

# UberCaseStudy

May 13, 2019

## Uber Supply-Demand Gap

### 0.0.1 Importing Necessary libraries and Datafile

```
In [113]: import pandas as pd
import os
df = pd.read_csv(os.getcwd()+'\\Uber Request Data.csv')
```

```
In [114]: df = df[df.isna().sum(axis=1)<=2]
df.head()
```

```
Out[114]:
```

	Request id	Pickup point	Driver id	Status	Request timestamp	\
0	619	Airport	1.0	Trip Completed	11/7/2016 11:51	
1	867	Airport	1.0	Trip Completed	11/7/2016 17:57	
2	1807	City	1.0	Trip Completed	12/7/2016 9:17	
3	2532	Airport	1.0	Trip Completed	12/7/2016 21:08	
4	3112	City	1.0	Trip Completed	13-07-2016 08:33:16	

	Drop timestamp
0	11/7/2016 13:00
1	11/7/2016 18:47
2	12/7/2016 9:58
3	12/7/2016 22:03
4	13-07-2016 09:25:47

### 0.0.2 Removing Rows that contains NA values less than equal to 2 and printing out dataframe head

```
In [115]: df['Request timestamp'] = pd.to_datetime(df['Request timestamp'])
df['Drop timestamp'] = pd.to_datetime(df['Drop timestamp'])
```

### 0.0.3 Converting Dataframe series to Data and Time for the columns 'Request timestamp' and 'Drop timestamp'

```
In [116]: df['dayname'] = df['Request timestamp'].dt.dayofweek
df['Hour'] = df['Request timestamp'].dt.hour
```

#### 0.0.4 Adding a new column 'dayname' that contains day of the week and 'Hour' that contains hour at which request was made

```
In [117]: df['date'] = df['Request timestamp'].dt.date
```

#### 0.0.5 Adding a date column of Request timestamp

```
In [118]: df.head()
```

```
Out[118]:
```

	Request id	Pickup point	Driver id	Status	Request timestamp	\
0	619	Airport	1.0	Trip Completed	2016-11-07 11:51:00	
1	867	Airport	1.0	Trip Completed	2016-11-07 17:57:00	
2	1807	City	1.0	Trip Completed	2016-12-07 09:17:00	
3	2532	Airport	1.0	Trip Completed	2016-12-07 21:08:00	
4	3112	City	1.0	Trip Completed	2016-07-13 08:33:16	

	Drop timestamp	dayname	Hour	date
0	2016-11-07 13:00:00	0	11	2016-11-07
1	2016-11-07 18:47:00	0	17	2016-11-07
2	2016-12-07 09:58:00	2	9	2016-12-07
3	2016-12-07 22:03:00	2	21	2016-12-07
4	2016-07-13 09:25:47	2	8	2016-07-13

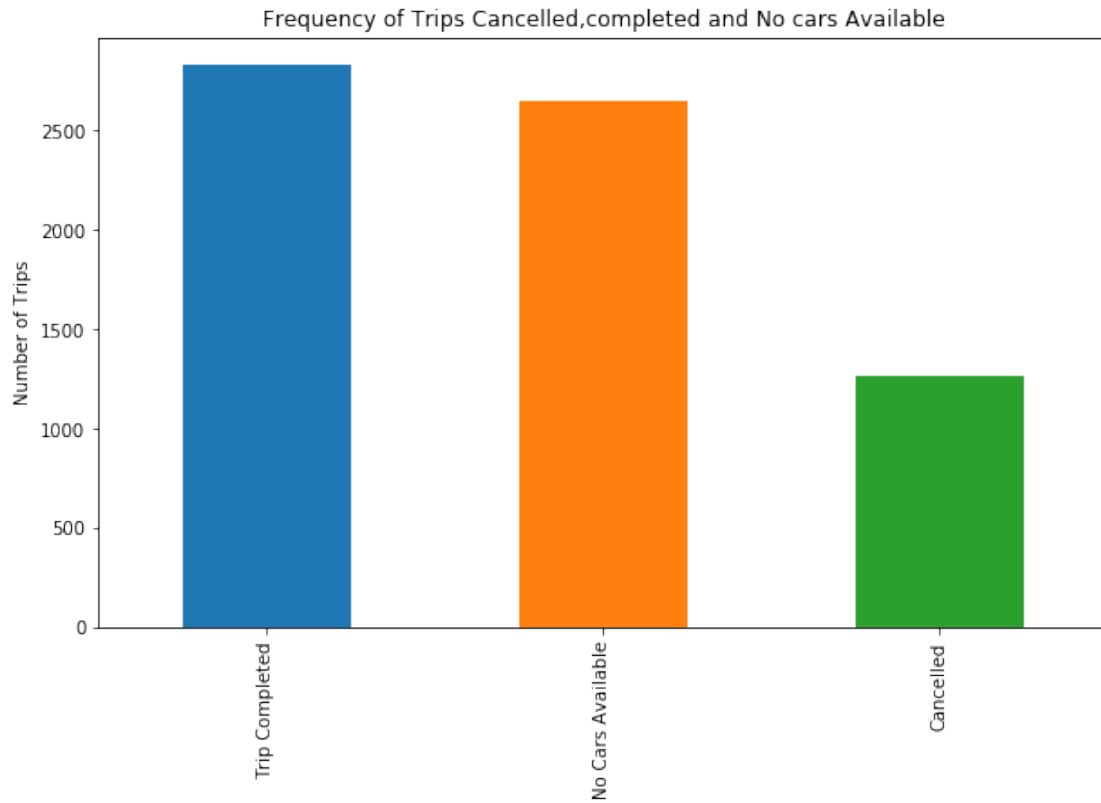
```
In [119]: df['Status'].value_counts()
```

```
Out[119]:
```

Trip Completed	2831
No Cars Available	2650
Cancelled	1264
Name: Status, dtype: int64	

#### 0.0.6 A simple breakdown of all the trips

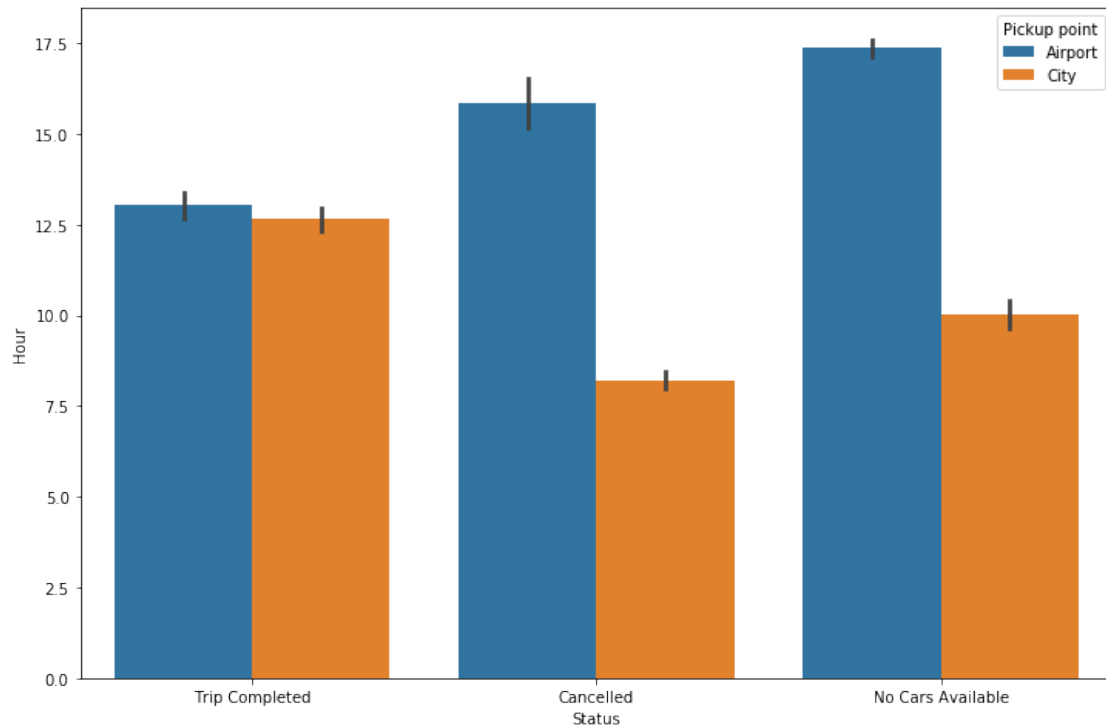
```
In [105]: plt.figure(figsize=(10,6))
trips = df['Status'].value_counts()
trips.plot(kind='bar')
plt.ylabel('Number of Trips')
plt.title('Frequency of Trips Cancelled,completed and No cars Available')
plt.show()
```



0.0.7 The above values of status is showing a simple breakdown of all the trips in a bar plot

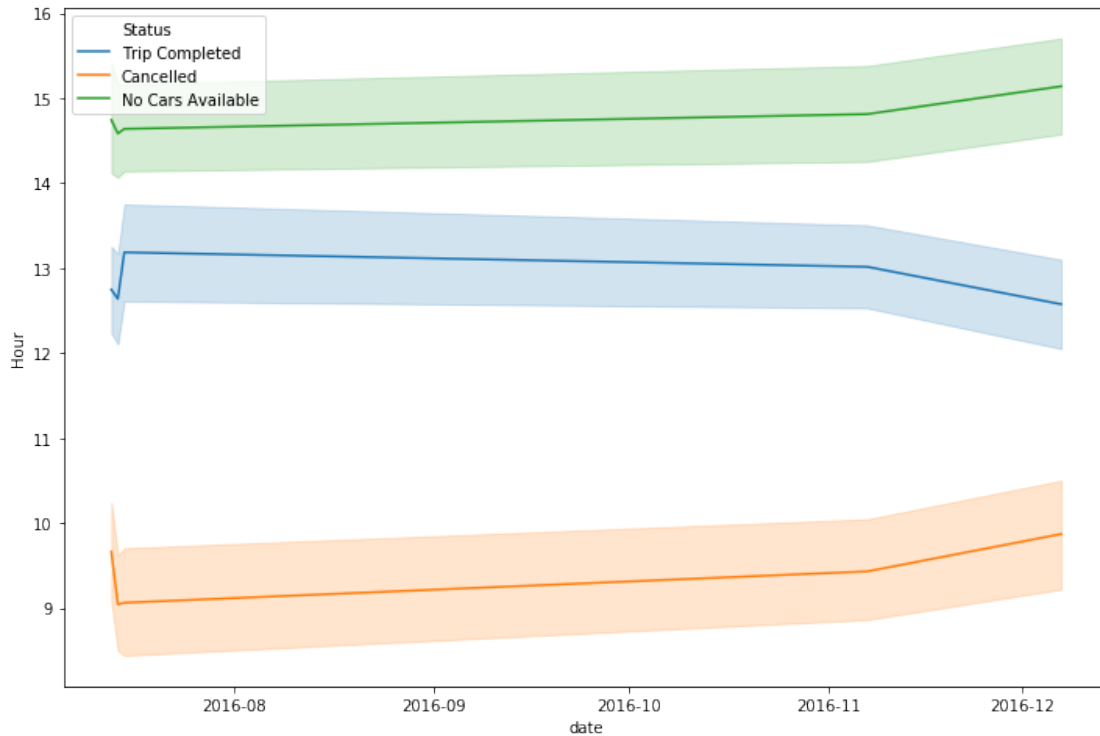
```
In [58]: import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [75]: plt.figure(figsize=(12,8))
sns.barplot(df['Status'],df['Hour'],hue=(df['Pickup point']))
plt.show()
```



Cancelled Status : The above figure gives us a glimpse that cancellation occurs mostly when the pickup point is Airport at a time around 15:00 hours and In the city mostly around 7:30 hours No cars available: It is mostly at 17:30 hours at the airport and 10:00 hours in the city Trip completed: Successful trips are at around 12:30 hours

```
In [88]: plt.figure(figsize=(12,8))
sns.lineplot(x=df['date'],y=df['Hour'] ,hue=df['Status'])
plt.show()
```



**0.0.8 The above line chart shows a much simpler picture to understand Trip status during different hours in a timeline**

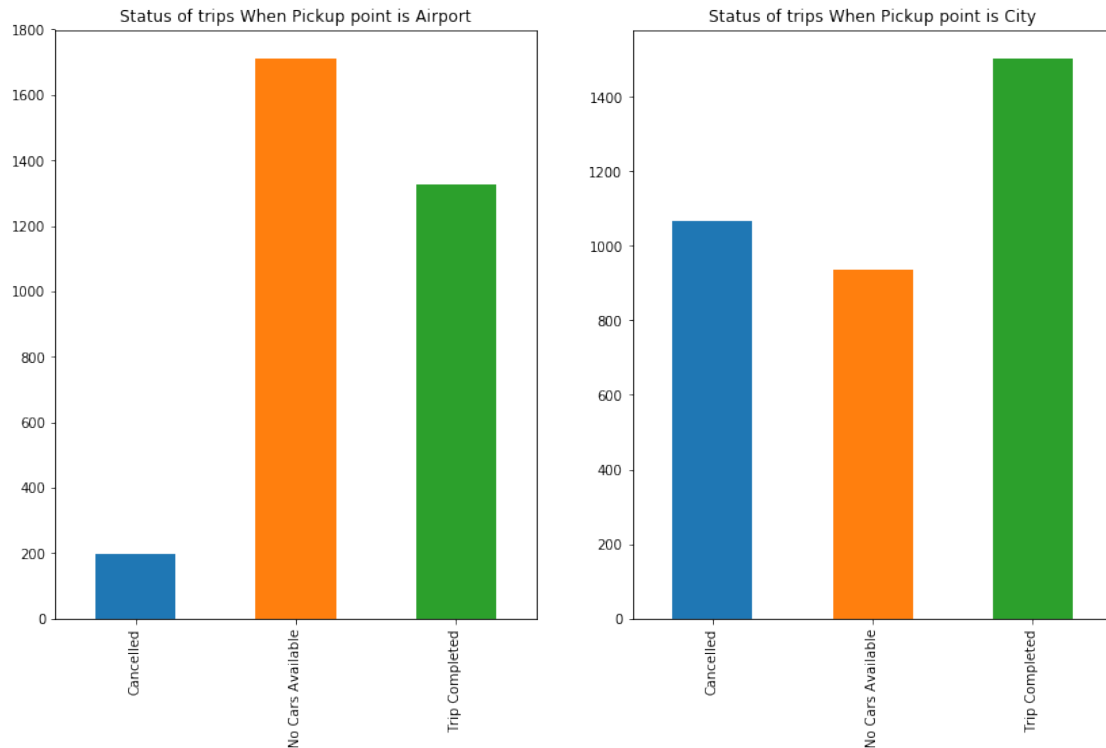
```
In [148]: df2 = df[df['Pickup point']=='Airport']
          airport = df2['Status'].value_counts()
          totalAirport = df2['Pickup point'].value_counts()
          df3 = df[df['Pickup point']=='City']
          city = df3['Status'].value_counts()
          totalcity = df2['Pickup point'].value_counts()
```

*#percentage of trip completed*

```
In [156]: airport.sort_index(inplace=True)
          city.sort_index(inplace=True)
```

```
In [158]: fig = plt.figure(figsize=(14,8))
          ax1 = fig.add_subplot(121)
          ax2 = fig.add_subplot(122)
```

```
airport.plot(kind='bar', ax=ax1, title='Status of trips When Pickup point is Airport')
city.plot(kind='bar', ax=ax2, title='Status of trips When Pickup point is City')
plt.show()
```



The above bar chart shows that frequency of Trips cancelled when pickup Point is City is much more compared to when pickup point is Airport frequency of 'No Cars Available' is more at the Airport as compared to City Number of 'Trips Completed' is more when pickup point is City compared to Airport

**0.1 It Looks Like Cab Drivers dont want to take the taxi's to Airport. which explains that there are more 'No cars Available' at the Airport**

Reason could be > Distance Between Airport and City is much more and makes it less economical( They may be charging high value per kilometer in city compared to airport) > Thier could be a chance that they might return empty handed

**0.2 A good way to resolve this situation could be promoting Shared taxis during critical hours at the airport**

**0.3 Another way could be starting a feature of advanced booking of taxis which will give a glimpse of how many taxis are required at a particular time. It may help in mitigating the problem in advance.**