In [53]:

```
#sampling
import numpy as np
import pandas as pd
```

In [60]:

```
df=pd.read_csv("sampling.csv")
```

In [61]:

```
df.head()
```

Out[61]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distanc of hous from \ institutio
0	1	M	22	4	100000	3.30	6.0	100	
1	2	M	18	4	60000	3.20	1.5	80	1
2	3	M	16	4	50000	3.43	2.0	90	
3	4	M	16	5	80000	2.58	1.0	90	
4	5	M	16	4	200000	2.99	1.0	90	

In [62]:

```
df.describe()
```

Out[62]:

	ID \n Number	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	i
count	49.00000	49.000000	49.000000	49.000000	49.000000	49.000000	49.000000	4
mean	25.00000	16.571429	4.632653	105306.122449	3.260204	2.346939	89.061224	5
std	14.28869	1.080123	1.481748	51310.007179	0.337259	1.287779	10.749822	1(
min	1.00000	16.000000	2.000000	30000.000000	2.580000	1.000000	60.000000	
25%	13.00000	16.000000	4.000000	70000.000000	3.100000	1.000000	85.000000	
50%	25.00000	16.000000	4.000000	100000.000000	3.200000	2.000000	90.000000	1
75%	37.00000	17.000000	5.000000	140000.000000	3.430000	3.000000	97.000000	3
max	49.00000	22.000000	10.000000	200000.000000	4.190000	6.000000	100.000000	5(

In [63]:

```
bins = np.linspace(min(df['Monthly Family Income']), max(df['Monthly Family Income']
groupNames = ["Stratum_1", "Stratum_2", "Stratum_3","Stratum_4"]
df['Strata'] = pd.cut(df['Monthly Family Income'], bins, labels = groupNames, includ
df.head()
```

Out[63]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distanc of hous from \ institutio
0	1	M	22	4	100000	3.30	6.0	100	
1	2	M	18	4	60000	3.20	1.5	80	1
2	3	M	16	4	50000	3.43	2.0	90	
3	4	M	16	5	80000	2.58	1.0	90	
4	5	M	16	4	200000	2.99	1.0	90	

In [73]:

```
Stratum_1=df[df['Strata']=='Stratum_1']
Stratum_1
```

Out[73]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
1	2	M	18	4	60000	3.20	1.5	80	
2	3	M	16	4	50000	3.43	2.0	90	
11	12	M	18	5	70000	3.45	3.5	99	
14	15	M	16	5	70000	3.58	5.0	95	
15	16	M	16	5	70000	3.42	2.0	90	
16	17	M	16	2	60000	4.19	4.0	90	
18	19	M	16	4	50000	2.72	2.0	90	2
20	21	F	18	6	60000	3.19	2.0	98	
21	22	M	16	5	30000	2.86	2.0	97	
25	26	M	17	5	50000	3.11	3.0	85	1
31	32	M	16	4	50000	2.97	1.0	60	
33	34	M	16	4	50000	3.93	4.0	85	
35	36	M	16	4	50000	3.02	1.0	60	
38	39	F	16	4	70000	3.50	4.0	98	1
43	44	M	16	4	50000	2.70	2.0	97	

In [76]:

```
Stratum_2=df[df['Strata']=='Stratum_2']
Stratum_2
```

Out[76]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
0	1	M	22	4	100000	3.30	6.0	100	
3	4	M	16	5	80000	2.58	1.0	90	
7	8	M	16	4	100000	3.28	3.5	95	3
8	9	M	16	5	80000	3.16	4.5	90	5
9	10	M	16	10	100000	3.10	3.0	90	
10	11	M	16	4	100000	3.26	4.5	100	2
12	13	M	18	4	100000	3.14	1.5	99	
19	20	M	18	5	110000	3.10	2.0	100	
22	23	M	16	5	80000	2.69	2.0	90	
24	25	M	17	4	80000	3.22	1.0	90	1
29	30	M	17	3	80000	3.14	1.0	70	
36	37	M	16	4	75000	2.80	2.0	70	
37	38	M	17	2	80000	3.10	2.0	75	
39	40	F	16	5	100000	3.60	3.0	99	
40	41	F	16	6	80000	3.40	2.0	97	1
41	42	M	17	4	100000	3.10	2.0	98	
45	46	M	16	6	100000	3.40	2.0	95	
46	47	M	16	5	75000	3.10	1.0	95	
47	48	M	16	4	100000	3.40	1.0	90	

In [77]:

```
Stratum_3=df[df['Strata']=='Stratum_3']
Stratum_3
```

Out[77]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
5	6	M	16	6	140000	3.25	1.0	70	
17	18	M	18	4	150000	3.42	2.0	98	
23	24	M	18	5	120000	3.48	4.0	97	
28	29	M	16	4	130000	2.90	1.0	80	
30	31	M	16	4	150000	3.80	1.0	90	2

In [78]:

```
Stratum_4=df[df['Strata']=='Stratum_4']
Stratum_4
```

Out[78]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
4	5	M	16	4	200000	2.99	1.0	90	
6	7	M	16	3	170000	3.15	1.0	80	
13	14	M	16	3	200000	3.43	4.0	98	
26	27	M	17	4	200000	3.96	3.0	95	
27	28	M	17	6	180000	3.12	1.0	80	
32	33	M	16	10	200000	3.20	2.0	65	
34	35	M	16	4	160000	3.21	1.0	90	
42	43	M	16	5	200000	3.70	3.0	95	
44	45	M	17	4	200000	3.20	2.0	90	
48	49	M	16	7	200000	3.80	4.0	99	

In [105]:

```
#exporting_file
Stratum_4.to_excel(r'Stratum_1.xlsx', index=False)
Stratum_4.to_excel(r'Stratum_2.xlsx', index=False)
Stratum_4.to_excel(r'Stratum_3.xlsx', index=False)
Stratum_4.to_excel(r'Stratum_4.xlsx', index=False)
```

In [81]:

```
df['Strata'].value_counts()
```

Out[81]:

```
Stratum_2    19
Stratum_1    15
Stratum_4    10
Stratum_3     5
Name: Strata, dtype: int64
```

In [29]:

```
population_size=49
sample_size=10
sample_propotion=sample_size/population_size
sample_propotion
```

Out[29]:

```
0.20408163265306123
```

In [30]:

```
Stratum_2=19
Stratum_1=15
Stratum_4=10
Stratum_3=5
```

In [34]:

```
stratum_1=round(Stratum_1*sample_propotion)
stratum_2=round(Stratum_2*sample_propotion)
stratum_3=round(Stratum_3*sample_propotion)
stratum_4=round(Stratum_4*sample_propotion)
```

In [82]:

```
print(f"**Propotional Allocation**\nstratum_1= {stratum_1}\nstratum_2= {stratum_2}\r
```

```
**Propotional Allocation**
stratum_1= 3
stratum_2= 4
stratum_3= 1
stratum_4= 2
```

In [90]:

```
# random sampling in Stratum_1
stratum_1=Stratum_1.sample(n = 3,random_state = 0)
stratum_1
```

Out[90]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
2	3	M	16	4	50000	3.43	2.0	90	
18	19	M	16	4	50000	2.72	2.0	90	2
21	22	M	16	5	30000	2.86	2.0	97	

In [112]:

```
# random sampling in Stratum_2
stratum_2=Stratum_2.sample(n =4,random_state = 2)
stratum_2
```

Out[112]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
24	25	M	17	4	80000	3.22	1.0	90	1
9	10	M	16	10	100000	3.10	3.0	90	
40	41	F	16	6	80000	3.40	2.0	97	1
0	1	M	22	4	100000	3.30	6.0	100	

In [92]:

```
# random sampling in Stratum_3
stratum_3=Stratum_3.sample(n =1,random_state = 2)
stratum_3
```

Out[92]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
23	24	M	18	5	120000	3.48	4.0	97	

In [93]:

```
# random sampling in Stratum_4
stratum_4=Stratum_4.sample(n =2,random_state = 2)
stratum_4
```

Out[93]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distan of hou from instituti
27	28	M	17	6	180000	3.12	1.0	80	
6	7	M	16	3	170000	3.15	1.0	80	

In [116]:

```
#sample Dataframe
sample=pd.DataFrame()

sample=sample.append([stratum_1,stratum_2,stratum_3,stratum_4], ignore_index=True)
sample
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_4104\3917195880.py:4: FutureWar
ning: The frame.append method is deprecated and will be removed from panda
s in a future version. Use pandas.concat instead.
  sample=sample.append([stratum_1,stratum_2,stratum_3,stratum_4], ignore_i
ndex=True)
```

Out[116]:

	ID \n Number	Gender	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	Distanc of hous from \ institutio
0	3	M	16	4	50000	3.43	2.0	90	
1	19	M	16	4	50000	2.72	2.0	90	25
2	22	M	16	5	30000	2.86	2.0	97	
3	25	M	17	4	80000	3.22	1.0	90	12
4	10	M	16	10	100000	3.10	3.0	90	
5	41	F	16	6	80000	3.40	2.0	97	15
6	1	M	22	4	100000	3.30	6.0	100	
7	24	M	18	5	120000	3.48	4.0	97	
8	28	M	17	6	180000	3.12	1.0	80	1
9	7	M	16	3	170000	3.15	1.0	80	

In [117]:

```
1 sample.describe()
```

Out[117]:

	ID \n Number	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	18.000000	17.000000	5.100000	96000.000000	3.178000	2.400000	91.100000
std	12.605114	1.885618	1.969207	49710.271686	0.245348	1.577621	6.951419
min	1.000000	16.000000	3.000000	30000.000000	2.720000	1.000000	80.000000
25%	7.750000	16.000000	4.000000	57500.000000	3.105000	1.250000	90.000000
50%	20.500000	16.000000	4.500000	90000.000000	3.185000	2.000000	90.000000
75%	24.750000	17.000000	5.750000	115000.000000	3.375000	2.750000	97.000000
max	41.000000	22.000000	10.000000	180000.000000	3.480000	6.000000	100.000000

In [122]:

```
1 freq_table_1 = pd.crosstab(sample['Highest Years of schooling'], sample['Monthly Fam
2
3 freq_table_2
```

Out[122]:

Monthly Family Income	30000	50000	80000	100000	120000	170000	180000
Highest Years of schooling							
16	1	2	1	1	0	1	0
17	0	0	1	0	0	0	1
18	0	0	0	0	1	0	0
22	0	0	0	1	0	0	0

In [127]:

```
freq_table_2 = pd.crosstab(sample['Total Family Member'], sample['Monthly Family Inc

freq_table_2
```

Out[127]:

Monthly Family Income	30000	50000	80000	100000	120000	170000	180000
Total Family Member							
3	0	0	0	0	0	1	0
4	0	2	1	1	0	0	0
5	1	0	0	0	1	0	0
6	0	0	1	0	0	0	1
10	0	0	0	1	0	0	0

In [126]:

```
freq_table_3 = pd.crosstab(sample['Daily hour studied'],sample['GPA in Graduation'])

freq_table_3
```

Out[126]:

GPA in Graduation	2.72	2.86	3.10	3.12	3.15	3.22	3.30	3.40	3.43	3.48
Daily hour studied										
1.0	0	0	0	1	1	1	0	0	0	0
2.0	1	1	0	0	0	0	0	1	1	0
3.0	0	0	1	0	0	0	0	0	0	0
4.0	0	0	0	0	0	0	0	0	0	1
6.0	0	0	0	0	0	0	1	0	0	0

In [130]:

```
freq_table_4 = pd.crosstab(sample['% of attendance'],sample['GPA in Graduation'])

freq_table_4
```

Out[130]:

GPA in Graduation	2.72	2.86	3.10	3.12	3.15	3.22	3.30	3.40	3.43	3.48
% of attendance										
80	0	0	0	1	1	0	0	0	0	0
90	1	0	1	0	0	1	0	0	1	0
97	0	1	0	0	0	0	0	1	0	1
100	0	0	0	0	0	0	1	0	0	0

In [161]:

```
freq1=sample['Total Family Member'].value_counts()
freq1
freq1=pd.DataFrame(freq1)
freq1
```

Out[161]:

	Total Family Member
4	4
5	2
6	2
10	1
3	1

In [158]:

```
freq2=sample['Monthly Family Income'].value_counts()
freq2=pd.DataFrame(freq2)
freq2
```

Out[158]:

	Monthly Family Income
50000	2
80000	2
100000	2
30000	1
120000	1
180000	1
170000	1

In [162]:

```
freq3=sample['Highest Years of schooling'].value_counts()
freq3=pd.DataFrame(freq3)
freq3
```

Out[162]:

	Highest Years of schooling
16	6
17	2
22	1
18	1

In [163]:

```
freq4=sample['Daily hour studied'].value_counts()
freq4=pd.DataFrame(freq4)
freq4
```

Out[163]:

	Daily hour studied
2.0	4
1.0	3
3.0	1
6.0	1
4.0	1

In [167]:

```
freq5=sample['% of attendance'].value_counts()
freq5=pd.DataFrame(freq5)
freq5
```

Out[167]:

	% of attendance
90	4
97	3
80	2
100	1

In [168]:

```
freq6=sample['Working status (Yes/No)'].value_counts()
freq6=pd.DataFrame(freq6)
freq6
```

Out[168]:

	Working status (Yes/No)
Y	8
N	2

In [182]:

```
freq7=sample['Gender'].value_counts()
freq7=pd.DataFrame(freq7)
freq7
```

Out[182]:

	Gender
M	9
F	1

In [172]:

```
freq8=sample['Distance of house from \n institution'].value_counts()
freq8=pd.DataFrame(freq8)
freq8
```

Out[172]:

	Distance of house from \n institution
5	2
1	2
8	1
250	1
122	1
150	1
10	1
2	1

In [174]:

```
sample.columns
```

Out[174]:

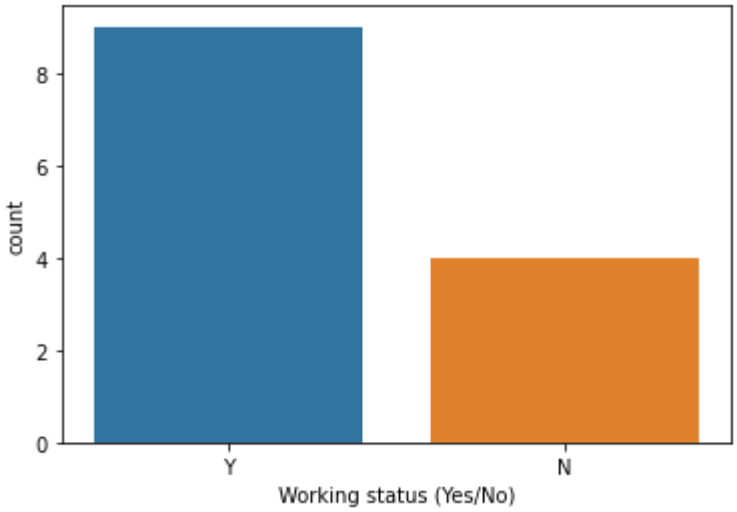
```
Index(['ID \n Number', 'Gender', 'Highest Years of schooling',
       'Total Family Member', 'Monthly Family Income', 'GPA in Graduatio
n',
       'Daily hour studied', '% of attendance',
       'Distance of house from \n institution', 'Working status (Yes/No)',
       'Strata'],
      dtype='object')
```

In [195]:

```
#graphs
import matplotlib.pyplot as plt
import seaborn as sns
```

In [212]:

```
Columns=[ 'Gender', 'Highest Years of schooling',
       'Total Family Member', 'Monthly Family Income', 'GPA in Graduation',
       'Daily hour studied', '% of attendance', 'Working status (Yes/No)']
for name in Columns:
    fig=sns.countplot(data=sample,x=name)
    if name=='Working status (Yes/No)':
        name='working status'
        plt.savefig("fig{name}.png".format(name=name),transparent=False)
    else:
        plt.savefig("fig{name}.png".format(name=name),transparent=False)
```



In [220]:

```
import statistics as st
Columns=['Highest Years of schooling',
       'Total Family Member', 'Monthly Family Income', 'GPA in Graduation',
       'Daily hour studied', '% of attendance','Distance of house from \n institutic
for name in Columns:
    var=st.variance(sample[name])
    print(f"Variance of {name}={var}")
```

```
Variance of Highest Years of schooling=3.5555555555555554
Variance of Total Family Member=3.8777777777777778
Variance of Monthly Family Income=2471111111.111111
Variance of GPA in Graduation=0.060195555555555544
Variance of Daily hour studied=2.488888888888889
Variance of % of attendance=48.32222222222222
Variance of Distance of house from
 institution=7712.488888888888
```

In [223]:

```
Columns=['Highest Years of schooling',
       'Total Family Member', 'Monthly Family Income', 'GPA in Graduation',
       'Daily hour studied', '% of attendance','Distance of house from \n institutic
for name in Columns:
    std=np.sqrt(st.variance(sample[name]))
    print(f"Standard Deviation of {name}={var}")
```

```
Standard Deviation of Highest Years of schooling=87.82077709112399
Standard Deviation of Total Family Member=87.82077709112399
Standard Deviation of Monthly Family Income=87.82077709112399
Standard Deviation of GPA in Graduation=87.82077709112399
Standard Deviation of Daily hour studied=87.82077709112399
Standard Deviation of % of attendance=87.82077709112399
Standard Deviation of Distance of house from
 institution=87.82077709112399
```

In [228]:

```
Columns=['Highest Years of schooling',
       'Total Family Member', 'Monthly Family Income', 'GPA in Graduation',
       'Daily hour studied', '% of attendance','Distance of house from \n institutic
for name in Columns:
    mean=st.mean(sample[name])
    std=np.sqrt(st.variance(sample[name]))
    cv=std/mean
    print(f"Coeficient Variance of {name}={var}")
```

```
Coeficient Variance of Highest Years of schooling=87.82077709112399
Coeficient Variance of Total Family Member=87.82077709112399
Coeficient Variance of Monthly Family Income=87.82077709112399
Coeficient Variance of GPA in Graduation=87.82077709112399
Coeficient Variance of Daily hour studied=87.82077709112399
Coeficient Variance of % of attendance=87.82077709112399
Coeficient Variance of Distance of house from
 institution=87.82077709112399
```

In [232]:

```
Columns=['Highest Years of schooling',
  'Total Family Member', 'Monthly Family Income', 'GPA in Graduation',
  'Daily hour studied', '% of attendance','Distance of house from \n institution']
for name in Columns:
    pearsons_coefficient = np.corrcoef(sample['Monthly Family Income'], sample[name])
    print(f"The pearson's coeffient of the Monthly Family Income  and {name} are: {pearso
```

```
The pearson's coeffient of the Monthly Family Income  and Highest Years of
schooling are: [[1.         0.16595321]
 [0.16595321 1.        ]]
The pearson's coeffient of the Monthly Family Income  and Total Family Mem
ber are: [[1.         0.04994276]
 [0.04994276 1.        ]]
The pearson's coeffient of the Monthly Family Income  and Monthly Family I
ncome are: [[1. 1.]
 [1. 1.]]
The pearson's coeffient of the Monthly Family Income  and GPA in Graduatio
n are: [[1.         0.24342523]
 [0.24342523 1.        ]]
The pearson's coeffient of the Monthly Family Income  and Daily hour studi
ed are: [[ 1.         -0.10484321]
 [-0.10484321  1.        ]]
The pearson's coeffient of the Monthly Family Income  and % of attendance
are: [[ 1.         -0.62572101]
 [-0.62572101  1.        ]]
The pearson's coeffient of the Monthly Family Income  and Distance of hous
e from
 institution are: [[ 1.        -0.3948552]
 [-0.3948552  1.       ]]
```

In [237]:

```
1 sample_corr=sample.corr()
2 sample_corr.to_excel(r'sample_correlation.xlsx', index=False)
3 sample_corr
```

Out[237]:

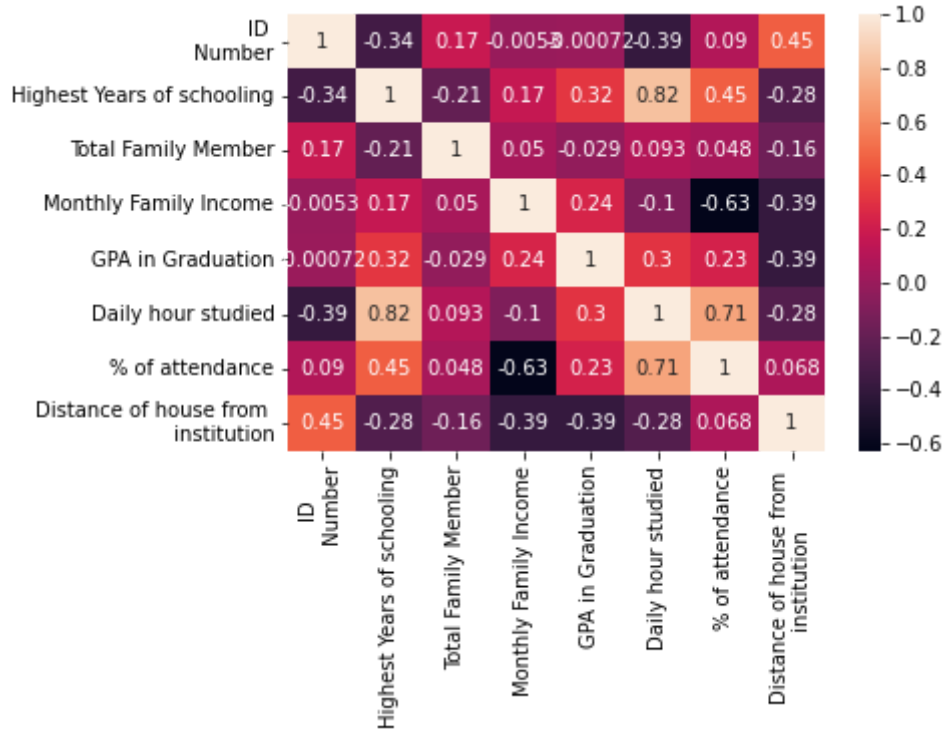
	ID \n Number	Highest Years of schooling	Total Family Member	Monthly Family Income	GPA in Graduation	Daily hour studied	% of attendance	in
ID \n Number	1.000000	-0.341256	0.174576	-0.005320	-0.000719	-0.391116	0.090032	(
Highest Years of schooling	-0.341256	1.000000	-0.209464	0.165953	0.317026	0.821720	0.449269	-(
Total Family Member	0.174576	-0.209464	1.000000	0.049943	-0.029437	0.092990	0.047890	-(
Monthly Family Income	-0.005320	0.165953	0.049943	1.000000	0.243425	-0.104843	-0.625721	-(
GPA in Graduation	-0.000719	0.317026	-0.029437	0.243425	1.000000	0.303709	0.229452	-(
Daily hour studied	-0.391116	0.821720	0.092990	-0.104843	0.303709	1.000000	0.705165	-(
% of attendance	0.090032	0.449269	0.047890	-0.625721	0.229452	0.705165	1.000000	(
Distance of house from \n institution	0.449768	-0.277784	-0.164093	-0.394855	-0.393214	-0.276358	0.067816	

In [241]:

```
1 sns.heatmap(sample_corr,annot=True)
```

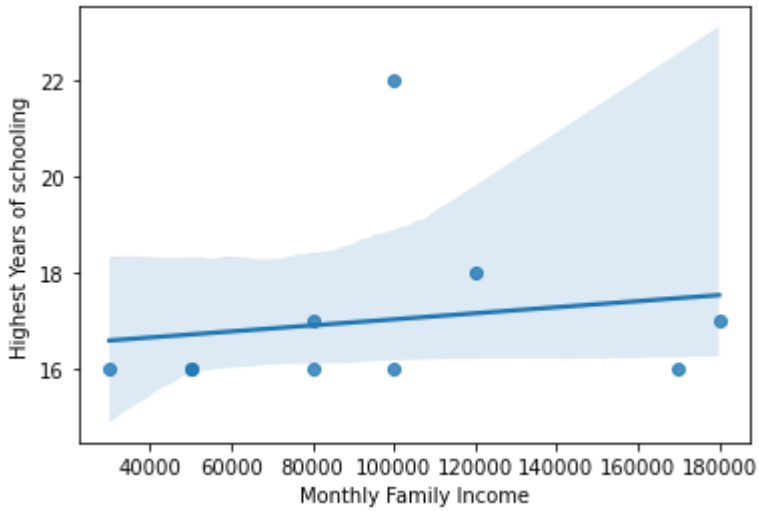
Out[241]:

<AxesSubplot:>



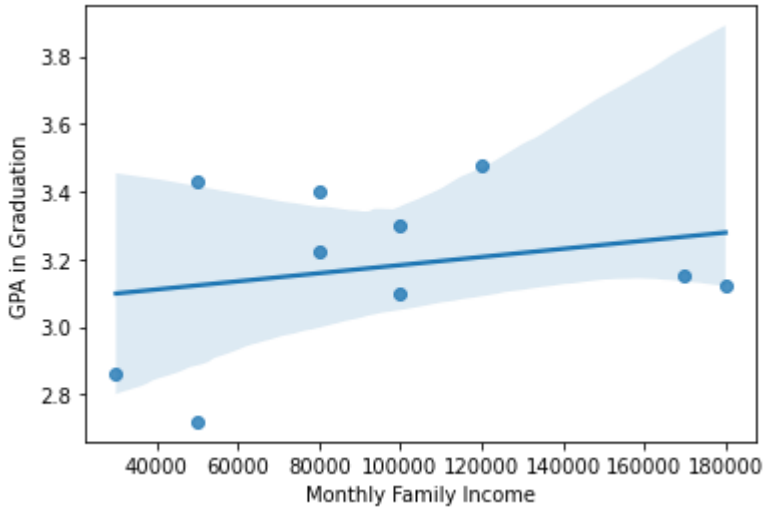
In [257]:

```
1 fig=sns.regplot(data=sample,x='Monthly Family Income',y='Highest Years of schooling'
2 plt.savefig("Regression_HighestYear.png",transparent=False)
```



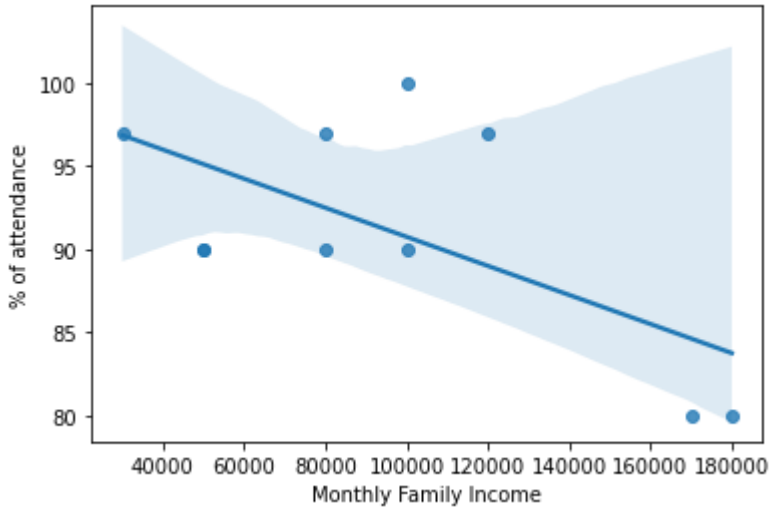
In [255]:

```
fig=sns.regplot(data=sample,x='Monthly Family Income',y='GPA in Graduation')
plt.savefig("Regression_GPA.png",transparent=False)
```



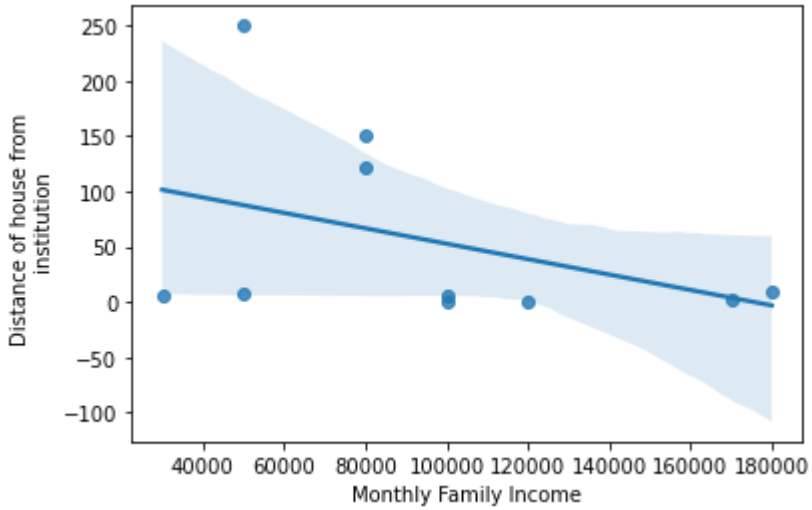
In [256]:

```
fig=sns.regplot(data=sample,x='Monthly Family Income',y='% of attendance')
plt.savefig("Regression_attendance.png",transparent=False)
```



In [260]:

```
fig=sns.regplot(data=sample,x='Monthly Family Income',y='Distance of house from \n i
plt.savefig("Regression_Distance.png",transparent=False)
```



In [ ]:

```
1
```