Exploratory Data Analysis (EDA)
on
Cab Industry Datasets

By: Alhanouf Alghamdi

Date : August 9th, 2021

Agenda

- Executive Summary
- Problem Statement
- EDA
- EDA Summary
- Recommendation

Executive Summary

A private company named XYZ in USA want to make an investment in Cab industry due to its popularity in the market. They want to choose between two different cab industries that are Pink cab and Yellow cab.

There are four datasets that contain all the required information to help in making the decision.

This project is about to perform Exploratory Data Analysis known as EDA on these four datasets.

4 Datasets

- CabData
- City
- Customer_ID
- Transaction_ID

Problem Statement

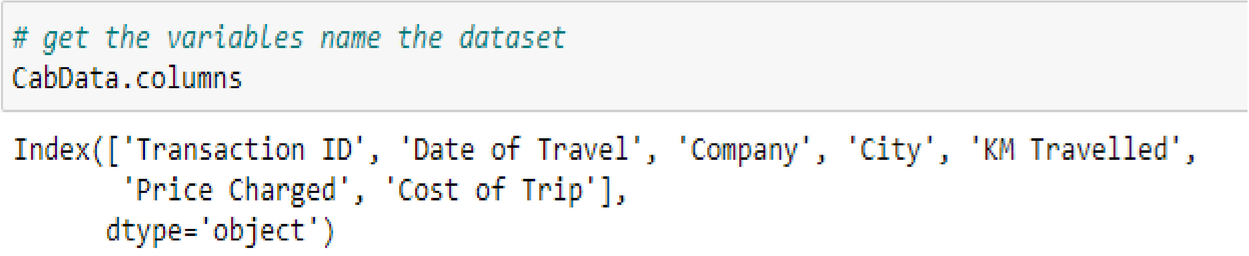
- Helping XYZ Company to decide which company (Pink or Yellow cabs) should they invest in based on different factors.

EDA

- Three steps were followed in this EDA for the all four datasets

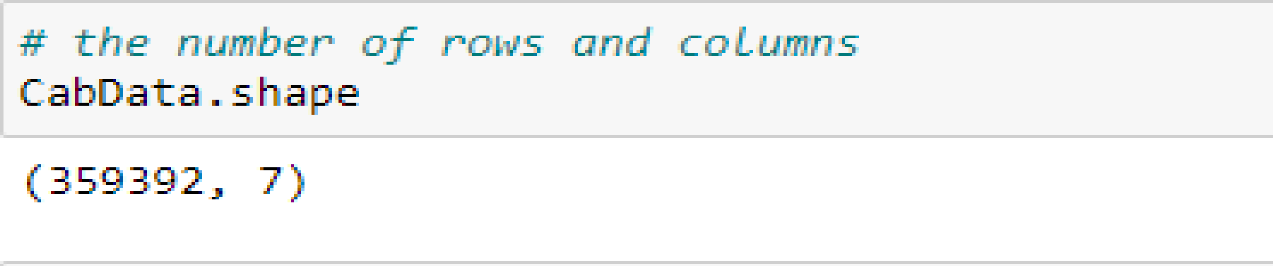
1. Understanding and Cleaning the data
2. Analysis
3. Visualization

EDA >> Understanding and Cleaning the data



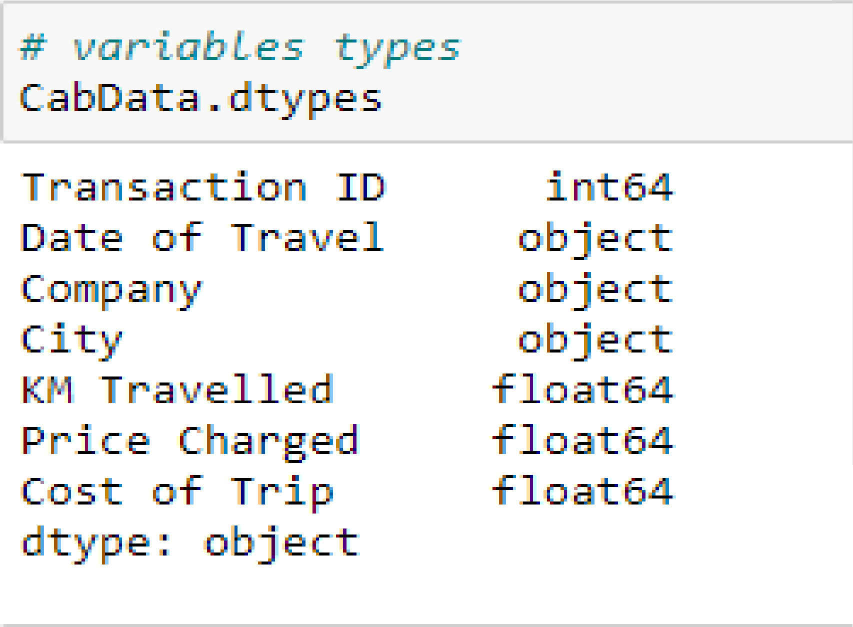
```
# get the variables name the dataset
CabData.columns
```

```
Index(['Transaction ID', 'Date of Travel', 'Company', 'City', 'KM Travelled',
       'Price Charged', 'Cost of Trip'],
      dtype='object')
```



```
# the number of rows and columns
CabData.shape
```

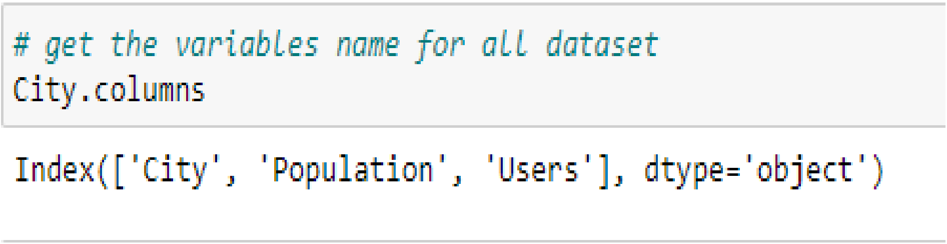
```
(359392, 7)
```



```
# variables types
CabData.dtypes
```

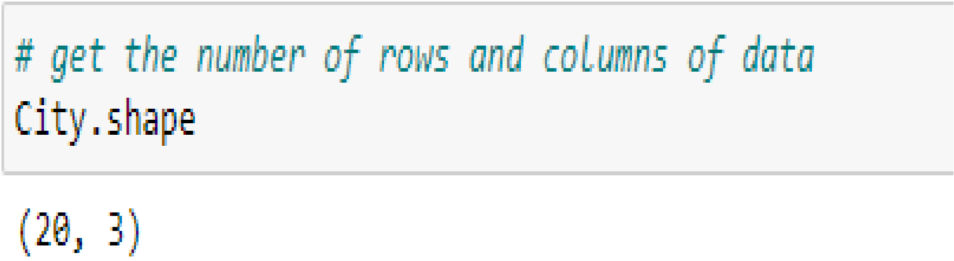
```
Transaction ID       int64
Date of Travel      object
Company             object
City                object
KM Travelled       float64
Price Charged      float64
Cost of Trip       float64
dtype: object
```

EDA >> Understanding and Cleaning the data



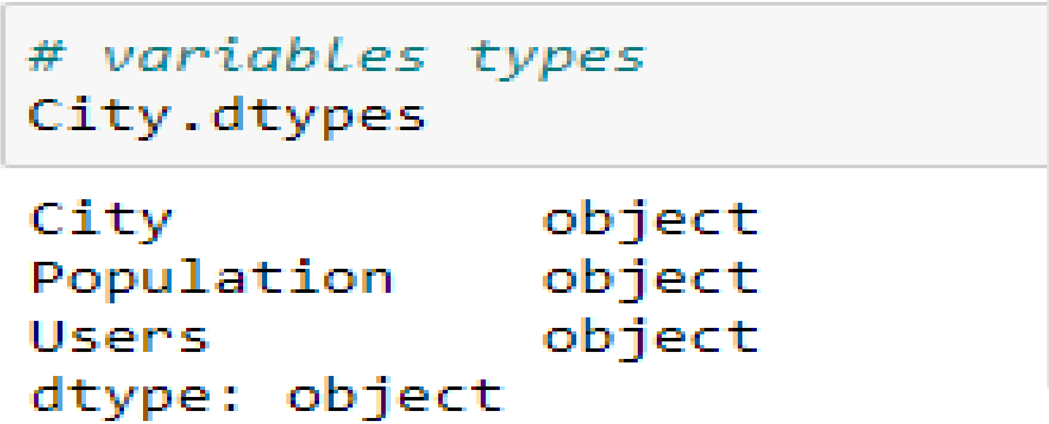
```
# get the variables name for all dataset
City.columns
```

```
Index(['City', 'Population', 'Users'], dtype='object')
```



```
# get the number of rows and columns of data
City.shape
```

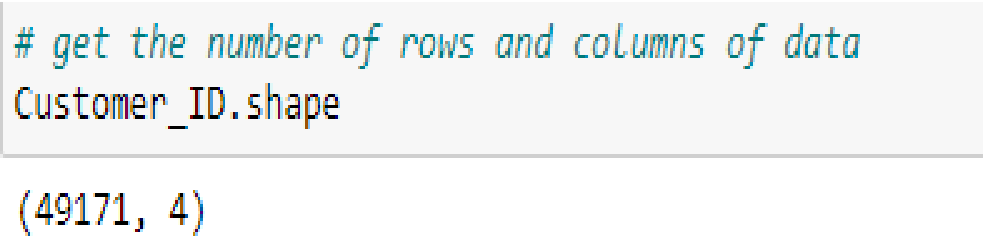
```
(20, 3)
```



```
# variables types
City.dtypes
```

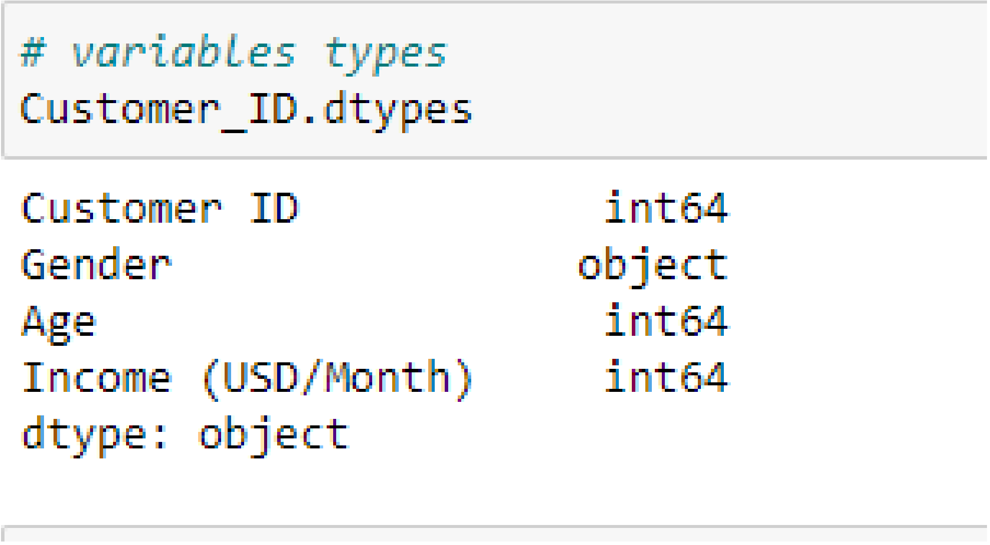
```
City          object
Population    object
Users         object
dtype: object
```

EDA >> Understanding and Cleaning the data



```
# get the number of rows and columns of data
Customer_ID.shape
```

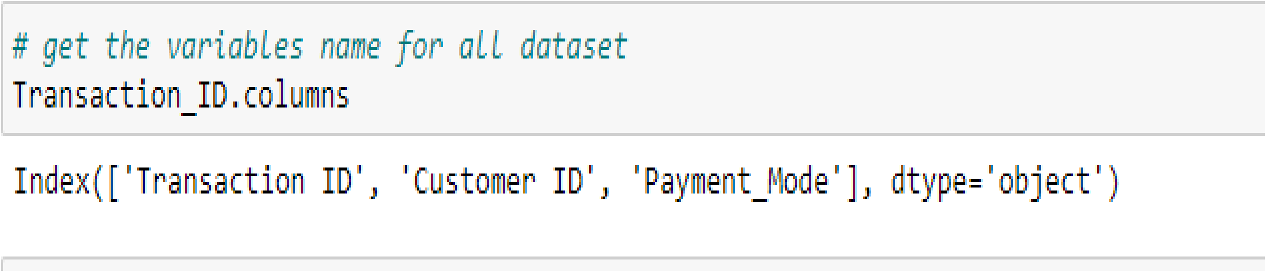
```
(49171, 4)
```



```
# variables types
Customer_ID.dtypes
```

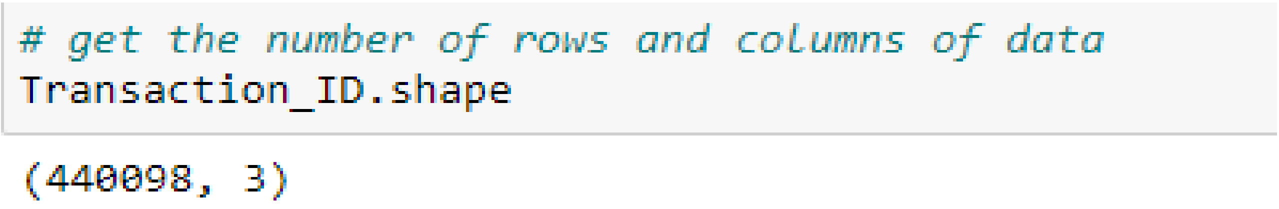
```
Customer ID               int64
Gender                   object
Age                       int64
Income (USD/Month)        int64
dtype: object
```

EDA >> Understanding and Cleaning the data



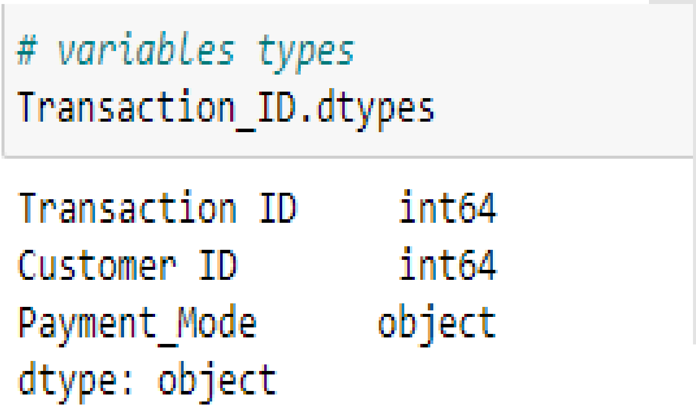
```
# get the variables name for all dataset
Transaction_ID.columns
```

```
Index(['Transaction ID', 'Customer ID', 'Payment_Mode'], dtype='object')
```



```
# get the number of rows and columns of data
Transaction_ID.shape
```

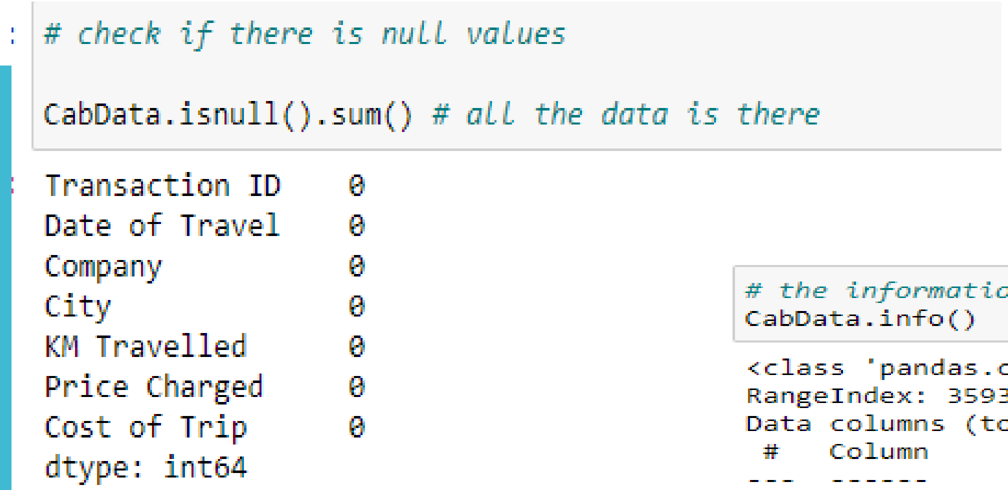
```
(440098, 3)
```



```
# variables types
Transaction_ID.dtypes
```

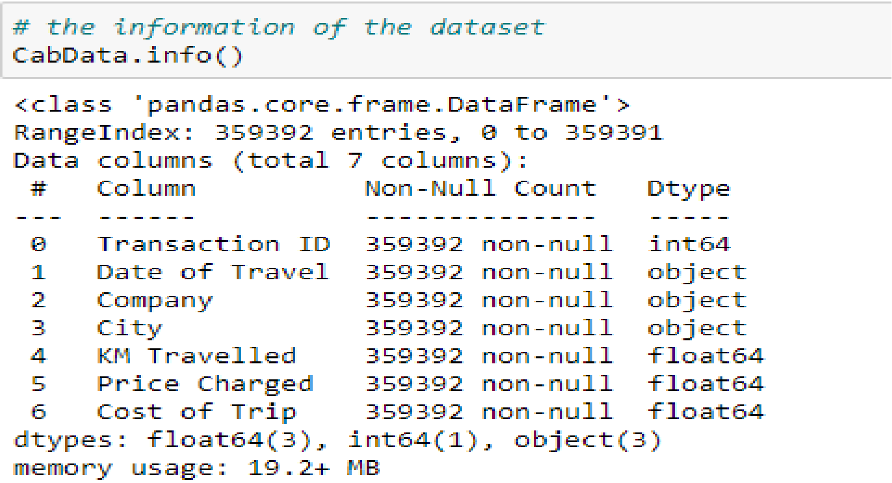
```
Transaction ID      int64
Customer ID         int64
Payment_Mode       object
dtype: object
```

EDA >> Understanding and Cleaning the data



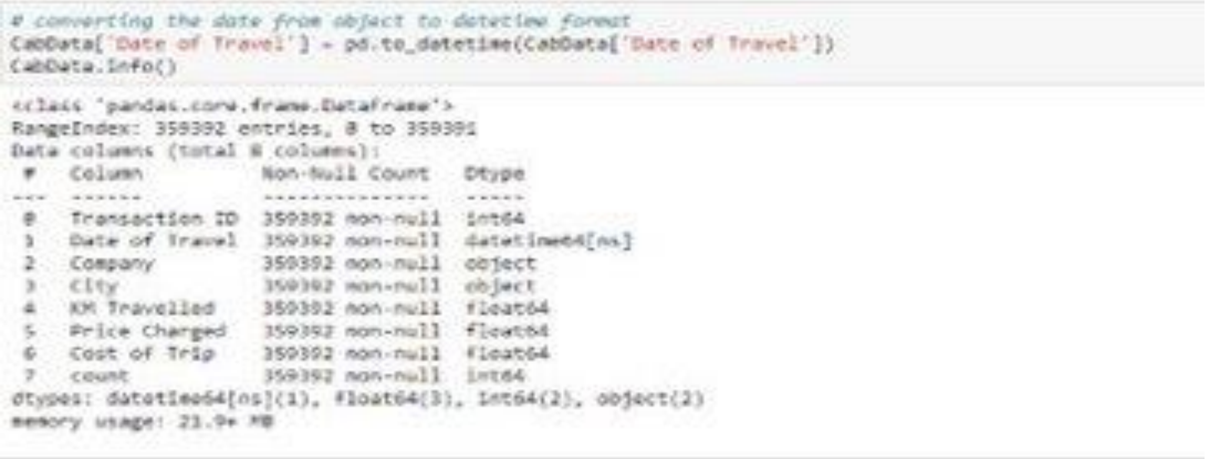
```
# check if there is null values

CabData.isnull().sum() # all the data is there
```

```
Transaction ID    0
Date of Travel    0
Company           0
City              0
KM Travelled      0
Price Charged     0
Cost of Trip      0
dtype: int64
```



```
# the information of the dataset
CabData.info()
```

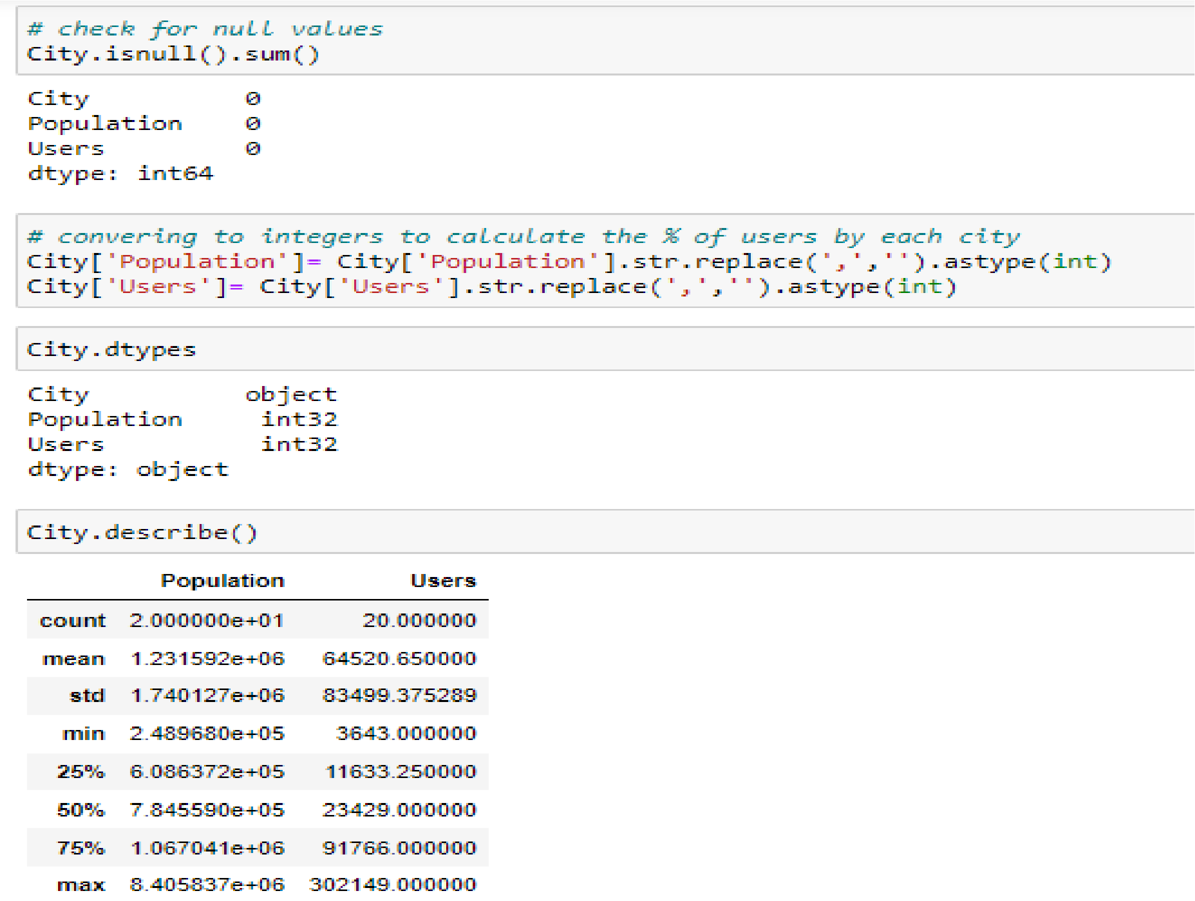
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 359392 entries, 0 to 359391
Data columns (total 7 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   Transaction ID  359392 non-null  int64  
 1   Date of Travel  359392 non-null  object 
 2   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(1), object(3)
memory usage: 19.2+ MB
```



```
# converting the date from object to datetime format
CabData['Date of Travel'] = pd.to_datetime(CabData['Date of Travel'])
CabData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 359392 entries, 0 to 359391
Data columns (total 8 columns):
 #   Column          Non-Null Count   Dtype         
---  ------          --------------   -----         
 0   Transaction ID  359392 non-null  int64         
 1   Date of Travel  359392 non-null  datetime64[ns]
 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       
 7   count           359392 non-null  int64         
dtypes: datetime64[ns](1), float64(3), int64(2), object(2)
memory usage: 21.9+ MB
```

EDA >> Understanding and Cleaning the data



```
# check for null values
City.isnull().sum()
```

```
City          0
Population    0
Users         0
dtype: int64
```

```
# convering to integers to calculate the % of users by each city
City['Population']= City['Population'].str.replace(',','').astype(int)
City['Users']= City['Users'].str.replace(',','').astype(int)
```

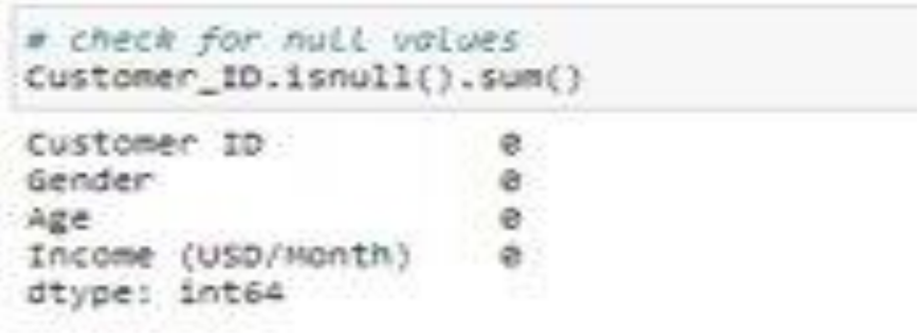
```
City.dtypes
```

```
City          object
Population     int32
Users          int32
dtype: object
```

```
City.describe()
```

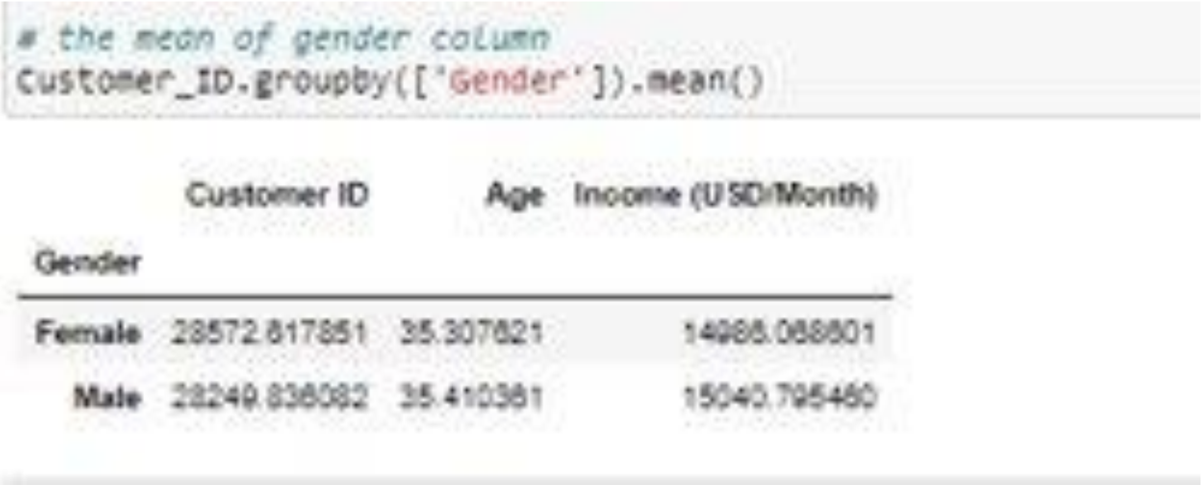
	Population	Users
count	2.000000e+01	20.000000
mean	1.231592e+06	64520.650000
std	1.740127e+06	83499.375289
min	2.489680e+05	3643.000000
25%	6.086372e+05	11633.250000
50%	7.845590e+05	23429.000000
75%	1.067041e+06	91766.000000
max	8.405837e+06	302149.000000

EDA >> Understanding and Cleaning the data



```
# check for null values
Customer_ID.isnull().sum()
```

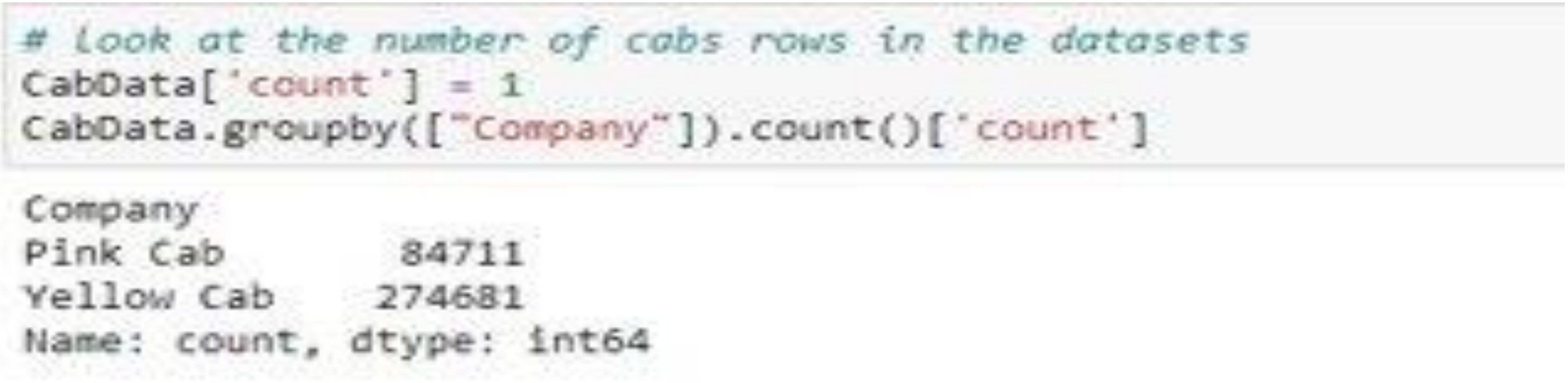
```
Customer ID              0
Gender                   0
Age                      0
Income (USD/Month)       0
dtype: int64
```



```
# the mean of gender column
Customer_ID.groupby(['Gender']).mean()
```

	Customer ID	Age	Income (USD/Month)
Gender			
Female	28572.817851	35.307821	14986.068801
Male	28249.836082	35.410361	15040.795460

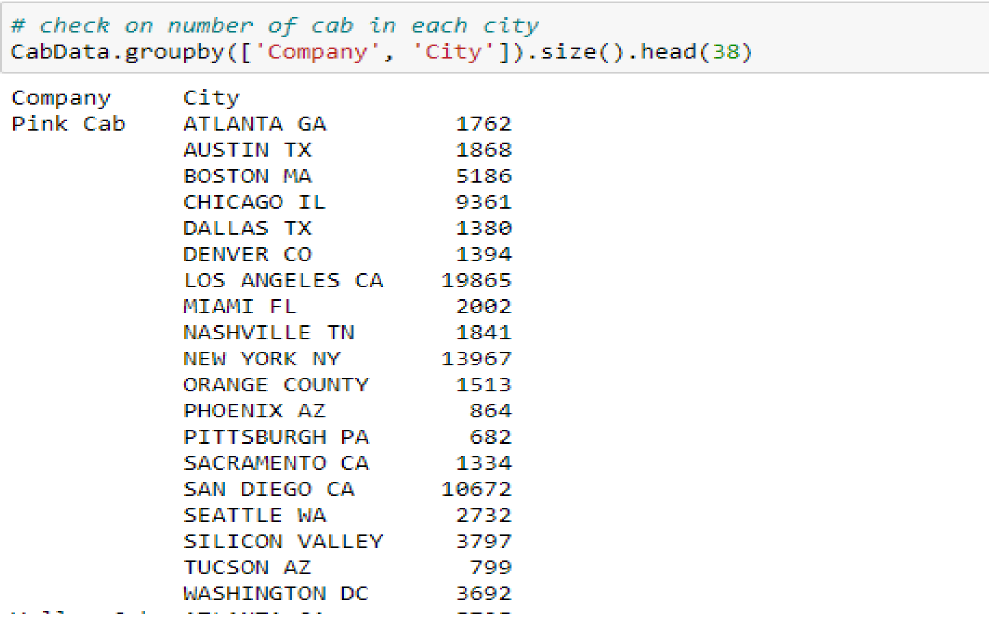
EDA >> Analysis



```
# look at the number of cabs rows in the datasets
CabData['count'] = 1
CabData.groupby(["Company"]).count()['count']
```

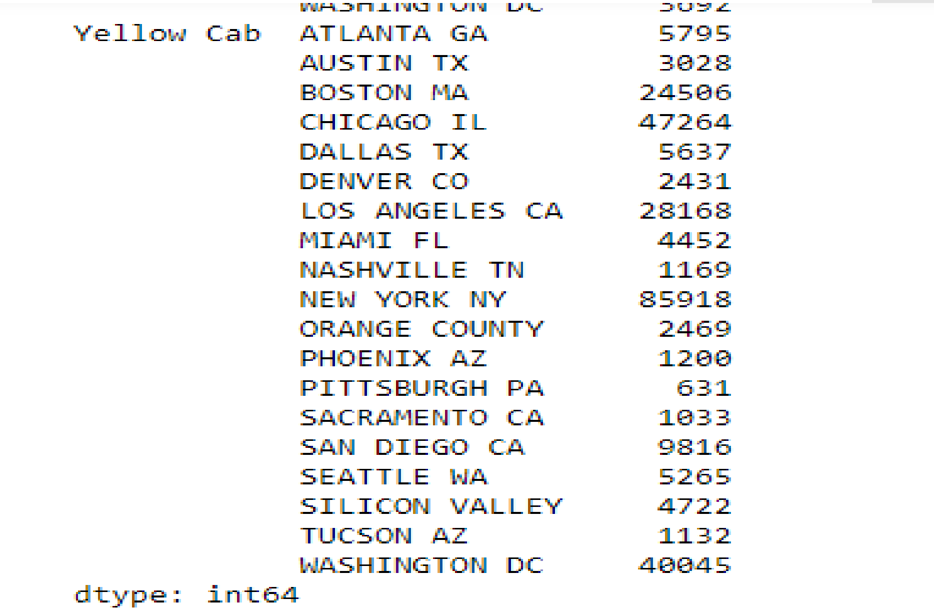
```
Company
Pink Cab        84711
Yellow Cab     274681
Name: count, dtype: int64
```

EDA >> Analysis



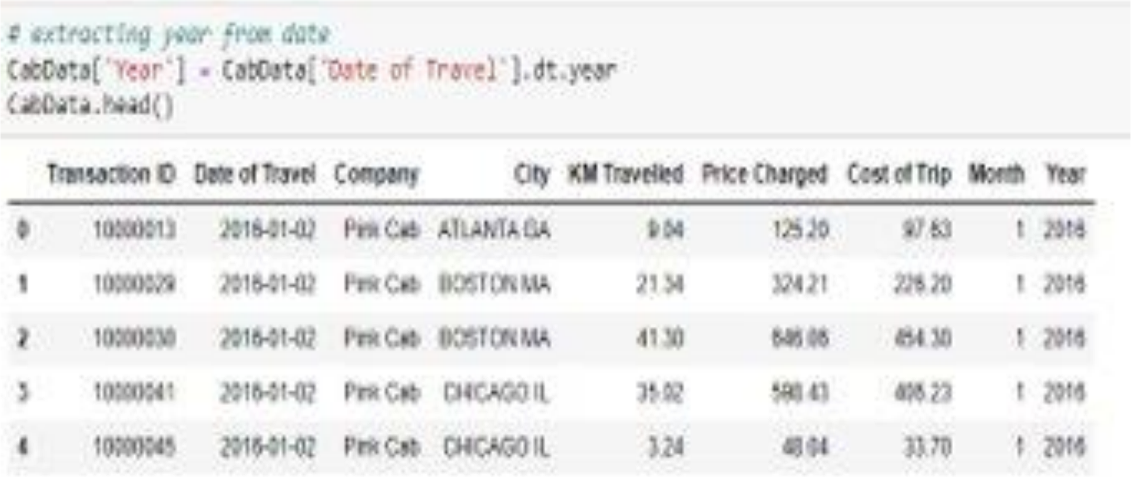
```
# check on number of cab in each city
CabData.groupby(['Company', 'City']).size().head(38)
```

```
Company     City
Pink Cab    ATLANTA GA         1762
            AUSTIN TX          1868
            BOSTON MA          5186
            CHICAGO IL         9361
            DALLAS TX          1380
            DENVER CO          1394
            LOS ANGELES CA    19865
            MIAMI FL           2002
            NASHVILLE TN       1841
            NEW YORK NY       13967
            ORANGE COUNTY      1513
            PHOENIX AZ          864
            PITTSBURGH PA       682
            SACRAMENTO CA      1334
            SAN DIEGO CA      10672
            SEATTLE WA         2732
            SILICON VALLEY     3797
            TUCSON AZ           799
            WASHINGTON DC      3692
```



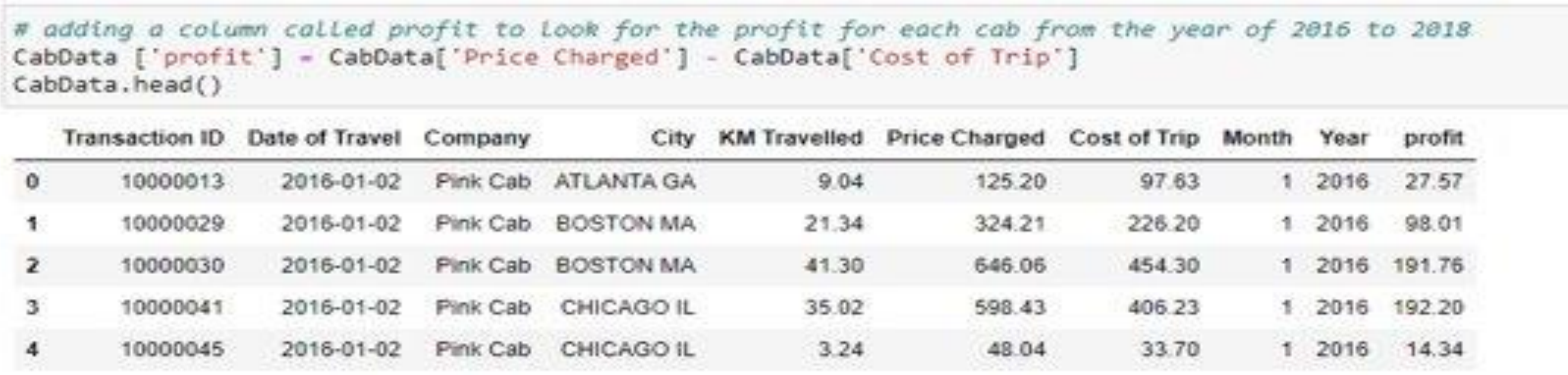
```
Yellow Cab  ATLANTA GA         5795
            AUSTIN TX          3028
            BOSTON MA         24506
            CHICAGO IL        47264
            DALLAS TX          5637
            DENVER CO          2431
            LOS ANGELES CA    28168
            MIAMI FL           4452
            NASHVILLE TN       1169
            NEW YORK NY       85918
            ORANGE COUNTY      2469
            PHOENIX AZ         1200
            PITTSBURGH PA       631
            SACRAMENTO CA      1033
            SAN DIEGO CA       9816
            SEATTLE WA         5265
            SILICON VALLEY     4722
            TUCSON AZ          1132
            WASHINGTON DC     40045
dtype: int64
```

EDA >> Analysis



```
# extracting year from date
CabData['Year'] = CabData['Date of Travel'].dt.year
CabData.head()
```

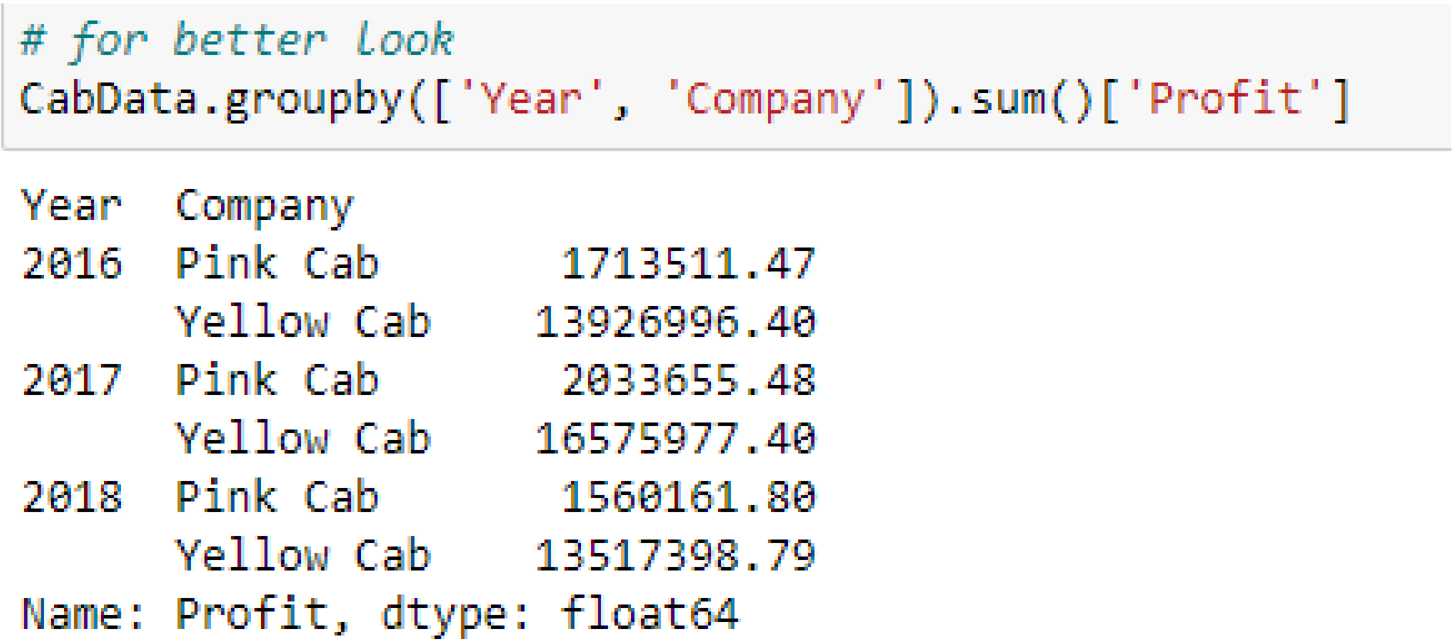
	Transaction ID	Date of Travel	Company	City	KM Traveled	Price Charged	Cost of Trip	Month	Year
0	10000013	2016-01-02	Pink Cab	ATLANTA GA	9.04	125.20	97.63	1	2016
1	10000029	2016-01-02	Pink Cab	BOSTON MA	21.34	324.21	226.20	1	2016
2	10000030	2016-01-02	Pink Cab	BOSTON MA	41.30	646.06	454.30	1	2016
3	10000041	2016-01-02	Pink Cab	CHICAGO IL	35.02	598.43	406.23	1	2016
4	10000045	2016-01-02	Pink Cab	CHICAGO IL	3.24	48.04	33.70	1	2016



```
# adding a column called profit to look for the profit for each cab from the year of 2016 to 2018
CabData ['profit'] = CabData['Price Charged'] - CabData['Cost of Trip']
CabData.head()
```

	Transaction ID	Date of Travel	Company	City	KM Travelled	Price Charged	Cost of Trip	Month	Year	profit
0	10000013	2016-01-02	Pink Cab	ATLANTA GA	9.04	125.20	97.63	1	2016	27.57
1	10000029	2016-01-02	Pink Cab	BOSTON MA	21.34	324.21	226.20	1	2016	98.01
2	10000030	2016-01-02	Pink Cab	BOSTON MA	41.30	646.06	454.30	1	2016	191.76
3	10000041	2016-01-02	Pink Cab	CHICAGO IL	35.02	598.43	406.23	1	2016	192.20
4	10000045	2016-01-02	Pink Cab	CHICAGO IL	3.24	48.04	33.70	1	2016	14.34

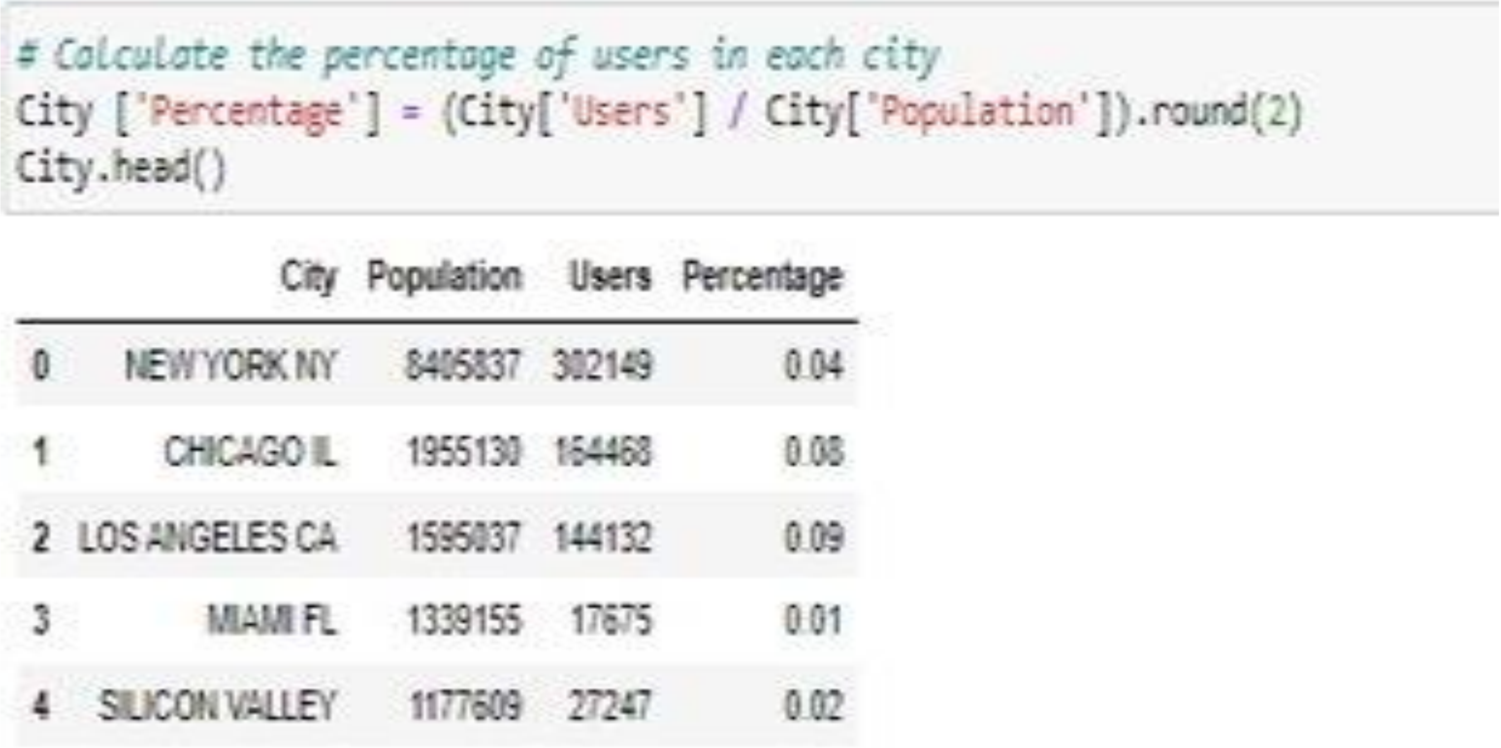
EDA >> Analysis



```python
# for better look
CabData.groupby(['Year', 'Company']).sum()['Profit']
```

```
Year  Company   
2016  Pink Cab       1713511.47
      Yellow Cab    13926996.40
2017  Pink Cab       2033655.48
      Yellow Cab    16575977.40
2018  Pink Cab       1560161.80
      Yellow Cab    13517398.79
Name: Profit, dtype: float64
```

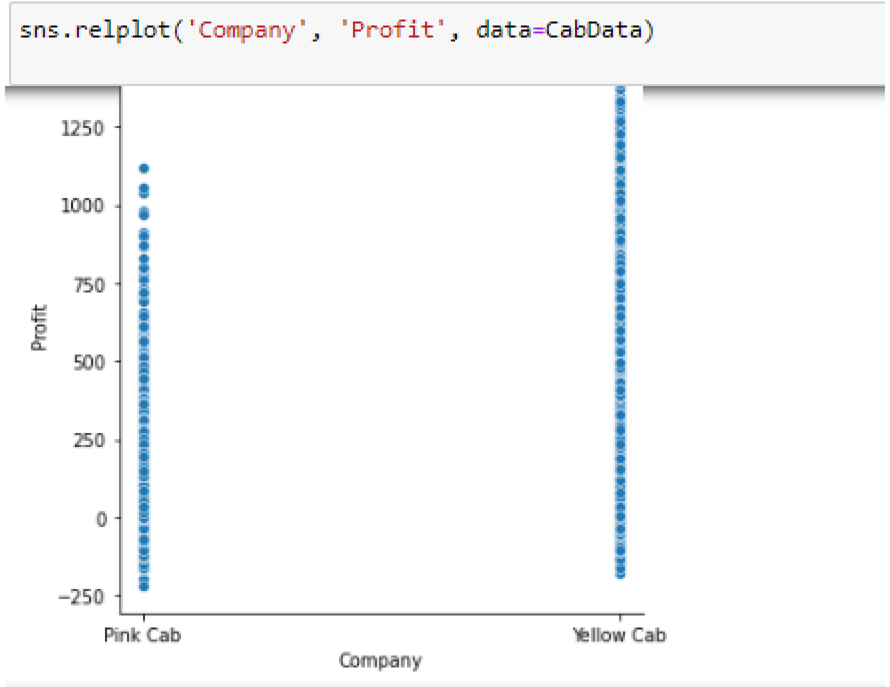
EDA >> Analysis



```
# Calculate the percentage of users in each city
City ['Percentage'] = (City['Users'] / City['Population']).round(2)
City.head()
```

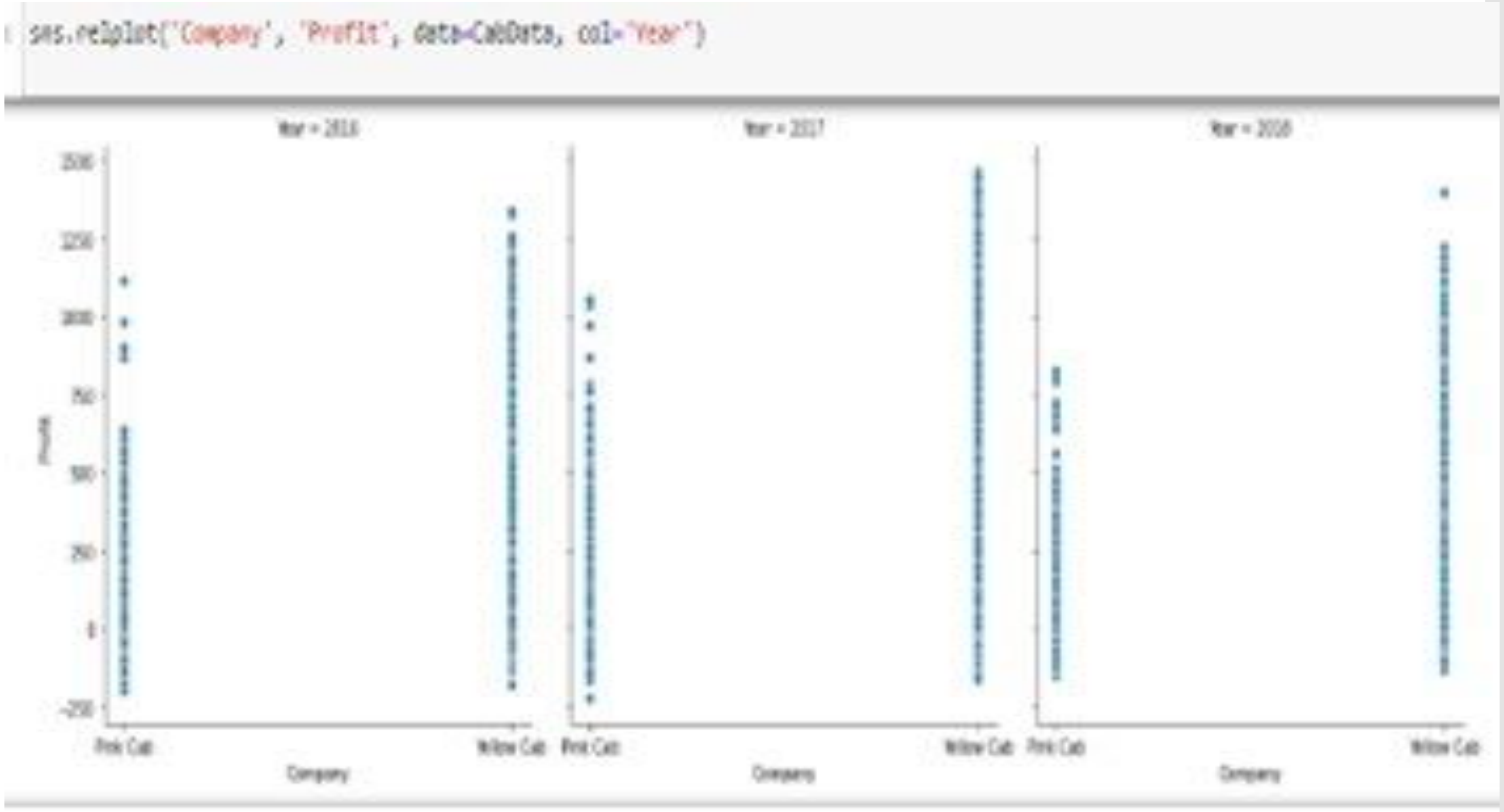
	City	Population	Users	Percentage
0	NEW YORK NY	8405837	302149	0.04
1	CHICAGO IL	1955130	164468	0.08
2	LOS ANGELES CA	1595037	144132	0.09
3	MIAMI FL	1339155	17675	0.01
4	SILICON VALLEY	1177609	27247	0.02

EDA >> Visualization



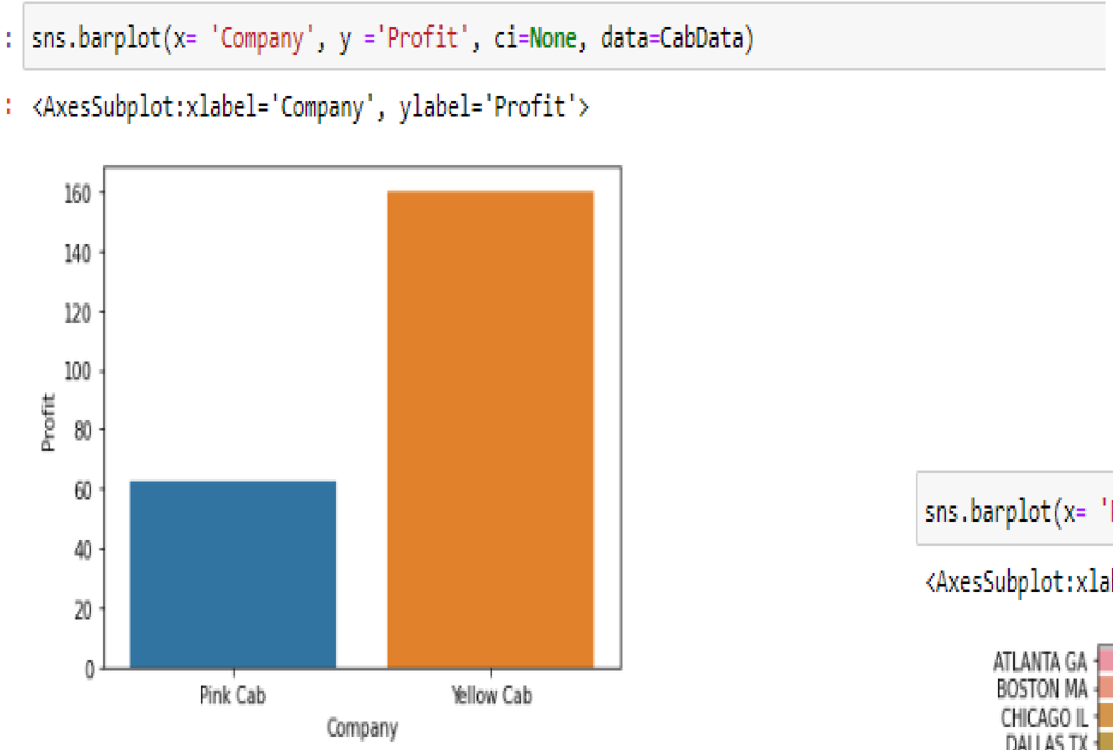
```
sns.relplot('Company', 'Profit', data=CabData)
```

EDA >> Visualization



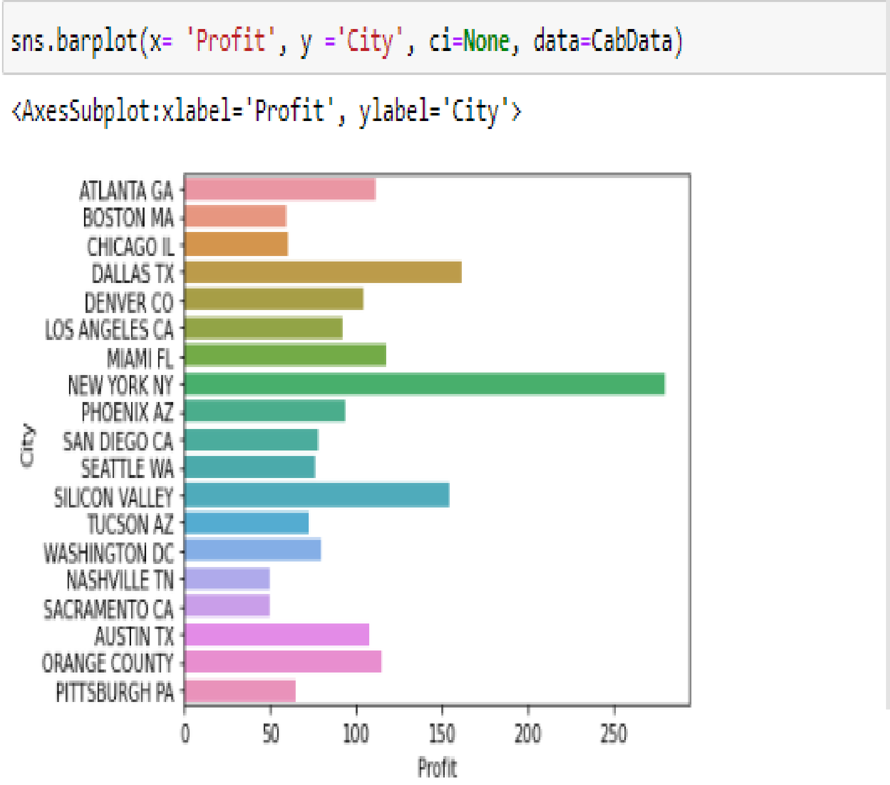
```
sns.relplot('Company', 'Profit', data=CabData, col='Year')
```

EDA >> Visualization



```
: sns.barplot(x= 'Company', y ='Profit', ci=None, data=CabData)
```

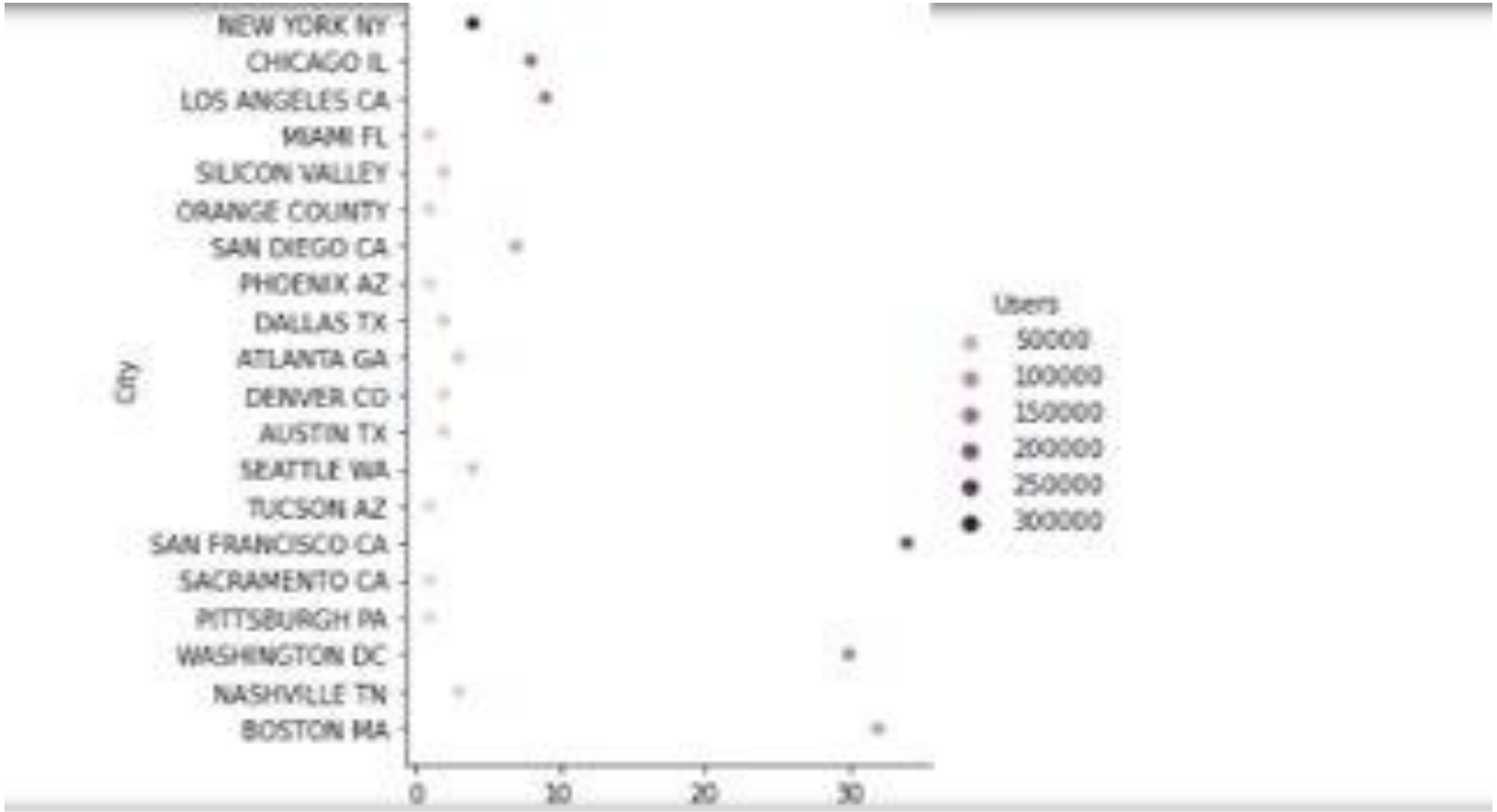
```
: <AxesSubplot:xlabel='Company', ylabel='Profit'>
```



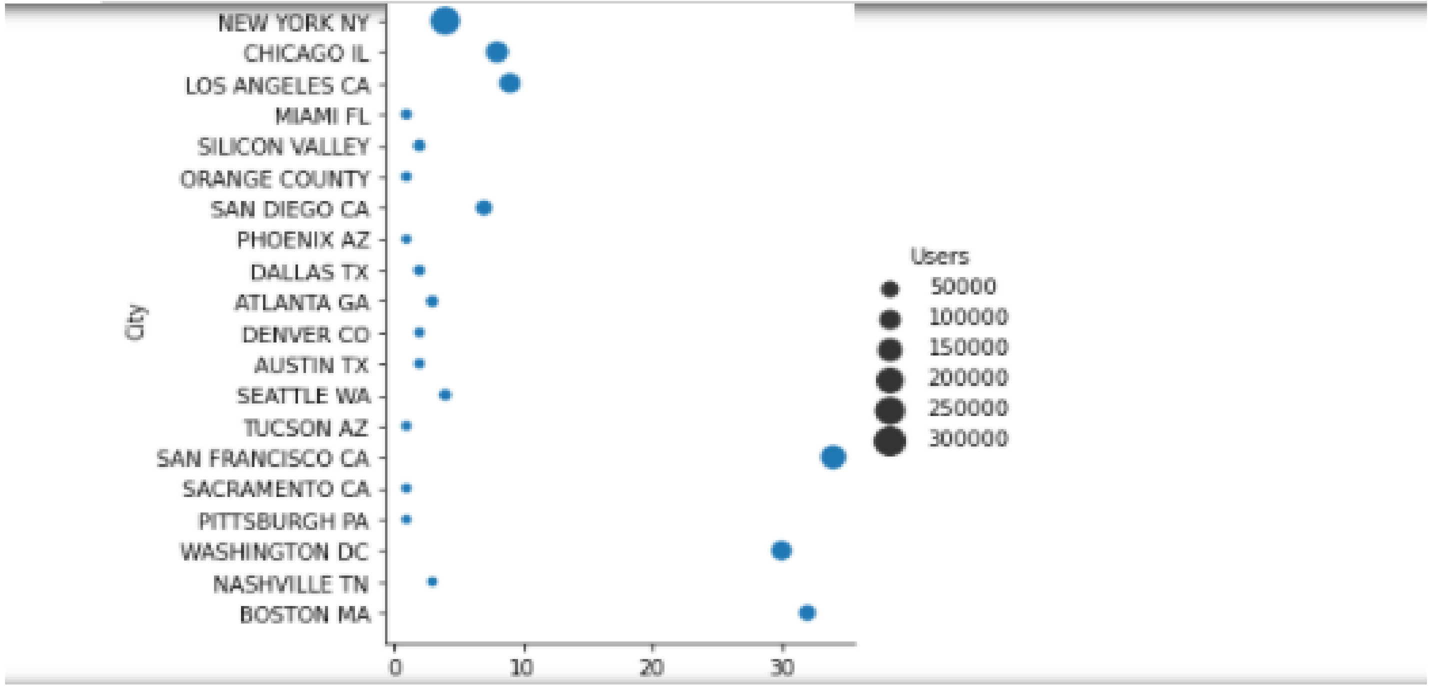
```
sns.barplot(x= 'Profit', y ='City', ci=None, data=CabData)
```

```
<AxesSubplot:xlabel='Profit', ylabel='City'>
```

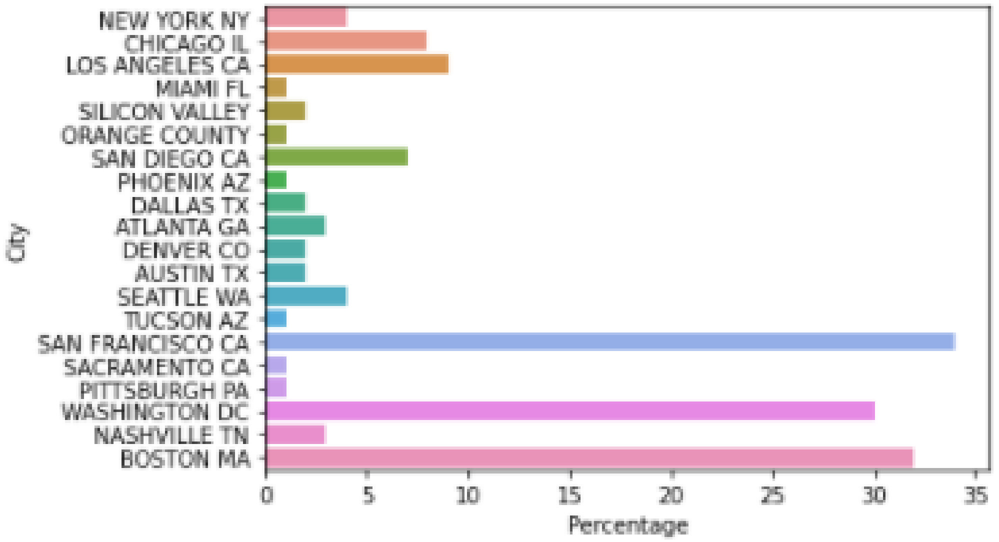
EDA >> Visualization



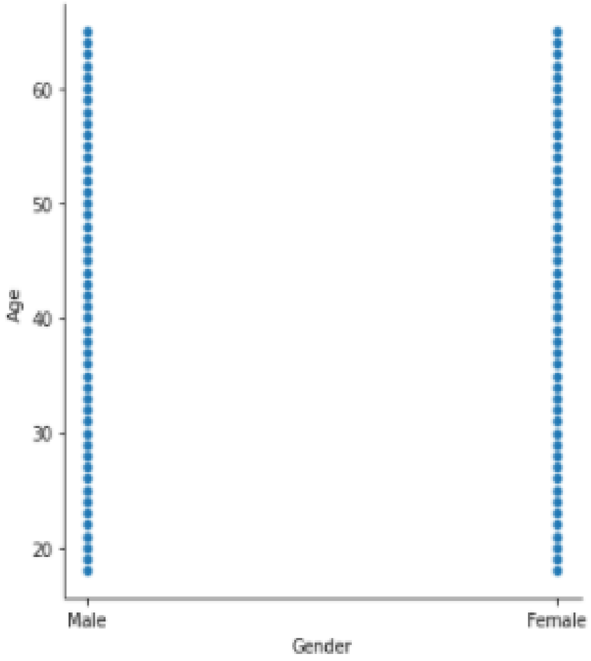
EDA >> Visualization

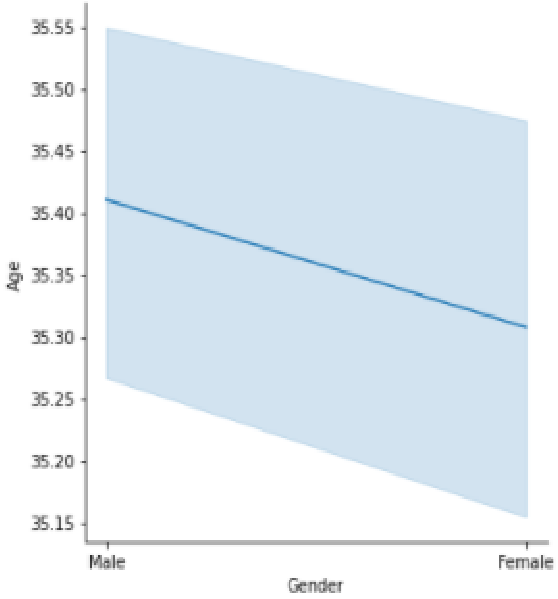


EDA >> Visualization

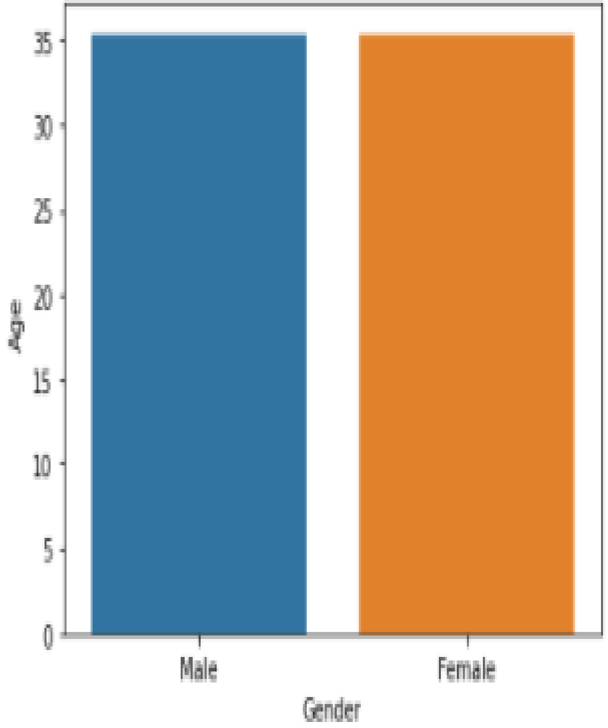


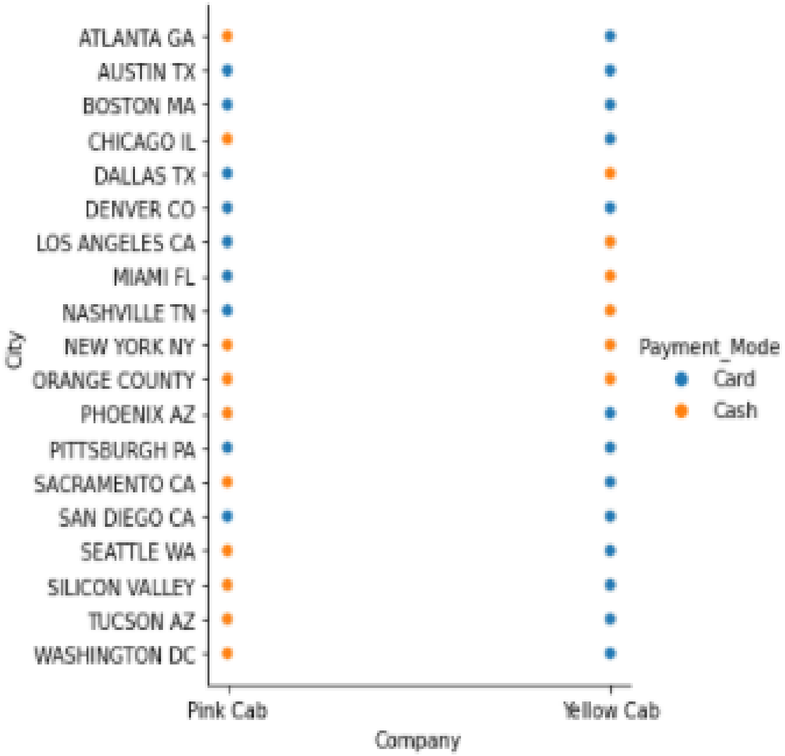
EDA >> Visualization





EDA >> Visualization





EDA Summary Pink or Yellow Cabs ??

- Yellow Cab profits from the years of 2016 to 2018 was higher than Pink Cab.
- Number of customers who are using Yellow Cab is more than the Pink Cab.

Recommendations

- According to the data that has been provided from the years of 2016 to 2018, investing in Yellow Cab would be the suitable choice for the XYZ Company.

Thank You !