greatlearning
Power Ahead

PDS PROJECT SOLUTION

Please note that there are several techniques to answer a few questions of PDS Project. You will be rewarded marks for the question if your answer is matching with the output given in this solution file

Load the necessary libraries. Import and load the dataset with a name uber_drives .

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Get the Data
uber_drives=pd.read_csv("uberdrive.csv")
```

We have read the data and stored the data in "uber_drives" variable

Q1. Show the last 10 records of the dataset. (2 point)

In [3]:
```
uber_drives.tail(10)
```

Out[3]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1145	12/30/2016 10:15	12/30/2016 10:33	Business	Karachi	Karachi	2.8	Errand/Supplies
1146	12/30/2016 11:31	12/30/2016 11:56	Business	Karachi	Karachi	2.9	Errand/Supplies
1147	12/30/2016 15:41	12/30/2016 16:03	Business	Karachi	Karachi	4.6	Errand/Supplies
1148	12/30/2016 16:45	12/30/2016 17:08	Business	Karachi	Karachi	4.6	Meeting
1149	12/30/2016 23:06	12/30/2016 23:10	Business	Karachi	Karachi	0.8	Customer Visit
1150	12/31/2016 1:07	12/31/2016 1:14	Business	Karachi	Karachi	0.7	Meeting
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Karachi	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

Q2. Show the first 10 records of the dataset. (2 points)

In [4]:
```
uber_drives.head(10)
```

Out[4]:

	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
5	01-06-2016 17:15	01-06-2016 17:19	Business	West Palm Beach	West Palm Beach	4.3	Meal/Entertain
6	01-06-2016 17:30	01-06-2016 17:35	Business	West Palm Beach	Palm Beach	7.1	Meeting
7	01-07-2016 13:27	01-07-2016 13:33	Business	Cary	Cary	0.8	Meeting
8	01-10-2016 08:05	01-10-2016 08:25	Business	Cary	Morrisville	8.3	Meeting
9	01-10-2016 12:17	01-10-2016 12:44	Business	Jamaica	New York	16.5	Customer Visit

Q3. Show the dimension(number of rows and columns) of the dataset. (2 points)

In [5]:
```
print(uber_drives.shape)
print("The number of rows in the dataset are",uber_drives.shape[0])
print("The number of columns in the dataset are",uber_drives.shape[1])
```

```
(1155, 7)
The number of rows in the dataset are 1155
The number of columns in the dataset are 7
```

Q4. Show the size (Total number of elements) of the dataset. (2 points)

In [6]:
```
print(uber_drives.size)
```

```
8085
```

The total elements in the dataset are 8085 which is a product of number of rows and number of columns i.e. 1155*7 = 8085

Q5. Display the information about all the variables of the data set. What can you infer from the output?(1 +2 points)

Hint: Information includes - Total number of columns,variable data-types, number of non-null values in a variable, and usage

In [7]:
```
uber_drives.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1155 entries, 0 to 1154
Data columns (total 7 columns):
START_DATE*    1155 non-null object
END_DATE*      1155 non-null object
CATEGORY*      1155 non-null object
START*         1155 non-null object
STOP*          1155 non-null object
MILES*         1155 non-null float64
PURPOSE*       653 non-null object
dtypes: float64(1), object(6)
memory usage: 63.2+ KB
```

The data contains 6 object type variable and 1 float64 type variable.

We can observe that there are few Non-Null values in the Purpose column as Purpose" has lesser Non-Null values as compared to other variables

Q6. Check for missing values. (2 points)

Note: Output should contain only one boolean value

In [8]:
```
uber_drives.isna().values.any()
```

Out[8]:
```
True
```

isna.any() function will check if there is any missing value in the dataset. "True" indicates there is atleast one missing value in the dataset and "False" indicates there is no missing value in the dataset.

Here the code gives an output "True" which indicates there is atleast 1 missing value present in the datsaet

Q7. How many missing values are present in the entire dataset? (2 points)

In [9]:
```
uber_drives.isna().values  # This code will check for missing values for each element of a dataset and gives boolean output
```

Out[9]:
```
array([[False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False,  True],
       [False, False, False, ..., False, False, False],
       ...,
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False]])
```

In [10]:
```
uber_drives.isna().values.sum()   # this code will give the sum of all True values of the above code
```

Out[10]:
```
502
```

There are 502 missing values in the dataset

Q8. Get the summary of the original data. (2 points).

Hint: Summary includes- Count,Mean, Std, Min, 25%,50%,75% and max

In [11]:
```
uber_drives.describe()
```

Out[11]:

	MILES*
count	1155.000000
mean	10.566840
std	21.579106
min	0.500000
25%	2.900000
50%	6.000000
75%	10.400000
max	310.300000

The output gives summary of one variable Miles" as all other variables were of Object datatype.

Q9. Drop the missing values and store the data in a new dataframe (name it"df") (2-points)

Note: Dataframe "df" will not contain any missing value

In [12]:
```
df=uber_drives.dropna()
df.isnull().values.any()
```

Out[12]:
```
False
```

The new dataframe df do not contain any missing values

Q10. Check the information of the dataframe(df). (1 points)

Hint: Information includes - Total number of columns,variable data-types, number of non-null values in a variable, and usage

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 653 entries, 0 to 1154
Data columns (total 7 columns):
START_DATE*     653 non-null object
END_DATE*       653 non-null object
CATEGORY*       653 non-null object
START*          653 non-null object
STOP*           653 non-null object
MILES*          653 non-null float64
PURPOSE*        653 non-null object
dtypes: float64(1), object(6)
memory usage: 40.8+ KB
```

The "df" dataset does not contain any missing values as all the variables has 653 non-null values.

Q11. Get the unique start locations. (2 points)

Note: This question is based on the dataframe with no 'NA' values

Hint- You need to print the unique start locations place names in this and not the count.

In [14]:
```
df["START*"].unique()
```

```
Out[14]: array(['Fort Pierce', 'West Palm Beach', 'Cary', 'Jamaica', 'New York',
       'Elmhurst', 'Midtown', 'East Harlem', 'Flatiron District',
       'Midtown East', 'Hudson Square', 'Lower Manhattan',
       "Hell's Kitchen", 'Downtown', 'Gulfton', 'Houston', 'Eagan Park',
       'Morrisville', 'Durham', 'Farmington Woods', 'Lake Wellingborough',
       'Fayetteville Street', 'Raleigh', 'Whitebridge', 'Hazelwood',
       'Fairmont', 'Meredith Townes', 'Apex', 'Chapel Hill', 'Northwoods',
       'Edgehill Farms', 'Eastgate', 'East Elmhurst', 'Long Island City',
       'Katunayaka', 'Colombo', 'Nugegoda', 'Unknown Location',
       'Islamabad', 'R?walpindi', 'Noorpur Shahan', 'Preston',
       'Heritage Pines', 'Tanglewood', 'Waverly Place', 'Wayne Ridge',
       'Westpark Place', 'East Austin', 'The Drag', 'South Congress',
       'Georgian Acres', 'North Austin', 'West University', 'Austin',
       'Katy', 'Sharpstown', 'Sugar Land', 'Galveston', 'Port Bolivar',
       'Washington Avenue', 'Briar Meadow', 'Latta', 'Jacksonville',
       'Lake Reams', 'Orlando', 'Kissimmee', 'Daytona Beach', 'Ridgeland',
       'Florence', 'Meredith', 'Holly Springs', 'Chessington', 'Burtrose',
       'Parkway', 'Mcvan', 'Capitol One', 'University District',
       'Seattle', 'Redmond', 'Bellevue', 'San Francisco', 'Palo Alto',
       'Sunnyvale', 'Newark', 'Menlo Park', 'Old City', 'Savon Height',
       'Kilarney Woods', 'Townes at Everett Crossing', 'Huntington Woods',
       'Weston', 'Seaport', 'Medical Centre', 'Rose Hill', 'Soho',
       'Tribeca', 'Financial District', 'Oakland', 'Emeryville',
       'Berkeley', 'Kenner', 'CBD', 'Lower Garden District', 'Storyville',
       'New Orleans', 'Chalmette', 'Arabi', 'Pontchartrain Shores',
       'Metairie', 'Summerwinds', 'Parkwood', 'Banner Elk', 'Boone',
       'Stonewater', 'Lexington Park at Amberly', 'Winston Salem',
       'Asheville', 'Topton', 'Renaissance', 'Santa Clara', 'Ingleside',
       'West Berkeley', 'Mountain View', 'El Cerrito', 'Krendle Woods',
       'Fuquay-Varina', 'Rawalpindi', 'Lahore', 'Karachi', 'Katunayake',
       'Gampaha'], dtype=object)
```

Q12. What is the total number of unique start locations? (2 points)

Note: Use the original dataframe without dropping 'NA' values

```
In [15]: uber_drives["START*"].nunique()  # nunique() function will give the count of observations
```

```
Out[15]: 176
```

There are a total of 176 unique start locations

Q13. What is the total number of unique stop locations. (2 points)

Note: Use the original dataframe without dropping 'NA' values.

```
In [16]: uber_drives["STOP*"].nunique()
```

```
Out[16]: 187
```

There are a total of 187 unique stop locations

Q14. Display all the Uber trips that has the starting point of San Francisco. (2 points)

Note: Use the original dataframe without dropping the 'NA' values.

Hint: You need to display the rows which has starting point of San Francisco.

In [17]:
```
uber_drives.loc[uber_drives["START*"]=="San Francisco"]
```

Out[17]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
362	05-09-2016 14:39	05-09-2016 15:06	Business	San Francisco	Palo Alto	20.5	Between Offices
440	6/14/2016 16:09	6/14/2016 16:39	Business	San Francisco	Emeryville	11.6	Meeting
836	10/19/2016 14:02	10/19/2016 14:31	Business	San Francisco	Berkeley	10.8	NaN
917	11-07-2016 19:17	11-07-2016 19:57	Business	San Francisco	Berkeley	13.2	Between Offices
919	11-08-2016 12:16	11-08-2016 12:49	Business	San Francisco	Berkeley	11.3	Meeting
927	11-09-2016 18:40	11-09-2016 19:17	Business	San Francisco	Oakland	12.7	Customer Visit
933	11-10-2016 15:17	11-10-2016 15:22	Business	San Francisco	Oakland	9.9	Temporary Site
966	11/15/2016 20:44	11/15/2016 21:00	Business	San Francisco	Berkeley	11.8	Temporary Site

Q15. What is the most popular starting point for the Uber drivers? (2 points)

Note: Use the original dataframe without dropping the 'NA' values.

Hint:Popular means the place that is visited the most

In [18]:
```
uber_drives["START*"].value_counts()
```

Out[18]:
```
Cary                   201
Unknown Location       148
Morrisville             85
Whitebridge             68
Islamabad               57
Durham                  37
Lahore                  36
Karachi                 31
Raleigh                 28
Apex                    17
Westpark Place          17
Berkeley                16
Midtown                 14
Kenner                  11
R?walpindi              11
Kissimmee               11
New Orleans             10
Emeryville              10
Downtown                 9
Edgehill Farms           8
Central                  8
Orlando                  8
```

Cary is the most popular starting point

Q16. What is the most popular dropping point for the Uber drivers? (2 points)

Note: Use the original dataframe without dropping the 'NA' values.

Hint: Popular means the place that is visited the most

In [19]:
```
uber_drives["STOP*"].value_counts()
```

Out[19]:
```
Cary                    203
Unknown Location        149
Morrisville              84
Whitebridge              65
Islamabad                58
Durham                   36
Lahore                   36
Raleigh                  29
Karachi                  28
Apex                     17
Westpark Place           16
Berkeley                 16
R?walpindi               13
Kissimmee                12
Midtown                  11
Kenner                   10
New Orleans              10
Edgehill Farms           10
Central                   9
```

Cary is the most popular dropping point

Q17. What is the most frequent route taken by Uber drivers. (3 points)

Note: This question is based on the new dataframe with no 'na' values.

Hint-Print the most frequent route taken by Uber drivers (Route= combination of START & END points present in the Data set).

In [20]:
```
df.groupby(["START*","STOP*"]).size().sort_values(ascending=False).head(10)
```

Out[20]:
```
START*            STOP*
Cary              Morrisville         52
Morrisville       Cary                51
Cary              Cary                44
Unknown Location  Unknown Location    30
Cary              Durham              30
Durham            Cary                29
Karachi           Karachi             20
Cary              Raleigh             17
Lahore            Lahore              16
Raleigh           Cary                15
dtype: int64
```

In [21]:
```
df.groupby(["START*","STOP*"]).size().sort_values(ascending=False).head(1)  # this will give us the first observation only
```

Out[21]:
```
START*  STOP*
Cary    Morrisville    52
dtype: int64
```

The most frequent/ popular route taken by Uber Drivers is from Cary to Morrisville

Q18. Display all types of purposes for the trip in an array. (2 points)

Note: This question is based on the new dataframe with no 'NA' values.

In [22]:
```
print(np.array(df['PURPOSE*'].unique()))
```

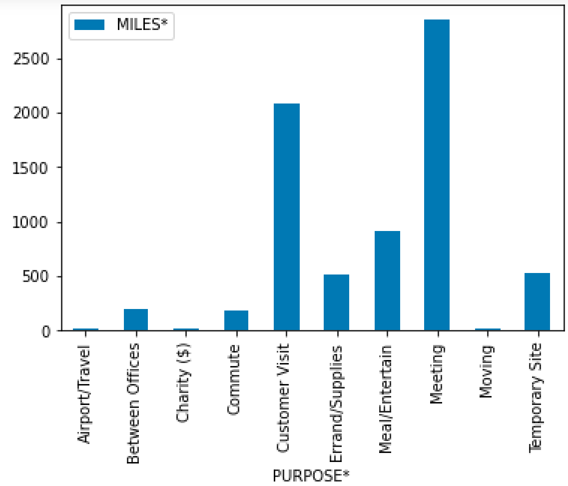
```
['Meal/Entertain' 'Errand/Supplies' 'Meeting' 'Customer Visit'
 'Temporary Site' 'Between Offices' 'Charity ($)' 'Commute' 'Moving'
 'Airport/Travel']
```

Q19. Plot a bar graph of Purpose vs Miles(Distance). What can you infer from the plot(2 +2 points)

Note: Use the original dataframe without dropping the 'NA' values.

Hint:You have to plot total/sum miles per purpose

In [23]:
```
df1=pd.DataFrame(uber_drives["MILES*"]).groupby(uber_drives["PURPOSE*"]).sum()
df1.plot(kind= "bar")
plt.show()
```



Maximum miles were clocked for Meeting Purpose followed by Customer Visit Purpose. Airport/Travel, Charity and Moving are the purposes where least miles were clocked

Q20. Display a dataframe of Purpose and the total distance travelled for that particular Purpose. (3 points)

Note: Use the original dataframe without dropping "NA" values

In [24]:
```
uber_drives.groupby("PURPOSE*").sum()
```
Out[24]:

PURPOSE*	MILES*
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	2851.3
Moving	18.2
Temporary Site	523.7

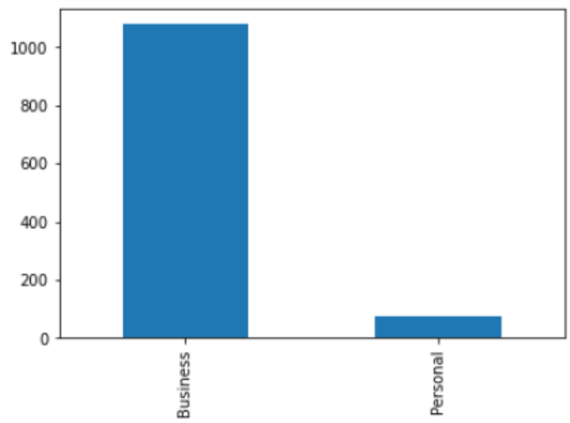
The maximum Miles were clocked for Meeting Purpose and the minimum Miles were clocked for Charity($) Purpose.

Q21. Generate a plot showing count of trips vs category of trips. What can you infer from the plot (2 +1 points)

Note: Use the original dataframe without dropping the 'NA' values.

In [25]:
```
uber_drives["CATEGORY*"].value_counts()
```
Out[25]:
```
Business    1078
Personal      77
Name: CATEGORY*, dtype: int64
```

In [26]:
```
uber_drives["CATEGORY*"].value_counts().plot(kind="bar")
```
Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1fd4fb592e8>



The majority of Uber trips were of Business Category and a few were of Personal Category

Q22. What percentage of Miles were clocked under Business Category and what percentage of Miles were clocked under Personal Category ? (3 points)

Note:Use the original dataframe without dropping the 'NA' values.

In [27]:
```
uber_drives.groupby("CATEGORY*").sum()
```

Out[27]:

CATEGORY*	MILES*
Business	11487.0
Personal	717.7

In [28]:
```
uber_drives.groupby("CATEGORY*").sum()/ uber_drives["MILES*"].sum() # to calculate proportion
```

Out[28]:

CATEGORY*	MILES*
Business	0.941195
Personal	0.058805

In [29]:
```
uber_drives.groupby("CATEGORY*").sum()/ uber_drives["MILES*"].sum() *100   # To calcuate percentage
```

Out[29]:

CATEGORY*	MILES*
Business	94.119479
Personal	5.880521

94.12% of the Miles were clocked for Business Category whereas 5.88% of the Mies were cloceked for Personal Category

THE END