In [1]:

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

In [2]:

```
df = pd.read_csv(r"/content/Inc_Exp_Data.csv")
```

In [3]:

```
df
```

Out[3]:

	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

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt
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

```
In [4]: df.describe()
```

Out[4]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_A
count	50.000000	50.000000	50.000000	50.0000
mean	41558.000000	18818.000000	4.060000	3060.0000
std	26097.908979	12090.216824	1.517382	6241.434
min	5000.000000	2000.000000	1.000000	0.0000
25%	23550.000000	10000.000000	3.000000	0.0000
50%	35000.000000	15500.000000	4.000000	0.0000
75%	50375.000000	25000.000000	5.000000	3500.0000
max	100000.000000	50000.000000	7.000000	35000.0000

In [5]:
```
df.head()
```

Out[5]:

	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

Analyze data

In [6]:
```
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 [8]:
```
df.shape
```

Out[8]:
```
(50, 7)
```

In [9]:
```
df.describe().T
```

Out[9]:

	count	mean	std	min	25%	
Mthly_HH_Income	50.0	41558.00	26097.908979	5000.0	23550.0	35
Mthly_HH_Expense	50.0	18818.00	12090.216824	2000.0	10000.0	15
No_of_Fly_Members	50.0	4.06	1.517382	1.0	3.0	
Emi_or_Rent_Amt	50.0	3060.00	6241.434948	0.0	0.0	
Annual_HH_Income	50.0	490019.04	320135.792123	64200.0	258750.0	447
No_of_Earning_Members	50.0	1.46	0.734291	1.0	1.0	

In [10]:
```
df.isna().any()#check missing values
```

Out[10]:

	0
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 [14]:
```
#what is mean expence of houshold
df["Mthly_HH_Expense"].mean()
```

Out[14]:
```
np.float64(18818.0)
```

In [15]:
```
#what is median expence of houshold
df["Mthly_HH_Expense"].median()
```

Out[15]:
```
15500.0
```

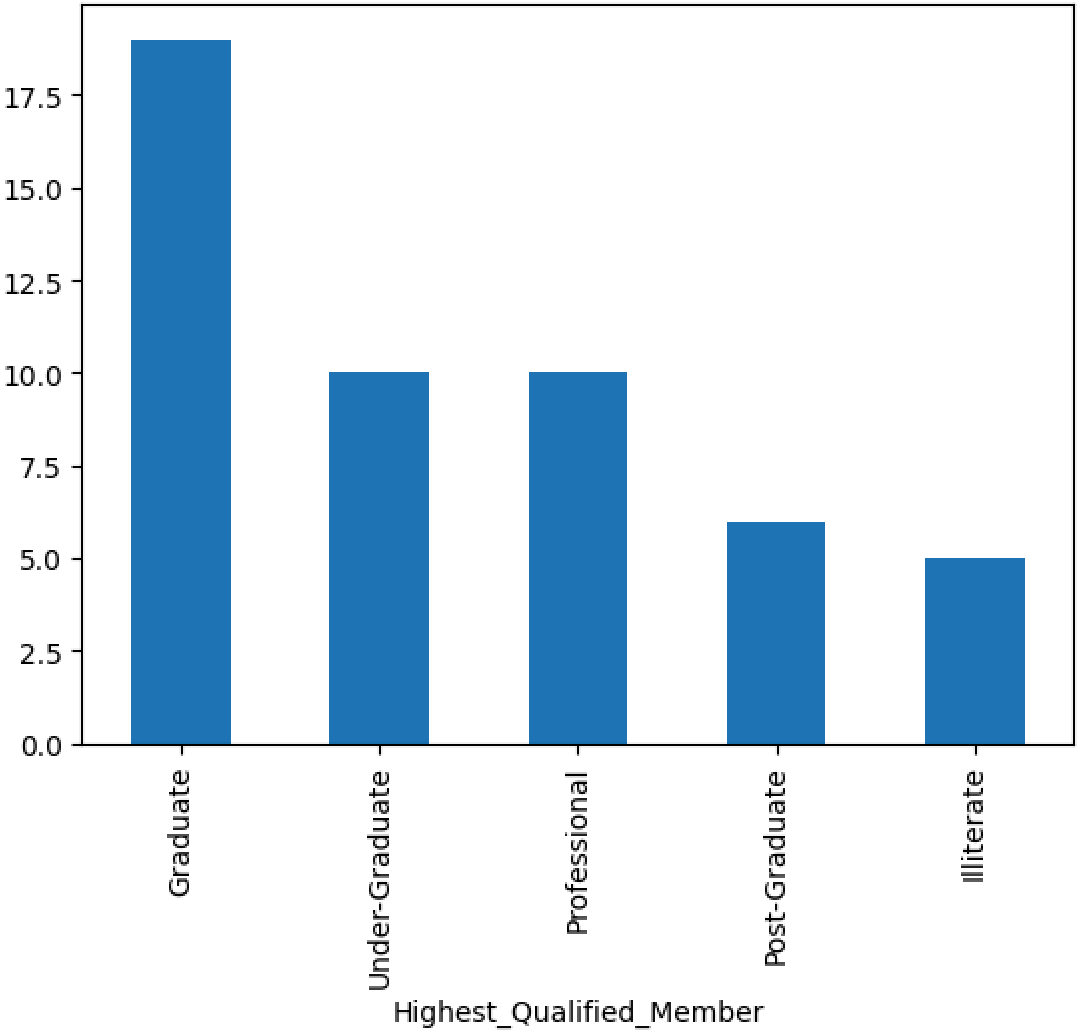
In [16]:
```
#what is montly
mth_exp_tmp = pd.crosstab(index= df['Mthly_HH_Expense'], columns="count")
mth_exp_tmp.reset_index(inplace=True)
mth_exp_tmp[mth_exp_tmp['count'] == df.Mthly_HH_Expense.value_counts().max()]
```

Out[16]:

col_0	Mthly_HH_Expense	count
18	25000	8

In [17]:
```
#Plot the Histogram tocount the Higest Qualified member
df["Highest_Qualified_Member"].value_counts().plot(kind="bar")
```

Out[17]:
```
<Axes: xlabel='Highest_Qualified_Member'>
```

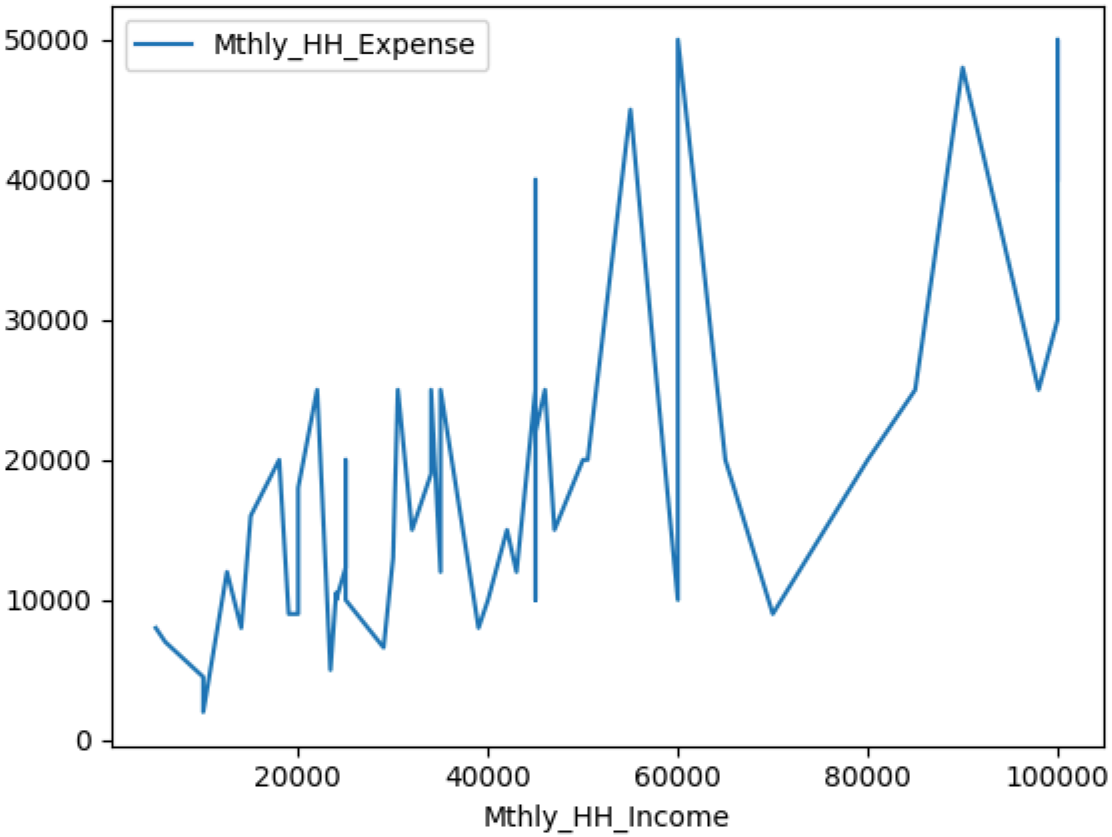


In [20]:

```
#Calculate IQR (inter quantile range) 75% and 25%
df.plot(x="Mthly_HH_Income", y='Mthly_HH_Expense')
IQR = df['Mthly_HH_Expense'].quantile(0.75)-df['Mthly_HH_Expense'].quantile(0.
IQR
```

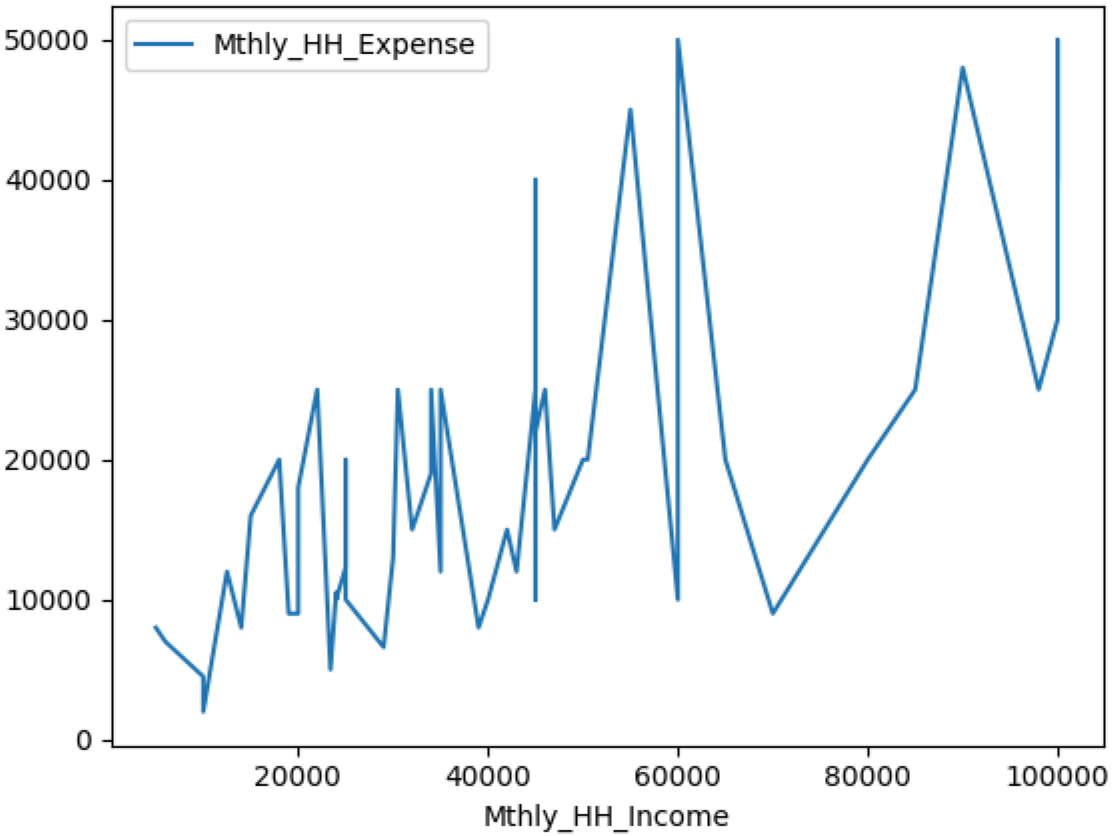
Out[20]:

```
np.float64(15000.0)
```



```
In [19]: df.plot(x="Mthly_HH_Income", y='Mthly_HH_Expense')
         IQR = df['Mthly_HH_Expense'].quantile(0.50)-df['Mthly_HH_Expense'].quantile(0.
         IQR
```

```
Out[19]: np.float64(5500.0)
```



In [22]:
```
#calculate SD for first 4 columns
pd.DataFrame(df.iloc[:,0:5].std().to_frame()).T # iloc - index location
```

Out[22]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt
0	26097.908979	12090.216824	1.517382	6241.434948

In [24]:
```
#calculate variance for first 3 columns
pd.DataFrame(df.iloc[:,0:4].std().to_frame()).T
```

Out[24]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt
0	26097.908979	12090.216824	1.517382	6241.434948

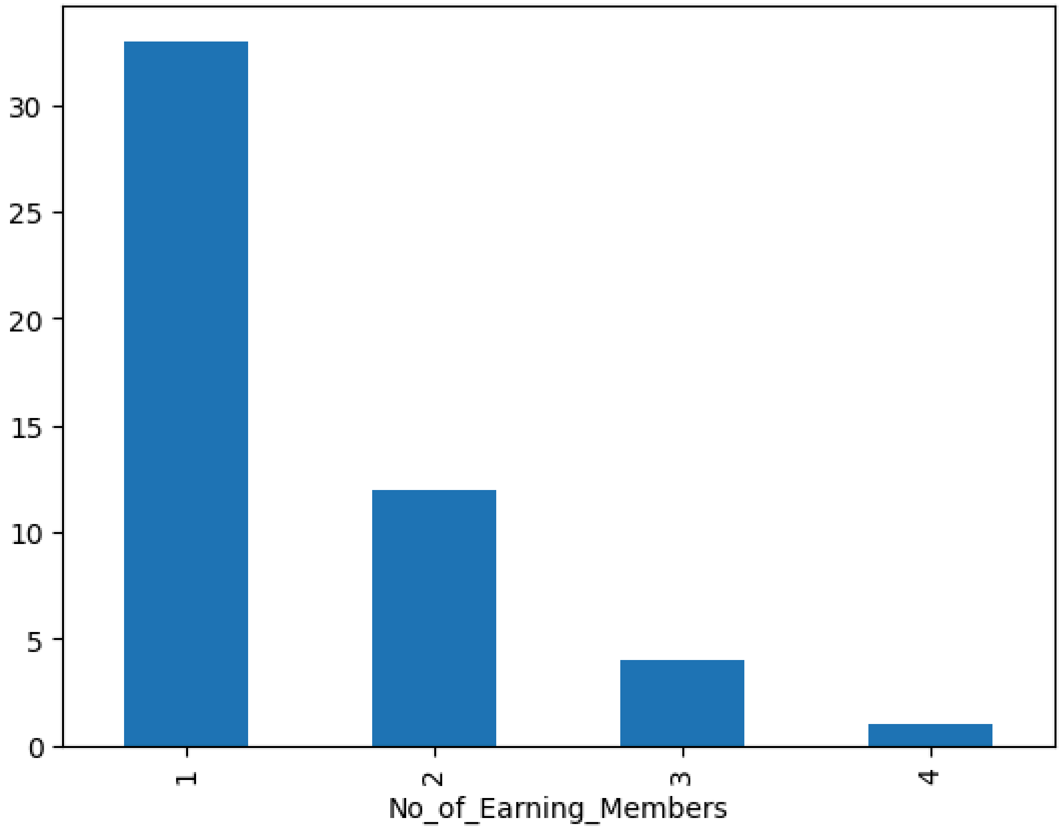
In [28]:
```
#Calculate the count of highest qualified member
df["Highest_Qualified_Member"].value_counts().to_frame()
```

```
Out[28]:
```

	count
Highest_Qualified_Member	
Graduate	19
Under-Graduate	10
Professional	10
Post-Graduate	6
Illiterate	5

```
In [29]: #Plot the histogram to count the number of earning member
         df['No_of_Earning_Members'].value_counts().plot(kind="bar")
```

```
Out[29]: <Axes: xlabel='No_of_Earning_Members'>
```



```
In [33]: #Suppose you have option to invest in stock A or stock B. The stock have a dif
         Coeff_of_var_StockA = 10/15
         print(Coeff_of_var_StockA)
         Coeff_of_var_StockB=5/10
         print(Coeff_of_var_StockB)
```

```
0.6666666666666666
0.5
```