Import libraries

In [1]:
```
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
```

Import dataset

In [2]:
```
df = pd.read_csv(r"C:\Users\ADMIN\Desktop\DATA SCIENCE NOTES\NOVEMBER MONTH\27TH NOV\24th, 25th -  Intro to Stats, De
```

In [3]:
```
df
```

Out[3]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Membe
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
5	14000	8000	2	0	196560	Graduate
6	15000	16000	3	35000	167400	Post-Graduate
7	18000	20000	5	8000	216000	Graduate
8	19000	9000	2	0	218880	Under-Graduate
9	20000	9000	4	0	220800	Under-Graduate
10	20000	18000	4	8000	278400	Under-Graduate
11	22000	25000	6	12000	279840	Illiterate
12	23400	5000	3	0	292032	Illiterate
13	24000	10500	6	0	316800	Graduate
14	24000	10000	4	0	244800	Graduate
15	25000	12300	3	0	246000	Graduate
16	25000	20000	3	3500	261000	Graduate
17	25000	10000	6	0	258000	Under-Graduate
18	29000	6600	2	2000	348000	Graduate
19	30000	13000	4	0	385200	Graduate
20	30500	25000	5	5000	351360	Under-Graduate
21	32000	15000	4	0	445440	Professiona

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Membe
22	34000	19000	6	0	330480	Professiona
23	34000	25000	3	4000	469200	Professiona
24	35000	12000	3	0	466200	Graduate
25	35000	25000	4	0	449400	Professiona
26	39000	8000	4	0	556920	Under-Graduate
27	40000	10000	4	0	412800	Under-Graduate
28	42000	15000	4	0	488880	Graduate
29	43000	12000	4	0	619200	Graduate
30	45000	25000	6	0	523800	Graduate
31	45000	40000	6	3500	507600	Professiona
32	45000	10000	2	1000	437400	Post-Graduate
33	45000	22000	4	2500	610200	Post-Graduate
34	46000	25000	5	3500	596160	Graduate
35	47000	15000	7	0	456840	Professiona
36	50000	20000	4	0	570000	Professiona
37	50500	20000	3	0	581760	Professiona
38	55000	45000	6	12000	600600	Graduate
39	60000	10000	3	0	590400	Post-Graduate
40	60000	50000	6	10000	590400	Graduate
41	65000	20000	4	5000	647400	Illiterate
42	70000	9000	2	0	756000	Graduate
43	80000	20000	4	0	1075200	Graduate

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Membe
44	85000	25000	5	0	1142400	Under-Graduate
45	90000	48000	7	0	885600	Post-Graduate
46	98000	25000	5	0	1152480	Professiona
47	100000	30000	6	0	1404000	Graduate
48	100000	50000	4	20000	1032000	Professiona
49	100000	40000	6	10000	1320000	Post-Graduate

summary of dataset

In [4]:
```
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
```

shape of dataset

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

Out[5]: (50, 7)

In [6]:
```
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(df.columns)
```

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

descriptive statistics

In [8]:
```
df.describe().T
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
Mthly_HH_Income	50.0	41558.00	26097.908979	5000.0	23550.0	35000.0	50375.0	100000.0
Mthly_HH_Expense	50.0	18818.00	12090.216824	2000.0	10000.0	15500.0	25000.0	50000.0
No_of_Fly_Members	50.0	4.06	1.517382	1.0	3.0	4.0	5.0	7.0
Emi_or_Rent_Amt	50.0	3060.00	6241.434948	0.0	0.0	0.0	3500.0	35000.0
Annual_HH_Income	50.0	490019.04	320135.792123	64200.0	258750.0	447420.0	594720.0	1404000.0
No_of_Earning_Members	50.0	1.46	0.734291	1.0	1.0	1.0	2.0	4.0

checking null values

In [9]:
```
df.isna()
```

Out[9]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Membe
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	False	False	False	False	False	False
7	False	False	False	False	False	False
8	False	False	False	False	False	False
9	False	False	False	False	False	False
10	False	False	False	False	False	False
11	False	False	False	False	False	False
12	False	False	False	False	False	False
13	False	False	False	False	False	False
14	False	False	False	False	False	False
15	False	False	False	False	False	False
16	False	False	False	False	False	False
17	False	False	False	False	False	False
18	False	False	False	False	False	False
19	False	False	False	False	False	False
20	False	False	False	False	False	False
21	False	False	False	False	False	False

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Membe
22	False	False	False	False	False	False
23	False	False	False	False	False	False
24	False	False	False	False	False	False
25	False	False	False	False	False	False
26	False	False	False	False	False	False
27	False	False	False	False	False	False
28	False	False	False	False	False	False
29	False	False	False	False	False	False
30	False	False	False	False	False	False
31	False	False	False	False	False	False
32	False	False	False	False	False	False
33	False	False	False	False	False	False
34	False	False	False	False	False	False
35	False	False	False	False	False	False
36	False	False	False	False	False	False
37	False	False	False	False	False	False
38	False	False	False	False	False	False
39	False	False	False	False	False	False
40	False	False	False	False	False	False
41	False	False	False	False	False	False
42	False	False	False	False	False	False
43	False	False	False	False	False	False

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Membe
44	False	False	False	False	False	False
45	False	False	False	False	False	False
46	False	False	False	False	False	False
47	False	False	False	False	False	False
48	False	False	False	False	False	False
49	False	False	False	False	False	False

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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
```

finding mean for monthly house expense

In [11]:
```
df["Mthly_HH_Expense"].mean()
```

Out[11]:
```
18818.0
```

finding median for monthly house expense

In [12]:
```
df["Mthly_HH_Expense"].median()
```

Out[12]:
```
15500.0
```

finding mode for monthly house expense

In [13]:
```
df["Mthly_HH_Expense"].mode()
```

Out[13]:
```
0    25000
Name: Mthly_HH_Expense, dtype: int64
```

counting monthly house expense

In [14]:
```
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()]
```

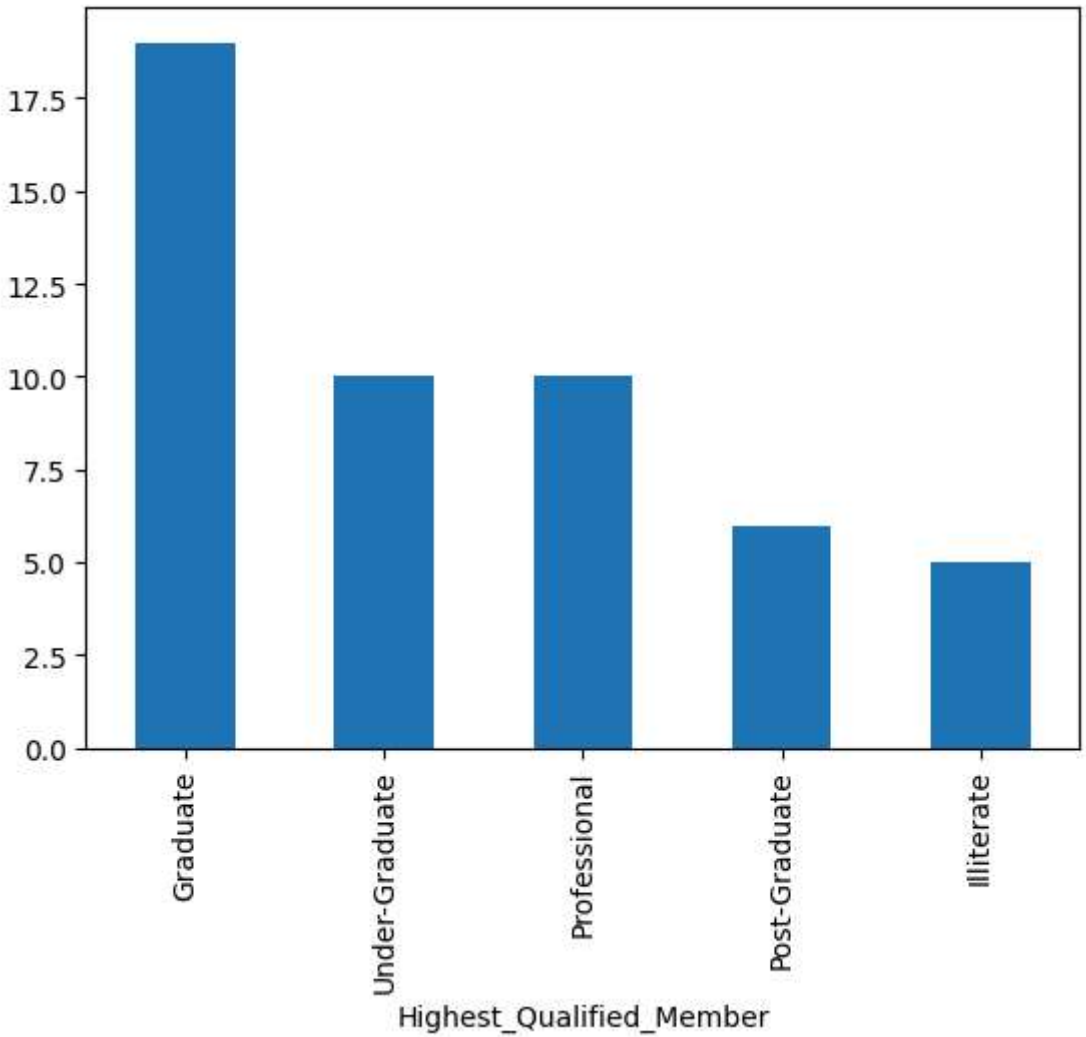
Out[14]:

col_0	Mthly_HH_Expense	count
18	25000	8

visualization

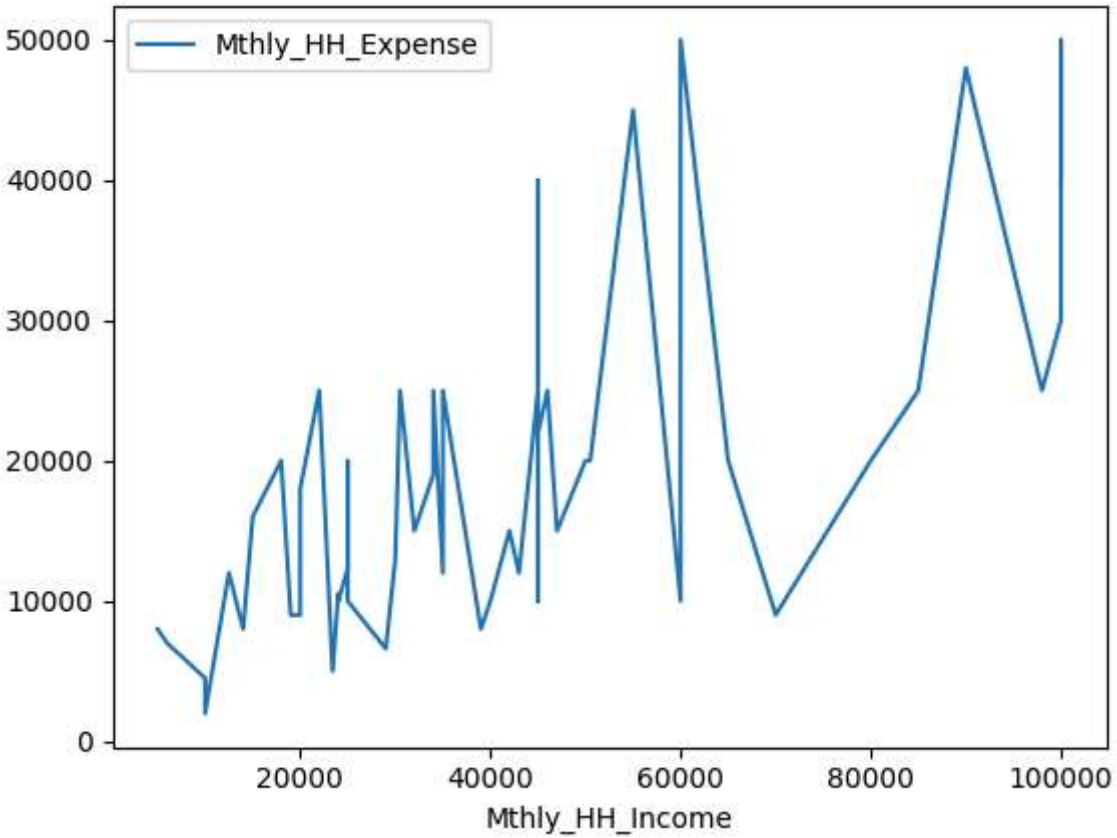
In [15]:
```
df["Highest_Qualified_Member"].value_counts().plot(kind="bar")
```

Out[15]:
```
<Axes: xlabel='Highest_Qualified_Member'>
```



In [16]:
```
df.plot(x='Mthly_HH_Income',y='Mthly_HH_Expense')
```

Out[16]: <Axes: xlabel='Mthly_HH_Income'>



finding 50% quantile

In [17]:
```
IQR = df['Mthly_HH_Expense'].quantile(0.75)-df['Mthly_HH_Expense'].quantile(0.25)
```

In [18]:
```
IQR
```

Out[18]: 15000.0

finding 25% quantile

In [19]:
```
IQR = df['Mthly_HH_Expense'].quantile(0.50)-df['Mthly_HH_Expense'].quantile(0.25)
```

In [20]:
```
IQR
```

Out[20]:
```
5500.0
```

finding monthly house expence variance

In [21]:
```
df['Mthly_HH_Expense'].var()
```

Out[21]:
```
146173342.85714287
```

finding monthly house expence standard deviation

In [22]:
```
df['Mthly_HH_Expense'].std()
```

Out[22]:
```
12090.216824240286
```

finding monthly house expence co-efficient

In [23]:
```
coef = df['Mthly_HH_Expense'].std() / df['Mthly_HH_Expense'].mean()
```

In [24]:
```
coef
```

Out[24]:
```
0.6424814977277227
```