EDA - TRAVEL

In [23]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [25]:
```
df = pd.read_csv("Travel.csv")
```

In [27]:
```
df
```

Out[27]:

	CustomerID	ProdTaken	Age	TypeofContact	CityTier	DurationOfPitch	Occupation
0	200000	1	41.0	Self Enquiry	3	6.0	Salaried
1	200001	0	49.0	Company Invited	1	14.0	Salaried
2	200002	1	37.0	Self Enquiry	1	8.0	Free Lancer
3	200003	0	33.0	Company Invited	1	9.0	Salaried
4	200004	0	NaN	Self Enquiry	1	8.0	Small Business
...	...	...	...	...	...	...	...
4883	204883	1	49.0	Self Enquiry	3	9.0	Small Business
4884	204884	1	28.0	Company Invited	1	31.0	Salaried
4885	204885	1	52.0	Self Enquiry	3	17.0	Salaried
4886	204886	1	19.0	Self Enquiry	3	16.0	Small Business
4887	204887	1	36.0	Self Enquiry	1	14.0	Salaried

4888 rows × 20 columns

In [29]:
```
df.dtypes
```

```
Out[29]: CustomerID                    int64
         ProdTaken                     int64
         Age                         float64
         TypeofContact                object
         CityTier                      int64
         DurationOfPitch             float64
         Occupation                   object
         Gender                       object
         NumberOfPersonVisiting        int64
         NumberOfFollowups           float64
         ProductPitched               object
         PreferredPropertyStar       float64
         MaritalStatus                object
         NumberOfTrips               float64
         Passport                      int64
         PitchSatisfactionScore        int64
         OwnCar                        int64
         NumberOfChildrenVisiting    float64
         Designation                  object
         MonthlyIncome               float64
         dtype: object
```

```
In [33]: df.columns
```

```
Out[33]: Index(['CustomerID', 'ProdTaken', 'Age', 'TypeofContact', 'CityTier',
                'DurationOfPitch', 'Occupation', 'Gender', 'NumberOfPersonVisiting',
                'NumberOfFollowups', 'ProductPitched', 'PreferredPropertyStar',
                'MaritalStatus', 'NumberOfTrips', 'Passport', 'PitchSatisfactionScore',
                'OwnCar', 'NumberOfChildrenVisiting', 'Designation', 'MonthlyIncome'],
               dtype='object')
```

```
In [31]: df['Gender'].unique()
```

```
Out[31]: array(['Female', 'Male', 'Fe Male'], dtype=object)
```

```
In [10]: df.replace("Fe Male","Female",inplace=True)
```

```
In [33]: df['Gender'].unique()
```

```
Out[33]: array(['Female', 'Male', 'Fe Male'], dtype=object)
```

```
In [47]: df['Gender'].value_counts()
```

```
Out[47]: Gender
         Male      2916
         Female    1972
         Name: count, dtype: int64
```

```
In [51]: df.isnull().sum().sort_values(ascending=False)
```

```
Out[51]: DurationOfPitch            251
         MonthlyIncome              233
         Age                        226
         NumberOfTrips              140
         NumberOfChildrenVisiting    66
         NumberOfFollowups           45
         PreferredPropertyStar       26
         TypeofContact               25
         Designation                  0
         OwnCar                       0
         PitchSatisfactionScore       0
         Passport                     0
         CustomerID                   0
         MaritalStatus                0
         ProdTaken                    0
         NumberOfPersonVisiting       0
         Gender                       0
         Occupation                   0
         CityTier                     0
         ProductPitched               0
         dtype: int64
```

```
In [61]: round(100*(df.isnull().sum()/ len(df.index)),2)
```

```
Out[61]: CustomerID                 0.00
         ProdTaken                  0.00
         Age                        4.62
         TypeofContact              0.51
         CityTier                   0.00
         DurationOfPitch            5.14
         Occupation                 0.00
         Gender                     0.00
         NumberOfPersonVisiting     0.00
         NumberOfFollowups          0.92
         ProductPitched             0.00
         PreferredPropertyStar      0.53
         MaritalStatus              0.00
         NumberOfTrips              2.86
         Passport                   0.00
         PitchSatisfactionScore     0.00
         OwnCar                     0.00
         NumberOfChildrenVisiting   1.35
         Designation                0.00
         MonthlyIncome              4.77
         dtype: float64
```

```
In [74]: df.dropna(axis=0,inplace=True)
```

```
In [76]: round(100*(df.isnull().sum()/ len(df.index)),2) #All null values are dropped
```

Out[76]:
```
CustomerID                 0.0
ProdTaken                  0.0
Age                        0.0
TypeofContact              0.0
CityTier                   0.0
DurationOfPitch            0.0
Occupation                 0.0
Gender                     0.0
NumberOfPersonVisiting     0.0
NumberOfFollowups          0.0
ProductPitched             0.0
PreferredPropertyStar      0.0
MaritalStatus              0.0
NumberOfTrips              0.0
Passport                   0.0
PitchSatisfactionScore     0.0
OwnCar                     0.0
NumberOfChildrenVisiting   0.0
Designation                0.0
MonthlyIncome              0.0
dtype: float64
```

- Separated categorical and numerical columns

In [42]:
```
cats = ['ProdTaken', 'CityTier', 'PreferredPropertyStar', 'Passport',
         'OwnCar', 'PitchSatisfactionScore', 'TypeofContact', 'Occupation', 'Gender'
         'MaritalStatus', 'Designation']
nums = ['CustomerID', 'Age', 'DurationOfPitch',
        'NumberOfPersonVisiting', 'NumberOfFollowups',
        'NumberOfTrips',
        'NumberOfChildrenVisiting', 'MonthlyIncome']
```

In [44]:
```
cats
```

Out[44]:
```
['ProdTaken',
 'CityTier',
 'PreferredPropertyStar',
 'Passport',
 'OwnCar',
 'PitchSatisfactionScore',
 'TypeofContact',
 'Occupation',
 'Gender',
 'ProductPitched',
 'MaritalStatus',
 'Designation']
```

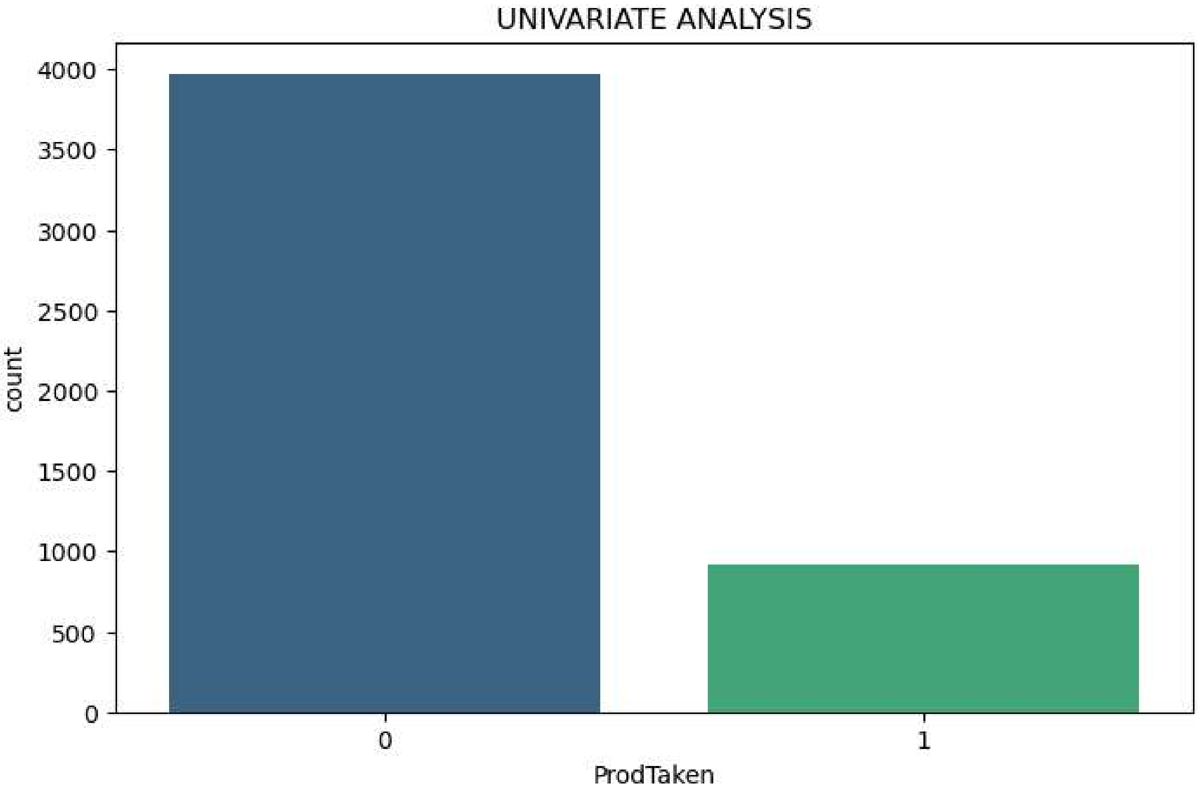
In [46]:
```
nums
```

```
Out[46]: ['CustomerID',
          'Age',
          'DurationOfPitch',
          'NumberOfPersonVisiting',
          'NumberOfFollowups',
          'NumberOfTrips',
          'NumberOfChildrenVisiting',
          'MonthlyIncome']
```

UNIVARIATE ANALYSIS

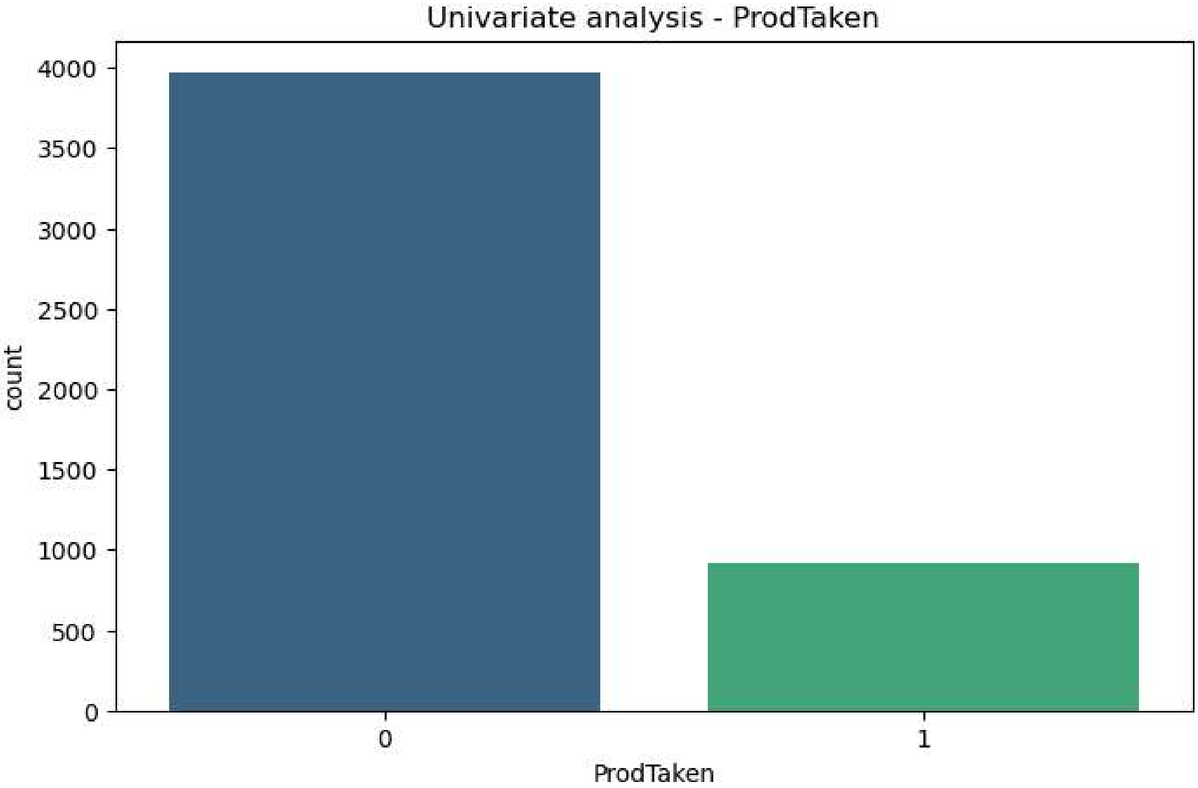
- FOR CATEGORICAL DATA

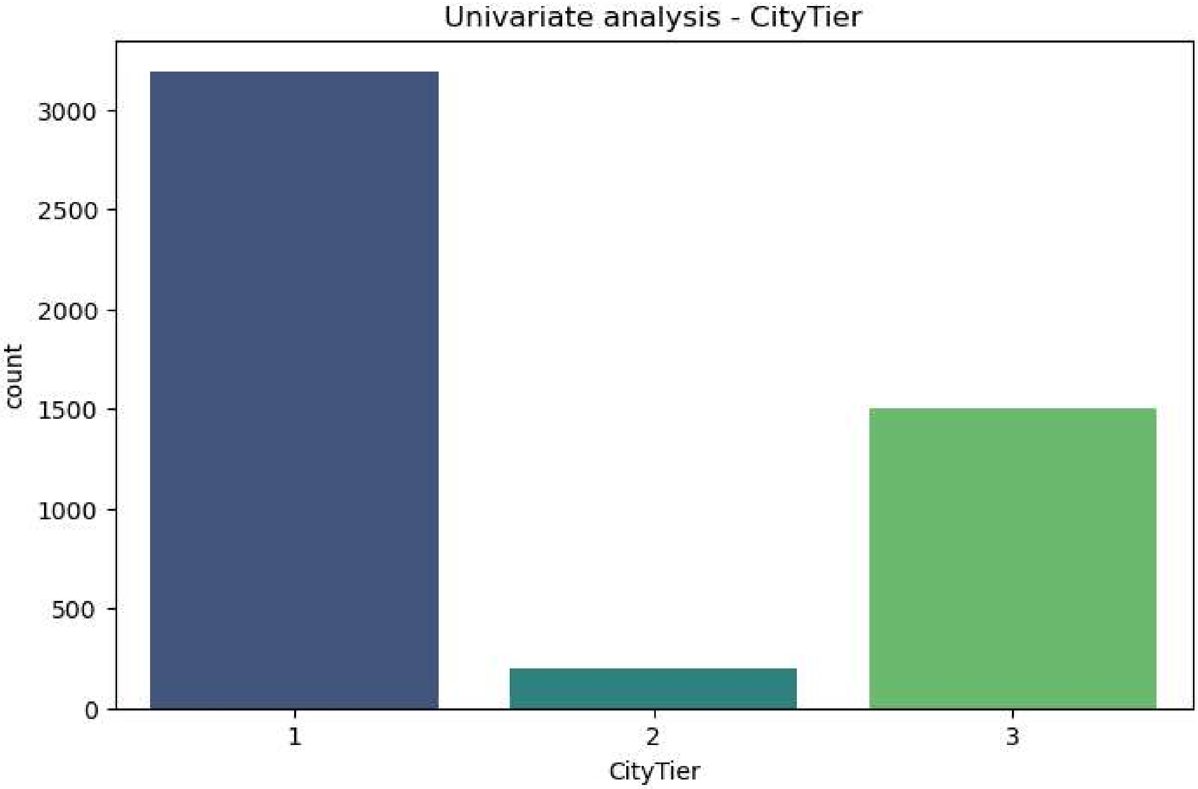
```
In [52]: plt.figure(figsize=(8,5))
         sns.countplot(x='ProdTaken',data=df,palette='viridis')
         plt.title("UNIVARIATE ANALYSIS")
         plt.show()
```

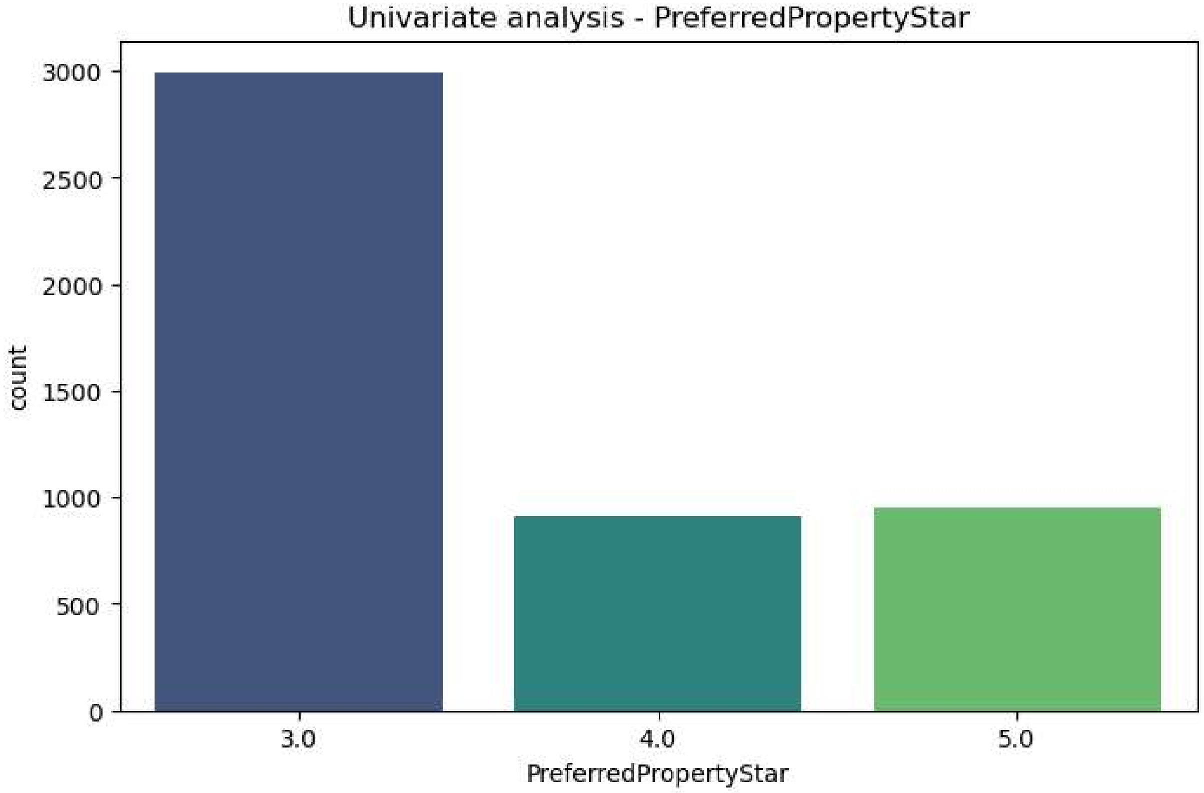


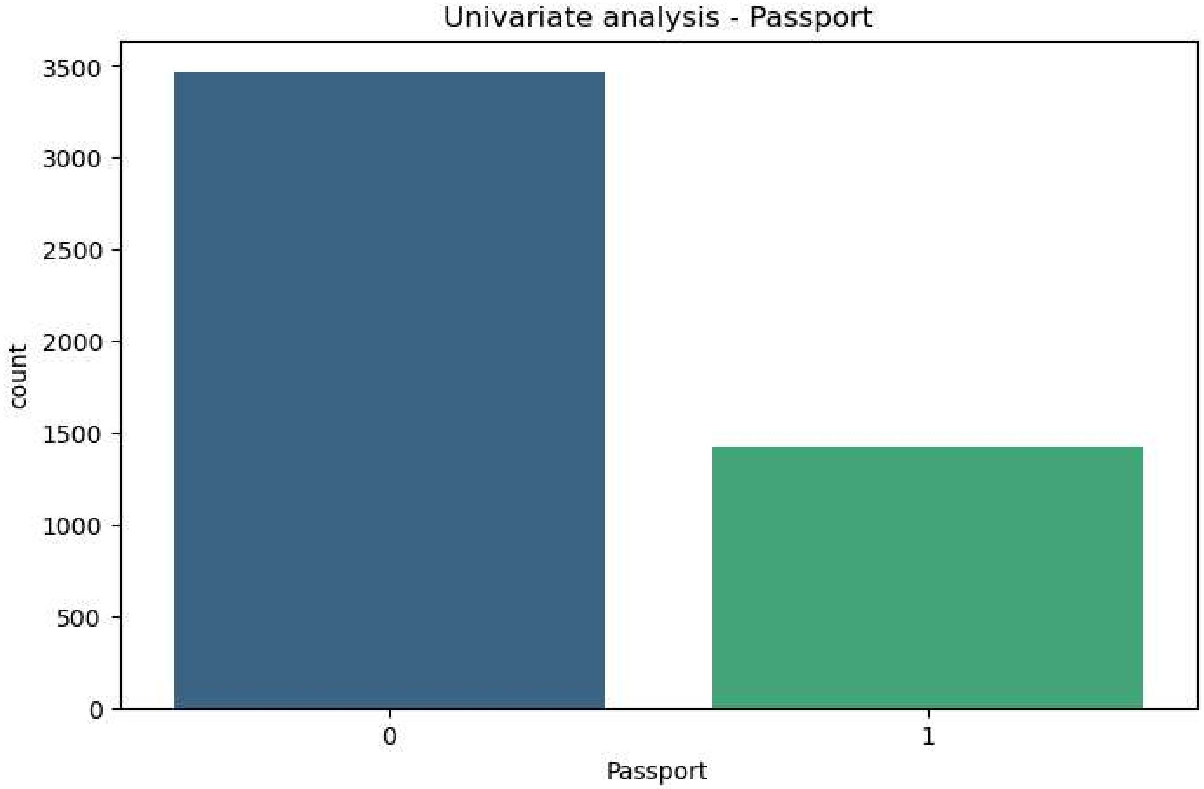
- ALL CATEGORICAL DATA COUNTPLOTS

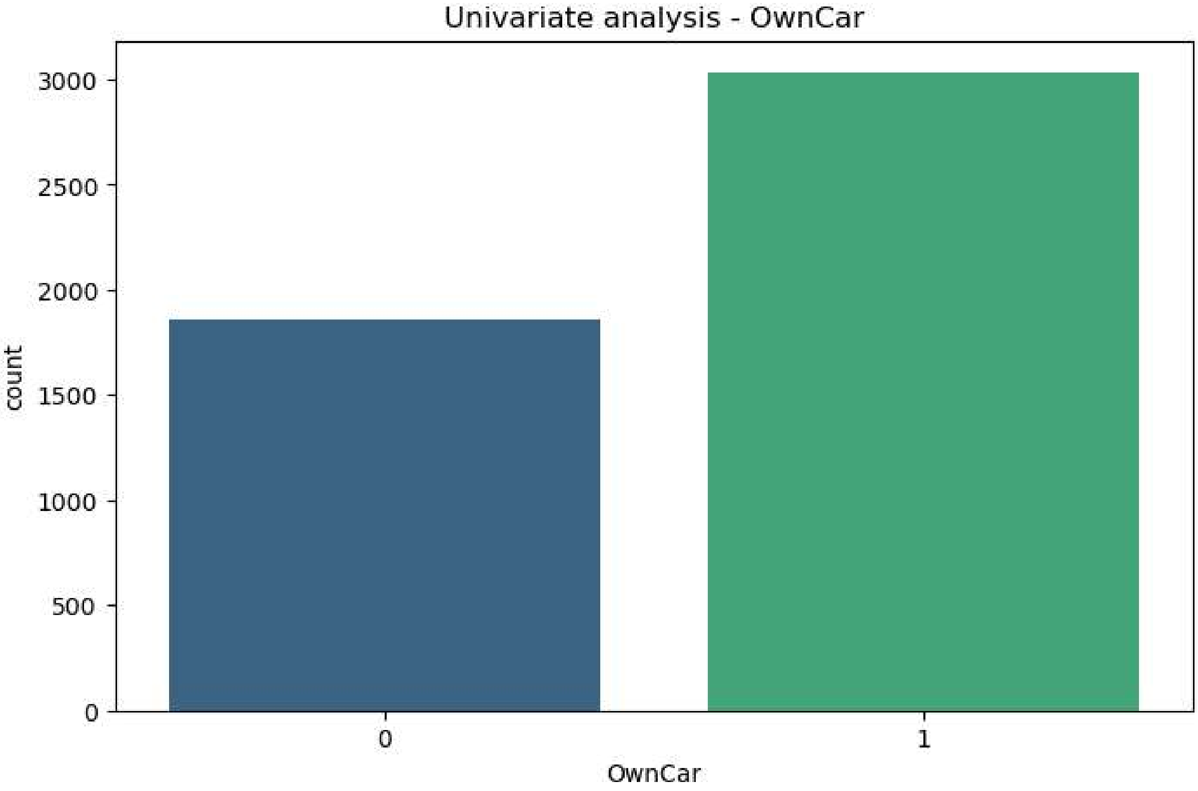
```
In [54]: for cat_column in cats:
             plt.figure(figsize=(8, 5))
             sns.countplot(x = cat_column, data=df, palette = 'viridis')
             plt.title(f"Univariate analysis - {cat_column}")
             plt.show()
```

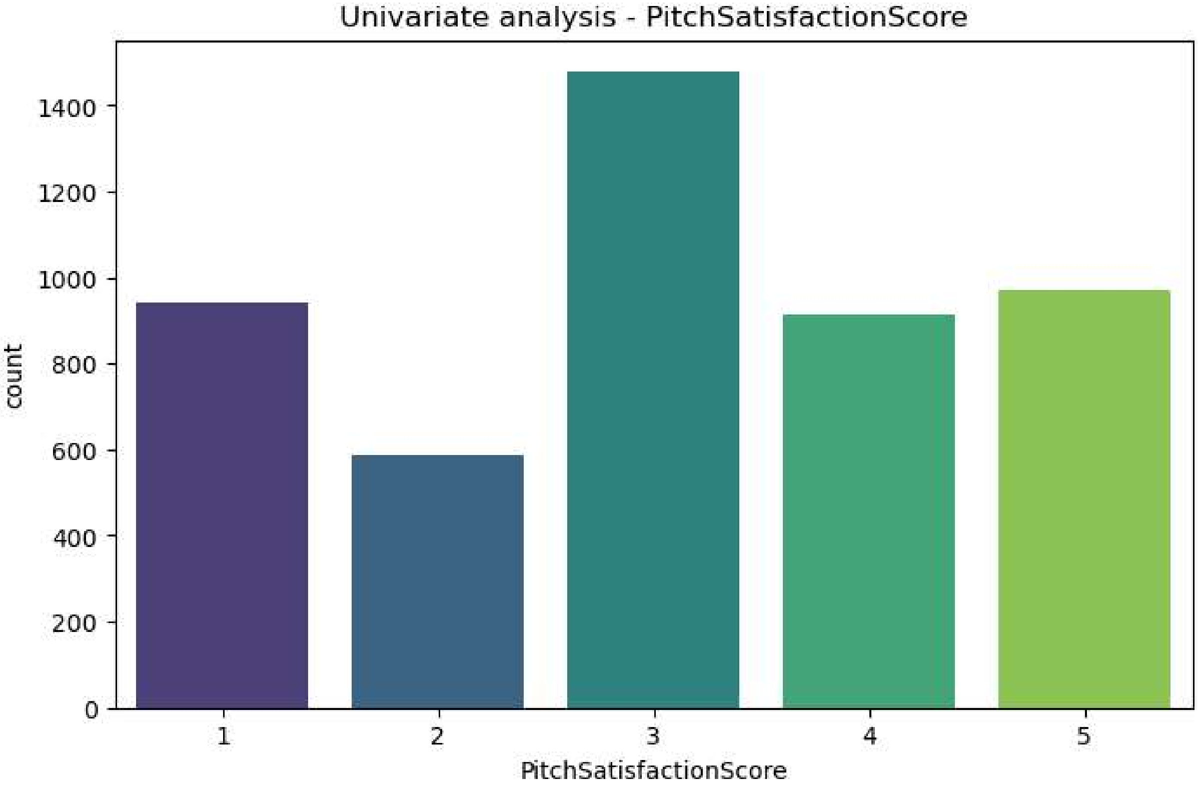

Univariate analysis - ProdTaken
count
4000
3500
3000
2500
2000
1500
1000
500
0
0
1
ProdTaken

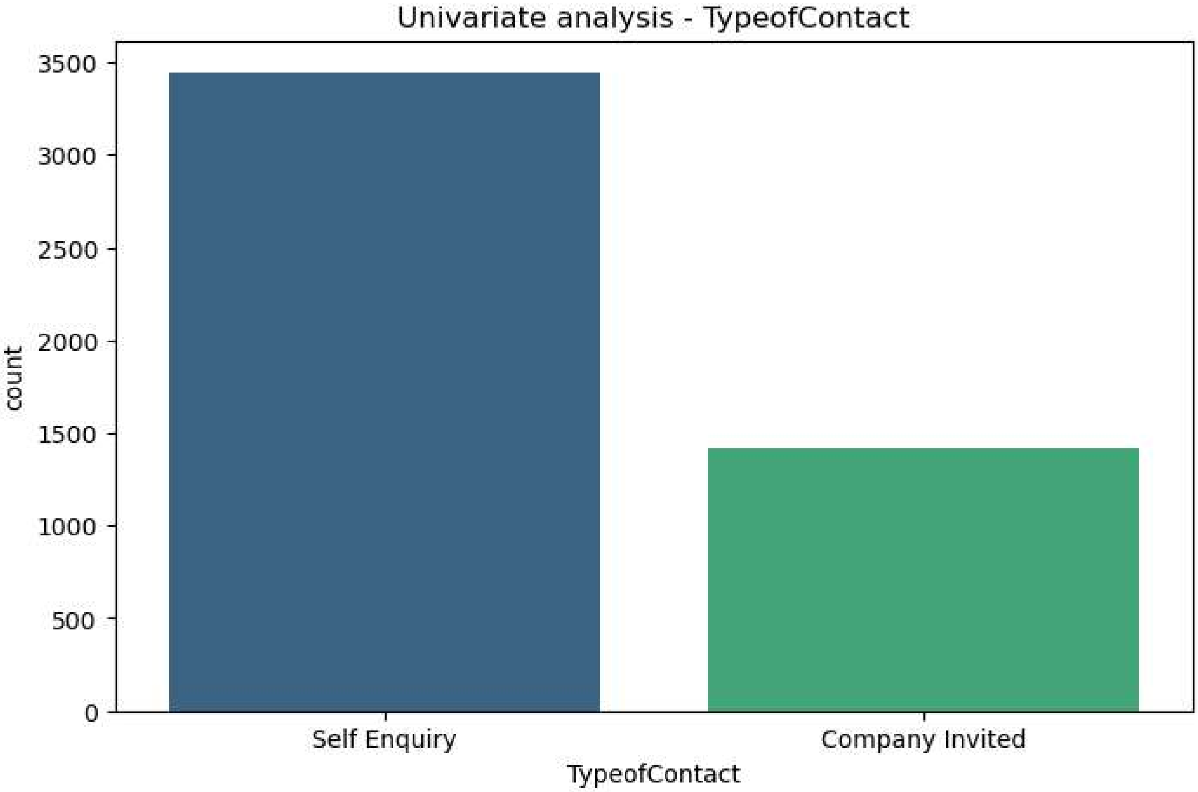

Univariate analysis - CityTier
count
3000
2500
2000
1500
1000
500
0
1
2
3
CityTier

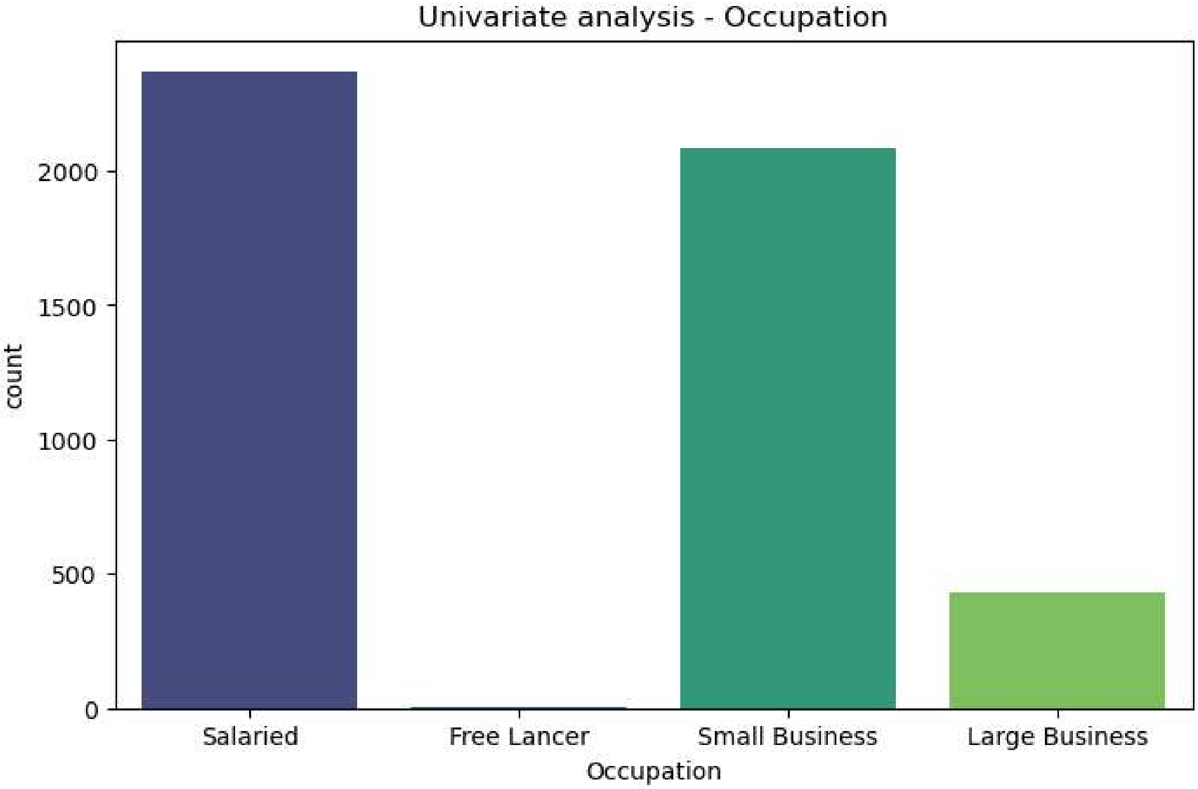

Univariate analysis - PreferredPropertyStar
count
3000
2500
2000
1500
1000
500
0
3.0
4.0
5.0
PreferredPropertyStar

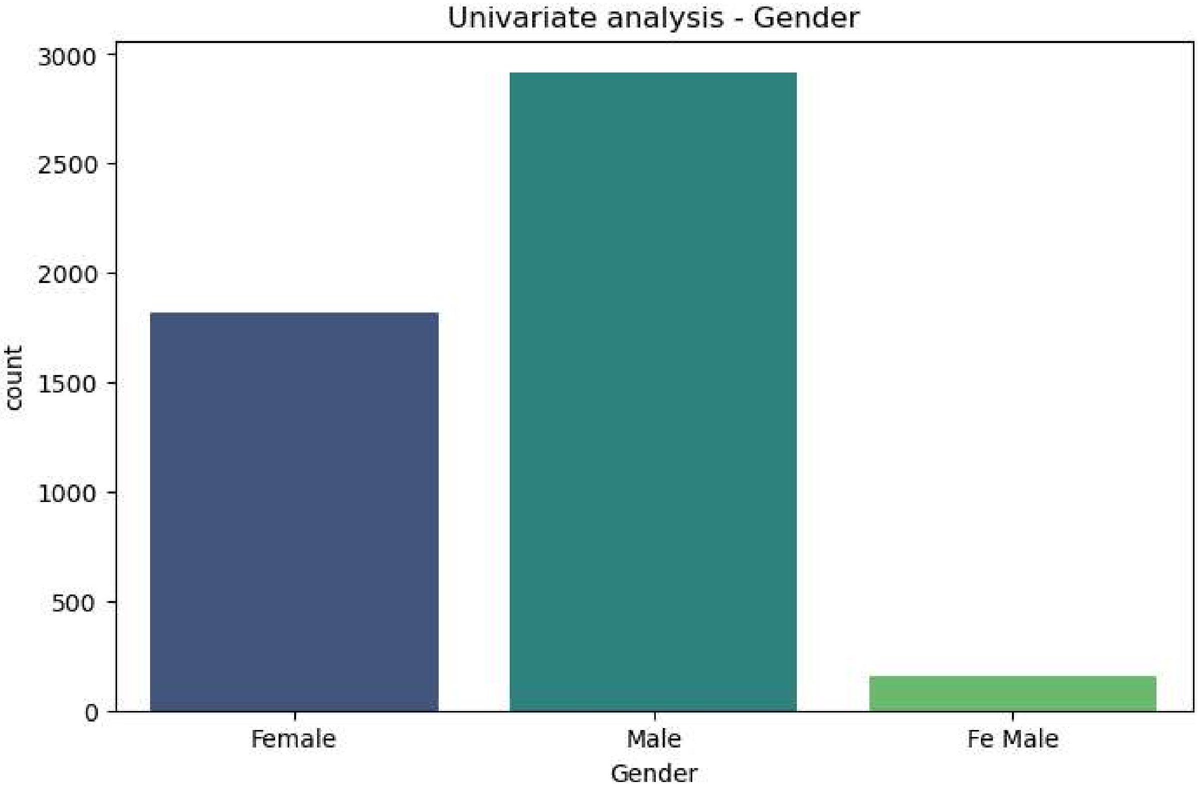

Univariate analysis - Passport
count
3500
3000
2500
2000
1500
1000
500
0
0
1
Passport

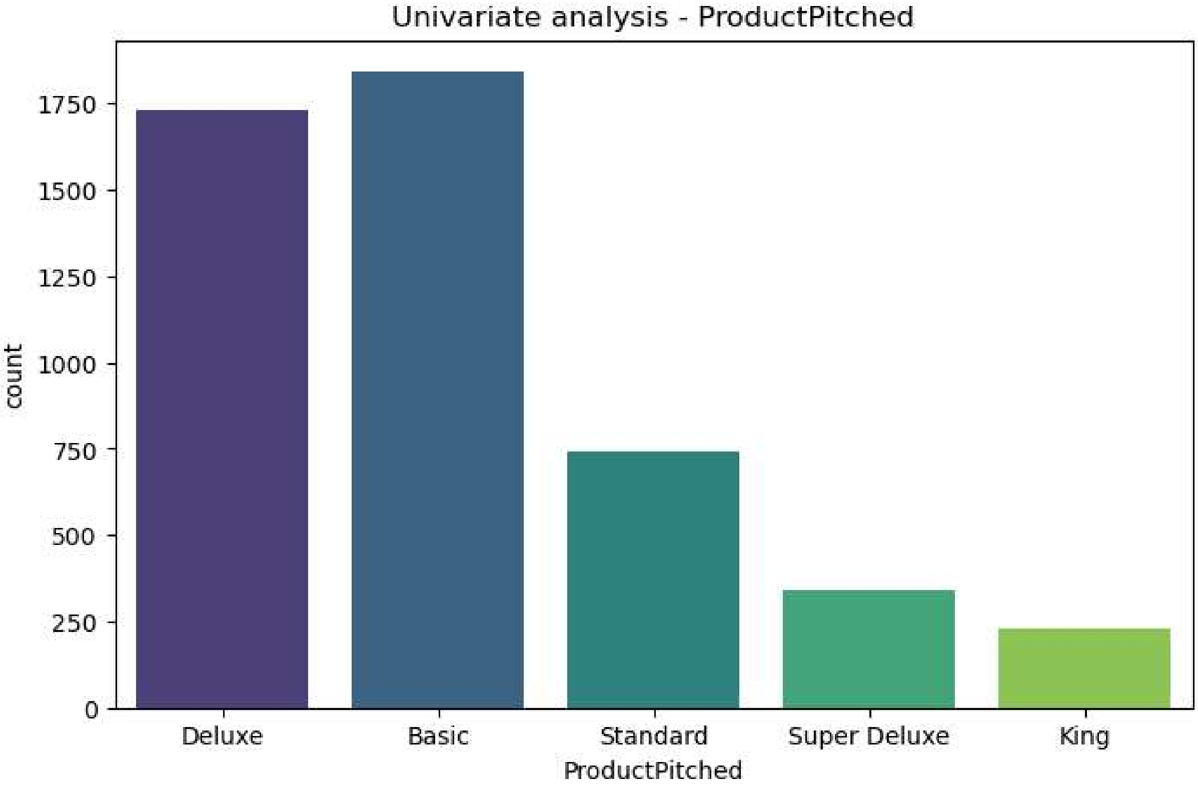

Univariate analysis - OwnCar
3000
2500
2000
1500
1000
500
0
count
0
1
OwnCar

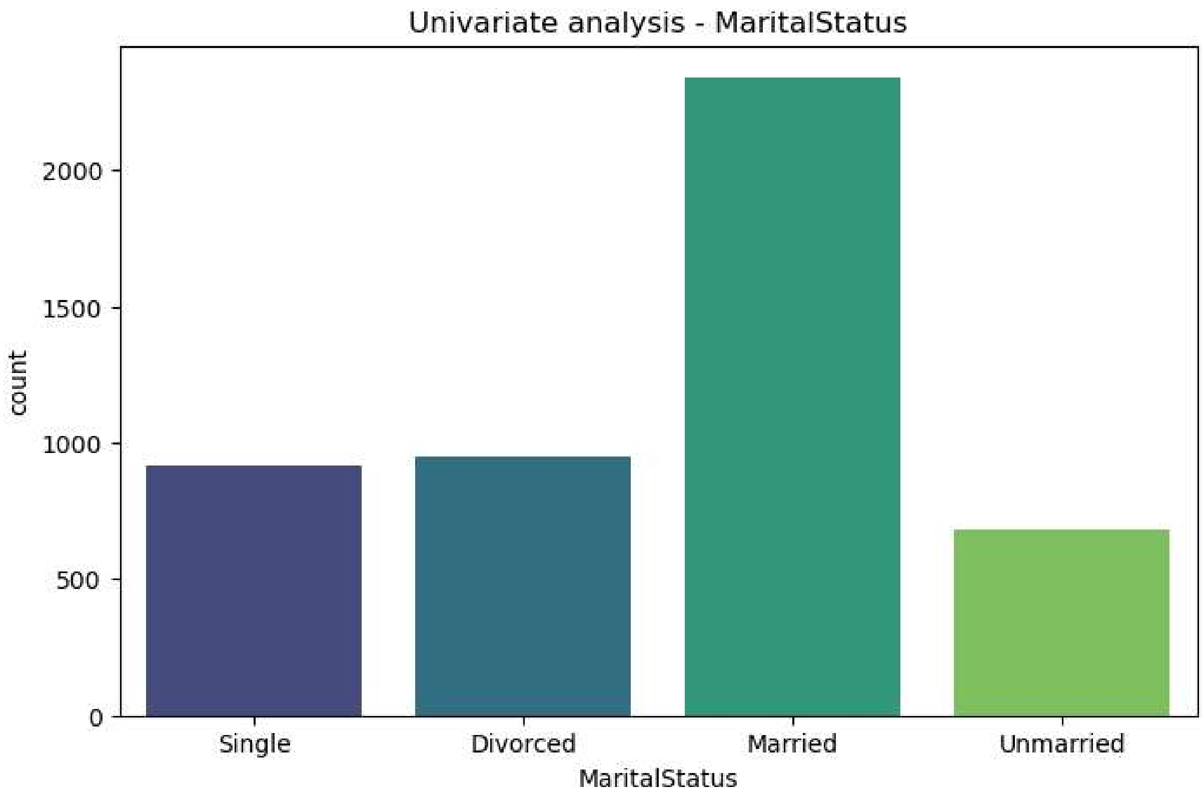

Univariate analysis - PitchSatisfactionScore
1400
1200
1000
800
600
400
200
0
count
1
2
3
4
5
PitchSatisfactionScore

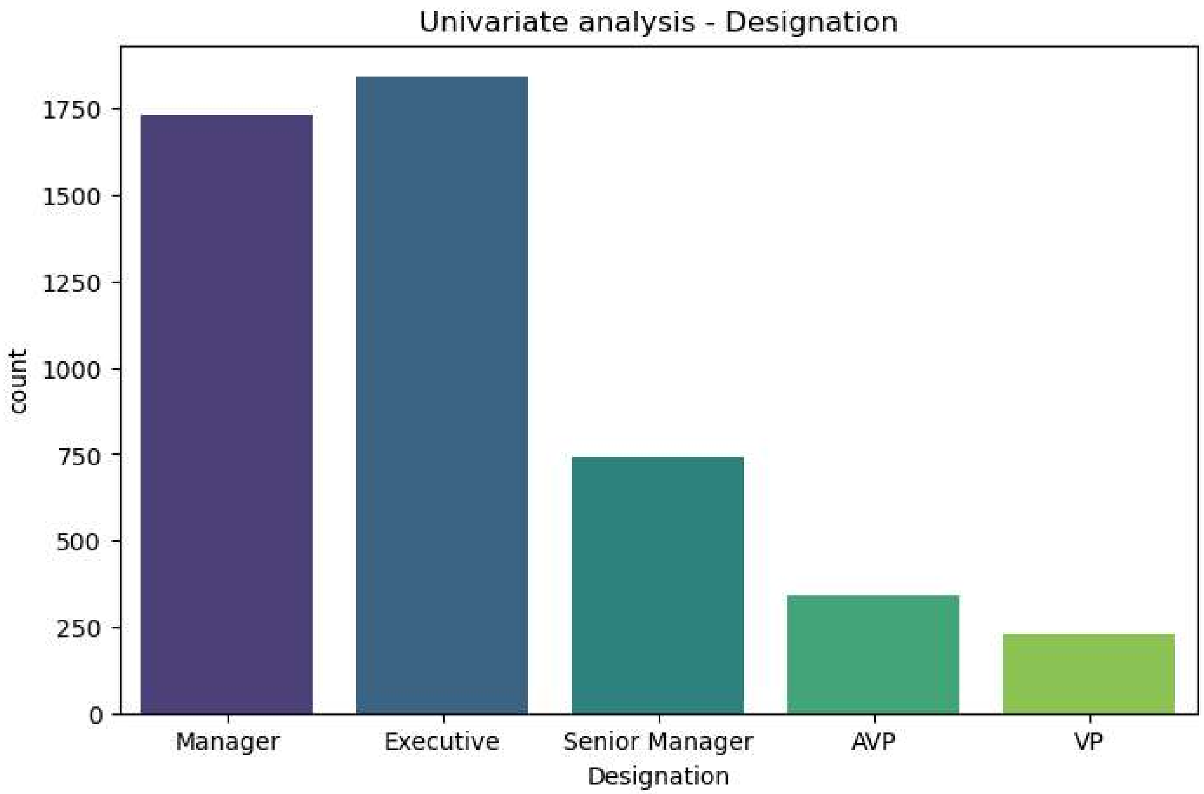

Univariate analysis - TypeofContact
count
0
500
1000
1500
2000
2500
3000
3500
Self Enquiry
Company Invited
TypeofContact


Univariate analysis - Occupation
count
0
500
1000
1500
2000
Salaried
Free Lancer
Small Business
Large Business
Occupation


Univariate analysis - Gender
count
3000
2500
2000
1500
1000
500
0
Female
Male
Fe Male
Gender


Univariate analysis - ProductPitched
count
1750
1500
1250
1000
750
500
250
0
Deluxe
Basic
Standard
Super Deluxe
King
ProductPitched

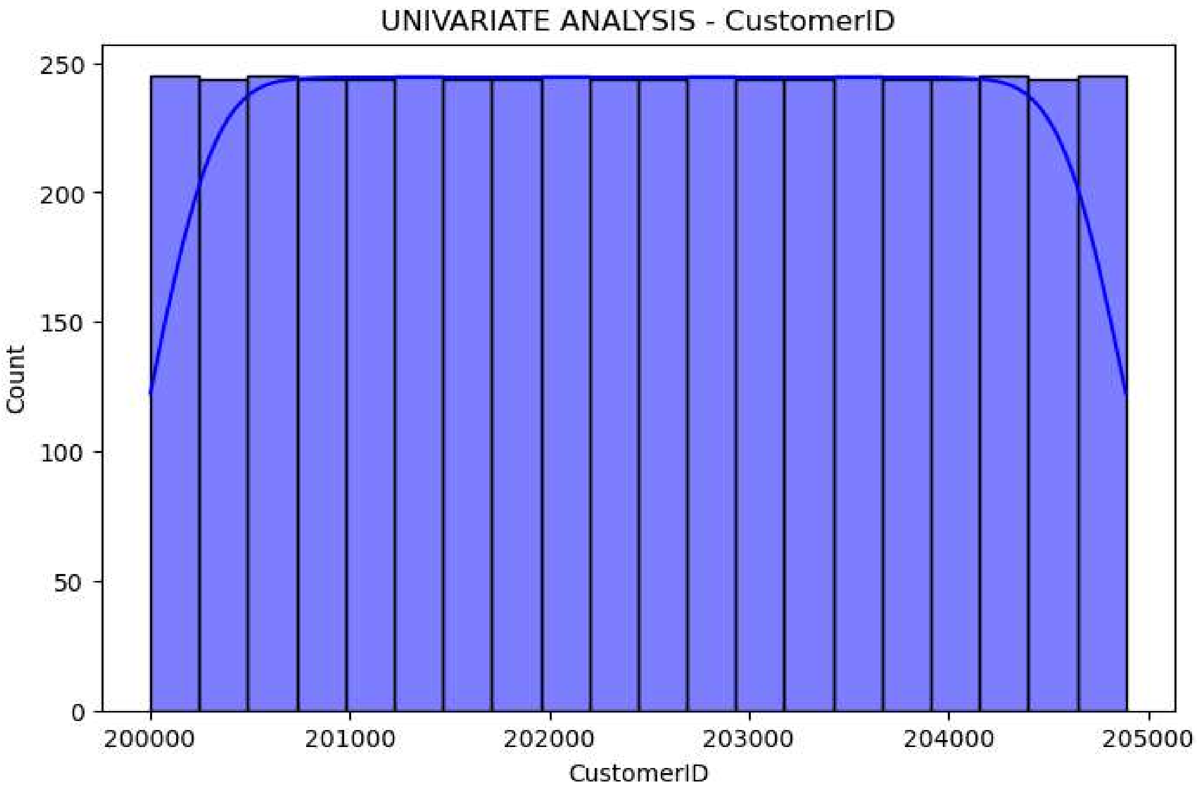


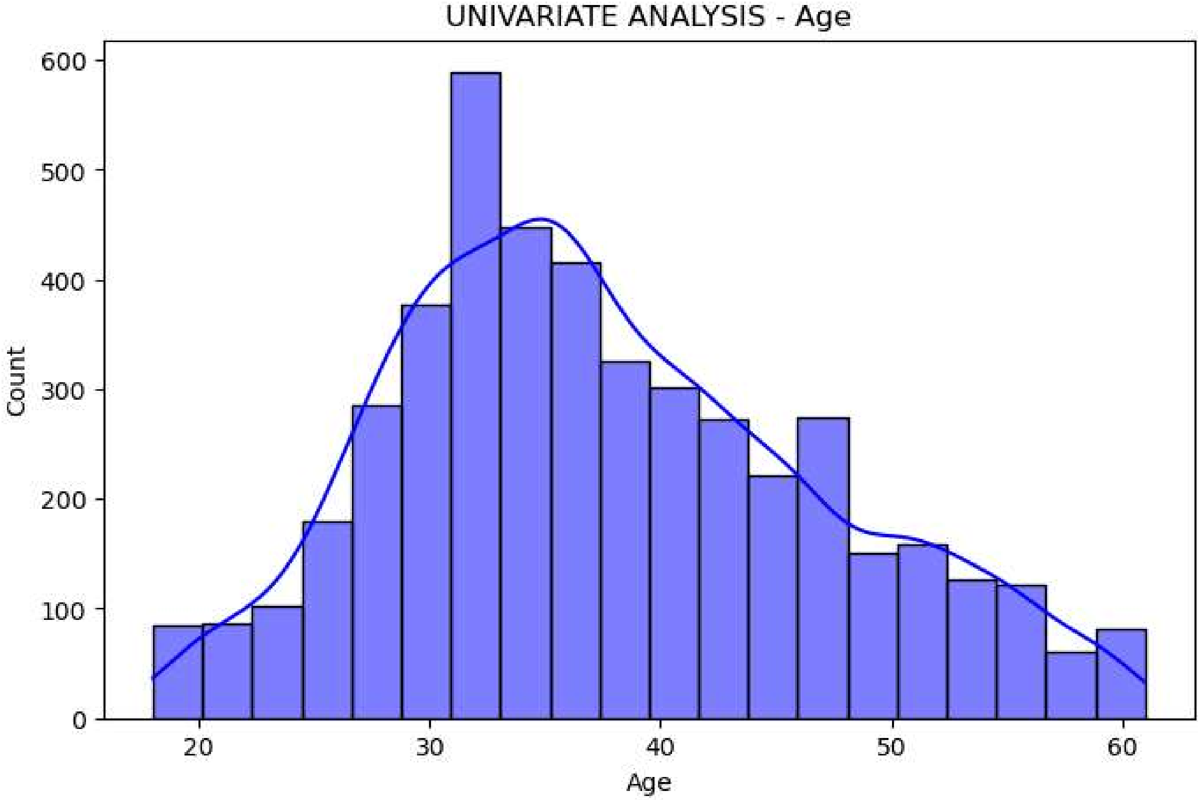


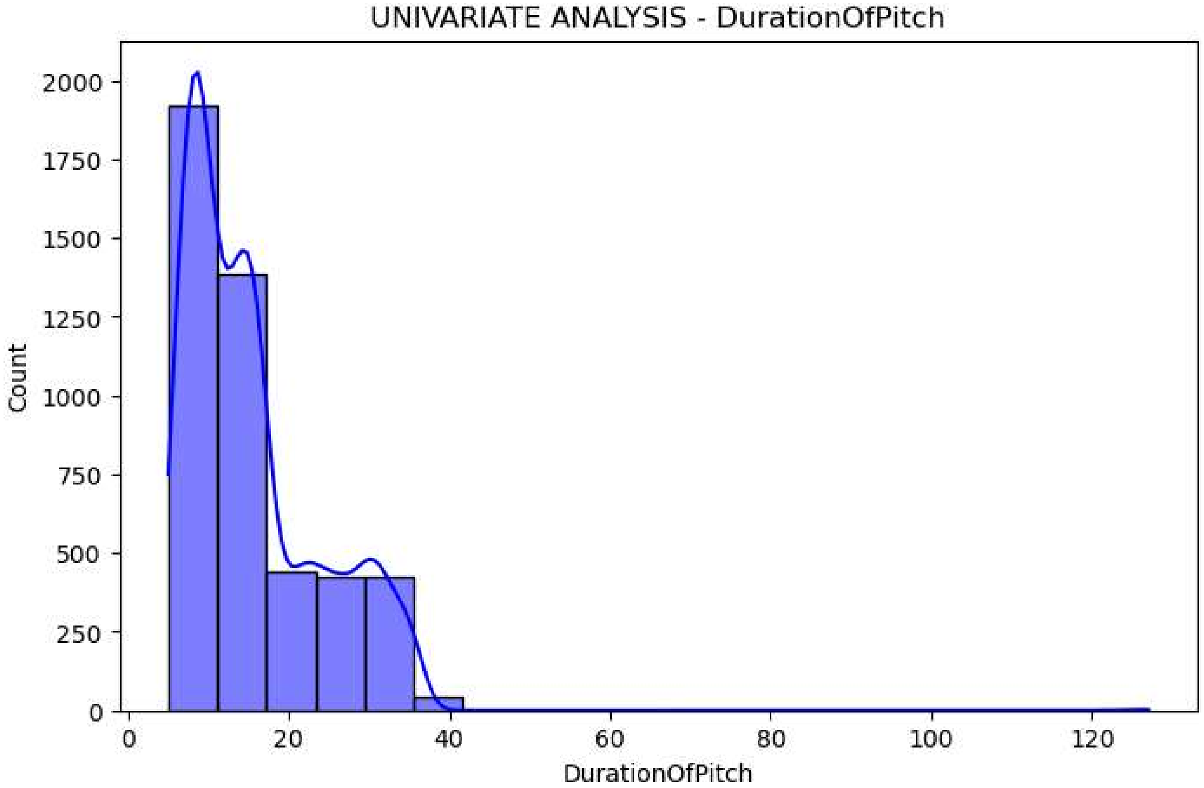
- UNIVARIATE ANALYSIS OF ALL NUMERICAL DATA

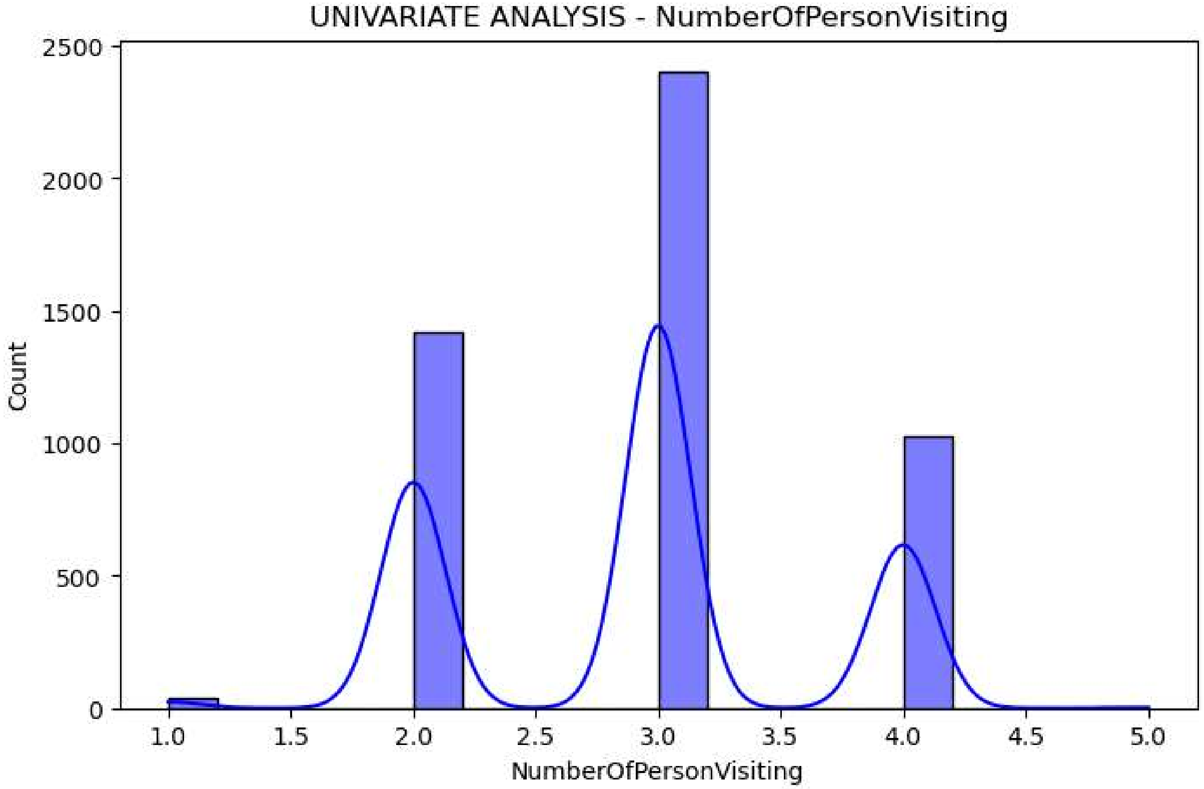
```
In [56]: for num_column in nums:
             plt.figure(figsize=(8,5))
             sns.histplot(df[num_column],color='blue',kde=True,bins=20)
```

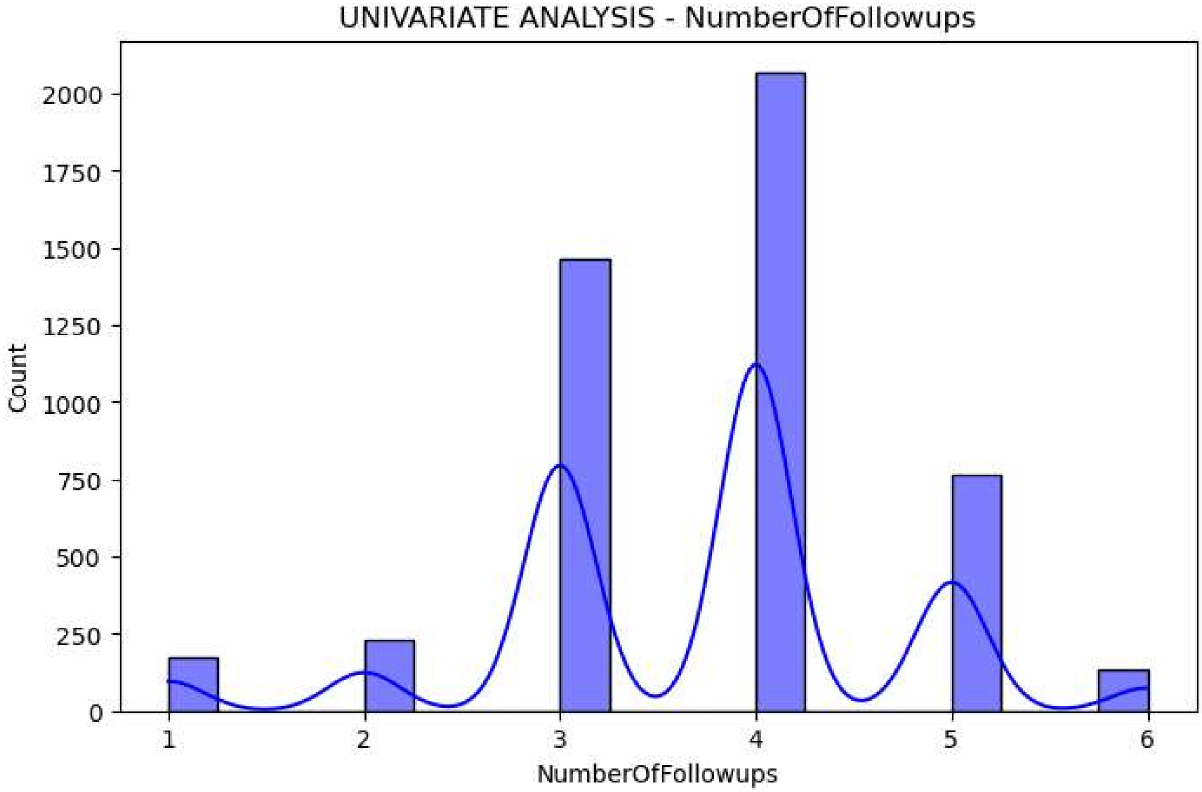
```
    plt.title(f"UNIVARIATE ANALYSIS - {num_column}")
    plt.show()
```

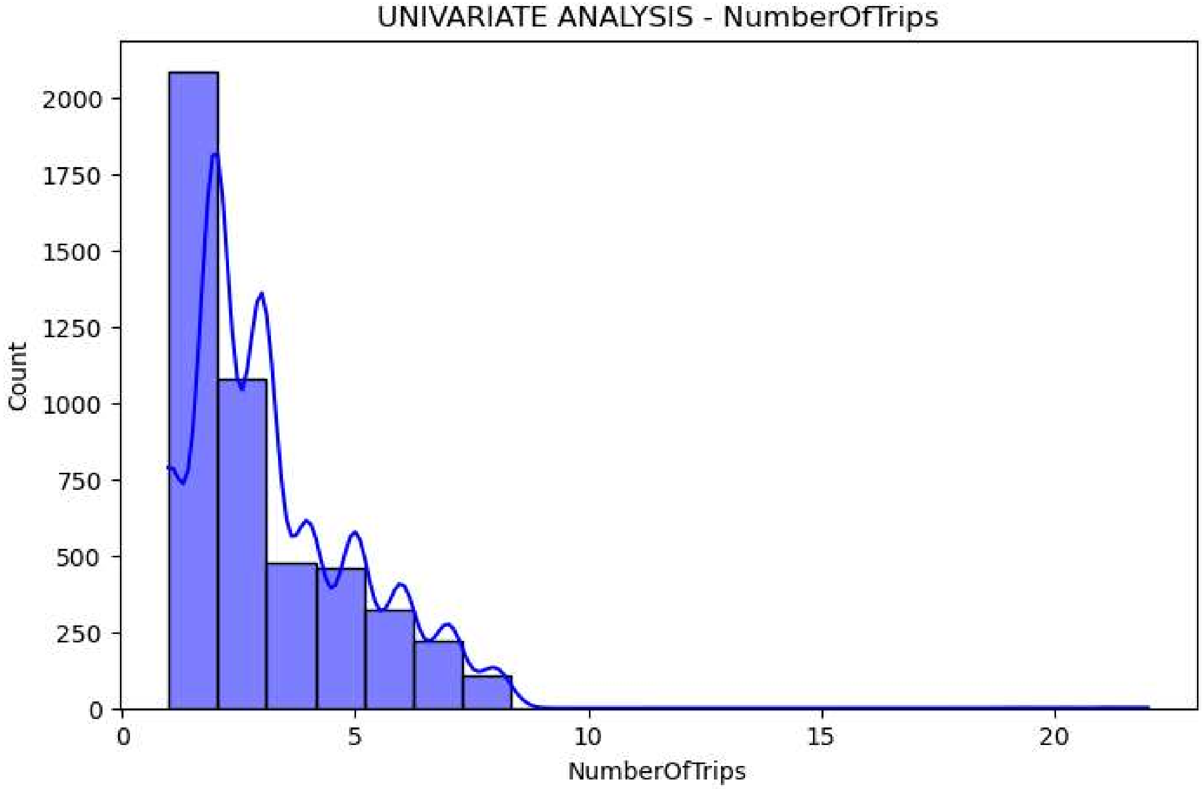


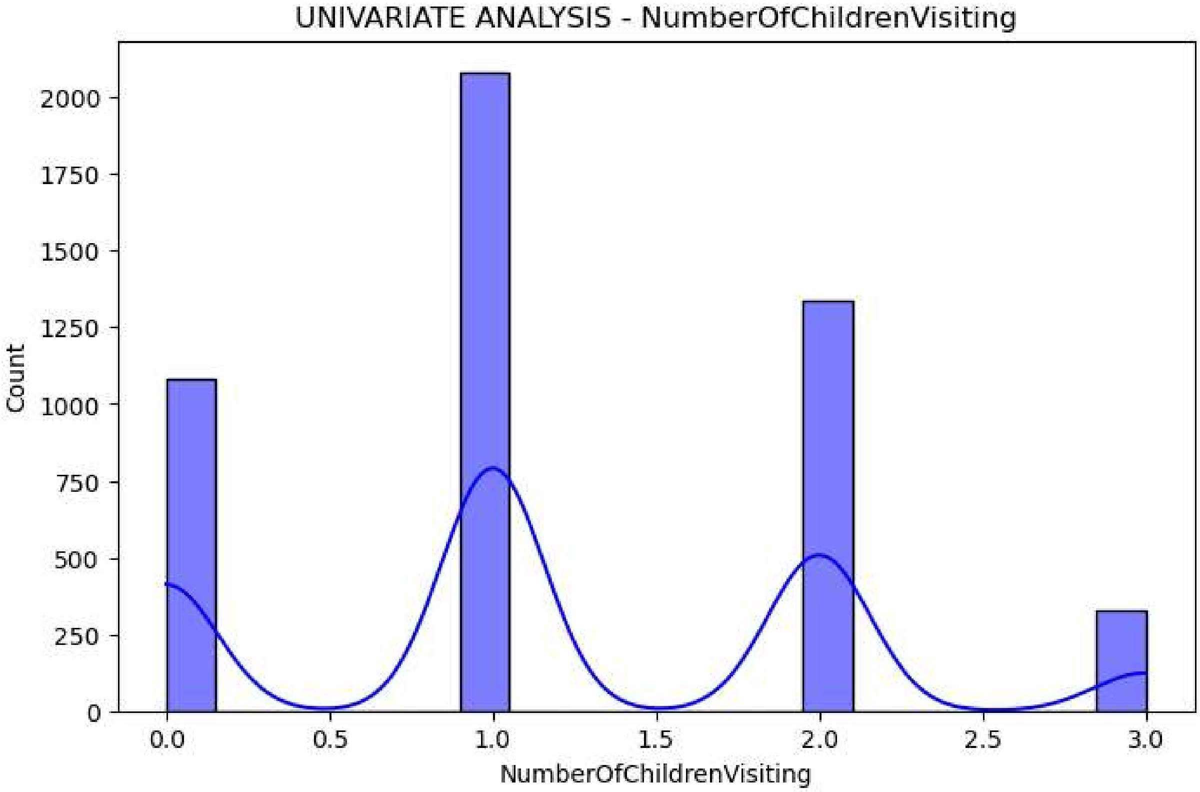


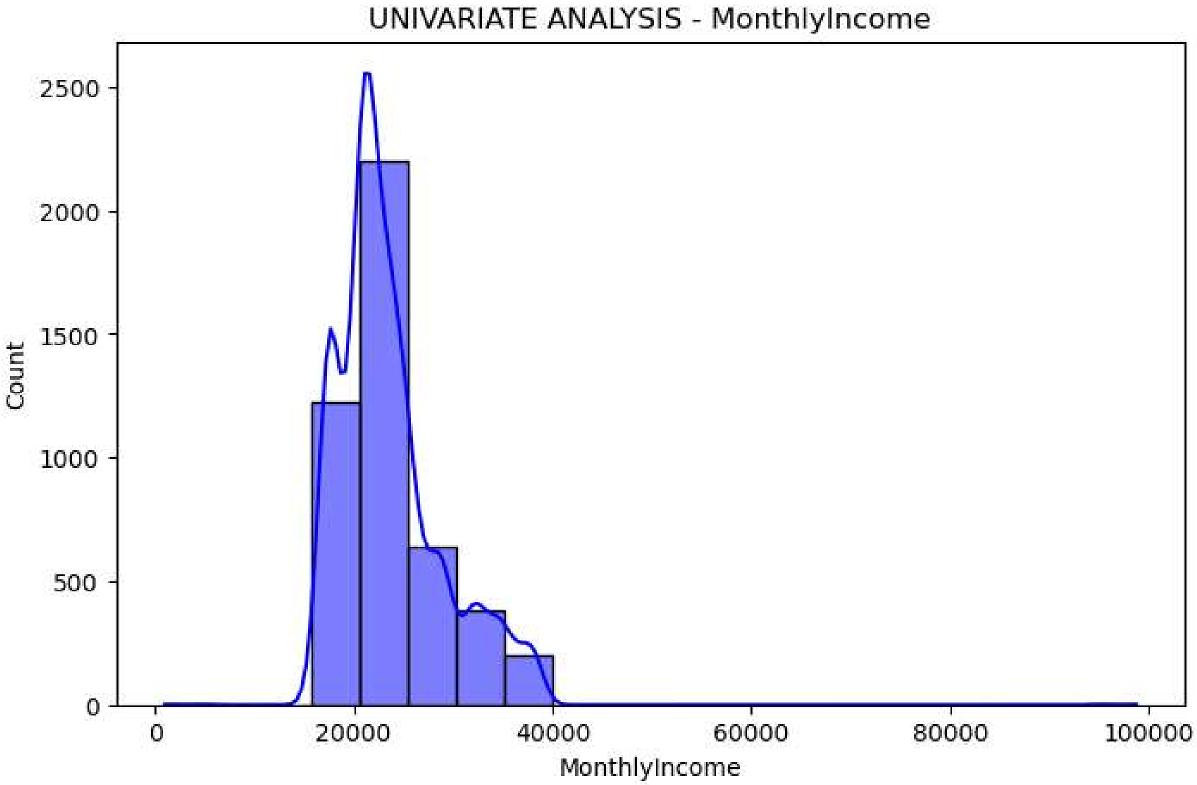

UNIVARIATE ANALYSIS - DurationOfPitch
Count
0
250
500
750
1000
1250
1500
1750
2000
0
20
40
60
80
100
120
DurationOfPitch


UNIVARIATE ANALYSIS - NumberOfPersonVisiting
Count
0
500
1000
1500
2000
2500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
NumberOfPersonVisiting


UNIVARIATE ANALYSIS - NumberOfFollowups
Count
0
250
500
750
1000
1250
1500
1750
2000
1
2
3
4
5
6
NumberOfFollowups


UNIVARIATE ANALYSIS - NumberOfTrips
Count
0
250
500
750
1000
1250
1500
1750
2000
0
5
10
15
20
NumberOfTrips





INSIGHTS :-

- PEOPLE FROM AGE GROUP 30-40 ARE TRAVELLING MORE
- MAXINUM NO. OF TRIPS ARE MORE THAN 2000

- MORE THAN 2000 PEOPLE ARE TAKING MONTHLY INCOME OF 20000+

BIVARIATE ANALYSIS

In [77]:

```
nums
```

Out[77]:

```
['CustomerID',
 'Age',
 'DurationOfPitch',
 'NumberOfPersonVisiting',
 'NumberOfFollowups',
 'NumberOfTrips',
 'NumberOfChildrenVisiting',
 'MonthlyIncome']
```

In [79]:

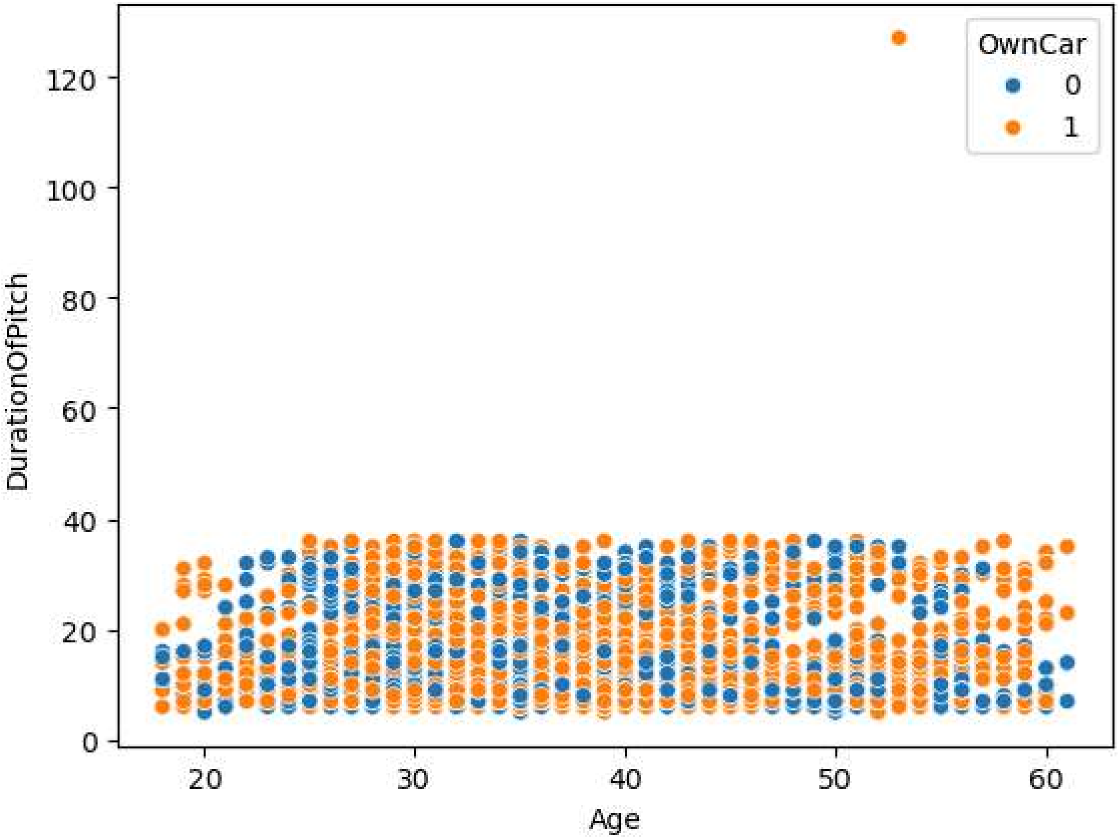
```
cats
```

Out[79]:

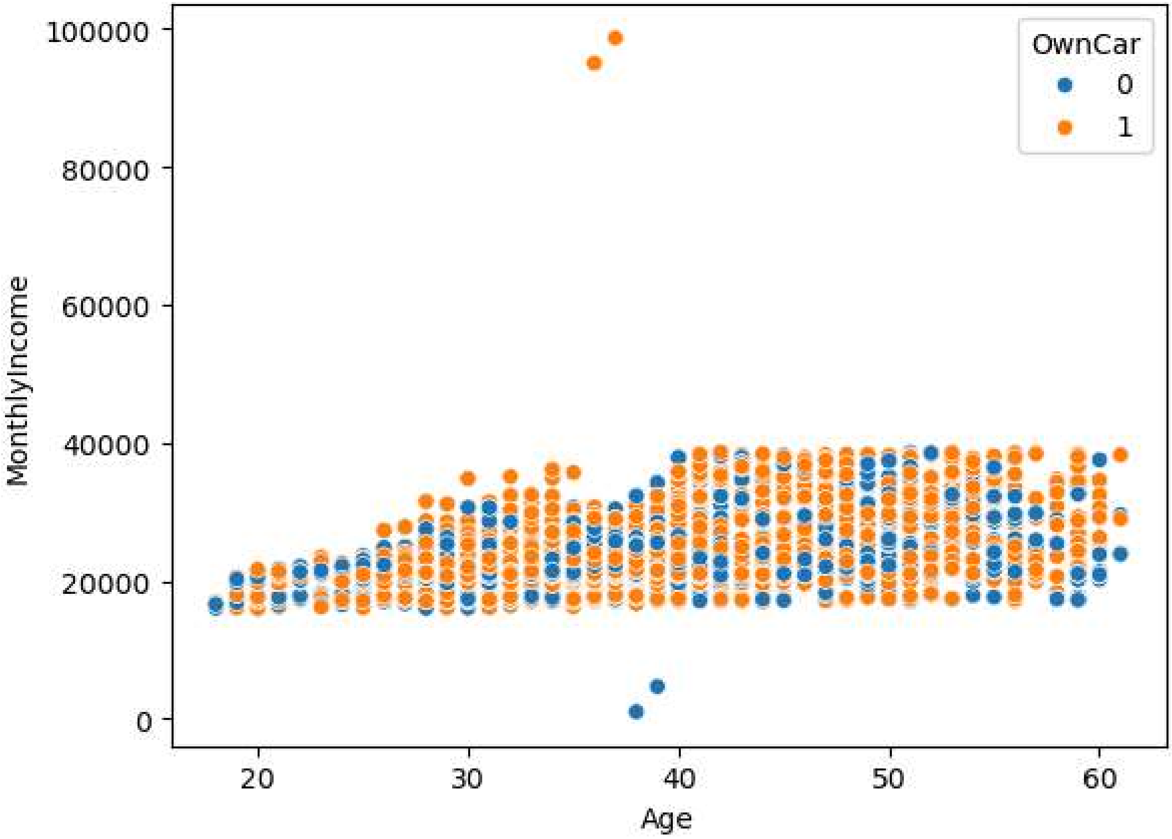
```
['ProdTaken',
 'CityTier',
 'PreferredPropertyStar',
 'Passport',
 'OwnCar',
 'PitchSatisfactionScore',
 'TypeofContact',
 'Occupation',
 'Gender',
 'ProductPitched',
 'MaritalStatus',
 'Designation']
```

In [108…

```
sns.scatterplot(x='Age',y= 'DurationOfPitch', data=df,hue='OwnCar' )
plt.show()
```

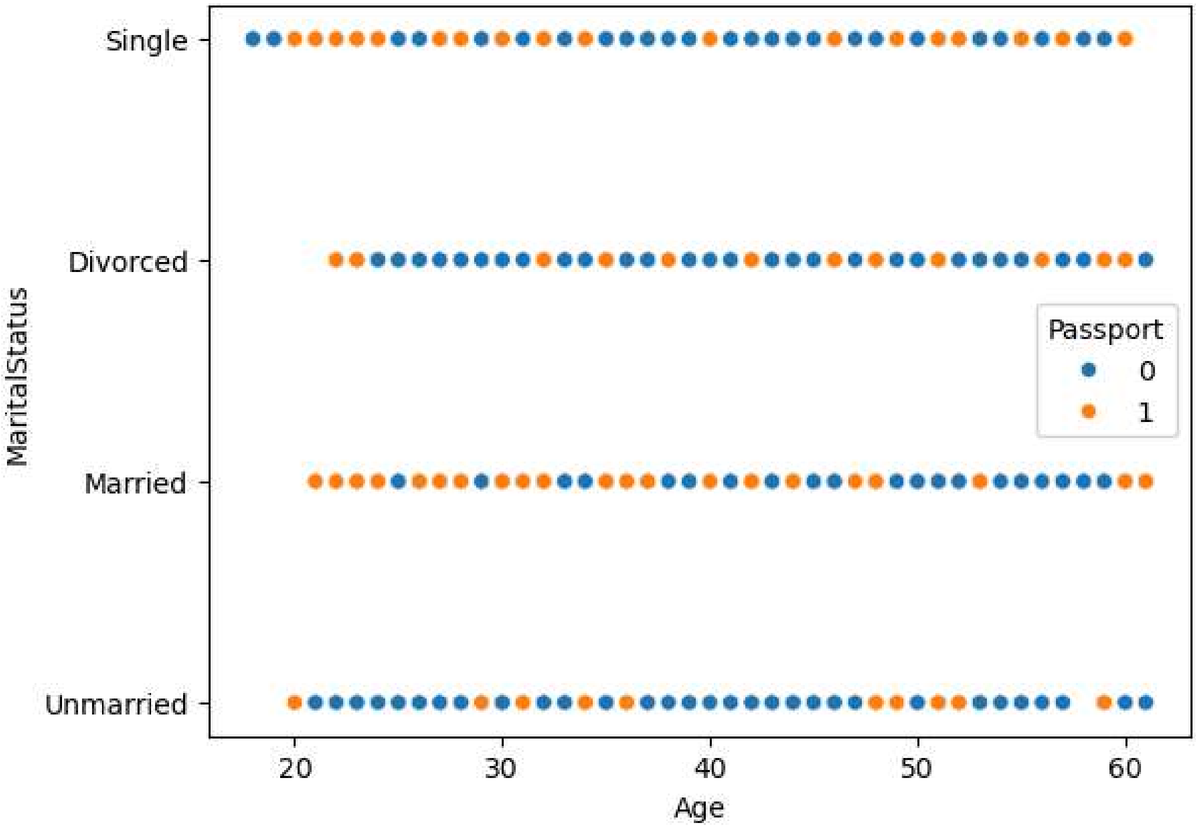


```
In [99]: sns.scatterplot(x='Age',y= 'MonthlyIncome', data=df,hue='OwnCar' )
         plt.show()
```



In [103…

```python
sns.scatterplot(x='Age',y= 'MaritalStatus', data=df , hue="Passport")
plt.show()
```



- FOR CATEGORICAL DATA :- WE USE CROSSTAB IN THIS CASE

In [149…

```python
# STACKED BAR CHART BETWEEN MARITAL STATUS ANS PRODUCT TAKEN
cross_tab = pd.crosstab(df['MaritalStatus'],df['ProdTaken'],normalize='index') # No
```

In [151…

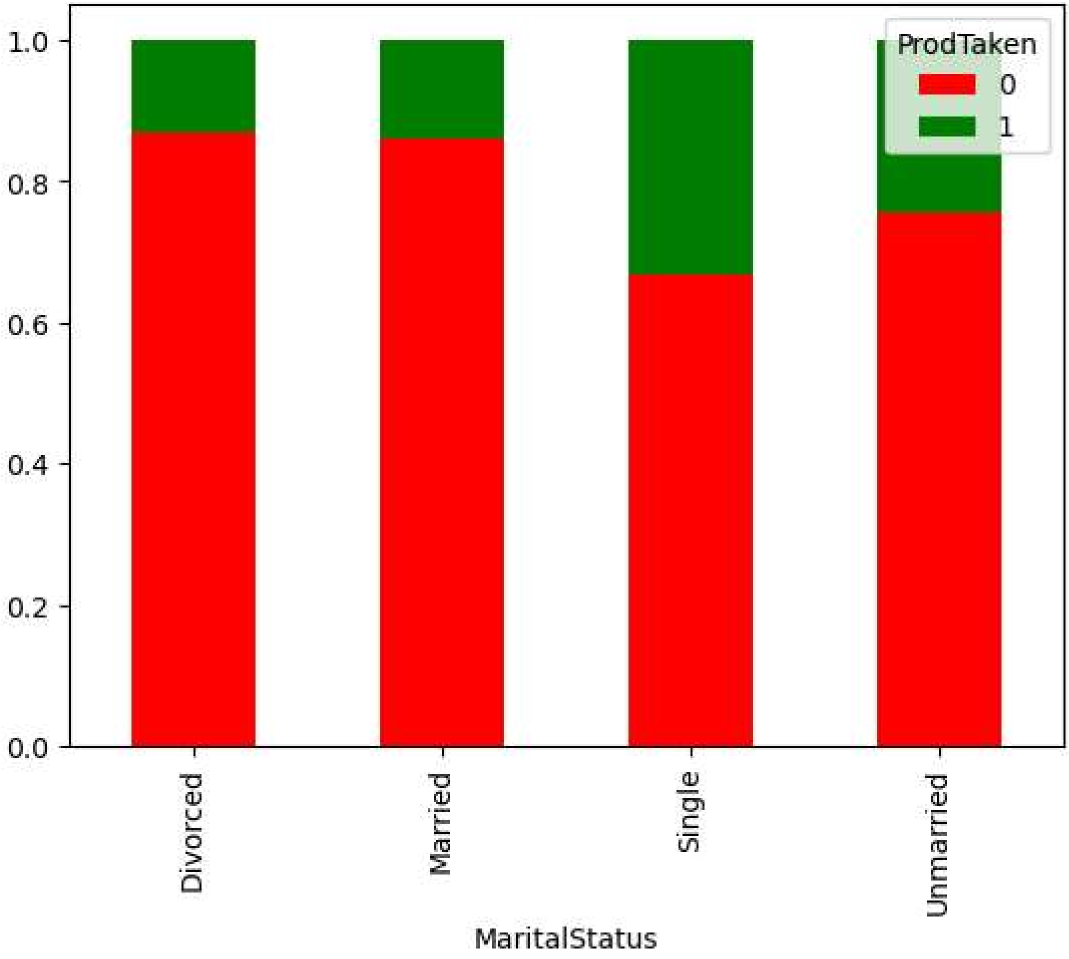
```python
cross_tab
```

Out[151…

ProdTaken	0	1
MaritalStatus		
Divorced	0.869474	0.130526
Married	0.860684	0.139316
Single	0.668122	0.331878
Unmarried	0.756598	0.243402

In [157…

```python
cross_tab.plot(kind="bar", stacked=True,color=['red','green'])
plt.show()
```



In [159…

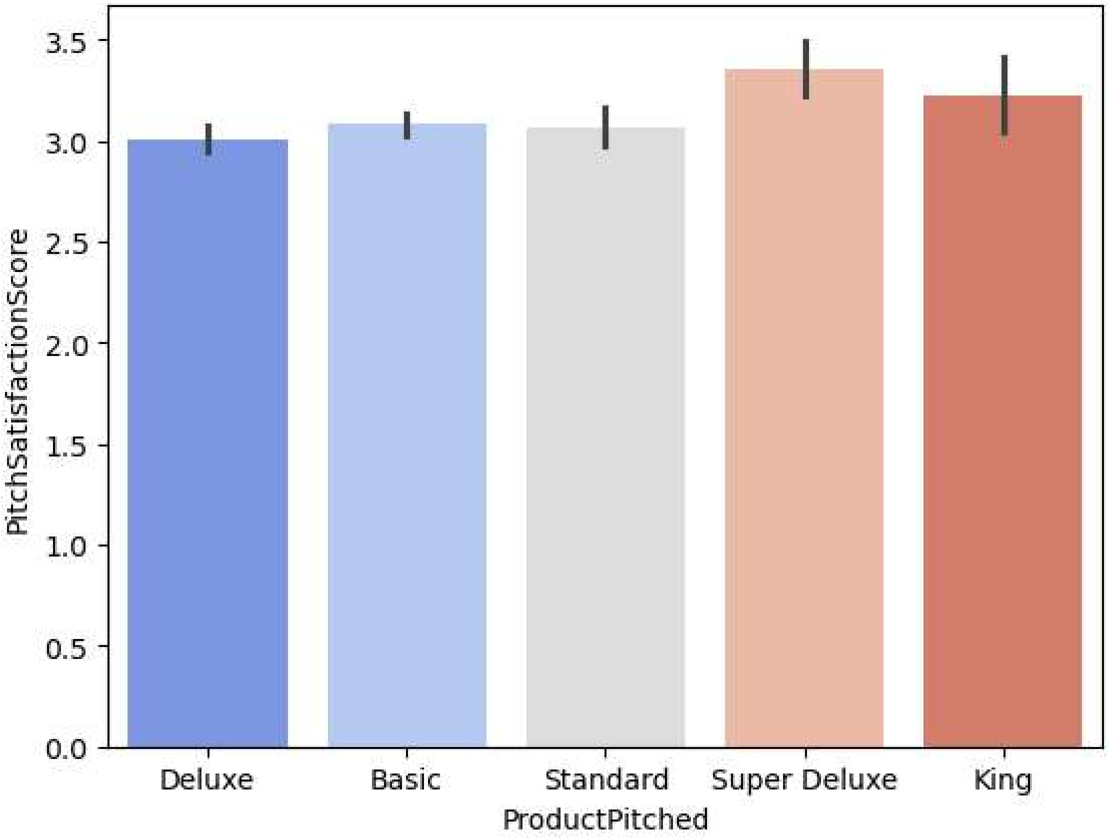
```
cats
```

Out[159…

```
['ProdTaken',
 'CityTier',
 'PreferredPropertyStar',
 'Passport',
 'OwnCar',
 'PitchSatisfactionScore',
 'TypeofContact',
 'Occupation',
 'Gender',
 'ProductPitched',
 'MaritalStatus',
 'Designation']
```

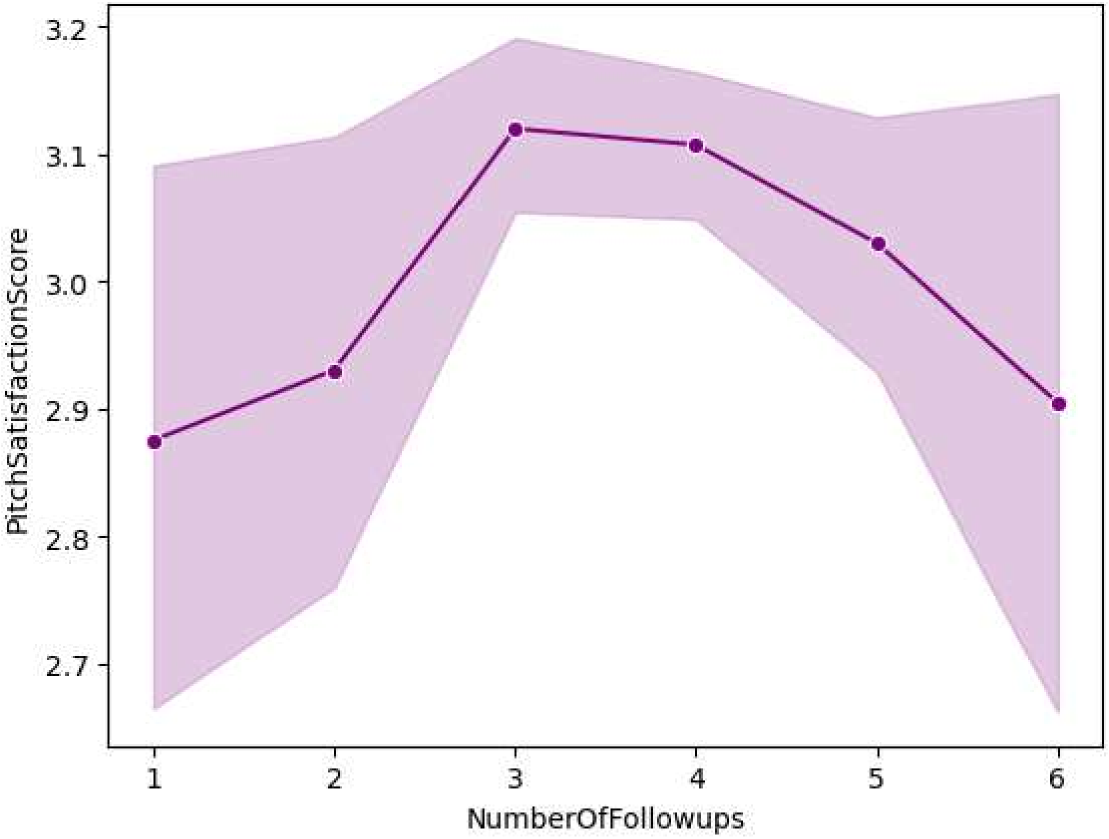
In [171…

```
sns.barplot(x='ProductPitched' , y='PitchSatisfactionScore', data=df,palette='coolw
plt.show()
```



In [175…

```
sns.lineplot(x = 'NumberOfFollowups', y = 'PitchSatisfactionScore', data=df, marker
plt.show()
```


PitchSatisfactionScore
NumberOfFollowups