

## 1. Import necessary packages

```
In [ ]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

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
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

```
In [24]: df = pd.read_csv(r'/Users/aviswe/Desktop/830/Datasets/Inc_Exp_Data.csv')
```

```
In [25]: df.head()
```

```
Out[25]:
```

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

```
In [26]: df.shape
```

```
Out[26]: (50, 7)
```

```
In [27]: df.describe().T
```

```
Out[27]:
```

	count	mean	std	min	25%	50%	75%
Mthly_HH_Income	50.0	41558.00	26097.908979	5000.0	23550.0	35000.0	50375.0
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	594725.0
No_of_Earning_Members	50.0	1.46	0.734291	1.0	1.0	1.0	2.0

```
In [28]: df.isna().any()
```

```
Out[28]: 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
```

## what is mean expenses of a Household?

```
In [7]: df["Mthly_HH_Expense"].mean()
```

```
Out[7]: 18818.0
```

1. What is the Median Household Expense?

```
In [8]: df["Mthly_HH_Expense"].median()
```

```
Out[8]: 15500.0
```

What is Monthly Expense for most of the Household

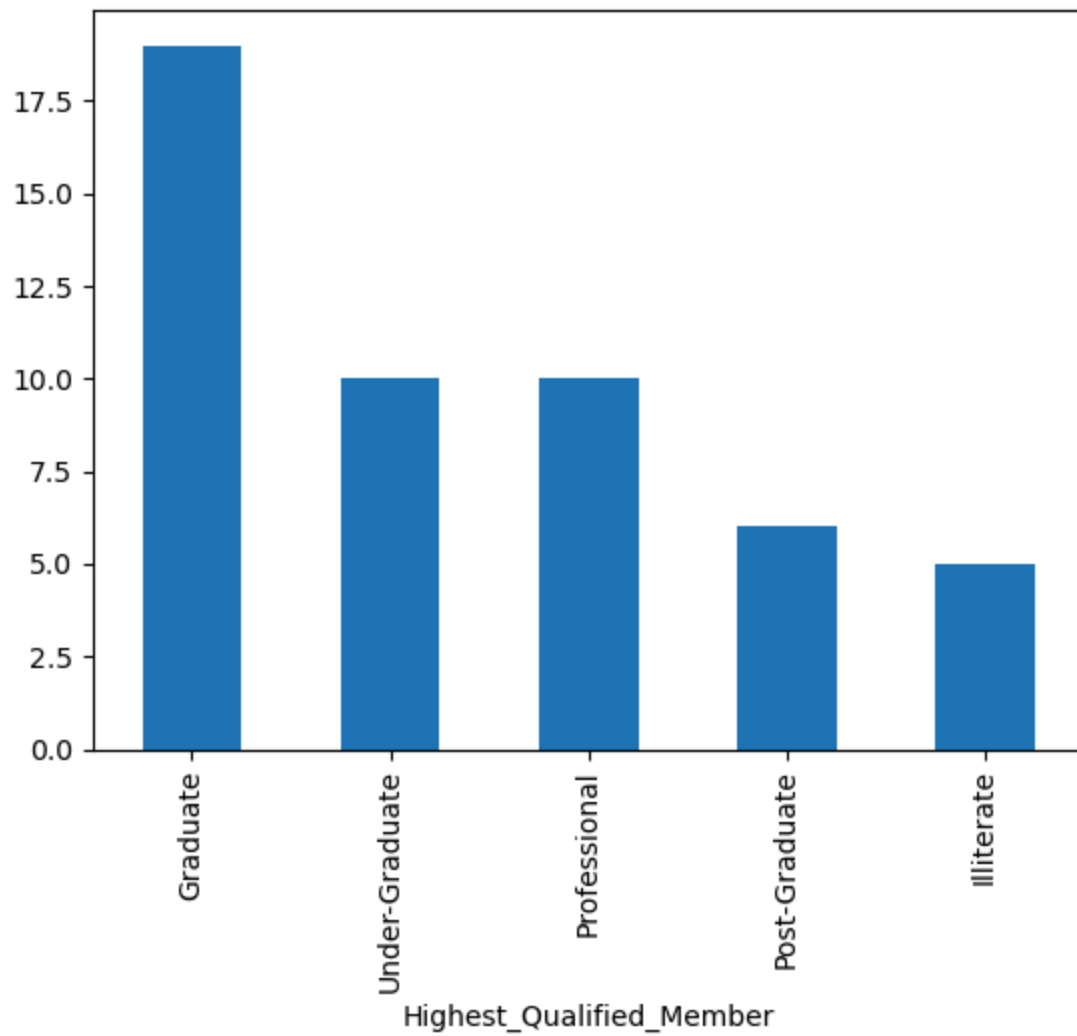
```
In [9]: mth_exp_tmp = pd.crosstab(index=df["Mthly_HH_Expense"], columns="count")
         mth_exp_tmp.reset_index(inplace=True)
         mth_exp_tmp[mth_exp_tmp['count']==df.Mthly_HH_Expense.value_counts().max()]
```

```
Out[9]: col_0  Mthly_HH_Expense  count
         18                25000      8
```

7. Plot the Histogram to count the Highest qualified memeber

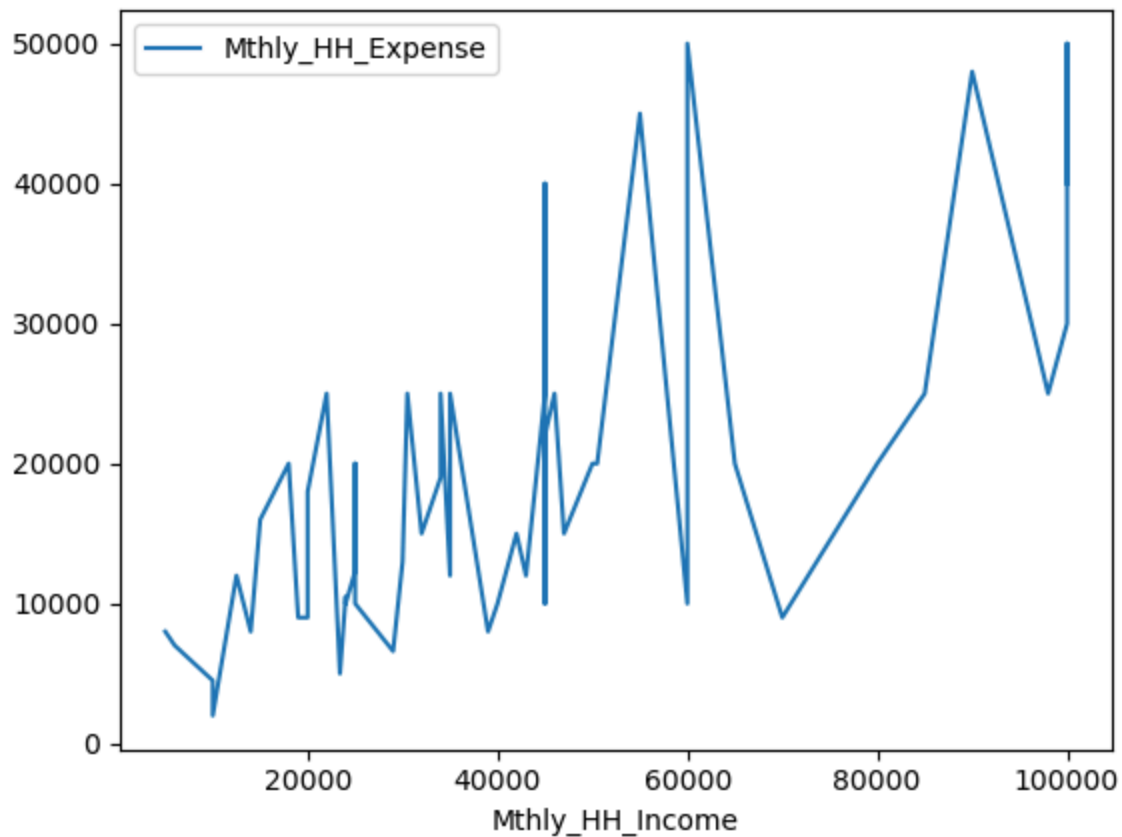
```
In [10]: df["Highest_Qualified_Member"].value_counts().plot(kind="bar")
```

```
Out[10]: <Axes: xlabel='Highest_Qualified_Member'>
```



8 Calculate the IQR (difference between 75% and 25%)

```
In [11]: df.plot(x="Mthly_HH_Income",y="Mthly_HH_Expense")  
IQR=df["Mthly_HH_Expense"].quantile(0.75)-df["Mthly_HH_Expense"].quantile(0.25)
```



1. Calculate Standard Deviation for first 4 columns

```
In [14]: #pd.DataFrame(df.iloc[:,0.5].std().to_frame())
pd.DataFrame(df.iloc[:,0:5].std().to_frame()).T
```

```
Out[14]:
```

	Mthly_HH_Income	Mthly_HH_Expense	No_of_Fly_Members	Emi_or_Rent_Amt	Annual_HH_Ir
0	26097.908979	12090.216824	1.517382	6241.434948	320135.7

10. Calculate Variance for first 3 columns

```
In [17]: #pd.DataFrame(df.iloc[:,0.4].var().to_frame()).T
pd.DataFrame(df.iloc[:,0:4].var().to_frame()).T
```

```
Out[17]:
```

	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 [18]: df["Highest_Qualified_Member"].value_counts().to_frame().T
```

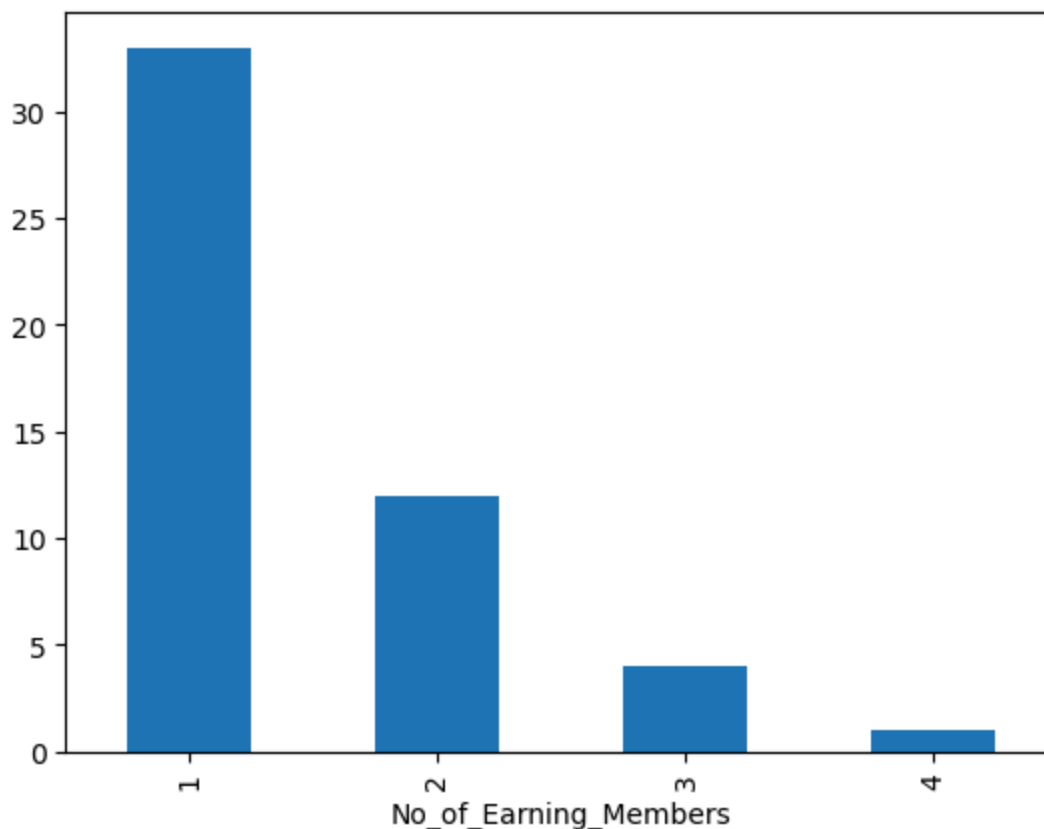
```
Out[18]:
```

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 [19]: df["No_of_Earning_Members"].value_counts().plot(kind="bar")
```

```
Out[19]: <Axes: xlabel='No_of_Earning_Members'>
```



13. Suppose you have option to invest in Stock A or Stock B. The stocks have different expected returns and standard deviations. The expected return of Stock A is 15% and Stock B is 10%. Standard Deviation of the returns of these stocks is 10% and 5% respectively.

```
In [20]: Coeff_of_var_StockA=10/15  
print(Coeff_of_var_StockA)  
Coeff_of_var_StockB=5/10  
print(Coeff_of_var_StockB)
```

```
0.6666666666666666  
0.5
```

```
In [ ]:
```