

# 1. Import Necessary Packages

```
In [11]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

## 2. Load the File

```
In [9]: income_df=pd.read_csv(r"C:\Users\admin\Desktop\Data Science 7pm/12th Sep-Inc_Exp_Da
```

```
In [12]: income_df.head()
```

```
Out[12]:
```

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HI
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	

## 3. Analyze the Data

```
In [10]: 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 [13]: income_df.shape
```

```
Out[13]: (50, 7)
```

```
In [14]: income_df.describe().T
```

Out[14]:

	count	mean	std	min	25%	50%	
<b>Mthly_HH_Income</b>	50.0	41558.00	26097.908979	5000.0	23550.0	35000.0	50
<b>Mthly_HH_Expense</b>	50.0	18818.00	12090.216824	2000.0	10000.0	15500.0	25
<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	3
<b>Annual_HH_Income</b>	50.0	490019.04	320135.792123	64200.0	258750.0	447420.0	594
<b>No_of_Earning_Members</b>	50.0	1.46	0.734291	1.0	1.0	1.0	

In [15]: `income_df.isna().any()`

```
Out[15]: 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
```

## 4. What is the Mean Expense of a Household?

In [18]: `income_df["Mthly_HH_Expense"].mean()`Out[18]: `np.float64(18818.0)`

## 5. What is the Median Household Expense?

In [20]: `income_df["Mthly_HH_Expense"].median()`Out[20]: `15500.0`

## 6. What is the Monthly Expense for Most of the Households?

```
In [24]: mth_exp_tmp = pd.crosstab(index=income_df["Mthly_HH_Expense"], columns="count")
mth_exp_tmp.reset_index(inplace=True)

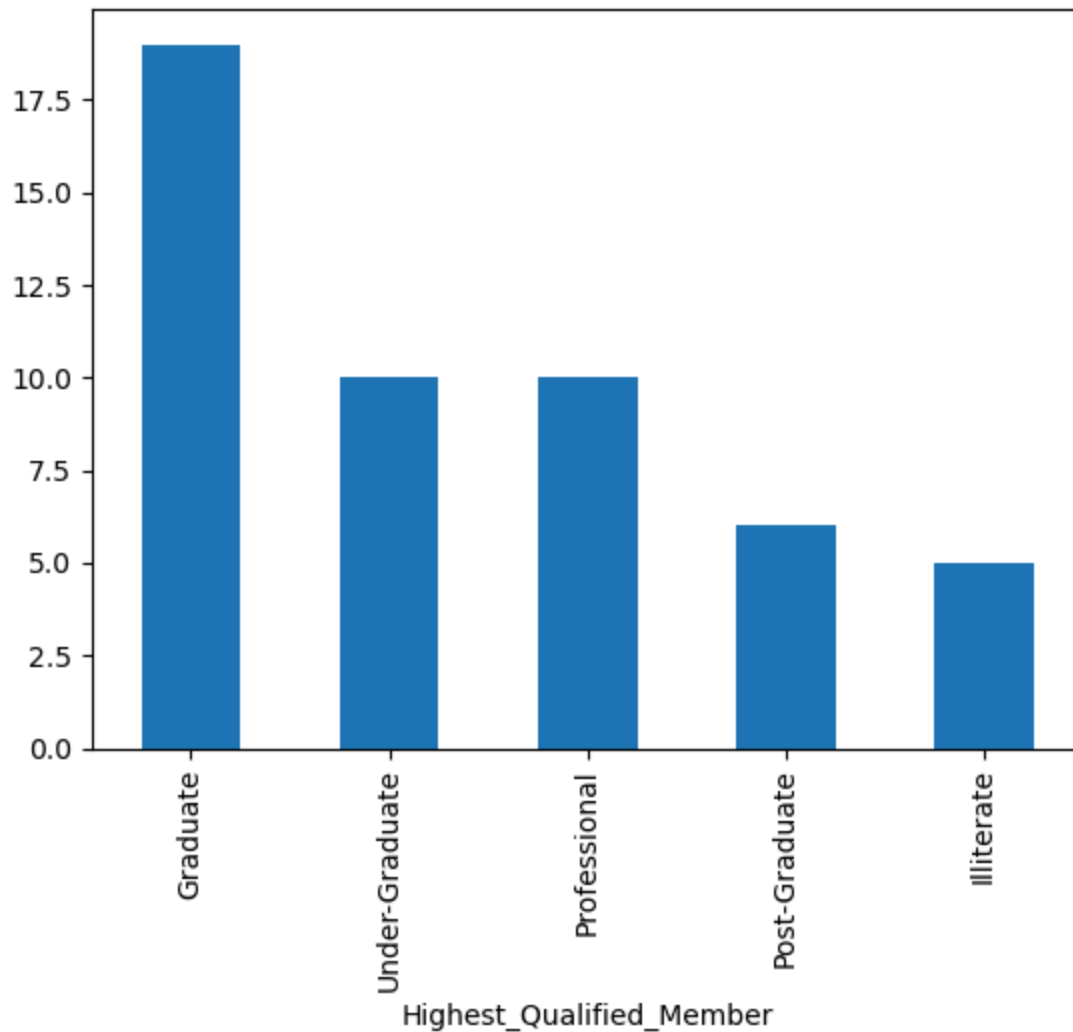
mth_exp_tmp[mth_exp_tmp['count'] == income_df["Mthly_HH_Expense"].value_counts().ma
```

```
Out[24]: col_0  Mthly_HH_Expense  count
         18                25000      8
```

## 7. Plot the Histogram to count the Highest Qualified Member

```
In [25]: income_df["Highest_Qualified_Member"].value_counts().plot(kind="bar")
```

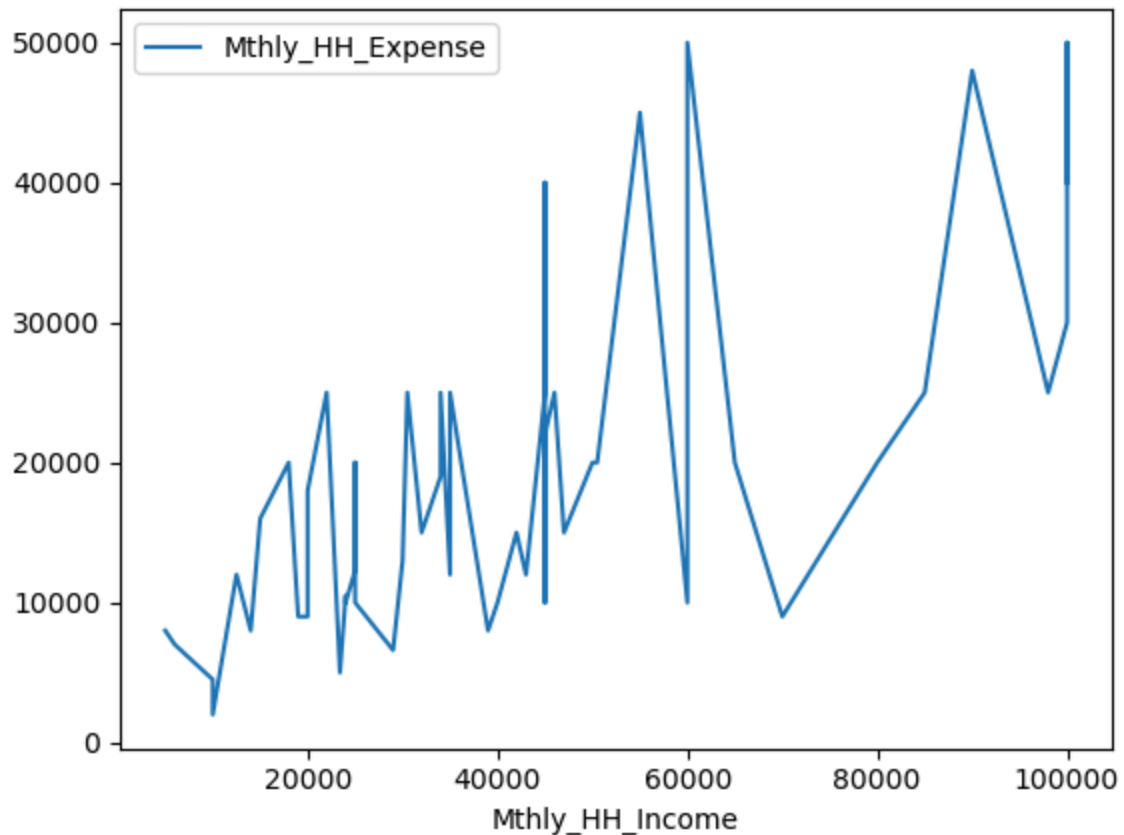
```
Out[25]: <Axes: xlabel='Highest_Qualified_Member'>
```



## 8. Calculate Iqr (different between

```
In [32]: 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"].quan
IQR
```

```
Out[32]: np.float64(15000.0)
```



## 9. Calculate Variance for first 3 Columns.

```
In [33]: pd.DataFrame(income_df.iloc[:,0:5].std().to_frame()).T
```

```
Out[33]:
```

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_Hi
0	26097.908979	12090.216824	1.517382	6241.434948	3201:

## 10. Calculate Variance for first 3 columns.

```
In [35]: pd.DataFrame(income_df.iloc[:,0:4].var().to_frame()).T
```

```
Out[35]:
```

	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

## 11. Calculate the count of Highest Qualified Member.

```
In [36]: income_df["Highest_Qualified_Member"].value_counts().to_frame().T
```

Out[36]:

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

## 12. Plot the Histogram to Count the No\_of\_Earning\_Members

```
In [38]: income_df["No_of_Earning_Members"].value_counts().plot(kind="bar")
```

```
Out[38]: <Axes: xlabel='No_of_Earning_Members'>
```

