In [1]:

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

In [2]:

```
income_df=pd.read_csv(r'C:\Users\Acer\Downloads\Inc_Exp_Data\Inc_Exp_Data.csv')
```

In [3]:

```
income_df
```

Out[3]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annu
0	5000	8000	3	2000	
1	6000	7000	2	3000	
2	10000	4500	2	0	
3	10000	2000	1	0	
4	12500	12000	2	3000	
5	14000	8000	2	0	
6	15000	16000	3	35000	
7	18000	20000	5	8000	
8	19000	9000	2	0	
9	20000	9000	4	0	
10	20000	18000	4	8000	
11	22000	25000	6	12000	
12	23400	5000	3	0	
13	24000	10500	6	0	
14	24000	10000	4	0	
15	25000	12300	3	0	
16	25000	20000	3	3500	
17	25000	10000	6	0	
18	29000	6600	2	2000	
19	30000	13000	4	0	
20	30500	25000	5	5000	
21	32000	15000	4	0	
22	34000	19000	6	0	
23	34000	25000	3	4000	
24	35000	12000	3	0	
25	35000	25000	4	0	
26	39000	8000	4	0	
27	40000	10000	4	0	
28	42000	15000	4	0	
29	43000	12000	4	0	
30	45000	25000	6	0	
31	45000	40000	6	3500	
32	45000	10000	2	1000	

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annu
33	45000	22000	4	2500	
34	46000	25000	5	3500	
35	47000	15000	7	0	
36	50000	20000	4	0	
37	50500	20000	3	0	
38	55000	45000	6	12000	
39	60000	10000	3	0	
40	60000	50000	6	10000	
41	65000	20000	4	5000	
42	70000	9000	2	0	
43	80000	20000	4	0	
44	85000	25000	5	0	
45	90000	48000	7	0	
46	98000	25000	5	0	
47	100000	30000	6	0	
48	100000	50000	4	20000	
49	100000	40000	6	10000	

```
In [4]: income_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Mthly_HH_Income          50 non-null     int64 
 1   Mthly_HH_Expense         50 non-null     int64 
 2   No_of_Fly_Members        50 non-null     int64 
 3   Emi_or_Rent_Amt          50 non-null     int64 
 4   Annual_HH_Income         50 non-null     int64 
 5   Highest_Qualified_Member 50 non-null     object
 6   No_of_Earning_Members    50 non-null     int64 
dtypes: int64(6), object(1)
memory usage: 2.9+ KB
```

```
In [5]: income_df.shape
```

```
Out[5]: (50, 7)
```

```
In [6]: income_df.columns
```

```
Out[6]:  Index(['Mthly_HH_Income', 'Mthly_HH_Expense', 'No_of_Fly_Members',
               'Emi_or_Rent_Amt', 'Annual_HH_Income', 'Highest_Qualified_Member',
               'No_of_Earning_Members'],
              dtype='object')
```

```
In [7]: income_df.describe
```

Out[7]:

```
<bound method NDFrame.describe of     Mthly_HH_Income  Mthly_HH_Expense  No_of_
Fly_Members  Emi_or_Rent_Amt  \
0              5000              8000                  3             2000
1              6000              7000                  2             3000
2             10000              4500                  2                0
3             10000              2000                  1                0
4             12500             12000                  2             3000
5             14000              8000                  2                0
6             15000             16000                  3            35000
7             18000             20000                  5             8000
8             19000              9000                  2                0
9             20000              9000                  4                0
10            20000             18000                  4             8000
11            22000             25000                  6            12000
12            23400              5000                  3                0
13            24000             10500                  6                0
14            24000             10000                  4                0
15            25000             12300                  3                0
16            25000             20000                  3             3500
17            25000             10000                  6                0
18            29000              6600                  2             2000
19            30000             13000                  4                0
20            30500             25000                  5             5000
21            32000             15000                  4                0
22            34000             19000                  6                0
23            34000             25000                  3             4000
24            35000             12000                  3                0
25            35000             25000                  4                0
26            39000              8000                  4                0
27            40000             10000                  4                0
28            42000             15000                  4                0
29            43000             12000                  4                0
30            45000             25000                  6                0
31            45000             40000                  6             3500
32            45000             10000                  2             1000
33            45000             22000                  4             2500
34            46000             25000                  5             3500
35            47000             15000                  7                0
36            50000             20000                  4                0
37            50500             20000                  3                0
38            55000             45000                  6            12000
39            60000             10000                  3                0
40            60000             50000                  6            10000
41            65000             20000                  4             5000
42            70000              9000                  2                0
43            80000             20000                  4                0
44            85000             25000                  5                0
45            90000             48000                  7                0
46            98000             25000                  5                0
47           100000             30000                  6                0
48           100000             50000                  4            20000
49           100000             40000                  6            10000

    Annual_HH_Income Highest_Qualified_Member  No_of_Earning_Members
0              64200           Under-Graduate                      1
1              79920               Illiterate                      1
2             112800           Under-Graduate                      1
3              97200               Illiterate                      1
4             147000                 Graduate                      1
5             196560                 Graduate                      1
```

```
6             167400            Post-Graduate                      1
7             216000                 Graduate                      1
8             218880           Under-Graduate                      1
9             220800           Under-Graduate                      2
10            278400           Under-Graduate                      2
11            279840               Illiterate                      1
12            292032               Illiterate                      1
13            316800                 Graduate                      2
14            244800                 Graduate                      2
15            246000                 Graduate                      1
16            261000                 Graduate                      1
17            258000           Under-Graduate                      3
18            348000                 Graduate                      1
19            385200                 Graduate                      1
20            351360           Under-Graduate                      1
21            445440             Professional                      1
22            330480             Professional                      1
23            469200             Professional                      1
24            466200                 Graduate                      1
25            449400             Professional                      2
26            556920           Under-Graduate                      1
27            412800           Under-Graduate                      1
28            488880                 Graduate                      1
29            619200                 Graduate                      1
30            523800                 Graduate                      3
31            507600             Professional                      2
32            437400            Post-Graduate                      1
33            610200            Post-Graduate                      1
34            596160                 Graduate                      1
35            456840             Professional                      4
36            570000             Professional                      1
37            581760             Professional                      2
38            600600                 Graduate                      2
39            590400            Post-Graduate                      1
40            590400                 Graduate                      1
41            647400               Illiterate                      2
42            756000                 Graduate                      1
43           1075200                 Graduate                      1
44           1142400           Under-Graduate                      2
45            885600            Post-Graduate                      3
46           1152480             Professional                      2
47           1404000                 Graduate                      3
48           1032000             Professional                      2
49           1320000            Post-Graduate                      1  >
```

In [8]:

```
income_df.describe()
```

Out[8]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Ar
count	50.000000	50.000000	50.000000	50.000000	
mean	41558.000000	18818.000000	4.060000	3060.000000	
std	26097.908979	12090.216824	1.517382	6241.434948	
min	5000.000000	2000.000000	1.000000	0.000000	
25%	23550.000000	10000.000000	3.000000	0.000000	
50%	35000.000000	15500.000000	4.000000	0.000000	
75%	50375.000000	25000.000000	5.000000	3500.000000	
max	100000.000000	50000.000000	7.000000	35000.000000	

In [9]:

```
income_df.describe(include='all')
```

Out[9]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	A
count	50.000000	50.000000	50.000000	50.000000	
unique	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	
mean	41558.000000	18818.000000	4.060000	3060.000000	
std	26097.908979	12090.216824	1.517382	6241.434948	
min	5000.000000	2000.000000	1.000000	0.000000	
25%	23550.000000	10000.000000	3.000000	0.000000	
50%	35000.000000	15500.000000	4.000000	0.000000	
75%	50375.000000	25000.000000	5.000000	3500.000000	
max	100000.000000	50000.000000	7.000000	35000.000000	

In [11]:

```
#Transpose to convert the rows into columns
income_df.describe().T
```

Out[11]:

	count	mean	std	min	25%	50%
Mthly_HH_Income	50.0	41558.00	26097.908979	5000.0	23550.0	35000.0
Mthly_HH_Expense	50.0	18818.00	12090.216824	2000.0	10000.0	15500.0
No_of_Fly_Members	50.0	4.06	1.517382	1.0	3.0	4.0
Emi_or_Rent_Amt	50.0	3060.00	6241.434948	0.0	0.0	0.0
Annual_HH_Income	50.0	490019.04	320135.792123	64200.0	258750.0	447420.0
No_of_Earning_Members	50.0	1.46	0.734291	1.0	1.0	1.0

In [12]:

```
income_df.isna()
```

Out[12]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annu
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	False	False	False	False	
4	False	False	False	False	
5	False	False	False	False	
6	False	False	False	False	
7	False	False	False	False	
8	False	False	False	False	
9	False	False	False	False	
10	False	False	False	False	
11	False	False	False	False	
12	False	False	False	False	
13	False	False	False	False	
14	False	False	False	False	
15	False	False	False	False	
16	False	False	False	False	
17	False	False	False	False	
18	False	False	False	False	
19	False	False	False	False	
20	False	False	False	False	
21	False	False	False	False	
22	False	False	False	False	
23	False	False	False	False	
24	False	False	False	False	
25	False	False	False	False	
26	False	False	False	False	
27	False	False	False	False	
28	False	False	False	False	
29	False	False	False	False	
30	False	False	False	False	
31	False	False	False	False	
32	False	False	False	False	

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annu
33	False	False	False	False	
34	False	False	False	False	
35	False	False	False	False	
36	False	False	False	False	
37	False	False	False	False	
38	False	False	False	False	
39	False	False	False	False	
40	False	False	False	False	
41	False	False	False	False	
42	False	False	False	False	
43	False	False	False	False	
44	False	False	False	False	
45	False	False	False	False	
46	False	False	False	False	
47	False	False	False	False	
48	False	False	False	False	
49	False	False	False	False	

In [13]:

```
income_df.isna().any()
```

Out[13]:

```
Mthly_HH_Income            False
Mthly_HH_Expense           False
No_of_Fly_Members          False
Emi_or_Rent_Amt            False
Annual_HH_Income           False
Highest_Qualified_Member   False
No_of_Earning_Members      False
dtype: bool
```

In [16]:

```
income_df['Mthly_HH_Expense'].mean()
```

Out[16]:

```
np.float64(18818.0)
```

In [15]:

```
income_df['Mthly_HH_Expense'].median()
```

Out[15]:

```
15500.0
```

In [17]:

```
income_df['Mthly_HH_Expense'].mode()
```

Out[17]:

```
0    25000
Name: Mthly_HH_Expense, dtype: int64
```

In [21]:

```
# Create a frequency table of monthly household expenses
exp_mean = pd.crosstab(index=income_df['Mthly_HH_Expense'],columns='count')

# Convert the index to a regular column for easier manipulation
exp_mean.reset_index(inplace=True)

# Find and display the row(s) with the most frequent monthly household expense v
# This identifies the mode (most common value) of the monthly household expenses
exp_mean[exp_mean['count']==income_df.Mthly_HH_Expense.value_counts().max()]
```

Out[21]:

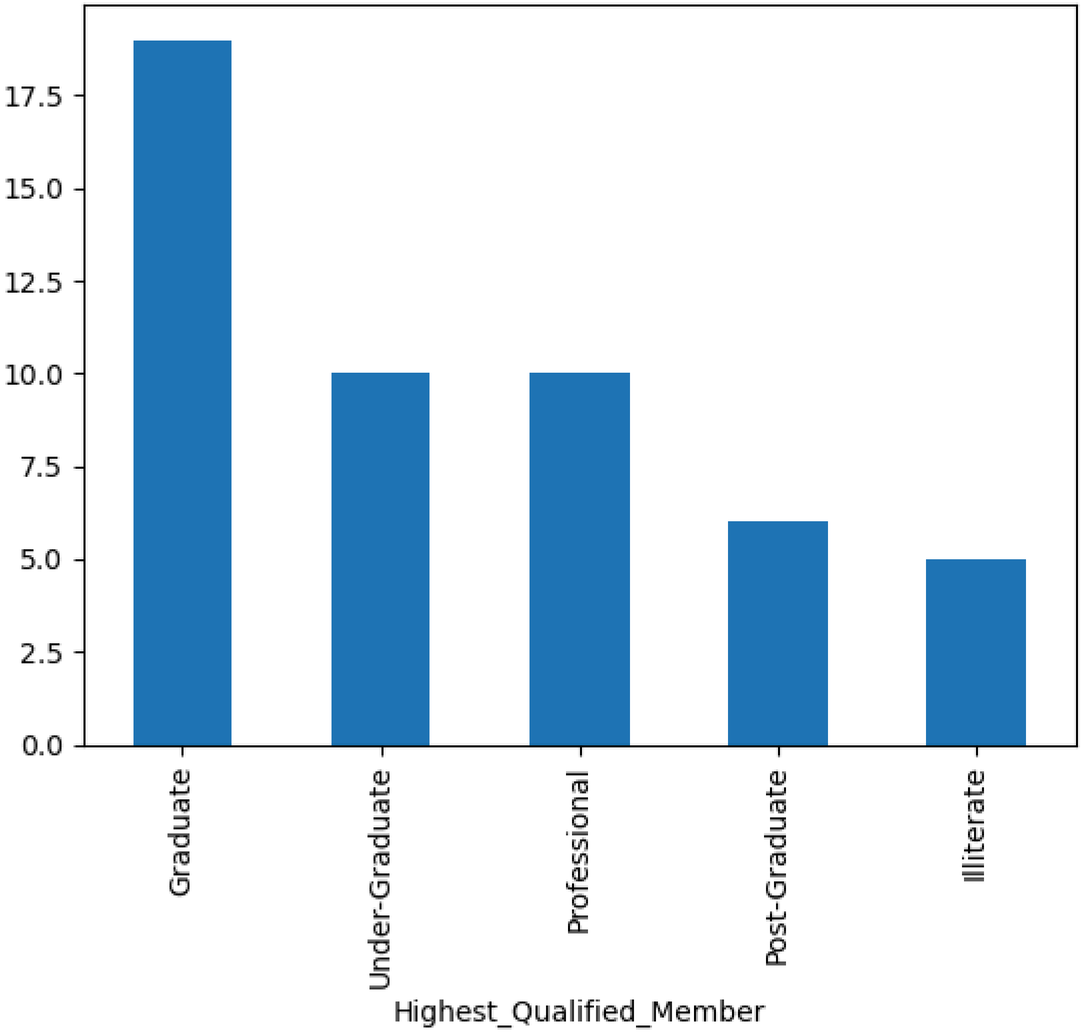
col_0	Mthly_HH_Expense	count
18	25000	8

In [22]:

```
income_df["Highest_Qualified_Member"].value_counts().plot(kind='bar')
```

Out[22]:

```
<Axes: xlabel='Highest_Qualified_Member'>
```

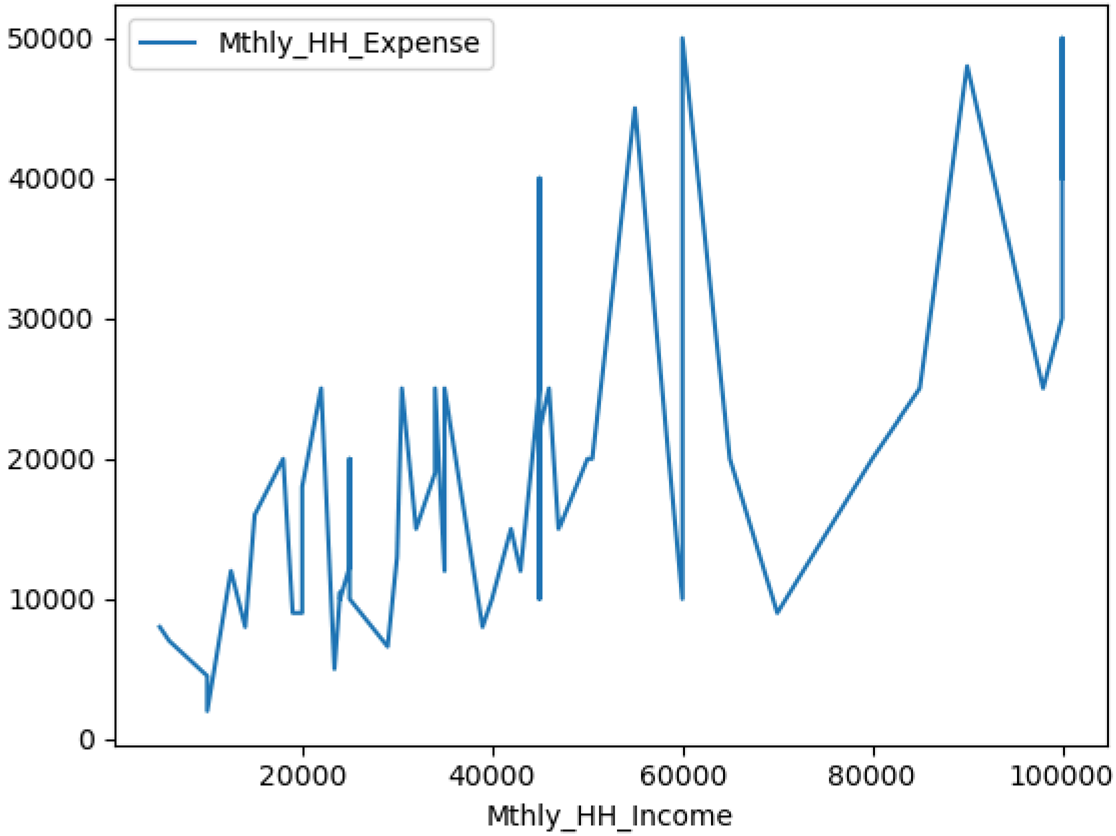


In [24]:

```
income_df.plot(x='Mthly_HH_Income', y='Mthly_HH_Expense')
```

Out[24]:

```
<Axes: xlabel='Mthly_HH_Income'>
```



```
In [25]: IQR = income_df['Mthly_HH_Expense'].quantile(0.75)-income_df['Mthly_HH_Expense']
         IQR
```

```
Out[25]: np.float64(15000.0)
```

```
In [26]: IQR1 = income_df['Mthly_HH_Expense'].quantile(0.5)-income_df['Mthly_HH_Expense']
         IQR1
```

```
Out[26]: np.float64(5500.0)
```

```
In [28]: income_df['Mthly_HH_Expense'].var()
```

```
Out[28]: 146173342.85714287
```

```
In [29]: income_df['Mthly_HH_Expense'].std()
```

```
Out[29]: 12090.216824240286
```

```
In [30]: #coefficient of covariation formula, std deviation / mean
         coef=income_df['Mthly_HH_Expense'].std() / income_df['Mthly_HH_Expense'].mean()
         coef
```

```
Out[30]: np.float64(0.6424814977277227)
```

```
In [ ]:
```