uber-trips-eda

June 24, 2024

1 Project Title: Uber Trips Analysis

Problem Statement

Uber has a constant imbalance in the demand and supply of rides which promotes poor customer retention. To achieve a balance and ensure there is a sufficient supply of rides to match the demand of customers, we will identify peak hours of the day in the most occurring start locations. This will help to know the locations to dispatch more riders to and what hours to do so. Customer satisfaction and customer retention will increase in such areas, and once this is achieved, it'll mean more profit for Uber.

Project Description

The aim of this project is to carryout an Exploratory Data Analysis (EDA) on an Uber trips dataset from New York to derive insights and patterns on which day(s) have the highest and the lowest trips or the busiest hour. New York has a highly complex transportation system coupled with a large residential populace.

This dataset consists of 1156 rows and 7 columns.

[1]:
```
#Import python libraries for data manipulation and visualization

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

[2]:
```
#Load the data
uber_trips = pd.read_csv('uber_drives.csv')
```

[3]:
```
#Inspect the data
uber_trips.head()
```

[3]:
```
      START_DATE*       END_DATE* CATEGORY*       START*            STOP*  \
0  1/1/2016 21:11  1/1/2016 21:17  Business  Fort Pierce      Fort Pierce
1   1/2/2016 1:25   1/2/2016 1:37  Business  Fort Pierce      Fort Pierce
2  1/2/2016 20:25  1/2/2016 20:38  Business  Fort Pierce      Fort Pierce
3  1/5/2016 17:31  1/5/2016 17:45  Business  Fort Pierce      Fort Pierce
```

```
4  1/6/2016 14:42  1/6/2016 15:49  Business  Fort Pierce  West Palm Beach

   MILES*         PURPOSE*
0     5.1   Meal/Entertain
1     5.0              NaN
2     4.8  Errand/Supplies
3     4.7          Meeting
4    63.7   Customer Visit
```

[4]:
```
uber_trips.tail()
```

[4]:
```
           START_DATE*         END_DATE* CATEGORY*            START*  \
1151  12/31/2016 13:24  12/31/2016 13:42  Business           Kar?chi
1152  12/31/2016 15:03  12/31/2016 15:38  Business  Unknown Location
1153  12/31/2016 21:32  12/31/2016 21:50  Business        Katunayake
1154  12/31/2016 22:08  12/31/2016 23:51  Business           Gampaha
1155            Totals               NaN       NaN               NaN

                 STOP*   MILES*        PURPOSE*
1151  Unknown Location      3.9  Temporary Site
1152  Unknown Location     16.2         Meeting
1153           Gampaha      6.4  Temporary Site
1154         Ilukwatta     48.2  Temporary Site
1155               NaN  12204.7             NaN
```

[5]:
```
#Dimension of the dataset
uber_trips.shape
```

[5]:
```
(1156, 7)
```

[6]:
```
#Size of the dataset
uber_trips.size
```

[6]:
```
8092
```

[7]:
```
uber_trips.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1156 entries, 0 to 1155
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   START_DATE*  1156 non-null   object
 1   END_DATE*    1155 non-null   object
 2   CATEGORY*    1155 non-null   object
 3   START*       1155 non-null   object
 4   STOP*        1155 non-null   object
```

```
 5   MILES*        1156 non-null   float64
 6   PURPOSE*      653 non-null    object
dtypes: float64(1), object(6)
memory usage: 63.3+ KB
```

[8]:
```
#Find null values in dataset
uber_trips.isna().sum()
```

[8]:
```
START_DATE*      0
END_DATE*        1
CATEGORY*        1
START*           1
STOP*            1
MILES*           0
PURPOSE*       503
dtype: int64
```

[9]:
```
# Summary Statistics of original datset
uber_trips.describe()
```

[9]:
```
             MILES*
count   1156.000000
mean      21.115398
std      359.299007
min        0.500000
25%        2.900000
50%        6.000000
75%       10.400000
max    12204.700000
```

Print total number of unique START locations

[10]:
```
start_dest = uber_trips['START*'].dropna()
unique_start = set(start_dest)
unique_start
len(unique_start)
```

[10]:
```
177
```

Print total number of unique STOP locations

[11]:
```
stop_dest = uber_trips['STOP*'].dropna()
unique_stop = set(stop_dest)
len(unique_stop)
```

[11]:
```
188
```

2 Data Cleaning

Some of the data cleaning steps includes: * Dropping missing values (NaN) * Renaming column headers * Splitting the Start_date and End_start into Hour, Day, Day of the week, month and weekday

[12]:
```
uber = uber_trips.copy()
uber.head()
```

[12]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	1/5/2016 17:31	1/5/2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	1/6/2016 14:42	1/6/2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit

[13]:
```
#Confirm if there are any null values
uber.isnull().values.any()
```

[13]:
```
True
```

[14]:
```
# How many missing values are present
uber.isnull().values.sum()
```

[14]:
```
507
```

Dropping NaN values.

[15]:
```
# Drop NAN values
uber.dropna(inplace=True)
uber
```

[15]:

	START_DATE*	END_DATE*	CATEGORY*	START*	…
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	
3	1/5/2016 17:31	1/5/2016 17:45	Business	Fort Pierce	
4	1/6/2016 14:42	1/6/2016 15:49	Business	Fort Pierce	
5	1/6/2016 17:15	1/6/2016 17:19	Business	West Palm Beach	
…	…	…	…	…	
1150	12/31/2016 1:07	12/31/2016 1:14	Business	Kar?chi	
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	

```
1152  12/31/2016 15:03  12/31/2016 15:38  Business  Unknown Location
1153  12/31/2016 21:32  12/31/2016 21:50  Business         Katunayake
1154  12/31/2016 22:08  12/31/2016 23:51  Business            Gampaha

                 STOP*  MILES*         PURPOSE*
0           Fort Pierce     5.1   Meal/Entertain
2           Fort Pierce     4.8  Errand/Supplies
3           Fort Pierce     4.7          Meeting
4       West Palm Beach    63.7   Customer Visit
5       West Palm Beach     4.3   Meal/Entertain
…                   …     …              …
1150            Kar?chi     0.7          Meeting
1151  Unknown Location      3.9   Temporary Site
1152  Unknown Location     16.2          Meeting
1153            Gampaha     6.4   Temporary Site
1154          Ilukwatta    48.2   Temporary Site

[653 rows x 7 columns]
```

[16]:
```
#Renaming original columns for easy referencing
uber.rename(columns = {'START_DATE*':'Start_Date', 'END_DATE*':'End_Date',
                       'CATEGORY*':'Category', 'START*':'Start', 'STOP*':
 ↪'Stop', 'MILES*':'Miles', 'PURPOSE*':'Purpose'}, inplace = True)
```

[17]:
```
uber.head()
```

[17]:
```
       Start_Date        End_Date  Category            Start             Stop  \
0  1/1/2016 21:11  1/1/2016 21:17  Business      Fort Pierce      Fort Pierce
2  1/2/2016 20:25  1/2/2016 20:38  Business      Fort Pierce      Fort Pierce
3  1/5/2016 17:31  1/5/2016 17:45  Business      Fort Pierce      Fort Pierce
4  1/6/2016 14:42  1/6/2016 15:49  Business      Fort Pierce  West Palm Beach
5  1/6/2016 17:15  1/6/2016 17:19  Business  West Palm Beach  West Palm Beach

   Miles          Purpose
0    5.1   Meal/Entertain
2    4.8  Errand/Supplies
3    4.7          Meeting
4   63.7   Customer Visit
5    4.3   Meal/Entertain
```

[18]:
```
#Import datetime and calender
import datetime
import calendar
```

[19]:
```
#Split the start_date and end_date into hour,day,day of the week, month and␣
 ↪weekday
uber['Start_Date'] = pd.to_datetime(uber['Start_Date'], format="%m/%d/%Y %H:%M")
```

```
uber['End_Date'] = pd.to_datetime(uber['End_Date'], format="%m/%d/%Y %H:%M")
hour=[]
day=[]
dayofweek=[]
month=[]
weekday=[]
for x in uber['Start_Date']:
    hour.append(x.hour)
    day.append(x.day)
    dayofweek.append(x.dayofweek)
    month.append(x.month)
    weekday.append(calendar.day_name[dayofweek[-1]])
uber['HOUR']=hour
uber['DAY']=day
uber['DAY_OF_WEEK']=dayofweek
uber['MONTH']=month
uber['WEEKDAY']=weekday
```

[20]:
```
uber.head()
```

[20]:

	Start_Date	End_Date	Category	Start	Stop	Miles	Purpose	HOUR	DAY	DAY_OF_WEEK	MONTH	WEEKDAY
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	21	1	4	1	Friday
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	20	2	5	1	Saturday
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4.7	Meeting	17	5	1	1	Tuesday
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	14	6	2	1	Wednesday
5	2016-01-06 17:15:00	2016-01-06 17:19:00	Business	West Palm Beach	West Palm Beach	4.3	Meal/Entertain	17	6	2	1	Wednesday

- The table above displays splitting of the Start_date and End_start into Hour, Day, Day of the week, month and weekday

[21]:
```
uber.dtypes
```

[21]:
```
Start_Date      datetime64[ns]
End_Date        datetime64[ns]
Category                object
Start                   object
Stop                    object
Miles                  float64
Purpose                 object
HOUR                     int64
DAY                      int64
DAY_OF_WEEK              int64
MONTH                    int64
WEEKDAY                 object
dtype: object
```

2.1 INSIGHTS

What is the most popular start destination for the uber drivers?

[22]:
```
#Get the start destination and the count
start_point = uber['Start']
df = pd.DataFrame(start_point.value_counts())
df.sort_values(['Start'], ascending=False)

df = df.reset_index()
df = df.rename(columns = {'index':'Start Destination', 'Start': 'Count'})
df.loc[df['Count'] == max(df['Count'])]
```

[22]:

	Start Destination	Count
0	Cary	161

- Cary city happens to be the most popular start destination.

What is the most popular stop/drop off destination for the uber drivers?

[23]:
```
#Get the stop destination and count
stop_point = uber['Stop']
df = pd.DataFrame(stop_point.value_counts())
df.sort_values(['Stop'], ascending=False)

df = df.reset_index()
df = df.rename(columns = {'index':'Stop Destination', 'Stop': 'Count'})
df.loc[df['Count'] == max(df['Count'])]
```

[23]:

	Stop Destination	Count
0	Cary	155

- Cary is also the location where people drop off the most.

What is the most frequent route taken by Uber drivers

[24]:
```
#Count of trips for the most frequest route
df = uber
df = pd.DataFrame(df.groupby(['Start', 'Stop']).size())
df = df.rename(columns = {0:'Count'})
df = df.sort_values(['Count'], ascending = False)
df.loc[df['Count'] == max(df['Count'])]
```

[24]:
```
                  Count
Start Stop
Cary  Morrisville     52
```

- The most frequented route for people who travel is from Cary to Morrisville.

Get the types of purpose and the mileage covered

[25]:
```
#find out the number of miles for eaach trip purpose
uber_df = uber['Miles'].groupby(uber['Purpose']).sum().sort_values(ascending =␣
 ↪False)
uber_df
```

[25]:
```
Purpose
Meeting            2851.3
Customer Visit     2089.5
Meal/Entertain      911.7
Temporary Site      523.7
Errand/Supplies     508.0
Between Offices     197.0
Commute             180.2
Moving               18.2
Airport/Travel       16.5
Charity ($)          15.1
Name: Miles, dtype: float64
```

- This gives us a hint as to what reasons people use Uber services; apparently, its mostly used for meeting trips.

[26]:
```
#Display top 10 start and stop locations
Start_Point = uber.groupby(['Start']).Stop.value_counts().nlargest(10)
Start_Point
```

[26]:
```
Start             Stop
Cary              Morrisville         52
Morrisville       Cary                51
Cary              Cary                44
                  Durham              30
Unknown Location  Unknown Location    30
Durham            Cary                29
Kar?chi           Kar?chi             20
```

```
Cary              Raleigh           17
Lahore            Lahore            16
Raleigh           Cary              15
Name: Stop, dtype: int64
```

- Displaying top 10 popular start and stop locations.

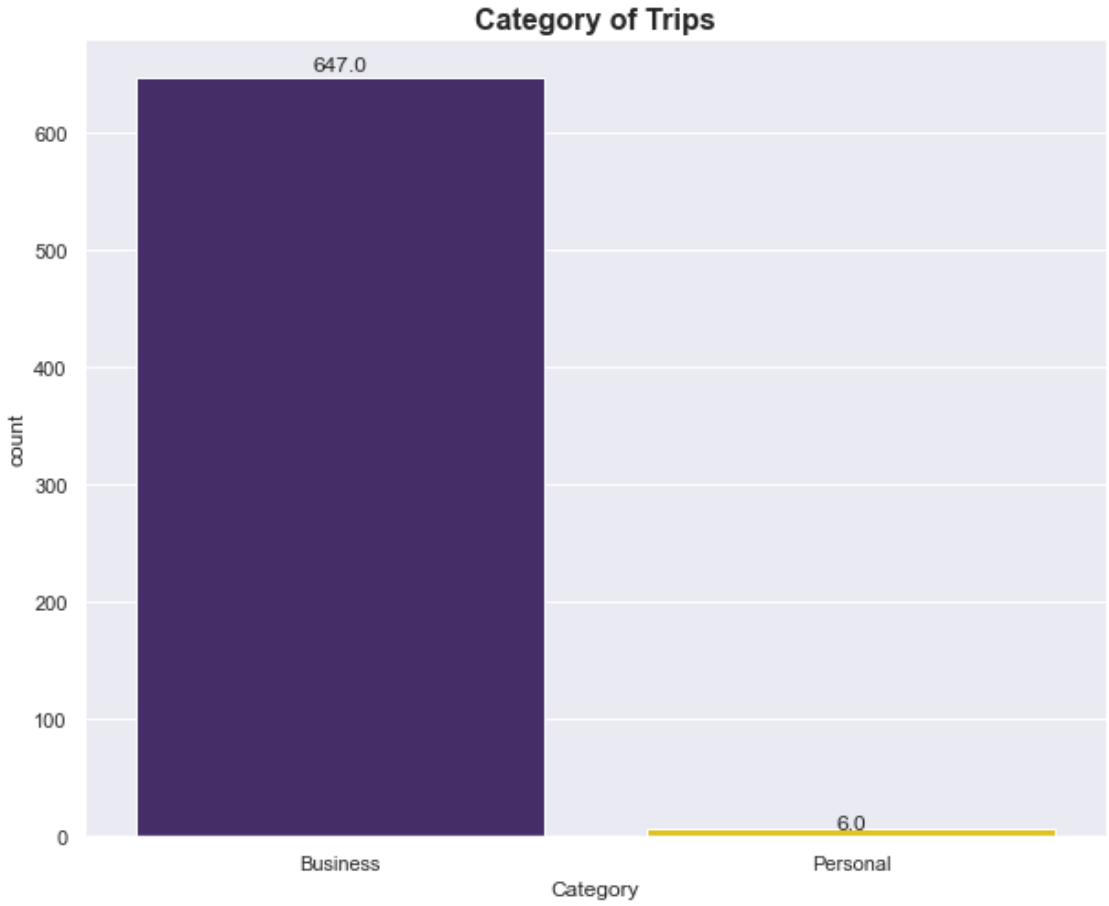
3 Data Visualization

3.1 Univariate Exploration

[159]:
```
#function to display values on each plot
def show_values(axs, orient="v", space=.01):
    def _single(ax):
        if orient == "v":
            for p in ax.patches:
                _x = p.get_x() + p.get_width() / 2
                _y = p.get_y() + p.get_height() + (p.get_height()*0.01)
                value = '{:.1f}'.format(p.get_height())
                ax.text(_x, _y, value, ha="center")
        elif orient == "h":
            for p in ax.patches:
                _x = p.get_x() + p.get_width() + float(space)
                _y = p.get_y() + p.get_height() - (p.get_height()*0.5)
                value = '{:.1f}'.format(p.get_width())
                ax.text(_x, _y, value, ha="left")

    if isinstance(axs, np.ndarray):
        for idx, ax in np.ndenumerate(axs):
            _single(ax)
    else:
        _single(axs)
```

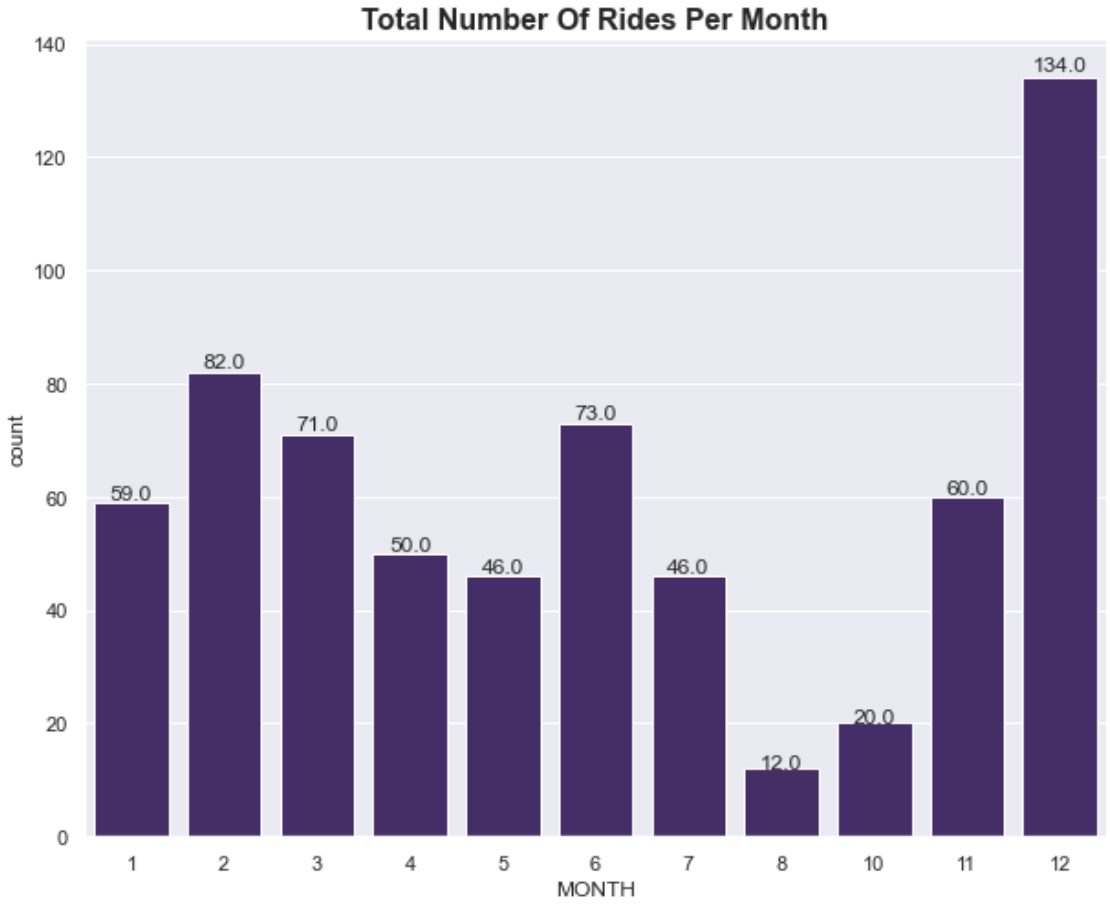
[160]:
```
#plot for category column
a = sns.countplot(x='Category', data=uber, palette=['#432371', 'gold'])
plt.title('Category of Trips', weight='bold').set_fontsize('16')
show_values(a)
```



- The Uber usage categories can be grouped into 2; the Business and Personal category. From the chart, it can be gleaned that most people patronize uber for Business.

3.1.1 What is the total number of rides per month?

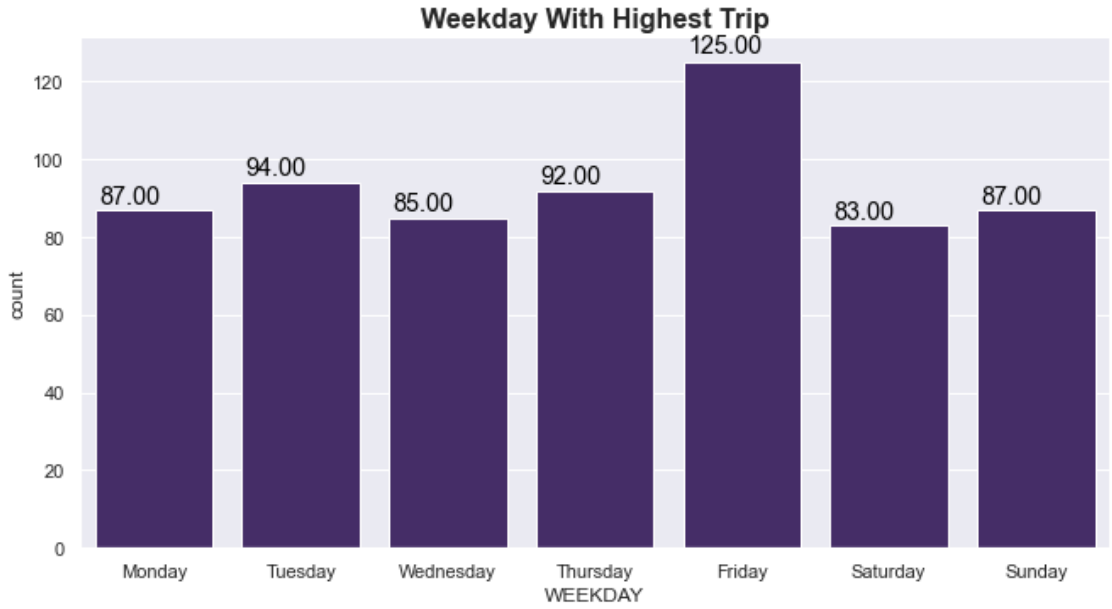
[182]:
```
#Plot for number of trips in a month
a = sns.countplot(x='MONTH', data=uber, palette=['#432371'])
plt.title('Total Number Of Rides Per Month', weight='bold').set_fontsize('16')
show_values(a)
```



- Based on the chart, there were more rides in the month of December.

[187]:
```
#plot to see the weekday with highest trip

g=sns.catplot(data=uber, x="WEEKDAY", kind="count", palette=sns.
 ↪color_palette(['#432371']), aspect=15/8,order=order)
plt.title('Weekday With Highest Trip', weight='bold').set_fontsize('16')
#function to display values
ax = g.facet_axis(0,0)
for p in ax.patches:
    ax.text(p.get_x() + 0.025,
            p.get_height() * 1.02,
            '{0:.2f}'.format(p.get_height()),
            color='black', rotation='horizontal', size='large')
order = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",␣
 ↪"Saturday","Sunday"]
```

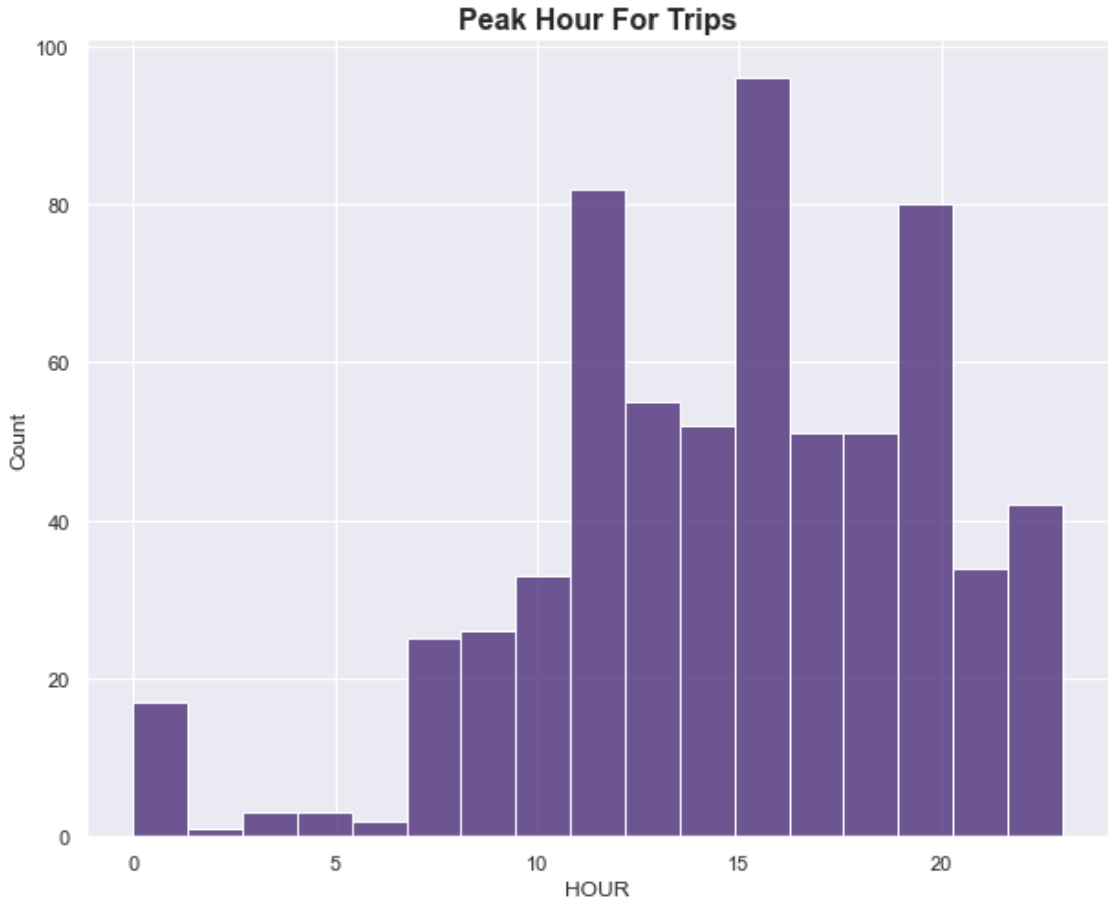


- Friday is the day with the highest trips every week.

[64]:
```
#Number of trips for each day of the week
uber.WEEKDAY.value_counts()
```

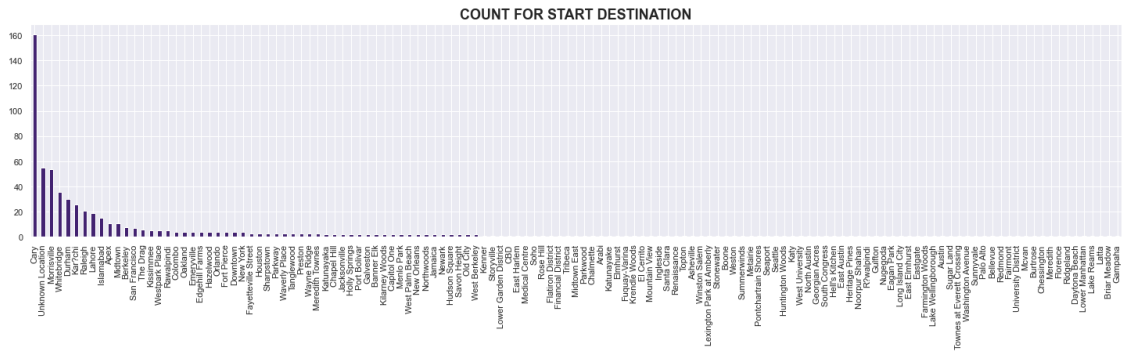
[64]:
```
Friday       125
Tuesday       94
Thursday      92
Sunday        87
Monday        87
Wednesday     85
Saturday      83
Name: WEEKDAY, dtype: int64
```

[193]:
```
sns.histplot(x='HOUR',data=uber,color=['#432371']).set(title='HOUR MOST PEOPLE␣
↪USE UBER')
sns.set(rc={'figure.figsize':(10,8)})
plt.title('Peak Hour For Trips', weight='bold').set_fontsize('16')
```



- Within the hours of 1pm - 4pm, the drivers appear to be busiest.

[128]:
```
#plot to check travel milage for each start destination
uber['Start'].value_counts().plot(kind='bar', figsize=(25,5),color='#432371')
plt.title('COUNT FOR START DESTINATION', weight='bold').set_fontsize('18')
```



The highest number of people which use Uber are located in Cary city. So we need to ensure there

is provision for enough cabs in that city. Also, more should be invested in advertisement to improve patronage.

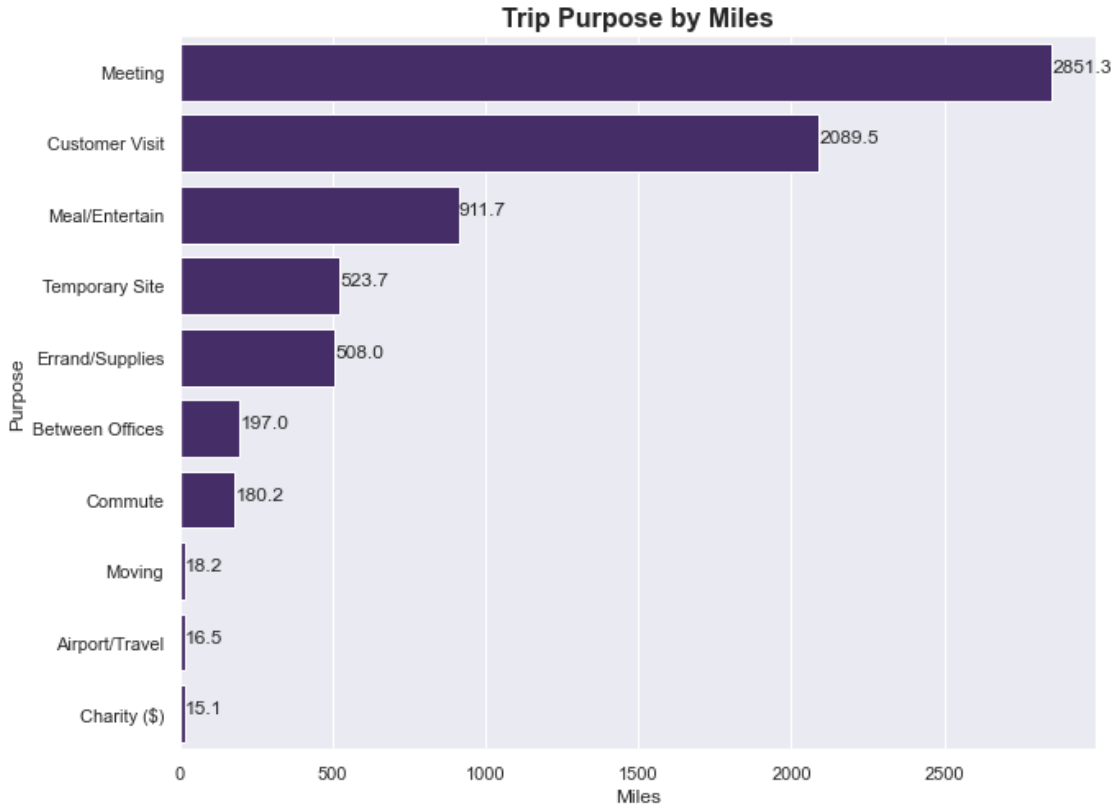
[34]:
```
uber.Start.value_counts()
```

[34]:
```
Cary                161
Unknown Location     55
Morrisville          54
Whitebridge          36
Durham               30
                    ...
Lower Manhattan       1
Lake Reams            1
Latta                 1
Briar Meadow          1
Gampaha               1
Name: Start, Length: 131, dtype: int64
```

3.2 Bivariate Exploration

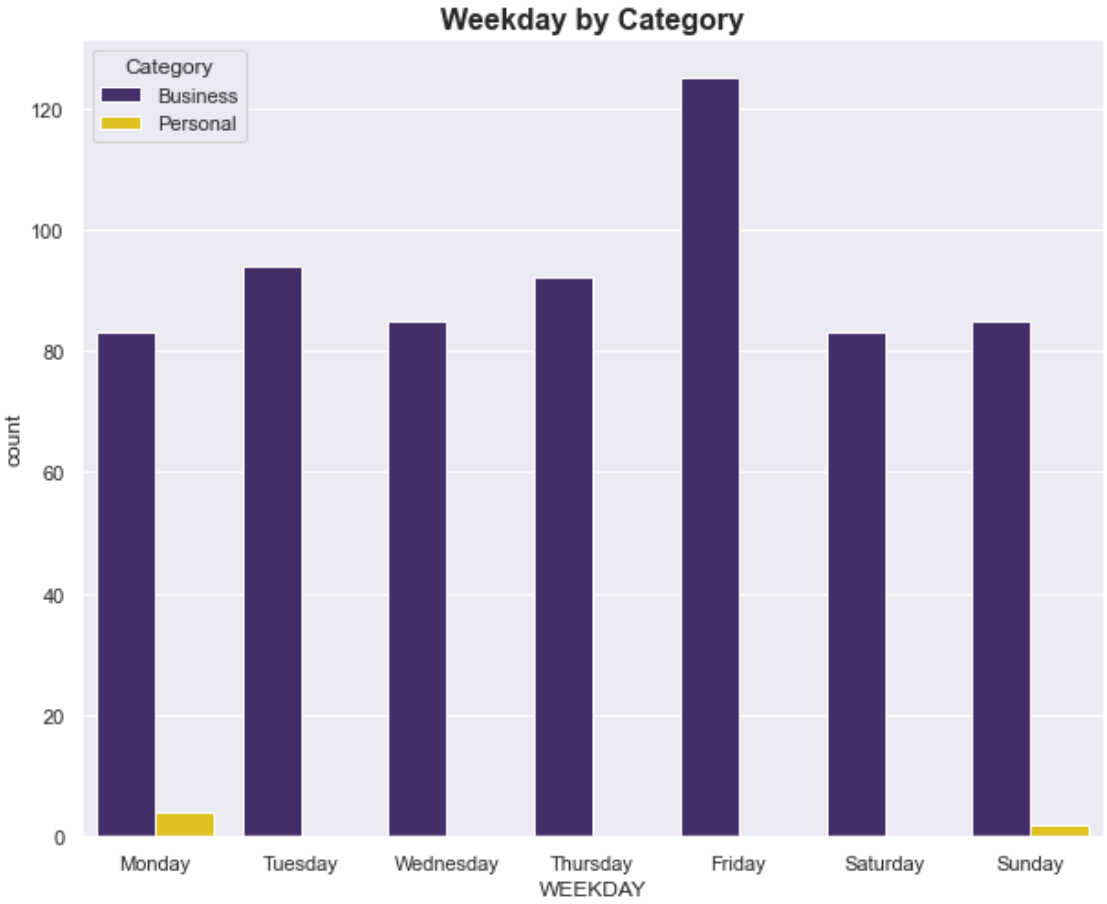
3.2.1 Purpose of trips vs distance

[179]:
```
#Purpose of trips against miles
df = uber_df.reset_index()
a = sns.barplot(x=df['Miles'], y=df['Purpose'], palette=['#432371'])
plt.title('Trip Purpose by Miles', weight='bold').set_fontsize('16')
show_values(a, "h", space=0)
```



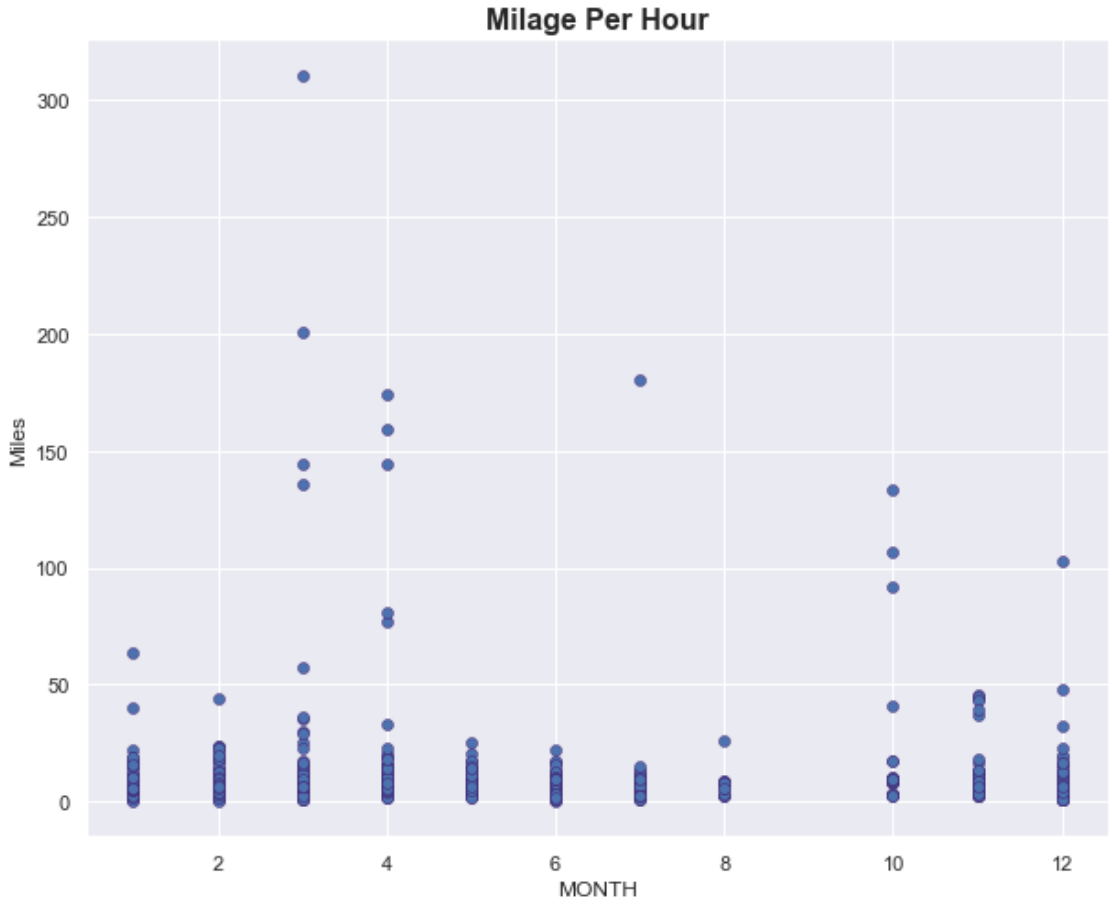
- Most people use Uber for Meetings followed by customer visit.

[190]:
```
#Plot of weekday against Category
order = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",␣
 ↪"Saturday","Sunday"]
sns.countplot(x='WEEKDAY', data=uber, hue='Category', order=order,␣
 ↪palette=['#432371','gold']).set_title('Weekday by Category', weight='bold').
 ↪set_fontsize('16')
```



- The weekdays is predominantly replete with business rides, together with pockets of personal transits on Mondays and Sundays.

[192]:
```
#Plot of milage per hour
sns.scatterplot(x='MONTH', y='Miles',data=uber,palette = "#432371",␣
↪edgecolor='#432371')
plt.title('Milage Per Hour', weight='bold').set_fontsize('16')
```

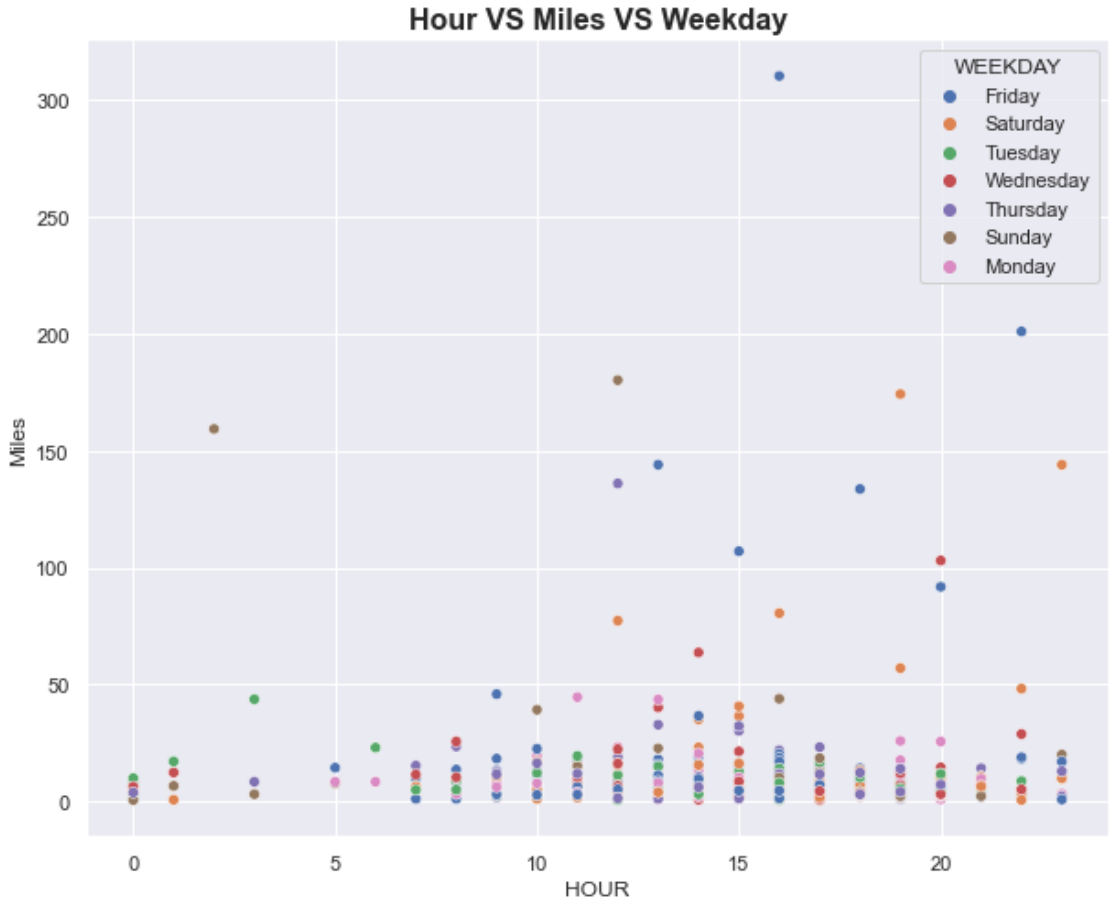


- The highest milage recorded by Uber was in the 3rd month - March.

3.3 Multivariate Analysis

3.3.1 What is the relationship between the hour of the day, distance traveled and day of the week?

[82]:
```
#Plot for hour vs weekday vs miles
sns.scatterplot(x='HOUR', y='Miles',hue='WEEKDAY',data=uber, palette='deep')
plt.title('Hour VS Miles VS Weekday', weight='bold').set_fontsize('16')
```



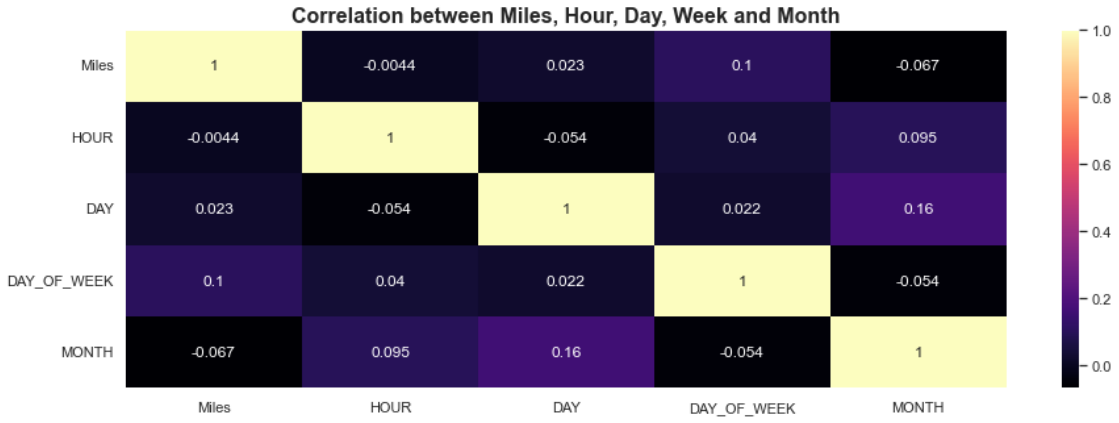
- It appears that the highest milage occured by 4pm on a Friday and from the previous plot of milage per hour we can deduce that this occurred in the month of March. This tallies with our univariate hour histplot.

[39]:
```
#Correlation for numeric columns
uber.corr()
```

[39]:

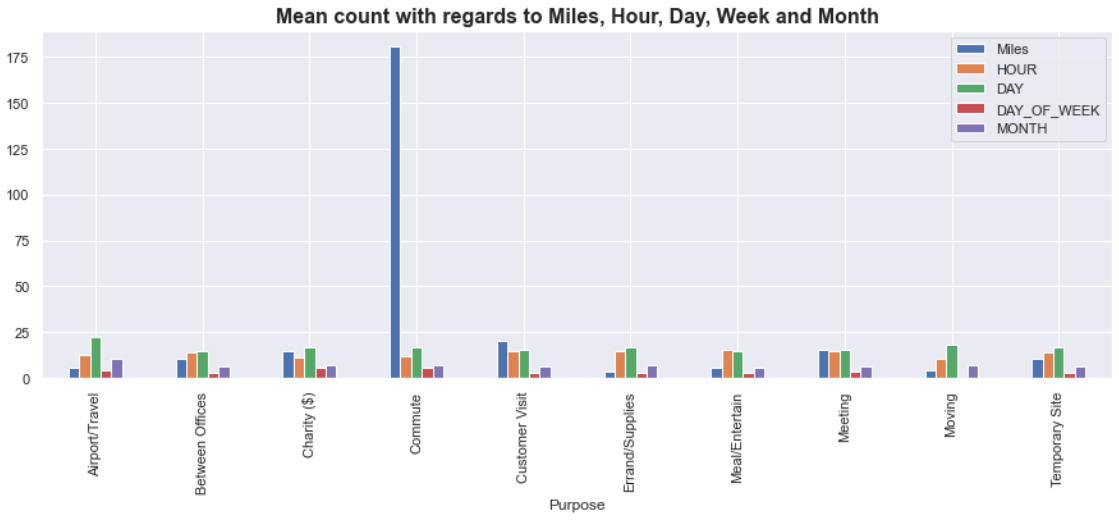
	Miles	HOUR	DAY	DAY_OF_WEEK	MONTH
Miles	1.000000	-0.004370	0.022724	0.104710	-0.067137
HOUR	-0.004370	1.000000	-0.053598	0.039669	0.095491
DAY	0.022724	-0.053598	1.000000	0.021694	0.160147
DAY_OF_WEEK	0.104710	0.039669	0.021694	1.000000	-0.054480
MONTH	-0.067137	0.095491	0.160147	-0.054480	1.000000

[195]:
```
#Correlation plot between Miles, Hour, Day, Day_of_week and Month
plt.figure(figsize = [15, 5])
sns.heatmap(uber.corr(),cmap='magma', annot=True)
plt.title('Correlation between Miles, Hour, Day, Week and Month',␣
 ↪weight='bold').set_fontsize('16')
```



- The correlation that exist between miles, hour, weekday, day and month is either low negative correlation or low positive correlation.

[194]:
```
uber.groupby('Purpose').mean().plot(kind='bar',figsize=(15,5))
plt.title('Mean count with regards to Miles, Hour, Day, Week and Month',␣
↪weight='bold').set_fontsize('16')
```



4 Summary

1. The highest pick-up point location is Cary.
2. The day of the week with most patronage is Friday, and the least is Saturday.
3. 1pm - 4pm is the time with the highest surge.
4. Most of the start & stop transit is between Cary and Morrisville.
5. Most trips are for business.

6. It appears that the highest milage that occured in the month of March was on a Friday, and on the 4pm mark. This tallies with our univariate hour histplot.