UBER DRIVE 2016 ANALYSIS

INTM572: Data Exploration and Preparation

Mehak Gupta
Registration Number: 12221647
Section: Q2255
Roll Number: RQ2255B41

Student Declaration And Teacher's Remarks

Declaration:

I declare that this Assignment is my individual work. I have not copied it from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for me by any other person.

Evaluator's Signature and Date:

General Observations	Suggestions for Improvement	Best Part of the Assignment

Marks Obtained: _____ Max. Marks: _____

INDEX:

01

INTRODUCTION

INTRODUCTION

Uber Technologies Inc provides ride hailing services. The Company develops applications for road transportation, navigation, ride sharing, and payment processing solutions. Uber Technologies serves customers worldwide.

02

DATA UNDERSTANDING AND CLEANSING

UNDERSTANDING THE DATA

Mehak_df.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
```

The following can the inferred:

- The total sample sets in the data are 1156, with some of them being 1155 indicating the existence of null samples. (number of rows)
- The total number of features is 7. (number of columns)
- Feature "purpose" has less sample sets than others which means that it has a lot of null values.
- There is only 1 float data type and 6 of object.
- The data-type of start date and end date should be datetime, however, it is showing it as string (object).

CLEANSING THE DATA

1. Checking the head and tail of the data frame.

Mehak_df.head()

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	01-01-2016 21:11	01-01-2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	01-02-2016 01:25	01-02-2016 01:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	01-02-2016 20:25	01-02-2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	01-05-2016 17:31	01-05-2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	01-06-2016 14:42	01-06-2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit

Mehak_df.tail()

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/2016 15:03	12/31/2016 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	12/31/2016 21:32	12/31/2016 21:50	Business	Katunayake	Gampaha	6.4	Temporary Site
1154	12/31/2016 22:08	12/31/2016 23:51	Business	Gampaha	Ilukwatta	48.2	Temporary Site
1155	Totals	NaN	NaN	NaN	NaN	12204.7	NaN

Since the last row is unnecessary and irrelevant for the data analysis, the following code is used to delete it

```
#deleting unnecessary data aka the last row
Mehak_df=Mehak_df[:-1]
Mehak_df.tail()
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1150	12/31/2016 1:07	12/31/2016 1:14	Business	Kar?chi	Kar?chi	0.7	Meeting
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/2016 15:03	12/31/2016 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	12/31/2016 21:32	12/31/2016 21:50	Business	Katunayake	Gampaha	6.4	Temporary Site
1154	12/31/2016 22:08	12/31/2016 23:51	Business	Gampaha	Ilukwatta	48.2	Temporary Site

CLEANSING THE DATA

2. Checking and deleting duplicate records.

The duplicate records can lead to inaccuracy in the analysis of data, thus, the same should be deleted from the data set to increase the appropriateness of the analysis done.

The uber dataset had one duplicate record at row number 492 which has been deleted.

```
#checking for duplicate records
Mehak_df[Mehak_df.duplicated()]
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
492	6/28/2016 23:34	6/28/2016 23:59	Business	Durham	Cary	9.9	Meeting

```
#deleting the duplicate record
Mehak_df.drop_duplicates(inplace=True)
```

```
#getting information about the dataset
Mehak_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1154 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   START_DATE*  1154 non-null   object
 1   END_DATE*    1154 non-null   object
 2   CATEGORY*    1154 non-null   object
 3   START*       1154 non-null   object
 4   STOP*        1154 non-null   object
 5   MILES*       1154 non-null   float64
 6   PURPOSE*     652 non-null    object
dtypes: float64(1), object(6)
memory usage: 72.1+ KB
```

CLEANSING THE DATA

3. Fixing the data types

The data types for start date and end date should be "datetime" but it is shown as string (object) in the dataset.

Firstly, we have to define the column names and then use the appropriate code to change the data type as shown.

```
Mehak_df.columns = ['START_DATE', 'END_DATE', 'CATEGORY', 'START', 'STOP', 'MILES', 'PURPOSE']
print(Mehak_df.columns);
```

```
Index(['START_DATE', 'END_DATE', 'CATEGORY', 'START', 'STOP', 'MILES',
       'PURPOSE'],
      dtype='object')
```

```
#the start and end dates should be in date format, however, they are shown as object.
#this needs to be fixed before proceeding with the data analysis
Mehak_df['START_DATE'] = pd.to_datetime(Mehak_df['START_DATE'])
Mehak_df['END_DATE'] = pd.to_datetime(Mehak_df['END_DATE'])
Mehak_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1154 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   START_DATE  1154 non-null   datetime64[ns]
 1   END_DATE    1154 non-null   datetime64[ns]
 2   CATEGORY    1154 non-null   object        
 3   START       1154 non-null   object        
 4   STOP        1154 non-null   object        
 5   MILES       1154 non-null   float64       
 6   PURPOSE     652 non-null    object        
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 72.1+ KB
```

CLEANSING THE DATA

4. Removing the cancelled rides and/or the null samples

The rows where the start date and time is the same as end date and time are either the cancelled rides or the samples where the values are null. Having these in the data set might give inaccurate analysis, hence, a new data frame called "Adj" has been created to proceed with the analysis part after dropping the null or cancelled rows.

```
#checking for cancelled rides or rides where the duration is zero
filtered_data = Mehak_df[Mehak_df["END_DATE"]==Mehak_df["START_DATE"]]
print(filtered_data)
```

```
          START_DATE         END_DATE CATEGORY          START  \
751 2016-09-06 17:49:00 2016-09-06 17:49:00  Business  Unknown Location
761 2016-09-16 07:08:00 2016-09-16 07:08:00  Business  Unknown Location
798 2016-10-08 15:03:00 2016-10-08 15:03:00  Business          Karachi
807 2016-10-13 13:02:00 2016-10-13 13:02:00  Business        Islamabad

              STOP  MILES PURPOSE
751  Unknown Location   69.1     NaN
761  Unknown Location    1.6     NaN
798          Karachi    3.6     NaN
807        Islamabad    0.7     NaN
```

```
#creating a new dataframe for analysis pruposes
AdjMehak_df = Mehak_df
```

```
#dropping the rows where the rides are cancelled
AdjMehak_df.drop([751,761,798,807], axis=0, inplace=True)
```

```
AdjMehak_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1150 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   START_DATE  1150 non-null   datetime64[ns]
 1   END_DATE    1150 non-null   datetime64[ns]
 2   CATEGORY    1150 non-null   object
 3   START       1150 non-null   object
 4   STOP        1150 non-null   object
 5   MILES       1150 non-null   float64
 6   PURPOSE     652 non-null    object
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 71.9+ KB
```

03

QUESTIONS:

THE DATASET CAN ANSWER THE FOLLOWING:

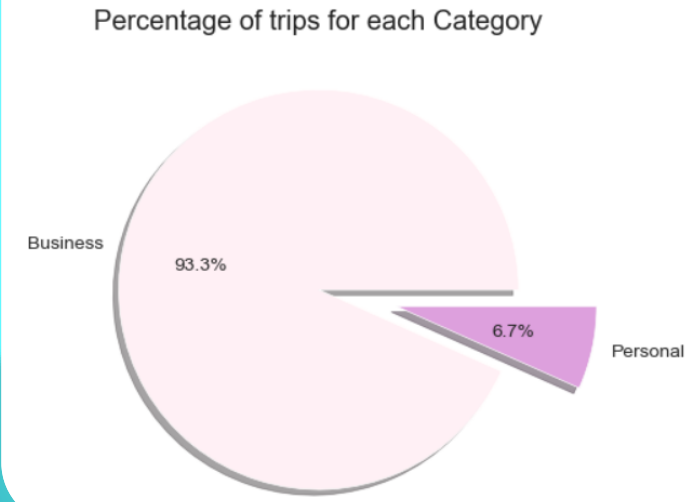
1. What percentage of trips were taken for business and personal reasons respectively?
2. How many trips were taken each month and what is the peak hour of usage?
3. For what purpose the maximum trips were taken?
4. What are the most popular pick-up and drop destinations?
5. What is the average duration and distance for different purposes of the trips?
6. What is the average speed?
7. How many rides were cancelled?

04

VISAULIZATIONS:

PERCENTAGE OF TRIPS FOR EACH CATEGORY

```
#percentage for each category
category_value = AdjMehak_df['CATEGORY'].value_counts()
labels_category=category_value.index
explode = (0,0.4)
colors = ('lavenderblush','plum')
plt.pie(category_value, labels=labels_category, explode = explode, colors = colors, autopct='%1.1f%%', shadow=True)
plt.title("Percentage of trips for each Category");
```



93.3% of the trips were for business purposes while 6.7% were for personal reasons.

```
Category_labels = AdjMehak_df.CATEGORY.value_counts()
print(Category_labels);
```

```
Business    1073
Personal      77
Name: CATEGORY, dtype: int64
```

It shows that out of the non-cancelled trips, 1073 were for business while the rest were for personal reasons.

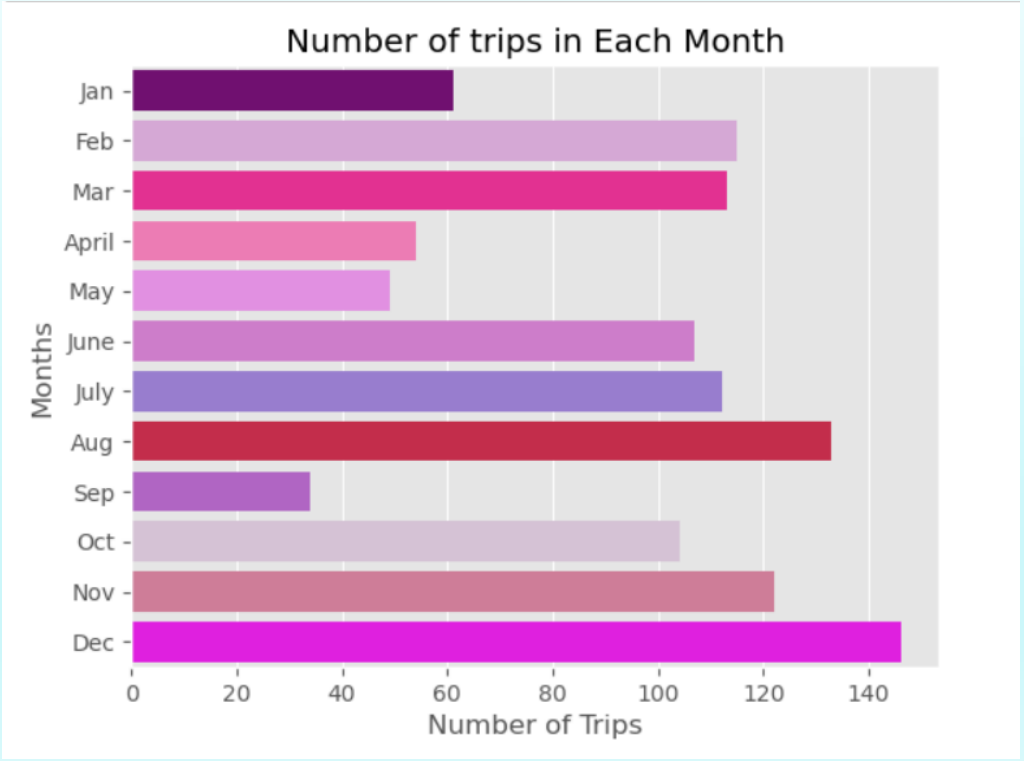
Monthly and Hourly Trips

For getting the monthly trips, the months has been extracted first and then the same has been plotted on a bargraph.
It can be seen that the maximum trips were made in December.

```
#extracting month from start date
AdjMehak_df['MONTH'] = pd.DatetimeIndex(AdjMehak_df['START_DATE']).month
month_label = {1.0: 'Jan', 2.0: 'Feb', 3.0: 'Mar', 4.0: 'April', 5.0: 'May', 6.0: 'June', 7.0: 'July', 8.0: 'Aug', 9.0: 'Sep',
    10.0: 'Oct', 11.0: 'Nov', 12.0: 'Dec' }
AdjMehak_df['MONTH'] = AdjMehak_df.MONTH.map(month_label)
AdjMehak_df.MONTH.unique()
```

```
array(['Jan', 'Feb', 'Mar', 'April', 'May', 'June', 'July', 'Aug', 'Sep',
       'Oct', 'Nov', 'Dec'], dtype=object)
```

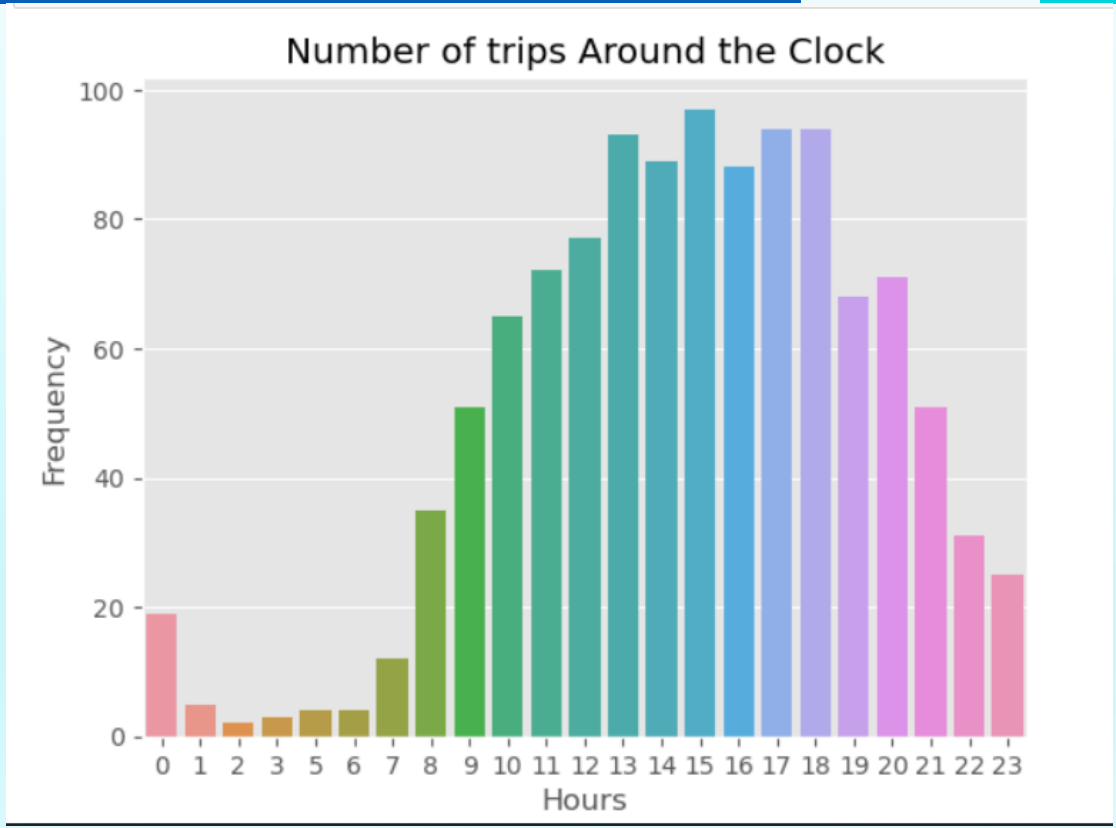
```
# plot number of trips at each month
month_count =AdjMehak_df.MONTH.value_counts()
colors = ('purple','plum','deeppink','hotpink','violet','orchid','mediumpurple','crimson','mediumorchid','thistle','palevioletred','magenta')
sns.barplot( x=month_count, y=month_count.index, order=['Jan','Feb','Mar','April','May','June','July','Aug','Sep','Oct','Nov','Dec'], palette = colors);
plt.xlabel('Number of Trips')
plt.ylabel('Months')
plt.title('Number of trips in Each Month');
```



Monthly and Hourly Trips

For getting the hourly trips, the following code is used.
It can be seen that the maximum trips were made around 3 pm

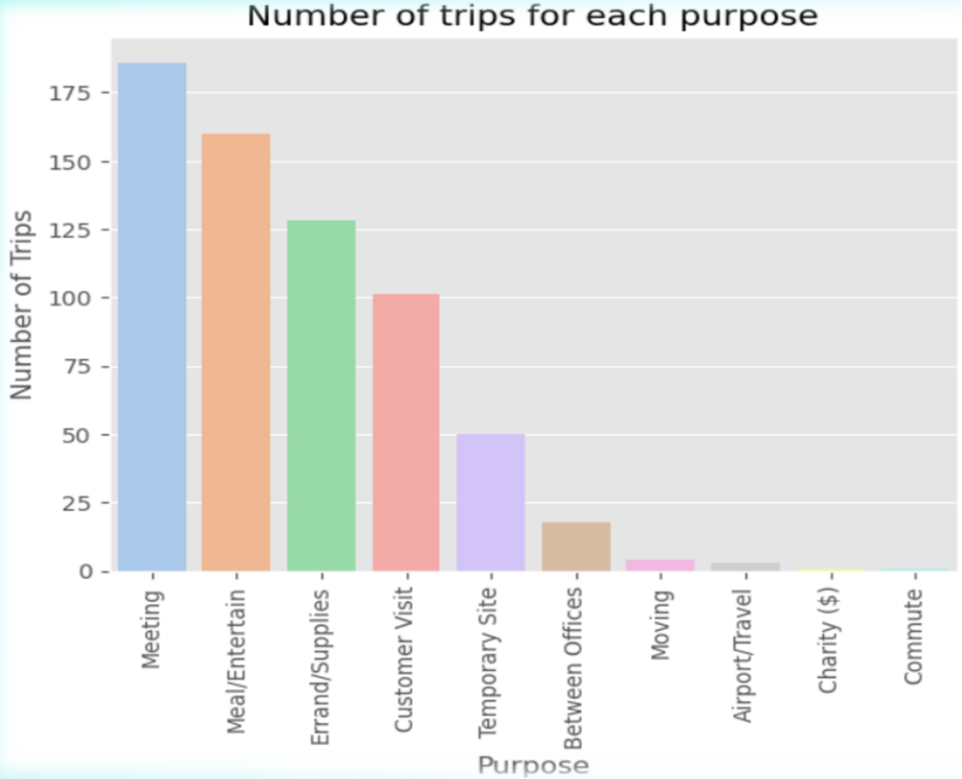
```
# I need to see how many trip made at each clock and as you see the clock which has the higest number of trips
hours = AdjMehak_df['START_DATE'].dt.hour.value_counts()
sns.barplot(x=hours.index, y=hours)
plt.xlabel('Hours')
plt.ylabel('Frequency')
plt.title('Number of trips Around the Clock');
```



Trips for each purpose

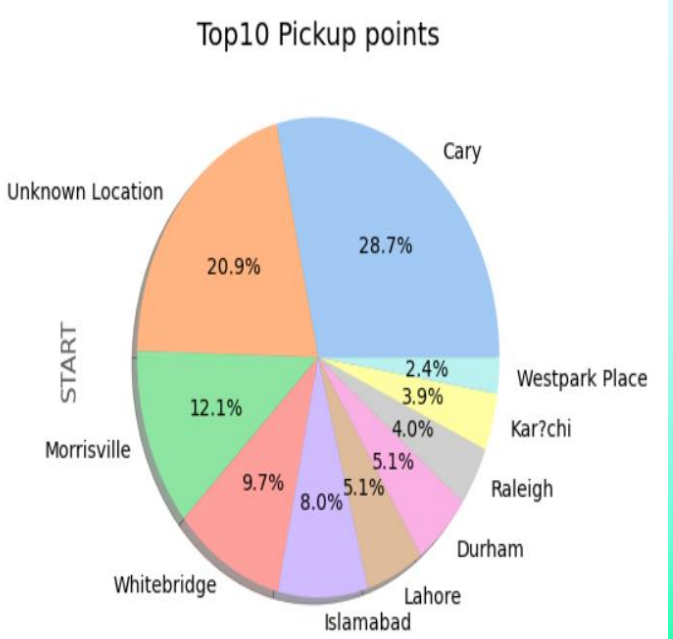
From the graph, it can be seen that the maximum trips were made for meeting purposes.

```
#number of trips for each purpose
purpose_labels = AdjMehak_df.PURPOSE.value_counts()
sns.barplot(x=purpose_labels.index, y=purpose_labels)
plt.xlabel('Purpose')
plt.ylabel('Number of Trips')
plt.title('Number of trips for each purpose');
plt.xticks(rotation=90);
```



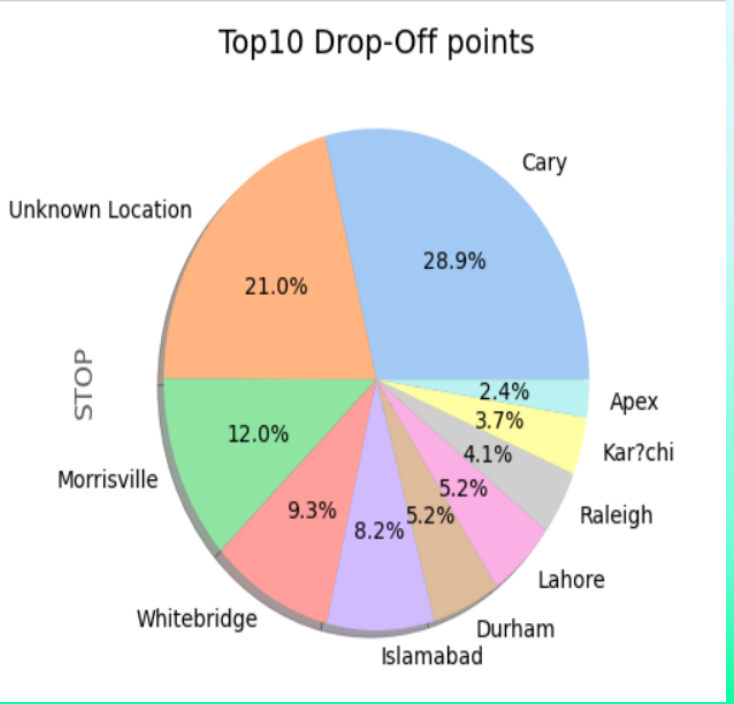
Popular pick up and drop off spots

```
months = AdjMehak_df['START'].value_counts().nlargest (10)
months.plot(kind='pie',autopct='%1.1f%%', shadow=True)
plt.title('Top10 Pickup points');
```



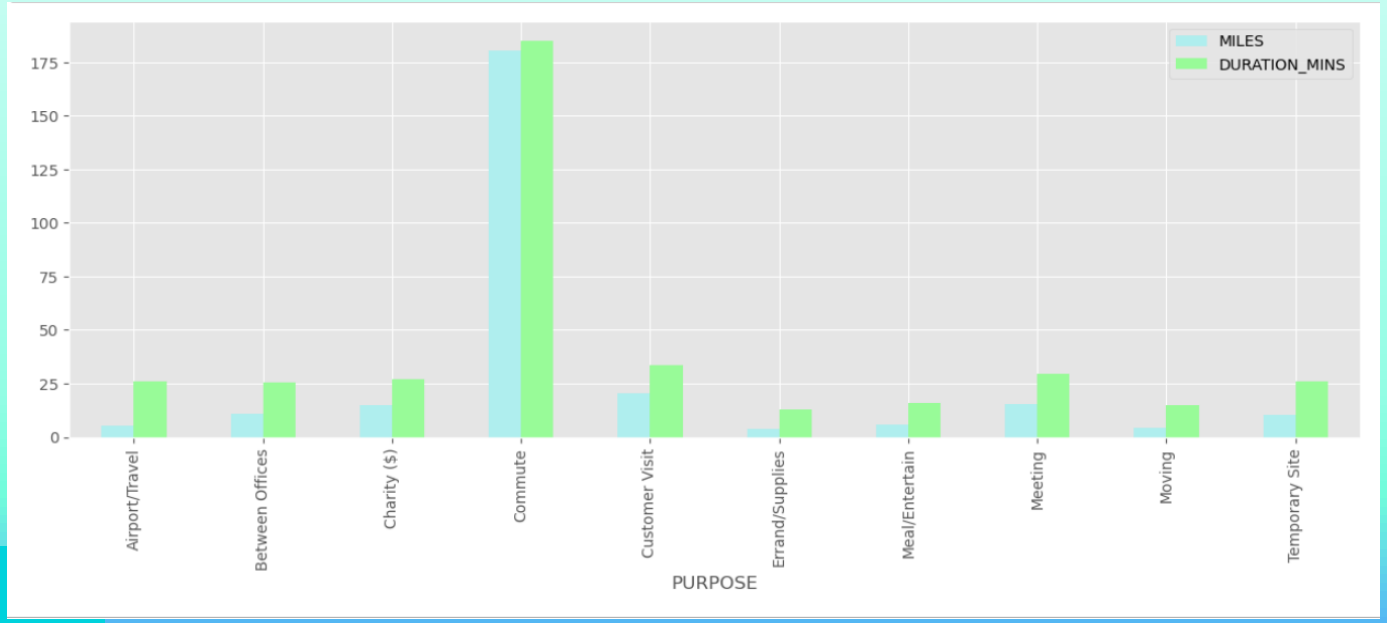
The most popular pick and drop off point is Cary, followed by unknown locations and Morrisville.

```
months = AdjMehak_df['STOP'].value_counts().nlargest (10)
months.plot(kind='pie',autopct='%1.1f%%', shadow=True)
plt.title('Top10 Drop-Off points');
```



Average Duration Distance for each purpose

```
#average time and distance sorted by purpose
purpose = AdjMehak_df.groupby('PURPOSE').mean()
purpose.plot(kind = 'bar',figsize=(15,5), color = ('paleturquoise', 'palegreen'));
```



Minimum, maximum and average speed

After computing the speed (as distance/time)
and making a column for it as Speed_km, the minimum, maximum and average speed has been computed in miles per hour

```
# calculate trip speed for each driver
AdjMehak_df['DURATION_HOURS'] = AdjMehak_df['DURATION_MINS']/60
AdjMehak_df['SPEED_KM'] = AdjMehak_df['MILES']/AdjMehak_df['DURATION_HOURS']
AdjMehak_df['SPEED_KM']
```

```
0       51.000000
1       25.000000
2       22.153846
3       20.142857
4       57.044776
          ...
1150     6.000000
1151    13.000000
1152    27.771429
1153    21.333333
1154    28.077670
Name: SPEED_KM, Length: 1150, dtype: float64
```

```
MinSpeed=AdjMehak_df['SPEED_KM'].min()
print(MinSpeed);
```

```
3.9173553719008267
```

```
AverageSpeed = AdjMehak_df['SPEED_KM'].mean()
print(AverageSpeed)
```

```
26.810348365851866
```

```
MaxSpeed=AdjMehak_df['SPEED_KM'].max()
print(MaxSpeed)
```

```
906.0
```

Number Of Cancelled trips

The cancelled trips are assumed to be the ones whose start and end date – time is same. The same has been filtered out using the given code and then has been counted.

As evident from the given image, the number of cancelled trips were 4 during the year.

```
#checking for cancelled rides or rides where the duration is zero
filtered_data = Mehak_df[Mehak_df["END_DATE"]==Mehak_df["START_DATE"]]
print(filtered_data)
```

```
      START_DATE        END_DATE  CATEGORY          START  \
751  09-06-2016 17:49  09-06-2016 17:49  Business  Unknown Location
761   9/16/2016 7:08    9/16/2016 7:08  Business  Unknown Location
798  10-08-2016 15:03  10-08-2016 15:03  Business         Karachi
807  10/13/2016 13:02  10/13/2016 13:02  Business       Islamabad

           STOP  MILES PURPOSE
751  Unknown Location   69.1     NaN
761  Unknown Location    1.6     NaN
798         Karachi    3.6     NaN
807       Islamabad    0.7     NaN
```

```
len(filtered_data)
```

```
4
```

05

CONCLUSION:

Conclusion

This activity helped us understand the service pattern of an average Uber user. It helps us explore its behaviours and opens a window for us to understand the user's needs. With this new found understanding of ours, we can consult the company in a more accurate manner.

The following is observed by us -

- 93.3% of the trips were for business purposes while 6.7% were for personal reasons, of which maximum were taken for meetings and least for charity.
- Maximum trips were made in December and most rides were hailed in the afternoon at 3pm.
- Most popular pickup spot and drop-off is Carry followed by unknown locations.
- In total 4 rides were cancelled during the year
- The Jupyter notebook can be downloaded by clicking here.
- PDF can be downloaded here

Thank you!