UBER

Uber Ride Exploratory Analysis

Presented by: Prateek Kumar Agarwal

Agenda

Tip: Use links to go to a different page inside presentation.

UBER

Company Overview

Uber is a transportation company with an app that allows passengers to hail a ride and drivers to charge fares and get paid. More specifically, Uber is a ridesharing company that hires independent contractors as drivers.

Back to Agenda

Objectives

- Understand how uber ride work based on avaiable data
- understand data and clean it if found any abnormality
- Analysis on data to find insights
- Give Recomdation based on Insights found

Back to Agenda

UBER

Resources

Google Collab

Using Cloub Python note Book for Analysis

CSV File

Data stored in aexcel CSV file

Google Drive

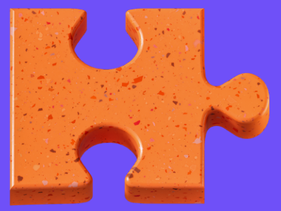
Files stored in Google drive Cloud Eniveronment

Back to Agenda

UBER

Approches

Data exploration



Data Cleaning

Data Analysing

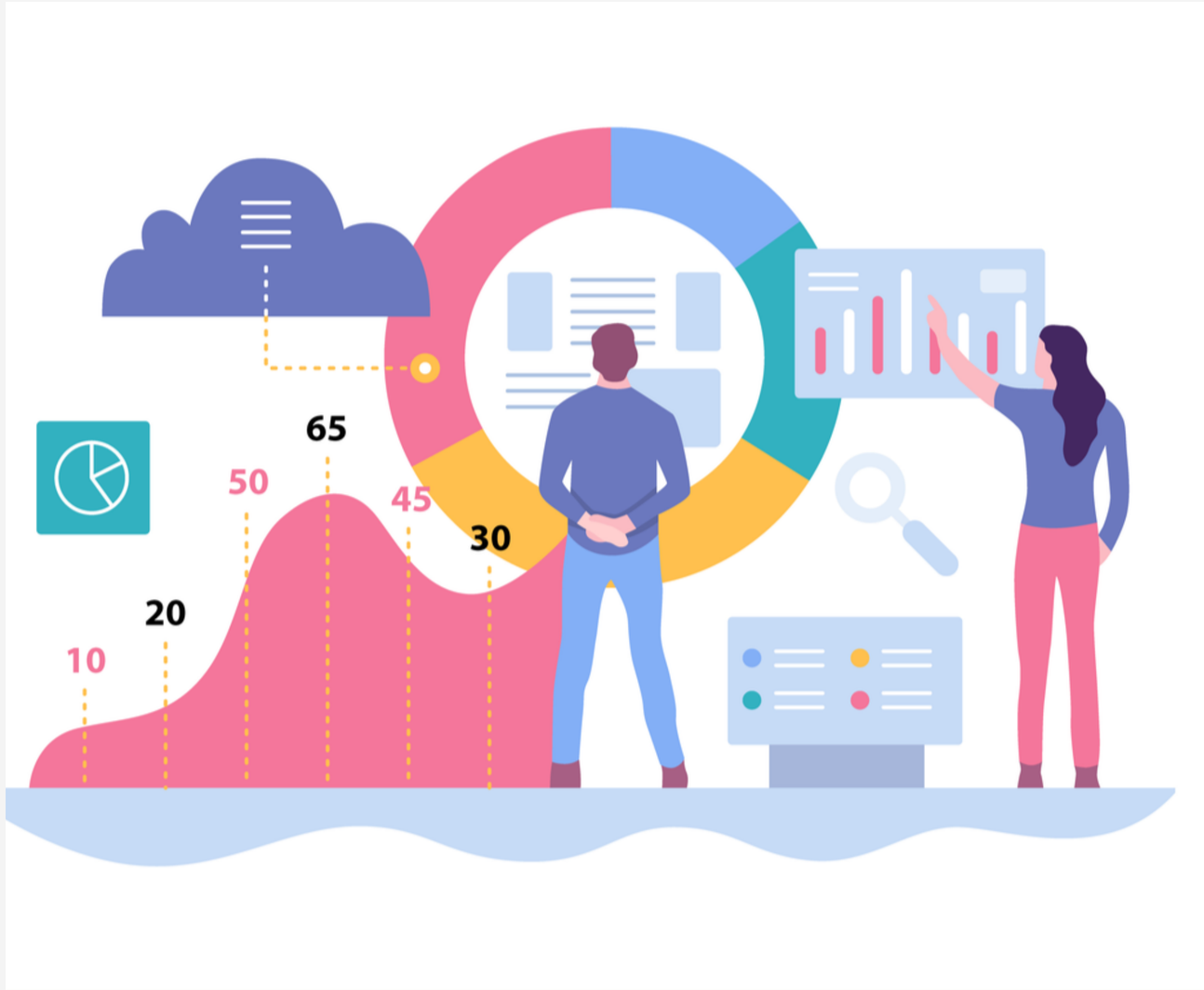
Back to Agenda

Back to Agenda

Approach

Data Exploartion

Importind data in platform and understanding what data is about , what each column describes .



Understanding the data in columns

START_DATE - Date of travel when ride start
END_DATE - Date of travel when ride End
CATEGORY - Type of ride (Business / Personal)
START - Location from where Ride started
STOP - Location where Ride stoped
MILES - Total distance covered in Miles
PURPOSE - For which type of purpose this ride is used for

```
# Viewing  data
data
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	08-06-2012	12-06-2012	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	24-07-2019	02-08-2019	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	01-05-2018	02-05-2018	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	25-08-2015	03-09-2015	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	03-12-2011	05-12-2011	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit
...	...	...	...	...	...	...	...
1150	12-08-2011	14-08-2011	Business	Kar?chi	Kar?chi	0.7	Meeting
1151	19-03-2011	29-03-2011	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	27-05-2010	27-05-2010	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	29-09-2014	04-10-2014	Business	Katunayake	Gampaha	6.4	Temporary Site
1154	04-01-2015	12-01-2015	Business	Gampaha	Ilukwatta	48.2	Temporary Site

1155 rows × 7 columns

Checking and understand total number of rows and columns in the dataset

```
data.info()
```

```
<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)
memory usage: 63.3+ KB
```

Total number of Non-NULL values across every column in the dataset.

```
data.notnull().sum()
```

```
START_DATE*     1155
END_DATE*       1155
CATEGORY*       1155
START*          1155
STOP*           1155
MILES*          1155
PURPOSE*         653
dtype: int64
```

Numerical data column - Miles

Exploring how data varry in Miles

```
data.describe()
```

	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

Maximum NULL values can be seen on 'Purpose' column

```
data[data['PURPOSE*'].isnull()]
```

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1	24-07-2019	02-08-2019	Business	Fort Pierce	Fort Pierce	5.0	NaN
32	25-03-2015	28-03-2015	Business	Whitebridge	Lake Wellingborough	7.2	NaN
85	28-09-2020	07-10-2020	Personal	Whitebridge	Northwoods	5.3	NaN
86	21-07-2018	26-07-2018	Personal	Northwoods	Tanglewood	3.0	NaN
87	23-01-2014	26-01-2014	Personal	Tanglewood	Preston	5.1	NaN
...	...	...	...	...	...	...	...
1065	07-02-2020	08-02-2020	Business	Unknown Location	Unknown Location	5.3	NaN
1066	23-08-2019	30-08-2019	Business	Unknown Location	Unknown Location	5.4	NaN
1069	18-05-2020	23-05-2020	Business	Islamabad	Unknown Location	2.2	NaN
1071	02-12-2018	09-12-2018	Business	Unknown Location	Rawalpindi	12.0	NaN
1143	13-07-2016	21-07-2016	Business	Kar?chi	Unknown Location	6.4	NaN

502 rows × 7 columns

Back to Agenda

Approach

Data Cleaning

After exploaring data , some cleaning operation are required on data

We can see header have some extra symbol in each column i.e. *
Removing them since it is unnecessary

START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*

AFTER

```
# Removing them from all columns
data.columns = data.columns.str.replace('*','')

data
```

```
<ipython-input-78-407876dc7bcc>:2: FutureWarning: The default value of regex will change from
  data.columns = data.columns.str.replace('*','')
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	08-06-2012	12-06-2012	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	24-07-2019	02-08-2019	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	01-05-2018	02-05-2018	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies

Datatypes of START_DATE and END_DATE columns to datetime.

```
data['START_DATE'] = pd.to_datetime(data['START_DATE'])

data['END_DATE'] = pd.to_datetime(data['END_DATE'])

data.dtypes
```

```
START_DATE    datetime64[ns]
END_DATE      datetime64[ns]
CATEGORY              object
START                 object
STOP                  object
MILES                float64
PURPOSE               object
dtype: object
```

In Start point and end point location their are many Unknown location mention which dosent make sense as location is needed to ride to take place .
So eliminating such rows

```
data = data[(data['START']!='Unknown Location') & (data['STOP']!='Unknown Location')]

data
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE	MONTH_No.	MONTH
0	2012-08-06	2012-12-06	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	8	August
1	2019-07-24	2019-02-08	Business	Fort Pierce	Fort Pierce	5.0	NaN	7	July
2	2018-01-05	2018-02-05	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	1	January
3	2015-08-25	2015-03-09	Business	Fort Pierce	Fort Pierce	4.7	Meeting	8	August
4	2011-03-12	2011-05-12	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit	3	March
...	...	...	...	...	...	...	...	...	...
1148	2013-03-25	2013-03-26	Business	Kar?chi	Kar?chi	4.6	Meeting	3	March
1149	2015-02-26	2015-05-03	Business	Kar?chi	Kar?chi	0.8	Customer Visit	2	February
1150	2011-12-08	2011-08-14	Business	Kar?chi	Kar?chi	0.7	Meeting	12	December
1153	2014-09-29	2014-04-10	Business	Katunayake	Gampaha	6.4	Temporary Site	9	September
1154	2015-04-01	2015-12-01	Business	Gampaha	Ilukwatta	48.2	Temporary Site	4	April

944 rows × 9 columns

Back to Agenda

Approach

Data Analysis

After cleaning data and getting more acurate data .

Analysis data to obtain insights which help to take decision to improve and progress

```python
# Total number of unique START points
print(len(data['START'].unique()))

# Unique START Places
print(data['START'].unique())
```

```
175
['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'
 'Whitebridge' 'Lake Wellingborough' 'Fayetteville Street' 'Raleigh'
 'Hazelwood' 'Fairmont' 'Meredith Townes' 'Apex' 'Chapel Hill'
 'Northwoods' 'Edgehill Farms' 'Tanglewood' 'Preston' 'Eastgate'
 'East Elmhurst' 'Jackson Heights' 'Long Island City' 'Colombo'
```

```python
# Total number of unique STOP points
print(len(data['STOP'].unique()))

# Unique STOP Places
print(data['STOP'].unique())
```

```
187
['Fort Pierce' 'West Palm Beach' 'Palm Beach' 'Cary' 'Morrisville'
 'New York' 'Queens' 'East Harlem' 'NoMad' 'Midtown' 'Midtown East'
 'Hudson Square' 'Lower Manhattan' "Hell's Kitchen" 'Queens County'
 'Gulfton' 'Downtown' 'Houston' 'Jamestown Court' 'Durham' 'Whitebridge'
 'Lake Wellingborough' 'Raleigh' 'Umstead' 'Hazelwood' 'Westpark Place'
 'Meredith Townes' 'Leesville Hollow' 'Apex' 'Chapel Hill'
 'Williamsburg Manor' 'Macgregor Downs' 'Edgehill Farms' 'Northwoods'
 'Tanglewood' 'Preston' 'Walnut Terrace' 'Jackson Heights' 'East Elmhurst'
```

Rides where we have the same START and STOP locations

```
data[data['START']==data['STOP']]
```

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES	PURPOSE
0	2012-08-06	2012-12-06	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	2019-07-24	2019-02-08	Business	Fort Pierce	Fort Pierce	5.0	NaN

```
# Total such rides
len(data[data['START']==(data['STOP'])])
```

```
202
```

Top Most visted starting point according the the total number of MILES covered.

```
data.groupby(['START'])['MILES'].sum().sort_values(ascending = False).head(5)
```

```
START
Cary           1784.6
Morrisville     671.7
Raleigh         433.0
Durham          384.4
Jacksonville    375.2
Name: MILES, dtype: float64
```

Top Most visted Stop point according the the total number of MILES covered.

```
data.groupby(['STOP'])['MILES'].sum().sort_values(ascending = False).head(5)
```

```
STOP
Cary             1913.1
Morrisville       524.3
Raleigh           393.7
Jacksonville      390.8
Durham            390.0
Name: MILES, dtype: float64
```

Between which place for the ride where maximum miles are covered.

```
data[['START','STOP','MILES']].sort_values(by='MILES',ascending = False).head(5)
```

	START	STOP	MILES
269	Latta	Jacksonville	310.3
270	Jacksonville	Kissimmee	201.0
881	Asheville	Mebane	195.9
546	Morrisville	Banner Elk	195.3
559	Boone	Cary	180.2

Month from START_DATE and the proportion of rides of different months.

```
# Month in numeric

data['MONTH_No.'] = data['START_DATE'].dt.month
data.groupby(['MONTH_No.'])['MILES'].count().sort_values(ascending = False)
```

```
MONTH_No.
1     101
11     86
3      85
8      84
7      82
12     76
4      75
10     74
6      72
5      70
9      70
2      69
Name: MILES, dtype: int64
```

Average distance covered each month.

```
data['MONTH'] = data['START_DATE'].dt.strftime('%B')
data.groupby(['MONTH'])['MILES'].mean().sort_values(ascending = False)
```

```
MONTH
October      16.994595
June         14.098611
July         11.310976
February     11.249275
March        10.107059
September     9.975714
August        8.955952
May           8.671429
April         8.132000
January       7.547525
November      6.938372
December      6.428947
Name: MILES, dtype: float64
```

Percentage of Business Miles covered and Personal miles covered

```
business_miles = data[data['CATEGORY']=='Business'].groupby(['CATEGORY'])['MILES'].sum()

personal_miles = data[data['CATEGORY']=='Personal'].groupby(['CATEGORY'])['MILES'].sum()

total_miles = data['MILES'].sum()
```

```
per_business_miles = (business_miles / total_miles)*100
print('Percentage of Business Miles Covered - ',per_business_miles)
```

```
Percentage of Business Miles Covered -  CATEGORY
Business    92.827198
Name: MILES, dtype: float64
```

```
per_personal_miles = (personal_miles / total_miles)*100

print('Percentage of Personal Miles Covered - ',per_personal_miles )
```

```
Percentage of Personal Miles Covered -  CATEGORY
Personal    7.172802
Name: MILES, dtype: float64
```

For which purpose Rides are most used for

```
data.groupby(['PURPOSE']).agg({'MILES':['sum','count']}).sort_values(('MILES','sum') , ascending=False)\
    .groupby(level=0)\
    .head(2)
```

	MILES	
	sum	count
PURPOSE		
Meeting	2435.2	164
Customer Visit	1995.2	92
Meal/Entertain	824.3	148
Errand/Supplies	430.9	111
Temporary Site	328.0	32
Between Offices	197.0	18
Commute	180.2	1
Moving	18.2	4
Charity ($)	15.1	1
Airport/Travel	4.1	1

Insights

- Most visited Start and End location is Carry
- Maximum rides were used for business purposes i.e. 97 %
- Two ways trips 202 out of 1156 total trips i.e 17.5 %

Back to Agenda

Back to Agenda

Recommendations

Data show which are peak months for travel miles , this helps to manage more rides at those peak months.

Two ways tripes should be increased to cover charges of one way travel as many places are same as start point and end point

Reference

All the resiurec to do analysis and present ypur analyis can be found in the resources shared here.

- Data set
- Python Notebook link
- Python notebook link with alllearning and different methods
- Python code images
- Graphic images

Add Company Name

Get In Touch

Email

prateek.kumar.agarwal25@gmail.com

Social Media

linkedin.com/in/prateek-kumar-agarwal-248015136/

Github

https://github.com/prateek071995

Back to Agenda