Uber Trips Analysis with Data Visualisation

Importing Important Libraries

In [1]:

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Number of files stored in Uber_Datasets

In [2]:

```
files = [file for file in os.listdir(r"C:/Users/archi/Kaggle_Competition/Uber_Dataset_Analysis/Raw_Uber_Datasets")]

for file in files:
    print(file)
```

```
uber-raw-data-apr14.csv
uber-raw-data-aug14.csv
uber-raw-data-jul14.csv
uber-raw-data-jun14.csv
uber-raw-data-may14.csv
uber-raw-data-sep14.csv
```

Concatenate data from each month into one CSV

In [3]:

```
files = [file for file in os.listdir(r"C:/Users/archi/Kaggle_Competition/Uber_Dataset_Analysis/Raw_Uber_Datasets")]

all_month_trips=pd.DataFrame()

for file in files:
    df=pd.read_csv(r"C:/Users/archi/Kaggle_Competition/Uber_Dataset_Analysis/Raw_Uber_Datasets/"+file)
    all_month_trips=pd.concat([all_month_trips, df])
```

```
all_month_trips.to_csv("all_trips.csv", index=False)
```

Name and Read the updated dataframe

In [4]:

```
all_trips=pd.read_csv("all_trips.csv")
all_trips.head()
```

Out[4]:

	Date/Time	Lat	Lon	Base
0	4/1/2014 0:11:00	40.7690	-73.9549	B02512
1	4/1/2014 0:17:00	40.7267	-74.0345	B02512
2	4/1/2014 0:21:00	40.7316	-73.9873	B02512
3	4/1/2014 0:28:00	40.7588	-73.9776	B02512
4	4/1/2014 0:33:00	40.7594	-73.9722	B02512

Dataset Information

In [5]:

```
all_trips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4534327 entries, 0 to 4534326
Data columns (total 4 columns):
 #   Column     Dtype  
---  ------     -----  
 0   Date/Time  object 
 1   Lat        float64
 2   Lon        float64
 3   Base       object 
dtypes: float64(2), object(2)
memory usage: 138.4+ MB
```

Conversion of 'Date/Time' column from string to datetime format

In [6]:

```
all_trips['Date/Time'] = pd.to_datetime(all_trips['Date/Time'])
```

In [7]:

```
all_trips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4534327 entries, 0 to 4534326
Data columns (total 4 columns):
 #   Column     Dtype         
---  ------     -----         
 0   Date/Time  datetime64[ns]
 1   Lat        float64       
 2   Lon        float64       
 3   Base       object        
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 138.4+ MB
```

Exploratory Data Analysis

Add Month, Day, and Year Column

In [8]:

```
all_trips['Month'] = pd.to_datetime(all_trips['Date/Time']).dt.month
all_trips['Day'] = pd.to_datetime(all_trips['Date/Time']).dt.day
all_trips['Year'] = pd.to_datetime(all_trips['Date/Time']).dt.year
all_trips.head()
```

Out[8]:

	Date/Time	Lat	Lon	Base	Month	Day	Year
0	2014-04-01 00:11:00	40.7690	-73.9549	B02512	4	1	2014
1	2014-04-01 00:17:00	40.7267	-74.0345	B02512	4	1	2014
2	2014-04-01 00:21:00	40.7316	-73.9873	B02512	4	1	2014
3	2014-04-01 00:28:00	40.7588	-73.9776	B02512	4	1	2014
4	2014-04-01 00:33:00	40.7594	-73.9722	B02512	4	1	2014

Add DayofWeek Column

In [9]:

```
all_trips['DayofWeek'] = pd.to_datetime(all_trips['Date/Time']).dt.day_name()
all_trips.head()
```

Out[9]:

	Date/Time	Lat	Lon	Base	Month	Day	Year	DayofWeek
0	2014-04-01 00:11:00	40.7690	-73.9549	B02512	4	1	2014	Tuesday
1	2014-04-01 00:17:00	40.7267	-74.0345	B02512	4	1	2014	Tuesday
2	2014-04-01 00:21:00	40.7316	-73.9873	B02512	4	1	2014	Tuesday
3	2014-04-01 00:28:00	40.7588	-73.9776	B02512	4	1	2014	Tuesday
4	2014-04-01 00:33:00	40.7594	-73.9722	B02512	4	1	2014	Tuesday

Add Hour, Minute, and Second Column

In [10]:

```
all_trips['Hour'] = pd.to_datetime(all_trips['Date/Time']).dt.hour
all_trips['Minute'] = pd.to_datetime(all_trips['Date/Time']).dt.minute
all_trips['Second'] = pd.to_datetime(all_trips['Date/Time']).dt.second
all_trips.head()
```

Out[10]:

	Date/Time	Lat	Lon	Base	Month	Day	Year	DayofWeek	Hour	Minute	Second
0	2014-04-01 00:11:00	40.7690	-73.9549	B02512	4	1	2014	Tuesday	0	11	0
1	2014-04-01 00:17:00	40.7267	-74.0345	B02512	4	1	2014	Tuesday	0	17	0
2	2014-04-01 00:21:00	40.7316	-73.9873	B02512	4	1	2014	Tuesday	0	21	0
3	2014-04-01 00:28:00	40.7588	-73.9776	B02512	4	1	2014	Tuesday	0	28	0
4	2014-04-01 00:33:00	40.7594	-73.9722	B02512	4	1	2014	Tuesday	0	33	0

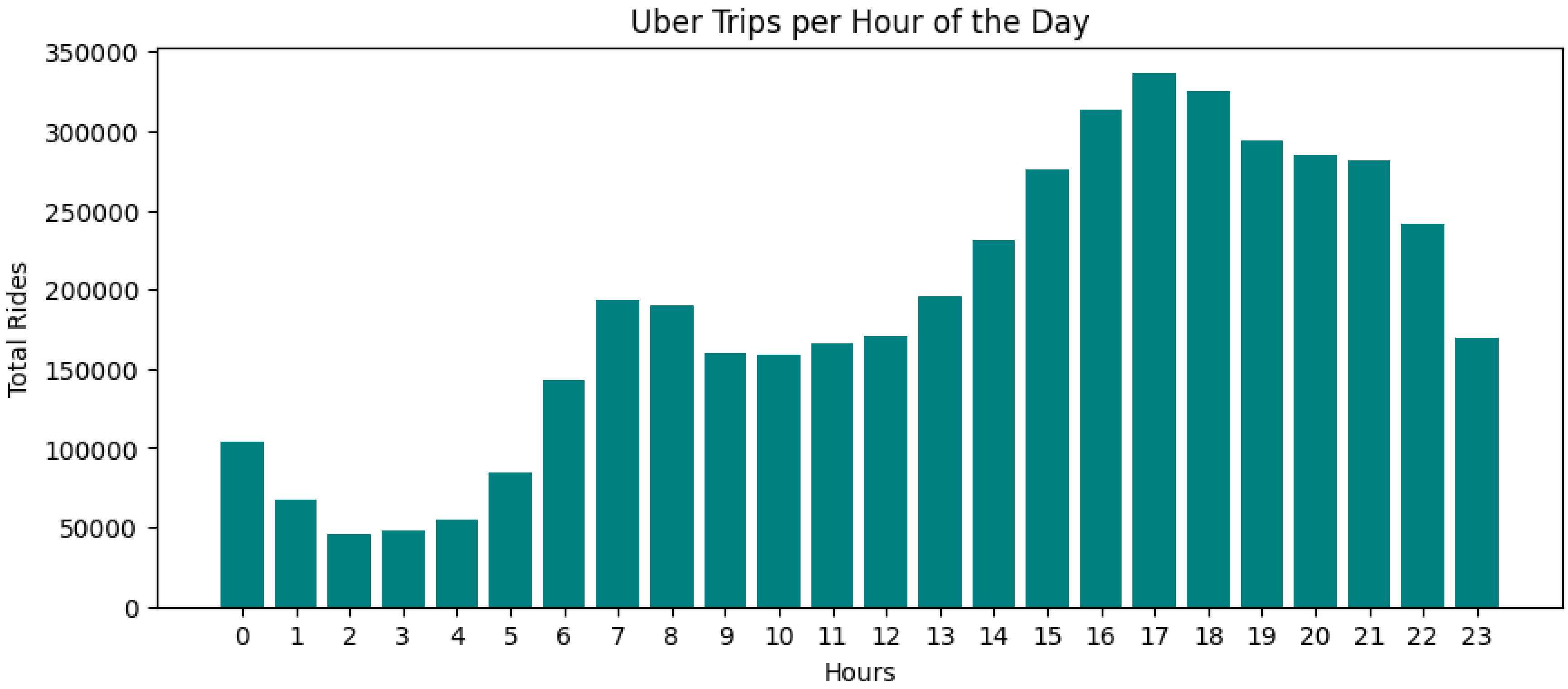
Data Visualisation

Uber Trips Per Hour

In this we want to analyse the peak times for Uber rides in a day taking all the dates into factor.

In [11]:
```
trips_per_hour = all_trips.groupby('Hour').count()

plt.figure(figsize=(10,4))
hourly = [hour for hour, df in all_trips.groupby('Hour')]
plt.bar(hourly, trips_per_hour['Day'], color='teal')
plt.xticks(hourly)
plt.title("Uber Trips per Hour of the Day")
plt.ylabel('Total Rides')
plt.xlabel('Hours')
plt.show()
```



So, Peak times for Uber rides in NYC during 2014 were mostly in the evening at 4PM to 7PM being the busiest.

Uber Trips Per Weekdays

In [12]:

```
all_trips['DayofWeek'].value_counts()
```

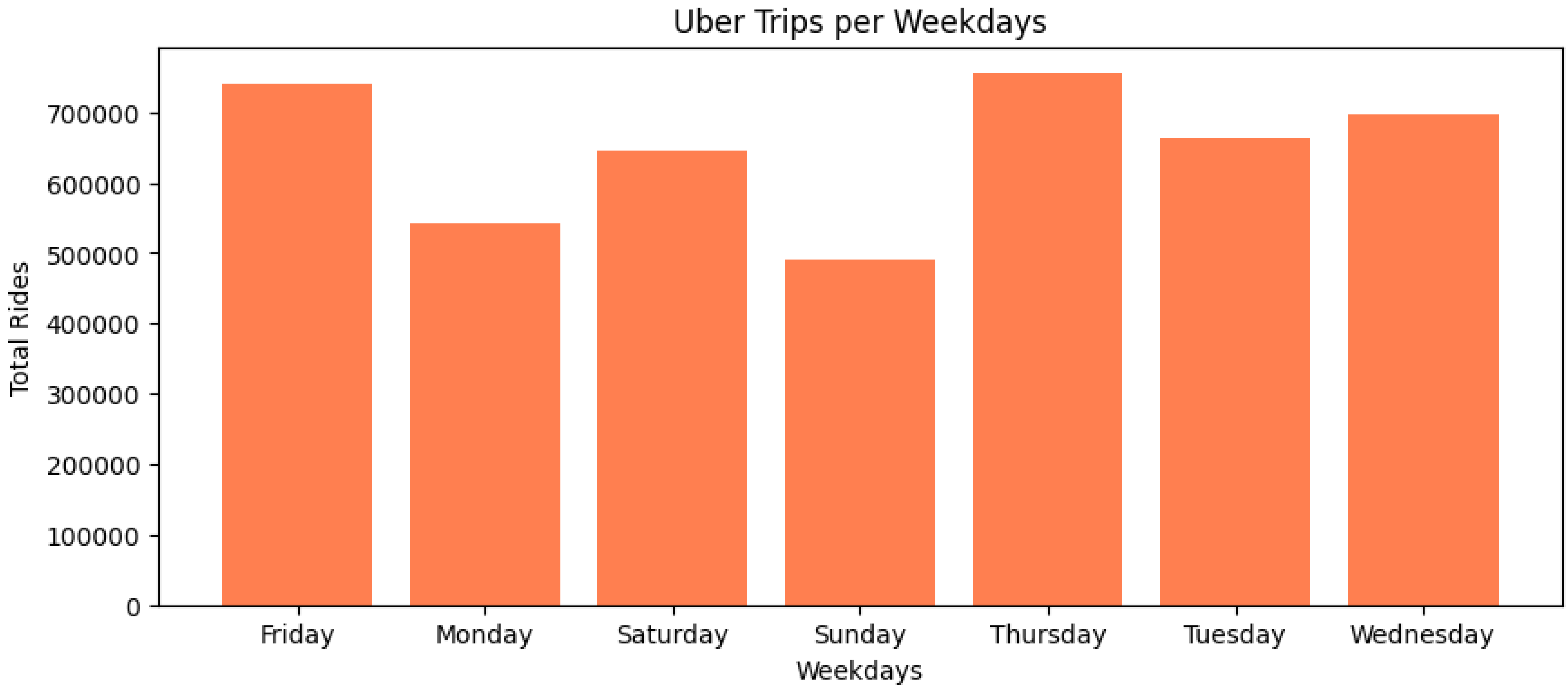
Out[12]:

```
Thursday     755145
Friday       741139
Wednesday    696488
Tuesday      663789
Saturday     646114
Monday       541472
Sunday       490180
Name: DayofWeek, dtype: int64
```

In [13]:

```
trips_per_weekday = all_trips.groupby('DayofWeek').count()

plt.figure(figsize=(10,4))
weekdays = [days for days, df in all_trips.groupby('DayofWeek')]
plt.bar(weekdays, trips_per_weekday['Day'], color='coral')
plt.xticks(weekdays)
plt.title("Uber Trips per Weekdays")
plt.ylabel('Total Rides')
plt.xlabel('Weekdays')
plt.show()
```



So, the peak weekdays are Friday and Thursday when a large number of Uber are booked in the year 2014.

Uber Trips Per Month

Let us add months to the table as well, to see how the months affect the data.

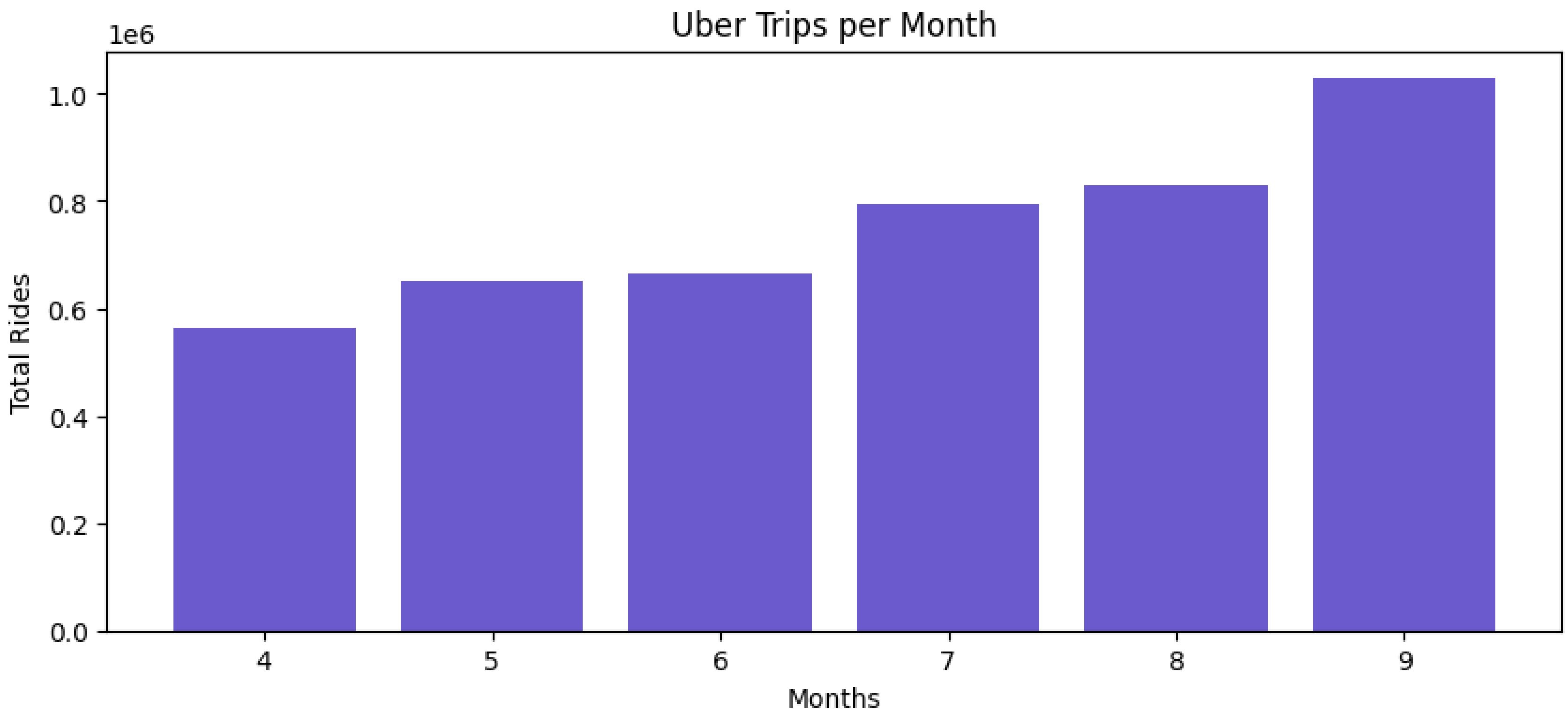
```
In [14]: all_trips['Month'].value_counts()
```

```
Out[14]: 9    1028136
8     829275
7     796121
6     663844
5     652435
4     564516
Name: Month, dtype: int64
```

In [15]:

```
trips_per_month = all_trips.groupby('Month').count()

plt.figure(figsize=(10,4))
monthly = [month for month, df in all_trips.groupby('Month')]
plt.bar(monthly, trips_per_month['Day'], color='slateblue')
plt.xticks(monthly)
plt.title("Uber Trips per Month")
plt.ylabel('Total Rides')
plt.xlabel('Months')
plt.show()
```



We can Say that the month to contribute the greatest number of trips at 4PM to 7PM was on September.

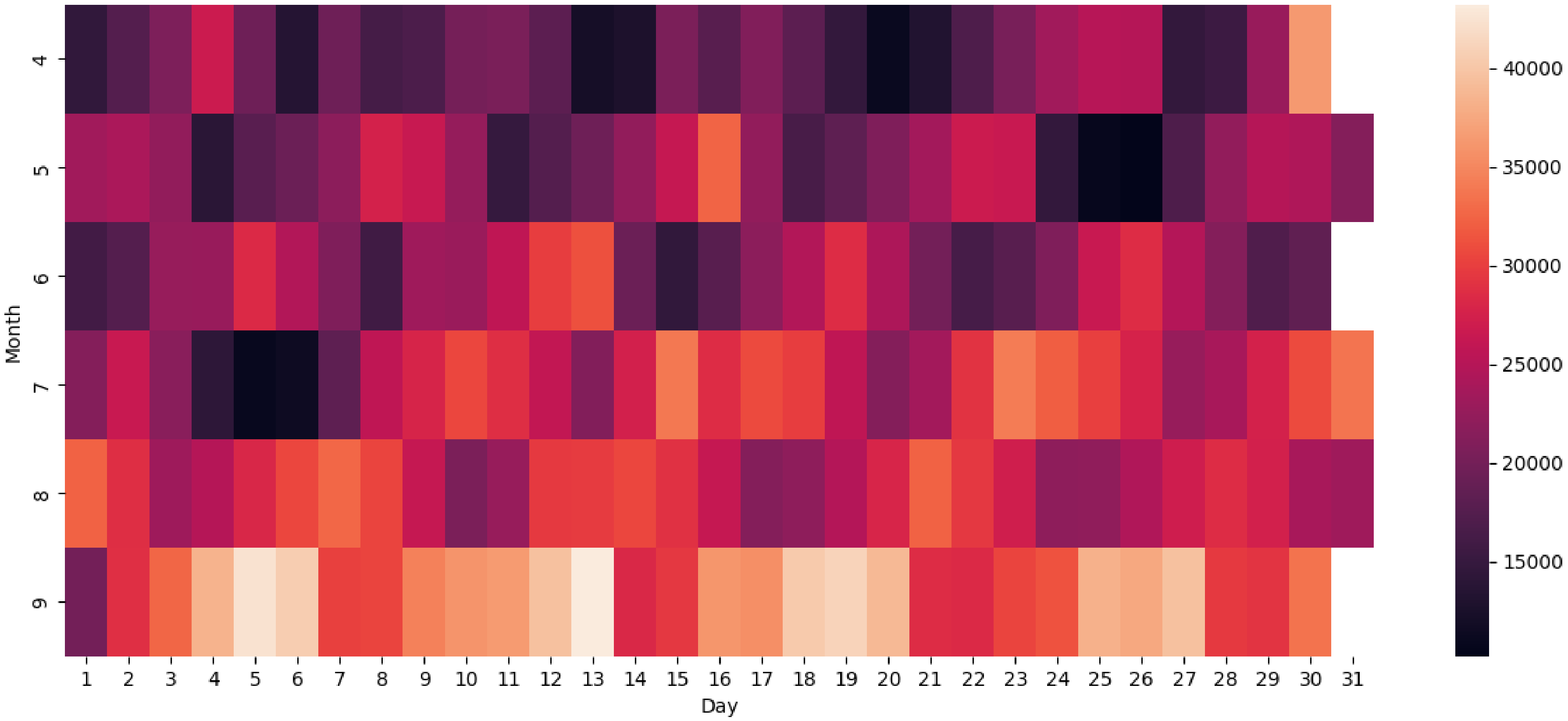
HeatMap Exploration

In [16]:
```
def heatmap(col1,col2):
    by_cross = all_trips.groupby([col1,col2]).size()
    pivot = by_cross.unstack()
    plt.figure(figsize=(15,6))
    return sns.heatmap(pivot)
```

Peak Date of Trips by Month

We know our peak times, let's find out which date of the month have the greatest number of trips from April to September.

In [17]:
```
heatmap('Month','Day')
plt.show()
```



From the heatmap, it shows that on 5th and 13th September the uber trips were it's peak because -

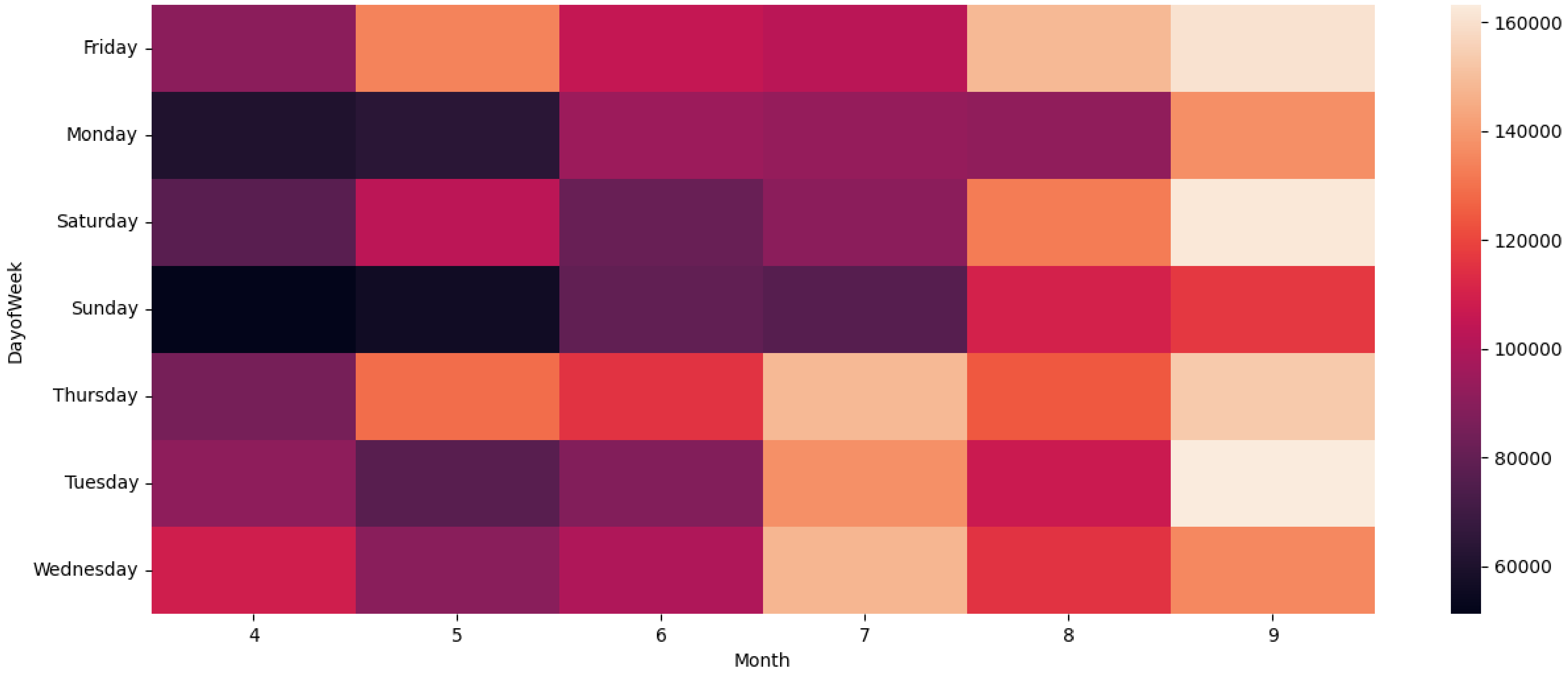
- On 5th September it's a "Labor Day" Celebration. It is a federal holiday in the USA. On this Day, everyone honor and recognize the American labor movement and the works and contributions of laborers to the development and achievements of the United States.
- On 13th September it's a "International Chocolate Day". To celebrate with everyone as we give them some cool facts and fun ideas to celebrate this well-loved treat.

Peak Day of the Week by Month

We know our peak times, let's find out which days of the week have the greatest number of trips from April to September.

In [18]:

```
heatmap('DayofWeek','Month')
plt.show()
```



It shows that in September the highest number of trips were booked on Tuesday, Friday, and Saturday. But from this graph, we can also tell that September had the greatest number of trips with Tuesday as the peak day.

Map Plot of New York City

Maximum and Minimum co-ordinate points in Longitudinal Column

```
In [19]: all_trips['Lon'].max()
```

```
Out[19]: -72.0666
```

```
In [20]: all_trips['Lon'].min()
```

```
Out[20]: -74.929
```

Maximum and Minimum co-ordinate points in Latitudinal Column

In [21]:

```
all_trips['Lat'].max()
```

Out[21]:

```
42.1166
```

In [22]:

```
all_trips['Lat'].min()
```

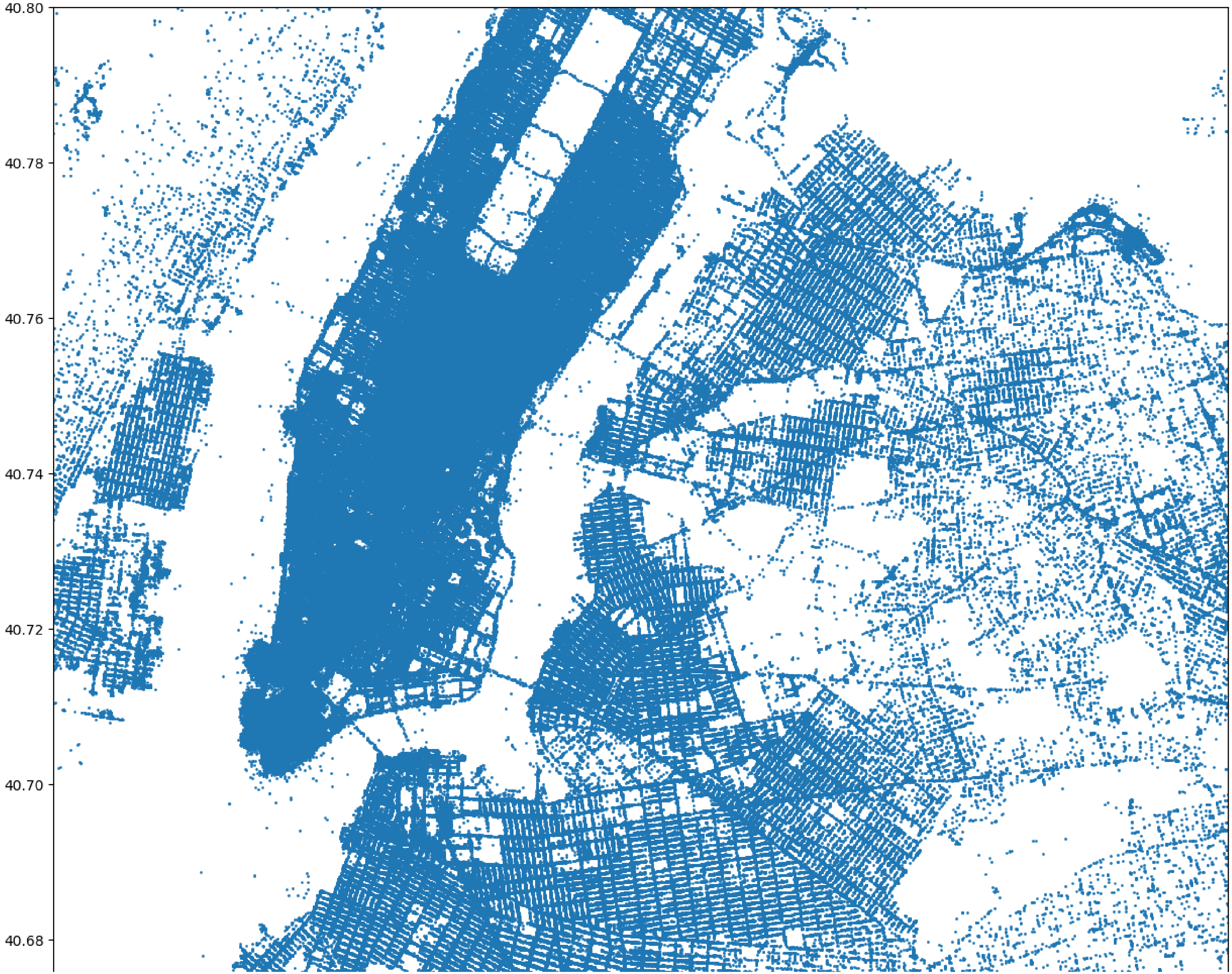
Out[22]:

```
39.6569
```

Scatter Plot of City:

In [24]:

```
plt.figure(figsize=(15,15))
plt.scatter(all_trips['Lon'],all_trips['Lat'],s=1)
plt.xlim(-74.05, -73.85)
plt.ylim(40.65, 40.80)
plt.show()
```


40.80
40.78
40.76
40.74
40.72
40.70
40.68

We can see from the map that, in the mid-region it shows that the majority of the uber trip bookings in NYC. This is because the visitors, tourists, and commuters fill the city during the day. There are various attractions in these regions as well as businesses, retail, and service jobs that bring people around this area.

With this analysis, Uber could do various changes to improve the business in this region.

- Uber could increase the number of drives in these areas during peak times and peak days to provide everyone.