

```
In [2]: import numpy as np # Linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [3]: income_df=pd.read_csv(r"C:\Users\kumar\Desktop\Inc_Exp_Data.csv")
```

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

```
Out[4]:
```

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income
0	5000	8000	3	2000	15000
1	6000	7000	2	3000	18000
2	10000	4500	2	0	30000
3	10000	2000	1	0	30000
4	12500	12000	2	3000	37500



```
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]: len(income_df.columns)
```

```
Out[7]: 7
```

```
In [8]: income_df.describe()
```

Out[8]:

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

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

Out[9]:

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

In [10]: `income_df.isna()`

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

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

```
In [11]: income_df.isna().sum()
```

```
Out[11]: Mthly_HH_Income      0
          Mthly_HH_Expense     0
          No_of_Fly_Members    0
          Emi_or_Rent_Amt       0
          Annual_HH_Income      0
          Highest_Qualified_Member 0
          No_of_Earning_Members   0
          dtype: int64
```

```
In [12]: income_df.isna().any()
```

```
Out[12]: 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 [13]: income_df["Mthly_HH_Expense"].mean()
```

```
Out[13]: np.float64(18818.0)
```

```
In [14]: income_df['Mthly_HH_Expense'].median()
```

```
Out[14]: 15500.0
```

```
In [15]: income_df['Mthly_HH_Expense'].mode()
```

```
Out[15]: 0    25000
Name: Mthly_HH_Expense, dtype: int64
```

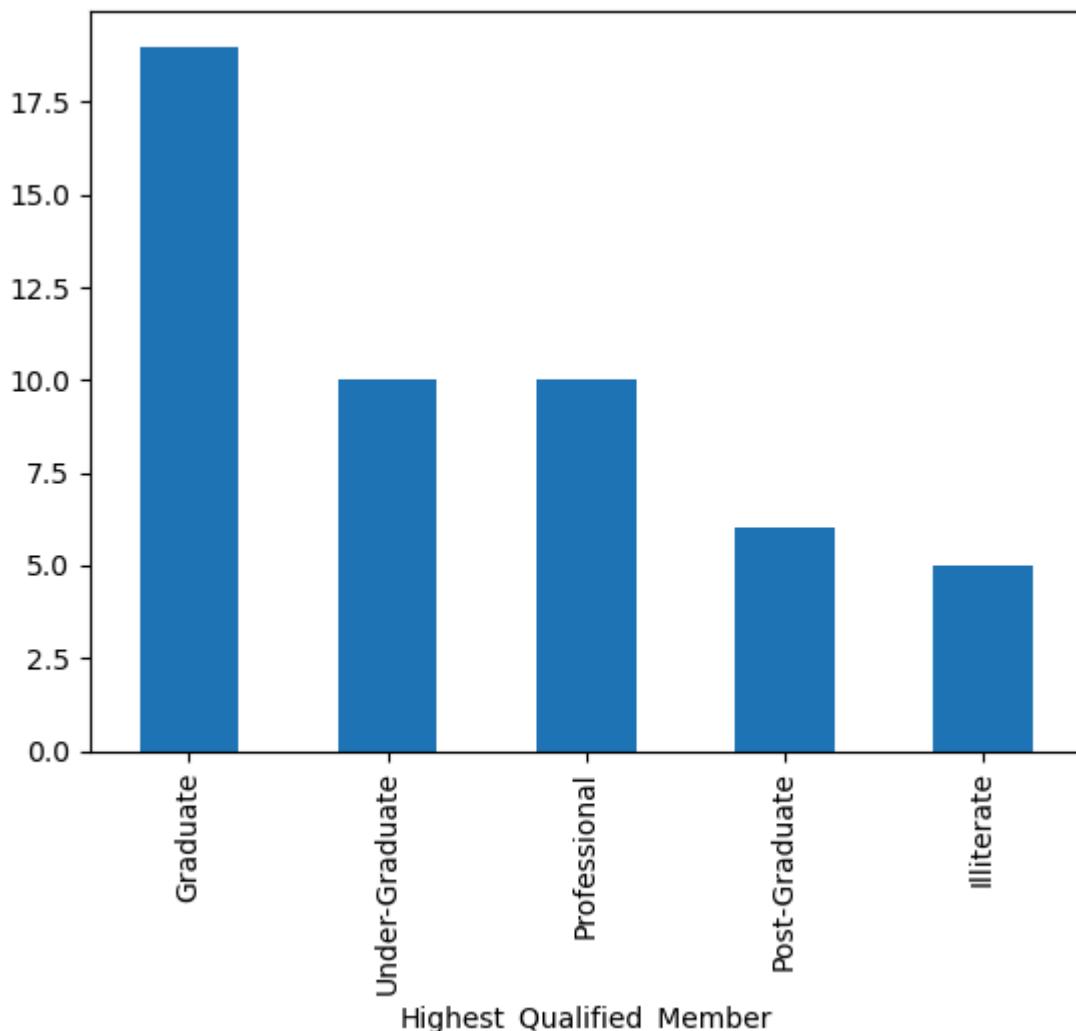
```
In [16]: mth_exp_tmp = pd.crosstab(index=income_df["Mthly_HH_Expense"], columns="count")
mth_exp_tmp.reset_index(inplace=True)
mth_exp_tmp[mth_exp_tmp['count']] == income_df.Mthly_HH_Expense.value_counts().ma
```

```
Out[16]: col_0  Mthly_HH_Expense  count
```

col_0	Mthly_HH_Expense	count
18	25000	8

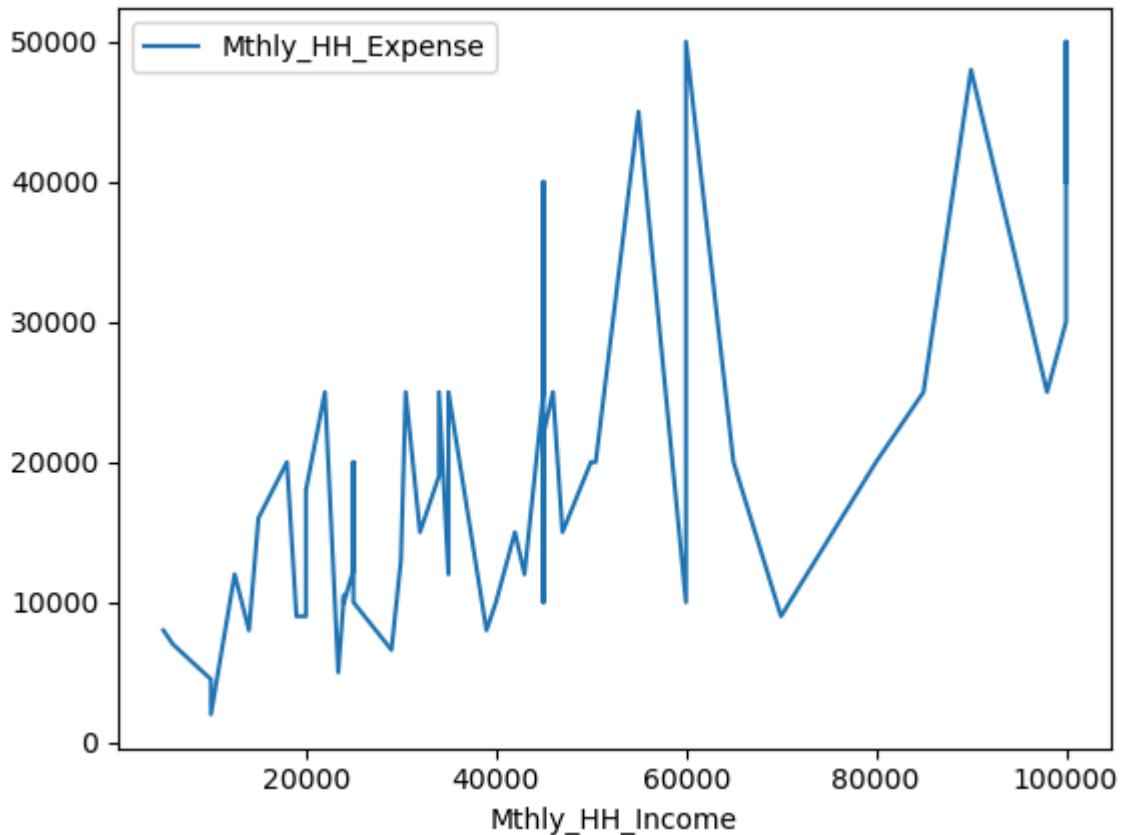
```
In [17]: income_df["Highest_Qualified_Member"].value_counts().plot(kind="bar")
```

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



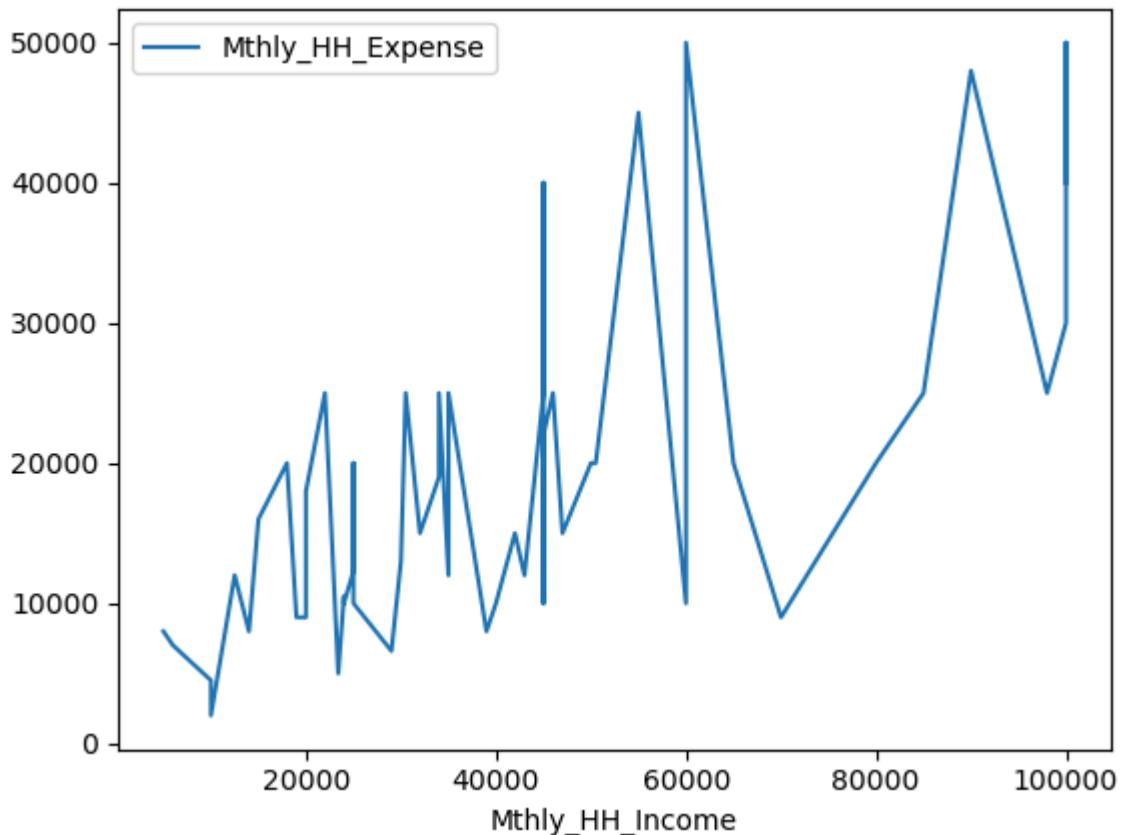
```
In [22]: income_df.plot(x="Mthly_HH_Income", y="Mthly_HH_Expense")
IQR=income_df["Mthly_HH_Expense"].quantile(0.75)-income_df["Mthly_HH_Expense"].q
IQR
```

Out[22]: np.float64(15000.0)



```
In [23]: income_df.plot(x="Mthly_HH_Income", y="Mthly_HH_Expense")
IQR=income_df["Mthly_HH_Expense"].quantile(0.25)+income_df["Mthly_HH_Expense"].q
IQR
```

Out[23]: np.float64(20000.0)



```
In [24]: IQR=income_df["Mthly_HH_Expense"].quantile(0.75)-income_df["Mthly_HH_Expense"].q  
IQR
```

```
Out[24]: np.float64(15000.0)
```

```
In [25]: IQR=income_df["Mthly_HH_Expense"].quantile(0.50)-income_df["Mthly_HH_Expense"].q  
IQR
```

```
Out[25]: np.float64(5500.0)
```

```
In [19]: income_df['Mthly_HH_Expense'].var()
```

```
Out[19]: 146173342.85714287
```

```
In [20]: income_df['Mthly_HH_Expense'].std()
```

```
Out[20]: 12090.216824240286
```

```
In [21]: coef = income_df['Mthly_HH_Expense'].std() / income_df['Mthly_HH_Expense'].mean()  
coef
```

```
Out[21]: np.float64(0.6424814977277227)
```

Descriptve stats we completed