Import Libraries and Dataset

```
import warnings
import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats
import matplotlib.pyplot as plt
import matplotlib

import scipy.stats as stat
from scipy.stats import ttest_1samp
from scipy.stats import ttest_ind

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (15, 9)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

Import Cab Data

```
[5] from google.colab import files
files.upload()
```

YORK NY,30.72,567.08,412.8768\r\n10438756,43103,Yellow Cab,NEW YORK NY,32.2,620.82,463.68\r\n10438758,43105,Yellow Cab,NEW YORK NY,9.99,213.1,121.0788\r\n10438759,43104,Yellow Cab,NEW YORK NY,20.9,450.11,295.944\r\n10438760,43105,Yellow Cab,NEW YORK NY,3.09,67.32,41.9004\r\n10438761,43106,Yellow Cab,NEW YORK NY,13.8,305.69,182.16\r\n10438762,43103,Yellow Cab,NEW YORK NY,28.08,604.74,350.4384\r\n10438763,43103,Yellow Cab,NEW YORK NY,32.01,623.72,449.4204\r\n10438764,43109,Yellow Cab,NEW YORK NY,33.92,758.34,451.8144\r\n10438766,43105,Yellow Cab,NEW YORK NY,28,661.25,399.84\r\n10438767,43105,Yellow Cab,NEW YORK NY,32.67,774.88,403.8012\r\n10438768,43105,Yellow Cab,NEW YORK NY,22.6,440.36,311.88\r\n10438769,43106,Yellow Cab,NEW YORK NY,33.6,661.59,431.424\r\n10438770,43109,Yellow Cab,NEW YORK NY,43.66,841.77,612.9864\r\n10438771,43106,Yellow Cab,NEW YORK NY,14.42,266.19,173.04\r\n10438772,43107,Yellow Cab,NEW YORK NY,17.1,368.27,219.564\r\n10438773,43105,Yellow Cab,NEW YORK NY,14.82,326.83,211.6296\r\n10438774,43109,Yellow Cab,NEW YORK NY,40.8,861.94,538.56\r\n10438775,43105,Yellow Cab,NEW YORK NY,16.24,403.01,196.8288\r\n10438776,43106,Yellow Cab,NEW YORK NY,8.26,157.56,104.076\r\n10438777,43107,Yellow Cab,NEW YORK NY,22.68,483.79,302.0976\r\n10438778,43106,Yellow Cab,NEW YORK NY,3.6,67.19,45.792\r\n10438779,43104,Yellow Cab,NEW YORK NY,11.5,221.72,149.04\r\n10438780,43102,Yellow Cab,NEW YORK NY,31.2,684.73,449.28\r\n10438781,43105,Yellow Cab,NEW YORK NY,29.4,590.95,352.8\r\n10438782,43103,Yellow Cab,NEW YORK NY,41.6,802.05,549.12\r\n10438783,43106,Yellow Cab,NEW YORK NY,19.72,404.47,274.5024\r\n10438784,43107,Yellow Cab,NEW YORK NY,7.07,133.41,97.566\r\n10438785,43106,Yellow Cab,NEW YORK NY,20.52,412.46,268.4016\r\n10438786,43109,Yellow Cab,NEW YORK NY,40.33,868.55,488.7996\r\n10438787,43108,Yellow Cab,NEW YORK NY,18.9,379.9,263.088\r\n10438788,43105,Yellow Cab,NEW YORK NY,23.23,493.14,314.9988\r\n10438789,43103,Yellow Cab,NEW YORK NY,2.85,53.78,34.2\r\n10438790,43465,Yellow Cab,NEW YORK NY,8.88,191.24,125.7408\r\n10438791,43106,Yellow Cab,NEW YORK NY,10.26,225.17,131.7384\r\n10438792,43105,Yellow Cab,NEW YORK NY,27.56,701.62,334.0272\r\n10438793,43104,Yellow Cab,NEW YORK NY,25.48,548.74,333.2784\r\n10438794,43105,Yellow Cab,NEW YORK NY,26.16,563.39,345.312\r\n10438795,43102,Yellow Cab,NEW YORK NY,25.92,510.37,342.144\r\n10438796,43105,Yellow Cab,NEW YORK NY,10,227.87,129.6\r\n10438797,43107,Yellow Cab,NEW YORK NY,8.8,187.71,107.712\r\n10438798,43105,Yellow Cab,NEW YORK NY,20.9,424.39,285.912\r\n10438799,43105,Yellow Cab,NEW YORK NY,7.42,199.22,105.0672\r\n10438800,43108,Yellow Cab,NEW YORK NY,38.85,852.62,480.186\r\n10438801,43104,Yellow Cab,NEW YORK NY,16.24,309.78,214.368\r\n10438802,43104,Yellow Cab,NEW YORK NY,3.21,62.55,45.8388\r\n10438803,43106,Yellow Cab,NEW YORK NY,26.25,511.48,327.6\r\n10438805,43104,Yellow Cab,NEW YORK NY,42.18,856.49,551.7144\r\n10438806,43107,Yellow Cab,NEW YORK NY,14.69,307.33,185.094\r\n10438807,43105,Yellow Cab,NEW YORK NY,28.86,568.26,415.584\r\n10438808,43107,Yellow Cab,NEW YORK NY,36.75,798.99,498.33\r\n10438809,43109,Yellow Cab,NEW YORK NY,12.48,263.65,155.7504\r\n10438810,43104,Yellow Cab,NEW YORK NY,10.8,217.08,136.08\r\n10438811,43108,Yellow Cab,NEW YORK NY,26.22,580.81,371.2752\r\n10438812,43107,Yellow Cab,NEW YORK NY,12.21,227.9,164.1024\r\n10438813,43105,Yellow Cab,NEW YORK NY,12.12,283.99,146.8944\r\n10438816,43104,Yellow Cab,NEW YORK NY,13.91,262.48,180.2736\r\n10438817,43106,Yellow Cab,NEW YORK NY,3.54,71.88,44.1792\r\n10438818,43106,Yellow Cab,NEW YORK NY,7.84,159.2,94.08\r\n10438819,43106,Yellow Cab,NEW YORK NY,15.52,350.16,210.4512\r\n10438820,43101,Yellow Cab,NEW YORK NY,26.4,503.58,329.472\r\n10438821,43102,Yellow Cab,NEW YORK NY,38.4,803.36,548.352\r\n10438822,43104,Yellow Cab,NEW YORK NY,2.1,40.49,26.46\r\n10438823,43108,Yellow Cab,NEW YORK NY,22.47,488.53,293.9076\r\n10438824,43105,Yellow Cab,NEW YORK NY,31.2,654.65,415.584\r\n10438825,43102,Yellow Cab,NEW YORK NY,46.8,883.11,645.84\r\n10438826,43102,Yellow Cab,NEW YORK NY,21.66,448.7,306.7056\r\n10438827,43109,Yellow Cab,NEW YORK NY,40.66,792.27,575.7456\r\n10438828,43104,Yellow Cab,NEW YORK NY,24.48,487.04,331.9488\r\n10438829,43105,Yellow Cab,NEW YORK NY,38.85,881.14,484.848\r\n10438830,43105,Yellow Cab,NEW YORK NY,31.86,620.8,420.552\r\n10438831,43104,Yellow Cab,NEW YORK NY,32.55,701,445.284\r\n10438833,43102,Yellow Cab,NEW YORK NY,24.78,533.67,350.8848\r\n10438834,43465,Yellow Cab,NEW YORK NY,31.64,590.55,429.0384\r\n10438835,43105,Yellow Cab,NEW YORK NY,9.81,203.22,128.3148\r\n10438836,43105,Yellow Cab,NEW YORK NY,19.8,455.25,254.232\r\n10438837,43104,Yellow Cab,NEW YORK NY,29.1,543.14,401.58\r\n10438838,43105,Yellow Cab,NEW YORK NY,9.2,198.13,117.024\r\n10438839,43108,Yellow Cab,NEW YORK NY,24,536.56,325.44\r\n10438840,43104,Yellow Cab,NEW YORK NY,10.17,204.42,137.9052\r\n10438842,43109,Yellow Cab,NEW YORK NY,22.88,511.52,326.7264\r\n10438843,43105,Yellow Cab,NEW YORK NY,28.8,773.24,369.792\r\n10438844,43465,Yellow Cab,NEW YORK NY,32.55,654.27,464.814\r\n10438845,43105,Yellow Cab,NEW YORK NY,12.87,319.38,171.4284\r\n10438846,43465,Yellow Cab,NEW YORK NY,14.16,296.24,190.3104\r\n10438847,43103,Yellow Cab,NEW YORK NY,36.04,731.81,445.4544\r\n10438849,43103,Yellow Cab,NEW YORK NY,3.92,82.01,52.6848\r\n10438851,43107,Yellow Cab,NEW YORK NY,3.03,59.04,41.4504\r\n10438852,43102,Yellow Cab,NEW YORK NY,8.55,178.87,123.12\r\n10438853,43106,Yellow Cab,NEW YORK NY,21.4,416.98,282.48\r\n10438854,43107,Yellow Cab,NEW YORK NY,16.8,358.36,205.632\r\n10438855,43107,Yellow Cab,NEW YORK NY,19.38,389.55,234.8856\r\n10438856,43465,Yellow Cab,NEW YORK NY,26.88,567.87,377.3952\r\n10438857,43104,Yellow Cab,NEW YORK NY,44.46,893.67,597.5424\r\n10438858,43105,Yellow Cab,NEW YORK NY,36.26,773.47,513.4416\r\n10438860,43105,Yellow Cab,NEW YORK NY,8.19,219.89,104.1768\r\n10438861,43105,Yellow Cab,NEW YORK NY,22.31,411.83,289.1376\r\n10438862,43105,Yellow Cab,NEW YORK NY,3.09,67.92,40.4172\r\n10438863,43103,Yellow Cab,NEW YORK NY,39.96,827.8,489.1104\r\n10438864,43465,Yellow Cab,NEW YORK NY,33.6,661.59,439.488\r\n10438865,43105,Yellow Cab,NEW YORK NY,45.22,871.84,613.1832\r\n10438866,43103,Yellow Cab,NEW YORK NY,4.56,94.46,64.5696\r\n10438867,43102,Yellow Cab,NEW YORK NY,4.2,80.11,51.408\r\n10438868,43108,Yellow Cab,NEW YORK NY,42.8,860.3,513.6\r\n10438869,43104,Yellow Cab,NEW YORK NY,13.68,252.53,192.0672\r\n10438870,43105,Yellow Cab,NEW YORK NY,30,658.39,388.8\r\n10438871,43101,Yellow Cab,NEW YORK NY,28.8,626.15,380.16\r\n10438873,43102,Yellow Cab,NEW YORK NY,27.75,597.63,356.31\r\n10438874,43104,Yellow Cab,NEW YORK NY,8.32,187.71,113.8176\r\n10438875,43104,Yellow Cab,NEW YORK

NY,8.32,163.82,112.8192\r\n10438876,43101,Yellow Cab,NEW YORK NY,27.12,550.69,354.7296\r\n10438877,43104,Yellow Cab,NEW YORK NY,33.32,717.59,407.8368\r\n10438879,43104,Yellow Cab,NEW YORK NY,10.89,241.23,145.0548\r\n10438880,43107,Yellow Cab,NEW YORK NY,12.43,280.44,174.5172\r\n10438881,43105,Yellow Cab,NEW YORK NY,35.7,901.89,479.808\r\n10438882,43105,Yellow Cab,NEW YORK NY,22,406.11,266.64\r\n10438883,43105,Yellow Cab,NEW YORK NY,18,398.73,233.28\r\n10438884,43105,Yellow Cab,NEW YORK NY,3.03,76.42,36.36\r\n10438885,43108,Yellow Cab,NEW YORK NY,1.92,39.77,26.0352\r\n10438886,43106,Yellow Cab,NEW YORK NY,33.92,653.98,407.04\r\n10438887,43109,Yellow Cab,NEW YORK NY,29.5,647.42,378.78\r\n10438891,43106,Yellow Cab,NEW YORK NY,34.88,765.49,443.6736\r\n10438892,43103,Yellow Cab,NEW YORK NY,28.28,585.84,349.5408\r\n10438893,43103,Yellow Cab,NEW YORK NY,43.68,904.86,618.5088\r\n10438894,43105,Yellow Cab,NEW YORK NY,11.9,219.67,162.792\r\n10438895,43102,Yellow Cab,NEW YORK NY,39.44,849.39,544.272\r\n10438896,43103,Yellow Cab,NEW YORK NY,25.2,465.18,302.4\r\n10438898,43107,Yellow Cab,NEW YORK NY,31.35,623.72,447.678\r\n10438899,43105,Yellow Cab,NEW YORK NY,32.32,649.65,442.1376\r\n10438901,43105,Yellow Cab,NEW YORK NY,5,95.37,67.2\r\n10438902,43105,Yellow Cab,NEW YORK NY,39.9,851.11,526.68\r\n10438903,43106,Yellow Cab,NEW YORK NY,25.65,515.58,369.36\r\n10438905,43101,Yellow Cab,NEW YORK NY,8.32,163.82,111.8208\r\n10438906,43105,Yellow Cab,NEW YORK NY,42.4,930.53,610.56\r\n10438907,43105,Yellow Cab,NEW YORK NY,13,266.37,176.28\r\n10438908,43465,Yellow Cab,NEW YORK NY,32.98,676.44,451.1664\r\n10438910,43106,Yellow Cab,NEW YORK NY,14.85,332,206.712\r\n10438912,43108,Yellow Cab,NEW YORK NY,23.92,515.15,318.6144\r\n10438913,43102,Yellow Cab,NEW YORK NY,17.51,362.73,247.9416\r\n10438914,43101,Yellow Cab,NEW YORK NY,24.99,466.43,320.8716\r\n10438915,43102,Yellow Cab,NEW YORK NY,25.08,534.98,340.0848\r\n10438916,43106,Yellow Cab,NEW YORK NY,25.48,496.48,308.8176\r\n10438917,43105,Yellow Cab,NEW YORK NY,20.52,527.28,293.0256\r\n10438918,43105,Yellow Cab,NEW YORK NY,29.4,566.83,388.08\r\n10438920,43104,Yellow Cab,NEW YORK NY,29.16,598.09,416.4048\r\n10438921,43102,Yellow Cab,NEW YORK NY,28.28,591.64,403.8384\r\n10438922,43109,Yellow Cab,NEW YORK NY,26.73,531.8,362.4588\r\n10438923,43108,Yellow Cab,NEW YORK NY,38.08,804.48,502.656\r\n10438924,43102,Yellow Cab,NEW YORK NY,35.4,769.64,475.776\r\n10438925,43106,Yellow Cab,NEW YORK NY,12.76,280.04,156.1824\r\n10438927,43103,Yellow Cab,NEW YORK NY,37.83,799.2,494.8164\r\n10438928,43105,Yellow Cab,NEW YORK NY,22.08,407.59,304.704\r\n10438929,43109,Yellow Cab,NEW YORK NY,18,402.42,241.92\r\n10438930,43105,Yellow Cab,NEW YORK NY,13.44,300.47,161.28\r\n10438931,43104,Yellow Cab,NEW YORK NY,45.63,1001.41,602.316\r\n10438932,43105,Yellow Cab,NEW YORK NY,5.95,143.15,72.828\r\n10438934,43107,Yellow Cab,NEW YORK NY,5.55,118.39,69.93\r\n10438935,43105,Yellow Cab,NEW YORK NY,38.22,705.53,490.7448\r\n10438936,43105,Yellow Cab,NEW YORK NY,17.68,422.06,216.4032\r\n10438937,43465,Yellow Cab,NEW YORK NY,17.85,340.49,231.336\r\n10438939,43103,Yellow Cab,NEW YORK NY,22.23,446.83,296.1036\r\n10438940,43107,Yellow Cab,NEW YORK NY,14.85,298.49,199.584\r\n10438941,43102,Yellow Cab,NEW YORK NY,16.15,361.06,213.18\r\n10438942,43105,Yellow Cab,NEW

```
cab_df = pd.read_csv('Cab_Data.csv')
cab_df.head(10)
```

	Transaction ID	Date of Travel	Company	City	KM Travelled	Price Charged	Cost of Trip
0	10000011	42377	Pink Cab	ATLANTA GA	30.45	370.95	313.635
1	10000012	42375	Pink Cab	ATLANTA GA	28.62	358.52	334.854
2	10000013	42371	Pink Cab	ATLANTA GA	9.04	125.20	97.632
3	10000014	42376	Pink Cab	ATLANTA GA	33.17	377.40	351.602
4	10000015	42372	Pink Cab	ATLANTA GA	8.73	114.62	97.776
5	10000016	42376	Pink Cab	ATLANTA GA	6.06	72.43	63.024
6	10000017	42372	Pink Cab	AUSTIN TX	44.00	576.15	475.200
7	10000018	42376	Pink Cab	AUSTIN TX	35.65	466.10	377.890
8	10000019	42381	Pink Cab	BOSTON MA	14.40	191.61	146.880
9	10000020	42375	Pink Cab	BOSTON MA	10.89	156.98	113.256

[7]
```
#Data Info
cab_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 359392 entries, 0 to 359391
Data columns (total 7 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   Transaction ID  359392 non-null  int64  
 1   Date of Travel  359392 non-null  int64  
 2   Company         359392 non-null  object 
 3   City            359392 non-null  object 
 4   KM Travelled    359392 non-null  float64
 5   Price Charged   359392 non-null  float64
 6   Cost of Trip    359392 non-null  float64
dtypes: float64(3), int64(2), object(2)
memory usage: 19.2+ MB
```

[8]
```
#Data Type
cab_df.dtypes
```

```
Transaction ID      int64
Date of Travel      int64
Company            object
City               object
KM Travelled      float64
Price Charged     float64
Cost of Trip      float64
dtype: object
```

[9]
```
#Datatype ofDate of Travel datatype --> DateTime
a = cab_df['Date of Travel'].to_list()
base_date = pd.Timestamp('1899-12-29')
dates = [base_date + pd.DateOffset(date_offset) for date_offset in a]
cab_df['Date of Travel'] = pd.to_datetime(dates, format='%y/%m/%d')
```

[10]
```
#Data Description
cab_df.describe
```

```
<bound method NDFrame.describe of         Transaction ID Date of Travel     Company           City  \
0             10000011     2016-01-07    Pink Cab     ATLANTA GA   
1             10000012     2016-01-05    Pink Cab     ATLANTA GA   
2             10000013     2016-01-01    Pink Cab     ATLANTA GA   
3             10000014     2016-01-06    Pink Cab     ATLANTA GA   
4             10000015     2016-01-02    Pink Cab     ATLANTA GA   
...                ...            ...         ...            ...   
359387        10440101     2018-01-07  Yellow Cab  WASHINGTON DC   
359388        10440104     2018-01-03  Yellow Cab  WASHINGTON DC   
359389        10440105     2018-01-04  Yellow Cab  WASHINGTON DC   
359390        10440106     2018-01-04  Yellow Cab  WASHINGTON DC   
359391        10440107     2018-01-01  Yellow Cab  WASHINGTON DC   

        KM Travelled  Price Charged  Cost of Trip  
0              30.45         370.95      313.6350  
1              28.62         358.52      334.8540  
2               9.04         125.20       97.6320  
3              33.17         377.40      351.6020  
4               8.73         114.62       97.7760  
...              ...            ...           ...  
359387          4.80          69.24       63.3600  
359388          8.40         113.75      106.8480  
359389         27.75         437.07      349.6500  
359390          8.80         146.19      114.0480  
359391         12.76         191.58      177.6192  

[359392 rows x 7 columns]>
```

```
[359392 rows x 7 columns]>
```

[11]
```
#Column of Company
cab_df['Company'].unique()
```

```
array(['Pink Cab', 'Yellow Cab'], dtype=object)
```

[12]
```
#Column of City
cab_df['City'].unique()
```

```
array(['ATLANTA GA', 'AUSTIN TX', 'BOSTON MA', 'CHICAGO IL', 'DALLAS TX',
       'DENVER CO', 'LOS ANGELES CA', 'MIAMI FL', 'NASHVILLE TN',
       'NEW YORK NY', 'ORANGE COUNTY', 'PHOENIX AZ', 'PITTSBURGH PA',
       'SACRAMENTO CA', 'SAN DIEGO CA', 'SEATTLE WA', 'SILICON VALLEY',
       'TUCSON AZ', 'WASHINGTON DC'], dtype=object)
```

Import City Data

[13]
```
from google.colab import files
files.upload()
```

Choose Files City.csv

- **City.csv**(text/csv) - 759 bytes, last modified: 10/1/2022 - 100% done

```
Saving City.csv to City.csv
{'City.csv': b'City,Population,Users\r\nNEW YORK NY," 8,405,837 "," 302,149 "\r\nCHICAGO IL," 1,955,130 "," 164,468 "\r\nLOS ANGELES CA," 1,595,037 "," 144,132 "\r\nMIAMI FL," 1,339,155
"," 17,675 "\r\nSILICON VALLEY," 1,177,609 "," 27,247 "\r\nORANGE COUNTY," 1,030,185 "," 12,994 "\r\nSAN DIEGO CA," 959,307 "," 69,995 "\r\nPHOENIX AZ," 943,999 "," 6,133 "\r\nDALLAS
TX," 942,908 "," 22,157 "\r\nATLANTA GA," 814,885 "," 24,701 "\r\nDENVER CO," 754,233 "," 12,421 "\r\nAUSTIN TX," 698,371 "," 14,978 "\r\nSEATTLE WA," 671,238 "," 25,063 "\r\nTUCSON
AZ," 631,442 "," 5,712 "\r\nSAN FRANCISCO CA," 629,591 "," 213,609 "\r\nSACRAMENTO CA," 545,776 "," 7,044 "\r\nPITTSBURGH PA," 542,085 "," 3,643 "\r\nWASHINGTON DC," 418,859 "," 127,001
"\r\nNASHVILLE TN," 327,225 "," 9,270 "\r\nBOSTON MA," 248,968 "," 80,021 "\r\n'}
```

[14]
```
city_df = pd.read_csv('City.csv')
city_df.head(5)
```

	City	Population	Users
0	NEW YORK NY	8,405,837	302,149
1	CHICAGO IL	1,955,130	164,468
2	LOS ANGELES CA	1,595,037	144,132
3	MIAMI FL	1,339,155	17,675
4	SILICON VALLEY	1,177,609	27,247

[15]
```
#Data Info
city_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        20 non-null     object
 1   Population  20 non-null     object
 2   Users       20 non-null     object
dtypes: object(3)
memory usage: 608.0+ bytes
```

[16]
```
# 'Population' --> integer
city_df['Population'] = [x.replace(',','') for x in city_df['Population']]
city_df['Population'] = city_df['Population'].astype(float)

#  'Users' --> integer
city_df['Users'] = [x.replace(',','') for x in city_df['Users']]
city_df['Users'] = city_df['Users'].astype(float)
```

[17]
```
#data types
city_df.dtypes
```

```
City           object
Population    float64
Users         float64
dtype: object
```

[18]
```
#Description
city_df.describe()
```

	Population	Users
count	2.000000e+01	20.000000
mean	1.231592e+06	64520.650000
std	1.740127e+06	83499.375289
min	2.489680e+05	3643.000000
25%	6.086372e+05	11633.250000
50%	7.845590e+05	23429.000000
75%	1.067041e+06	91766.000000
max	8.405837e+06	302149.000000

Import Transaction ID Data

[19]
```
from google.colab import files
files.upload()
```

Choose Files Transaction_ID.csv

- **Transaction_ID.csv**(text/csv) - 8998194 bytes, last modified: 10/1/2022 - 100% done

```
Saving Transaction_ID.csv to Transaction_ID.csv
{'Transaction_ID.csv': b'Transaction ID,Customer
ID,Payment_Mode\r\n10000011,29290,Card\r\n10000012,27703,Card\r\n10000013,28712,Cash\r\n10000014,28020,Cash\r\n10000015,27182,Card\r\n10000016,27318,Cash\r\n10000017,33788,Card\r\n10000
```

[20]
```
transaction_id_df = pd.read_csv('Transaction_ID.csv')
transaction_id_df.head()
```

	Transaction ID	Customer ID	Payment_Mode
0	10000011	29290	Card
1	10000012	27703	Card
2	10000013	28712	Cash
3	10000014	28020	Cash
4	10000015	27182	Card

[21]
```
#Data Info
transaction_id_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440098 entries, 0 to 440097
Data columns (total 3 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Transaction ID  440098 non-null  int64
 1   Customer ID     440098 non-null  int64
 2   Payment_Mode    440098 non-null  object
dtypes: int64(2), object(1)
memory usage: 10.1+ MB
```

[22]
```
#Data Description
transaction_id_df.describe(include = 'all', datetime_is_numeric=True)
```

	Transaction ID	Customer ID	Payment_Mode
count	4.400980e+05	440098.000000	440098
unique	NaN	NaN	2
top	NaN	NaN	Card
freq	NaN	NaN	263991
mean	1.022006e+07	23619.513120	NaN
std	1.270455e+05	21195.549816	NaN
min	1.000001e+07	1.000000	NaN
25%	1.011004e+07	3530.000000	NaN
50%	1.022006e+07	15168.000000	NaN
75%	1.033008e+07	43884.000000	NaN
max	1.044011e+07	60000.000000	NaN

Import Customer ID Data

[23]
```
from google.colab import files
files.upload()
```

Choose Files Customer_ID.csv

- **Customer_ID.csv**(text/csv) - 1051215 bytes, last modified: 10/1/2022 - 100% done

```
Saving Customer_ID.csv to Customer_ID.csv
{'Customer_ID.csv': b'Customer ID,Gender,Age,Income
(USD/Month)\r\n29290,Male,28,10813\r\n27703,Male,27,9237\r\n28712,Male,53,11242\r\n28020,Male,23,23327\r\n27182,Male,33,8536\r\n27318,Male,25,13984\r\n33788,Male,23,23788\r\n34106,Male,
```

[24]
```
customer_id_df = pd.read_csv('Customer_ID.csv')
customer_id_df.head()
```

	Customer ID	Gender	Age	Income (USD/Month)
0	29290	Male	28	10813
1	27703	Male	27	9237
2	28712	Male	53	11242
3	28020	Male	23	23327
4	27182	Male	33	8536

[25]
```
#Data Info
customer_id_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49171 entries, 0 to 49170
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Customer ID         49171 non-null  int64
 1   Gender              49171 non-null  object
 2   Age                 49171 non-null  int64
 3   Income (USD/Month)  49171 non-null  int64
dtypes: int64(3), object(1)
memory usage: 1.5+ MB
```

[26]
```
#Data Description
customer_id_df.describe( include = 'all')
```

	Customer ID	Gender	Age	Income (USD/Month)
count	49171.000000	49171	49171.000000	49171.000000
unique	NaN	2	NaN	NaN
top	NaN	Male	NaN	NaN
freq	NaN	26562	NaN	NaN
mean	28398.252283	NaN	35.363121	15015.631856
std	17714.137333	NaN	12.599066	8002.208253
min	1.000000	NaN	18.000000	2000.000000
25%	12654.500000	NaN	25.000000	8289.500000
50%	27631.000000	NaN	33.000000	14656.000000
75%	43284.500000	NaN	42.000000	21035.000000
max	60000.000000	NaN	65.000000	35000.000000

Data Exploration and Visualize the Whole Dataset

```
[27] df= cab_df.merge(transaction_id_df, on= 'Transaction ID').merge(customer_id_df, on ='Customer ID').merge(city_df, on = 'City')
     df.head(5)
```

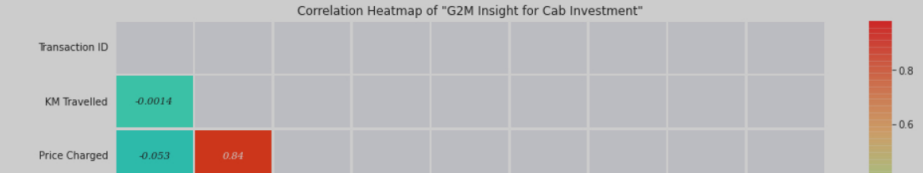
	Transaction ID	Date of Travel	Company	City	KM Travelled	Price Charged	Cost of Trip	Customer ID	Payment_Mode	Gender	Age	Income (USD/Month)	Population	Users
0	10000011	2016-01-07	Pink Cab	ATLANTA GA	30.45	370.95	313.6350	29290	Card	Male	28	10813	814885.0	24701.0
1	10351127	2018-07-20	Yellow Cab	ATLANTA GA	26.19	598.70	317.4228	29290	Cash	Male	28	10813	814885.0	24701.0
2	10412921	2018-11-22	Yellow Cab	ATLANTA GA	42.55	792.05	597.4020	29290	Card	Male	28	10813	814885.0	24701.0
3	10000012	2016-01-05	Pink Cab	ATLANTA GA	28.62	358.52	334.8540	27703	Card	Male	27	9237	814885.0	24701.0
4	10320494	2018-04-20	Yellow Cab	ATLANTA GA	36.38	721.10	467.1192	27703	Card	Male	27	9237	814885.0	24701.0

Correlation Check

```
[28] data_corr = df.corr()
     data_corr
```

	Transaction ID	KM Travelled	Price Charged	Cost of Trip	Customer ID	Age	Income (USD/Month)	Population	Users
Transaction ID	1.000000	-0.001429	-0.052902	-0.003462	-0.016912	-0.001267	-0.001570	0.023868	0.013526
KM Travelled	-0.001429	1.000000	0.835753	0.981848	0.000389	-0.000369	-0.000544	-0.002311	-0.000428
Price Charged	-0.052902	0.835753	1.000000	0.859812	-0.177324	-0.003084	0.003228	0.326589	0.281061
Cost of Trip	-0.003462	0.981848	0.859812	1.000000	0.003077	-0.000189	-0.000633	0.015108	0.023628
Customer ID	-0.016912	0.000389	-0.177324	0.003077	1.000000	-0.004735	-0.013608	-0.647052	-0.610742
Age	-0.001267	-0.000369	-0.003084	-0.000189	-0.004735	1.000000	0.003907	-0.009002	-0.005906
Income (USD/Month)	-0.001570	-0.000544	0.003228	-0.000633	-0.013608	0.003907	1.000000	0.011868	0.010464
Population	0.023868	-0.002311	0.326589	0.015108	-0.647052	-0.009002	0.011868	1.000000	0.915490
Users	0.013526	-0.000428	0.281061	0.023628	-0.610742	-0.005906	0.010464	0.915490	1.000000

```
[29] # Define the figure size
     plt.figure(figsize = (16, 9))

     # Cutomize the annot
     annot_kws={'fontsize':10,                    # To change the size of the font
                'fontstyle':'italic',             # To change the style of font
                'fontfamily': 'serif',            # To change the family of font
                'alpha':1 }                       # To change the transparency of the text


     # Customize the cbar
     cbar_kws = {"shrink":1,                      # To change the size of the color bar
                 'extend':'min',                  # To change the end of the color bar like pointed
                 'extendfrac':0.1,                # To adjust the extension of the color bar
                 "drawedges":True,                # To draw lines (edges) on the color bar
                }

     # take upper correlation matrix
     matrix = np.triu(data_corr)

     # Generate heatmap correlation
     ax = sns.heatmap(data_corr, mask = matrix, cmap = 'rainbow', annot = True, linewidth = 1.5 ,annot_kws= annot_kws, cbar_kws=cbar_kws)

     # Set the title etc
     plt.title('Correlation Heatmap of "G2M Insight for Cab Investment"')

     # Set the size of text
     sns.set(font_scale = 1.2)
```



From the heatmap correlation a strong -Correlation between

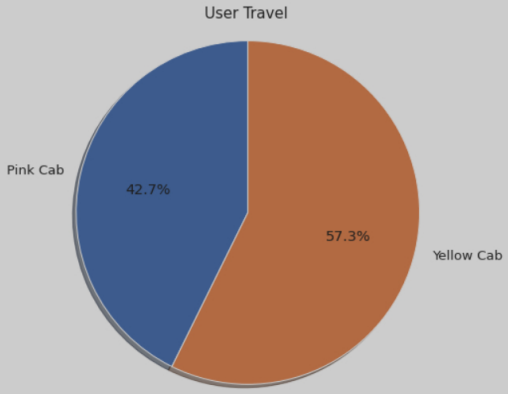
-Population vs Users

-Price Charged vs Cost of Trip vs KM Travelled

User's Travel in both Pink & Yellow Cab

```
[30] user=df.groupby('Company')
     avg_user = user.Users.mean()
     index = avg_user.index
     value = avg_user.values
```

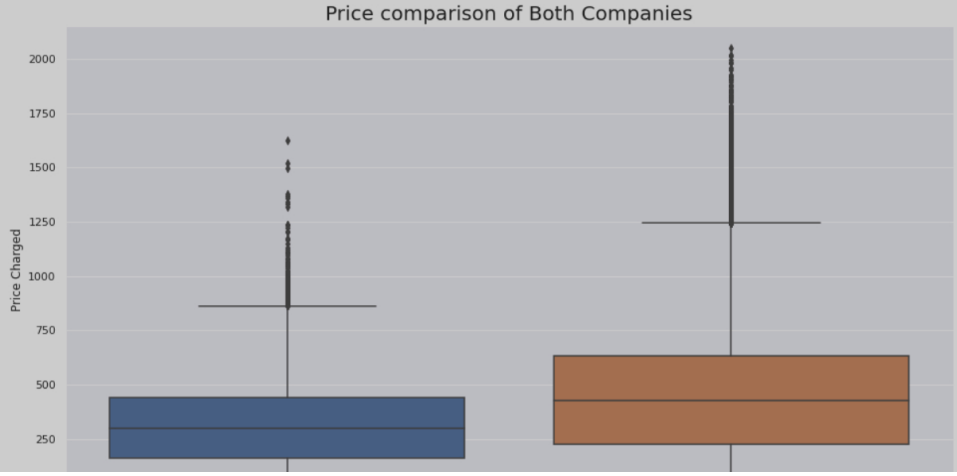
```
[31] figp, axp = plt.subplots(figsize=(10,7))
     axp.pie(value , labels=index, autopct='%1.1f%%',shadow=True, startangle=90,)
     axp.axis('equal')

     plt.title('User Travel', fontsize = 15)
     plt.show()
```



Users Prefer `Yellow` Cab than `Pink` Cab

Comparison Of Price

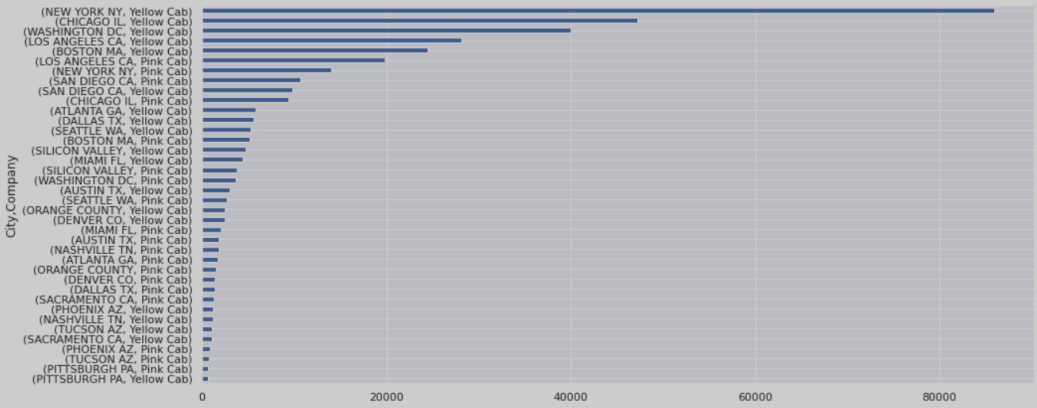
```
[32] sns.set(style = 'darkgrid')

     plt.figure(figsize = (16, 9))

     sns.boxplot(df['Company'], df['Price Charged'])
     plt.title('Price comparison of Both Companies', fontsize=20)
     plt.show()
```



So the `Yellow Cab` is higher as compared to `Pink Cab`

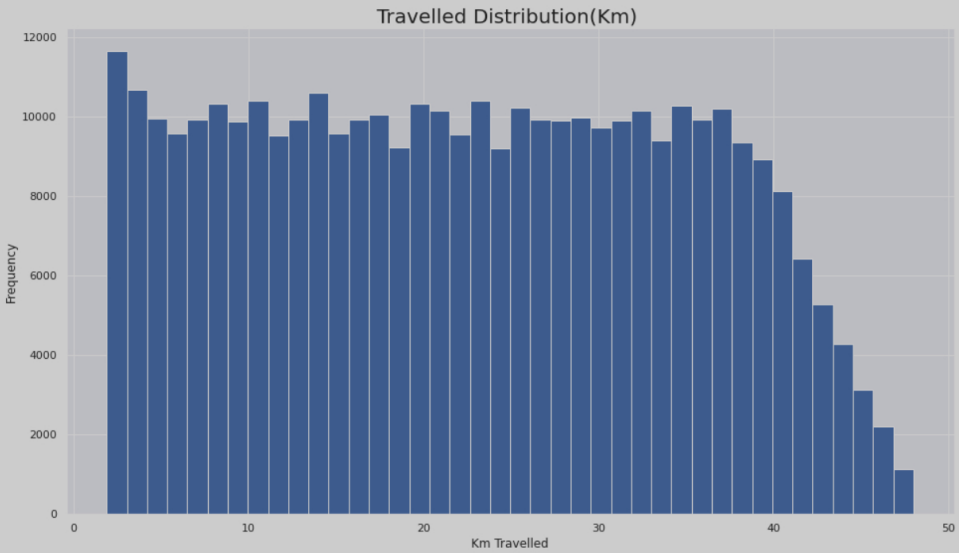
Demand of the 2 cab service providers

```
[33] plt.figure(figsize=(15, 7))
     df.groupby('City').Company.value_counts().sort_values(ascending=True).plot(kind='barh');
```



Travelled Distribution (KM)

```
[34] plt.figure(figsize = (16, 9))
     plt.hist(df['KM Travelled'], bins = 40)
     plt.title('Travelled Distribution(Km) ', fontsize=20)
     plt.ylabel('Frequency')
     plt.xlabel('Km Travelled')
     plt.show()
```



Most of the travelled rides varies from `2 - 48` KM.

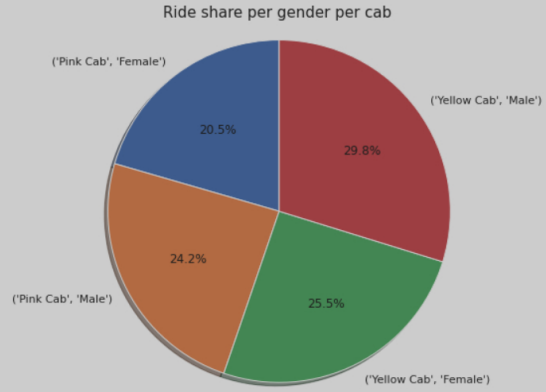
Gender

```
[35] gender_cab=df.groupby(['Company','Gender'])
     gender_cab  = gender_cab['Customer ID'].nunique()
     print(gender_cab)
```

```
Company     Gender
Pink Cab    Female    14819
            Male      17511
Yellow Cab  Female    18394
            Male      21502
Name: Customer ID, dtype: int64
```

```
[36] labs = gender_cab.index
     vals = gender_cab.values
     figp, axp = plt.subplots(figsize=(10,7))
```

```
axp.pie(vals , labels=labs, autopct= %1.1f%% ,shadow=True, startangle=90,)
axp.axis('equal')

plt.title('Ride share per gender per cab', fontsize = 15)
plt.show()
```

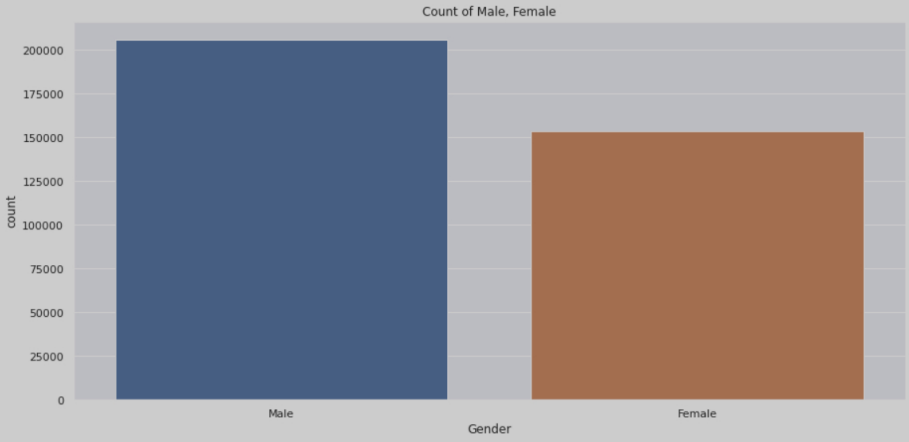


▶ *`Male` users are prefer more to travel in Cab*

▶ *Users prefer to travel in `Yellow Cab`*

No. of Male & female user

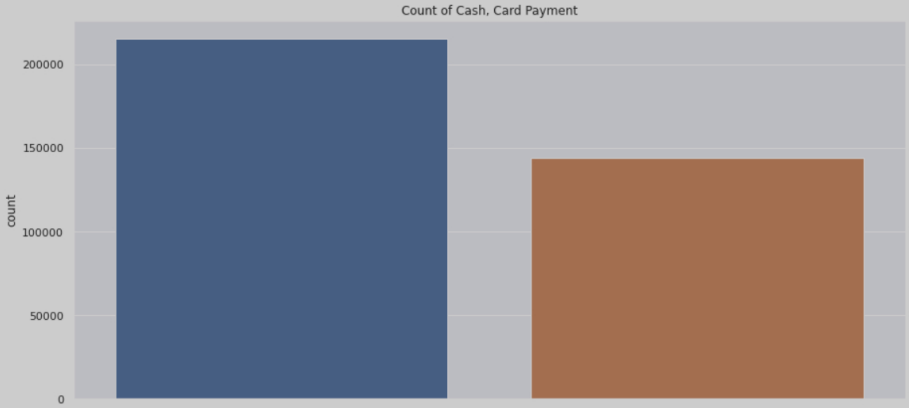
```
[37] plt.figure(figsize=(15,7))
     g=sns.countplot(x = 'Gender', data = df);
     g.set_title('Count of Male, Female')
     plt.show()
```



▶ *No. of male users are more than female users*

Payment Mode

```
[38] plt.figure(figsize=(15,7))
     g=sns.countplot(df['Payment_Mode']);
     g.set_title('Count of Cash, Card Payment')
     plt.show()
```



Card Cash

Payment_Mode

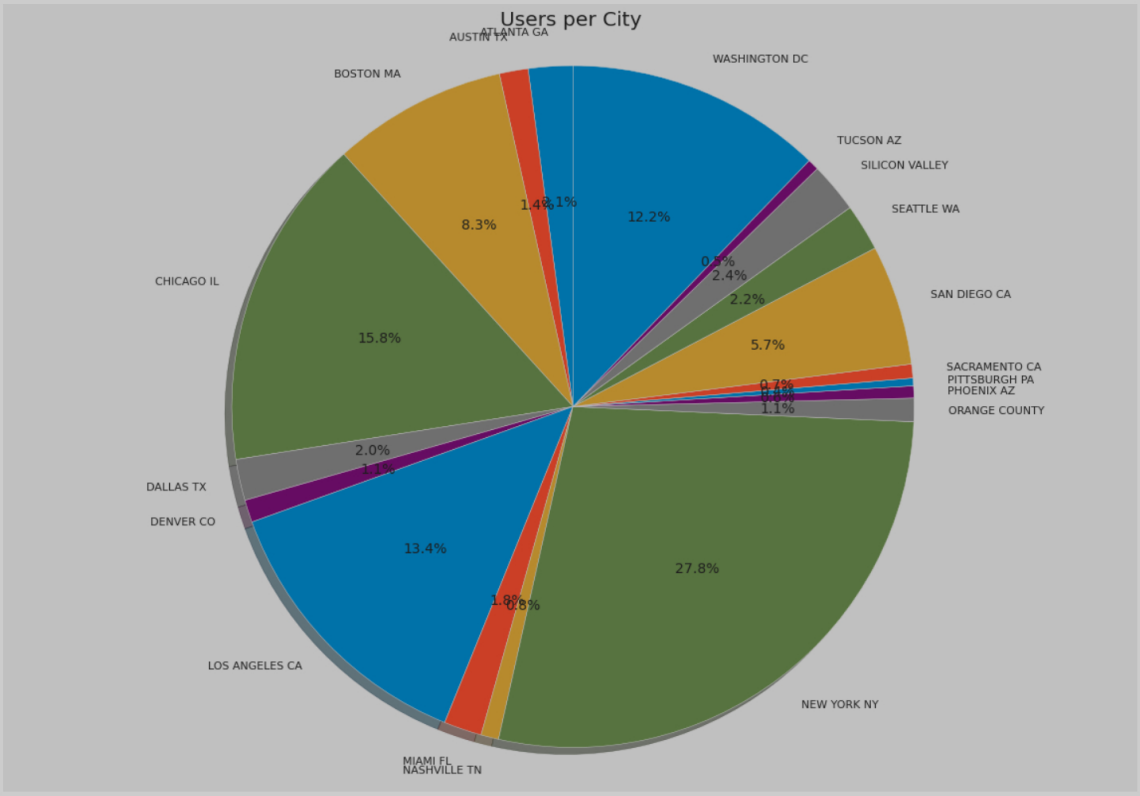
```
[39] df['Payment_Mode'].value_counts()
```

```
Card    215504
Cash    143888
Name: Payment_Mode, dtype: int64
```

▶ *No. of Card payment is more than no. of cash payment*

Users per City

```
[40] city_users = df.groupby('City')
     city_users = city_users.Users.count()
     labs = city_users.index
     vals = city_users.values

     plt.style.use('fivethirtyeight')
     figp, axp = plt.subplots(figsize=(18,13))
     axp.pie(vals , labels=labs, autopct='%1.1f%%',shadow=True, startangle=90,)
     axp.axis('equal')
     plt.title('Users per City')
     plt.show()
```

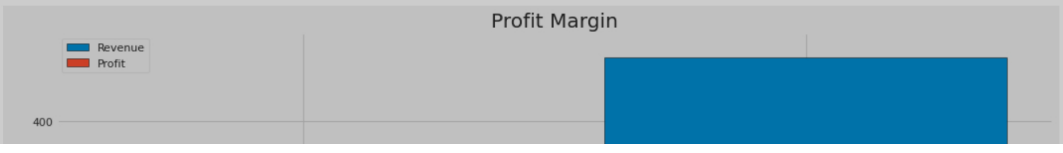


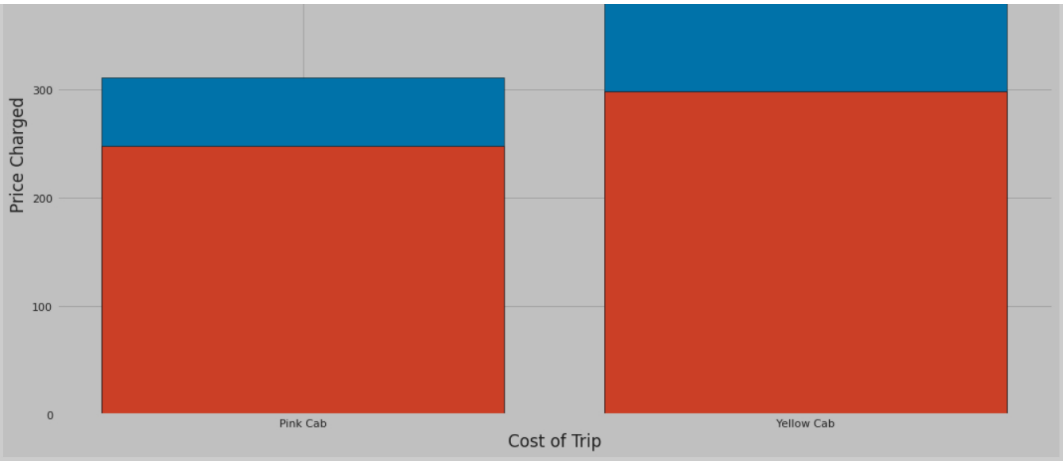
▶ *New York City - 28%* ▶*Chicago - 16%*` ▶*Los Angeles - 13%*

Profit & Cost Analysis

```
[41] company = df.groupby('Company')
     price_charged = company['Price Charged'].mean()
     cost_trip = company['Cost of Trip'].mean()
     c = cost_trip.index
     c_v = cost_trip.values
     c_p = price_charged.values
```

```
[42] plt.style.use('fivethirtyeight')
     plt.figure(figsize = (16, 9))
     plt.bar(c, c_p, edgecolor='black', label="Revenue")
     plt.bar(c, c_v, edgecolor='black', label="Profit")
     plt.title('Profit Margin')
     plt.ylabel('Price Charged')
     plt.xlabel('Cost of Trip')
     plt.legend()
     plt.show()
```



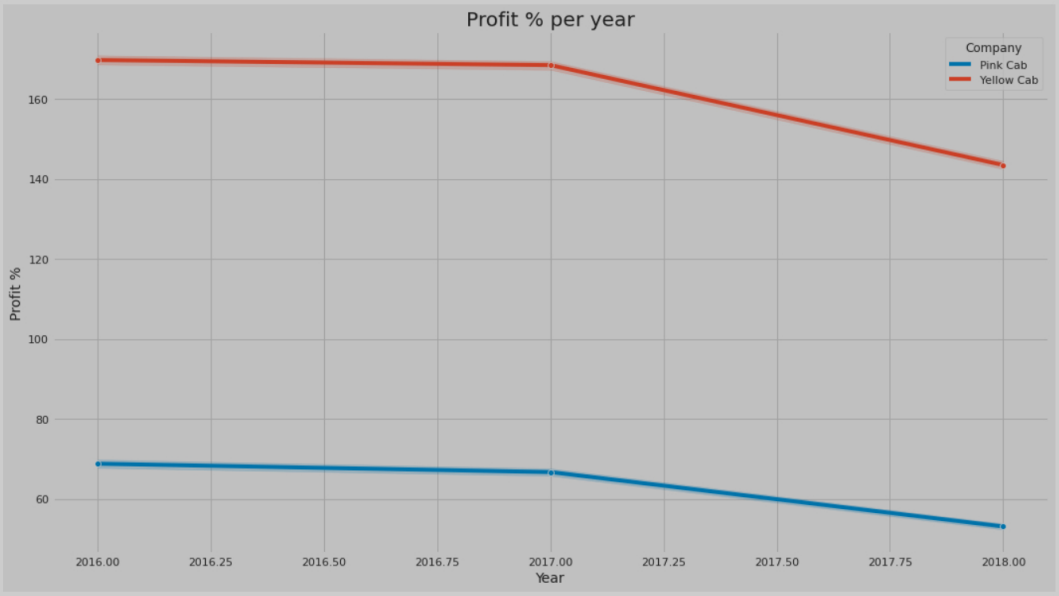


▶ *The `Yellow cab` has a higher Profit Margin compared to `Pink cab`*

Profit Per Year

```
[43] df['Year'] = df['Date of Travel'].dt.year
     df['Month'] = df['Date of Travel'].dt.month
     df['Day'] = df['Date of Travel'].dt.day
     df['Profit'] = df['Price Charged'] - df['Cost of Trip']
```

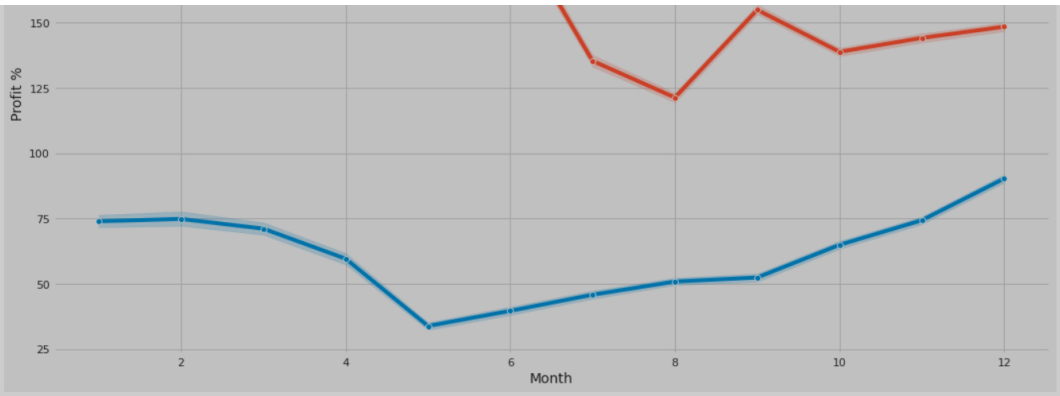
```
[44] plt.figure(figsize = (16, 9))
     sns.lineplot(x='Year', y='Profit', hue="Company", data=df, marker='o')
     plt.xlabel("Year", size=14)
     plt.ylabel("Profit %", size=14)
     plt.title("Profit % per year")
     plt.show()
```



▶ *The profit margin has `decreased` with respect to year*

Profit Per Month

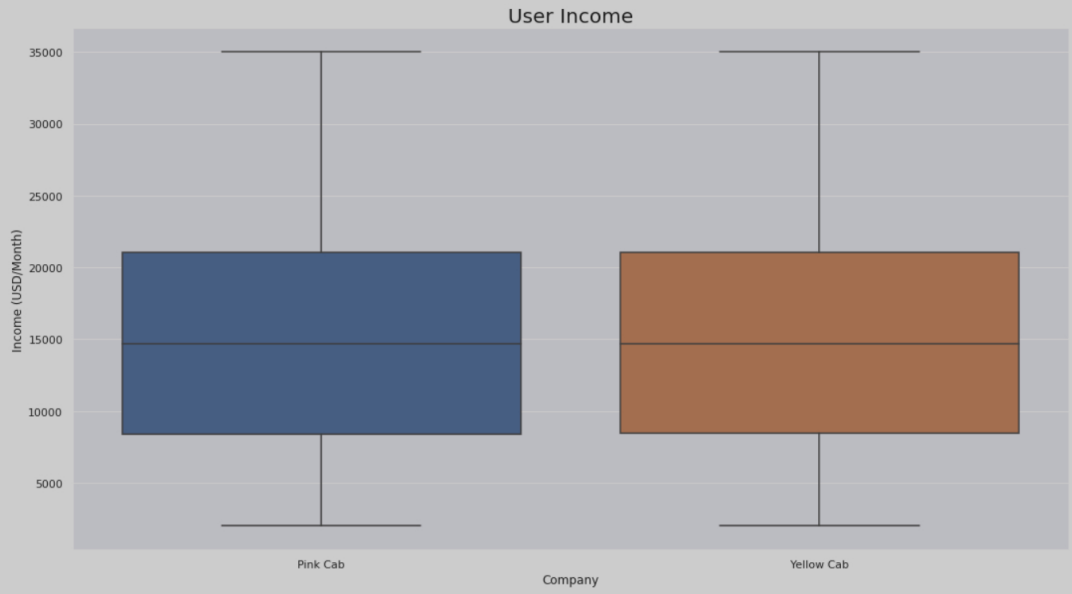
```
[45] plt.figure(figsize = (16, 9))
     sns.lineplot(x='Month', y='Profit', hue="Company", data=df, marker='o')
     plt.xlabel("Month", size=14)
     plt.ylabel("Profit %", size=14)
     plt.title("Profit % per month")
     plt.show()
```



▶ *The profit margin varies with respect to month*

Average Income

```
[46] sns.set(style = 'darkgrid')
     plt.figure(figsize = (16, 9))
     sns.boxplot(df['Company'], df['Income (USD/Month)'])
     plt.title('User Income', fontsize=20)
     plt.show()
```



▶ *Average income is around `USD15k` who use cab sevice*

Hypothesis

```
[47] df.describe()
```

	Transaction ID	KM Travelled	Price Charged	Cost of Trip	Customer ID	Age	Income (USD/Month)	Population	Users	Year	Month	Day	
count	3.593920e+05	359392.000000	359392.000000	359392.000000	359392.000000	359392.000000	359392.000000	3.593920e+05	359392.000000	359392.000000	359392.000000	359392.000000	359392
mean	1.022076e+07	22.567254	423.443311	286.190113	19191.652115	35.336705	15048.822937	3.132198e+06	158365.582267	2017.041693	7.509243	15.641700	137
std	1.268058e+05	12.233526	274.378911	157.993661	21012.412463	12.594234	7969.409482	3.315194e+06	100850.051020	0.801378	3.428929	8.838986	160
min	1.000001e+07	1.900000	15.600000	19.000000	1.000000	18.000000	2000.000000	2.489680e+05	3643.000000	2016.000000	1.000000	1.000000	-220
25%	1.011081e+07	12.000000	206.437500	151.200000	2705.000000	25.000000	8424.000000	6.712380e+05	80021.000000	2016.000000	5.000000	8.000000	28
50%	1.022104e+07	22.440000	386.360000	282.480000	7459.000000	33.000000	14685.000000	1.595037e+06	144132.000000	2017.000000	8.000000	16.000000	81
75%	1.033094e+07	32.960000	583.660000	413.683200	36078.000000	42.000000	21035.000000	8.405837e+06	302149.000000	2018.000000	11.000000	23.000000	190
max	1.044011e+07	48.000000	2048.030000	691.200000	60000.000000	65.000000	35000.000000	8.405837e+06	302149.000000	2018.000000	12.000000	31.000000	1463

Hypothesis-01 Charged Price Same or Not

▶ *H0 = Price charged by Pink & Yellow Cabs are same*

▶ H1 = Price charged by Pink & Yellow Cabs are not same

```
[48] sample_size = int((10/100)*359392) # Considering 10% values as sample data

     def T_Test(a, b):
         sample_a = np.random.choice(a, sample_size)
         sample_b = np.random.choice(b, sample_size)
         ttest, p_value = ttest_ind(sample_a, sample_b, equal_var = False)
         print(f'p-value: {p_value}')
         if p_value < 0.05:    # alpha value is 0.05 or 5%
             print("We are rejecting null hypothesis (H0)")
         else:
             print("We are accepting null hypothesis (H0)")
```

```
[49] df['KM Travelled'].groupby(df['Company']).mean()

     Company
     Pink Cab      22.559917
     Yellow Cab    22.569517
     Name: KM Travelled, dtype: float64
```

```
[50] T_Test(df[df['Company'] == 'Yellow Cab']['KM Travelled'], df[df['Company'] == 'Pink Cab']['KM Travelled'])

     p-value: 0.5033034260404716
     We are accepting null hypothesis (H0)
```

Hypothesis-02 Difference Between Age

```
[51] a = df[(df.Age <= 60)&(df.Company=='Pink Cab')].groupby('Transaction ID').Profit.mean()
     b = df[(df.Age >= 60)&(df.Company=='Pink Cab')].groupby('Transaction ID').Profit.mean()
     print(a.shape[0],b.shape[0])

     _, p_value = stats.ttest_ind(a.values,
                                  b.values,
                                  equal_var=True)

     print('P value is ', p_value)

     if(p_value<0.05):
         print('We accept alternative hypothesis (H1) that there is a difference regarding age for Pink Cab')
     else:
         print('We accept null hypothesis (H0) that there is no difference regarding age for Pink Cab')

     80125 5429
     P value is  0.4816748536155635
     We accept null hypothesis (H0) that there is no difference regarding age for Pink Cab
```

```
[52] a = df[(df.Age <= 60)&(df.Company=='Yellow Cab')].groupby('Transaction ID').Profit.mean()
     b = df[(df.Age >= 60)&(df.Company=='Yellow Cab')].groupby('Transaction ID').Profit.mean()
     print(a.shape[0],b.shape[0])

     _, p_value = stats.ttest_ind(a.values,
                                  b.values,
                                  equal_var=True)

     print('P value is ', p_value)

     if(p_value<0.05):
         print('We accept alternative hypothesis (H1) that there is a difference regarding age for Yellow Cab')
     else:
         print('We accept null hypothesis (H0) that there is no difference regarding age for Yellow Cab')

     260356 17257
     P value is  6.328485471267631e-05
     We accept alternative hypothesis (H1) that there is a difference regarding age for Yellow Cab
```

Hypothesis-03 Profit difference for both cab services

▶H0 = Profit is same for both cab services

▶H1 = Profit is not same for both cab services

```
[53] df['Profit'].groupby(df['Company']).mean()

     Company
     Pink Cab       62.652174
     Yellow Cab    160.259986
     Name: Profit, dtype: float64
```

```
[ ] T_Test(df[df['Company'] == 'Yellow Cab']['Profit'], df[df['Company'] == 'Pink Cab']['Profit'])

     p-value: 0.0
     We are rejecting null hypothesis (H0)
```

Hypothesis-04 Difference in Profit in Payment mode

▶H0 = Has profit difference in Payment_Mode in both cab companies

▶H1 = Has no profit difference in Payment_Mode in both cab companies

```
[54] a = df[(df['Payment_Mode']=='Cash')&(df.Company=='Pink Cab')].groupby('Transaction ID').Profit.mean()
     b = df[(df['Payment_Mode']=='Card')&(df.Company=='Pink Cab')].groupby('Transaction ID').Profit.mean()

     _, p_value = stats.ttest_ind(a.values,
                                  b.values,
                                  equal_var=True)

     print('P value is ', p_value)

     if(p_value<0.05):
         print('We accept alternative hypothesis (H1) that there is a difference in payment mode for Pink Cab')
     else:
         print('We accept null hypothesis (H0) that there is no difference in payment mode for Pink Cab')
```

```
P value is  0.7900465828793288
We accept null hypothesis (H0) that there is no difference in payment mode for Pink Cab
```

```
[55] a = df[(df['Payment_Mode']=='Cash')&(df.Company=='Yellow Cab')].groupby('Transaction ID').Profit.mean()
     b = df[(df['Payment_Mode']=='Card')&(df.Company=='Yellow Cab')].groupby('Transaction ID').Profit.mean()

     _, p_value = stats.ttest_ind(a.values,
                                  b.values,
                                  equal_var=True)

     print('P value is ', p_value)

     if(p_value<0.05):
         print('We accept alternative hypothesis (H1) that there is a difference in payment mode for Yellow Cab')
     else:
         print('We accept null hypothesis (H0) that there is no difference in payment mode for Yellow Cab')
```

```
P value is  0.2933060638298729
We accept null hypothesis (H0) that there is no difference in payment mode for Yellow Cab
```

Hypothesis-05 Travelled distance difference for both cab services

▶H0 = Distance travelled by both Cabs are same

▶H1 = Distance travelled by both Cabs are not same

```
[56] df['KM Travelled'].groupby(df['Company']).mean()
```

```
Company
Pink Cab      22.559917
Yellow Cab    22.569517
Name: KM Travelled, dtype: float64
```

```
[57] T_Test(df[df['Company'] == 'Yellow Cab']['KM Travelled'], df[df['Company'] == 'Pink Cab']['KM Travelled'])
```

```
p-value: 0.07571452137168612
We are accepting null hypothesis (H0)
```