Assignment on Uber dataset

Load the dataset

In [1]:
```
import pandas as pd
ud=pd.read_csv("Uber.csv")
type(ud)

#to load a csv file and view its type
```

Out[1]:
```
pandas.core.frame.DataFrame
```

Display basic info about dataset

In [2]:
```
ud.describe

#to get the complete description of our data
```

Out[2]:
```
<bound method NDFrame.describe of            START_DATE*         END_DATE* CATEGOR
Y*            START*  \
0       1/1/2016 21:11    1/1/2016 21:17  Business       Fort Pierce
1        1/2/2016 1:25     1/2/2016 1:37  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
...                ...               ...       ...               ...
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*
0          Fort Pierce      5.1   Meal/Entertain
1          Fort Pierce      5.0              NaN
2          Fort Pierce      4.8  Errand/Supplies
3          Fort Pierce      4.7          Meeting
4      West Palm Beach     63.7   Customer Visit
...                ...      ...              ...
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

[1156 rows x 7 columns]>
```

In [3]:
```
ud.info()

#info()- similar to stypes and descibe
```

```
<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
```

Check for missing values

In [19]:
```
print(ud.isnull().sum())

print(ud)

#isnull- to check for null values
```

```
START_DATE*    0
END_DATE*      0
CATEGORY*      0
START*         0
STOP*          0
MILES*         0
PURPOSE*       0
dtype: int64
             START_DATE*         END_DATE* CATEGORY*            START*  \
0    2016-01-01 21:11:00    1/1/2016 21:17  Business       Fort Pierce   
2    2016-01-02 20:25:00    1/2/2016 20:38  Business       Fort Pierce   
3    2016-01-05 17:31:00    1/5/2016 17:45  Business       Fort Pierce   
4    2016-01-06 14:42:00    1/6/2016 15:49  Business       Fort Pierce   
5    2016-01-06 17:15:00    1/6/2016 17:19  Business   West Palm Beach   
...                  ...               ...       ...               ...   
1150 2016-12-31 01:07:00   12/31/2016 1:14  Business           Kar?chi   
1151 2016-12-31 13:24:00  12/31/2016 13:42  Business           Kar?chi   
1152 2016-12-31 15:03:00  12/31/2016 15:38  Business  Unknown Location   
1153 2016-12-31 21:32:00  12/31/2016 21:50  Business       Katunayake   
1154 2016-12-31 22:08:00  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  

[652 rows x 7 columns]
```

Drop rows with missing values

In [18]:
```
ud = ud.dropna()
```

```
print(ud)

#dropna()-to drop rows
```

```
             START_DATE*         END_DATE* CATEGORY*            START*  \
0    2016-01-01 21:11:00    1/1/2016 21:17  Business       Fort Pierce
2    2016-01-02 20:25:00    1/2/2016 20:38  Business       Fort Pierce
3    2016-01-05 17:31:00    1/5/2016 17:45  Business       Fort Pierce
4    2016-01-06 14:42:00    1/6/2016 15:49  Business       Fort Pierce
5    2016-01-06 17:15:00    1/6/2016 17:19  Business   West Palm Beach
...                  ...               ...       ...               ...
1150 2016-12-31 01:07:00   12/31/2016 1:14  Business           Kar?chi
1151 2016-12-31 13:24:00  12/31/2016 13:42  Business           Kar?chi
1152 2016-12-31 15:03:00  12/31/2016 15:38  Business  Unknown Location
1153 2016-12-31 21:32:00  12/31/2016 21:50  Business       Katunayake
1154 2016-12-31 22:08:00  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

[652 rows x 7 columns]
```

fill missing values (propose column with unknown value)

In [12]:
```
ud['PURPOSE*'] = ud['PURPOSE*'].fillna('Unknown')
print(ud)

#fillna()-to fill resppective cells with a value
```

```
             START_DATE*         END_DATE* CATEGORY*            START*  \
0    2016-01-01 21:11:00    1/1/2016 21:17  Business       Fort Pierce
2    2016-01-02 20:25:00    1/2/2016 20:38  Business       Fort Pierce
3    2016-01-05 17:31:00    1/5/2016 17:45  Business       Fort Pierce
4    2016-01-06 14:42:00    1/6/2016 15:49  Business       Fort Pierce
5    2016-01-06 17:15:00    1/6/2016 17:19  Business   West Palm Beach
...                  ...               ...       ...               ...
1150 2016-12-31 01:07:00   12/31/2016 1:14  Business           Kar?chi
1151 2016-12-31 13:24:00  12/31/2016 13:42  Business           Kar?chi
1152 2016-12-31 15:03:00  12/31/2016 15:38  Business  Unknown Location
1153 2016-12-31 21:32:00  12/31/2016 21:50  Business       Katunayake
1154 2016-12-31 22:08:00  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

[652 rows x 7 columns]
```

Check and remove duplicates

In [22]:
```
duplicates = ud[ud.duplicated()]
print(duplicates)

#duplicated()-to identify duplicates
```

```
Empty DataFrame
Columns: [START_DATE*, END_DATE*, CATEGORY*, START*, STOP*, MILES*, PURPOSE*]
Index: []
```

In [23]:
```
ud = ud.drop_duplicates()
print(ud)

#drop_duplicates()-to drop duplicates in our dataset
```

```
             START_DATE*           END_DATE* CATEGORY*            START*  \
0    2016-01-01 21:11:00 2016-01-01 21:17:00  Business       Fort Pierce
2    2016-01-02 20:25:00 2016-01-02 20:38:00  Business       Fort Pierce
3    2016-01-05 17:31:00 2016-01-05 17:45:00  Business       Fort Pierce
4    2016-01-06 14:42:00 2016-01-06 15:49:00  Business       Fort Pierce
5    2016-01-06 17:15:00 2016-01-06 17:19:00  Business   West Palm Beach
...                  ...                 ...       ...               ...
1150 2016-12-31 01:07:00 2016-12-31 01:14:00  Business           Kar?chi
1151 2016-12-31 13:24:00 2016-12-31 13:42:00  Business           Kar?chi
1152 2016-12-31 15:03:00 2016-12-31 15:38:00  Business  Unknown Location
1153 2016-12-31 21:32:00 2016-12-31 21:50:00  Business       Katunayake
1154 2016-12-31 22:08:00 2016-12-31 23:51:00  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

[652 rows x 7 columns]
```

Convert START_DATE and END_DATE to datetime

In [20]:
```
ud['START_DATE*'] = pd.to_datetime(ud['START_DATE*'])
```

In [21]:
```
ud['END_DATE*'] = pd.to_datetime(ud['END_DATE*'])

#to_date_time()-to change from object datype to datetime
```

Total number of rides per category:

In [24]:
```
total_rides_category = ud['CATEGORY*'].value_counts()
print(total_rides_category)

#value.counts-to count values in our column
```

```
CATEGORY*
Business    646
Personal      6
Name: count, dtype: int64
```

Total miles traveled for each purpose:

In [25]:
```
total_miles_purpose = ud.groupby('PURPOSE*')['MILES*'].sum()
print(total_miles_purpose)

#groupby- to group the resultset with a respective column
```

```
PURPOSE*
Airport/Travel       16.5
Between Offices     197.0
Charity ($)          15.1
Commute             180.2
Customer Visit     2089.5
Errand/Supplies     508.0
Meal/Entertain      911.7
Meeting            2841.4
Moving               18.2
Temporary Site      523.7
Name: MILES*, dtype: float64
```

Average distance for business vs. personal rides:

In [26]:
```
average_distance_category = ud.groupby('CATEGORY*')['MILES*'].mean()
print(average_distance_category)
```

```
CATEGORY*
Business    10.971827
Personal    35.583333
Name: MILES*, dtype: float64
```

Add a column for cost estimation (assuming $2 per mile):

In [27]:
```
ud['COST_ESTIMATION'] = ud['MILES*'] * 2
print(ud)

#new column creation
```

```
             START_DATE*           END_DATE* CATEGORY*            START*  \
0    2016-01-01 21:11:00 2016-01-01 21:17:00  Business       Fort Pierce
2    2016-01-02 20:25:00 2016-01-02 20:38:00  Business       Fort Pierce
3    2016-01-05 17:31:00 2016-01-05 17:45:00  Business       Fort Pierce
4    2016-01-06 14:42:00 2016-01-06 15:49:00  Business       Fort Pierce
5    2016-01-06 17:15:00 2016-01-06 17:19:00  Business   West Palm Beach
...                  ...                 ...       ...               ...
1150 2016-12-31 01:07:00 2016-12-31 01:14:00  Business           Kar?chi
1151 2016-12-31 13:24:00 2016-12-31 13:42:00  Business           Kar?chi
1152 2016-12-31 15:03:00 2016-12-31 15:38:00  Business  Unknown Location
1153 2016-12-31 21:32:00 2016-12-31 21:50:00  Business        Katunayake
1154 2016-12-31 22:08:00 2016-12-31 23:51:00  Business           Gampaha

                STOP*  MILES*         PURPOSE*  COST_ESTIMATION
0        Fort Pierce     5.1   Meal/Entertain             10.2
2        Fort Pierce     4.8  Errand/Supplies              9.6
3        Fort Pierce     4.7          Meeting              9.4
4      West Palm Beach    63.7   Customer Visit           127.4
5      West Palm Beach     4.3   Meal/Entertain              8.6
...                ...     ...              ...              ...
1150           Kar?chi     0.7          Meeting              1.4
1151  Unknown Location     3.9   Temporary Site              7.8
1152  Unknown Location    16.2          Meeting             32.4
1153           Gampaha     6.4   Temporary Site             12.8
1154         Ilukwatta    48.2   Temporary Site             96.4

[652 rows x 8 columns]
```

Filter rides longer than 50 miles:

In [28]:
```
long_rides = ud[ud['MILES*'] > 50]
print(long_rides)
```

```
#condition applied miles >50
```

```
            START_DATE*           END_DATE* CATEGORY*         START*  \
4    2016-01-06 14:42:00 2016-01-06 15:49:00  Business    Fort Pierce
232  2016-03-17 12:52:00 2016-03-17 15:11:00  Business         Austin
251  2016-03-19 19:33:00 2016-03-19 20:39:00  Business      Galveston
268  2016-03-25 13:24:00 2016-03-25 16:22:00  Business           Cary
269  2016-03-25 16:52:00 2016-03-25 22:22:00  Business          Latta
270  2016-03-25 22:54:00 2016-03-26 01:39:00  Business   Jacksonville
295  2016-04-02 12:21:00 2016-04-02 14:47:00  Business      Kissimmee
296  2016-04-02 16:57:00 2016-04-02 18:09:00  Business  Daytona Beach
297  2016-04-02 19:38:00 2016-04-02 22:36:00  Business   Jacksonville
298  2016-04-02 23:11:00 2016-04-03 01:34:00  Business      Ridgeland
299  2016-04-03 02:00:00 2016-04-03 04:16:00  Business       Florence
559  2016-07-17 12:20:00 2016-07-17 15:25:00  Personal          Boone
869  2016-10-28 15:53:00 2016-10-28 17:59:00  Business           Cary
870  2016-10-28 18:13:00 2016-10-28 20:07:00  Business  Winston Salem
871  2016-10-28 20:13:00 2016-10-28 22:00:00  Business      Asheville
1088 2016-12-21 20:56:00 2016-12-21 23:42:00  Business     Rawalpindi

                STOP*  MILES*        PURPOSE*  COST_ESTIMATION
4      West Palm Beach    63.7  Customer Visit            127.4
232              Katy   136.0  Customer Visit            272.0
251           Houston    57.0  Customer Visit            114.0
268             Latta   144.0  Customer Visit            288.0
269      Jacksonville   310.3  Customer Visit            620.6
270         Kissimmee   201.0         Meeting            402.0
295      Daytona Beach    77.3  Customer Visit            154.6
296       Jacksonville    80.5  Customer Visit            161.0
297          Ridgeland   174.2  Customer Visit            348.4
298           Florence   144.0         Meeting            288.0
299               Cary   159.3         Meeting            318.6
559               Cary   180.2         Commute            360.4
869      Winston Salem   107.0         Meeting            214.0
870          Asheville   133.6         Meeting            267.2
871             Topton    91.8         Meeting            183.6
1088  Unknown Location   103.0         Meeting            206.0
```

Filter by specific purpose (e.g., meetings):

In [29]:
```
meetings = ud[ud['PURPOSE*'] == 'Meeting']
print(meetings)

#filtering data where purpose ==meetings
```

```
             START_DATE*           END_DATE* CATEGORY*            START*  \
3    2016-01-05 17:31:00 2016-01-05 17:45:00  Business       Fort Pierce
6    2016-01-06 17:30:00 2016-01-06 17:35:00  Business   West Palm Beach
7    2016-01-07 13:27:00 2016-01-07 13:33:00  Business              Cary
8    2016-01-10 08:05:00 2016-01-10 08:25:00  Business              Cary
10   2016-01-10 15:08:00 2016-01-10 15:51:00  Business          New York
...                  ...                 ...       ...             ...
1142 2016-12-29 20:15:00 2016-12-29 20:45:00  Business           Kar?chi
1144 2016-12-29 23:14:00 2016-12-29 23:47:00  Business  Unknown Location
1148 2016-12-30 16:45:00 2016-12-30 17:08:00  Business           Kar?chi
1150 2016-12-31 01:07:00 2016-12-31 01:14:00  Business           Kar?chi
1152 2016-12-31 15:03:00 2016-12-31 15:38:00  Business  Unknown Location

                 STOP*  MILES* PURPOSE*  COST_ESTIMATION
3          Fort Pierce     4.7  Meeting              9.4
6           Palm Beach     7.1  Meeting             14.2
7                 Cary     0.8  Meeting              1.6
8          Morrisville     8.3  Meeting             16.6
10              Queens    10.8  Meeting             21.6
...                ...     ...      ...              ...
1142           Kar?chi     7.2  Meeting             14.4
1144           Kar?chi    12.9  Meeting             25.8
1148           Kar?chi     4.6  Meeting              9.2
1150           Kar?chi     0.7  Meeting              1.4
1152  Unknown Location    16.2  Meeting             32.4

[186 rows x 8 columns]
```

What is the total number of business trips versus personal trips?

In [30]:
```
business_trips = ud[ud['CATEGORY*'] == 'Business'].shape[0]
personal_trips = ud[ud['CATEGORY*'] == 'Personal'].shape[0]
print(f"Business trips: {business_trips}, Personal trips: {personal_trips}")

#count of busines trips and personal trips
```

```
Business trips: 646, Personal trips: 6
```

What percentage of trips are business versus personal?

In [31]:
```
total_trips = ud.shape[0]
business_per = (business_trips / total_trips) * 100
print(business_per)
```

```
99.079754601227
```

In [33]:
```
personal_per = (personal_trips / total_trips) * 100
print(personal_per)

#percentage of business trips and personal trips
```

```
0.9202453987730062
```

In [ ]:
```

```