# PROBLEM STATEMENT

**Analyze any datasheet using NumPy, Pandas and Matplotlib**

# METHODOLOGY /PROCEDURE

### We decided to analyze the datasheet of Uber Technologies, Inc. (Uber), based in San Francisco for the mini-project.

The datasheet has information related to the name of the driver and the customer, pick-up and drop times, start and end location, miles travelled, purpose of travel i.e for personal purpose of business purpose, method of payment and the amount paid.

# CODING (PYTHON)

import pandas as pd import numpy as np import seaborn as sns

from matplotlib import pyplot as plt

%matplotlib inline

data=pd.read\_csv('uberdrives.csv')

ľhe fiíst 10 íecoíds of the dataset.

data.head(10)

#### ľhe last 10 íecoíds of the dataset.

data.tail(10)

#### ľhe dimension(numbeí of íows and columns) of the dataset.

data.shape

print('The Number of Rows are {} and the number of Columns are {}'.form

at(data.shape[0],data.shape[1]))

ľotal numbeí of elements in the dataset.

print(data.size)

#### ľhe infoímation about all the vaíiables of the data set.

data.info()

#### Missing values aíe píesent in the entiíe dataset

data.isnull().values.sum()

#### ľhe summaíy of the oíiginal data.

data.describe().T

#### Díopping the missing values and stoíe the data in a new datafíame

df=data.dropna()

df.isnull().values.any()

Infoímation of the datafíame

df.info()

df.columns

#### ľhe total numbeí of unique staít locations

data['START\*'].nunique()

#### ľhe total numbeí of unique stop locations.

data['STOP\*'].nunique()

#### All Ubeí tíips that has the staíting point as San Fíancisco

data.loc[data['START\*']== "San Francisco"]

#### Most populaí staíting point foí the Ubeí díiveís

data['START\*'].value\_counts().head(1)

#### Most populaí díopping point foí the Ubeí díiveís

data['STOP\*'].value\_counts().head(1)

ľhe most fíequent íoute taken by Ubeí díiveís. df.groupby(["START\*","STOP\*"]).size().sort\_values(ascending=False).head (5)

#### All types of puíposes foí the tíip in an aííay.

print(np.array(df["PURPOSE\*"].unique()))

#### Baí gíaph of Puípose vs Miles(Distance)

df2=pd.DataFrame(data["MILES\*"]).groupby(data["PURPOSE\*"]).sum() df2.plot(kind="bar")

plt.show()

print('From the above chart it is infered that the most rides are taken for the purpose of MEETING and the least amount of rides are made for

MOVIG, AIRPORT TRAVEL & CHARITY')

from google.colab import drive

drive.mount('/content/drive')

Datafíame of Puípose and the total distance tíavelled foí that paíticulaí Puípose.

data.groupby("PURPOSE\*").sum()

#### A plot showing count of tíips vs categoíy of tíips.

data["CATEGORY\*"].value\_counts().plot(kind="bar") plt.show()

print("From the above chart it is inferred that the most trips are take n for Bussiness related and least trips are made for personal work ")

weíe clocked undeí Peísonal Categoíy

data.groupby("CATEGORY\*").sum()

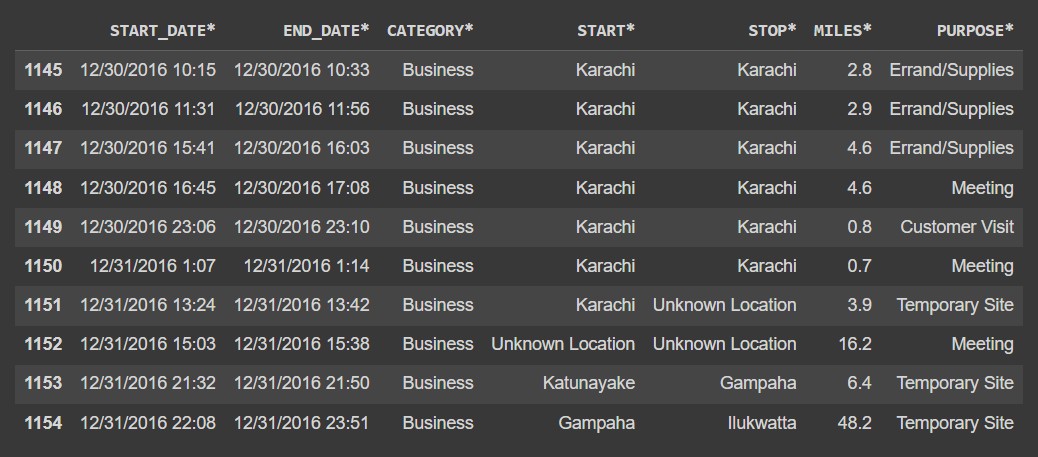
avg = data.groupby("CATEGORY\*").sum()/data["MILES\*"].sum()

print(avg\*100)

***THE END***

# RESULTS

#### the first 10 seconds of the dataset.

1. the last 10 seconds of the dataset.

#### the dimension(number of rows and columns) of the dataset.

The Number of Rows are 1155 and the number of Columns are 7

#### total number of elements in the dataset.

8085

#### the information about all the variables of the data set.

<class 'pandas.core.frame.DataFrame'> RangeIndex: 1155 entries, 0 to 1154 Data columns (total 7 columns):

1. START\_DATE\* 1155 non-null object
2. END\_DATE\* 1155 non-null object
3. CATEGORY\* 1155 non-null object
4. START\* 1155 non-null object
5. STOP\* 1155 non-null object

# Column Non-Null Count Dtype

5 MILES\*

1155 non-null float64

6 PURPOSE\* 653 non-null object dtypes: float64(1), object(6)

#### Missing values are present in the entire dataset

502

#### the summary of the original data.

1. Dropping the missing values and store the data in a new dataframe

False

#### Information of the dataframe

<class 'pandas.core.frame.DataFrame'> Int64Index: 653 entries, 0 to 1154 Data columns (total 7 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 |  | START\_DATE\* | 653 non-null |  | object |
| 1 |  | END\_DATE\* | 653 non-null |  | object |
| 2 |  | CATEGORY\* | 653 non-null |  | object |
| 3 |  | START\* | 653 non-null |  | object |
| 4 |  | STOP\* | 653 non-null |  | object |

5 MILES\* 653 non-null float64

6 PURPOSE\* 653 non-null object dtypes: float64(1), object(6)

#### the unique start locations.

Index(['START\_DATE\*', 'END\_DATE\*', 'CATEGORY\*', 'START\*', 'STOP\*',

'MILES\*', 'PURPOSE\*'], dtype='object')

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', 'Rawalpindi', '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)

#### the total number of unique start locations

176

#### the total number of unique stop locations.

187

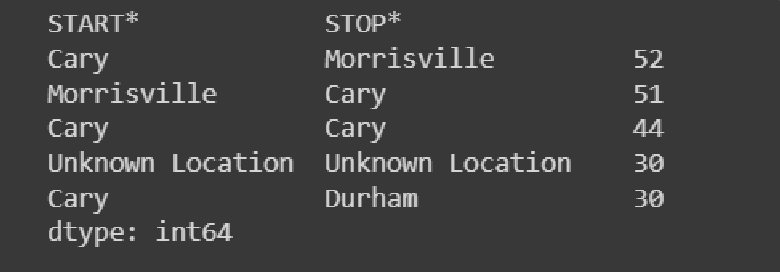
#### All Uber trips that has the starting point as San Francisco

1. Most popular starting point for the Uber drivers

Cary 201 Name: START\*, dtype: int64

#### Most popular dropping point for the Uber drivers

Cary 203 Name: STOP\*, dtype: int64



#### All types of purposes for the trip in an array.

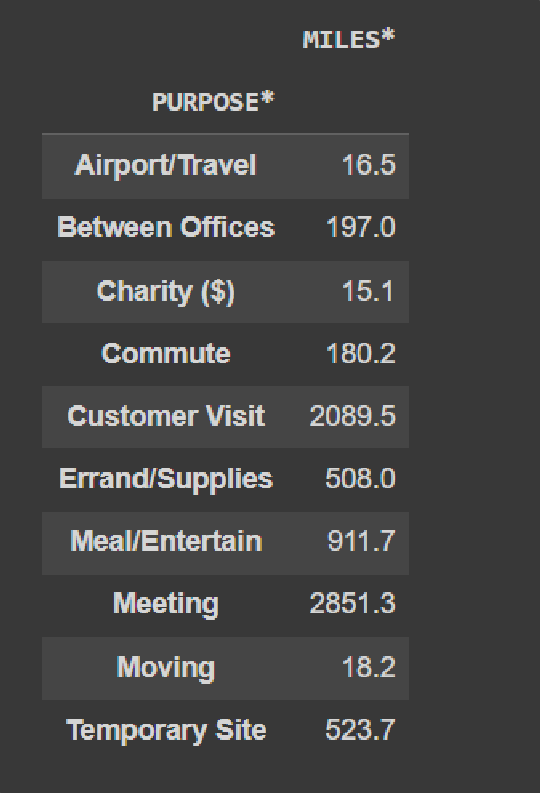
['Meal/Entertain' 'Errand/Supplies' 'Meeting' 'Customer Visit'

'Temporary Site' 'Between Offices' 'Charity ($)' 'Commute' 'Moving'

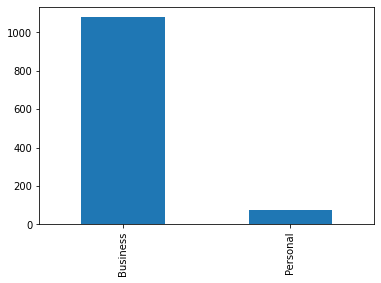
'Airport/Travel']

#### Bar graph of Purpose vs Miles(Distance)

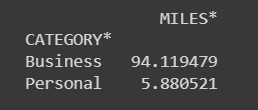
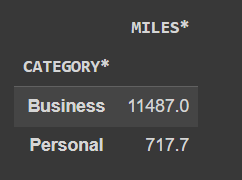
|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| From the above chart it is inferred that the most rides are taken for | | | |
| the purpose of MEETING and the least amount of rides are made for | | |  |
| MOVIG, AIRPORT TRAVEL & CHARITY |  | |



20.A plot showing count of trips vs category of trips.



From the above chart it is inferred that the most trips are taken for Business related and least trips are made for personal work



#### the information about all the variables of the data set.

<class 'pandas.core.frame.DataFrame'> RangeIndex: 1155 entries, 0 to 1154 Data columns (total 7 columns):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # |  | Column | Non-Null Count |  | Dtype |
|  |  |  |  |  |  |
| 0 |  | START\_DATE\* | 1155 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\* 1155 non-null float64

6 PURPOSE\* 653 non-null object dtypes: float64(1), object(6)

#### Missing values are present in the entire dataset

502

#### the summary of the original data.

# CONCLUSION

# We were successfully able to analyze the data we collected from Uber company. We were able to find information to questions like – Which was the most frequent start point/ end point? Which was the most frequent route taken? What was the purpose of travel? What was the most common purpose of travel etc.

# We concluded that the most frequent start and end point was Cary. The most frequent route taken was Cary too. The most common purpose of travel was for business meetings. Apart from travelling for work, customers travelled for meals/ entertainment purposes.

# Many customers used Uber to commute to the airport and to shuttle between home and office.

# We were also able to plot graphs from the data we got from Uber using Python. Graphs aided us to understand the given data better and make better analysis.

Load the necessary libraries. Import and load the dataset with a name uber\_drives .

import pandas as pd import numpy as np

import seaborn as sns

from matplotlib import pyplot as plt

%matplotlib inline

data=pd.read\_csv('uberdrives.csv')

The first 10 records of the dataset.



data.head(10)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |

The last 10 records of the dataset.

data.tail(10)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 |

The dimension(number of rows and columns) of the dataset.

data.shape

print('The Number of Rows are {} and the number of Columns are {}'.format(data.shape[0],data.shape[1]))

The Number of Rows are 1155 and the number of Columns are 7

Total number of elements in the dataset.

print(data.size)

8085

The information about all the variables of the data set.

https://colab.research.google.com/drive/1RcNAWC3EeqFXjTIzmgUyxRALhFLKCMA6#printMode=tr 2/10

data.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 1155 entries, 0 to 1154

Data columns (total 7 columns):

# Column Non-Null Count Dtype

* 1. START\_DATE\* 1155 non-null object
  2. END\_DATE\* 1155 non-null object
  3. CATEGORY\* 1155 non-null object
  4. START\* 1155 non-null object
  5. STOP\* 1155 non-null object
  6. MILES\* 1155 non-null float64
  7. PURPOSE\* 653 non-null object dtypes: float64(1), object(6)

memory usage: 63.3+ KB

Missing values are present in the entire dataset

data.isnull().values.sum()

502

The summary of the original data.

data.describe().T

**count mean std min 25% 50% 75% max**

**MILES\*** 1155.0 10.56684 21.579106 0.5 2.9 6.0 10.4 310.3

Dropping the missing values and store the data in a new dataframe

df=data.dropna()

df.isnull().values.any()

False

Information of the dataframe

https://colab.research.google.com/drive/1RcNAWC3EeqFXjTIzmgUyxRALhFLKCMA6#printMode=true 3/10

df.info()

<class 'pandas.core.frame.DataFrame'> Int64Index: 653 entries, 0 to 1154

Data columns (total 7 columns):

# Column Non-Null Count Dtype

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 |  | START\_DATE\* | 653 | non-null |  | object |
| 1 |  | END\_DATE\* | 653 | non-null |  | object |
| 2 |  | CATEGORY\* | 653 | non-null |  | object |
| 3 |  | START\* | 653 | non-null |  | object |
| 4 |  | STOP\* | 653 | non-null |  | object |
| 5 |  | MILES\* | 653 | non-null |  | float64 |
| 6 |  | PURPOSE\* | 653 | non-null |  | object |

dtypes: float64(1), object(6) memory usage: 40.8+ KB

The unique start locations.

df.columns

Index(['START\_DATE\*', 'END\_DATE\*', 'CATEGORY\*', 'START\*', 'STOP\*', 'MILES\*', 'PURPOSE\*'],

dtype='object')

df['START\*'].unique()

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)

The total number of unique start locations

data['START\*'].nunique()

176

The total number of unique stop locations.

data['STOP\*'].nunique()

187

All Uber trips that has the starting point as San Francisco

data.loc[data['START\*']== "San Francisco"]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 |

Most popular starting point for the Uber drivers

**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

data['START\*'].value\_counts().head(1)

Cary 201

Name: START\*, dtype: int64

**966** 11/15/2016 20:44 11/15/2016 21:00 Business San Francisco Berkeley 11.8 Temporary Site

Most popular dropping point for the Uber drivers

Cary 203

data['STOP\*'].value\_counts().head(1)

Name: STOP\*, dtype: int64

The most frequent route taken by Uber drivers. (3 points)

df.groupby(["START\*","STOP\*"]).size().sort\_values(ascending=False).head(5)

|  |  |  |
| --- | --- | --- |
| START\*  Cary | STOP\*  Morrisville | 52 |
| Morrisville | Cary | 51 |
| Cary | Cary | 44 |
| Unknown Location | Unknown Location | 30 |
| Cary | Durham | 30 |

dtype: int64

All types of purposes for the trip in an array.

print(np.array(df["PURPOSE\*"].unique()))

['Meal/Entertain' 'Errand/Supplies' 'Meeting' 'Customer Visit'

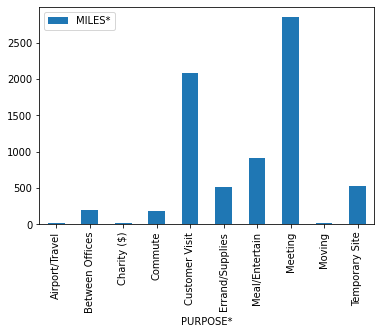
'Temporary Site' 'Between Offices' 'Charity ($)' 'Commute' 'Moving' 'Airport/Travel']

Bar graph of Purpose vs Miles(Distance)

df2=pd.DataFrame(data["MILES\*"]).groupby(data["PURPOSE\*"]).sum() df2.plot(kind="bar")

plt.show()

print('From the above chart it is infered that the most rides are taken for the purpose of MEETING and the least amount of rides are made for MOVIG, AIRPORT TRAVEL & CHARITY')



From the above chart it is infered that the most rides are taken for the purpose of MEETING and the least amount of ri

from google.colab import drive drive.mount('/content/drive')

Mounted at /content/drive

Dataframe of Purpose and the total distance travelled for that particular Purpose.

data.groupby("PURPOSE\*").sum()

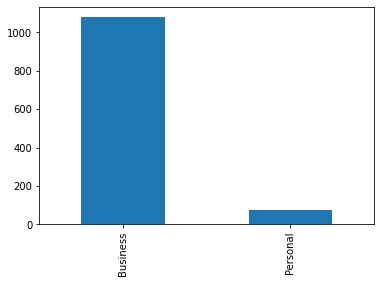
|  |  |
| --- | --- |
| **PURPOSE\*** | **MILES\*** |
| **Airport/Travel** | 16.5 |
| **Between Offices** | 197.0 |
| **Charity ($)** | 15.1 |
| **Commute** | 180.2 |
| **Customer Visit** | 2089.5 |
| **Errand/Supplies** | 508.0 |
| **Meal/Entertain** | 911.7 |
| **Meeting** | 2851.3 |
| **Moving** | 18.2 |
| **Temporary Site** | 523.7 |

A plot showing count of trips vs category of trips.

data["CATEGORY\*"].value\_counts().plot(kind="bar") plt.show()

print("From the above chart it is inferred that the most trips are taken for Bussiness related and least trips are made for personal work ")

8/10



Percentage of Miles were clocked under Business Category and what percentage of Miles were clocked under Personal Category

**MILES\***

**CATEGORY\***

**Business** 11487.0

**Personal**

717.7

data.groupby("CATEGORY\*").sum()

avg = data.groupby("CATEGORY\*").sum()/data["MILES\*"].sum() print(avg\*100)

MILES\*

CATEGORY\*

From the above chart it is inferred that the most trips are taken for Bussiness related and least trips are made for p

Business 94.119479

Personal 5.880521

***THE END***