NYC taxi trip duration:Project 2

In [42]:
```
# loading libraries to jupyter not book

%matplotlib inline
import numpy as np
import pandas as pd
from datetime import timedelta
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [43]:
```
# reading file
data = pd.read_csv('Documents/nyc_taxi_trip_duration.csv')
```

In [44]:
```
data.head()
```

Out[44]:

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

In [45]:
```
# getting shape of date
data.shape
```

Out[45]:
```
(729322, 11)
```

In [46]:
```
# getting cilumn details
data.columns
```

Out[46]:
```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration'],
      dtype='object')
```

In [47]:
```
# finind any null values is availble and dataypes
data.isna().sum(), data.dtypes
```

Out[47]:
```
(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,
 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)
```

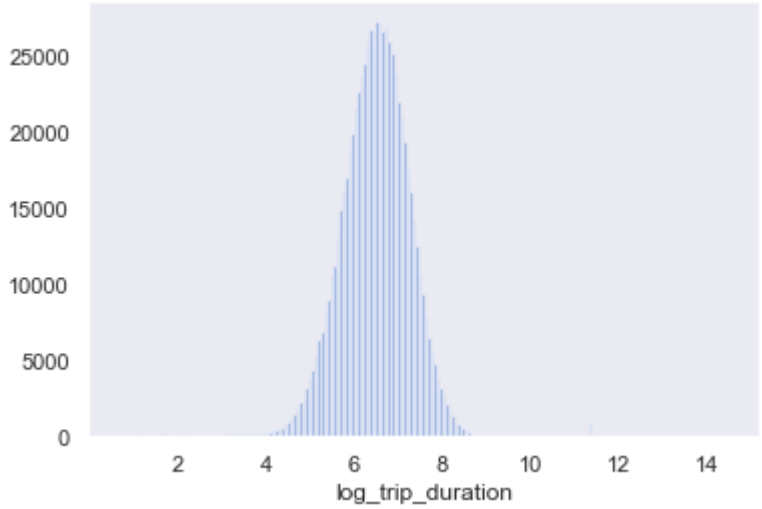
In [48]:
```
# converting strings to datetime features
data['pickup_datetime'] = pd.to_datetime(data.pickup_datetime)
data['dropoff_datetime'] = pd.to_datetime(data.dropoff_datetime)

# Converting yes/no flag to 1 and 0
data['store_and_fwd_flag'] = 1 * (data.store_and_fwd_flag.values == 'Y')

data['check_trip_duration'] = (data['dropoff_datetime'] - data['pickup_datet

duration_difference = data[np.abs(data['check_trip_duration'].values  - data
duration_difference.shape
```

Out[48]:
```
(0, 12)
```

In [49]:
```
# calculating tip duration in hours for better understanding
data['trip_duration'].describe()/3600
```

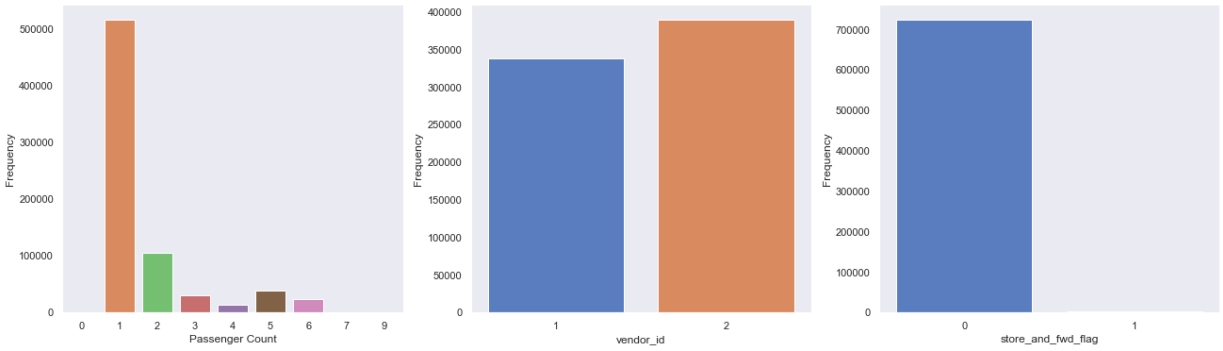
Out[49]:
```
count    202.589444
mean       0.264508
std        1.073507
min        0.000278
25%        0.110278
50%        0.184167
75%        0.298611
max      538.815556
Name: trip_duration, dtype: float64
```

In [50]:
```
data['log_trip_duration'] = np.log(data['trip_duration'].values + 1)
sns.distplot(data['log_trip_duration'], kde = False, bins = 200)
plt.show()
```



In [51]:
```
plt.figure(figsize=(22, 6))
#fig, axs = plt.subplot(ncols=2)

# Passenger Count
plt.subplot(131)
sns.countplot(data['passenger_count'])
plt.xlabel('Passenger Count')
plt.ylabel('Frequency')


# vendor_id
plt.subplot(132)
sns.countplot(data['vendor_id'])
plt.xlabel('vendor_id')
plt.ylabel('Frequency')

# store_and_fwd_flag
plt.subplot(133)
sns.countplot(data['store_and_fwd_flag'])
plt.xlabel('store_and_fwd_flag')
plt.ylabel('Frequency')
```

Out[51]:
```
Text(0, 0.5, 'Frequency')
```



Observations:

1. Most of the trips involve only 1 passenger. There are trips with 7-9 passengers but they are very low in number.
2. Vendor 2 has more number of trips as compared to vendor 1
3. The store_and_fwd_flag values, indicating whether the trip data was sent immediately to the vendor (“0”) or held in the memory of the taxi because there was no connection to the server (“1”), show that there was almost no storing taking place

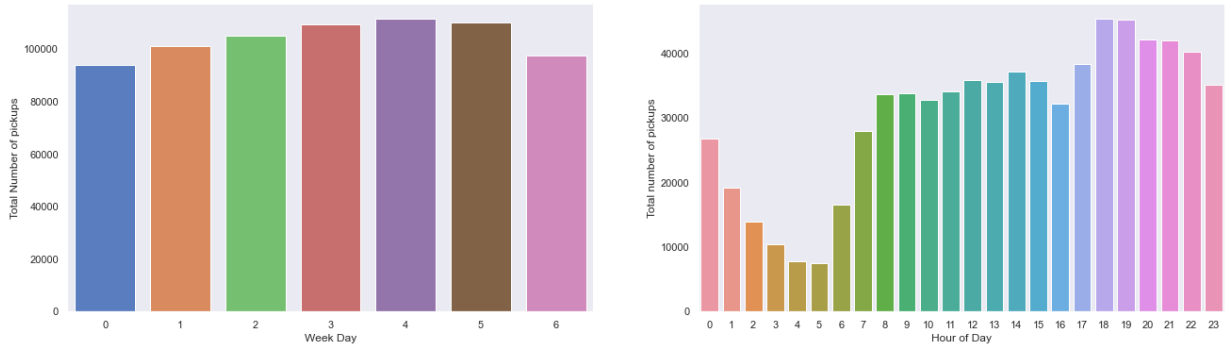
In [52]:
```
# getting most frequent duration trips
data['pickup_datetime'].min(), data['pickup_datetime'].max()
```

Out[52]:
```
(Timestamp('2016-01-01 00:01:14'), Timestamp('2016-06-30 23:59:37'))
```

found these trips happened in first 6 months of 2016

In [53]:
```
# we can extract weekdays and hour of days
data['day_of_week'] = data['pickup_datetime'].dt.weekday
data['hour_of_day'] = data['pickup_datetime'].dt.hour

# Datetime features
plt.figure(figsize=(22, 6))

# Passenger Count
plt.subplot(121)
sns.countplot(data['day_of_week'])
plt.xlabel('Week Day')
plt.ylabel('Total Number of pickups')

# vendor_id
plt.subplot(122)
sns.countplot(data['hour_of_day'])
plt.xlabel('Hour of Day')
plt.ylabel('Total number of pickups')
```

Out[53]: Text(0, 0.5, 'Total number of pickups')

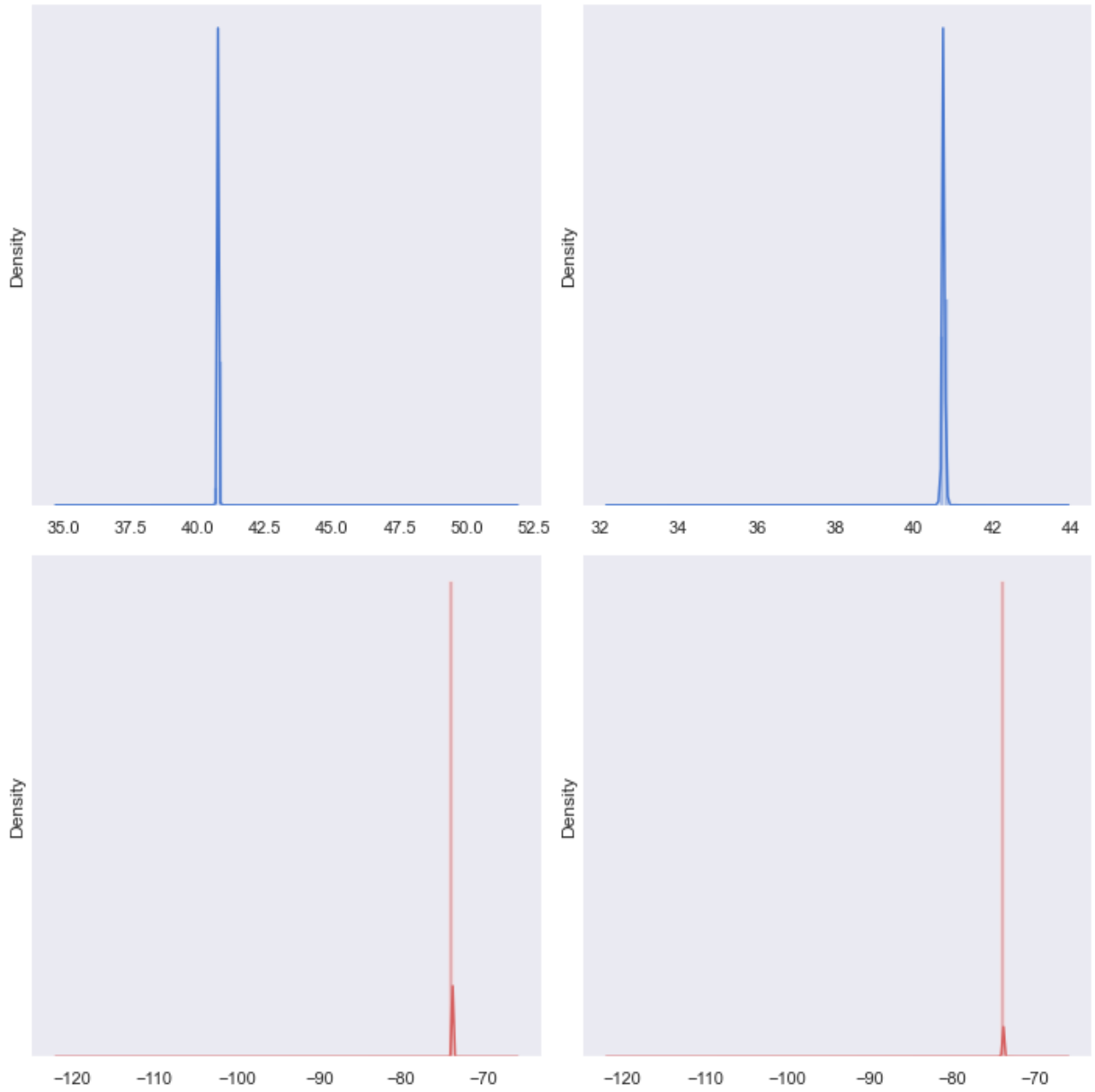


- Number of pickups for weekends is much lower than week days with a peak on Thursday (4). Note that here weekday is a decimal number, where 0 is Sunday and 6 is Saturday.
- Number of pickups as expected is highest in late evenings. However, it is much lower during the morning peak hours.

Location consistancy check throug Longitude and lattitude

In [54]:

```
# finding location consistancy through longitude and lattitude

sns.set(style="dark", 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['pickup_latitude'].values, label = 'pickup_latitude',color
sns.distplot(data['pickup_longitude'].values, label = 'pickup_longitude',col
sns.distplot(data['dropoff_latitude'].values, label = 'dropoff_latitude',col
sns.distplot(data['dropoff_longitude'].values, label = 'dropoff_longitude',c
plt.setp(axes, yticks=[])
plt.tight_layout()
plt.show()
```

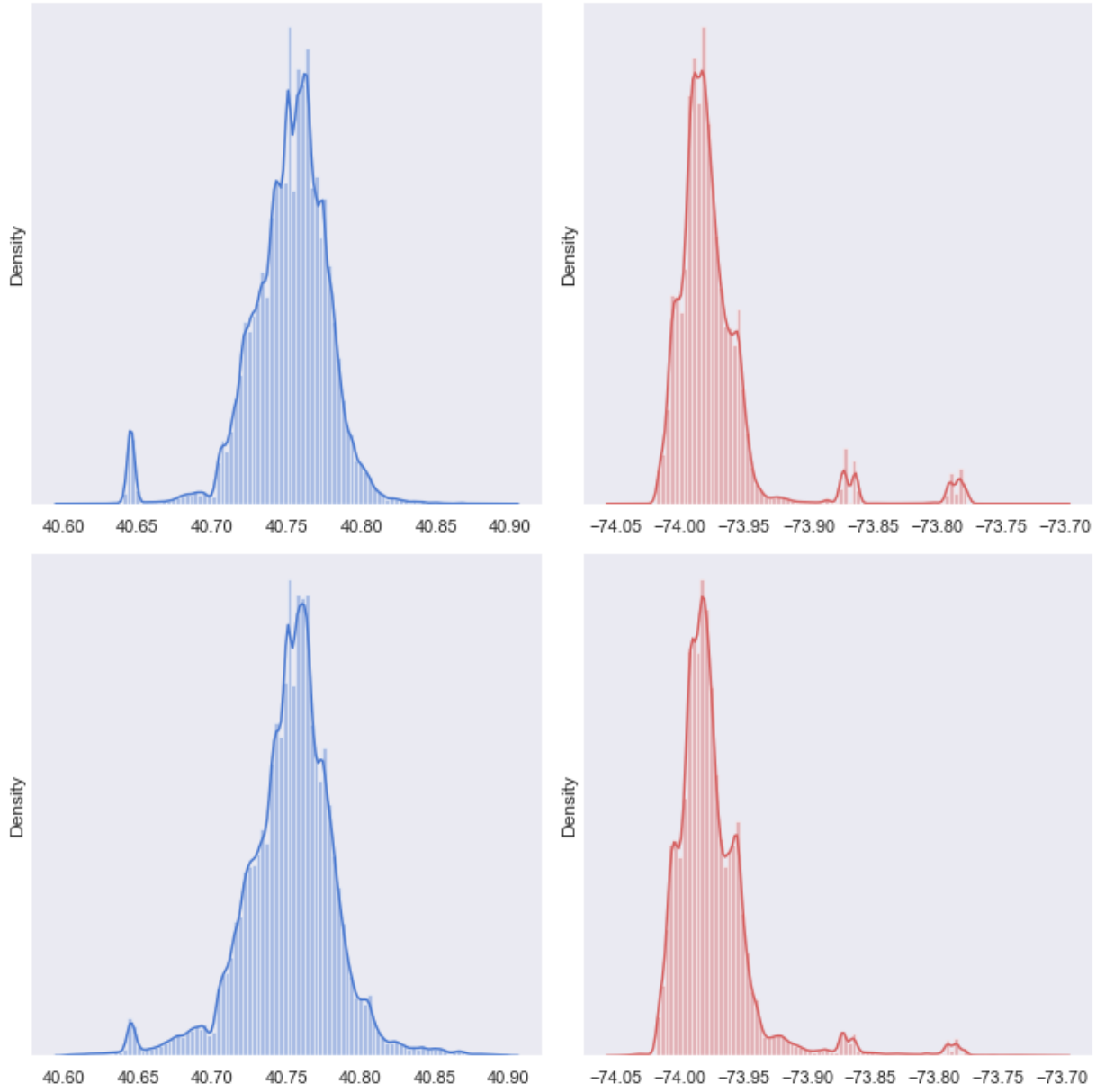


1. From the plot above it is clear that pick and drop latitude are centered around 40 to 41, and longitude are situated around -74 to -73.
2. Some extreme co-ordinates has squeezed the plot such that we see a spike here
3. A good idea is to remove these outliers and look at the distribution more closely

Removing ouliers

In [55]:

```
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
data = data.loc[(data.pickup_longitude > -74.05) & (data.pickup_longitude <
data_new = data.copy()
sns.set(style="dark", 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',c
sns.distplot(data_new['pickup_longitude'].values, label = 'pickup_longitude'
sns.distplot(data_new['dropoff_latitude'].values, label = 'dropoff_latitude'
sns.distplot(data_new['dropoff_longitude'].values, label = 'dropoff_longitud
plt.setp(axes, yticks=[])
plt.tight_layout()

plt.show()
```

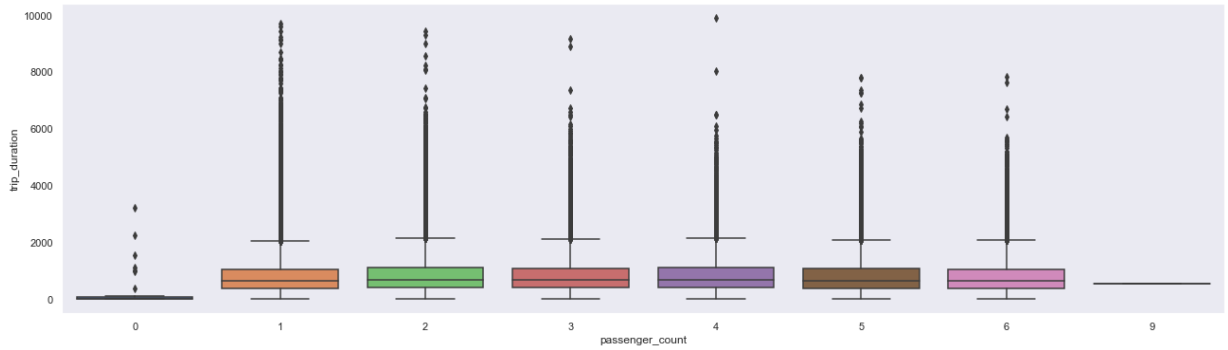


In [56]:
```
1 data.columns
```

Out[56]:
```
Index(['id', 'vendor_id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count', 'pickup_longitude', 'pickup_latitude',
       'dropoff_longitude', 'dropoff_latitude', 'store_and_fwd_flag',
       'trip_duration', 'check_trip_duration', 'log_trip_duration',
       'day_of_week', 'hour_of_day'],
      dtype='object')
```

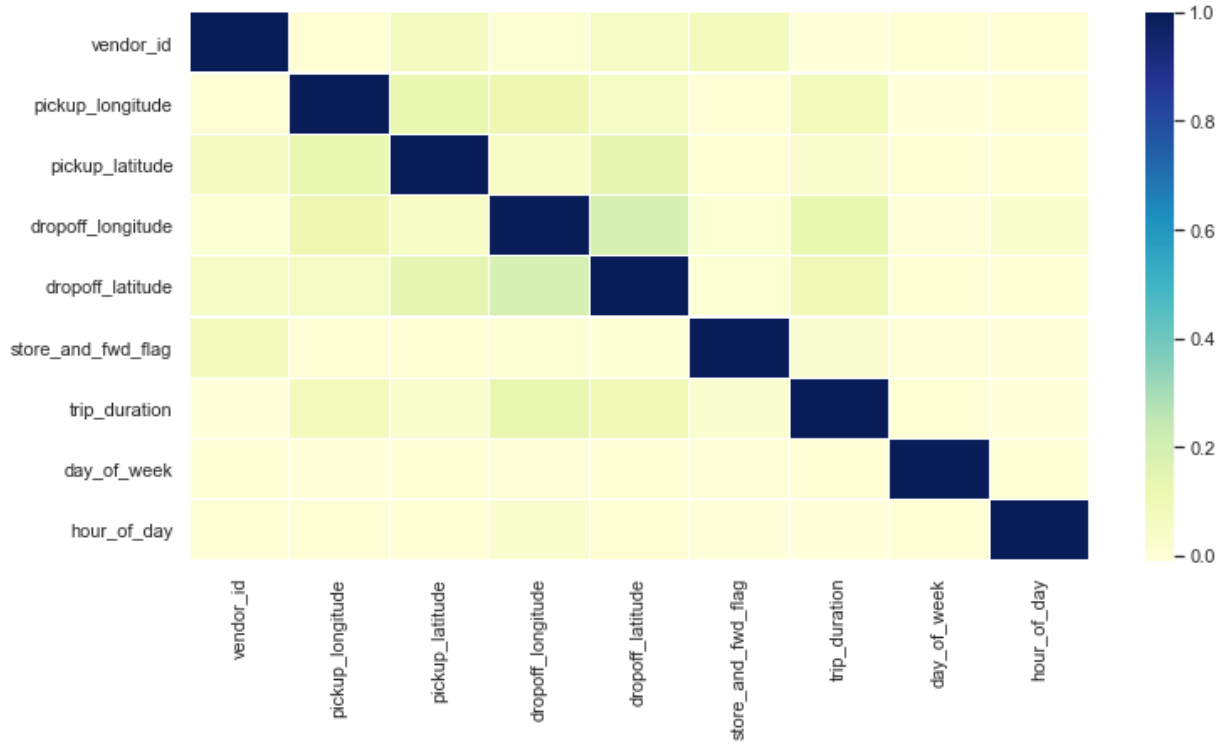
Time duration and passenger count

In [57]:
```
1 data.passenger_count.value_counts()
2 plt.figure(figsize=(22, 6))
3 data_sub = data[data['trip_duration'] < 10000]
4 sns.boxplot(x="passenger_count", y="trip_duration", data=data_sub)
5 plt.show()
```



corelating with Heat map

In [58]:
```
# Corelatig to heat map

plt.figure(figsize=(12, 6))
data = data.drop(['id', 'pickup_datetime', 'dropoff_datetime',
       'passenger_count','check_trip_duration','log_trip_duration'],
        axis=1)
corr = data.apply(lambda x: pd.factorize(x)[0]).corr()
ax = sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns,
                 linewidths=.2, cmap="YlGnBu")
```



Results

1. The majority of rides follow a rather smooth distribution that looks almost log-normal with a peak just around exp(6.5) i.e. about 17 minutes.
2. There are several suspiciously short rides with less than 10 seconds duration.
3. As discussed earlier, there are a few huge outliers near 12.
4. Most of the trips involve only 1 passenger. There are trips with 7-9 passengers but they are very low in number.
5. Vendor 2 has more number of trips as compared to vendor 1
6. Number of pickups for weekends is much lower than week days with a peak on Thursday (4). Note that here weekday is a decimal number, where 0 is Sunday and 6 is Saturday.
7. From the correlation heatmap we see that the lattitude and longitude features have higher correlation with the target as compared to the other features.