# The G2B Cab Investement Company. The goal of every organization is to make profits and to see trends that will help maximize the resources invested. In a bid to help the Executive make ground-breaking profits decision, I will be analyzing these data to bring out insights for the organization.

In [51]:

```
#Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)
```

In [2]:

```
#Load the data set
cab_data = pd.read_csv('Cab_Data.csv')
city = pd.read_csv('City.csv')
Customer_ID = pd.read_csv('Customer_ID.csv')
Transaction = pd.read_csv('Transaction_ID.csv')
```

In [3]:

```
cab_data
```

Out[3]:

| | Transaction ID | Date of Travel | Company | City | KM Travelled | Price Charged | Cost of Trip |
|---|---|---|---|---|---|---|---|
| **0** | 10000011 | 08/01/2016 | Pink Cab | ATLANTA GA | 30.45 | 370.95 | 313.6350 |
| **1** | 10000012 | 06/01/2016 | Pink Cab | ATLANTA GA | 28.62 | 358.52 | 334.8540 |
| **2** | 10000013 | 02/01/2016 | Pink Cab | ATLANTA GA | 9.04 | 125.20 | 97.6320 |
| **3** | 10000014 | 07/01/2016 | Pink Cab | ATLANTA GA | 33.17 | 377.40 | 351.6020 |
| **4** | 10000015 | 03/01/2016 | Pink Cab | ATLANTA GA | 8.73 | 114.62 | 97.7760 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **359387** | 10440101 | 08/01/2018 | Yellow Cab | WASHINGTON DC | 4.80 | 69.24 | 63.3600 |
| **359388** | 10440104 | 04/01/2018 | Yellow Cab | WASHINGTON DC | 8.40 | 113.75 | 106.8480 |

In [4]:

```
city
```

Out[4]:

| | 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 |
| **5** | ORANGE COUNTY | 1,030,185 | 12,994 |
| **6** | SAN DIEGO CA | 959,307 | 69,995 |
| **7** | PHOENIX AZ | 943,999 | 6,133 |
| **8** | DALLAS TX | 942,908 | 22,157 |
| **9** | ATLANTA GA | 814,885 | 24,701 |
| **10** | DENVER CO | 754,233 | 12,421 |
| **11** | AUSTIN TX | 698,371 | 14,978 |
| **12** | SEATTLE WA | 671,238 | 25,063 |
| **13** | TUCSON AZ | 631,442 | 5,712 |
| **14** | SAN FRANCISCO CA | 629,591 | 213,609 |
| **15** | SACRAMENTO CA | 545,776 | 7,044 |
| **16** | PITTSBURGH PA | 542,085 | 3,643 |
| **17** | WASHINGTON DC | 418,859 | 127,001 |
| **18** | NASHVILLE TN | 327,225 | 9,270 |
| **19** | BOSTON MA | 248,968 | 80,021 |

In [5]:

```
Transaction
```

| | 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 |
| **...** | ... | ... | ... |
| **440093** | 10440104 | 53286 | Cash |
| **440094** | 10440105 | 52265 | Cash |
| **440095** | 10440106 | 52175 | Card |
| **440096** | 10440107 | 52917 | Card |
| **440097** | 10440108 | 51587 | Card |

In [6]:

```
Customer_ID
```

Out[6]:

| | 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 |
| **...** | ... | ... | ... | ... |
| **49166** | 12490 | Male | 33 | 18713 |
| **49167** | 14971 | Male | 30 | 15346 |
| **49168** | 41414 | Male | 38 | 3960 |
| **49169** | 41677 | Male | 23 | 19454 |
| **49170** | 39761 | Female | 32 | 10128 |

49171 rows × 4 columns

In [7]:

```
#Check for data duplicates, empty cells in the four datasets
```

In [8]:

```
cab_data.isnull().sum()
```

Out[8]:

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

In [9]:

```
cab_data.duplicated().sum()
```

Out[9]:

```
0
```

In [10]:

```
city.isnull().sum()
```

Out[10]:

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

In [11]:

```
city.duplicated().sum()
```

Out[11]:

```
0
```

In [12]:

```
Transaction.duplicated().sum()
```

Out[12]:

```
0
```

In [13]:

```
Transaction.isnull().sum()
```

Out[13]:

```
Transaction ID    0
Customer ID       0
Payment_Mode      0
dtype: int64
```

In [14]:

```
Customer_ID.duplicated().sum()
```

Out[14]:

```
0
```

In [15]:

```
Customer_ID.isnull().sum()
```

Out[15]:

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

In [16]:

```
cab_data.info
```

Out[16]:

```
<bound method DataFrame.info of         Transaction ID Date of Travel     Co
mpany           City  \
0             10000011     08/01/2016    Pink Cab     ATLANTA GA
1             10000012     06/01/2016    Pink Cab     ATLANTA GA
2             10000013     02/01/2016    Pink Cab     ATLANTA GA
3             10000014     07/01/2016    Pink Cab     ATLANTA GA
4             10000015     03/01/2016    Pink Cab     ATLANTA GA
...                ...            ...         ...            ...
359387        10440101     08/01/2018  Yellow Cab  WASHINGTON DC
359388        10440104     04/01/2018  Yellow Cab  WASHINGTON DC
359389        10440105     05/01/2018  Yellow Cab  WASHINGTON DC
359390        10440106     05/01/2018  Yellow Cab  WASHINGTON DC
359391        10440107     02/01/2018  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]>
```

In [17]:

```
Transaction.info
```

Out[17]:

```
<bound method DataFrame.info of         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
...                ...          ...        ...
440093        10440104        53286       Cash
440094        10440105        52265       Cash
440095        10440106        52175       Card
440096        10440107        52917       Card
440097        10440108        51587       Card

[440098 rows x 3 columns]>
```

In [18]:

```
city.info
```

Out[18]:

```
<bound method DataFrame.info of                 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
5      ORANGE COUNTY  1,030,185     12,994
6       SAN DIEGO CA    959,307     69,995
7         PHOENIX AZ    943,999      6,133
8          DALLAS TX    942,908     22,157
9         ATLANTA GA    814,885     24,701
10         DENVER CO    754,233     12,421
11         AUSTIN TX    698,371     14,978
12        SEATTLE WA    671,238     25,063
13         TUCSON AZ    631,442      5,712
14  SAN FRANCISCO CA    629,591    213,609
15     SACRAMENTO CA    545,776      7,044
16     PITTSBURGH PA    542,085      3,643
17     WASHINGTON DC    418,859    127,001
18      NASHVILLE TN    327,225      9,270
19         BOSTON MA    248,968     80,021 >
```

Data Preparation and Cleaning in order to merge the dataset into one comprehensive dataset

In [19]:

```
#Count the number of unique cities
counts=city.nunique()
```

In [20]:

```
counts
```

Out[20]:

```
City          20
Population    20
Users         20
dtype: int64
```

In [21]:

```
count=cab_data.City.nunique()
```

In [22]:

```
count
```

Out[22]:

```
19
```

In [23]:

```
#Since there are different number in cities, Check to see which city in the city dataset is
np.setdiff1d(city.City, cab_data.City)
```

Out[23]:

```
array(['SAN FRANCISCO CA'], dtype=object)
```

In [24]:

```
#San Francisco is not included in the cab-dataset,
```

In [25]:

```
#Combine the transaction ID in transaction with the transaction Id in cab data

len(np.setdiff1d(Transaction["Transaction ID"], cab_data["Transaction ID"]))
```

Out[25]:

```
80706
```

There are 80706 transactions not listed in the cab data

In [26]:

```
#1st merge the cab_data and transaction data

merge_cab_transaction = pd.merge(cab_data, Transaction, on = 'Transaction ID')
```

In [27]:

```
#Check and compare the Customer ID in Transaction for the ones not recorded in the merge_ca

len(np.setdiff1d(Customer_ID['Customer ID'], merge_cab_transaction['Customer ID']))
```

Out[27]:

```
3023
```

In [28]:

```
#Since there are 3023 not included in the Customer ID, we will need to drop them to make a
df = pd.merge(merge_cab_transaction, Customer_ID, on = 'Customer ID')
df
```

Out[28]:

| | Transaction ID | Date of Travel | Company | City | KM Travelled | Price Charged | Cost of Trip | Custom |
|---|---|---|---|---|---|---|---|---|
| **0** | 10000011 | 08/01/2016 | Pink Cab | ATLANTA GA | 30.45 | 370.95 | 313.6350 | 292 |
| **1** | 10351127 | 21/07/2018 | Yellow Cab | ATLANTA GA | 26.19 | 598.70 | 317.4228 | 292 |
| **2** | 10412921 | 23/11/2018 | Yellow Cab | ATLANTA GA | 42.55 | 792.05 | 597.4020 | 292 |
| **3** | 10000012 | 06/01/2016 | Pink Cab | ATLANTA GA | 28.62 | 358.52 | 334.8540 | 277 |
| **4** | 10320494 | 21/04/2018 | Yellow Cab | ATLANTA GA | 36.38 | 721.10 | 467.1192 | 277 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **359387** | 10439790 | 07/01/2018 | Yellow Cab | SEATTLE WA | 16.66 | 261.18 | 213.9144 | 385 |
| **359388** | 10439799 | 03/01/2018 | Yellow Cab | SILICON VALLEY | 13.72 | 277.97 | 172.8720 | 124 |
| **359389** | 10439838 | 04/01/2018 | Yellow Cab | TUCSON AZ | 19.00 | 303.77 | 232.5600 | 414 |
| **359390** | 10439840 | 06/01/2018 | Yellow Cab | TUCSON AZ | 5.60 | 92.42 | 70.5600 | 416 |
| **359391** | 10439846 | 04/01/2018 | Yellow Cab | TUCSON AZ | 13.30 | 244.65 | 180.3480 | 397 |

359392 rows × 12 columns

Now that we have a comprehensive dataset, we begin EDA

In [29]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 359392 entries, 0 to 359391
Data columns (total 12 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
 7   Customer ID         359392 non-null  int64  
 8   Payment_Mode        359392 non-null  object 
 9   Gender              359392 non-null  object 
 10  Age                 359392 non-null  int64  
 11  Income (USD/Month)  359392 non-null  int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 35.6+ MB
```

In [30]:

```
df.describe()
```

Out[30]:

| | Transaction ID | KM Travelled | Price Charged | Cost of Trip | Customer ID | Ag |
|---|---|---|---|---|---|---|
| **count** | 3.593920e+05 | 359392.000000 | 359392.000000 | 359392.000000 | 359392.000000 | 359392.0000 |
| **mean** | 1.022076e+07 | 22.567254 | 423.443311 | 286.190113 | 19191.652115 | 35.3367 |
| **std** | 1.268058e+05 | 12.233526 | 274.378911 | 157.993661 | 21012.412463 | 12.5942 |
| **min** | 1.000001e+07 | 1.900000 | 15.600000 | 19.000000 | 1.000000 | 18.0000 |
| **25%** | 1.011081e+07 | 12.000000 | 206.437500 | 151.200000 | 2705.000000 | 25.0000 |
| **50%** | 1.022104e+07 | 22.440000 | 386.360000 | 282.480000 | 7459.000000 | 33.0000 |
| **75%** | 1.033094e+07 | 32.960000 | 583.660000 | 413.683200 | 36078.000000 | 42.0000 |
| **max** | 1.044011e+07 | 48.000000 | 2048.030000 | 691.200000 | 60000.000000 | 65.0000 |

In [31]:

```
df.isnull().sum()
```

Out[31]:

```
Transaction ID        0
Date of Travel        0
Company               0
City                  0
KM Travelled          0
Price Charged         0
Cost of Trip          0
Customer ID           0
Payment_Mode          0
Gender                0
Age                   0
Income (USD/Month)    0
dtype: int64
```

In [32]:

```
from pandas_profiling import ProfileReport
```

In [35]:

```
df.City.unique()
```

Out[35]:

```
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)
```

In [37]:

```
#From the Data set, the dates are not arranged in any order, hence we rearrange according t

df.sort_values(['Date of Travel', 'Transaction ID'], ignore_index=True, inplace = True)
```

In [38]:

```
df
```

Out[38]:

| | Transaction ID | Date of Travel | Company | City | KM Travelled | Price Charged | Cost of Trip | Custo |
|---|---|---|---|---|---|---|---|---|
| **0** | 10128150 | 01/01/2017 | Pink Cab | ATLANTA GA | 24.00 | 379.79 | 280.8000 | 28 |
| **1** | 10128153 | 01/01/2017 | Pink Cab | ATLANTA GA | 38.88 | 609.62 | 443.2320 | 29 |
| **2** | 10128188 | 01/01/2017 | Pink Cab | BOSTON MA | 41.44 | 630.76 | 464.1280 | 59 |
| **3** | 10128190 | 01/01/2017 | Pink Cab | BOSTON MA | 31.36 | 463.01 | 373.1840 | 58 |
| **4** | 10128192 | 01/01/2017 | Pink Cab | BOSTON MA | 40.46 | 597.36 | 457.1980 | 57 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **359387** | 10439960 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 33.93 | 474.47 | 411.2316 | 52 |
| **359388** | 10439984 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 40.00 | 641.78 | 484.8000 | 51 |
| **359389** | 10440028 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 26.22 | 405.25 | 327.2256 | 52 |
| **359390** | 10440034 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 34.68 | 505.38 | 470.2608 | 51 |
| **359391** | 10440093 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 4.32 | 60.41 | 55.4688 | 53 |

359392 rows × 12 columns

In [39]:

```
profile = ProfileReport(df, title="Pandas Profiling Report")
```

In [40]:

```
profile
```

100% Completed]

Generate report structure: 1/1 [00:09<00:00,
100% 9.26s/it]

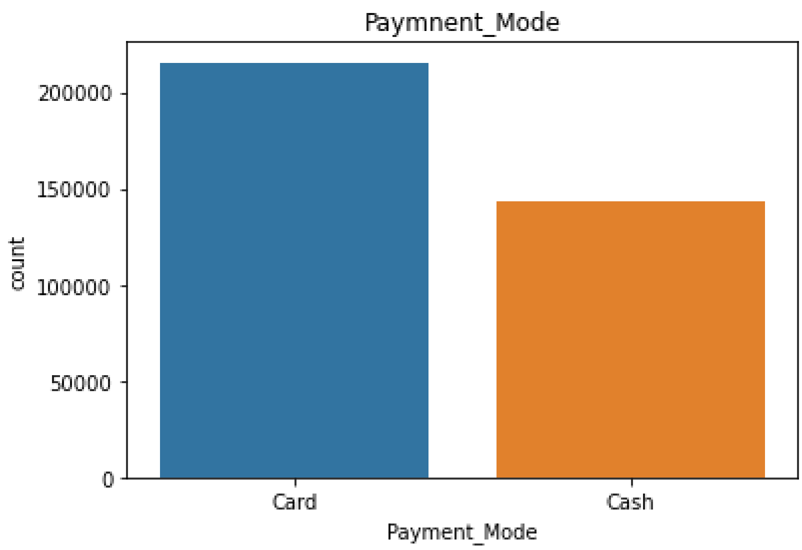
Render HTML: 1/1 [00:03<00:00,
100% 3.55s/it]

EDA

In [57]:

```
#Visualize the cab distribution

print(f'Proportion of Total Card Payment: {df.Payment_Mode.value_counts(normalize = True)[0
print(f'Proportion of Total Cash Payment: {df.Payment_Mode.value_counts(normalize = True)[1

sns.countplot(df['Payment_Mode'])
plt.title('Paymnent_Mode')
plt.show()
```

```
Proportion of Total Card Payment: 59.96 %
Proportion of Total Cash Payment: 40.04 %
```



In [59]:

```
#Let's visualize the number of trips and the frequencies they occured

no_of_journeys = df.groupby(['Date of Travel', 'Company']).size().reset_index().rename(colu

no_of_journeys
```

Out[59]:

| | Date of Travel | Company | count |
|---|---|---|---|
| **0** | 01/01/2017 | Pink Cab | 205 |
| **1** | 01/01/2017 | Yellow Cab | 698 |
| **2** | 01/01/2018 | Pink Cab | 89 |
| **3** | 01/01/2018 | Yellow Cab | 268 |
| **4** | 01/02/2016 | Pink Cab | 13 |
| **...** | ... | ... | ... |
| **2185** | 31/12/2016 | Yellow Cab | 630 |
| **2186** | 31/12/2017 | Pink Cab | 180 |
| **2187** | 31/12/2017 | Yellow Cab | 547 |
| **2188** | 31/12/2018 | Pink Cab | 58 |
| **2189** | 31/12/2018 | Yellow Cab | 198 |

2190 rows × 3 columns

In [63]:

```
#Confirming which day and company had the highest number of travels
no_of_journeys.loc[no_of_journeys['count'].idxmax()]
```

Out[63]:

```
Date of Travel    05/01/2018
Company           Yellow Cab
count                   1494
Name: 291, dtype: object
```

In [65]:

```
#Due to the large number of travel days, we will simply by categorizing the days into month
```

Out[65]:

```
Date of Travel    05/01/2018
Company           Yellow Cab
count                   1494
Name: 291, dtype: object
```

In [66]:

```
ates or missing values, let's calculate the profits made based on Cost of trip and price cha
Charged"] - df["Cost of Trip"]
```

In [67]:

```
#Let us take a look at the profits
df
```

Out[67]:

| | Transaction ID | Date of Travel | Company | City | KM Travelled | Price Charged | Cost of Trip | Custo |
|---|---|---|---|---|---|---|---|---|
| **0** | 10128150 | 01/01/2017 | Pink Cab | ATLANTA GA | 24.00 | 379.79 | 280.8000 | 28 |
| **1** | 10128153 | 01/01/2017 | Pink Cab | ATLANTA GA | 38.88 | 609.62 | 443.2320 | 29 |
| **2** | 10128188 | 01/01/2017 | Pink Cab | BOSTON MA | 41.44 | 630.76 | 464.1280 | 59 |
| **3** | 10128190 | 01/01/2017 | Pink Cab | BOSTON MA | 31.36 | 463.01 | 373.1840 | 58 |
| **4** | 10128192 | 01/01/2017 | Pink Cab | BOSTON MA | 40.46 | 597.36 | 457.1980 | 57 |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **359387** | 10439960 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 33.93 | 474.47 | 411.2316 | 52 |
| **359388** | 10439984 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 40.00 | 641.78 | 484.8000 | 51 |
| **359389** | 10440028 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 26.22 | 405.25 | 327.2256 | 52 |
| **359390** | 10440034 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 34.68 | 505.38 | 470.2608 | 51 |
| **359391** | 10440093 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 4.32 | 60.41 | 55.4688 | 53 |

359392 rows × 13 columns

In [68]:

```
#Let us see the profits generated monthly and yearly

from datetime import datetime, timedelta
def to_date_format(n):
    date_str =(datetime(1899,12,30) + timedelta(n-1)).strftime("%d-%m-%Y")
    date_date =  datetime.strptime(date_str, "%d-%m-%Y")
    return date_date
```

In [74]:

```
df[["day", "month", "year"]] = df["Date of Travel"].str.split("/", expand = True)
```

In [75]:

```
df
```

| Date of Travel | Company | City | KM Travelled | Price Charged | Cost of Trip | Customer ID | Payment_Mode | Gender | Age | (US |
|---|---|---|---|---|---|---|---|---|---|---|
| /01/2017 | Pink Cab | ATLANTA GA | 24.00 | 379.79 | 280.8000 | 28154 | Card | Male | 49 | |
| /01/2017 | Pink Cab | ATLANTA GA | 38.88 | 609.62 | 443.2320 | 29383 | Card | Female | 57 | |
| /01/2017 | Pink Cab | BOSTON MA | 41.44 | 630.76 | 464.1280 | 59347 | Card | Male | 26 | |
| /01/2017 | Pink Cab | BOSTON MA | 31.36 | 463.01 | 373.1840 | 58311 | Card | Female | 19 | |
| /01/2017 | Pink Cab | BOSTON MA | 40.46 | 597.36 | 457.1980 | 57940 | Cash | Female | 30 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| /12/2018 | Yellow Cab | WASHINGTON DC | 33.93 | 474.47 | 411.2316 | 52449 | Card | Female | 40 | |
| /12/2018 | Yellow | WASHINGTON | 40.00 | 641.78 | 484.8000 | 51614 | Card | Female | 55 | |

In [77]:

```
#For easy access, we replace every column header with space _

for col in df.columns:
    if ' ' in col:
        df = df.rename(columns={col:col.replace(' ','_')})
```

In [78]:

```
df
```

Out[78]:

| City | KM_Travelled | Price_Charged | Cost_of_Trip | Customer_ID | Payment_Mode | Gender | Age | Inc |
|---|---|---|---|---|---|---|---|---|
| \ GA | 24.00 | 379.79 | 280.8000 | 28154 | Card | Male | 49 | |
| \ GA | 38.88 | 609.62 | 443.2320 | 29383 | Card | Female | 57 | |
| ɪ MA | 41.44 | 630.76 | 464.1280 | 59347 | Card | Male | 26 | |
| ɪ MA | 31.36 | 463.01 | 373.1840 | 58311 | Card | Female | 19 | |
| ɪ MA | 40.46 | 597.36 | 457.1980 | 57940 | Cash | Female | 30 | |
| ... | ... | ... | ... | ... | ... | ... | ... | |
| TON DC | 33.93 | 474.47 | 411.2316 | 52449 | Card | Female | 40 | |
| TON DC | 40.00 | 641.78 | 484.8000 | 51614 | Card | Female | 55 | |
| TON DC | 26.22 | 405.25 | 327.2256 | 52389 | Card | Female | 29 | |
| TON DC | 34.68 | 505.38 | 470.2608 | 51877 | Cash | Male | 46 | |
| TON DC | 4.32 | 60.41 | 55.4688 | 53810 | Cash | Male | 23 | |

In [79]:

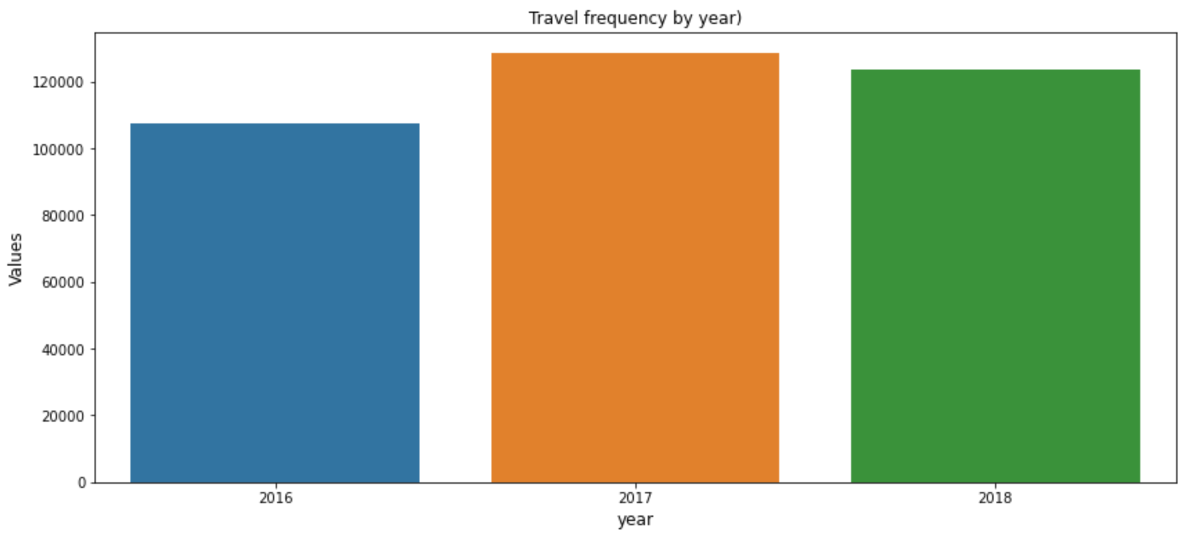
```
#What year and months had the highest number of trips

plot0 = df.groupby(['year']).Transaction_ID.count()
plot0
```

Out[79]:

```
year
2016    107319
2017    128510
2018    123563
Name: Transaction_ID, dtype: int64
```

In [87]:

```
plt.figure(figsize=(14,6))
sns.barplot(x=plot0.index,y=plot0.values)
plt.title('Travel frequency by year)',fontsize = 12)
plt.xlabel('year', fontsize = 12)
plt.ylabel('Values',fontsize = 12)
plt.show()
```




In [93]:

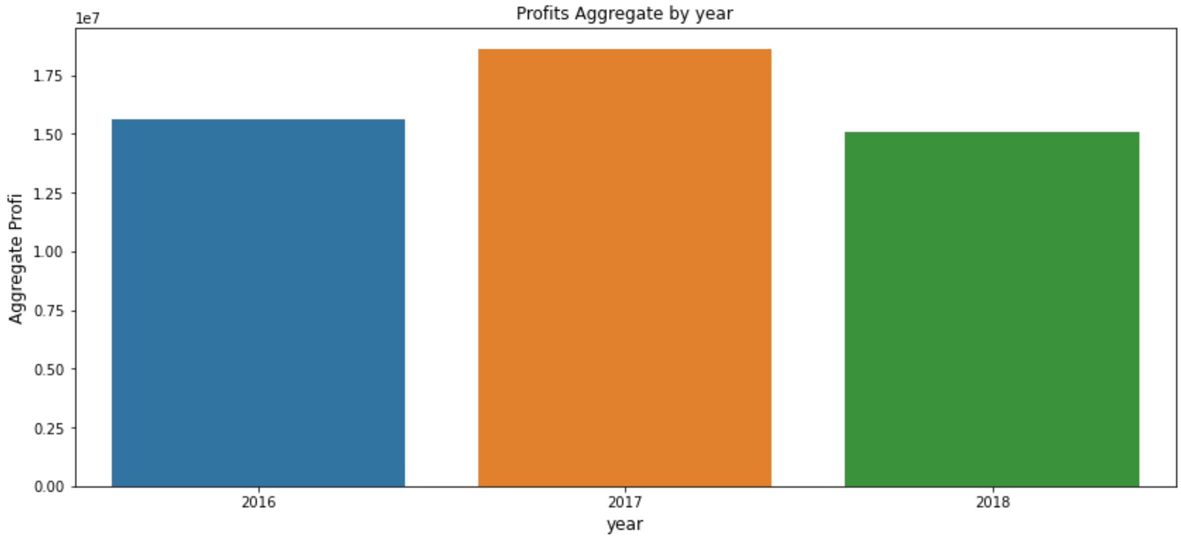
```
plot1 = df.groupby(['year']).Profits.sum()
plot1
```

Out[93]:

```
year
2016    1.564051e+07
2017    1.860963e+07
2018    1.507756e+07
Name: Profits, dtype: float64
```

In [94]:

```
plt.figure(figsize=(14,6))
sns.barplot(x=plot1.index,y=plot1.values)
plt.title('Profits Aggregate by year',fontsize = 12)
plt.xlabel('year', fontsize = 12)
plt.ylabel('Aggregate Profi',fontsize = 12)
plt.show()
```




In [81]:

```
plot2 = df.groupby(['month']).Profits.sum()
plot2
```

Out[81]:

```
month
01    3.746490e+06
02    3.228020e+06
03    3.450311e+06
04    3.321361e+06
05    4.152157e+06
06    3.813054e+06
07    3.163447e+06
08    3.244522e+06
09    4.636818e+06
10    4.946949e+06
11    5.419647e+06
12    6.204925e+06
Name: Profits, dtype: float64
```

In [82]:

```
plot3 = df.groupby(['Company']).Profits.sum()
plot3
```

Out[82]:

```
Company
Pink Cab      5.307328e+06
Yellow Cab    4.402037e+07
Name: Profits, dtype: float64
```

In [86]:

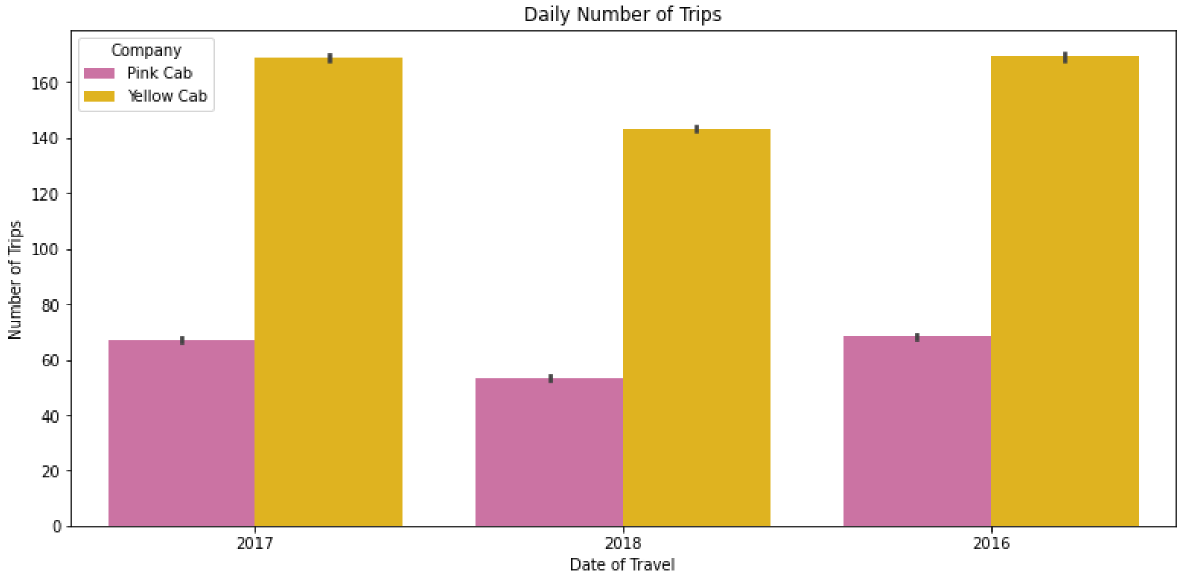
```
plot4 = df.groupby(['City']).Profits.mean()
plot4
```

```
City
ATLANTA GA        111.477158
AUSTIN TX         107.577824
BOSTON MA          59.568883
CHICAGO IL         59.820104
DALLAS TX         160.856957
DENVER CO         103.943793
LOS ANGELES CA     91.847452
MIAMI FL          117.493220
NASHVILLE TN       49.678478
NEW YORK NY       279.947491
ORANGE COUNTY     114.766920
PHOENIX AZ         93.479109
PITTSBURGH PA      64.863638
SACRAMENTO CA      49.567466
SAN DIEGO CA       77.467955
SEATTLE WA         75.613962
SILICON VALLEY    154.561013
TUCSON AZ          72.636300
WASHINGTON DC      79.860762
```

In [ ]:

In [102]:

```
#Assigning Colors for companies
palette = ['#d965a4', '#ffc400']

plt.figure(figsize = (13,6))
sns.barplot(x = 'year', y = 'Profits', data = df, hue = 'Company',
             palette = palette);
plt.title('Profits return by Company')
plt.xlabel('Date of Travel')
plt.ylabel('Profits')
```



In [106]:

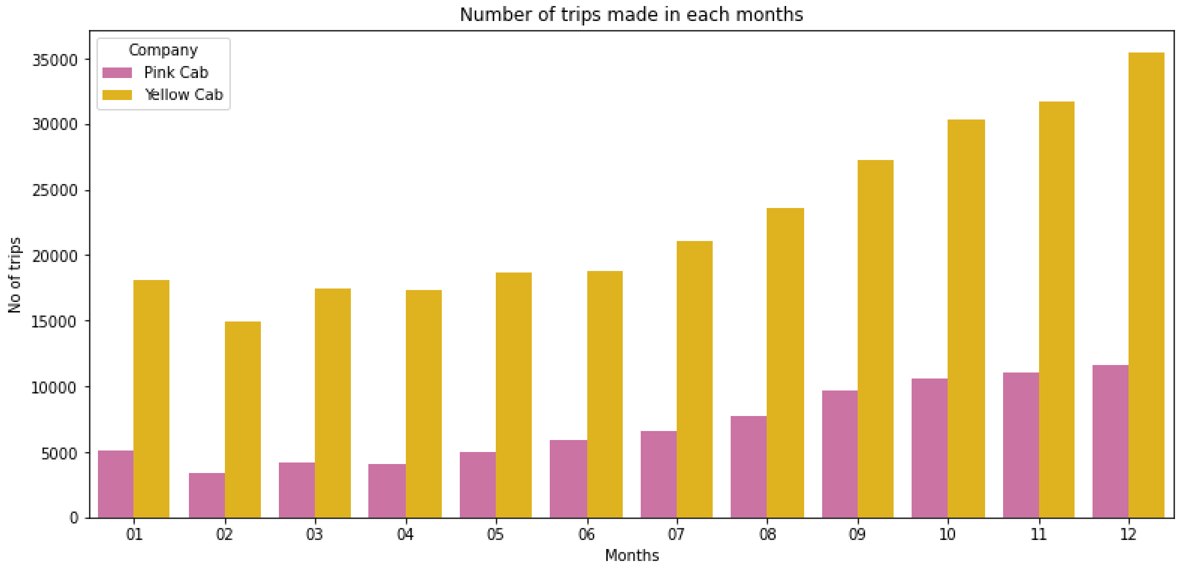
```
trip = df.groupby(['month', 'Company']).size().reset_index().rename(columns = {0 : 'count'}

trip
```

Out[106]:

| | month | Company | count |
|---|---|---|---|
| **0** | 01 | Pink Cab | 5057 |
| **1** | 01 | Yellow Cab | 18117 |
| **2** | 02 | Pink Cab | 3375 |
| **3** | 02 | Yellow Cab | 14932 |
| **4** | 03 | Pink Cab | 4223 |
| **5** | 03 | Yellow Cab | 17423 |
| **6** | 04 | Pink Cab | 4083 |
| **7** | 04 | Yellow Cab | 17351 |
| **8** | 05 | Pink Cab | 4960 |
| **9** | 05 | Yellow Cab | 18741 |
| **10** | 06 | Pink Cab | 5877 |
| **11** | 06 | Yellow Cab | 18836 |
| **12** | 07 | Pink Cab | 6590 |
| **13** | 07 | Yellow Cab | 21086 |
| **14** | 08 | Pink Cab | 7739 |
| **15** | 08 | Yellow Cab | 23584 |
| **16** | 09 | Pink Cab | 9658 |
| **17** | 09 | Yellow Cab | 27201 |
| **18** | 10 | Pink Cab | 10576 |
| **19** | 10 | Yellow Cab | 30276 |
| **20** | 11 | Pink Cab | 11005 |
| **21** | 11 | Yellow Cab | 31695 |
| **22** | 12 | Pink Cab | 11568 |
| **23** | 12 | Yellow Cab | 35439 |

In [108]:

```
#Assigning Colors for companies
palette = ['#d965a4', '#ffc400']

plt.figure(figsize = (13,6))
sns.barplot(x = 'month', y = 'count', data = trip, hue = 'Company',
            palette = palette);
plt.title('Number of trips made in each months');
plt.xlabel('Months');
plt.ylabel('No of trips');
```



In [110]:

```
monthly_city_trips = df.groupby(['year', 'month', 'City', 'Company']).size().\
                                    reset_index().rename(columns = {0:'count'})

monthly_city_trips['month_level'] = monthly_city_trips['year'].astype('str') + "_" + \
                                                monthly_city_trips['month'].astype('str')

monthly_city_trips
```

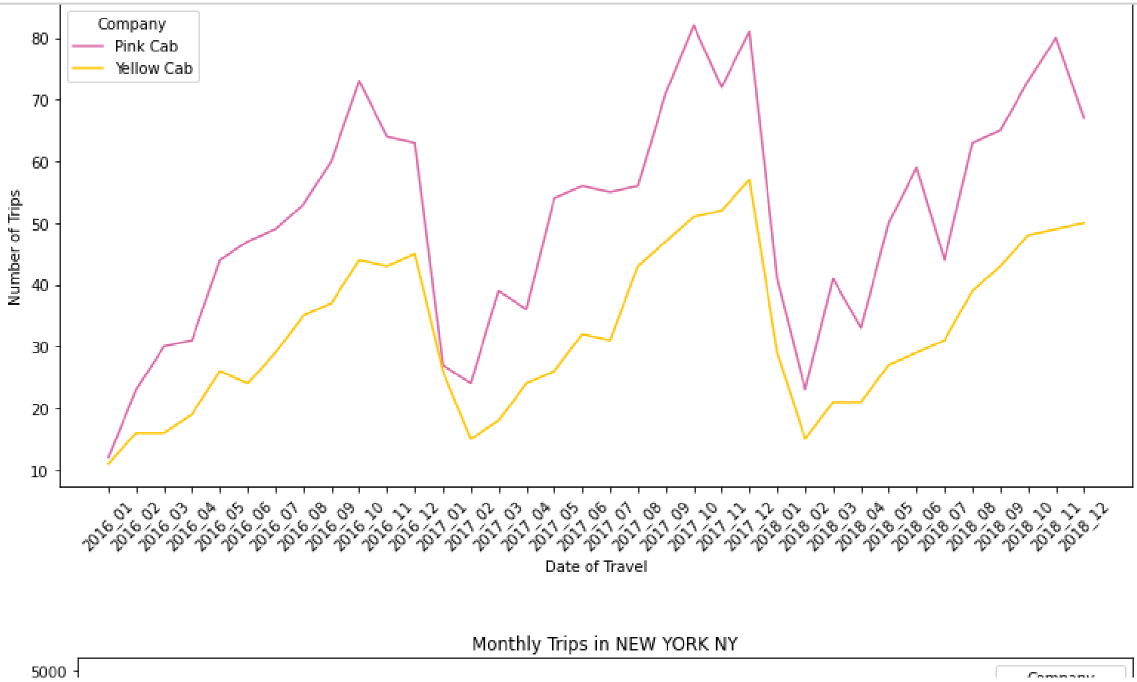
Out[110]:

| | year | month | City | Company | count | month_level |
|---|---|---|---|---|---|---|
| **0** | 2016 | 01 | ATLANTA GA | Pink Cab | 21 | 2016_01 |
| **1** | 2016 | 01 | ATLANTA GA | Yellow Cab | 85 | 2016_01 |
| **2** | 2016 | 01 | AUSTIN TX | Pink Cab | 7 | 2016_01 |
| **3** | 2016 | 01 | AUSTIN TX | Yellow Cab | 24 | 2016_01 |
| **4** | 2016 | 01 | BOSTON MA | Pink Cab | 72 | 2016_01 |
| **...** | ... | ... | ... | ... | ... | ... |
| **1363** | 2018 | 12 | SILICON VALLEY | Yellow Cab | 205 | 2018_12 |
| **1364** | 2018 | 12 | TUCSON AZ | Pink Cab | 29 | 2018_12 |
| **1365** | 2018 | 12 | TUCSON AZ | Yellow Cab | 50 | 2018_12 |
| **1366** | 2018 | 12 | WASHINGTON DC | Pink Cab | 188 | 2018_12 |
| **1367** | 2018 | 12 | WASHINGTON DC | Yellow Cab | 1500 | 2018_12 |

1368 rows × 6 columns

In [113]:

```
for i in monthly_city_trips.City.unique():
    plt.figure(figsize = (13,6))
    monthly_df = monthly_city_trips.query(f"City == '{i}'")

    sns.lineplot(x = 'month_level', y = 'count', data = monthly_df, hue = 'Company',
                 palette = palette);

    plt.title(f'Monthly Trips in {i}');
    plt.xlabel('Date of Travel');
    plt.ylabel('Number of Trips');
    plt.xticks(rotation = 45)
```



In [114]:

```
loss = df.query("Profits <= 0")
loss
```

Out[114]:

| | Transaction_ID | Date_of_Travel | Company | City | KM_Travelled | Price_Charged |
|---|---|---|---|---|---|---|
| **44** | 10128619 | 01/01/2017 | Pink Cab | ORANGE COUNTY | 23.00 | 242.47 |
| **71** | 10129181 | 01/01/2017 | Yellow Cab | BOSTON MA | 21.80 | 302.30 |
| **76** | 10129219 | 01/01/2017 | Yellow Cab | BOSTON MA | 37.80 | 524.16 |
| **139** | 10129921 | 01/01/2017 | Yellow Cab | MIAMI FL | 14.43 | 197.65 |
| **201** | 10130766 | 01/01/2017 | Yellow Cab | SACRAMENTO CA | 24.50 | 335.03 |
| **...** | ... | ... | ... | ... | ... | ... |
| **359335** | 10438112 | 31/12/2018 | Yellow Cab | CHICAGO IL | 22.42 | 269.98 |
| **359336** | 10438116 | 31/12/2018 | Yellow Cab | CHICAGO IL | 36.58 | 435.76 |
| **359338** | 10438152 | 31/12/2018 | Yellow Cab | CHICAGO IL | 19.62 | 233.72 |
| **359340** | 10438192 | 31/12/2018 | Yellow Cab | CHICAGO IL | 38.61 | 469.94 |
| **359386** | 10439934 | 31/12/2018 | Yellow Cab | WASHINGTON DC | 38.11 | 510.48 |

24823 rows × 16 columns

In [ ]:

```
#Let us take a look at the correlation of the variables
dataplot = sns.heatmap(df.corr(), cmap="YlGnBu", annot=True)

# displaying heatmap
plt.show()
```

From the heatmap, there is a high correlation between KM traveled, Price charged and cost of trip. Also, there is a high correlation between profit made and price charged

In [ ]:

```
# I am yet to complete my analysis and findings and it is already a day after the due date.

#But I have to submit this now
```