariate Analysis:

We will check our hypothesis using univariate analysis of variables.

In [20]:
```
data['distance'].value_counts()
```

Out[20]:
```
0.000000    2901
0.000424      20
0.000424      19
0.000424      16
0.000424      11
            ...
2.929161       1
0.977650       1
0.925223       1
4.112012       1
5.945846       1
Name: distance, Length: 726243, dtype: int64
```

We see there are 2893 trips with 0 km distance.

Possible reasons for 0 km distance can be:

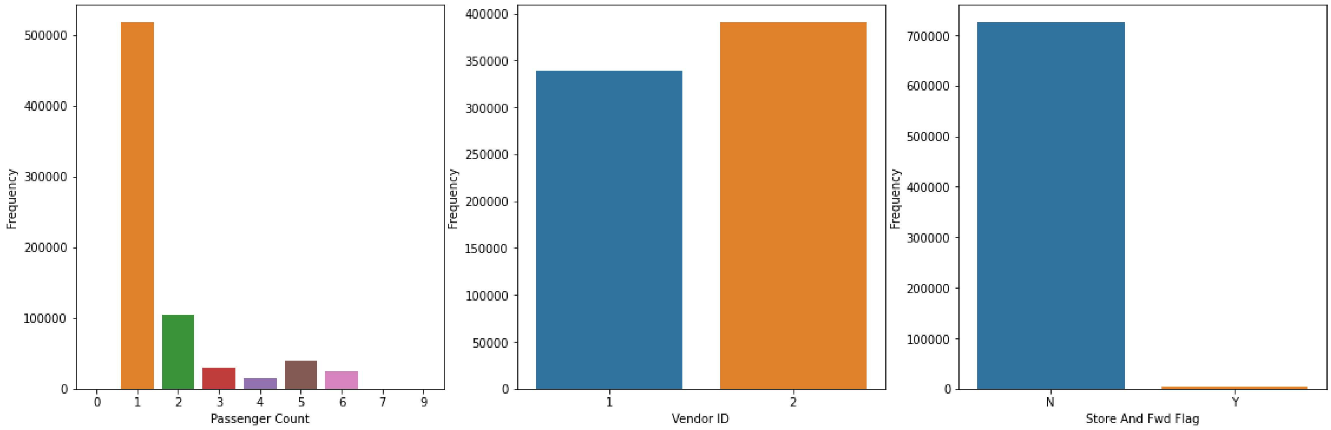
- The dropoff location couldn't be tracked.
- The driver deliberately took this ride to complete a target ride number.
- The passengers canceled the trip.

In [21]:
```
plt.figure(figsize=(19, 6))

plt.subplot(1, 3, 1)
sns.countplot(data['passenger_count'])
plt.xlabel('Passenger Count')
plt.ylabel('Frequency')

plt.subplot(1, 3, 2)
sns.countplot(data['vendor_id'])
plt.xlabel('Vendor ID')
plt.ylabel('Frequency')

plt.subplot(1, 3, 3)
sns.countplot(data['store_and_fwd_flag'])
plt.xlabel('Store And Fwd Flag')
plt.ylabel('Frequency')
plt.show()
```



Observation -

1. Most frequent rides include only 1 passenger, while some of the rides include 7 to 9 passengers too and they are very low in number.
2. Most of the rides have been completed by vendor 2 as compared to vendor 1.
3. There is almost no storing of data taking place in the taxi and being updated later. (Y - Yes, N - No)

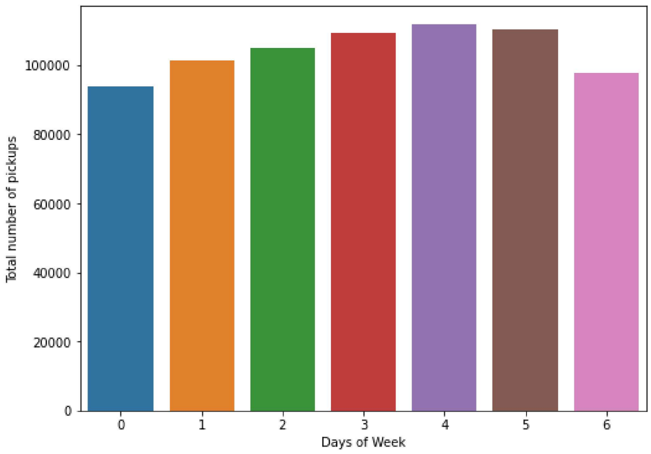
Observing trend in pickup datetime of trips

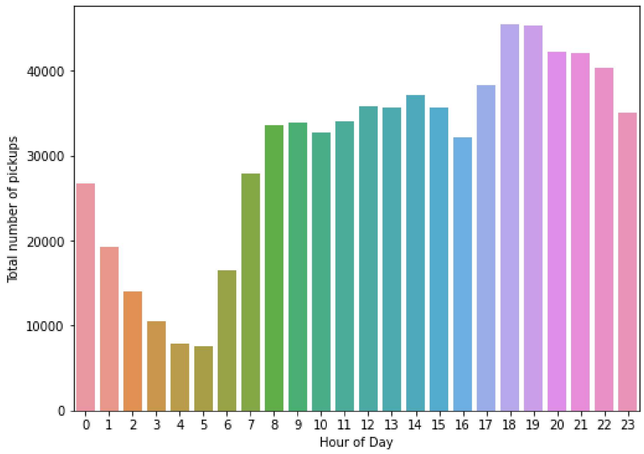
In [22]:
```
data['day_of_week'] = data['pickup_datetime'].dt.weekday
data['hour_of_day'] = data['pickup_datetime'].dt.hour
```

In [23]:
```
plt.figure(figsize=(18, 6))

plt.subplot(1, 2, 1)
sns.countplot(data['day_of_week'])
plt.xlabel('Days of Week')
plt.ylabel('Total number of pickups')

plt.subplot(1, 2, 2)
sns.countplot(data['hour_of_day'])
plt.xlabel('Hour of Day')
plt.ylabel('Total number of pickups')
plt.show()
```



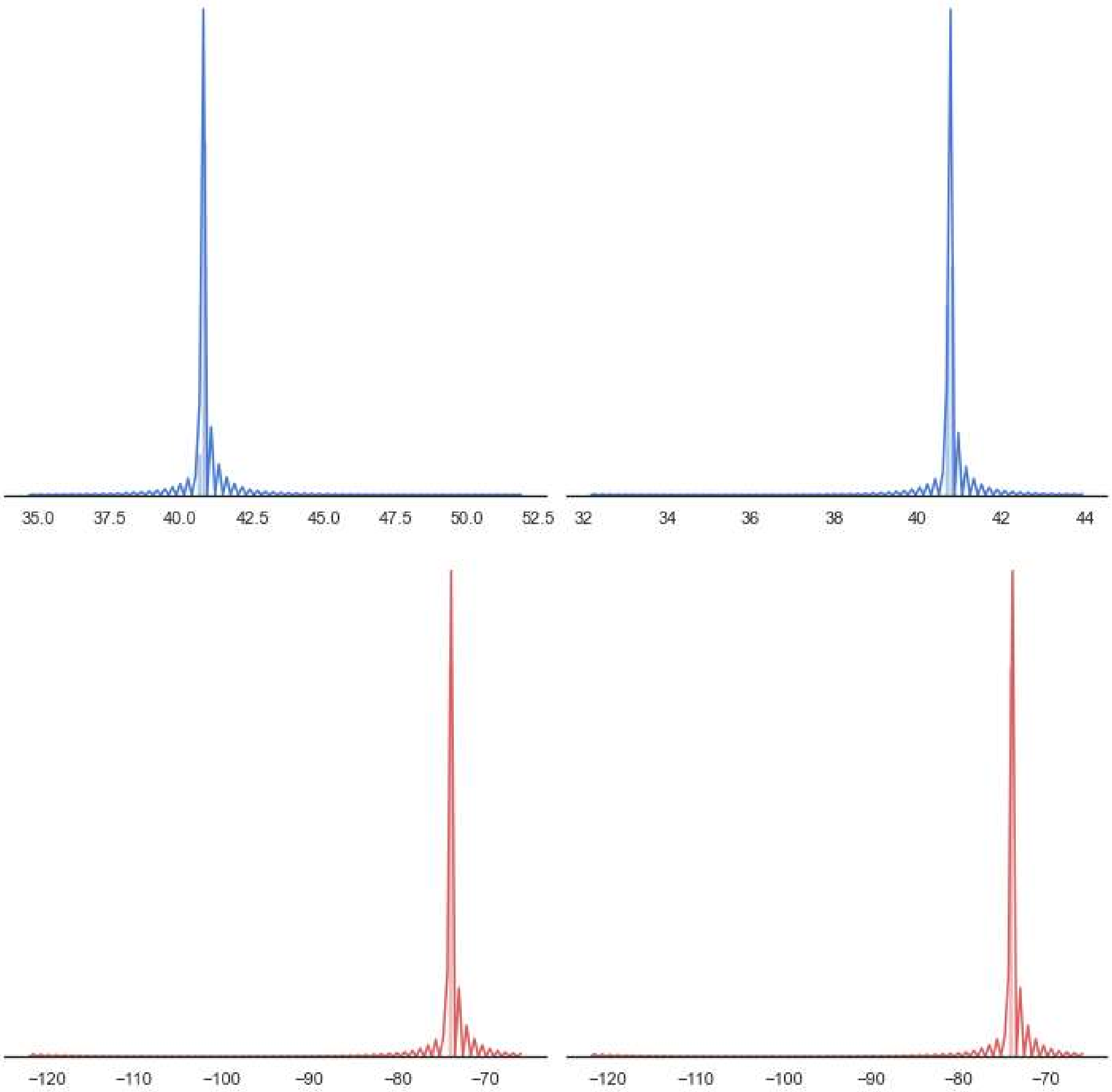


Observation -

1. Observing the above trend, we can see that the most of the rides are on Thursday, while on the weekends, there is lowest number of rides (0 is Sunday).
2. Total number of rides in 24 hours are mostly around 18-19 hours, i.e. evening. While in the morning peak hour, it is lower than expected.

Observing location of pickup and dropoff

In [24]:
```
sns.set(style="white", palette="muted", color_codes=True)
f, axes = plt.subplots(nrows = 2, ncols = 2, figsize=(10, 10), sharex = False, sharey =
sns.despine(left=True)
sns.distplot(data['pickup_latitude'].values, label = 'pickup_latitude', color="b", bins
sns.distplot(data['pickup_longitude'].values, label = 'pickup_longitude', color="r", bi
sns.distplot(data['dropoff_latitude'].values, label = 'dropoff_latitude', color="b", bi
sns.distplot(data['dropoff_longitude'].values, label = 'dropoff_longitude', color="r",
plt.setp(axes, yticks=[])
plt.tight_layout()
plt.show()
```



- Latitude - Blue
- Longitude - Red

Observation -

1. Pickup and drop latitudes are denser around 40 to 41, and longitude are denser around -74 to -73.
2. Extreme values are present in the data which depicts higher value of distance.

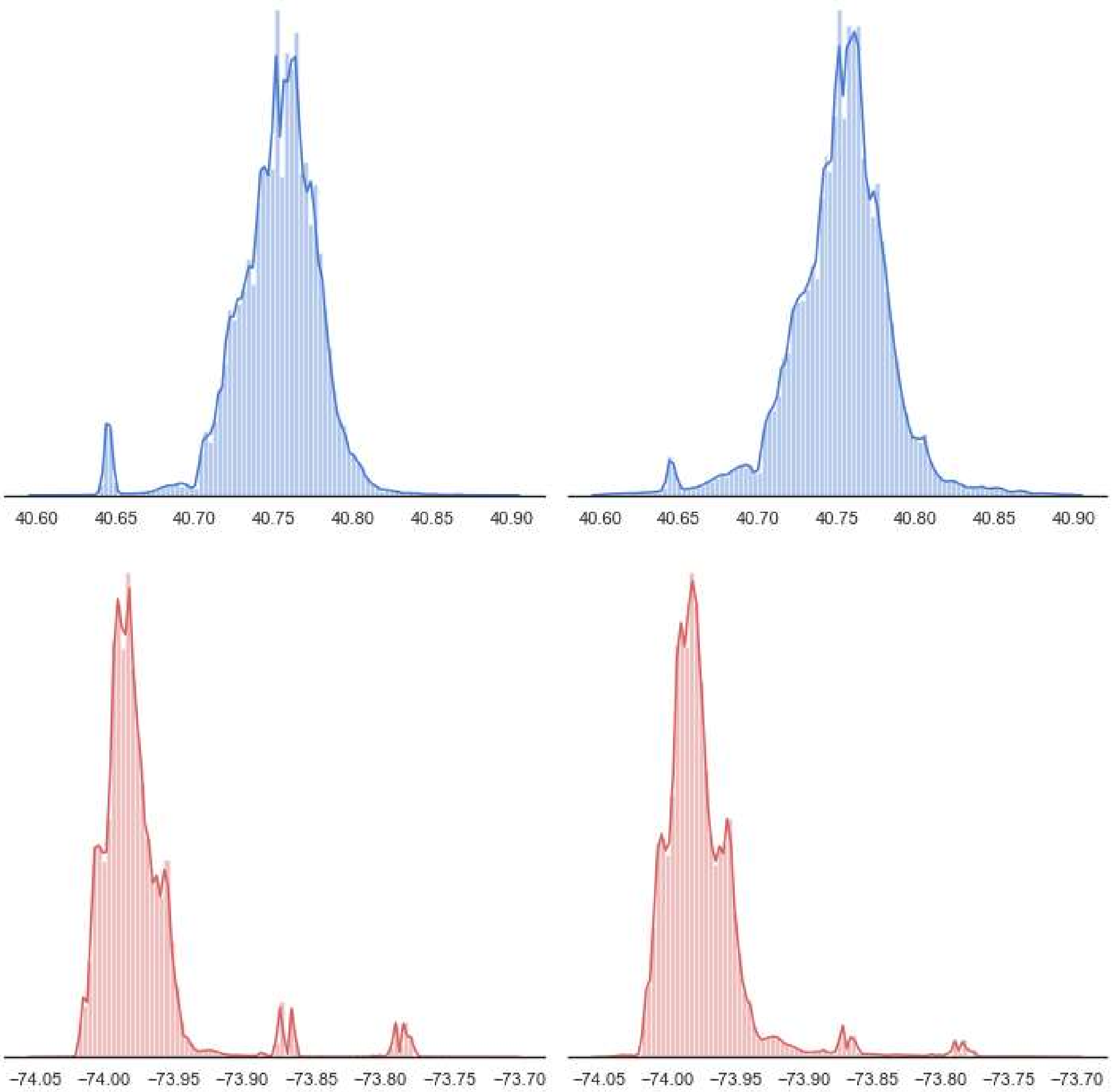
We will remove these outliers or extreme values and observe the data closely.

In [25]:
```
# Removal of outliers.

data = data.loc[(data.pickup_latitude > 40.6) & (data.pickup_latitude < 40.9)]
data = data.loc[(data.dropoff_latitude>40.6) & (data.dropoff_latitude < 40.9)]
data = data.loc[(data.dropoff_longitude > -74.05) & (data.dropoff_longitude < -73.7)]
data = data.loc[(data.pickup_longitude > -74.05) & (data.pickup_longitude < -73.7)]
data_new = data.copy()
```

In [26]:
```
# Visualisation after removing outliers

sns.set(style="white", palette="muted", color_codes=True)
f, axes = plt.subplots(2,2,figsize=(10, 10), sharex=False, sharey = False)
sns.despine(left=True)
sns.distplot(data_new['pickup_latitude'].values, label = 'pickup_latitude',color="b",bi
sns.distplot(data_new['pickup_longitude'].values, label = 'pickup_longitude',color="r",
sns.distplot(data_new['dropoff_latitude'].values, label = 'dropoff_latitude',color="b",
sns.distplot(data_new['dropoff_longitude'].values, label = 'dropoff_longitude',color="r'
plt.setp(axes, yticks=[])
plt.tight_layout()
plt.show()
```



Observation -

- Most of the rides are located between these locations, apart from few outliers outside the above range.

Bivariate Analysis:

We will compare each of the variables with the target variable, 'trip_duration', in order to derive the relation between them.

In [27]:
```
data.columns
```

Out[27]:
```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration', 'distance', 'log_trip_duration', 'day_of_week',
       'hour_of_day'],
      dtype='object')
```

In [28]:
```
data.head()
```

Out[28]:

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_lati
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01	1	-73.953918	40.778
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57	2	-73.988312	40.731
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48	2	-73.997314	40.721
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32	6	-73.961670	40.759
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31	1	-74.017120	40.708

Trip duration and Weekdays

Do the trips on weekdays have higher trip duration?

- We will use Time series plot, 'tsplot', to plot between datetime and a continuous variable.
- For plotting each day, we will take central tendency, i.e. median of each day's trip_duration and plot it against the days of week.

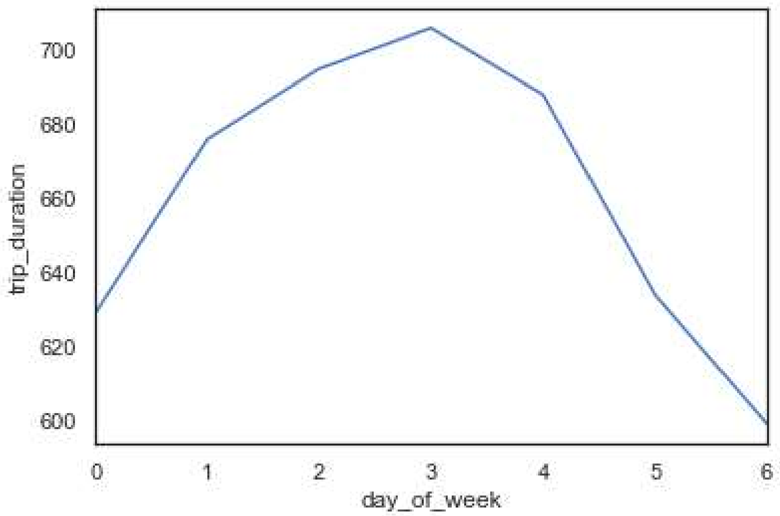
In [29]:
```
average_duration_day = pd.DataFrame(data.groupby(['day_of_week'])['trip_duration'].media
average_duration_day.reset_index(inplace = True)
average_duration_day['unit']=1
```

In [30]:
```
average_duration_day
```

Out[30]:

	day_of_week	trip_duration	unit
0	0	629.0	1
1	1	676.0	1
2	2	695.0	1
3	3	706.0	1
4	4	688.0	1
5	5	634.0	1
6	6	599.0	1

In [31]:
```
sns.tsplot(data=average_duration_day, time="day_of_week", unit = 'unit', value="trip_du
```

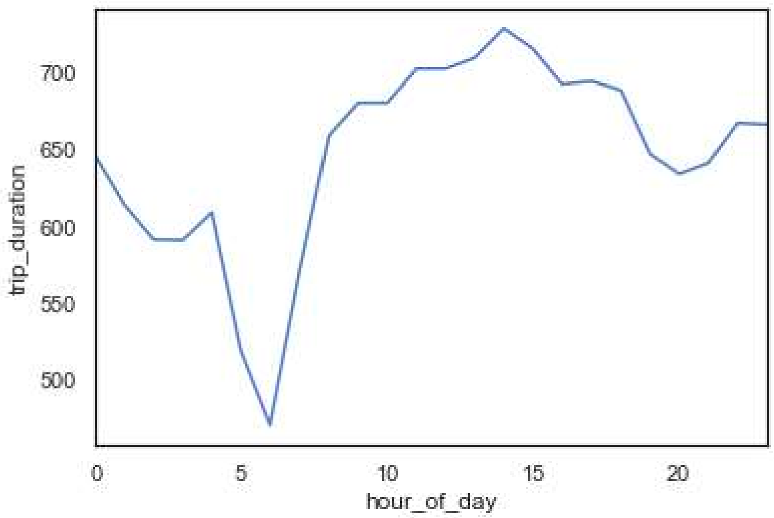


Observation -

1. Longest trip duration has been observed on Wednesday.
2. Opposite to expectation, trip duration on weekends are lowest.

In [32]:
```
average_duration_hour = pd.DataFrame(data.groupby(['hour_of_day'])['trip_duration'].med
average_duration_hour.reset_index(inplace = True)
average_duration_hour['unit']=1
```

In [33]:
```
sns.tsplot(data=average_duration_hour, time='hour_of_day', unit = 'unit', value='trip_du
plt.show()
```



Observation -

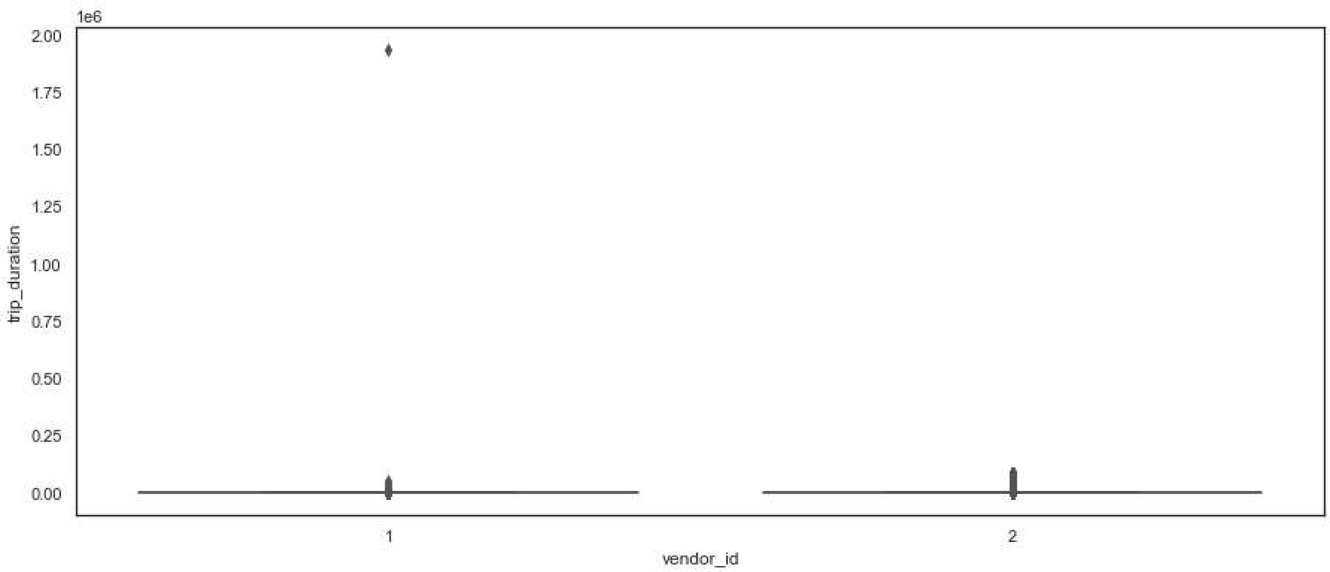
- Trip duration during early morning are comparatively lesser which may be because of low traffic, and highest during evening peak hour.
- There is a correlation between the number of pickups and trip duration as it follows the similar trend.

Trip Duration and Vendor ID

Do the vendors have any impact on the trip duration?

We will check the duration of trip for each vendor.

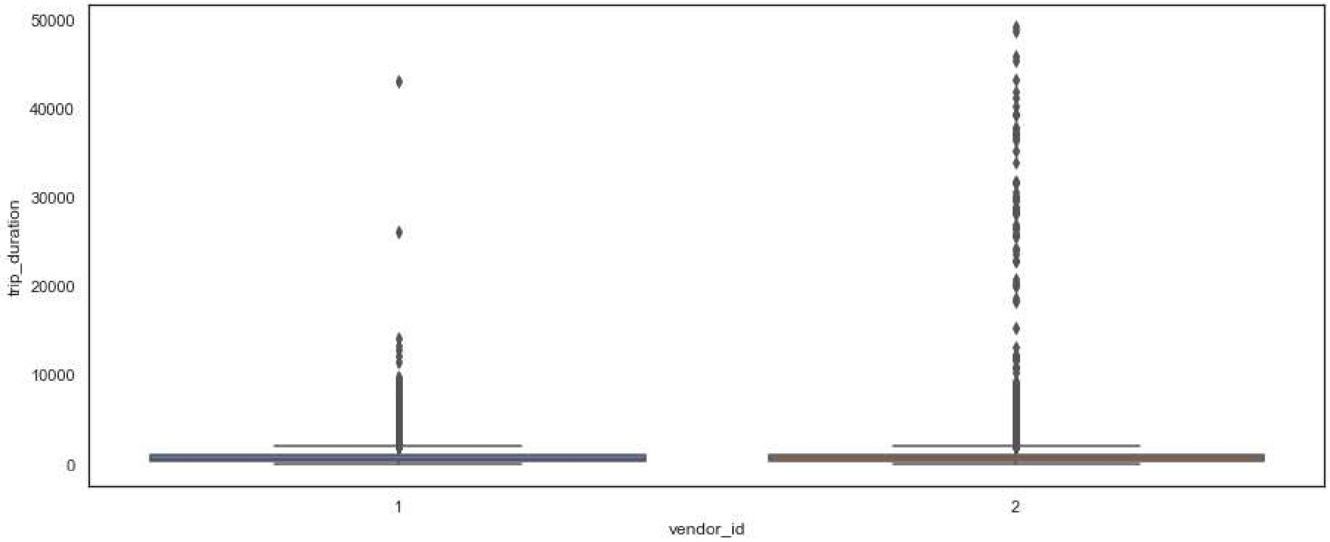
In [34]:
```
plt.figure(figsize=(15, 6))
sns.boxplot(x = 'vendor_id', y = 'trip_duration', data = data)
plt.show()
```



Observation -

- As we can see, there is a huge outlier/extreme point for vendor 1 as compared to vendor 2.
- Let's remove the outliers and observe the above data closely.

In [35]:
```
plt.figure(figsize=(15, 6))
trip_no_outliers = data[data['trip_duration'] < 50000]
sns.boxplot(x = 'vendor_id', y = 'trip_duration', data = trip_no_outliers)
plt.show()
```



Observation -

- Here we can see that vendor 2 has much outliers than vendor 1, and we know that the median for trip duration lies around 600.

Trip duration and Passenger Count

Are passengers with higher count, taking longer duration to complete the trip?

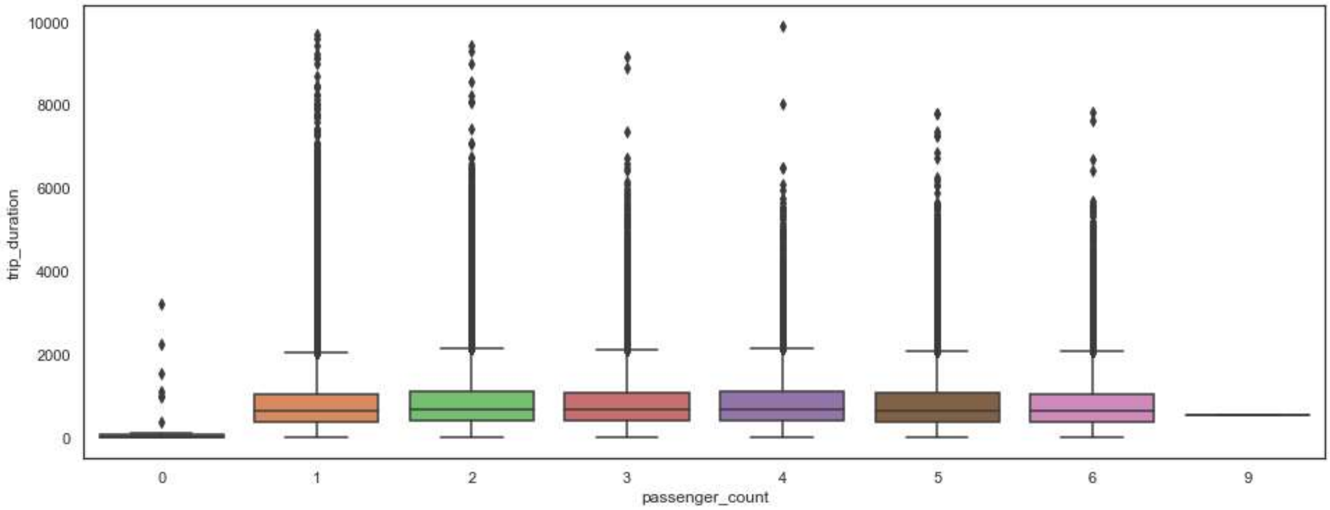
We will check the trend in duration of trips as compared to the number of passengers for the trip.

In [36]:
```
data.passenger_count.value_counts()
```

Out[36]:
```
1    515243
2    104576
5     38776
3     29561
6     24035
4     13972
0        31
9         1
Name: passenger_count, dtype: int64
```

As we know the median of trip_duration lies around 600 and we have huge outliers present in the trip_duration data, we will consider the trip_duration data of only less than 10,000 seconds.

In [37]:
```python
plt.figure(figsize=(16, 6))
trip_duration_new = data[data['trip_duration'] < 10000]
sns.boxplot(x="passenger_count", y="trip_duration", data = trip_duration_new)
plt.show()
```



Observation -

1. There are few trips recorded without any passenger.
2. Trips with 1 and 2 numbers of passengers have high amount of outliers present.
3. As the number of passengers are increasing, the outliers are decreasing.

Correlation Heatmap

We will check the correlations amongst all of the variables using heatmap.

In [38]:
```python
data.head()
```

Out[38]:

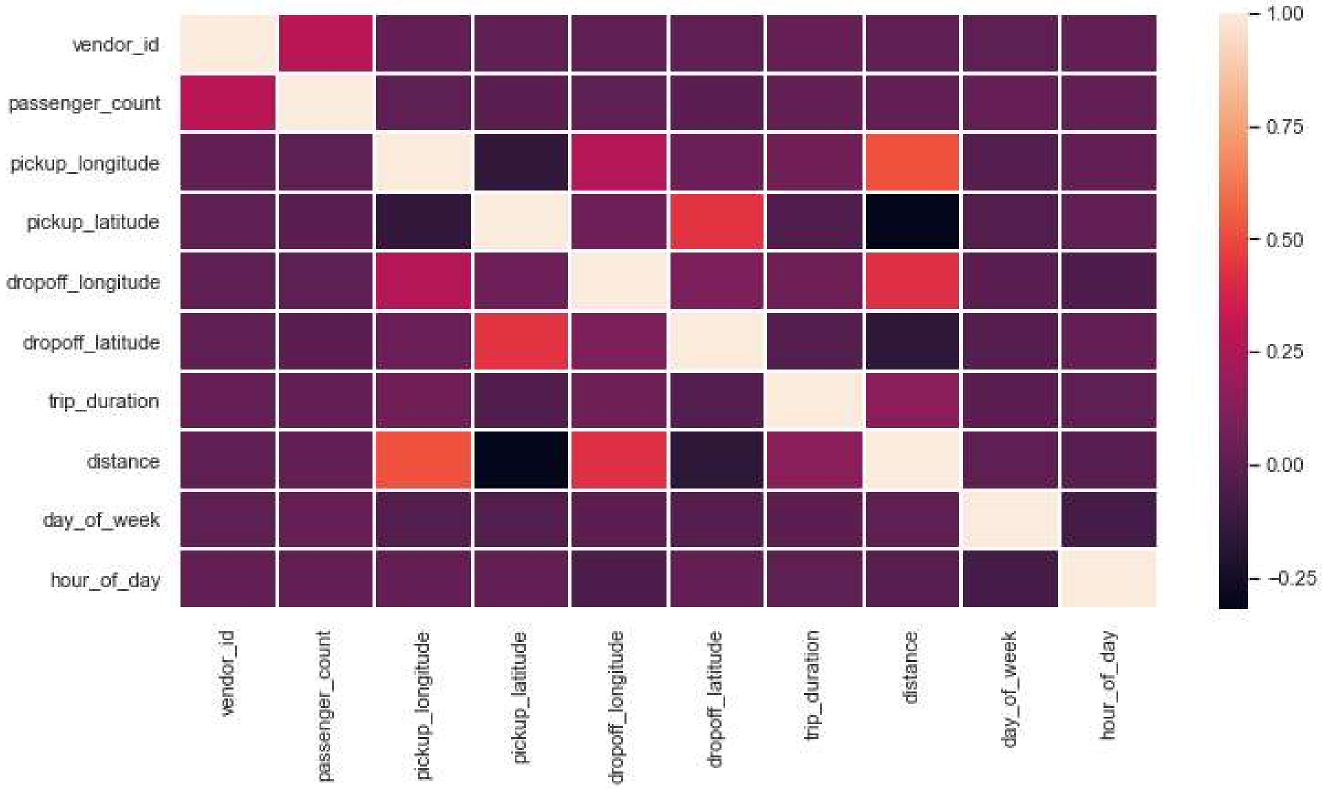
	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_lati
0	id1080784	2	2016-02-29 16:40:21	2016-02-29 16:47:01	1	-73.953918	40.778
1	id0889885	1	2016-03-11 23:35:37	2016-03-11 23:53:57	2	-73.988312	40.731
2	id0857912	2	2016-02-21 17:59:33	2016-02-21 18:26:48	2	-73.997314	40.721
3	id3744273	2	2016-01-05 09:44:31	2016-01-05 10:03:32	6	-73.961670	40.759
4	id0232939	1	2016-02-17 06:42:23	2016-02-17 06:56:31	1	-74.017120	40.708

In [39]:

```
# From the above dataset, we will drop those columns which are irrelevant with our targe

data_drop = data.drop(['id', 'pickup_datetime','dropoff_datetime', 'log_trip_duration']

plt.figure(figsize=(12, 6))
corr = data_drop.corr('pearson')
sns.heatmap(corr, linewidth=2)
plt.show()
```



Observation -

- From the above correlation heatmap, we see that the latitude and longitude have higher correlation with the target as compared to the others.

Conclusion

1. The trip duration of rides are forming almost normal curve.
2. As noticed earlier, there is an outlier present near 12.
3. Also there are very short rides present which are of less than 10 seconds, which are suspicious.
4. Most frequent rides include only 1 passenger, while some of the rides include 7 to 9 passengers too and they are very low in number.
5. Most of the rides have been completed by vendor 2 as compared to vendor 1.
6. There is almost no storing of data taking place in the taxi and being updated later. (Y - Yes, N - No)
7. Observing the above trend, we can see that the most of the rides are on Thursday, while on the weekends, there is lowest number of rides (0 is Sunday).
8. Total number of rides in 24 hours are mostly around 18-19 hours, i.e. evening. While in the morning peak hour, it is lower than expected.
9. Pickup and drop latitudes are denser around 40 to 41, and longitude are denser around -74 to -73.
10. Extreme values are present in the data which depicts higher value of distance.

10. Extreme values are present in the data which depicts higher value of distance.
11. Most of the rides are located between these locations, apart from few outliers outside the above range.
12. Longest trip duration has been observed on Wednesday.
13. Opposite to expectation, trip duration on weekends are lowest.
14. Trip duration during early morning are comparatively lesser which may be because of low traffic, and highest during evening peak hour.
15. There is a correlation between the number of pickups and trip duration as it follows the similar trend.
16. Here we can see that vendor 2 has much outliers than vendor 1, and we know that the median for trip duration lies around 600.
17. There are few trips recorded without any passenger.
18. Trips with 1 and 2 numbers of passengers have high amount of outliers present.
19. As the number of passengers are increasing, the outliers are decreasing.
20. From the above correlation heatmap, we see that the latitude and longitude have higher correlation with the target as compared to the others.

In [ ]: