Background

A person’s creditworthiness is often associated (conversely) with the likelihood they may default on loans.

We’re giving you anonymized data on about 1000 loan applications, along with a certain set of attributes about the applicant itself, and whether they were considered high risk.

0 = Low credit risk i.e high chance of paying back the loan amount #non defaulters

1 = High credit risk i.e low chance of paying back the loan amount #defaulter

Dataset Description

```
The dataset has two files:

1. `applicant.csv`: This file contains personal data about
the (primary) applicant
- Unique ID: `applicant_id` (string)
- Other fields:
    - Primary_applicant_age_in_years (numeric)
    - Gender (string)
    - Marital_status (string)
    - Number_of_dependents (numeric)
    - Housing (string)
    - Years_at_current_residence (numeric)
    - Employment_status (string)
    - Has_been_employed_for_at_least (string)
    - Has_been_employed_for_at_most (string)
    - Telephone (string)
    - Foreign_worker (numeric)
    - Savings_account_balance (string)
    -
Balance_in_existing_bank_account_(lower_limit_of_bucket)
(string)
    -
Balance_in_existing_bank_account_(upper_limit_of_bucket)
(string)

1. `loan.csv`: This file contains data more specific to the
loan application
- Target: `high_risk_application` (numeric)
- Other fields:
    - applicant_id (string)
    - Months_loan_taken_for (numeric)
    - Purpose (string)
    - Principal_loan_amount (numeric)
    - EMI_rate_in_percentage_of_disposable_income (numeric)
    - Property (string)
    - Has_coapplicant (numeric)
    - Has_guarantor (numeric)
    - Other_EMI_plans (string)
    - Number_of_existing_loans_at_this_bank (numeric)
    - Loan_history (string)
```

In [1]:

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
appdata = pd.read_csv('applicant.csv')
```

In [3]:
```
appdata.head()
```

Out[3]:

	applicant_id	Primary_applicant_age_in_years	Gender	Marital_status	Numbe
0	1469590	67	male	single	
1	1203873	22	female	divorced/separated/married	
2	1432761	49	male	single	
3	1207582	45	male	single	
4	1674436	53	male	single	

In [4]:
```
loan = pd.read_csv('loan.csv')
```

In [5]:
```
loan.head()
```

Out[5]:

	loan_application_id	applicant_id	Months_loan_taken_for	Purpose	Principal_loan_amo
0	d68d975e-edad-11ea-8761-1d6f9c1ff461	1469590	6	electronic equipment	1169(
1	d68d989e-edad-11ea-b1d5-2bcf65006448	1203873	48	electronic equipment	5951(
2	d68d995c-edad-11ea-814a-1b6716782575	1432761	12	education	2096(
3	d68d99fc-edad-11ea-8841-17e8848060ae	1207582	42	FF&E	7882(
4	d68d9a92-edad-11ea-9f3d-1f8682db006a	1674436	24	new vehicle	4870(

In [6]:
```
loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   loan_application_id                     1000 non-null   object
 1   applicant_id                            1000 non-null   int64
 2   Months_loan_taken_for                   1000 non-null   int64
 3   Purpose                                 988 non-null    object
 4   Principal_loan_amount                   1000 non-null   int64
 5   EMI_rate_in_percentage_of_disposable_income  1000 non-null   int64
 6   Property                                846 non-null    object
 7   Has_coapplicant                         1000 non-null   int64
 8   Has_guarantor                           1000 non-null   int64
 9   Other_EMI_plans                         186 non-null    object
 10  Number_of_existing_loans_at_this_bank   1000 non-null   int64
 11  Loan_history                            1000 non-null   object
 12  high_risk_applicant                     1000 non-null   int64
dtypes: int64(8), object(5)
memory usage: 101.7+ KB
```

In [7]:

```
appdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 15 columns):
 #   Column                                              Non-Null Cou
nt  Dtype
---  ------                                              ------------
--  -----
 0   applicant_id                                        1000 non-nul
l   int64
 1   Primary_applicant_age_in_years                      1000 non-nul
l   int64
 2   Gender                                              1000 non-nul
l   object
 3   Marital_status                                      1000 non-nul
l   object
 4   Number_of_dependents                                1000 non-nul
l   int64
 5   Housing                                             1000 non-nul
l   object
 6   Years_at_current_residence                          1000 non-nul
l   int64
 7   Employment_status                                   1000 non-nul
l   object
 8   Has_been_employed_for_at_least                      938 non-null
object
 9   Has_been_employed_for_at_most                       747 non-null
object
 10  Telephone                                           404 non-null
object
 11  Foreign_worker                                      1000 non-nul
l   int64
 12  Savings_account_balance                             817 non-null
object
 13  Balance_in_existing_bank_account_(lower_limit_of_bucket)  332 non-null
object
 14  Balance_in_existing_bank_account_(upper_limit_of_bucket)  543 non-null
object
dtypes: int64(5), object(10)
memory usage: 117.3+ KB
```

In [8]:

```
loan.Other_EMI_plans.unique()
```

Out[8]:

```
array([nan, 'bank', 'stores'], dtype=object)
```

In [9]:

```
loan.Purpose.unique()
```

Out[9]:

```
array(['electronic equipment', 'education', 'FF&E', 'new vehicle',
       'used vehicle', 'business', 'domestic appliances', 'repair costs',
       nan, 'career development'], dtype=object)
```

In [10]:

```
loan.Property.unique()
```

Out[10]:

```
array(['real estate', 'building society savings agreement/life insurance',
       nan, 'car or other'], dtype=object)
```

TASK-1

1. Do the Exploratory Data Analysis & share the insights.

2. How would you segment customers based on their risk (of default).

- We Can Segment Them As Follows :
- 0 - Non-Defaulters : high chance of paying back the loan amount.
- 1 - Defaulters : low chance of paying back the loan amount.

1. Which of these segments / sub-segments would you propose be approved?
   - For e.g. Would a person with critical credit history be more creditworthy? Are young people more creditworthy? Would a person with more credit accounts be more creditworthy?
2. Tell us what your observations were on the data itself (completeness, skews).

In [11]:

```
data = pd.merge(appdata,loan)
```

In [12]:

```
data.head()
```

Out[12]:

	applicant_id	Primary_applicant_age_in_years	Gender	Marital_status	Numbe
0	1469590	67	male	single	
1	1203873	22	female	divorced/separated/married	
2	1432761	49	male	single	
3	1207582	45	male	single	
4	1674436	53	male	single	

5 rows × 27 columns

In [13]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 27 columns):
```

```
 #   Column                                                     Non-Null Cou
nt  Dtype
---  ------                                                     ------------
--  -----
 0   applicant_id                                               1000 non-nul
l   int64
 1   Primary_applicant_age_in_years                             1000 non-nul
l   int64
 2   Gender                                                     1000 non-nul
l   object
 3   Marital_status                                             1000 non-nul
l   object
 4   Number_of_dependents                                       1000 non-nul
l   int64
 5   Housing                                                    1000 non-nul
l   object
 6   Years_at_current_residence                                 1000 non-nul
l   int64
 7   Employment_status                                          1000 non-nul
l   object
 8   Has_been_employed_for_at_least                             938 non-null
object
 9   Has_been_employed_for_at_most                              747 non-null
object
 10  Telephone                                                  404 non-null
object
 11  Foreign_worker                                             1000 non-nul
l   int64
 12  Savings_account_balance                                    817 non-null
object
 13  Balance_in_existing_bank_account_(lower_limit_of_bucket)  332 non-null
object
 14  Balance_in_existing_bank_account_(upper_limit_of_bucket)  543 non-null
object
 15  loan_application_id                                        1000 non-nul
l   object
 16  Months_loan_taken_for                                      1000 non-nul
l   int64
 17  Purpose                                                    988 non-null
object
 18  Principal_loan_amount                                      1000 non-nul
l   int64
 19  EMI_rate_in_percentage_of_disposable_income                1000 non-nul
l   int64
 20  Property                                                   846 non-null
object
 21  Has_coapplicant                                            1000 non-nul
l   int64
 22  Has_guarantor                                              1000 non-nul
l   int64
 23  Other_EMI_plans                                            186 non-null
object
 24  Number_of_existing_loans_at_this_bank                      1000 non-nul
l   int64
 25  Loan_history                                               1000 non-nul
l   object
 26  high_risk_applicant                                        1000 non-nul
```

```
l    int64
dtypes: int64(12), object(15)
memory usage: 218.8+ KB
```

In [14]:
```
data.duplicated().sum().any()
```

Out[14]:
```
False
```

In [15]:
```
data.isnull().sum().any()
```

Out[15]:
```
True
```

In [16]:
```
data.isnull().sum()
```

Out[16]:
```
applicant_id                                              0
Primary_applicant_age_in_years                            0
Gender                                                    0
Marital_status                                            0
Number_of_dependents                                      0
Housing                                                   0
Years_at_current_residence                                0
Employment_status                                         0
Has_been_employed_for_at_least                           62
Has_been_employed_for_at_most                           253
Telephone                                               596
Foreign_worker                                            0
Savings_account_balance                                 183
Balance_in_existing_bank_account_(lower_limit_of_bucket)    668
Balance_in_existing_bank_account_(upper_limit_of_bucket)    457
loan_application_id                                       0
Months_loan_taken_for                                     0
Purpose                                                  12
Principal_loan_amount                                     0
EMI_rate_in_percentage_of_disposable_income               0
Property                                                154
Has_coapplicant                                           0
Has_guarantor                                             0
Other_EMI_plans                                         814
Number_of_existing_loans_at_this_bank                     0
Loan_history                                              0
high_risk_applicant                                       0
dtype: int64
```

Dropping The Columns Having Missing Values, As These Columns Doesn't Make Much Difference Even If We Remove Them.

In [17]:
```
data.drop(['Has_been_employed_for_at_least'], axis=1, inplace=True)
```

In [18]:
```
data.drop(['Has_been_employed_for_at_most'], axis=1, inplace=True)
```

In [19]:
```
data.drop(['Telephone'], axis=1, inplace=True)
```

In [20]:
```
data.drop(['Savings_account_balance','Balance_in_existing_bank_account_(low
```

In [21]:
```
data.drop(['Balance_in_existing_bank_account_(upper_limit_of_bucket)','Purp
```

In [22]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column                                   Non-Null Count  Dtype 
---  ------                                   --------------  ----- 
 0   applicant_id                             1000 non-null   int64 
 1   Primary_applicant_age_in_years           1000 non-null   int64 
 2   Gender                                   1000 non-null   object
 3   Marital_status                           1000 non-null   object
 4   Number_of_dependents                     1000 non-null   int64 
 5   Housing                                  1000 non-null   object
 6   Years_at_current_residence               1000 non-null   int64 
 7   Employment_status                        1000 non-null   object
 8   Foreign_worker                           1000 non-null   int64 
 9   loan_application_id                      1000 non-null   object
 10  Months_loan_taken_for                    1000 non-null   int64 
 11  Principal_loan_amount                    1000 non-null   int64 
 12  EMI_rate_in_percentage_of_disposable_income  1000 non-null   int64 
 13  Has_coapplicant                          1000 non-null   int64 
 14  Has_guarantor                            1000 non-null   int64 
 15  Number_of_existing_loans_at_this_bank    1000 non-null   int64 
 16  Loan_history                             1000 non-null   object
 17  high_risk_applicant                      1000 non-null   int64 
dtypes: int64(12), object(6)
memory usage: 148.4+ KB
```

In [23]:
```
data.head()
```

Out[23]:

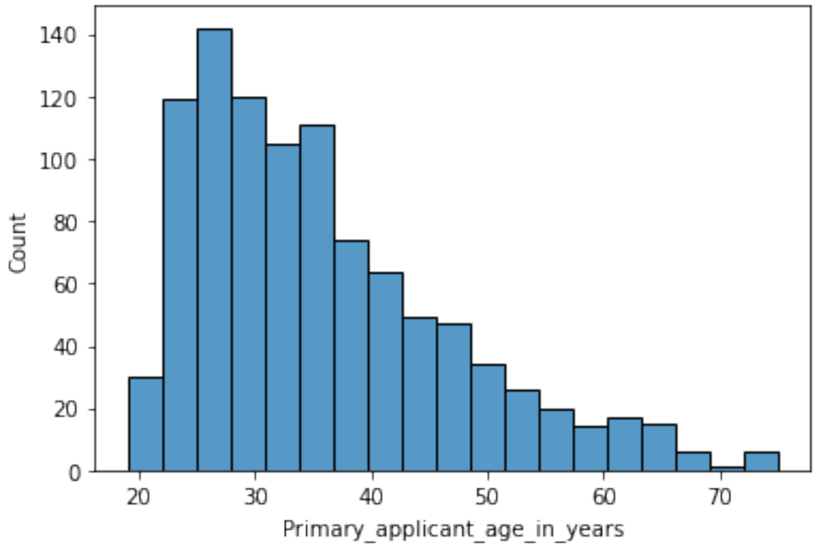
	applicant_id	Primary_applicant_age_in_years	Gender	Marital_status	Numbe
0	1469590	67	male	single	
1	1203873	22	female	divorced/separated/married	
2	1432761	49	male	single	
3	1207582	45	male	single	
4	1674436	53	male	single	

In [24]:

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [25]:

```
sns.histplot(data['Primary_applicant_age_in_years']);
```



In [26]:

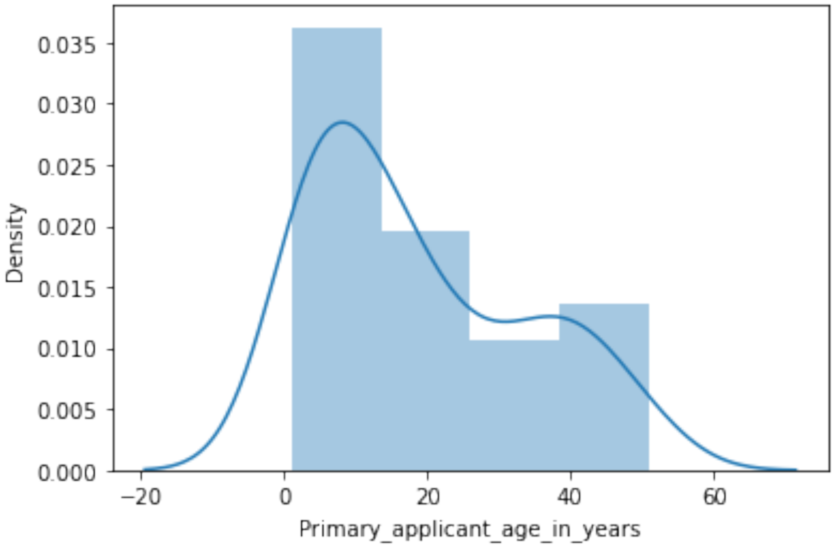
```
data['Primary_applicant_age_in_years'].value_counts().sort_index(ascending=
```

Out[26]:
```
19     2
20    14
21    14
22    27
23    48
24    44
25    41
26    50
27    51
28    43
29    37
30    40
31    38
32    34
33    33
34    32
35    40
36    39
37    29
38    24
39    21
40    25
41    17
42    22
43    17
44    17
45    15
46    18
47    17
48    12
49    14
50    12
51     8
52     9
53     7
54    10
55     8
56     3
57     9
58     5
59     3
60     6
61     7
62     2
63     8
64     5
65     5
66     5
67     3
68     3
70     1
74     4
75     2
Name: Primary_applicant_age_in_years, dtype: int64
```

- Applicants having 23 - 30 age are more in quantity as compared to others

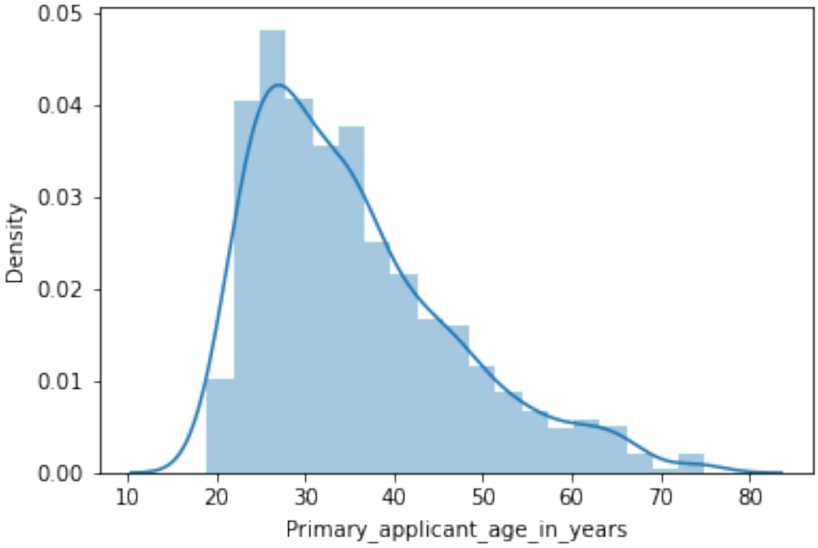
In [27]:

```
sns.distplot(data.Primary_applicant_age_in_years.value_counts());
```



In [28]:

```
sns.distplot(data['Primary_applicant_age_in_years']);
```



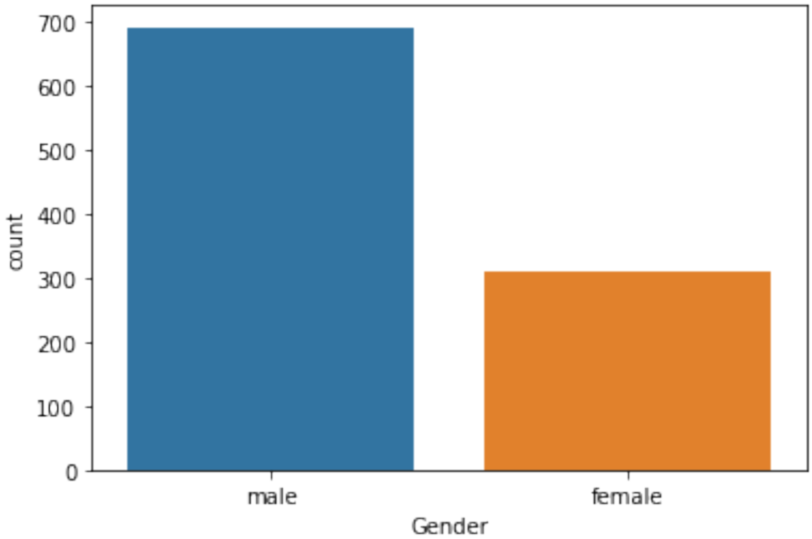
In [29]:

```
data['high_risk_applicant'].groupby(data['Primary_applicant_age_in_years']
```

Out[29]:

```
Primary_applicant_age_in_years  high_risk_applicant
19                              0                      1
                                1                      1
20                              0                      9
                                1                      5
21                              0                      9
                                                      ..
68                              0                      1
70                              0                      1
74                              0                      3
                                1                      1
75                              0                      2
Name: high_risk_applicant, Length: 100, dtype: int64
```

In [30]:
```
sns.countplot(data['Gender']);
```



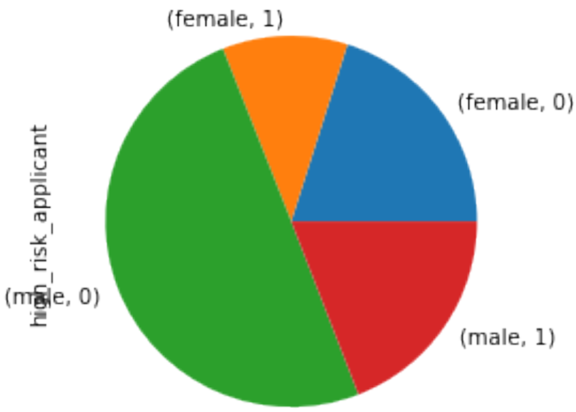
In [31]:
```
data.Gender.value_counts()
```

Out[31]:
```
male      690
female    310
Name: Gender, dtype: int64
```

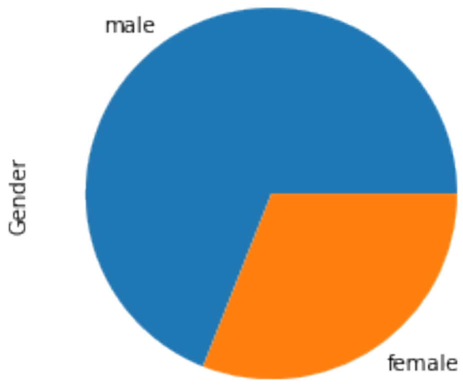
In [32]:
```
data['high_risk_applicant'].groupby(data['Gender']).value_counts()
```

Out[32]:
```
Gender  high_risk_applicant
female  0                      201
        1                      109
male    0                      499
        1                      191
Name: high_risk_applicant, dtype: int64
```

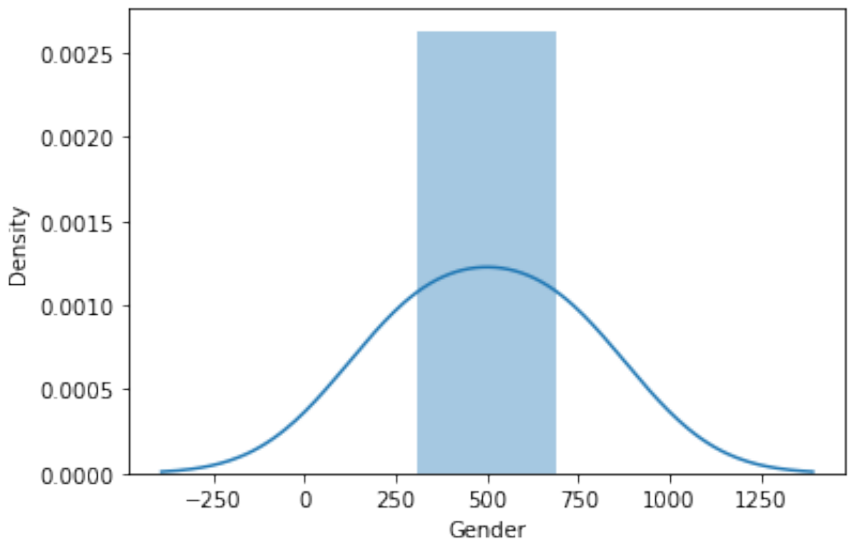
In [33]:
```
data['high_risk_applicant'].groupby(data['Gender']).value_counts().plot(ki
```



In [34]:
```
data.Gender.value_counts().plot(kind='pie');
```



In [35]:
```
sns.distplot(data.Gender.value_counts());
```



- Total Female Applicants - 310 Out Of Which 209 are in non-defaulter zone: low risk (high chance of paying back the loan), 109 are in defaulter zone: high risk(low chance of paying back the loan)
- Total male Applicants - 690 Out Of Which 499 are non-defaulter zone:low risk(high chance of paying back the loan), 191 are in defaulter zone: high risk(low chance of paying back the loan)

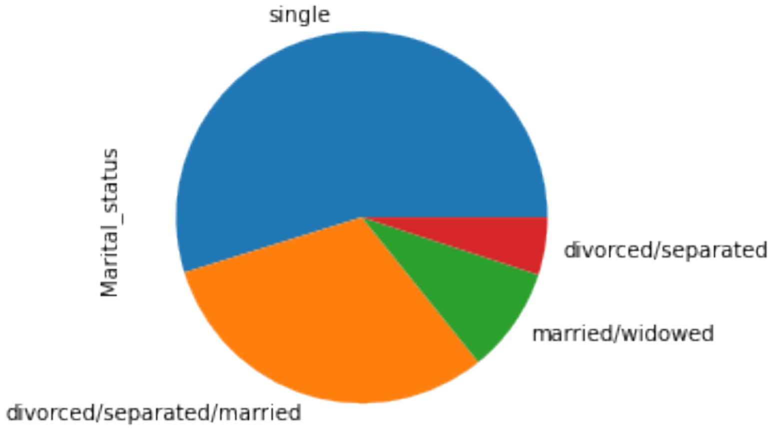
In [36]:
```
data['Marital_status'].unique()
```

Out[36]:
```
array(['single', 'divorced/separated/married', 'divorced/separated',
       'married/widowed'], dtype=object)
```

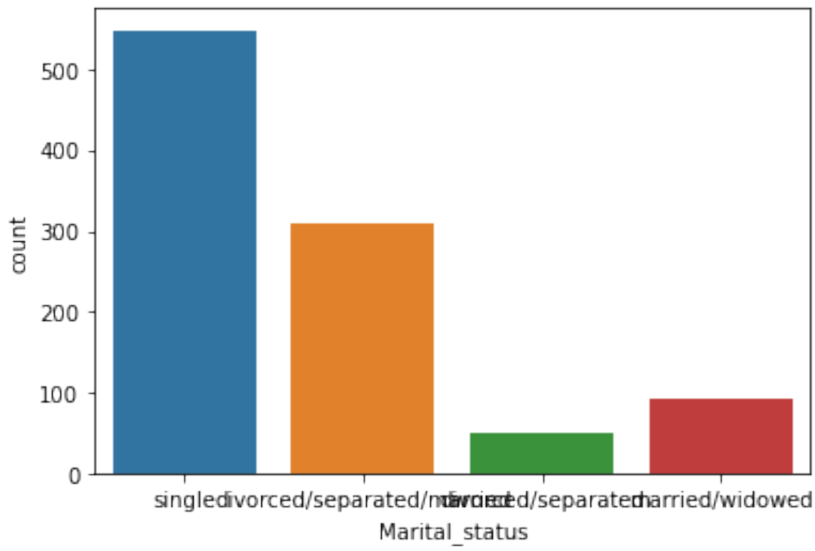
In [37]:
```
data.Marital_status.value_counts()
```

Out[37]:
```
single                            548
divorced/separated/married        310
married/widowed                    92
divorced/separated                 50
Name: Marital_status, dtype: int64
```

In [38]:
```
data.Marital_status.value_counts().plot(kind='pie');
```



In [39]:
```
sns.countplot(data['Marital_status']);
```



In [40]:
```
data['high_risk_applicant'].groupby(data['Marital_status']).value_counts()
```

Out[40]:
```
Marital_status             high_risk_applicant
divorced/separated         0                         30
                           1                         20
divorced/separated/married 0                        201
                           1                        109
married/widowed            0                         67
                           1                         25
single                     0                        402
                           1                        146
Name: high_risk_applicant, dtype: int64
```

- single : Total = 548, Defaulter Zone = 146, Non-Defaulter Zone = 402
- divorced/separated/married : Total = 310, Defaulter Zone = 109, Non-Defaulter Zone = 201
- married/widowed : Total = 92, Defaulter Zone = 25, Non-Defaulter Zone = 67
- divorced/separated : Total = 50, Defaulter Zone = 20, Non-Defaulter Zone = 30

In [41]:
```
548+310+92+50 #Total Applicants
```

Out[41]:
```
1000
```

In [42]:
```
146+109+25+20 #Defaulter Zone
```

Out[42]:
```
300
```

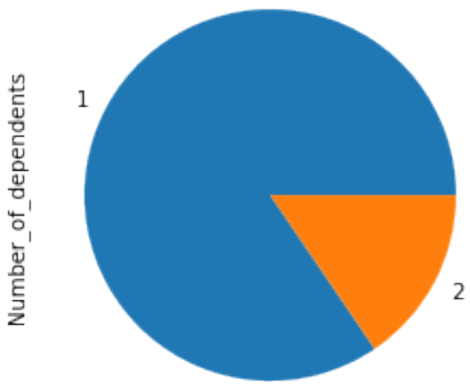
In [43]:
```
data['Number_of_dependents'].unique()
```

Out[43]:
```
array([1, 2])
```

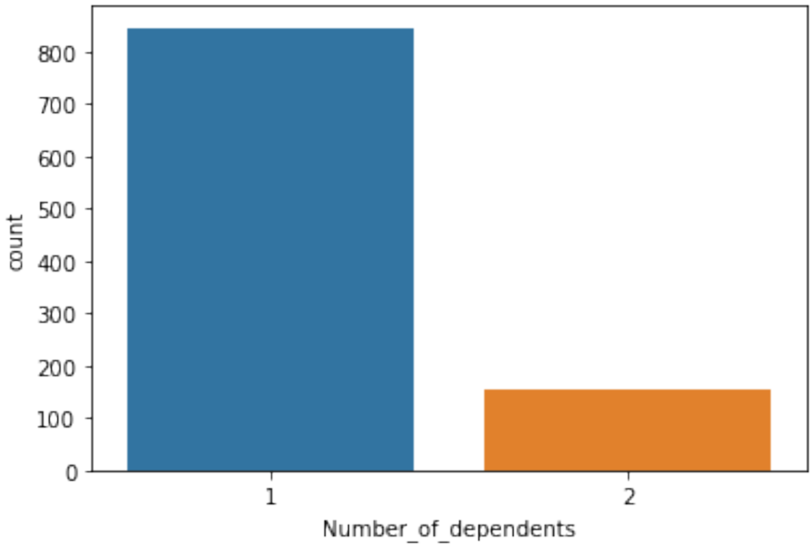
In [44]:
```
data.Number_of_dependents.value_counts()
```

Out[44]:
```
1    845
2    155
Name: Number_of_dependents, dtype: int64
```

In [45]:
```
data.Number_of_dependents.value_counts().plot(kind='pie');
```



In [46]:
```
sns.countplot(data['Number_of_dependents']);
```



In [47]:
```
data['high_risk_applicant'].groupby(data['Number_of_dependents']).value_cou
```

Out[47]:
```
Number_of_dependents  high_risk_applicant
1                     0                      591
                      1                      254
2                     0                      109
                      1                       46
Name: high_risk_applicant, dtype: int64
```

- People Having No. Of Dependents As 1 are 845 in Total Outoff Which 254 are in defaulter zone and 591 are in non-defaulter zone.
- People Having No. Of Dependents As 2 are 155 in Total Outoff Which 46 are in defaulter zone and 109 are in non-defaulter zone

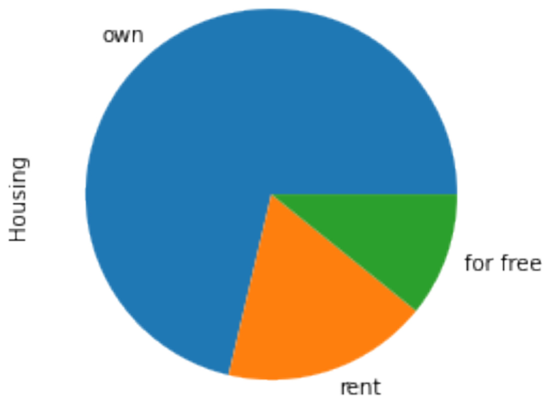
In [48]:
```
data['Housing'].unique()
```

Out[48]:
```
array(['own', 'for free', 'rent'], dtype=object)
```

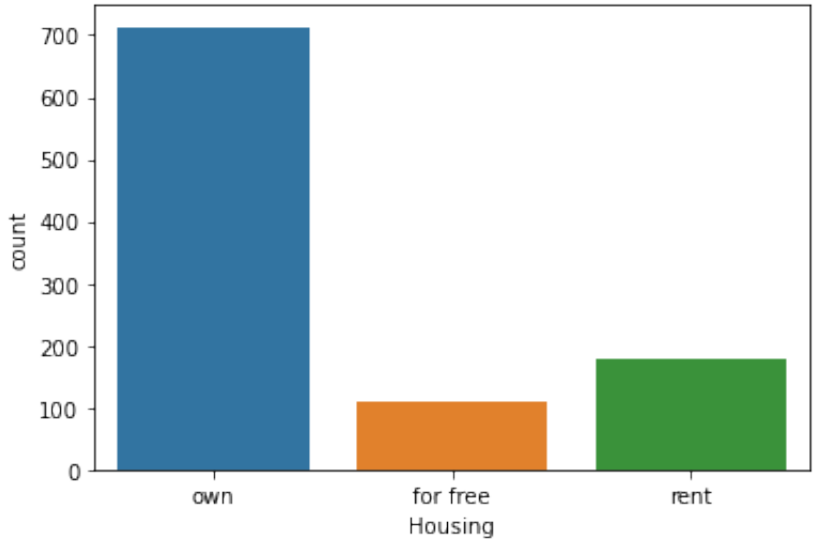
In [49]:
```
data.Housing.value_counts()
```

Out[49]:
```
own         713
rent        179
for free    108
Name: Housing, dtype: int64
```

In [50]:
```
data.Housing.value_counts().plot(kind='pie');
```



In [51]:
```
sns.countplot(data['Housing']);
```



In [52]:
```
data['high_risk_applicant'].groupby(data['Housing']).value_counts()
```

Out[52]:
```
Housing   high_risk_applicant
for free  0                       64
          1                       44
own       0                      527
          1                      186
rent      0                      109
          1                       70
Name: high_risk_applicant, dtype: int64
```

- Applicants Those Who Live In There "own" House are 713 in Total Out-off Which 186 are In Defaulter Zone & 527 are In Non-Defaulter Zone.
- Applicants Those Who Live Giving "rent" For House are 179 in Total Out-off Which 70 are In Defaulter Zone & 109 are In Non-Defaulter Zone.
- Applicants Those Who Live "for free" In House are 108 in Total Out-off Which 44 are In Defaulter Zone & 64 are In Non-Defaulter Zone.

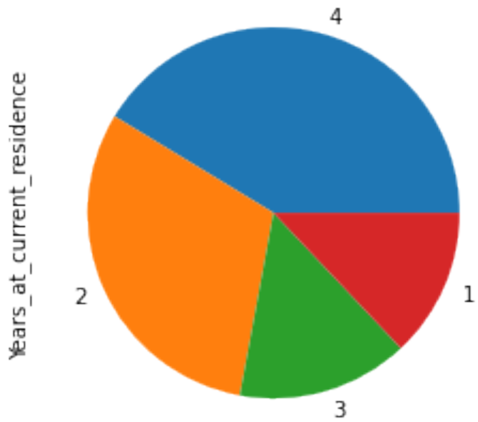
In [53]:
```
data['Years_at_current_residence'].unique()
```

Out[53]:
```
array([4, 2, 3, 1])
```

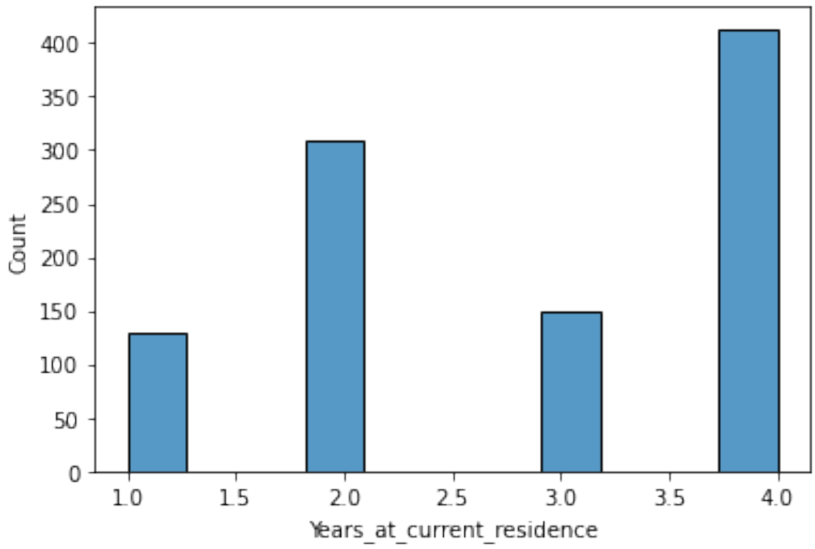
In [54]:
```
data.Years_at_current_residence.value_counts()
```

Out[54]:
```
4    413
2    308
3    149
1    130
Name: Years_at_current_residence, dtype: int64
```

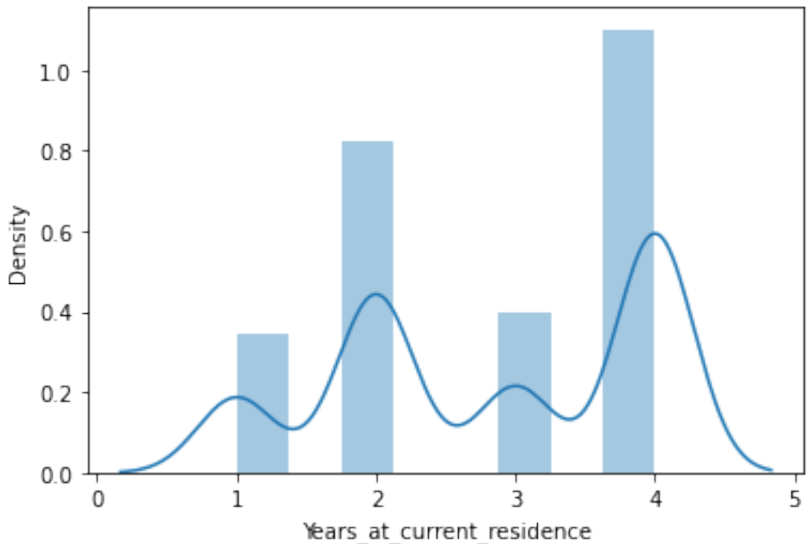
In [55]:
```
data.Years_at_current_residence.value_counts().plot(kind='pie');
```



In [56]:
```
sns.histplot(data['Years_at_current_residence']);
```



In [57]:
```
sns.distplot(data['Years_at_current_residence']);
```



In [58]:
```
data['high_risk_applicant'].groupby(data['Years_at_current_residence']).val
```

Out[58]:
```
Years_at_current_residence  high_risk_applicant
1                           0                       94
                            1                       36
2                           0                      211
                            1                       97
3                           0                      106
                            1                       43
4                           0                      289
                            1                      124
Name: high_risk_applicant, dtype: int64
```

Year At Current Residence :

- Applicants Those Who Completed 4 Years At Current Residence Are 413 In Total, Out-off Which 124 are in Defaulter Zone, 289 are in Non-Defaulter Zone.
- Applicants Those Who Completed 3 Years At Current Residence Are 149 In Total, Out-off Which 43 are in Defaulter Zone, 106 are in Non-Defaulter Zone.
- Applicants Those Who Completed 2 Years At Current Residence Are 308 In Total, Out-off Which 97 are in Defaulter Zone, 211 are in Non-Defaulter Zone.
- Applicants Those Who Completed 1 Years At Current Residence Are 130 In Total, Out-off Which 36 are in Defaulter Zone, 94 are in Non-Defaulter Zone.

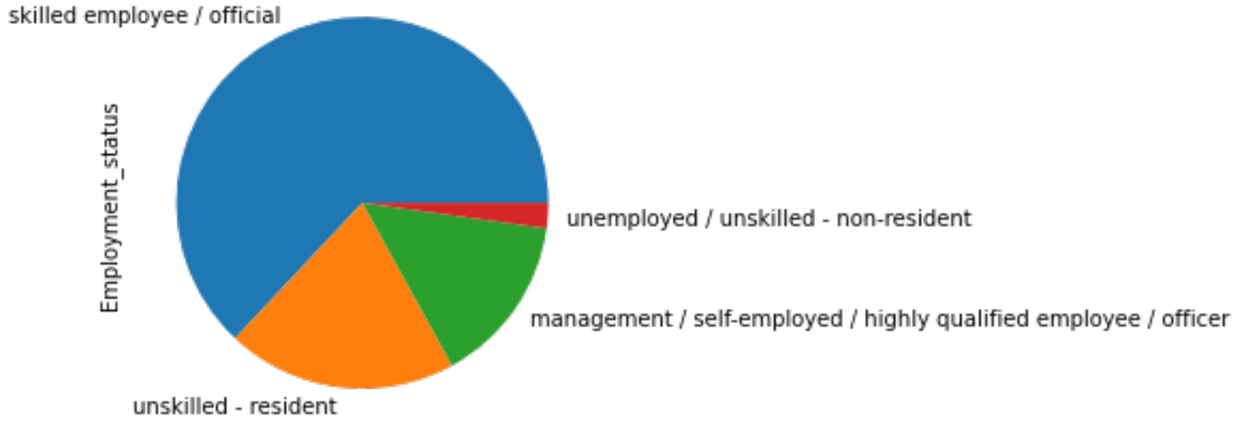
In [59]:
```
data['Employment_status'].unique()
```

Out[59]:
```
array(['skilled employee / official', 'unskilled - resident',
       'management / self-employed / highly qualified employee / officer',
       'unemployed / unskilled - non-resident'], dtype=object)
```

In [60]:
```
data.Employment_status.value_counts()
```

Out[60]:
```
skilled employee / official                                         630
unskilled - resident                                                200
management / self-employed / highly qualified employee / officer    148
unemployed / unskilled - non-resident                                22
Name: Employment_status, dtype: int64
```

In [61]:
```
data.Employment_status.value_counts().plot(kind='pie');
```



In [62]:
```
data['high_risk_applicant'].groupby(data['Employment_status']).value_counts
```

Out[62]:
```
Employment_status                                                   high_risk
_applicant
management / self-employed / highly qualified employee / officer   0
97
                                                                    1
51
skilled employee / official                                         0
444
                                                                    1
186
unemployed / unskilled - non-resident                               0
15
                                                                    1
7
unskilled - resident                                                0
144
                                                                    1
56
Name: high_risk_applicant, dtype: int64
```

Employment_status

- Applicants Those Who Are Marked as ''skilled employee / official'' Type Are 630 In Total Out-off Which 186 are In Defaulter Zone & 444 Are In Non-Defaulter Zone.
- Applicants Those Who Are Marked as ''unskilled - resident'' Type Are 200 In Total Out-off Which 56 are In Defaulter Zone & 144 Are In Non-Defaulter Zone.
- Applicants Those Who Are Marked as ''management / self-employed / highly qualified employee / officer'' Type Are 148 In Total Out-off Which 51 are In Defaulter Zone & 97 Are In Non-Defaulter Zone.
- Applicants Those Who Are Marked as ''unemployed / unskilled - non-resident'' Type Are 22 In Total Out-off Which 7 are In Defaulter Zone & 15 Are In Non-Defaulter Zone.

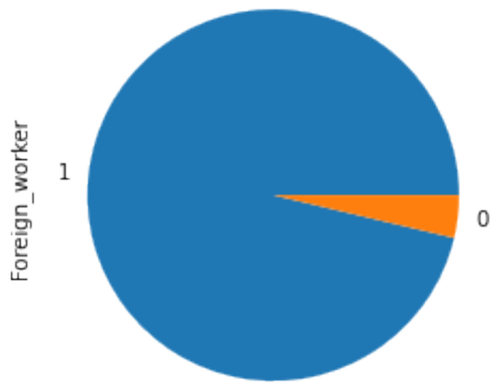
In [63]:
```
data['Foreign_worker'].unique()
```

Out[63]:
```
array([1, 0])
```

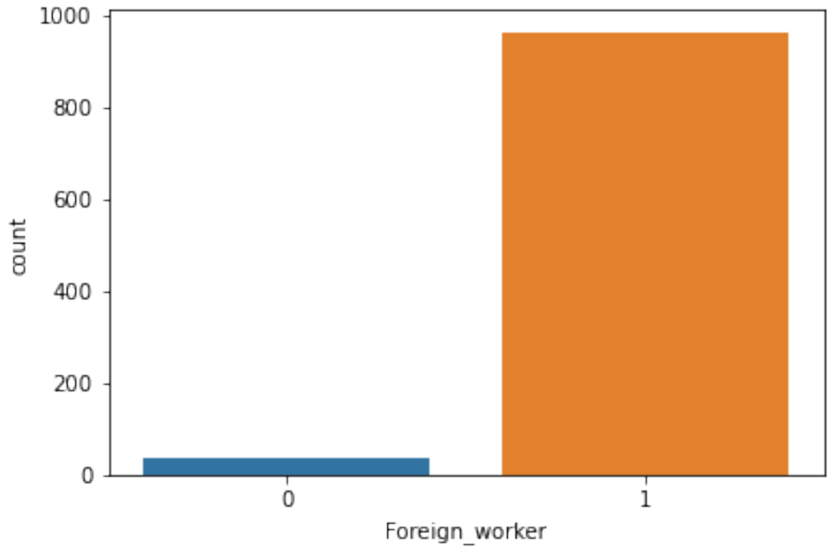
In [64]:
```
data.Foreign_worker.value_counts()
```

Out[64]:
```
1    963
0     37
Name: Foreign_worker, dtype: int64
```

In [65]:
```
data.Foreign_worker.value_counts().plot(kind='pie');
```



In [66]:
```
sns.countplot(data['Foreign_worker']);
```



In [67]:
```
data['high_risk_applicant'].groupby(data['Foreign_worker']).value_counts()
```

Out[67]:
```
Foreign_worker  high_risk_applicant
0               0                      33
                1                       4
1               0                     667
                1                     296
Name: high_risk_applicant, dtype: int64
```

- 963 are Marked As Foreign Worker Out Of Which 296 Are In Defaulter Zone & 667 Are In Non-Defaulter Zone.
- 37 are Not Marked As Foreign Worker Out Of Which 4 Are In Defaulter Zone & 33 Are In Non-Defaulter Zone.

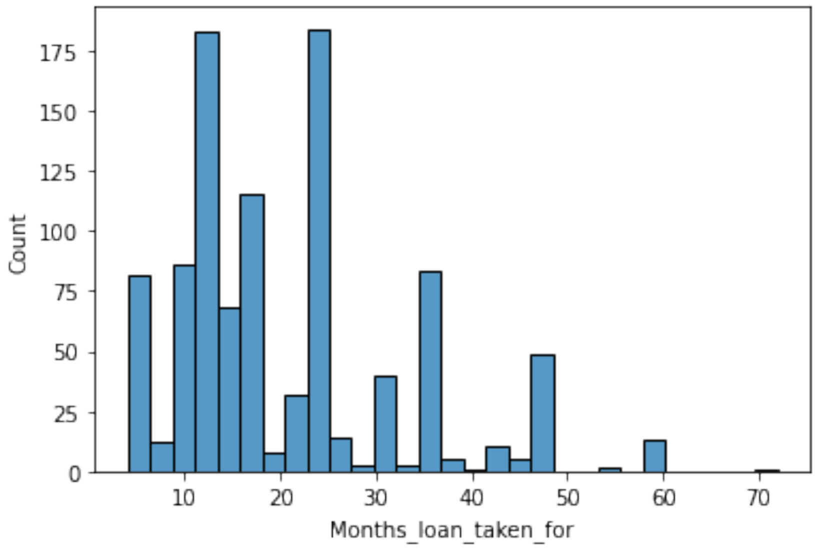
In [68]:
```
data['Months_loan_taken_for'].unique()
```

Out[68]:
```
array([ 6, 48, 12, 42, 24, 36, 30, 15,  9, 10,  7, 60, 18, 45, 11, 27,  8,
       54, 20, 14, 33, 21, 16,  4, 47, 13, 22, 39, 28,  5, 26, 72, 40])
```

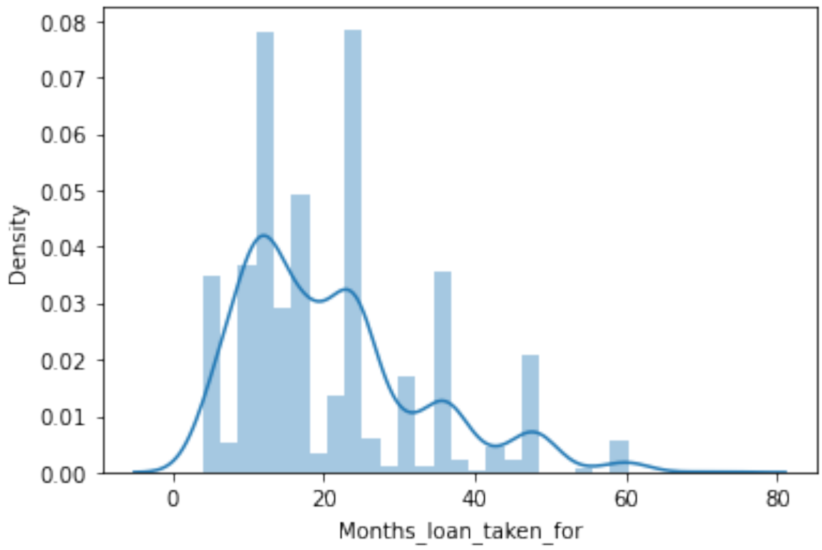
In [69]:
```
data.Months_loan_taken_for.value_counts()
```

Out[69]:
```
24    184
12    179
18    113
36     83
6      75
15     64
9      49
48     48
30     40
21     30
10     28
60     13
27     13
42     11
11      9
20      8
8       7
4       6
45      5
7       5
39      5
14      4
13      4
33      3
28      3
54      2
16      2
22      2
47      1
5       1
26      1
72      1
40      1
Name: Months_loan_taken_for, dtype: int64
```

In [70]:
```
sns.histplot(data['Months_loan_taken_for']);
```



In [71]:
```
sns.distplot(data['Months_loan_taken_for']);
```



In [72]:
```
data['high_risk_applicant'].groupby(data['Months_loan_taken_for']).value_c
```

Out[72]:
```
Months_loan_taken_for  high_risk_applicant
4                      0                        6
5                      0                        1
6                      0                       66
                       1                        9
7                      0                        5
8                      0                        6
                       1                        1
9                      0                       35
                       1                       14
10                     0                       25
                       1                        3
11                     0                        9
12                     0                      130
                       1                       49
13                     0                        4
```

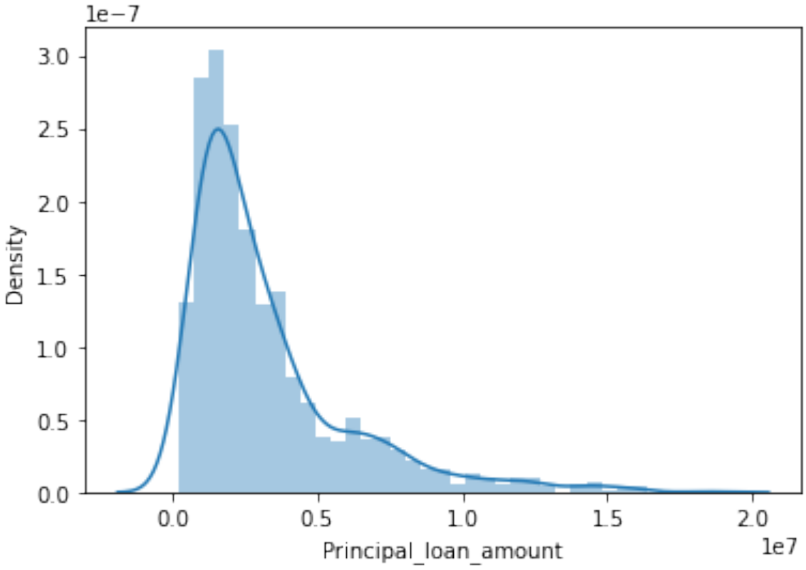
```
14              0       3
                1       1
15              0      52
                1      12
16              0       1
                1       1
18              0      71
                1      42
20              0       7
                1       1
21              0      21
                1       9
22              0       2
24              0     128
                1      56
26              0       1
27              0       8
                1       5
28              0       2
                1       1
30              0      27
                1      13
33              0       2
                1       1
36              0      46
                1      37
39              0       4
                1       1
40              1       1
42              0       8
                1       3
45              1       4
                0       1
47              0       1
48              1      28
                0      20
54              0       1
                1       1
60              0       7
                1       6
72              1       1
Name: high_risk_applicant, dtype: int64
```

Months_loan_taken_for :

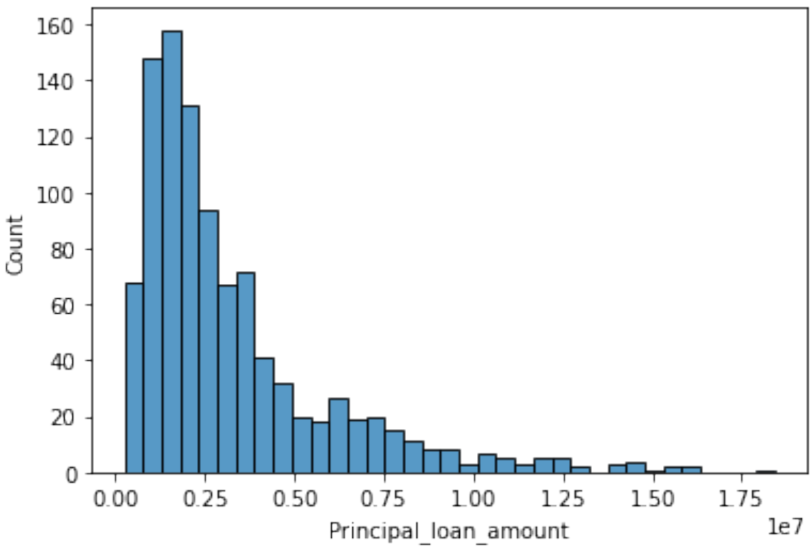
- 184 applicants taken loan for 24 months which is the highest, out of which 56 are in Defaulter Zone & 128 are in Non-Defgaulter Zone.
- 179 applicants taken loan for 12 months which is the 2nd highest, out of which 49 are in Defaulter Zone & 130 are in Non-Defgaulter Zone.
- 113 applicants taken loan for 18 months which is the 3rd highest, out of which 42 are in Defaulter Zone & 71 are in Non-Defgaulter Zone.

In [73]:

```
sns.distplot(data['Principal_loan_amount']);
```



In [74]:
```
sns.histplot(data['Principal_loan_amount']);
```



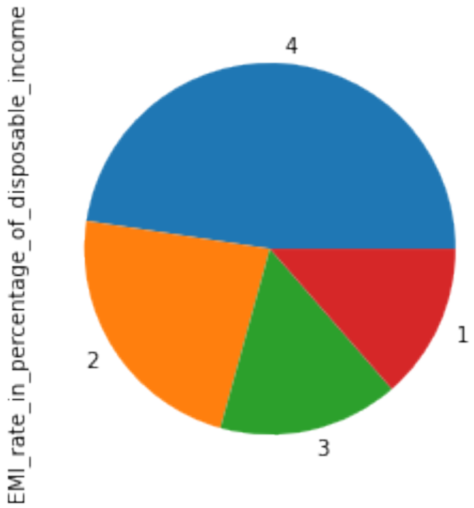
In [75]:
```
data['high_risk_applicant'].groupby(data['Principal_loan_amount']).value_c
```

Out[75]:
```
Principal_loan_amount  high_risk_applicant
250000                 0                      1
276000                 0                      1
338000                 0                      1
339000                 0                      1
343000                 0                      1
                                             ..
15653000               0                      1
15672000               1                      1
15857000               0                      1
15945000               1                      1
18424000               1                      1
Name: high_risk_applicant, Length: 949, dtype: int64
```

In [76]:
```
data.EMI_rate_in_percentage_of_disposable_income.value_counts()
```

Out[76]:
```
4    476
2    231
3    157
1    136
Name: EMI_rate_in_percentage_of_disposable_income, dtype: int64
```

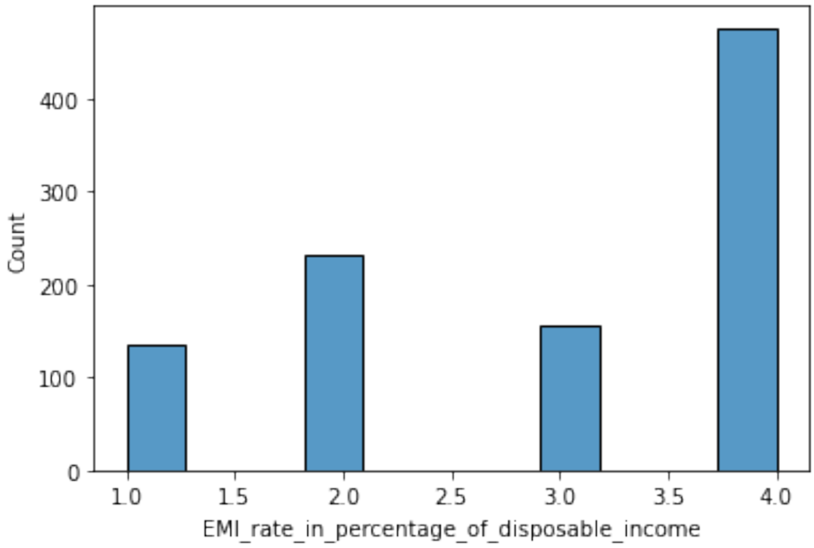
In [77]:
```
data.EMI_rate_in_percentage_of_disposable_income.value_counts().plot(kind=
```



In [78]:
```
data['EMI_rate_in_percentage_of_disposable_income'].unique()
```

Out[78]:
```
array([4, 2, 3, 1])
```

In [79]:
```
sns.histplot(data['EMI_rate_in_percentage_of_disposable_income']);
```



In [80]:
```
data['high_risk_applicant'].groupby(data['EMI_rate_in_percentage_of_disposa
```

```
Out[80]: EMI_rate_in_percentage_of_disposable_income  high_risk_applicant
         1                                            0                      102
                                                      1                       34
         2                                            0                      169
                                                      1                       62
         3                                            0                      112
                                                      1                       45
         4                                            0                      317
                                                      1                      159
         Name: high_risk_applicant, dtype: int64
```

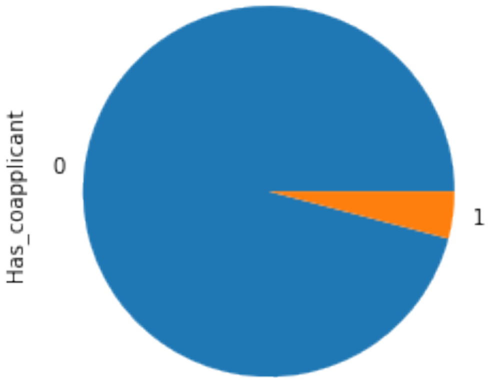
```
In [81]: data.Has_coapplicant.unique()
```

```
Out[81]: array([0, 1])
```

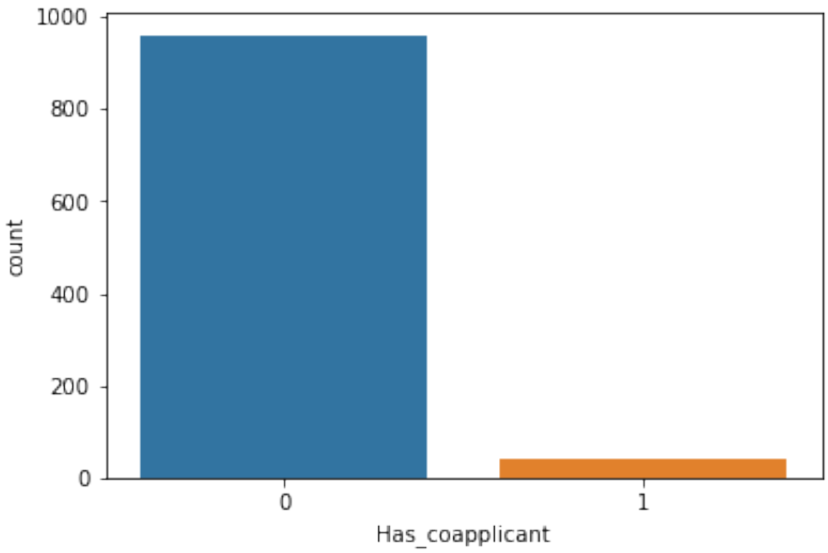
```
In [82]: data.Has_coapplicant.value_counts()
```

```
Out[82]: 0    959
         1     41
         Name: Has_coapplicant, dtype: int64
```

```
In [83]: data.Has_coapplicant.value_counts().plot(kind='pie');
```




```
In [84]: sns.countplot(data['Has_coapplicant']);
```



In [85]:
```
data['high_risk_applicant'].groupby(data['Has_coapplicant']).value_counts(
```

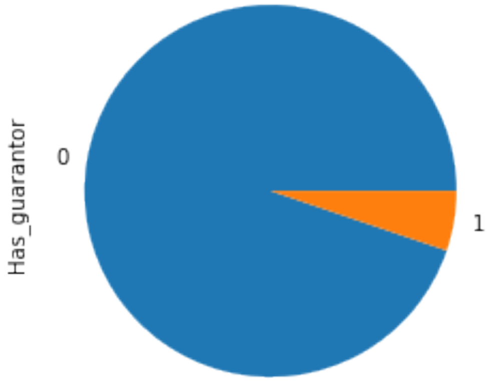
Out[85]:
```
Has_coapplicant  high_risk_applicant
0                0                      677
                 1                      282
1                0                       23
                 1                       18
Name: high_risk_applicant, dtype: int64
```

- 959 People Has No Coapplicant Out-off Which 282 Are In Defaulter's Zone & 677 Are In Non-Defaulter's Zone.
- 41 People Has Coapplicant Out-off Which 18 Are In Defaulter's Zone & 23 Are In Non-Defaulter's Zone.

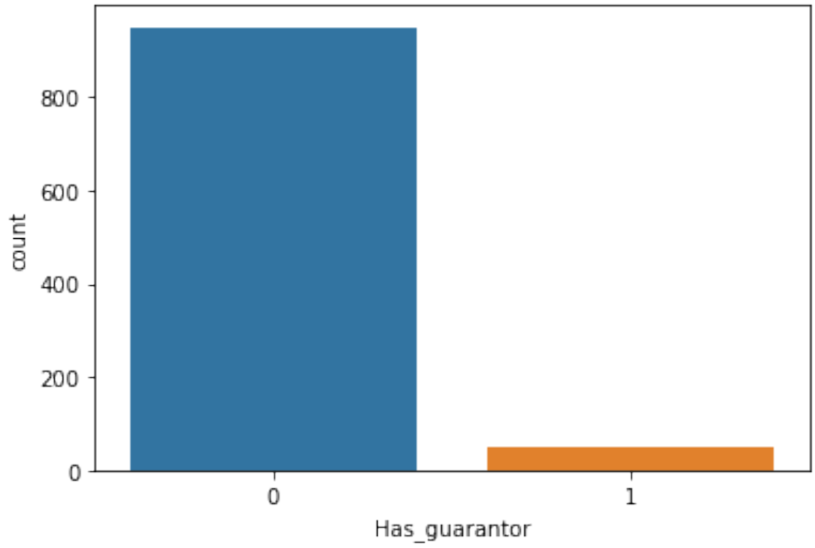
In [86]:
```
data.Has_guarantor.value_counts()
```

Out[86]:
```
0    948
1     52
Name: Has_guarantor, dtype: int64
```

In [87]:
```
data.Has_guarantor.value_counts().plot(kind='pie');
```



In [88]:
```
sns.countplot(data['Has_guarantor']);
```



In [89]:
```
data['high_risk_applicant'].groupby(data['Has_guarantor']).value_counts()
```

Out[89]:
```
Has_guarantor  high_risk_applicant
0              0                      658
               1                      290
1              0                       42
               1                       10
Name: high_risk_applicant, dtype: int64
```

- 948 People Has No Guarantor Out-off Which 290 Are In Defaulter's Zone & 658 Are In Non-Defaulter's Zone.
- 52 People Has Guarantor Out-off Which 10 Are In Defaulter's Zone & 42 Are In Non-Defaulter's Zone.

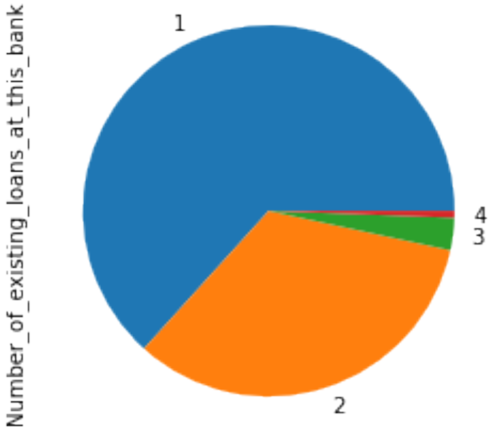
In [90]:
```
data['Number_of_existing_loans_at_this_bank'].unique()
```

Out[90]:
```
array([2, 1, 3, 4])
```

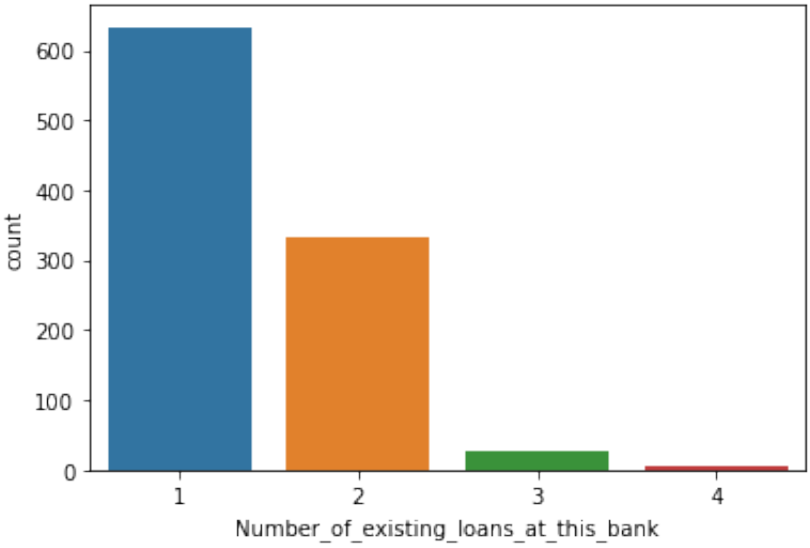
In [91]:
```
data.Number_of_existing_loans_at_this_bank.value_counts()
```

Out[91]:
```
1    633
2    333
3     28
4      6
Name: Number_of_existing_loans_at_this_bank, dtype: int64
```

In [92]:
```
data.Number_of_existing_loans_at_this_bank.value_counts().plot(kind='pie')
```



In [93]:
```
sns.countplot(data['Number_of_existing_loans_at_this_bank']);
```



In [94]:
```
data['high_risk_applicant'].groupby(data['Number_of_existing_loans_at_this_
```

Out[94]:
```
Number_of_existing_loans_at_this_bank  high_risk_applicant
1                                      0                      433
                                       1                      200
2                                      0                      241
                                       1                       92
3                                      0                       22
                                       1                        6
4                                      0                        4
                                       1                        2
Name: high_risk_applicant, dtype: int64
```

Number_of_existing_loans_at_this_bank:

- 633 Applicants Having 1 Number Of Loan At This Bank Out Of Which 200 Are In Defaulters Zone, 433 Are In Non-Defaulters Zone.
- 333 Applicants Having 2 Number Of Loans At This Bank Out Of Which 92 Are In Defaulters Zone, 241 Are In Non-Defaulters Zone.
- 28 Applicants Having 3 Number Of Loans At This Bank Out Of Which 6 Are In Defaulters Zone, 22 Are In Non-Defaulters Zone.
- 6 Applicants Having 4 Number Of Loans At This Bank Out Of Which 2 Are In Defaulters Zone, 4 Are In Non-Defaulters Zone.

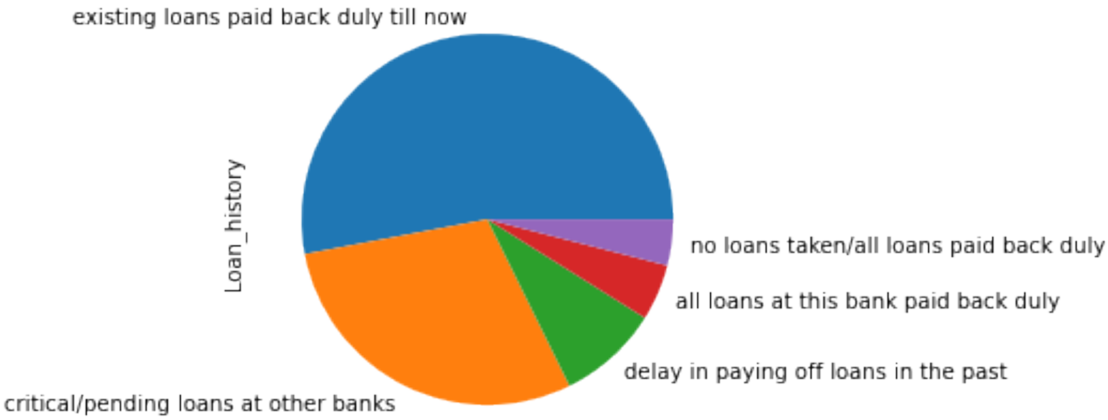
In [95]:
```
data['Loan_history'].unique()
```

Out[95]:
```
array(['critical/pending loans at other banks',
       'existing loans paid back duly till now',
       'delay in paying off loans in the past',
       'no loans taken/all loans paid back duly',
       'all loans at this bank paid back duly'], dtype=object)
```

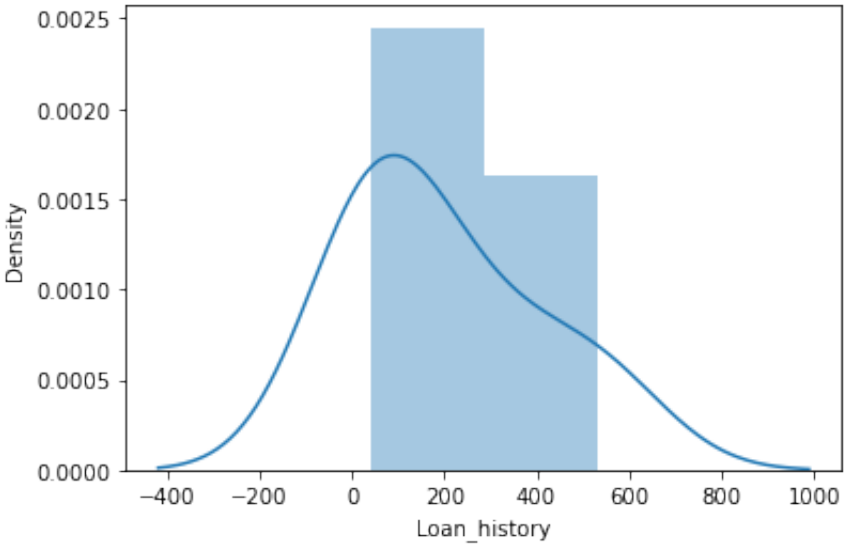
In [96]:
```
data['Loan_history'].value_counts()
```

Out[96]:
```
existing loans paid back duly till now     530
critical/pending loans at other banks      293
delay in paying off loans in the past       88
all loans at this bank paid back duly       49
no loans taken/all loans paid back duly     40
Name: Loan_history, dtype: int64
```

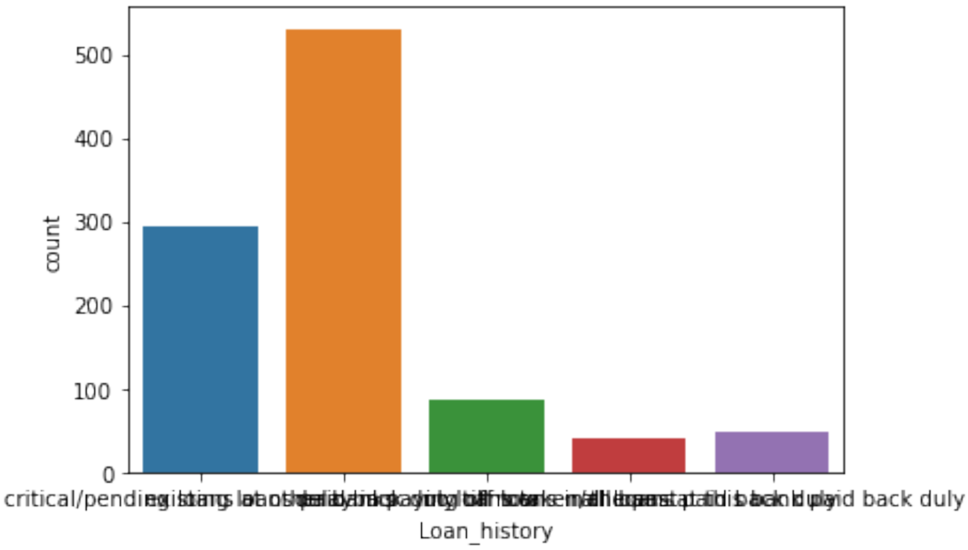
In [97]:
```
data.Loan_history.value_counts().plot(kind='pie');
```



In [98]:
```
sns.distplot(data.Loan_history.value_counts());
```



In [99]:
```
sns.countplot(data['Loan_history']);
```



In [100…

```
data['high_risk_applicant'].groupby(data['Loan_history']).value_counts()
```

Out[100…

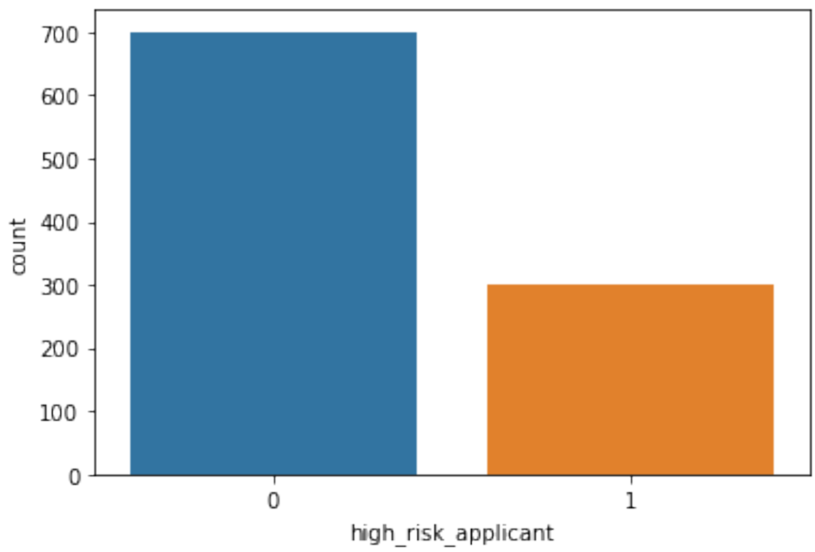
```
Loan_history                               high_risk_applicant
all loans at this bank paid back duly      1                       28
                                           0                       21
critical/pending loans at other banks      0                      243
                                           1                       50
delay in paying off loans in the past      0                       60
                                           1                       28
existing loans paid back duly till now     0                      361
                                           1                      169
no loans taken/all loans paid back duly    1                       25
                                           0                       15
Name: high_risk_applicant, dtype: int64
```

Loan_history

- all loans at this bank paid back duly : Total : 49, Non-Defaulter's Zone : 21, Defaulter's Zone : 28
- critical/pending loans at other banks : Total : 293, Non-Defaulter's Zone : 243, Defaulter's Zone : 50
- delay in paying off loans in the past : Total : 88, Non-Defaulter's Zone : 60, Defaulter's Zone : 28
- existing loans paid back duly till now : Total : 530, Non-Defaulter's Zone : 361, Defaulter's Zone : 169
- no loans taken/all loans paid back duly : Total : 40, Non-Defaulter's Zone : 15, Defaulter's Zone : 25

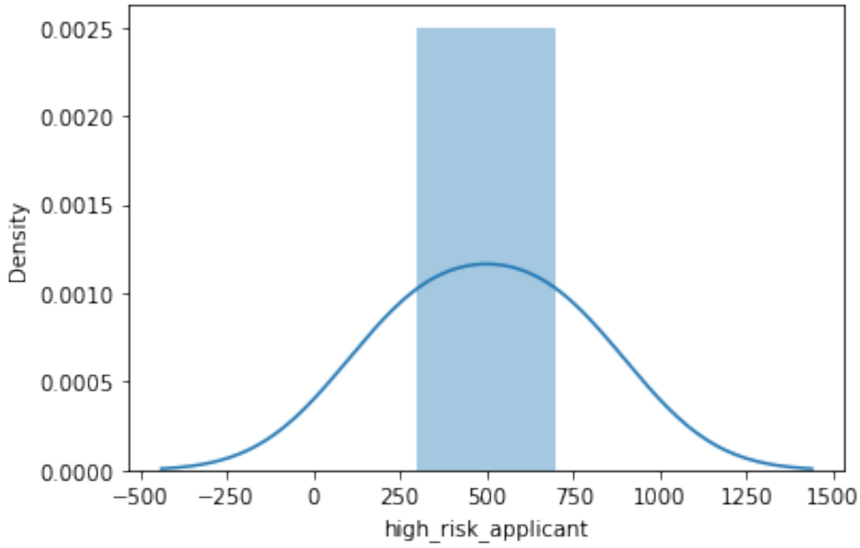
In [101…

```
sns.countplot(data['high_risk_applicant']);
```



In [102…

```
sns.distplot(data.high_risk_applicant.value_counts());
```



In [103…

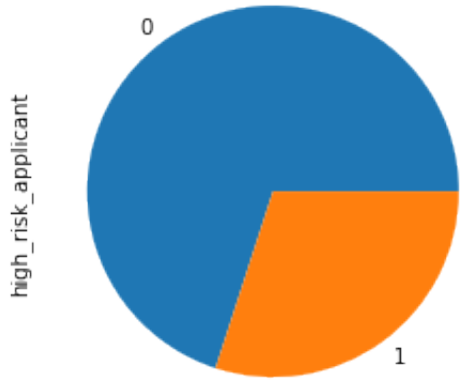
```
data.high_risk_applicant.value_counts()
```

Out[103…

```
0    700
1    300
Name: high_risk_applicant, dtype: int64
```

In [104…

```
data.high_risk_applicant.value_counts().plot(kind='pie');
```



In [105…

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column                                   Non-Null Count  Dtype 
---  ------                                   --------------  ----- 
 0   applicant_id                             1000 non-null   int64 
 1   Primary_applicant_age_in_years           1000 non-null   int64 
 2   Gender                                   1000 non-null   object
 3   Marital_status                           1000 non-null   object
 4   Number_of_dependents                     1000 non-null   int64 
 5   Housing                                  1000 non-null   object
 6   Years_at_current_residence               1000 non-null   int64 
 7   Employment_status                        1000 non-null   object
 8   Foreign_worker                           1000 non-null   int64 
 9   loan_application_id                      1000 non-null   object
 10  Months_loan_taken_for                    1000 non-null   int64 
 11  Principal_loan_amount                    1000 non-null   int64 
 12  EMI_rate_in_percentage_of_disposable_income  1000 non-null   int64 
 13  Has_coapplicant                          1000 non-null   int64 
 14  Has_guarantor                            1000 non-null   int64 
 15  Number_of_existing_loans_at_this_bank    1000 non-null   int64 
 16  Loan_history                             1000 non-null   object
 17  high_risk_applicant                      1000 non-null   int64 
dtypes: int64(12), object(6)
memory usage: 180.7+ KB
```

Applying Label Encoder For Categorical Variables

In [106…

```
from sklearn.preprocessing import LabelEncoder
```

In [107…

```
Le = LabelEncoder()

data['Gender'] = Le.fit_transform(data['Gender'])

data['Marital_status'] = Le.fit_transform(data['Marital_status'])

data['Housing'] = Le.fit_transform(data['Housing'])

data['Employment_status'] = Le.fit_transform(data['Employment_status'])

data['Loan_history'] = Le.fit_transform(data['Loan_history'])
```

In [108…

```
data['applicant_id'] = Le.fit_transform(data['applicant_id'])
data['loan_application_id'] = Le.fit_transform(data['loan_application_id']
```

In [109…

```
data.head()
```

Out[109...

	applicant_id	Primary_applicant_age_in_years	Gender	Marital_status	Number_of_depen
0	436	67	1	3	
1	115	22	0	1	
2	380	49	1	3	
3	117	45	1	3	
4	713	53	1	3	

In [110...

```
data.info()
```

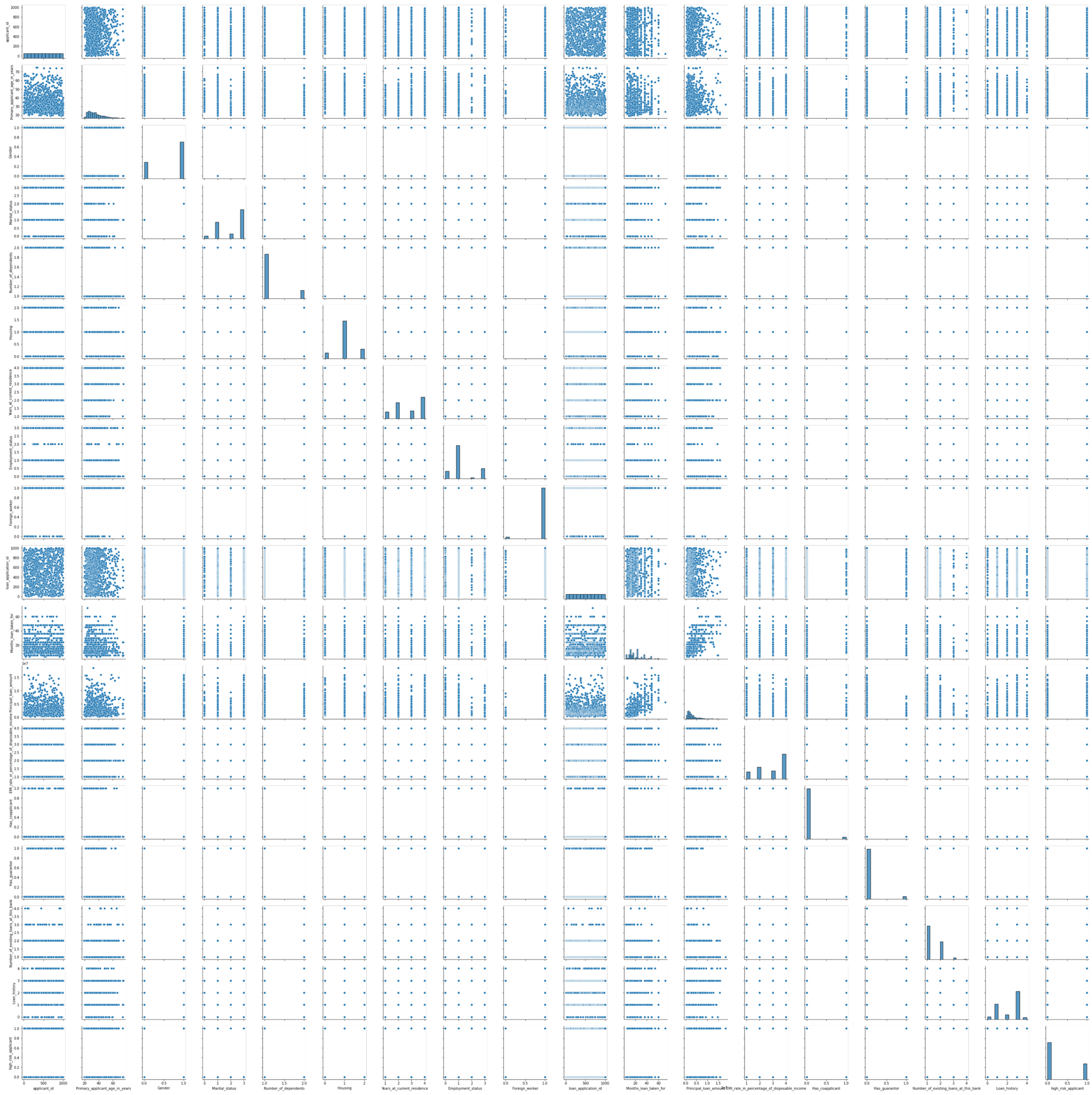
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   applicant_id                            1000 non-null   int64
 1   Primary_applicant_age_in_years          1000 non-null   int64
 2   Gender                                  1000 non-null   int64
 3   Marital_status                          1000 non-null   int64
 4   Number_of_dependents                    1000 non-null   int64
 5   Housing                                 1000 non-null   int64
 6   Years_at_current_residence              1000 non-null   int64
 7   Employment_status                       1000 non-null   int64
 8   Foreign_worker                          1000 non-null   int64
 9   loan_application_id                     1000 non-null   int64
 10  Months_loan_taken_for                   1000 non-null   int64
 11  Principal_loan_amount                   1000 non-null   int64
 12  EMI_rate_in_percentage_of_disposable_income  1000 non-null   int64
 13  Has_coapplicant                         1000 non-null   int64
 14  Has_guarantor                           1000 non-null   int64
 15  Number_of_existing_loans_at_this_bank   1000 non-null   int64
 16  Loan_history                            1000 non-null   int64
 17  high_risk_applicant                     1000 non-null   int64
dtypes: int64(18)
memory usage: 180.7 KB
```

In [111...

```
sns.pairplot(data)
```

Out[111…

```
<seaborn.axisgrid.PairGrid at 0x7f7ed38189d0>
```



Would a person with critical credit history be more creditworthy?

In [112…

```
data['high_risk_applicant'].groupby(data['Loan_history']).value_counts()
```

Out[112…

```
Loan_history  high_risk_applicant
0             1                       28
              0                       21
1             0                      243
              1                       50
2             0                       60
              1                       28
3             0                      361
              1                      169
4             1                       25
              0                       15
Name: high_risk_applicant, dtype: int64
```

Loan_history

- all loans at this bank paid back duly : Total : 49, Non-Defaulter's Zone : 21, Defaulter's Zone : 28
- critical/pending loans at other banks : Total : 293, Non-Defaulter's Zone : 243, Defaulter's Zone : 50
- delay in paying off loans in the past : Total : 88, Non-Defaulter's Zone : 60, Defaulter's Zone : 28
- existing loans paid back duly till now : Total : 530, Non-Defaulter's Zone : 361, Defaulter's Zone : 169
- no loans taken/all loans paid back duly : Total : 40, Non-Defaulter's Zone : 15, Defaulter's Zone : 25

According To Data We Can Assume That A Person With Critical Credit History Can Be More Creditworthy As Out-off 293 Only 50 Are In Defaulter's Zone, as Compared To Others It Seems To Be More Creditworthy.

Are young people more creditworthy?

- Applicants having 23 - 30 age are more in quantity as compared to others

In [113…

```
data['high_risk_applicant'].groupby(data['Primary_applicant_age_in_years']<
```

Out[113…

```
Primary_applicant_age_in_years  high_risk_applicant
False                           0                      466
                                1                      163
True                            0                      234
                                1                      137
Name: high_risk_applicant, dtype: int64
```

In [114…

```
data['high_risk_applicant'].groupby(data['Primary_applicant_age_in_years']>
```

Out[114…

```
Primary_applicant_age_in_years  high_risk_applicant
False                           0                      263
                                1                      148
True                            0                      437
                                1                      152
Name: high_risk_applicant, dtype: int64
```

In [115…

```
data['high_risk_applicant'].groupby(data['Primary_applicant_age_in_years']=
```

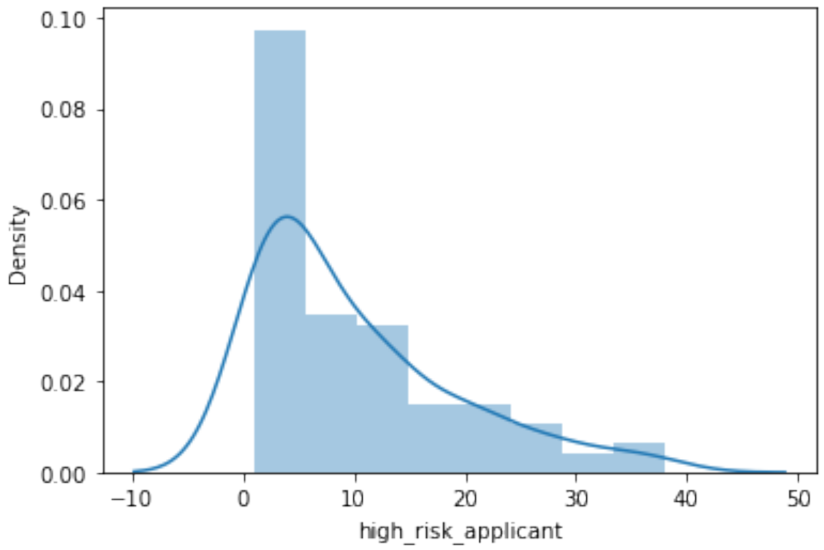
Out[115…

```
Primary_applicant_age_in_years  high_risk_applicant
False                           0                      672
                                1                      280
True                            0                       28
                                1                       20
Name: high_risk_applicant, dtype: int64
```

We can consider young people more creditworthy taking there quantity in consideration as compared to others.

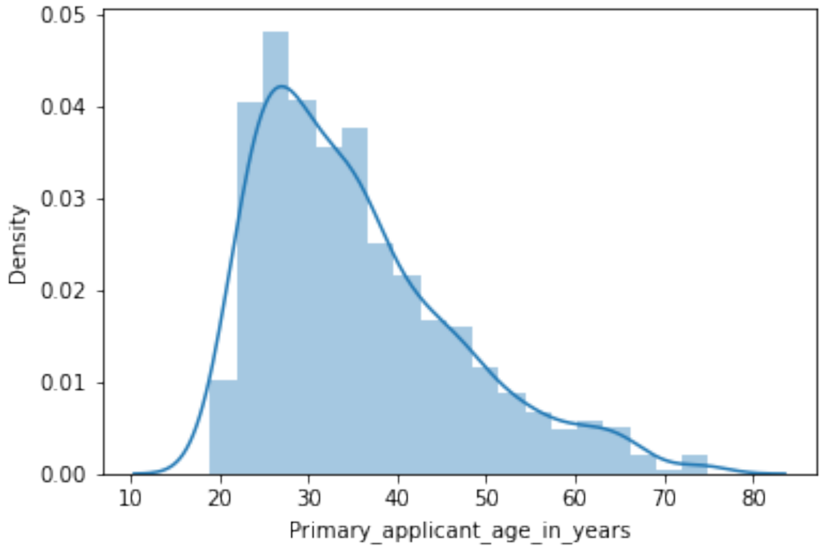
In [116…

```
sns.distplot(data['high_risk_applicant'].groupby(data['Primary_applicant_a
```



In [117…

```
sns.distplot(data['Primary_applicant_age_in_years']);
```



In [118…

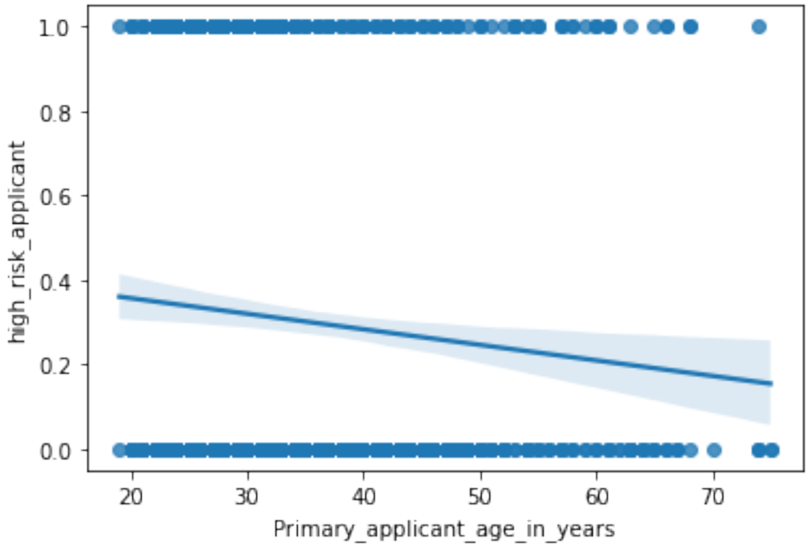
```
data[["high_risk_applicant", "Primary_applicant_age_in_years"]].corr()
```

Out[118...

	high_risk_applicant	Primary_applicant_age_in_years
high_risk_applicant	1.000000	-0.091127
Primary_applicant_age_in_years	-0.091127	1.000000

In [119...

```
sns.regplot(x=data['Primary_applicant_age_in_years'], y=data['high_risk_app
```



Would a person with more credit accounts be more creditworthy?

In [120...

```
data['high_risk_applicant'].groupby(data['Number_of_existing_loans_at_this_
```

Out[120...

```
Number_of_existing_loans_at_this_bank  high_risk_applicant
1                                      0                      433
                                       1                      200
2                                      0                      241
                                       1                       92
3                                      0                       22
                                       1                        6
4                                      0                        4
                                       1                        2
Name: high_risk_applicant, dtype: int64
```

In [121...

```
200/(200+433)
```

Out[121...

```
0.315955766192733
```

In [122...

```
92/(241+92)
```

Out[122...

```
0.27627627627627627
```

In [123...

```
6/(6+22)
```

Out[123…
```
0.21428571428571427
```

In [124…
```
2/(4+2)
```

Out[124…
```
0.3333333333333333
```

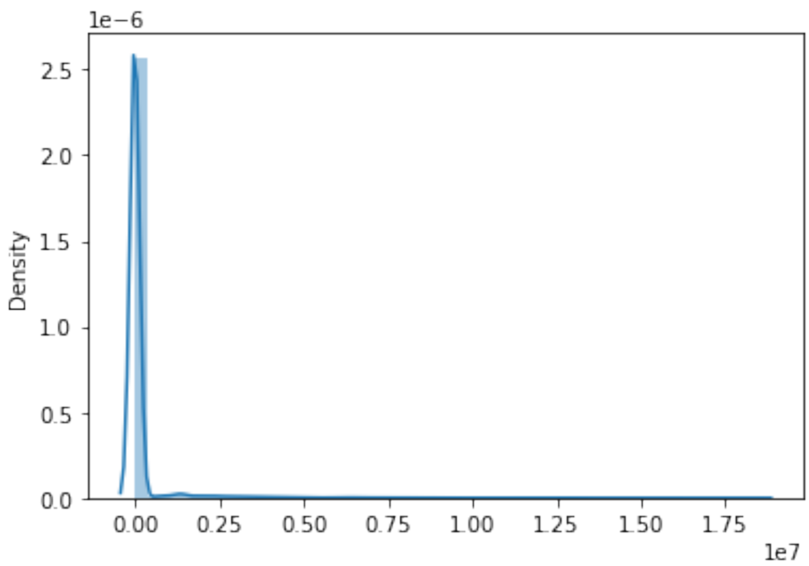
Number_of_existing_loans_at_this_bank:

- 633 Applicants Having 1 Number Of Loan At This Bank Out Of Which 200 Are In Defaulters Zone, 433 Are In Non-Defaulters Zone.
- 333 Applicants Having 2 Number Of Loans At This Bank Out Of Which 92 Are In Defaulters Zone, 241 Are In Non-Defaulters Zone.
- 28 Applicants Having 3 Number Of Loans At This Bank Out Of Which 6 Are In Defaulters Zone, 22 Are In Non-Defaulters Zone.
- 6 Applicants Having 4 Number Of Loans At This Bank Out Of Which 2 Are In Defaulters Zone, 4 Are In Non-Defaulters Zone.

We can assume a person with more credit accounts be more creditworthy according to the current scenario, but we can't be 100 % sure as we just have 1000 entries out of which only 6 applicants have 4 number of loans at this bank out of which 2 are in defaulter's zone and 4 are in non-defaulter's zone.

In [125…
```
sns.distplot(data)
```

Out[125…
```
<AxesSubplot:ylabel='Density'>
```



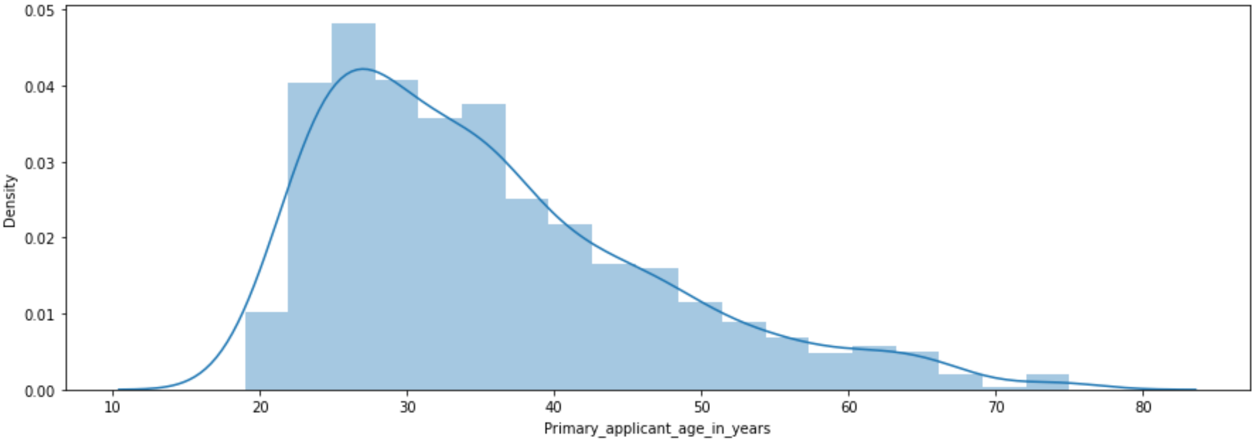
In [126…
```
data.head(2)
```

Out[126...

	applicant_id	Primary_applicant_age_in_years	Gender	Marital_status	Number_of_depen
0	436	67	1	3	
1	115	22	0	1	

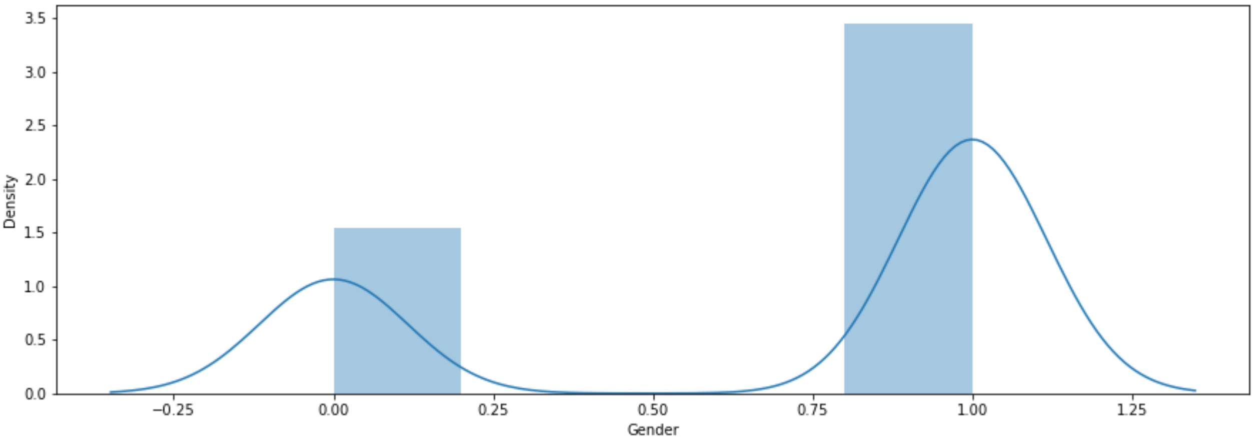
In [127...

```
plt.figure(figsize=(15,5))
sns.distplot(data['Primary_applicant_age_in_years']);
```



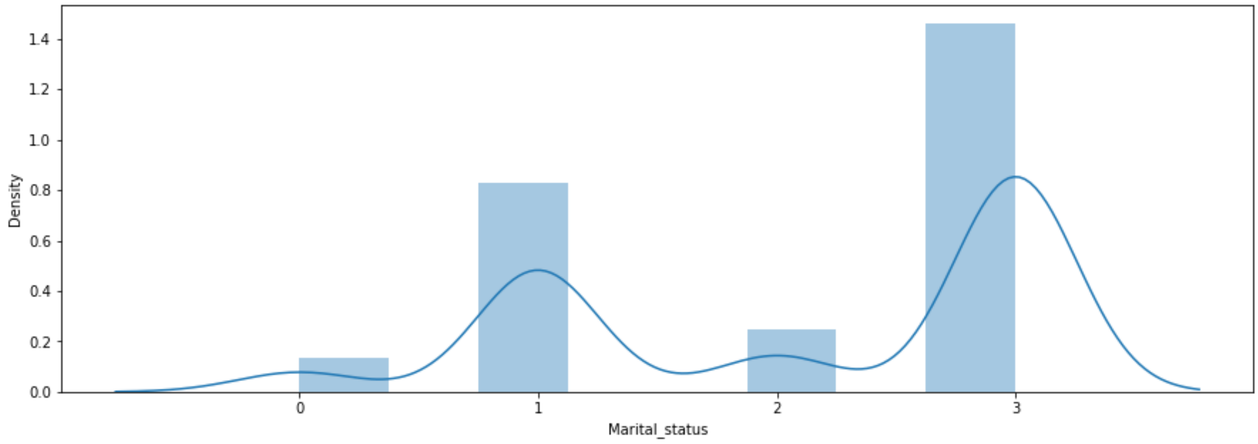
In [128...

```
plt.figure(figsize=(15,5))
sns.distplot(data['Gender']);
```



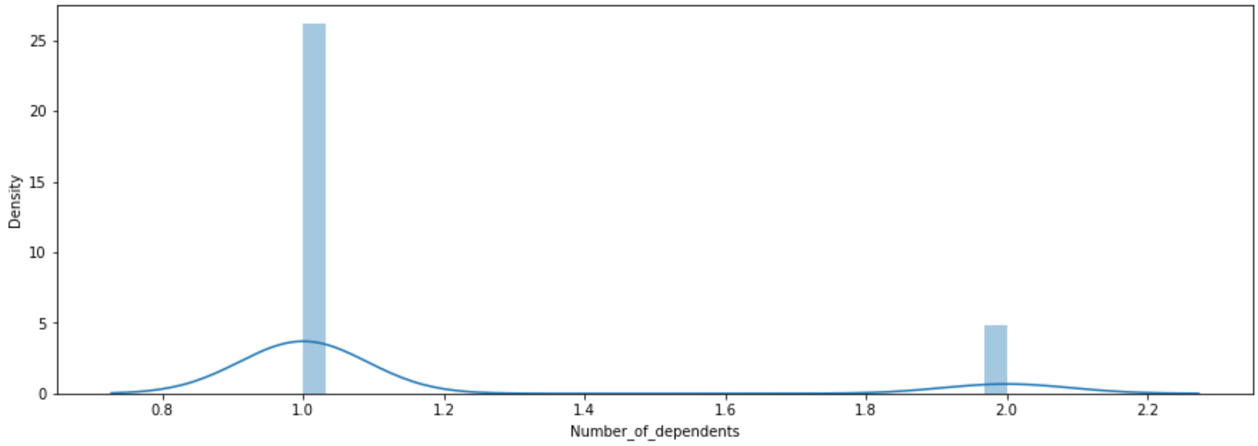
In [129...

```
plt.figure(figsize=(15,5))
sns.distplot(data['Marital_status']);
```



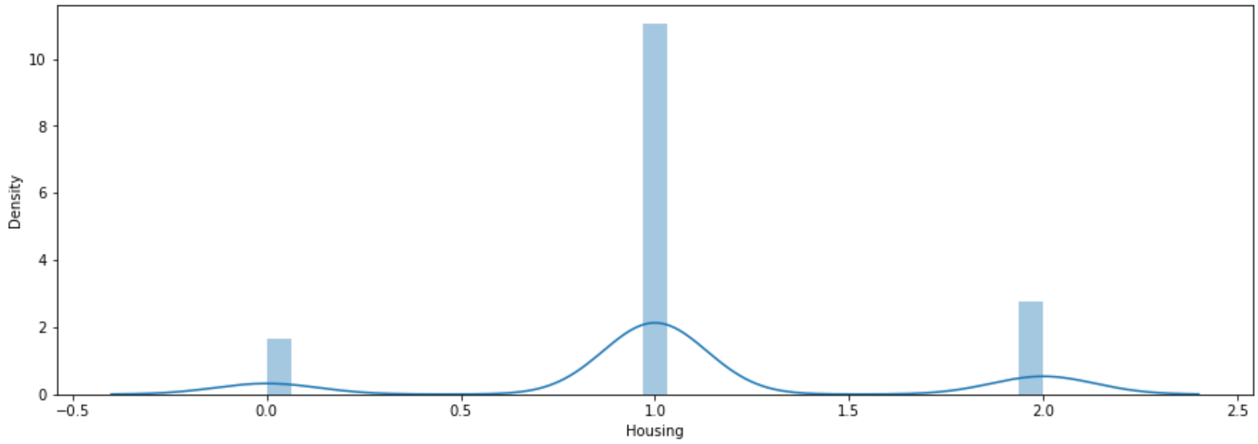
In [130…

```
plt.figure(figsize=(15,5))
sns.distplot(data['Number_of_dependents']);
```



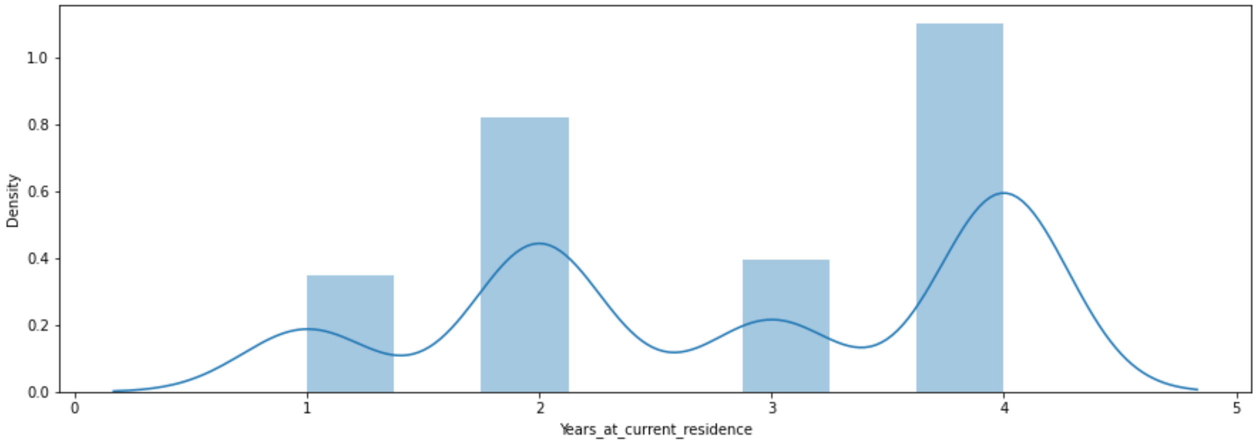
In [131…

```
plt.figure(figsize=(15,5))
sns.distplot(data['Housing']);
```



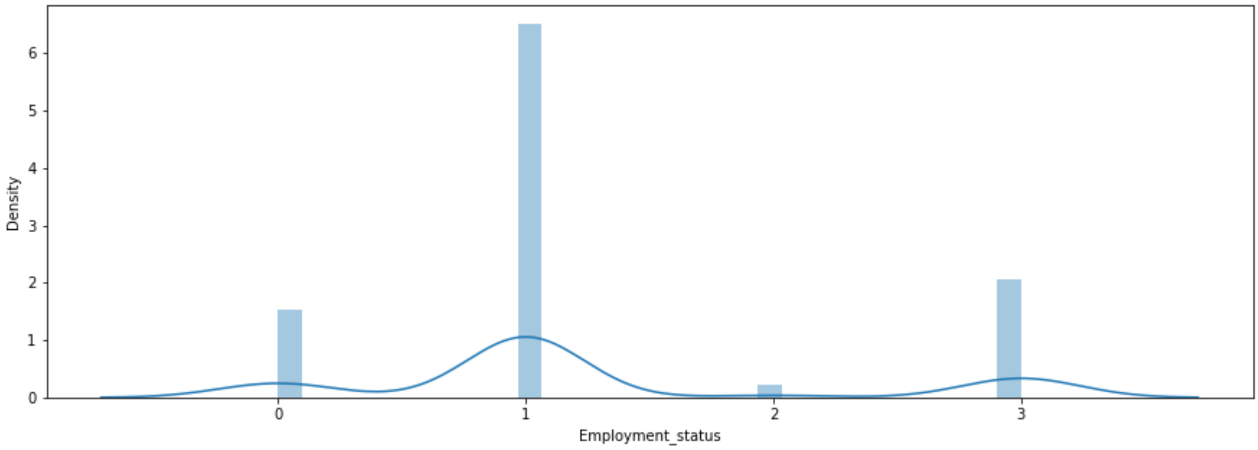
In [132…

```
plt.figure(figsize=(15,5))
sns.distplot(data['Years_at_current_residence']);
```



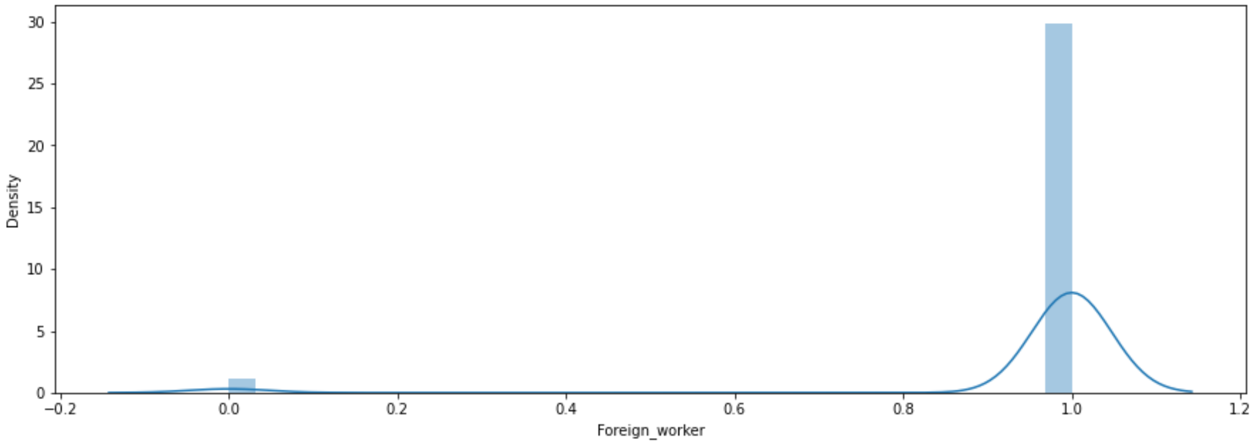
In [133…

```
plt.figure(figsize=(15,5))
sns.distplot(data['Employment_status']);
```



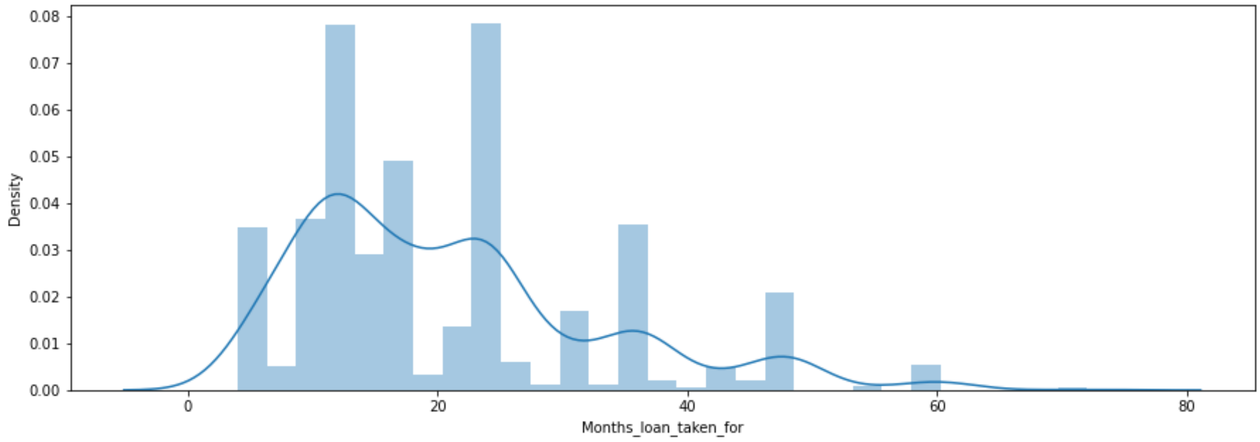
In [134…

```
plt.figure(figsize=(15,5))
sns.distplot(data['Foreign_worker']);
```



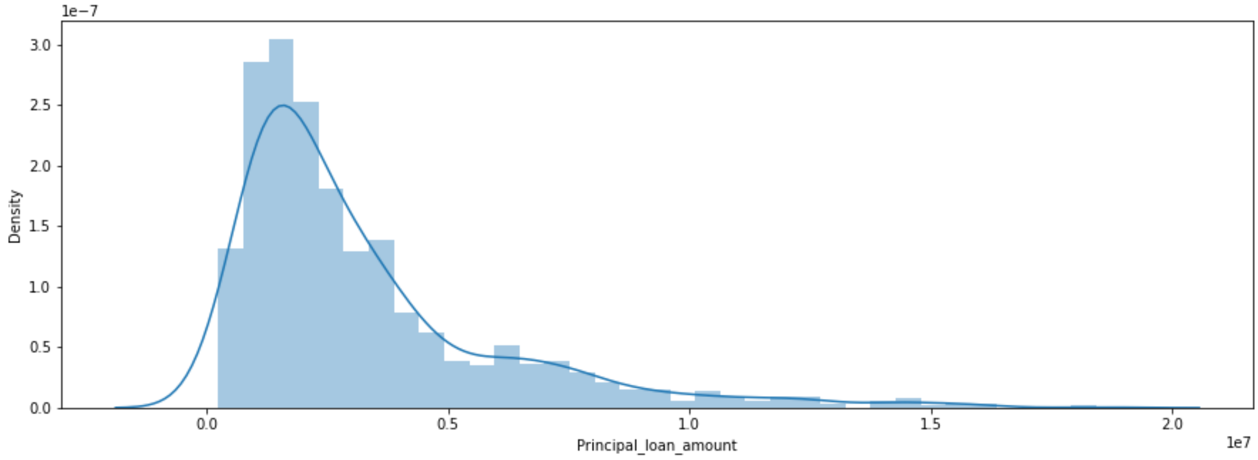
In [135…

```
plt.figure(figsize=(15,5))
sns.distplot(data['Months_loan_taken_for']);
```



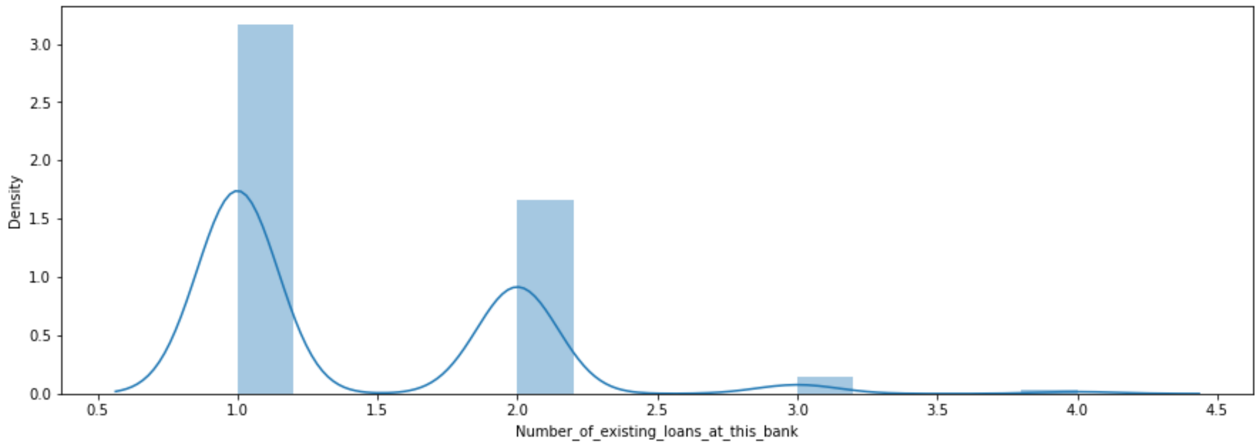
In [136...

```
plt.figure(figsize=(15,5))
sns.distplot(data['Principal_loan_amount']);
```



In [137...

```
plt.figure(figsize=(15,5))
sns.distplot(data['Number_of_existing_loans_at_this_bank']);
```



In [138...

```
data.head(1)
```

Out[138…

	applicant_id	Primary_applicant_age_in_years	Gender	Marital_status	Number_of_depen
0	436	67	1	3	

In [139…

```
data['high_risk_applicant'].groupby(data['Gender']).value_counts()
```

Out[139…

```
Gender  high_risk_applicant
0       0                      201
        1                      109
1       0                      499
        1                      191
Name: high_risk_applicant, dtype: int64
```

In [140…

```
109/310
```

Out[140…

```
0.35161290322580646
```

In [141…

```
191/690
```

Out[141…

```
0.2768115942028985
```

In [142…

```
data[['Primary_applicant_age_in_years','Gender','Marital_status','Number_o
```

Out[142...

	Primary_applicant_age_in_years	Gender
Primary_applicant_age_in_years	1.000000	0.161694
Gender	0.161694	1.000000
Marital_status	0.147954	0.748342
Number_of_dependents	0.118201	0.203431
Housing	-0.301419	-0.219844
Years_at_current_residence	0.266419	-0.013818
Employment_status	-0.001637	-0.041278
Foreign_worker	0.006151	-0.051202
Months_loan_taken_for	-0.036136	0.081432
Principal_loan_amount	0.032716	0.093482
EMI_rate_in_percentage_of_disposable_income	0.058266	0.086302
Has_coapplicant	-0.018357	0.007742
Has_guarantor	-0.023923	0.010907
Number_of_existing_loans_at_this_bank	0.149254	0.094260
Loan_history	-0.157261	-0.059183
high_risk_applicant	-0.091127	-0.075493

In [143...

```
data[['Primary_applicant_age_in_years','Gender','Marital_status','Number_o
```

Out[143...

	Primary_applicant_age_in_years	Gender
Primary_applicant_age_in_years	1.000000	0.161694
Gender	0.161694	1.000000
Marital_status	0.147954	0.748342
Number_of_dependents	0.118201	0.203431
Housing	-0.301419	-0.219844
Years_at_current_residence	0.266419	-0.013818
Employment_status	-0.001637	-0.041278
Foreign_worker	0.006151	-0.051202
Months_loan_taken_for	-0.036136	0.081432
Principal_loan_amount	0.032716	0.093482
EMI_rate_in_percentage_of_disposable_income	0.058266	0.086302
Has_coapplicant	-0.018357	0.007742
Has_guarantor	-0.023923	0.010907
Number_of_existing_loans_at_this_bank	0.149254	0.094260
Loan_history	-0.157261	-0.059183
high_risk_applicant	-0.091127	-0.075493

In [144...

```
plt.figure(figsize=(17,7))
sns.heatmap(data[['Primary_applicant_age_in_years','Gender','Marital_statu
plt.show()
```

In [145…
```
data.to_csv('data2')
```

TASK-2

Develop the ML model(s) to predict the credit risk(low or high) for a given applicant.

Business Constraint: Note that it is worse to state an applicant as a low credit risk when they are actually a high risk(Type2) - False Negative , than it is to state an applicant to be a high credit risk when they aren't(Type1) - False Positive.

In [146…
```
data.drop(['loan_application_id','applicant_id','Marital_status','Number_o
```

In [147…
```
data.head(2)
```

Out[147…

	Primary_applicant_age_in_years	Gender	Employment_status	Months_loan_taken_for	Nu
0	67	1	1	6	
1	22	0	1	48	

In [148…
```
X = data.loc[:, data.columns != 'high_risk_applicant' ] # independent vari
y = data.loc[:, data.columns == 'high_risk_applicant'] #target variable
```

In [149…
```
X = pd.get_dummies(X, drop_first=True)
```

In [150…
```
X.head()
```

Out[150…

	Primary_applicant_age_in_years	Gender	Employment_status	Months_loan_taken_for	Nu
0	67	1	1	6	
1	22	0	1	48	
2	49	1	3	12	
3	45	1	1	42	
4	53	1	1	24	

In [151…
```
y.head()
```

Out[151…

	high_risk_applicant
0	0
1	1
2	0
3	0
4	1

In [152…

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import LogisticRegression
```

In [153…

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_s
```

In [154…

```
X_train.head()
```

Out[154…

	Primary_applicant_age_in_years	Gender	Employment_status	Months_loan_taken_for
834	25	0	3	15
227	53	1	0	12
471	23	0	1	6
929	43	1	3	12
457	35	1	1	12

In [155…

```
X_test.head()
```

Out[155…

	Primary_applicant_age_in_years	Gender	Employment_status	Months_loan_taken_for
518	43	1	1	6
871	46	1	1	6
797	22	0	3	12
274	34	1	3	30
325	39	1	3	8

In [156…

```
X_train.shape,X_test.shape
```

Out[156…

```
((700, 6), (300, 6))
```

In [157…

```
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

Out[157…

```
▾ LogisticRegression
LogisticRegression()
```

In [158…

```
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.forı
```

```
Accuracy of logistic regression classifier on test set: 0.70
```

In [159…

```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
```

In [160…

```
confusion_matrix
```

Out[160…

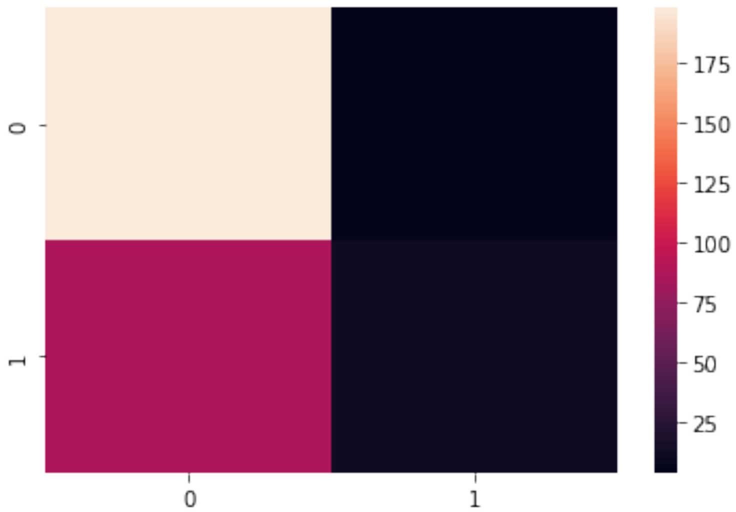
```
array([[198,   4],
       [ 87,  11]])
```

In [161…

```
sns.heatmap(confusion_matrix)
```

Out[161…

<AxesSubplot:>



In [162…

```
TN = 198
FP = 87
FN = 4
TP = 11
```

In [163…

```
TPR = 11/(11+4) #TPR = TP/P
TPR
```

Out[163...

```
0.7333333333333333
```

In [164...

```
TNR = 198/(198+87) #TNR = TN/N
TNR
```

Out[164...

```
0.6947368421052632
```

In [165...

```
FPR = 87/(198+87) #FPR = FP/N
FPR
```

Out[165...

```
0.30526315789473685
```

In [166...

```
FNR = 4/(11+4) #FNR = FN/p
FNR
```

Out[166...

```
0.26666666666666666
```

In [167...

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.69      0.98      0.81       202
           1       0.73      0.11      0.19        98

    accuracy                           0.70       300
   macro avg       0.71      0.55      0.50       300
weighted avg       0.71      0.70      0.61       300
```