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
```

In [3]:

```
income_df=pd.read_csv(r"C:\Users\Admin\Downloads\16th,17th\16th,17th\Descriptive 
```

In [4]:

```
income_df
```

Out[4]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income
0	5000	8000	3	2000	6420
1	6000	7000	2	3000	7992
2	10000	4500	2	0	11280
3	10000	2000	1	0	9720
4	12500	12000	2	3000	14700
5	14000	8000	2	0	19656
6	15000	16000	3	35000	16740
7	18000	20000	5	8000	21600
8	19000	9000	2	0	21888
9	20000	9000	4	0	22080
10	20000	18000	4	8000	27840
11	22000	25000	6	12000	27984
12	23400	5000	3	0	29203
13	24000	10500	6	0	31680
14	24000	10000	4	0	24480
15	25000	12300	3	0	24600
16	25000	20000	3	3500	26100
17	25000	10000	6	0	25800
18	29000	6600	2	2000	34800
19	30000	13000	4	0	38520
20	30500	25000	5	5000	35136
21	32000	15000	4	0	44544
22	34000	19000	6	0	33048
23	34000	25000	3	4000	46920
24	35000	12000	3	0	46620
25	35000	25000	4	0	44940
26	39000	8000	4	0	55692
27	40000	10000	4	0	41280
28	42000	15000	4	0	48888
29	43000	12000	4	0	61920
30	45000	25000	6	0	52380
31	45000	40000	6	3500	50760
32	45000	10000	2	1000	43740
33	45000	22000	4	2500	61020
34	46000	25000	5	3500	59616
35	47000	15000	7	0	45684

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income
36	50000	20000	4	0	570000
37	50500	20000	3	0	581760
38	55000	45000	6	12000	600600
39	60000	10000	3	0	590400
40	60000	50000	6	10000	590400
41	65000	20000	4	5000	647400
42	70000	9000	2	0	756000
43	80000	20000	4	0	1075200
44	85000	25000	5	0	1142400
45	90000	48000	7	0	885600
46	98000	25000	5	0	1152480
47	100000	30000	6	0	1404000
48	100000	50000	4	20000	1032000
49	100000	40000	6	10000	1320000

In [5]:

```
income_df.head()
```

Out[5]:

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

In [6]:

```
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 [7]:
```
income_df.shape
```

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

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

Out[8]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Inc
count	50.000000	50.000000	50.000000	50.000000	5.000000e
mean	41558.000000	18818.000000	4.060000	3060.000000	4.900190e
std	26097.908979	12090.216824	1.517382	6241.434948	3.201358e
min	5000.000000	2000.000000	1.000000	0.000000	6.420000e
25%	23550.000000	10000.000000	3.000000	0.000000	2.587500e
50%	35000.000000	15500.000000	4.000000	0.000000	4.474200e
75%	50375.000000	25000.000000	5.000000	3500.000000	5.947200e
max	100000.000000	50000.000000	7.000000	35000.000000	1.404000e

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

Out[9]:

	count	mean	std	min	25%	50%	75%	
Mthly_HH_Income	50.0	41558.00	26097.908979	5000.0	23550.0	35000.0	50375.0	1
Mthly_HH_Expense	50.0	18818.00	12090.216824	2000.0	10000.0	15500.0	25000.0	
No_of_Fly_Members	50.0	4.06	1.517382	1.0	3.0	4.0	5.0	
Emi_or_Rent_Amt	50.0	3060.00	6241.434948	0.0	0.0	0.0	3500.0	
Annual_HH_Income	50.0	490019.04	320135.792123	64200.0	258750.0	447420.0	594720.0	14
No_of_Earning_Members	50.0	1.46	0.734291	1.0	1.0	1.0	2.0	

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

Out[10]:
```
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 [11]:

```
income_df.isna().all()
```

Out[11]:

```
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 [12]:

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

Out[12]:

```
18818.0
```

In [13]:

```
income_df["Mthly_HH_Income"].mean()
```

Out[13]:

```
41558.0
```

In [14]:

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

Out[14]:

```
15500.0
```

In [23]:

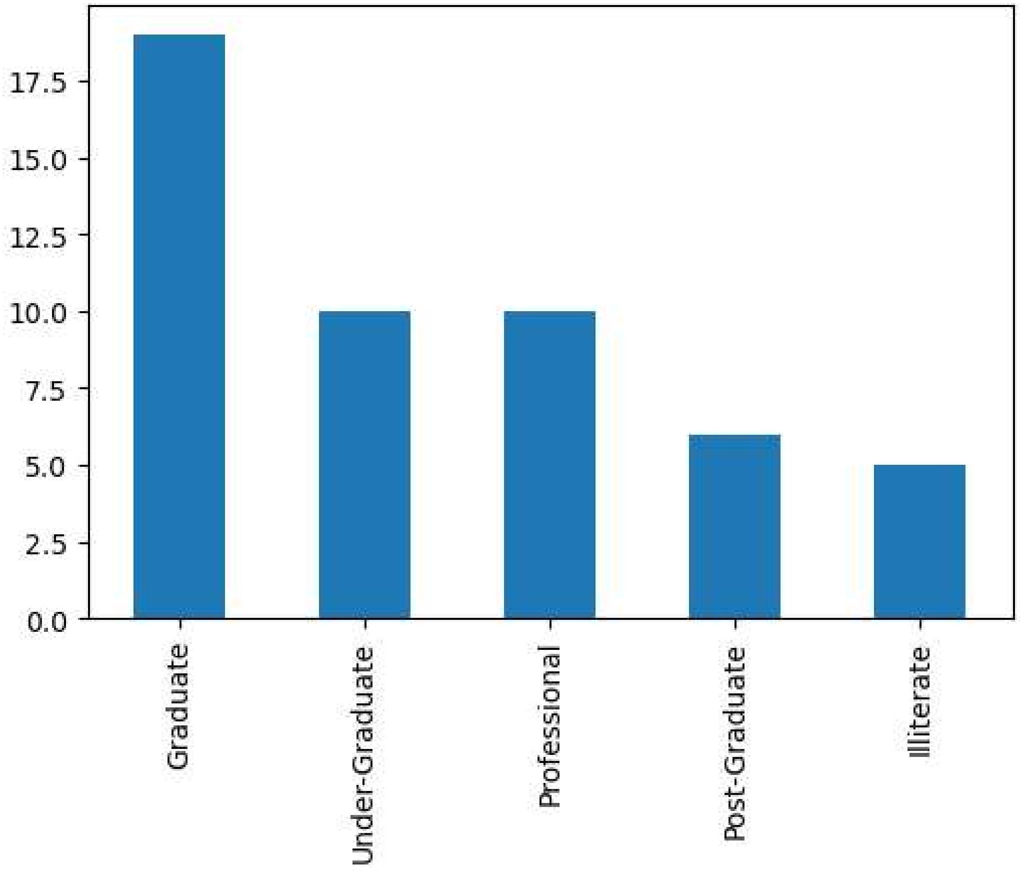
```
mth_exp_tmp=pd.crosstab(index=income_df["Mthly_HH_Expense"],columns="count")
mth_exp_tmp.reset_index(inplace=True)
mth_exp_tmp[mth_exp_tmp['count']==income_df.Mthly_HH_Expense.value_counts().max()
```

Out[23]:

col_0	Mthly_HH_Expense	count
18	25000	8

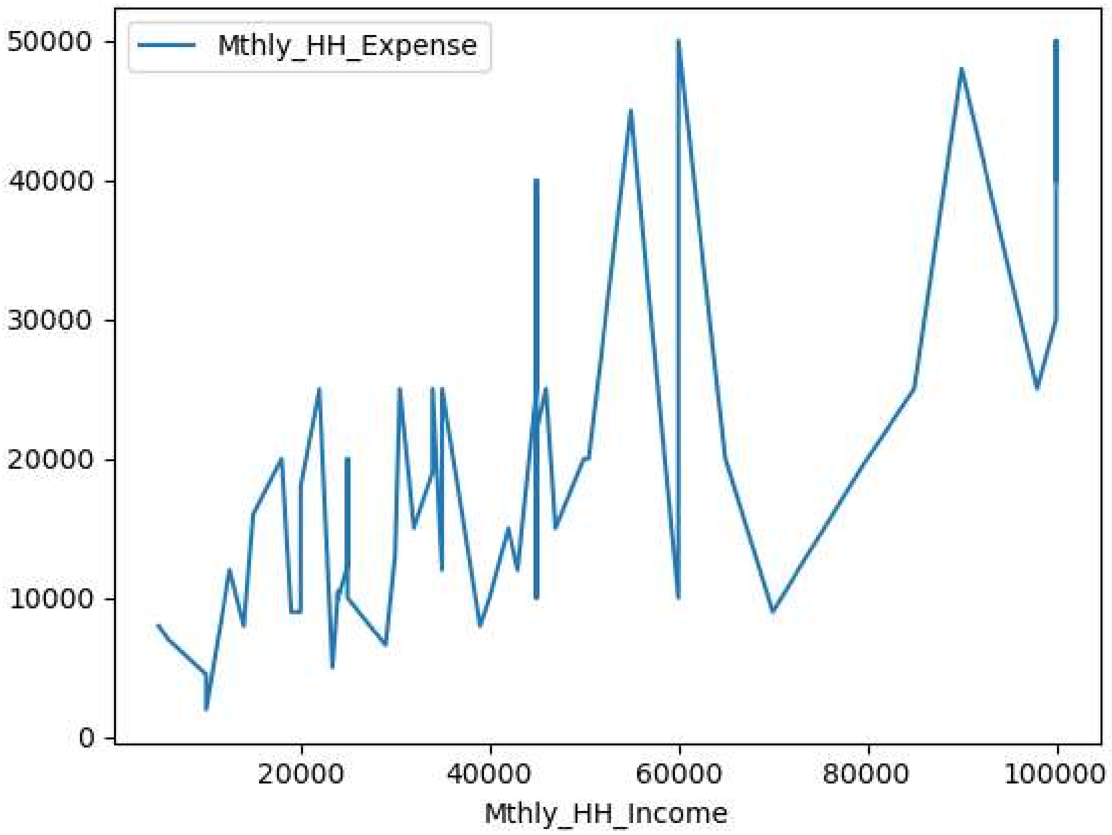
In [30]:
```
f,ax=plt.subplots(figsize=(6,4))
income_df["Highest_Qualified_Member"].value_counts().plot(kind='bar')
```

Out[30]:
```
<Axes: >
```



In [31]:
```
income_df.plot(x="Mthly_HH_Income", y="Mthly_HH_Expense")
```

Out[31]:
```
<Axes: xlabel='Mthly_HH_Income'>
```



In [32]:
```
IQR=income_df["Mthly_HH_Expense"].quantile(0.75)-income_df["Mthly_HH_Expense"].qu
IQR
```

Out[32]:
```
15000.0
```

In [36]:
```
pd.DataFrame(income_df.iloc[:,0:5].std().to_frame()).T
```

Out[36]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income
0	26097.908979	12090.216824	1.517382	6241.434948	320135.792123

In [37]:
```
pd.DataFrame(income_df.iloc[:,0:4].var().to_frame()).T
```

Out[37]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt
0	6.811009e+08	1.461733e+08	2.302449	3.895551e+07

In [38]:

```
income_df["Highest_Qualified_Member"].value_counts().to_frame().T
```

Out[38]:

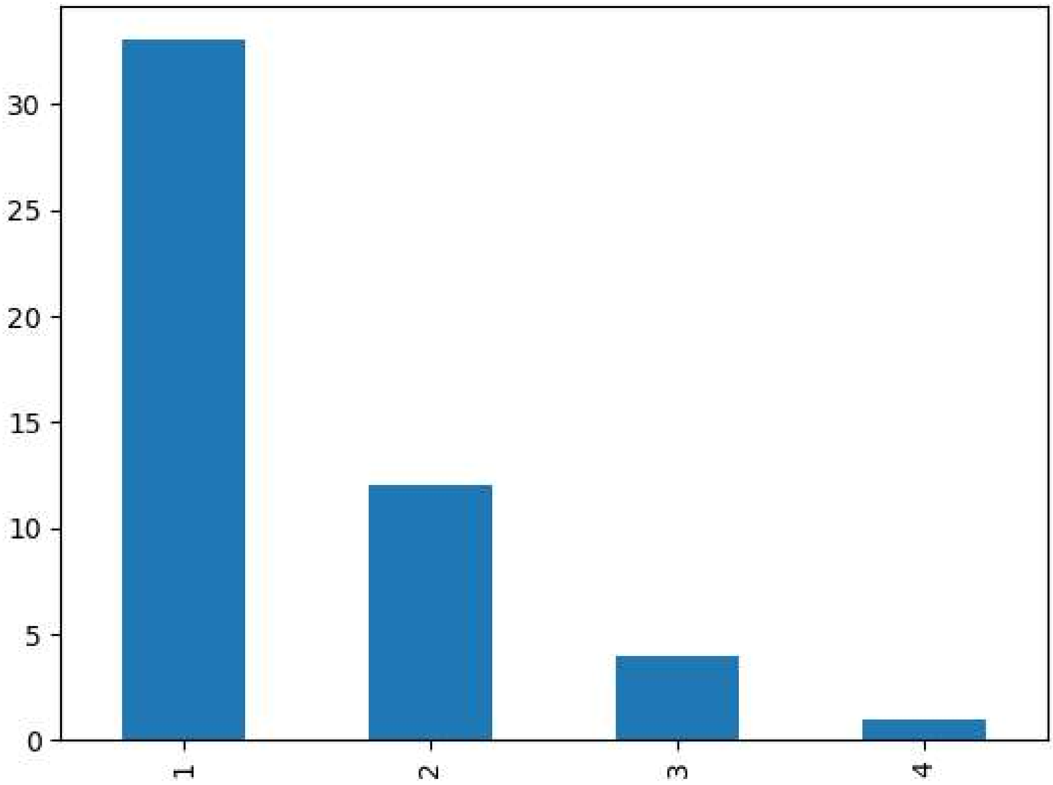
	Graduate	Under-Graduate	Professional	Post-Graduate	Illiterate
Highest_Qualified_Member	19	10	10	6	5

In [39]:

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

Out[39]:

```
<Axes: >
```



In [41]:

```
coeff_var_stockA=10/15
coeff_var_stockB=5/10
print(coeff_var_stockA)
print(coeff_var_stockB)
```

```
0.6666666666666666
0.5
```