In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
```

In [2]:

```python
Income=pd.read_csv(r'C:\Users\Dell\Downloads\16th,17th\16th,17th\Descriptive stats code- practivle\Inc_Exp_Data.csv
```

In [3]:

```
Income
```

Out[3]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Memb
0	5000	8000	3	2000	64200	Under-Gradua
1	6000	7000	2	3000	79920	Illitera
2	10000	4500	2	0	112800	Under-Gradua
3	10000	2000	1	0	97200	Illitera
4	12500	12000	2	3000	147000	Gradua
5	14000	8000	2	0	196560	Gradua
6	15000	16000	3	35000	167400	Post-Gradua
7	18000	20000	5	8000	216000	Gradua
8	19000	9000	2	0	218880	Under-Gradua
9	20000	9000	4	0	220800	Under-Gradua
10	20000	18000	4	8000	278400	Under-Gradua
11	22000	25000	6	12000	279840	Illitera
12	23400	5000	3	0	292032	Illitera
13	24000	10500	6	0	316800	Gradua
14	24000	10000	4	0	244800	Gradua
15	25000	12300	3	0	246000	Gradua
16	25000	20000	3	3500	261000	Gradua
17	25000	10000	6	0	258000	Under-Gradua
18	29000	6600	2	2000	348000	Gradua
19	30000	13000	4	0	385200	Gradua
20	30500	25000	5	5000	351360	Under-Gradua
21	32000	15000	4	0	445440	Profession
22	34000	19000	6	0	330480	Profession
23	34000	25000	3	4000	469200	Profession
24	35000	12000	3	0	466200	Gradua
25	35000	25000	4	0	449400	Profession
26	39000	8000	4	0	556920	Under-Gradua
27	40000	10000	4	0	412800	Under-Gradua
28	42000	15000	4	0	488880	Gradua
29	43000	12000	4	0	619200	Gradua
30	45000	25000	6	0	523800	Gradua
31	45000	40000	6	3500	507600	Profession
32	45000	10000	2	1000	437400	Post-Gradua
33	45000	22000	4	2500	610200	Post-Gradua
34	46000	25000	5	3500	596160	Gradua
35	47000	15000	7	0	456840	Profession
36	50000	20000	4	0	570000	Profession
37	50500	20000	3	0	581760	Profession
38	55000	45000	6	12000	600600	Gradua
39	60000	10000	3	0	590400	Post-Gradua
40	60000	50000	6	10000	590400	Gradua
41	65000	20000	4	5000	647400	Illitera
42	70000	9000	2	0	756000	Gradua
43	80000	20000	4	0	1075200	Gradua
44	85000	25000	5	0	1142400	Under-Gradua
45	90000	48000	7	0	885600	Post-Gradua
46	98000	25000	5	0	1152480	Profession

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Memb
47	100000	30000	6	0	1404000	Gradua
48	100000	50000	4	20000	1032000	Profession
49	100000	40000	6	10000	1320000	Post-Gradua

In [38]:

```
Income.head()
```

Out[38]:

	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

In [39]:

```
Income.tail()
```

Out[39]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	Highest_Qualified_Memb
45	90000	48000	7	0	885600	Post-Gradua
46	98000	25000	5	0	1152480	Profession
47	100000	30000	6	0	1404000	Gradua
48	100000	50000	4	20000	1032000	Profession
49	100000	40000	6	10000	1320000	Post-Gradua

In [4]:

```
Income.shape
```

Out[4]:

```
(50, 7)
```

In [5]:

```
Income.columns
```

Out[5]:

```
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 [6]:

```
Income.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]:

```
Income.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

In [9]:

```
Income.describe()
```

Out[9]:

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Income	No_of_Earning_Memb
count	50.000000	50.000000	50.000000	50.000000	5.000000e+01	50.000
mean	41558.000000	18818.000000	4.060000	3060.000000	4.900190e+05	1.460
std	26097.908979	12090.216824	1.517382	6241.434948	3.201358e+05	0.734
min	5000.000000	2000.000000	1.000000	0.000000	6.420000e+04	1.000
25%	23550.000000	10000.000000	3.000000	0.000000	2.587500e+05	1.000
50%	35000.000000	15500.000000	4.000000	0.000000	4.474200e+05	1.000
75%	50375.000000	25000.000000	5.000000	3500.000000	5.947200e+05	2.000
max	100000.000000	50000.000000	7.000000	35000.000000	1.404000e+06	4.000

In [13]:

```
Income.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 [ ]:

```
#mean & median expense of a household
```

In [16]:

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

Out[16]:

```
15500.0
```

In [40]:

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

Out[40]:

```
41558.0
```

In [41]:

```
Income["Mthly_HH_Income"].median()
```

Out[41]:

```
35000.0
```

In [17]:

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

Out[17]:

```
18818.0
```

In [20]:

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

Out[20]:

col_0	Mthly_HH_Expense	count
18	25000	8

In [42]:

```
mth_exp_tem=pd.crosstab(index=Income["Mthly_HH_Income"], columns="count")
mth_exp_tem.reset_index(inplace=True)
mth_exp_tem[mth_exp_tem['count']==Income.Mthly_HH_Income.value_counts().max()]
```

Out[42]:

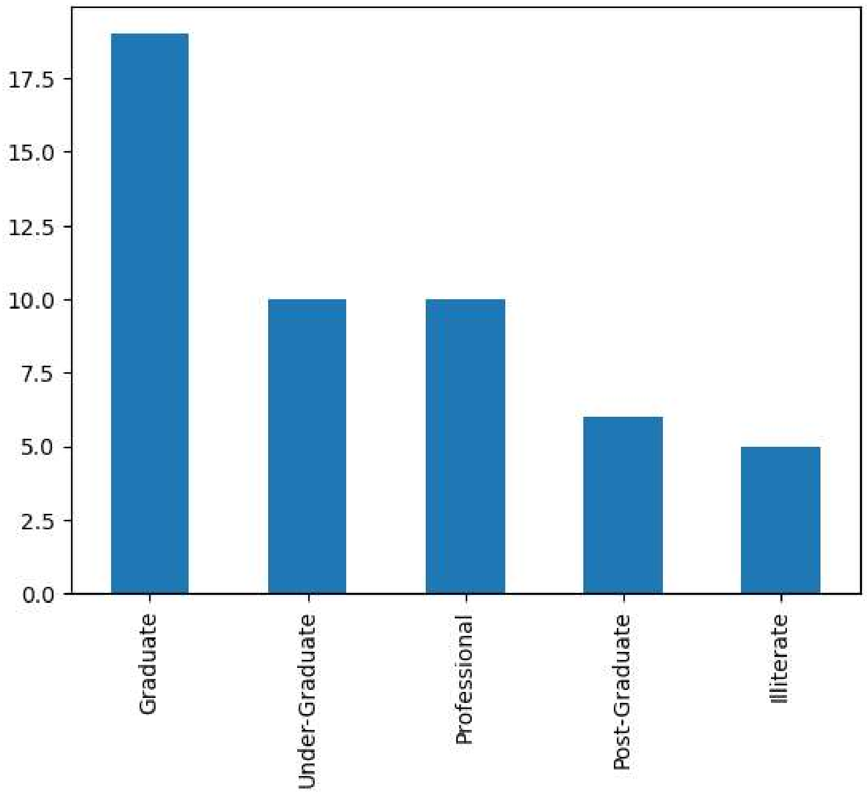
col_0	Mthly_HH_Income	count
23	45000	4

In [21]:

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

Out[21]:

```
<Axes: >
```

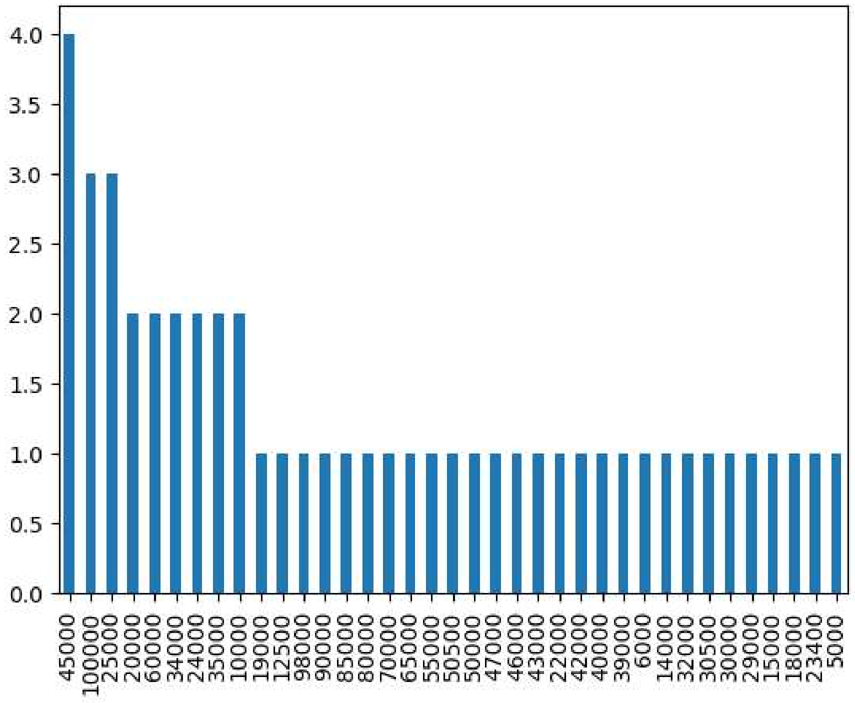


In [44]:

```
Income["Mthly_HH_Income"].value_counts().plot(kind="bar")
```

Out[44]:

```
<Axes: >
```

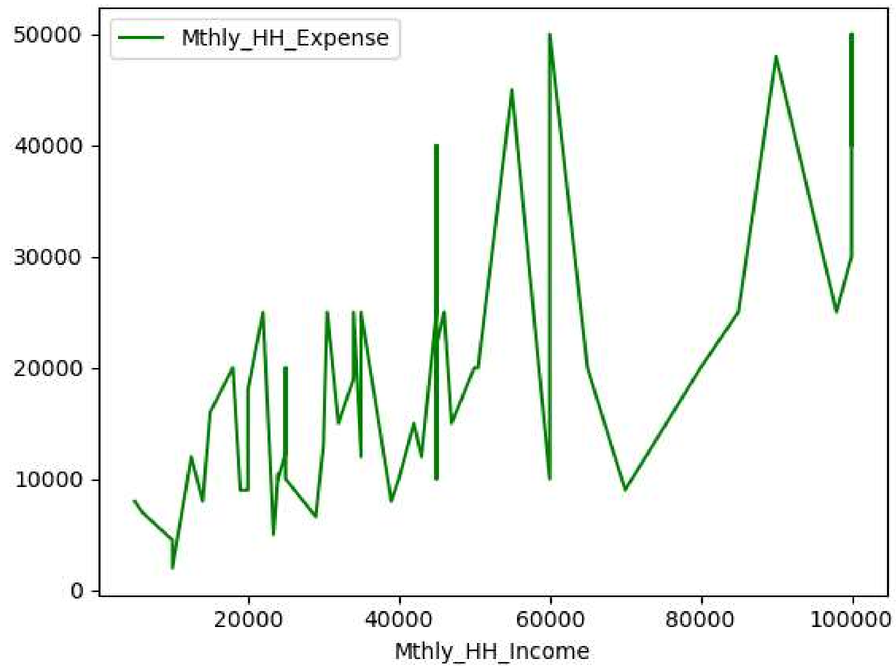


In [45]:

```
Income.plot(x="Mthly_HH_Income", y="Mthly_HH_Expense",color="green")
IQR=Income["Mthly_HH_Expense"].quantile(0.75)-Income["Mthly_HH_Expense"].quantile(0.25)
IQR
```

Out[45]:

```
15000.0
```

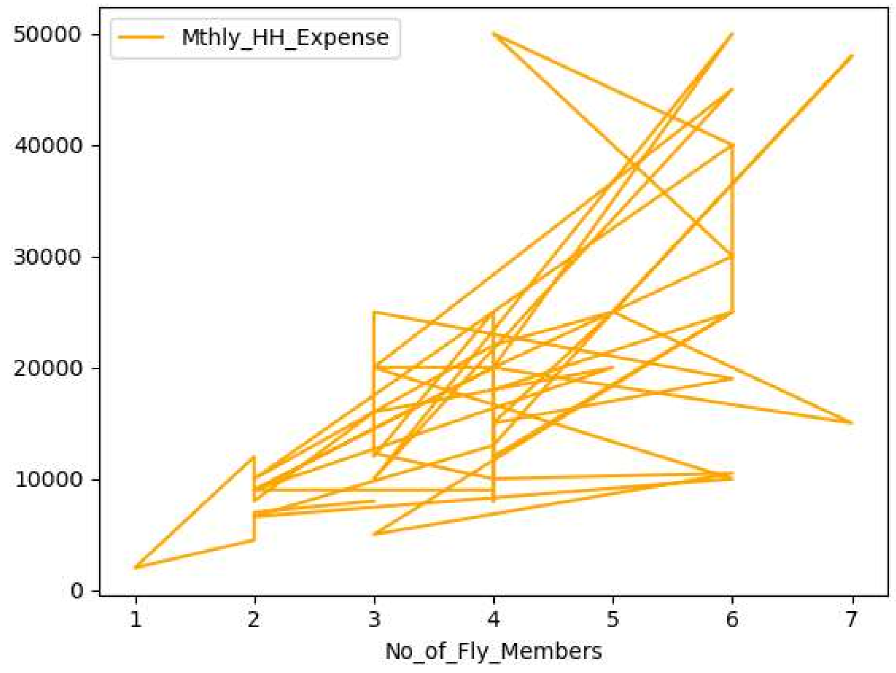


In [59]:

```
Income.plot(x="No_of_Fly_Members", y="Mthly_HH_Expense",color="orange")
IQR=Income["No_of_Fly_Members"].quantile(0.75)-Income["No_of_Fly_Members"].quantile(0.25)
IQR
```

Out[59]:

```
2.0
```



In [24]:

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

Out[24]:

	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 [28]:

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

Out[28]:

	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 [29]:

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

Out[29]:

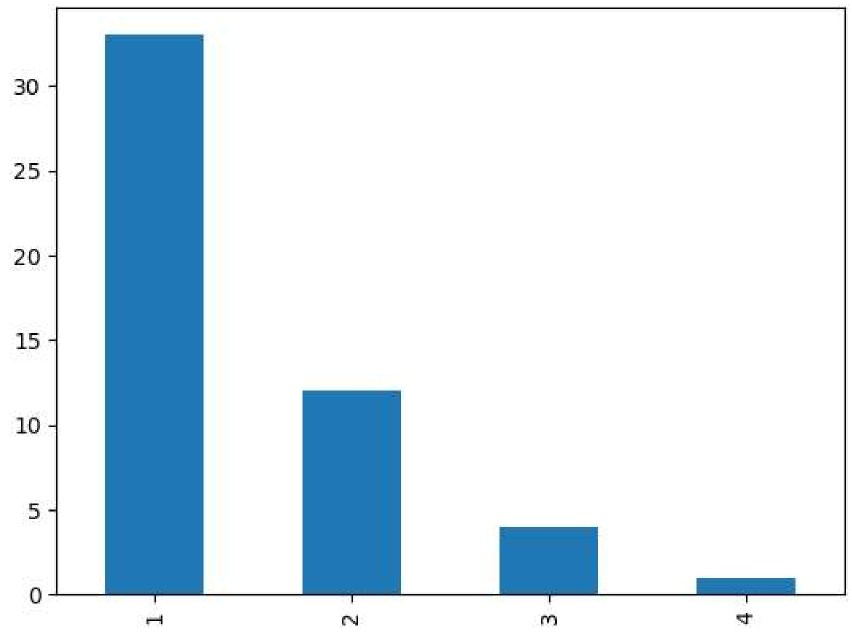
	Graduate	Under-Graduate	Professional	Post-Graduate	Illiterate
Highest_Qualified_Member	19	10	10	6	5

In [47]:

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

Out[47]:

```
<Axes: >
```

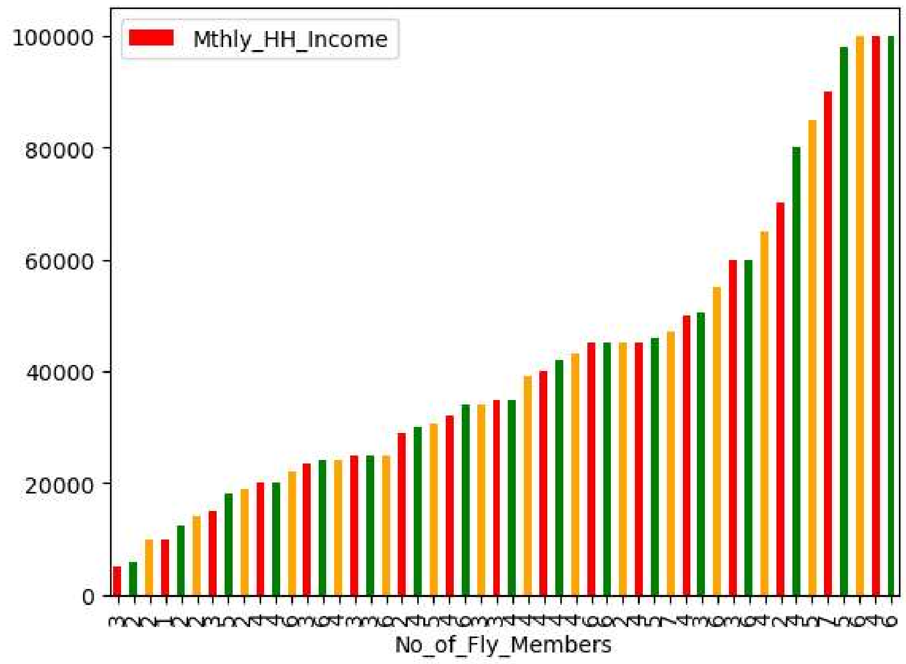


In [62]:

```
Income.plot(x='No_of_Fly_Members', y='Mthly_HH_Income',kind='bar',color=['red','green','orange'])
```

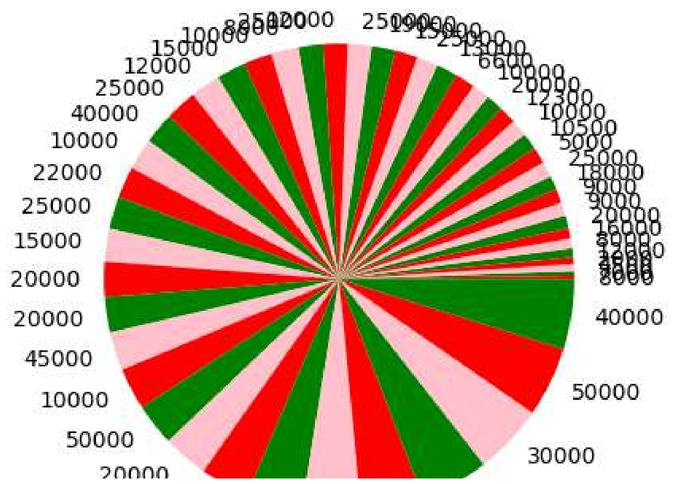
Out[62]:

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



In [65]:

```
plt.pie(Income["Mthly_HH_Income"],labels=Income["Mthly_HH_Expense"],colors = ['red','green','pink'])
```



In [53]:

```
Coeff_of_var_StockA=10/15
print(Coeff_of_var_StockA)
Coeff_of_var_StockB=15/10
print(Coeff_of_var_StockB)
```

```
0.6666666666666666
1.5
```