```
In [1]:   import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import seaborn as sn
          %matplotlib inline
```

```
In [2]:   incomeIndex = pd.read_csv(r'C:\Users\Adetola Fagbule\Documents\bluetooth\Inc_Exp_Data.csv')
```
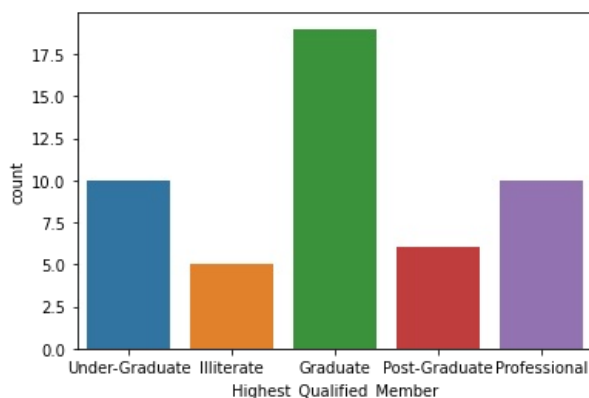
```
In [3]:   incomeIndex.head()
```

Out[3]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Member	No_of_Earning_
0	5000	8000	3	2000	64200	Under-Graduate	
1	6000	7000	2	3000	79920	Illiterate	
2	10000	4500	2	0	112800	Under-Graduate	
3	10000	2000	1	0	97200	Illiterate	
4	12500	12000	2	3000	147000	Graduate	

```
In [28]:   sn.countplot(x ='Highest_Qualified_Member', data =incomeIndex )
```

Out[28]:   <AxesSubplot:xlabel='Highest_Qualified_Member', ylabel='count'>


```
In [23]:   topAnnual = incomeIndex.sort_values('Annual_HH_Income', ascending = False)
```

```
In [27]:   ### Top 5 Household with highest annual income
           topAnnual.iloc[0:5]
```

Out[27]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Member	No_of_Earning
47	100000	30000	6	0	1404000	Graduate	
49	100000	40000	6	10000	1320000	Post-Graduate	
46	98000	25000	5	0	1152480	Professional	
44	85000	25000	5	0	1142400	Under-Graduate	
43	80000	20000	4	0	1075200	Graduate	

```
In [5]:   ###average hh income of highest qualified member, therefore the higher the qualification the higher the income.
          incomeIndex.groupby('Highest_Qualified_Member').mean()['Annual_HH_Income']
```
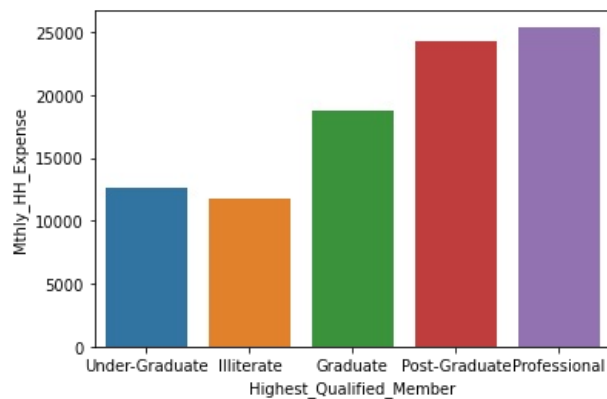
```
Out[5]:   Highest_Qualified_Member
          Graduate          499042.105263
          Illiterate        279278.400000
          Post-Graduate     668500.000000
          Professional      599520.000000
          Under-Graduate    361656.000000
          Name: Annual_HH_Income, dtype: float64
```

```
In [36]:   sn.barplot(x = 'Highest_Qualified_Member',y = 'Mthly_HH_Expense', data =incomeIndex, ci = False )
```

```
Out[36]:   <AxesSubplot:xlabel='Highest_Qualified_Member', ylabel='Mthly_HH_Expense'>
```



```
In [7]:    ### average number of family members by average Monthly hh expense showing that the higher the members'number the
           incomeIndex.groupby('No_of_Fly_Members').mean()['Mthly_HH_Expense']
```

```
Out[7]:    No_of_Fly_Members
           1      2000.000000
           2      8262.500000
           3     14255.555556
           4     17800.000000
           5     24000.000000
           6     29450.000000
           7     31500.000000
           Name: Mthly_HH_Expense, dtype: float64
```

```
In [37]:   sn.barplot(x = 'No_of_Fly_Members',y = 'Mthly_HH_Expense', data =incomeIndex, ci = False)
```

```
Out[37]:   <AxesSubplot:xlabel='No_of_Fly_Members', ylabel='Mthly_HH_Expense'>
```
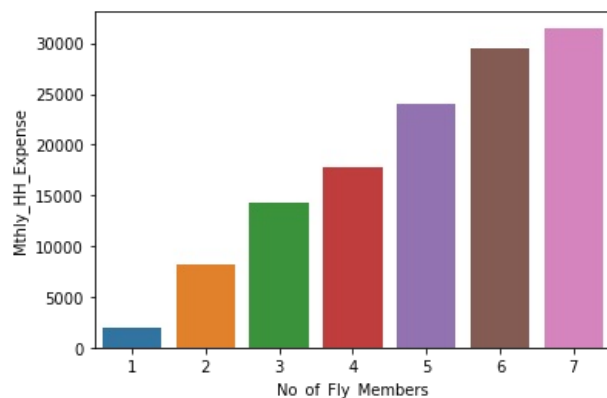


```
In [12]:   incomeIndex['Highest_Qualified_Member']
```

```
Out[12]:   0       Under-Graduate
           1           Illiterate
           2       Under-Graduate
           3           Illiterate
           4             Graduate
           5             Graduate
           6        Post-Graduate
           7             Graduate
           8       Under-Graduate
           9       Under-Graduate
           10      Under-Graduate
           11          Illiterate
           12          Illiterate
           13            Graduate
           14            Graduate
           15            Graduate
           16            Graduate
           17      Under-Graduate
           18            Graduate
           19            Graduate
           20      Under-Graduate
```

```
21       Professional
22       Professional
23       Professional
24           Graduate
25       Professional
26     Under-Graduate
27     Under-Graduate
28           Graduate
29           Graduate
30           Graduate
31       Professional
32      Post-Graduate
33      Post-Graduate
34           Graduate
35       Professional
36       Professional
37       Professional
38           Graduate
39      Post-Graduate
40           Graduate
41         Illiterate
42           Graduate
43           Graduate
44     Under-Graduate
45      Post-Graduate
46       Professional
47           Graduate
48       Professional
49      Post-Graduate
Name: Highest_Qualified_Member, dtype: object
```

In [14]:
```python
incomeIndex['Highest_Qualified_Member']=='Graduate'
```

Out[14]:
```
0      False
1      False
2      False
3      False
4       True
5       True
6      False
7       True
8      False
9      False
10     False
11     False
12     False
13      True
14      True
15      True
16      True
17     False
18      True
19      True
20     False
21     False
22     False
23     False
24      True
25     False
26     False
27     False
28      True
29      True
30      True
31     False
32     False
33     False
34      True
35     False
36     False
37     False
38      True
39     False
40      True
41     False
42      True
43      True
44     False
45     False
46     False
47      True
48     False
49     False
Name: Highest_Qualified_Member, dtype: bool
```

```
In [18]:    graduateIncome = incomeIndex.loc[incomeIndex['Highest_Qualified_Member']=='Graduate', :]
```

```
In [25]:    top5 = graduateIncome.sort_values('Mthly_HH_Income', ascending = False)
```

```
In [26]:    #### top 5 households' income with graduates as highest qualified
            top5.iloc[0:5]
```

Out[26]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Member	No_of_Earning
47	100000	30000	6	0	1404000	Graduate	
43	80000	20000	4	0	1075200	Graduate	
42	70000	9000	2	0	756000	Graduate	
40	60000	50000	6	10000	590400	Graduate	
38	55000	45000	6	12000	600600	Graduate	

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
In [18]:    graduateIncome = incomeIndex.loc[incomeIndex['Highest_Qualified_Member']=='Graduate', :]
```

```
In [25]:    top5 = graduateIncome.sort_values('Mthly_HH_Income', ascending = False)
```

```
In [26]:    #### top 5 households' income with graduates as highest qualified
            top5.iloc[0:5]
```

Out[26]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Member	No_of_Earning