Data Analysis and Visualization:

Programmer: Souparna Bose

Importing required libraries

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import scipy as sipi
import matplotlib.pyplot as plt

%matplotlib inline
```

Loading dataset into a pandas DataFrame

In [2]:
```
df = pd.read_csv('trainingData.csv')
```

Show important information about the dataset

In [3]:
```
df.columns
```

Out[3]:
```
Index(['Id', 'city', 'age', 'sex', 'social_class', 'primary_business',
       'secondary_business', 'annual_income', 'monthly_expenses',
       'old_dependents', 'young_dependents', 'home_ownership', 'type_of_house',
       'occupants_count', 'house_area', 'sanitary_availability',
       'water_availabity', 'loan_purpose', 'loan_tenure', 'loan_installments',
       'loan_amount'],
      dtype='object')
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
Data columns (total 21 columns):
Id                      40000 non-null int64
city                    38136 non-null object
age                     40000 non-null int64
sex                     40000 non-null object
social_class            34745 non-null object
primary_business        39974 non-null object
secondary_business      34759 non-null object
annual_income           40000 non-null float64
monthly_expenses        39880 non-null float64
old_dependents          40000 non-null int64
young_dependents        40000 non-null int64
home_ownership          39621 non-null float64
type_of_house           39306 non-null object
occupants_count         40000 non-null int64
house_area              40000 non-null float64
sanitary_availability   39792 non-null float64
water_availabity        34747 non-null float64
loan_purpose            39974 non-null object
loan_tenure             40000 non-null int64
loan_installments       40000 non-null int64
loan_amount             40000 non-null float64
dtypes: float64(7), int64(7), object(7)
memory usage: 6.4+ MB
```

In [5]:
```
df.dtypes
```

Out[5]:
```
Id                        int64
city                     object
age                       int64
sex                      object
social_class             object
primary_business         object
secondary_business       object
annual_income           float64
monthly_expenses        float64
old_dependents            int64
young_dependents          int64
home_ownership          float64
type_of_house            object
occupants_count           int64
house_area              float64
sanitary_availability   float64
water_availabity        float64
loan_purpose             object
loan_tenure               int64
loan_installments         int64
loan_amount             float64
dtype: object
```

From above, we can see that there are 3 formats of data types:

object: Object format indicates variables are categorical. Categorical variables in the dataset are city, sex, social_class, primary_business, secondary_business, type_of_house, loan_purpose.

int64: This represents integer variables. Integer variables in the dataset are Id, age, old_dependents, young_dependents, occupants_count, loan_tenure, loan_installments.

float64: This represents variables that have some decimal values involved. Float variables in the dataset are annual-income, monthly_expenses, home_ownership, house_area, sanitary_availability, water_availability, loan_amount.

Display all columns in the pandas DataFrame:

In [6]:
```
pd.set_option('display.max_columns',None)
```

In [7]:
```
df.head()
```

Out[7]:

	Id	city	age	sex	social_class	primary_business	secondary_business	annual_income	monthly_
0	1	Dhanbad	22	F	Mochi	Tailoring	Others	36000.0	
1	2	Manjapra	21	F	OBC	Tailoring	none	94000.0	
2	3	Dhanbad	24	M	Nai	Beauty salon	Others	48000.0	
3	4	NaN	26	F	OBC	Tailoring	none	7000.0	
4	5	Nuapada	23	F	OBC	General store	Agriculture	36000.0	

In [8]:
```
df.tail()
```

Out[8]:

	Id	city	age	sex	social_class	primary_business	secondary_business	annual_income	mo

	Id	city	age	sex	social_class	primary_business	secondary_business	annual_income	mo
39995	39996	Pusad	45	F	Muslim	Buffalo rearing	none	78000.0	
39996	39997	Pusad	35	F	ST	Tailoring	none	48000.0	
39997	39998	Pusad	35	F	Sc	Goat rearing	none	48000.0	
39998	39999	Pusad	28	F	Sc	Goat rearing	none	48000.0	
39999	40000	Pusad	32	F	Sc	Goat rearing	none	72000.0	

Check co-relation among various variables:

In [9]:
```
df.shape
```

Out[9]: (40000, 21)

In [10]:
```
df.describe()
```

Out[10]:

	Id	age	annual_income	monthly_expenses	old_dependents	young_depender
count	40000.00000	40000.00000	4.000000e+04	39880.000000	40000.000000	40000.000
mean	20000.50000	55.15990	3.764021e+04	3810.875401	0.044900	1.137
std	11547.14972	3830.35566	2.873912e+04	4592.958009	0.222003	1.073
min	1.00000	2.00000	0.000000e+00	2.000000	0.000000	0.000
25%	10000.75000	29.00000	1.440000e+04	2500.000000	0.000000	0.000
50%	20000.50000	35.00000	3.600000e+04	3500.000000	0.000000	1.000
75%	30000.25000	42.00000	5.600000e+04	4000.000000	0.000000	2.000
max	40000.00000	766105.00000	1.200000e+06	240000.000000	3.000000	7.000

In [11]:
```
df.corr()
```

Out[11]:

	Id	age	annual_income	monthly_expenses	old_dependents	young_de
Id	1.000000	-0.004114	0.472447	-0.021413	0.044053	
age	-0.004114	1.000000	-0.006414	-0.003101	-0.000691	
annual_income	0.472447	-0.006414	1.000000	0.112499	0.062216	
monthly_expenses	-0.021413	-0.003101	0.112499	1.000000	-0.003522	
old_dependents	0.044053	-0.000691	0.062216	-0.003522	1.000000	
young_dependents	0.109523	-0.005837	0.239864	0.028754	-0.093778	
home_ownership	0.095202	0.000937	0.011885	-0.047173	0.008586	

	Id	age	annual_income	monthly_expenses	old_dependents	young_de
occupants_count	0.007440	-0.000031	0.003999	0.001320	-0.000987	
house_area	0.037266	-0.000586	0.033902	-0.008270	0.010852	
sanitary_availability	0.003357	-0.007487	0.241509	0.059819	0.029027	
water_availabity	0.433107	-0.001627	0.280939	0.078061	-0.017931	
loan_tenure	-0.062596	-0.000233	-0.027618	-0.013020	-0.022390	
loan_installments	-0.225166	-0.003040	-0.119936	0.113914	-0.033921	
loan_amount	0.141249	-0.001969	0.085632	0.019569	0.006997	

Checking for missing data

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
Id                         0
city                    1864
age                        0
sex                        0
social_class            5255
primary_business          26
secondary_business      5241
annual_income              0
monthly_expenses         120
old_dependents             0
young_dependents           0
home_ownership           379
type_of_house            694
occupants_count            0
house_area                 0
sanitary_availability    208
water_availabity        5253
loan_purpose              26
loan_tenure                0
loan_installments          0
loan_amount                0
dtype: int64
```

Visualizing individual features: Uni/Bi/Multivariate data analysis

In [13]:
```
df.head(2)
```

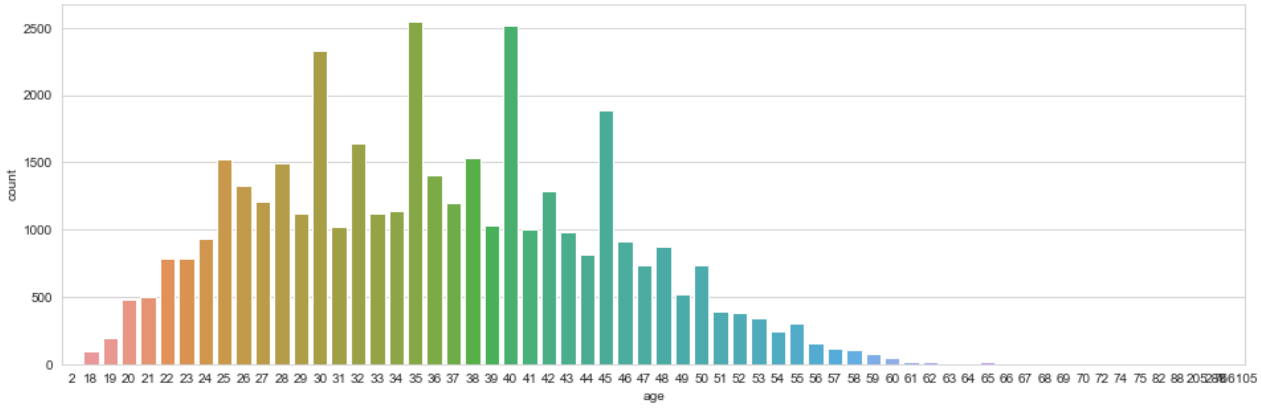
Out[13]:

	Id	city	age	sex	social_class	primary_business	secondary_business	annual_income	monthly_
0	1	Dhanbad	22	F	Mochi	Tailoring	Others	36000.0	
1	2	Manjapra	21	F	OBC	Tailoring	none	94000.0	

In [14]:
```
sns.set_style('whitegrid')
```

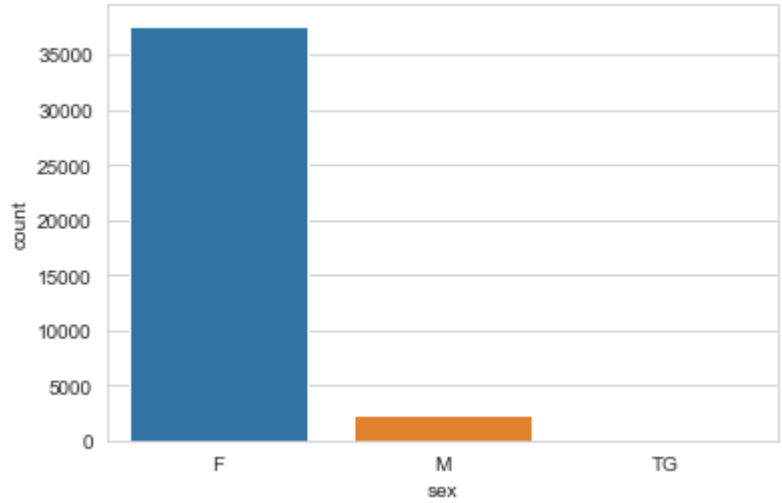
In [15]:
```
plt.figure(figsize=(16,5))
sns.countplot(df['age'])
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x297ab506b88>



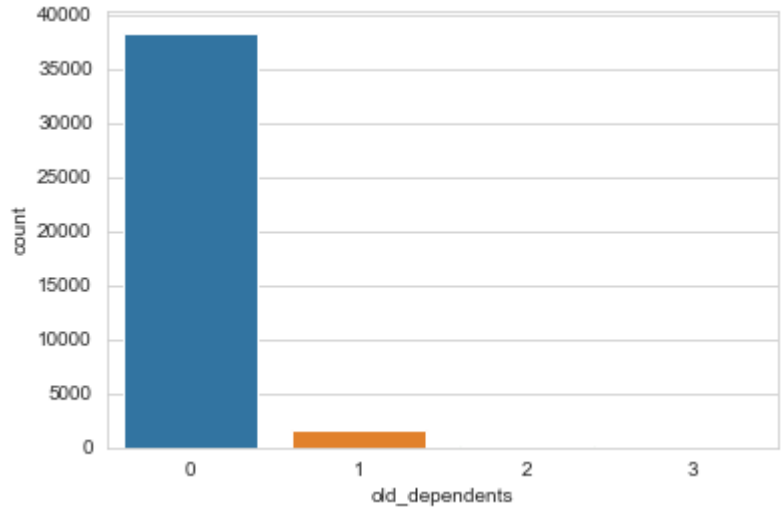
In [16]:
```
sns.countplot(df['sex'])
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x297ab638e88>



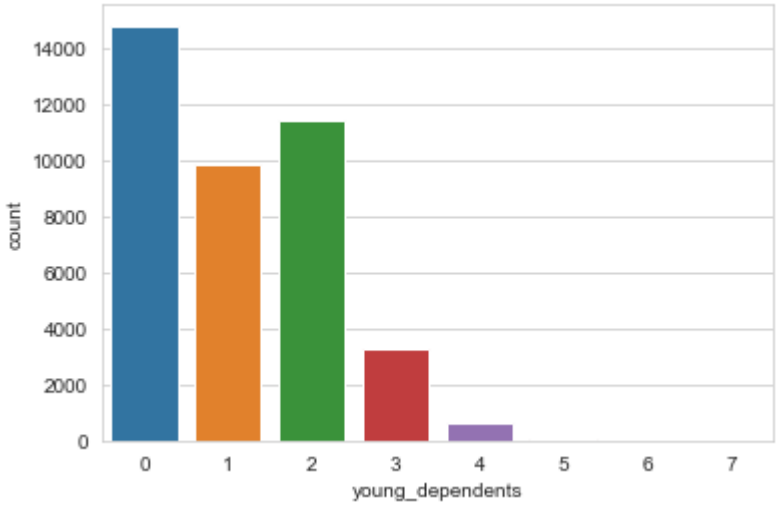
In [17]:
```
sns.countplot(df['old_dependents'])
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x297abe0d6c8>



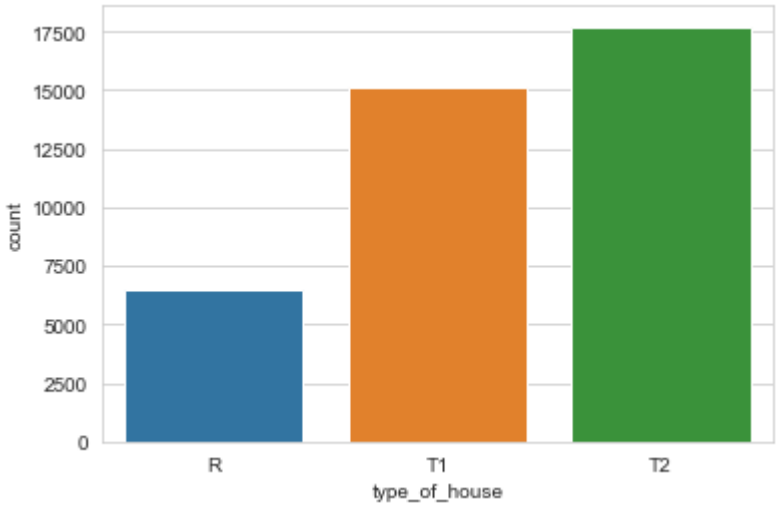
In [18]:
```
sns.countplot(df['young_dependents'])
```

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x297ab7a5f48>



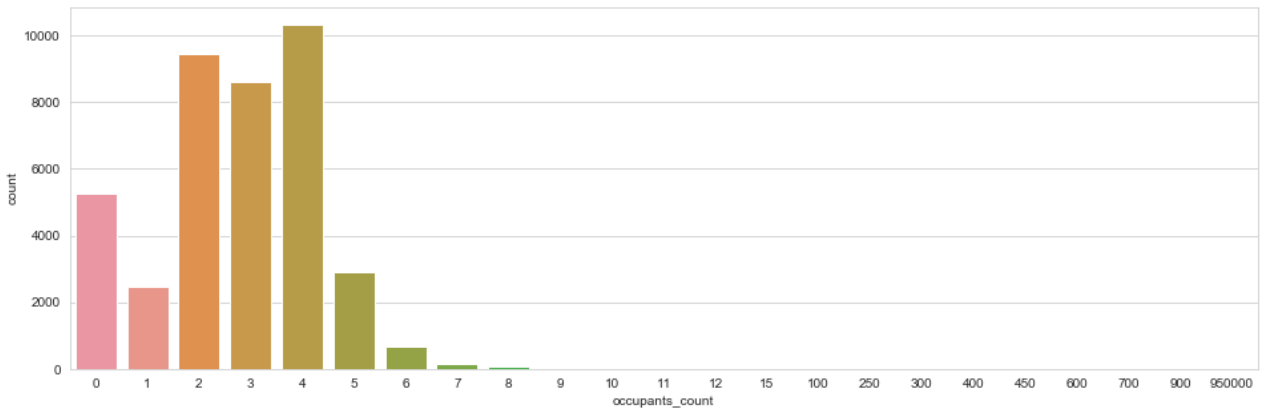
In [19]:
```
sns.countplot(df['type_of_house'])
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x297a9f31788>



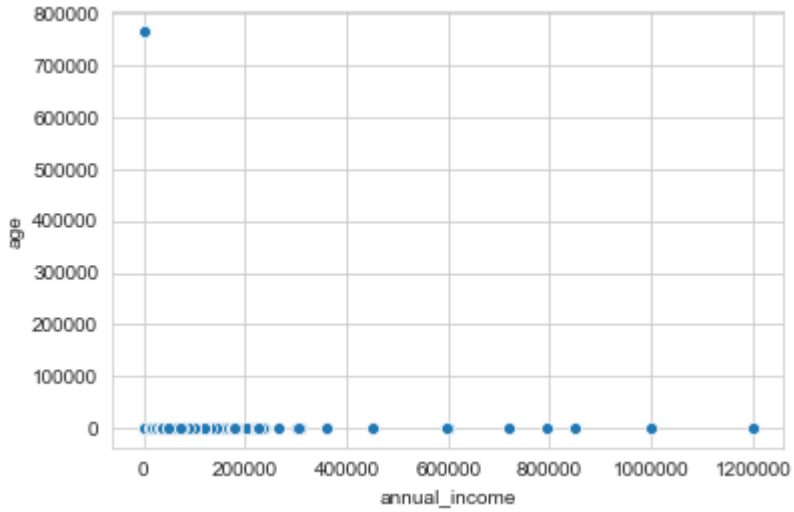
In [20]:
```
plt.figure(figsize=(16,5))
sns.countplot(df['occupants_count'])
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x297ab7b3788>



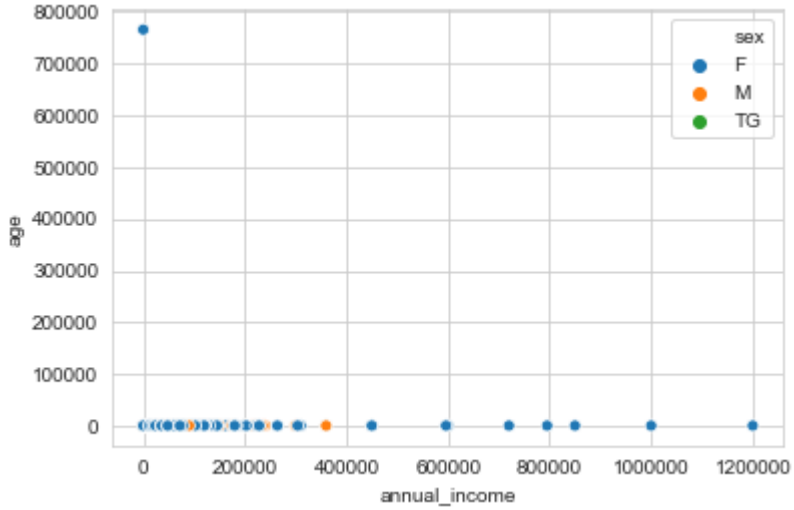
In [21]:
```
sns.scatterplot(x='annual_income',y='age',data=df)
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x297abf7ff48>



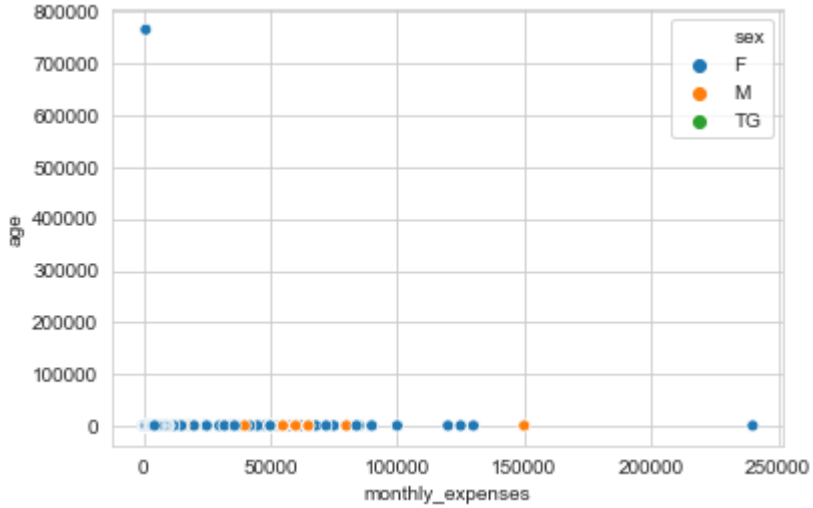
In [22]:
```
sns.scatterplot(x='annual_income',y='age',data=df,hue='sex')
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x297ab8104c8>



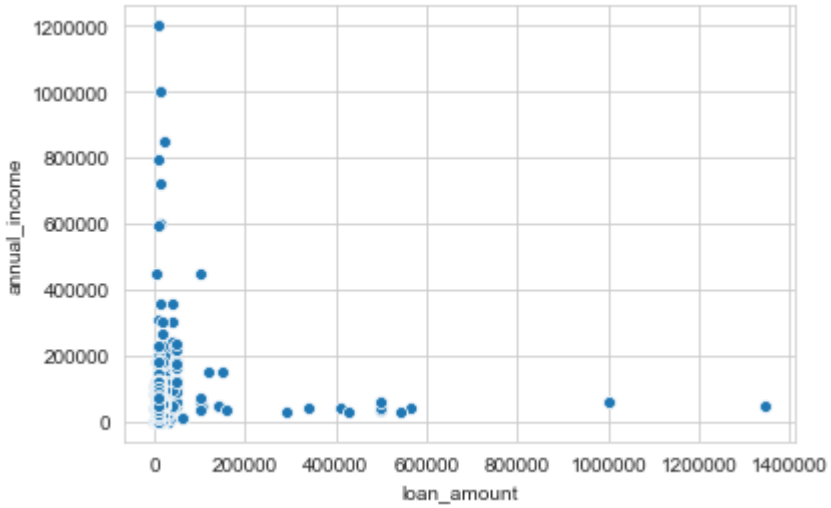
In [23]:
```
sns.scatterplot(x='monthly_expenses',y='age',data=df,hue='sex')
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x297ac0d9e08>



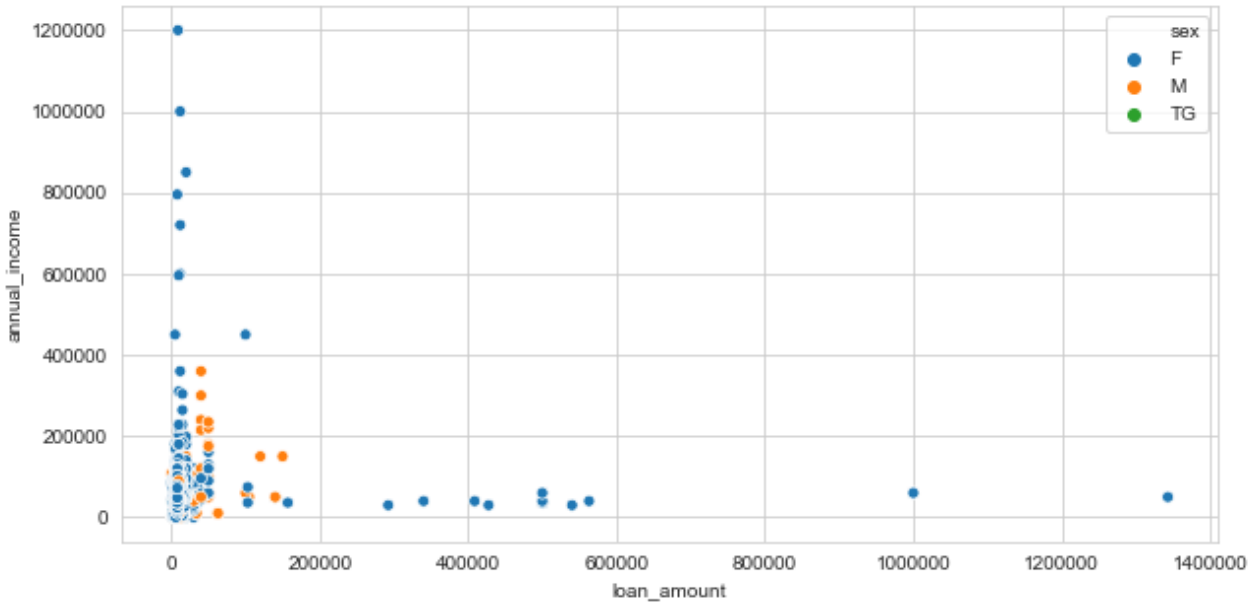
In [24]:
```
sns.scatterplot(x='loan_amount',y='annual_income',data=df)
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x297abdc87c8>



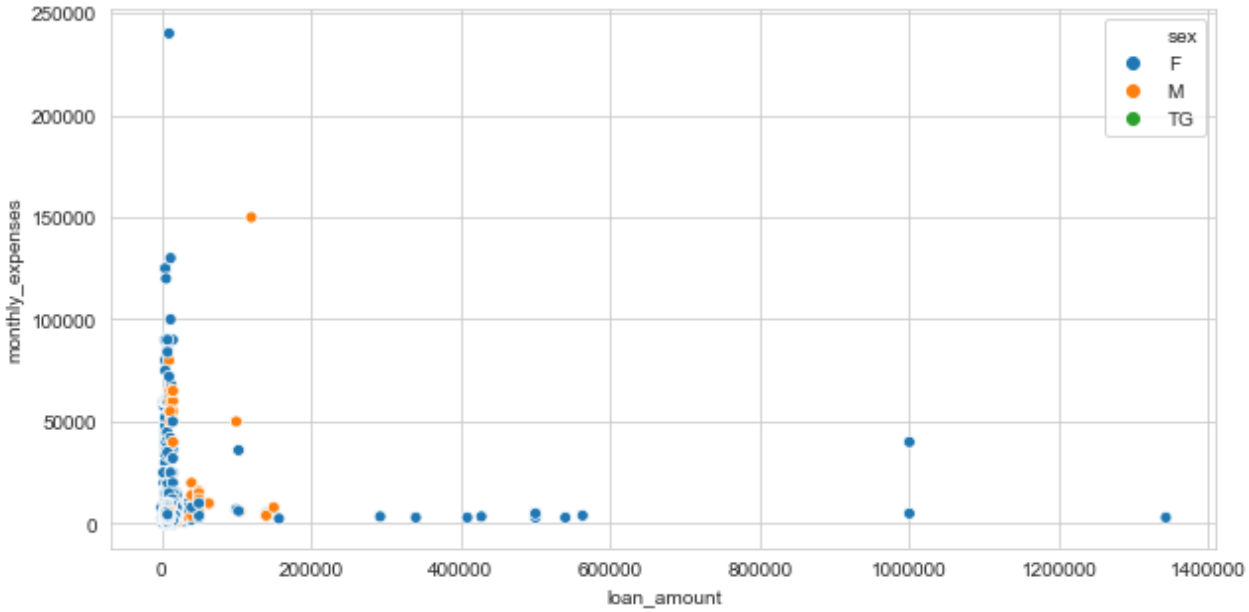
In [25]:
```
plt.figure(figsize=(10,5))
sns.scatterplot(x='loan_amount',y='annual_income',data=df,hue='sex')
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x297b4925d88>



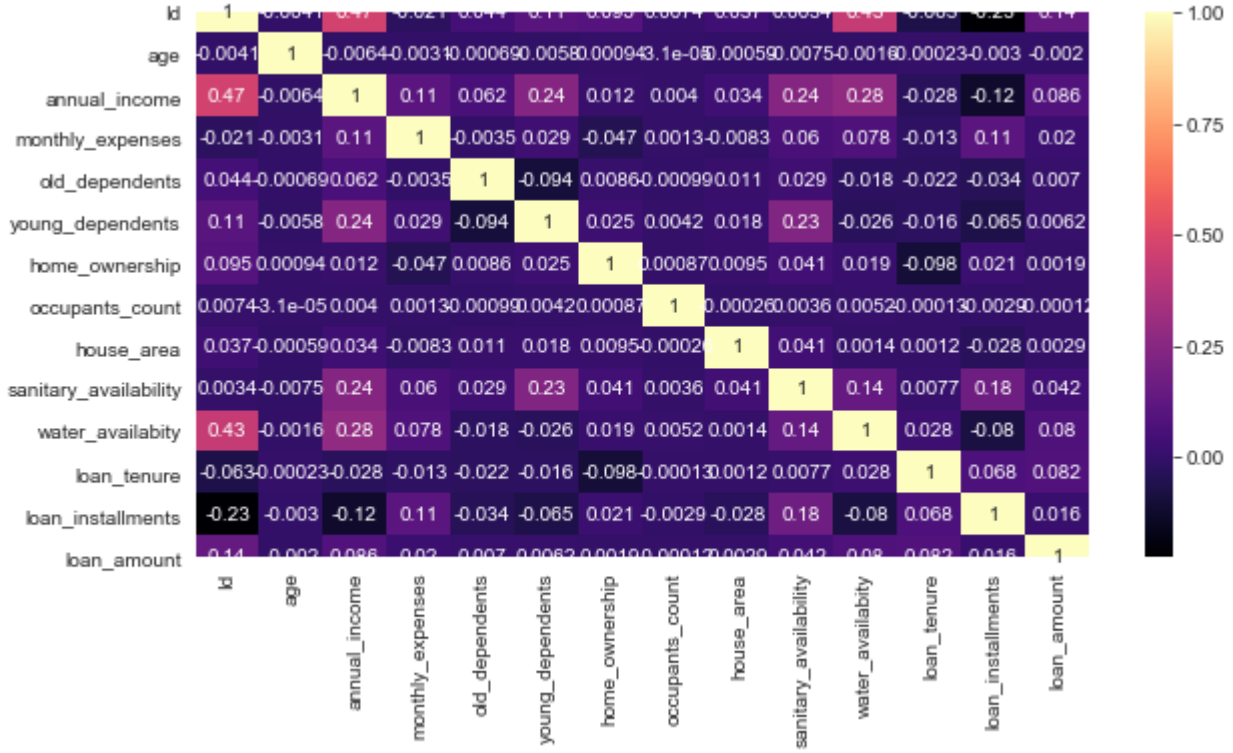
In [26]:
```
plt.figure(figsize=(10,5))
sns.scatterplot(x='loan_amount',y='monthly_expenses',data=df,hue='sex')
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x297ac2e4e88>



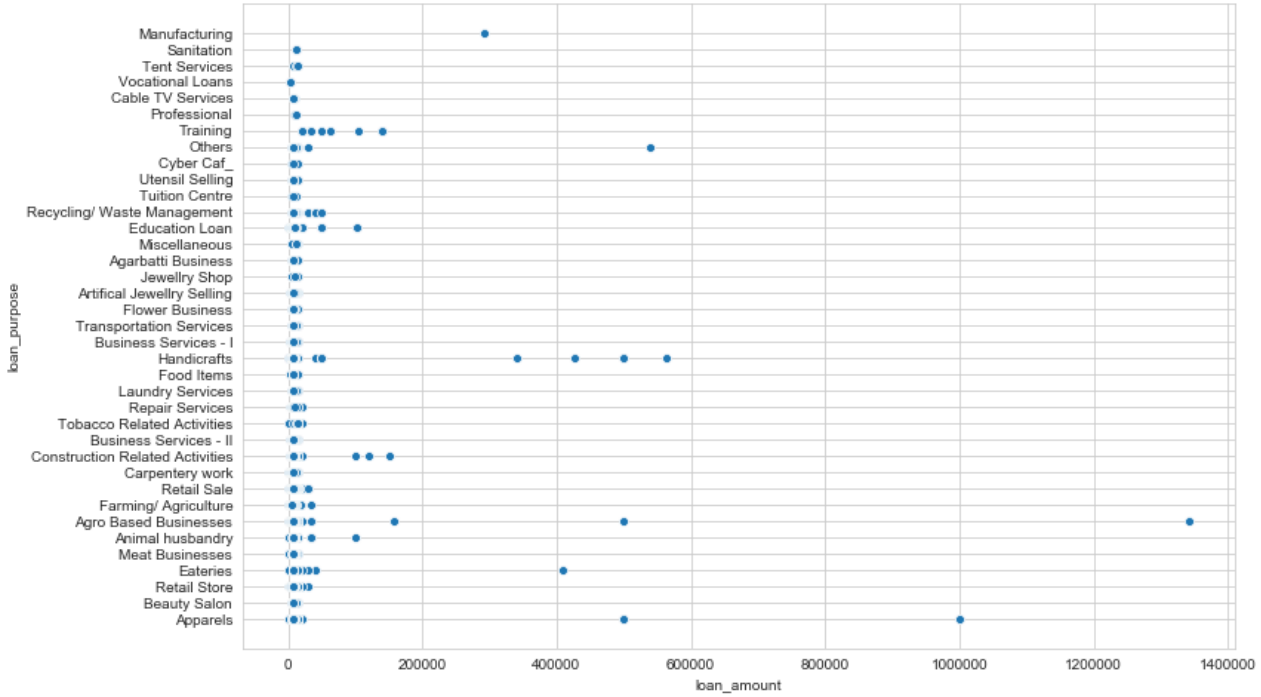
In [27]:
```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True,cmap='magma')
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x297ac2e6888>



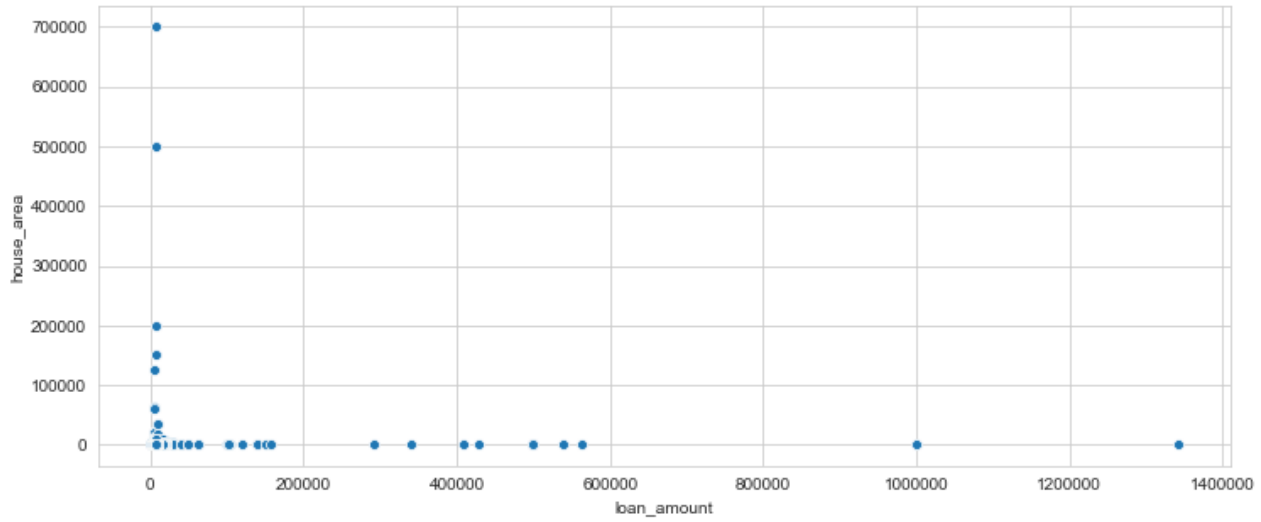
In [28]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x='loan_amount',y='loan_purpose',data=df)
```

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x297b5f095c8>



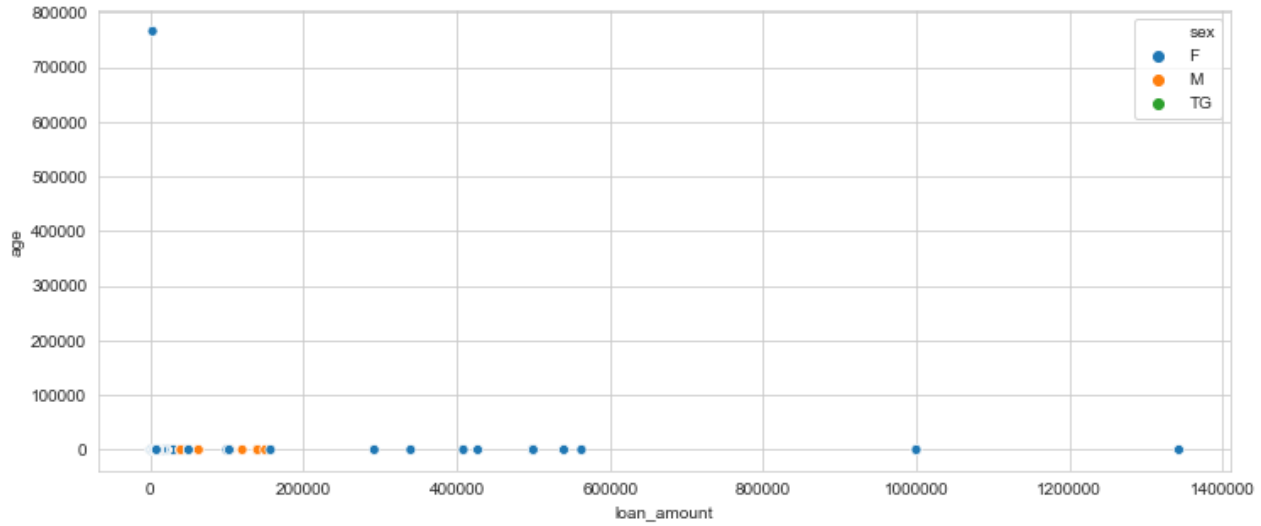
In [29]:
```
plt.figure(figsize=(12,5))
sns.scatterplot(x='loan_amount',y='house_area',data=df)
```

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x297b5f7d288>



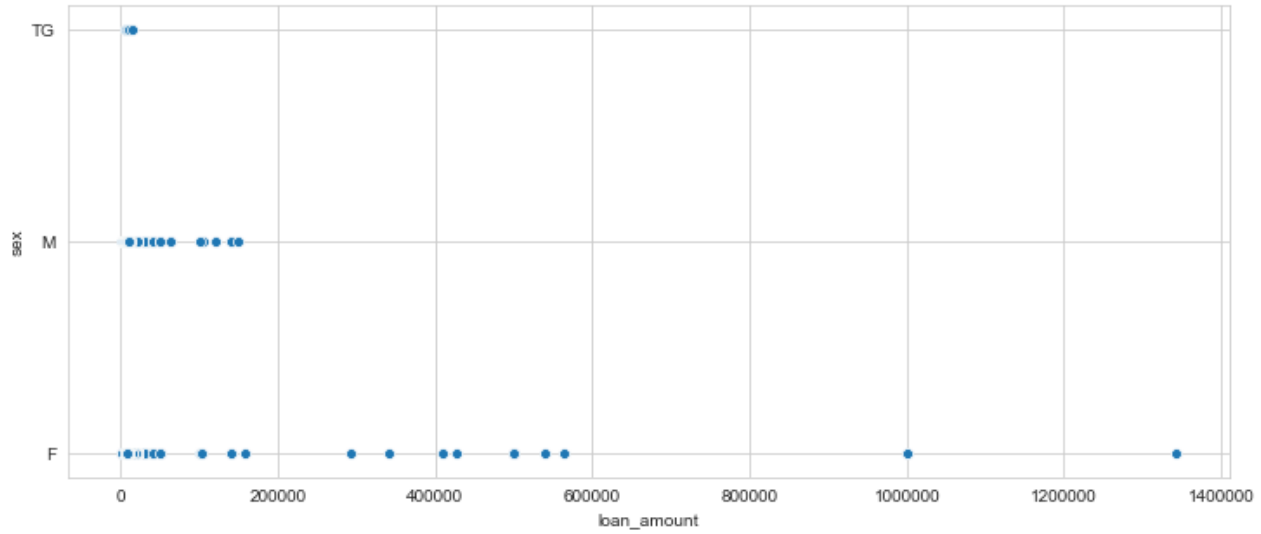
In [30]:
```
plt.figure(figsize=(12,5))
sns.scatterplot(x='loan_amount',y='age',data=df,hue='sex')
```

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x297b5ff75c8>



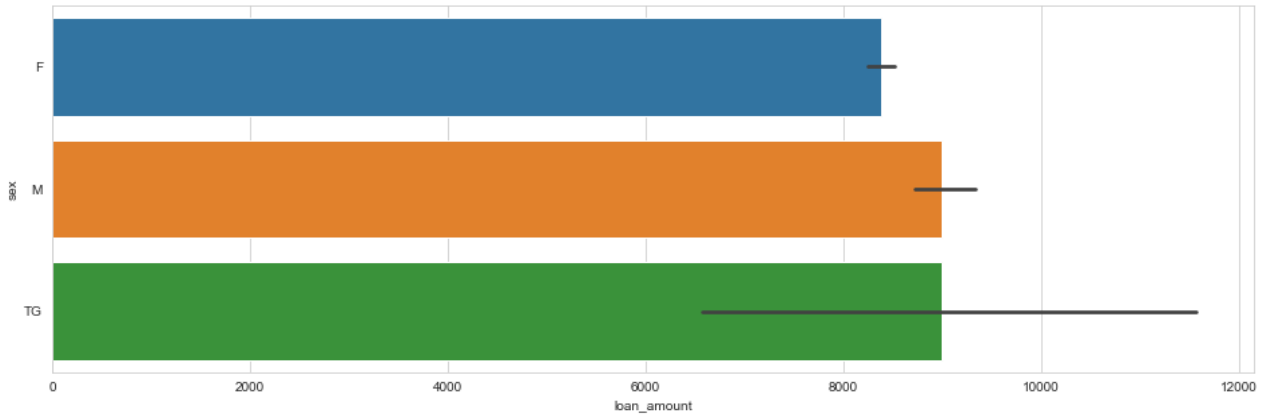
In [31]:
```
plt.figure(figsize=(12,5))
sns.scatterplot(x='loan_amount',y='sex',data=df)
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x297b66dad48>



In [32]:
```
plt.figure(figsize=(16,5))
sns.barplot(x='loan_amount',y='sex',data=df)
```

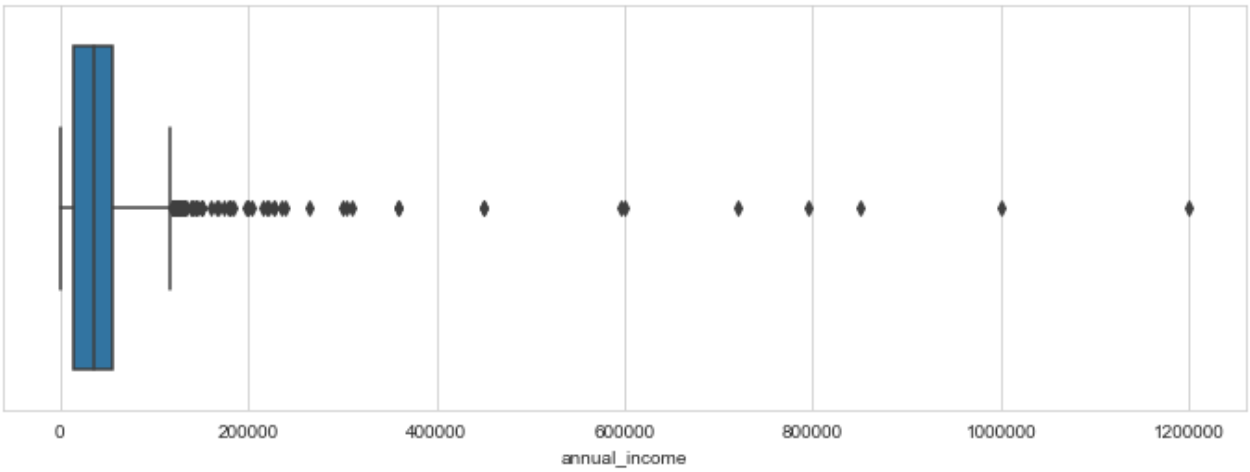
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x297b62c8a08>



Checking outliers for key features:

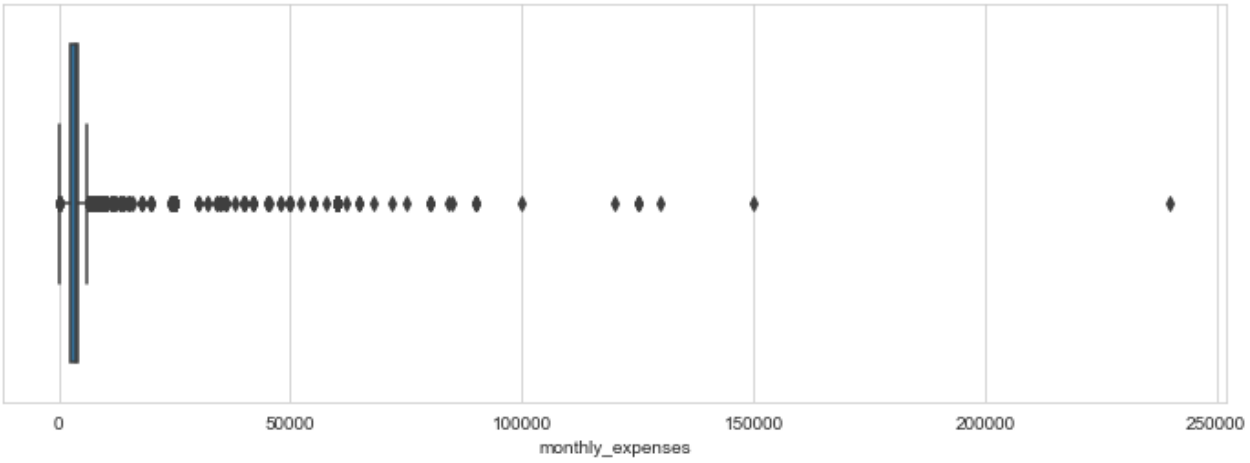
In [33]:
```
plt.figure(figsize=(12,4))
sns.boxplot(x=df['annual_income'])
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x297b6342048>



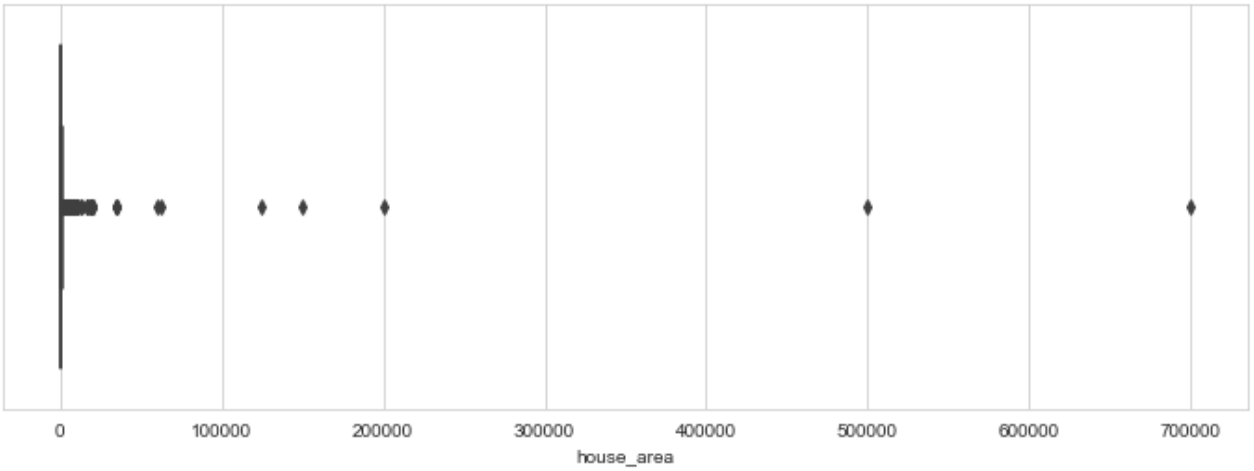
In [34]:
```
plt.figure(figsize=(12,4))
sns.boxplot(x=df['monthly_expenses'])
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x297b62dea88>



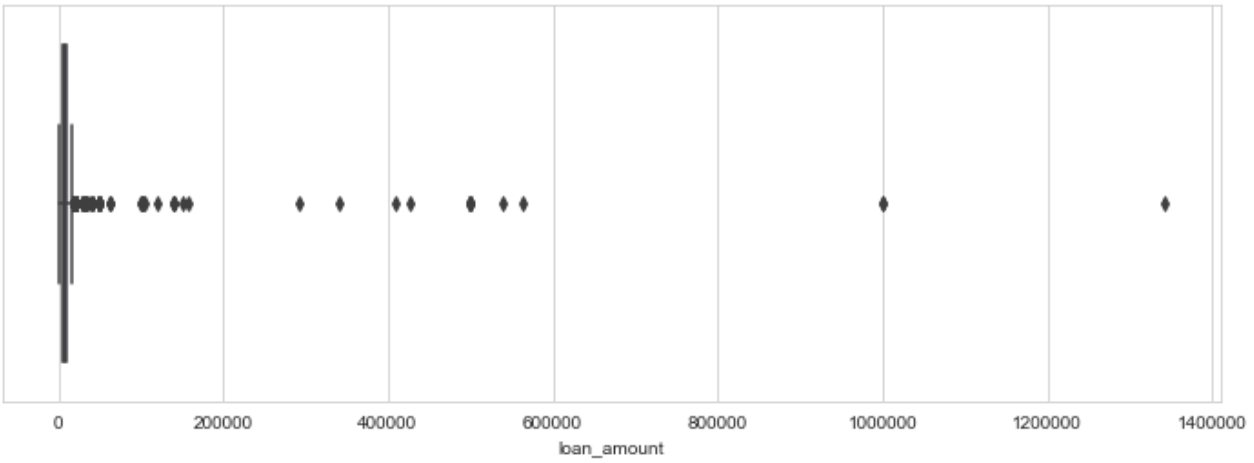
In [35]:
```
plt.figure(figsize=(12,4))
sns.boxplot(x=df['house_area'])
```

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x297b6bb3088>



In [36]:
```
plt.figure(figsize=(12,4))
sns.boxplot(x=df['loan_amount'])
```

Out[36]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x297b6342b08>
```



Checking value counts for all categorical data

In [37]:
```
df['social_class'].value_counts()
```

Out[37]:
```
OBC               10683
SC                 3136
ST                 2616
General            2299
Muslim             1743
                  ...
Madivlar shetty       1
Kumhaar               1
ONT                   1
Gowda shettru         1
Gen- BPL              1
Name: social_class, Length: 519, dtype: int64
```

In [38]:
```
df['city'].value_counts()
```

Out[38]:
```
Pusad         3154
Bahoriband    1979
PUSAD         1776
Shantipur     1727
Imphal        1699
              ...
```

```
nilgiri          1
Munidih          1
Vandazhy         1
Singjakmei       1
Raina            1
Name: city, Length: 856, dtype: int64
```

In [39]:
```
df['sex'].value_counts()
```

Out[39]:
```
F     37622
M      2371
TG        7
Name: sex, dtype: int64
```

In [40]:
```
df['primary_business'].value_counts()
```

Out[40]:
```
Tailoring                                              3971
Goat rearing                                           2268
Cow Rearing                                            2077
Handloom Work                                          2068
Vegetable cultivation                                  1704
                                                       ...
Stove Making                                              1
Kerosine                                                  1
Cycle Shop                                                1
Agricultural inputs to small and marginal farmers         1
Chumki stiching                                           1
Name: primary_business, Length: 441, dtype: int64
```

In [41]:
```
df['secondary_business'].value_counts()
```

Out[41]:
```
none                  27366
Others                 2564
Daily wage labourer    2545
Agriculture            2105
Livestock rearing       179
Name: secondary_business, dtype: int64
```

In [42]:
```
df['type_of_house'].value_counts()
```

Out[42]:
```
T2    17715
T1    15092
R      6499
Name: type_of_house, dtype: int64
```

In [43]:
```
df['loan_purpose'].value_counts()
```

Out[43]:
```
Apparels                        7064
Agro Based Businesses           4729
Animal husbandry                4421
Meat Businesses                 4302
Handicrafts                     4230
Farming/ Agriculture            3284
Education Loan                  2100
Retail Store                    1963
Eateries                        1831
Business Services - II           854
Tobacco Related Activities       853
Construction Related Activities  661
Retail Sale                      614
Artifical Jewellry Selling       556
Carpentery work                  299
Food Items                       285
Business Services - I            276
```

```
Transportation Services            245
Flower Business                    238
Beauty Salon                       204
Repair Services                    192
Laundry Services                   162
Agarbatti Business                 107
Utensil Selling                    104
Sanitation                         101
Recycling/ Waste Management        100
Others                              62
Vocational Loans                    40
Jewellry Shop                       30
Training                            23
Miscellaneous                       19
Cyber Caf_                           7
Tent Services                        6
Cable TV Services                    5
Tuition Centre                       3
Professional                         3
Manufacturing                        1
Name: loan_purpose, dtype: int64
```

In [ ]: