In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt,seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [18]:
```
df=pd.read_csv(r"D:\Python\EDA  ASSIGNMENT\application_data.csv")
df.head()
```

Out[18]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT
0	100002	1	Cash loans	M	N	Y	
1	100003	0	Cash loans	F	N	N	
2	100004	0	Revolving loans	M	Y	Y	
3	100006	0	Cash loans	F	N	Y	
4	100007	0	Cash loans	M	N	Y	

5 rows × 122 columns

In [12]:
```
df.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 122 columns):
 #    Column                       Dtype  
---   ------                       -----  
 0    SK_ID_CURR                   int64  
 1    TARGET                       int64  
 2    NAME_CONTRACT_TYPE           object 
 3    CODE_GENDER                  object 
 4    FLAG_OWN_CAR                 object 
 5    FLAG_OWN_REALTY              object 
 6    CNT_CHILDREN                 int64  
 7    AMT_INCOME_TOTAL             float64
 8    AMT_CREDIT                   float64
 9    AMT_ANNUITY                  float64
 10   AMT_GOODS_PRICE              float64
 11   NAME_TYPE_SUITE              object 
 12   NAME_INCOME_TYPE             object 
 13   NAME_EDUCATION_TYPE          object 
 14   NAME_FAMILY_STATUS           object 
 15   NAME_HOUSING_TYPE            object 
 16   REGION_POPULATION_RELATIVE   float64
 17   DAYS_BIRTH                   int64  
 18   DAYS_EMPLOYED                int64  
 19   DAYS_REGISTRATION            float64
 20   DAYS_ID_PUBLISH              int64  
 21   OWN_CAR_AGE                  float64
 22   FLAG_MOBIL                   int64  
 23   FLAG_EMP_PHONE               int64  
 24   FLAG_WORK_PHONE              int64  
 25   FLAG_CONT_MOBILE             int64  
 26   FLAG_PHONE                   int64  
 27   FLAG_EMAIL                   int64  
 28   OCCUPATION_TYPE              object 
      MEMBERS                      float64
```

```
30   REGION_RATING_CLIENT          int64
31   REGION_RATING_CLIENT_W_CITY   int64
32   WEEKDAY_APPR_PROCESS_START    object
33   HOUR_APPR_PROCESS_START       int64
34   REG_REGION_NOT_LIVE_REGION    int64
35   REG_REGION_NOT_WORK_REGION    int64
36   LIVE_REGION_NOT_WORK_REGION   int64
37   REG_CITY_NOT_LIVE_CITY        int64
38   REG_CITY_NOT_WORK_CITY        int64
39   LIVE_CITY_NOT_WORK_CITY       int64
40   ORGANIZATION_TYPE             object
41   EXT_SOURCE_1                  float64
42   EXT_SOURCE_2                  float64
43   EXT_SOURCE_3                  float64
44   APARTMENTS_AVG                float64
45   BASEMENTAREA_AVG              float64
46   YEARS_BEGINEXPLUATATION_AVG   float64
47   YEARS_BUILD_AVG               float64
48   COMMONAREA_AVG                float64
49   ELEVATORS_AVG                 float64
50   ENTRANCES_AVG                 float64
51   FLOORSMAX_AVG                 float64
52   FLOORSMIN_AVG                 float64
53   LANDAREA_AVG                  float64
54   LIVINGAPARTMENTS_AVG          float64
55   LIVINGAREA_AVG                float64
56   NONLIVINGAPARTMENTS_AVG       float64
57   NONLIVINGAREA_AVG             float64
58   APARTMENTS_MODE               float64
59   BASEMENTAREA_MODE             float64
60   YEARS_BEGINEXPLUATATION_MODE  float64
61   YEARS_BUILD_MODE              float64
62   COMMONAREA_MODE               float64
63   ELEVATORS_MODE                float64
64   ENTRANCES_MODE                float64
65   FLOORSMAX_MODE                float64
66   FLOORSMIN_MODE                float64
67   LANDAREA_MODE                 float64
68   LIVINGAPARTMENTS_MODE         float64
69   LIVINGAREA_MODE               float64
70   NONLIVINGAPARTMENTS_MODE      float64
71   NONLIVINGAREA_MODE            float64
72   APARTMENTS_MEDI               float64
73   BASEMENTAREA_MEDI             float64
74   YEARS_BEGINEXPLUATATION_MEDI  float64
75   YEARS_BUILD_MEDI              float64
76   COMMONAREA_MEDI               float64
77   ELEVATORS_MEDI                float64
78   ENTRANCES_MEDI                float64
79   FLOORSMAX_MEDI                float64
80   FLOORSMIN_MEDI                float64
81   LANDAREA_MEDI                 float64
82   LIVINGAPARTMENTS_MEDI         float64
83   LIVINGAREA_MEDI               float64
84   NONLIVINGAPARTMENTS_MEDI      float64
85   NONLIVINGAREA_MEDI            float64
86   FONDKAPREMONT_MODE            object
87   HOUSETYPE_MODE                object
88   TOTALAREA_MODE                float64
89   WALLSMATERIAL_MODE            object
90   EMERGENCYSTATE_MODE           object
91   OBS_30_CNT_SOCIAL_CIRCLE      float64
92   DEF_30_CNT_SOCIAL_CIRCLE      float64
93   OBS_60_CNT_SOCIAL_CIRCLE      float64
```

```
 94   DEF_60_CNT_SOCIAL_CIRCLE      float64
 95   DAYS_LAST_PHONE_CHANGE        float64
 96   FLAG_DOCUMENT_2               int64
 97   FLAG_DOCUMENT_3               int64
 98   FLAG_DOCUMENT_4               int64
 99   FLAG_DOCUMENT_5               int64
 100  FLAG_DOCUMENT_6               int64
 101  FLAG_DOCUMENT_7               int64
 102  FLAG_DOCUMENT_8               int64
 103  FLAG_DOCUMENT_9               int64
 104  FLAG_DOCUMENT_10              int64
 105  FLAG_DOCUMENT_11              int64
 106  FLAG_DOCUMENT_12              int64
 107  FLAG_DOCUMENT_13              int64
 108  FLAG_DOCUMENT_14              int64
 109  FLAG_DOCUMENT_15              int64
 110  FLAG_DOCUMENT_16              int64
 111  FLAG_DOCUMENT_17              int64
 112  FLAG_DOCUMENT_18              int64
 113  FLAG_DOCUMENT_19              int64
 114  FLAG_DOCUMENT_20              int64
 115  FLAG_DOCUMENT_21              int64
 116  AMT_REQ_CREDIT_BUREAU_HOUR    float64
 117  AMT_REQ_CREDIT_BUREAU_DAY     float64
 118  AMT_REQ_CREDIT_BUREAU_WEEK    float64
 119  AMT_REQ_CREDIT_BUREAU_MON     float64
 120  AMT_REQ_CREDIT_BUREAU_QRT     float64
 121  AMT_REQ_CREDIT_BUREAU_YEAR    float64
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

In [19]:
```
df.shape
```

Out[19]:
```
(307511, 122)
```

In [129…
```
df.describe()
```

Out[129…

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	

8 rows × 106 columns

In [130…
```
df.isnull().sum()*100/len(df)
```

Out[130…
```
SK_ID_CURR                    0.000000
TARGET                        0.000000
NAME_CONTRACT_TYPE            0.000000
CODE_GENDER                   0.000000
```

```
FLAG_OWN_CAR                  0.000000
                                ...
AMT_REQ_CREDIT_BUREAU_DAY    13.501631
AMT_REQ_CREDIT_BUREAU_WEEK   13.501631
AMT_REQ_CREDIT_BUREAU_MON    13.501631
AMT_REQ_CREDIT_BUREAU_QRT    13.501631
AMT_REQ_CREDIT_BUREAU_YEAR   13.501631
Length: 122, dtype: float64
```

In [20]:
```
df["SK_ID_CURR"].value_counts()
```

Out[20]:
```
100002    1
337664    1
337661    1
337660    1
337659    1
         ..
218992    1
218991    1
218990    1
218989    1
456255    1
Name: SK_ID_CURR, Length: 307511, dtype: int64
```

In [ ]:
```

```

In [25]:
```
df1['DAYS_BIRTH']=df1['DAYS_BIRTH'].abs()      ####Converting neegative into positive valu
df1['DAYS_EMPLOYED']=df1['DAYS_EMPLOYED'].abs()
df1['DAYS_ID_PUBLISH']=df1['DAYS_ID_PUBLISH'].abs()
df1.head()
```

Out[25]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT
0	100002	1	Cash loans	M	N	Y	
1	100003	0	Cash loans	F	N	N	
2	100004	0	Revolving loans	M	Y	Y	
3	100006	0	Cash loans	F	N	Y	
4	100007	0	Cash loans	M	N	Y	

5 rows × 90 columns

Checking for missing value

Making threshold as 30

In [128…
```
missing=df.isnull().sum()*100/len(df)
missing[missing>30]
```

Out[128…
```
OWN_CAR_AGE                   65.990810
OCCUPATION_TYPE               31.345545
EXT_SOURCE_1                  56.381073
APARTMENTS_AVG                50.749729
BASEMENTAREA_AVG              58.515956
YEARS_BEGINEXPLUATATION_AVG   48.781019
YEARS_BUILD_AVG               66.497784
COMMONAREA_AVG                69.872297
ELEVATORS_AVG                 53.295980
```

```
ENTRANCES_AVG                  50.348768
FLOORSMAX_AVG                  49.760822
FLOORSMIN_AVG                  67.848630
LANDAREA_AVG                   59.376738
LIVINGAPARTMENTS_AVG           68.354953
LIVINGAREA_AVG                 50.193326
NONLIVINGAPARTMENTS_AVG        69.432963
NONLIVINGAREA_AVG              55.179164
APARTMENTS_MODE                50.749729
BASEMENTAREA_MODE              58.515956
YEARS_BEGINEXPLUATATION_MODE   48.781019
YEARS_BUILD_MODE               66.497784
COMMONAREA_MODE                69.872297
ELEVATORS_MODE                 53.295980
ENTRANCES_MODE                 50.348768
FLOORSMAX_MODE                 49.760822
FLOORSMIN_MODE                 67.848630
LANDAREA_MODE                  59.376738
LIVINGAPARTMENTS_MODE          68.354953
LIVINGAREA_MODE                50.193326
NONLIVINGAPARTMENTS_MODE       69.432963
NONLIVINGAREA_MODE             55.179164
APARTMENTS_MEDI                50.749729
BASEMENTAREA_MEDI              58.515956
YEARS_BEGINEXPLUATATION_MEDI   48.781019
YEARS_BUILD_MEDI               66.497784
COMMONAREA_MEDI                69.872297
ELEVATORS_MEDI                 53.295980
ENTRANCES_MEDI                 50.348768
FLOORSMAX_MEDI                 49.760822
FLOORSMIN_MEDI                 67.848630
LANDAREA_MEDI                  59.376738
LIVINGAPARTMENTS_MEDI          68.354953
LIVINGAREA_MEDI                50.193326
NONLIVINGAPARTMENTS_MEDI       69.432963
NONLIVINGAREA_MEDI             55.179164
FONDKAPREMONT_MODE             68.386172
HOUSETYPE_MODE                 50.176091
TOTALAREA_MODE                 48.268517
WALLSMATERIAL_MODE             50.840783
EMERGENCYSTATE_MODE            47.398304
dtype: float64
```

Droping the missing value

In [125…
```
df1=df.drop(["OWN_CAR_AGE","OCCUPATION_TYPE","EXT_SOURCE_1","APARTMENTS_AVG","BASEMENTAREA
```

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

In [ ]:
```
df.isnull().sum()*100/len(df)
```

In [26]:
```
dropped_columns=["OWN_CAR_AGE","OCCUPATION_TYPE","EXT_SOURCE_1","APARTMENTS_AVG","BASEMENT
```

In [ ]:
```

```

In [ ]:
```

```

In [27]:
```
len(dropped_columns)
```

Out[27]:
```
32
```

In [ ]:
```
##### Making threshold as 30
```

In [127…
```
missing=df.isnull().sum()*100/len(df)
missing[missing>30]
```

Out[127…
```
OWN_CAR_AGE                   65.990810
OCCUPATION_TYPE               31.345545
EXT_SOURCE_1                  56.381073
APARTMENTS_AVG                50.749729
BASEMENTAREA_AVG              58.515956
YEARS_BEGINEXPLUATATION_AVG   48.781019
YEARS_BUILD_AVG               66.497784
COMMONAREA_AVG                69.872297
ELEVATORS_AVG                 53.295980
ENTRANCES_AVG                 50.348768
FLOORSMAX_AVG                 49.760822
FLOORSMIN_AVG                 67.848630
LANDAREA_AVG                  59.376738
LIVINGAPARTMENTS_AVG          68.354953
LIVINGAREA_AVG                50.193326
NONLIVINGAPARTMENTS_AVG       69.432963
NONLIVINGAREA_AVG             55.179164
APARTMENTS_MODE               50.749729
BASEMENTAREA_MODE             58.515956
YEARS_BEGINEXPLUATATION_MODE  48.781019
YEARS_BUILD_MODE              66.497784
COMMONAREA_MODE               69.872297
ELEVATORS_MODE                53.295980
ENTRANCES_MODE                50.348768
FLOORSMAX_MODE                49.760822
FLOORSMIN_MODE                67.848630
LANDAREA_MODE                 59.376738
LIVINGAPARTMENTS_MODE         68.354953
LIVINGAREA_MODE               50.193326
NONLIVINGAPARTMENTS_MODE      69.432963
NONLIVINGAREA_MODE            55.179164
APARTMENTS_MEDI               50.749729
BASEMENTAREA_MEDI             58.515956
YEARS_BEGINEXPLUATATION_MEDI  48.781019
YEARS_BUILD_MEDI              66.497784
COMMONAREA_MEDI               69.872297
ELEVATORS_MEDI                53.295980
ENTRANCES_MEDI                50.348768
FLOORSMAX_MEDI                49.760822
FLOORSMIN_MEDI                67.848630
LANDAREA_MEDI                 59.376738
LIVINGAPARTMENTS_MEDI         68.354953
LIVINGAREA_MEDI               50.193326
NONLIVINGAPARTMENTS_MEDI      69.432963
NONLIVINGAREA_MEDI            55.179164
FONDKAPREMONT_MODE            68.386172
HOUSETYPE_MODE                50.176091
TOTALAREA_MODE                48.268517
WALLSMATERIAL_MODE            50.840783
EMERGENCYSTATE_MODE           47.398304
dtype: float64
```

```
df1.shape
```

Out[132…
```
(307511, 90)
```

In [133…
```
df1.describe()
```

Out[133…

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	

8 rows × 79 columns

Checking for imbalance ratio

In [134…
```
df1['TARGET'].value_counts(normalize=True)*100
```

Out[134…
```
0    91.927118
1     8.072882
Name: TARGET, dtype: float64
```

Splitting the dataframe into two segment.

In [28]:
```
df0=df[df1.TARGET==0]
df2=df[df1.TARGET==1]
```

In [135…
```
df1.shape
```

Out[135…
```
(307511, 90)
```

In [23]:
```
df1=df.drop(["OWN_CAR_AGE","OCCUPATION_TYPE","EXT_SOURCE_1","APARTMENTS_AVG","BASEMENTAREA
```

In [ ]:
```

```

Taking 5 features to analyse

In [140…
```
df3=df1[['NAME_CONTRACT_TYPE','AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY','NAME_INCOME_T
df3.head()
```

Out[140…

	NAME_CONTRACT_TYPE	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	NAME_INCOME_TYPE	TARGET
0	Cash loans	202500.0	406597.5	24700.5	Working	1
1	Cash loans	270000.0	1293502.5	35698.5	State servant	0
2	Revolving loans	67500.0	135000.0	6750.0	Working	0

	NAME_CONTRACT_TYPE	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	NAME_INCOME_TYPE	TARGET
3	Cash loans	135000.0	312682.5	29686.5	Working	0
4	Cash loans	121500.0	513000.0	21865.5	Working	0

Univariate analysis

In [138…
```
####Identifying categorical and numerical variables
number= df1.select_dtypes(include="number").columns.to_list()
category =df1.select_dtypes(exclude="number").columns.to_list()
```

In [ ]:
```

```

In [ ]:
```

```

In [141…
```
[i for i in df3.columns  if i in category]
```

Out[141…
```
['NAME_CONTRACT_TYPE', 'NAME_INCOME_TYPE']
```

In [142…
```
df3[[i for i in df3.columns  if i in category]]
```

Out[142…

	NAME_CONTRACT_TYPE	NAME_INCOME_TYPE
0	Cash loans	Working
1	Cash loans	State servant
2	Revolving loans	Working
3	Cash loans	Working
4	Cash loans	Working
...	...	...
307506	Cash loans	Working
307507	Cash loans	Pensioner
307508	Cash loans	Working
307509	Cash loans	Commercial associate
307510	Cash loans	Commercial associate

307511 rows × 2 columns

In [143…
```
[i for i in df3.columns  if i in number]
```

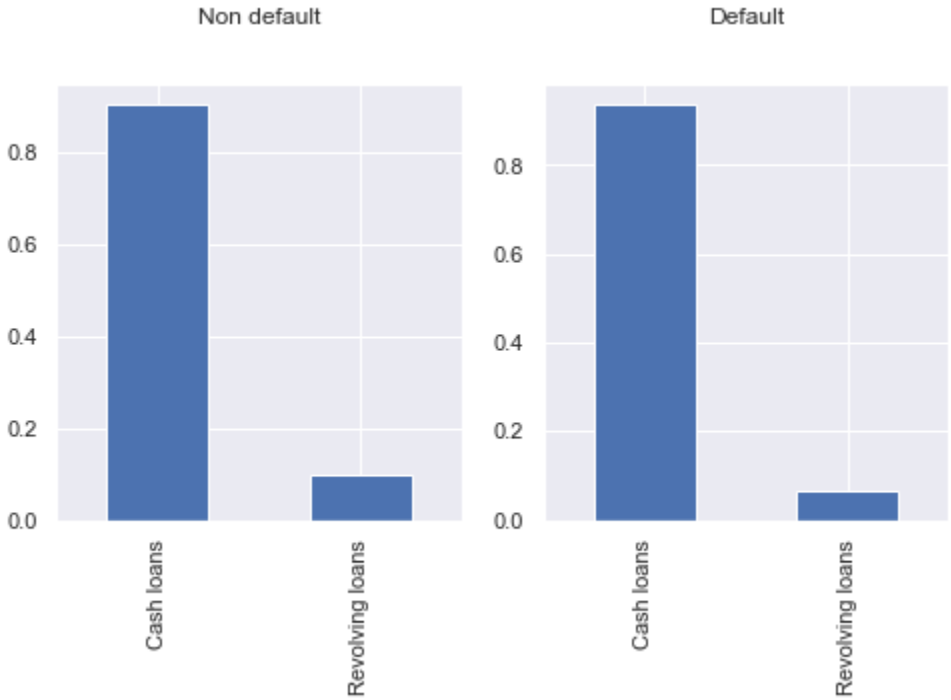
Out[143…
```
['AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'TARGET']
```

In [ ]:
```

```

NAME_CONTRACT_TYPE

In [144…
```
plt.figure(figsize=(8,4))
21)
```

```
df0.NAME_CONTRACT_TYPE.value_counts(normalize=True).plot.bar()
plt.title('Non default')
plt.subplot(122)
df2.NAME_CONTRACT_TYPE.value_counts(normalize=True).plot.bar()
plt.title('Default')
plt.show()
```



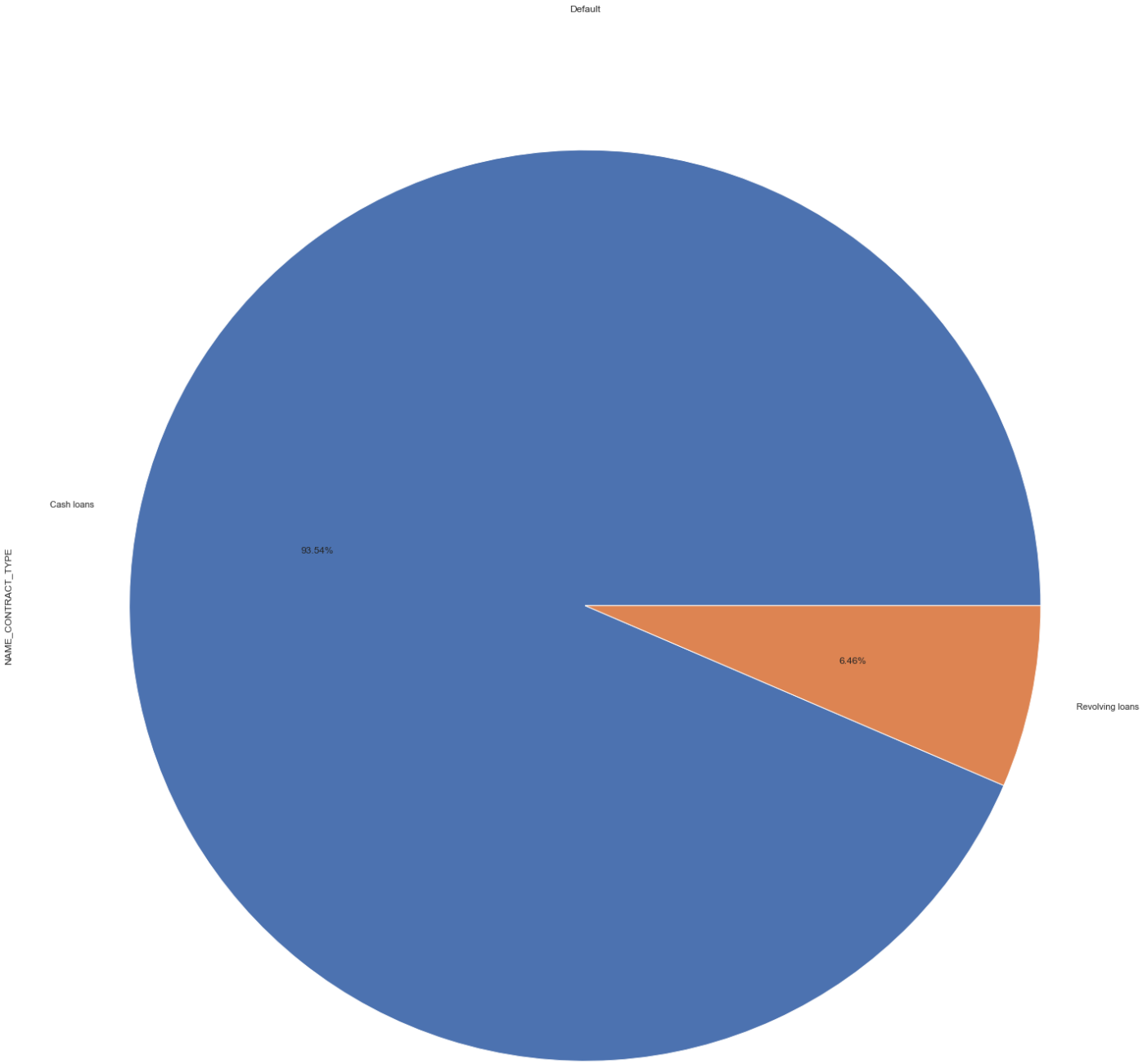
From the graph we can say for defaulters and non defaulters cash loans is more than revolving loans.

And non defaulters use more revolving loans than defaulters

NAME_CONTRACT_TYPE

In [146…

```
df0.NAME_CONTRACT_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Non default')
plt.show()
df2.NAME_CONTRACT_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Default')
plt.show()
```



We can say that cash loan is more for both defaulters and non defaulters than revolving loans.

Non defaulters

Cash loan=90.2%

Revilving=9.79%

Defaulters

Cash loan=93.54%

Revilving=6.46%

AMT_INCOME_TOTAL

Creating bin for AMT_INCOME_TOTAL

In [148…

```
bins = [0,25000,50000,75000,100000,125000,150000,175000,200000,225000,250000,275000,
        0,325000,350000,375000,400000,425000,450000,475000,500000,10000000000]
```

```
slot=['0-25000','25000-50000','50000-75000','75000-100000','100000-125000',
       '125000-150000','150000-175000','175000-200000','200000-225000','225000-250000',
      '250000-275000','275000-300000','300000-325000','325000-350000','350000-375000',
      '375000-400000','400000-425000','425000-450000','450000-475000','475000-500000', '50
df1['AMT_INCOME_RANGE']=pd.cut(df1['AMT_INCOME_TOTAL'],bins=bins,labels=slot)
```

In [149…

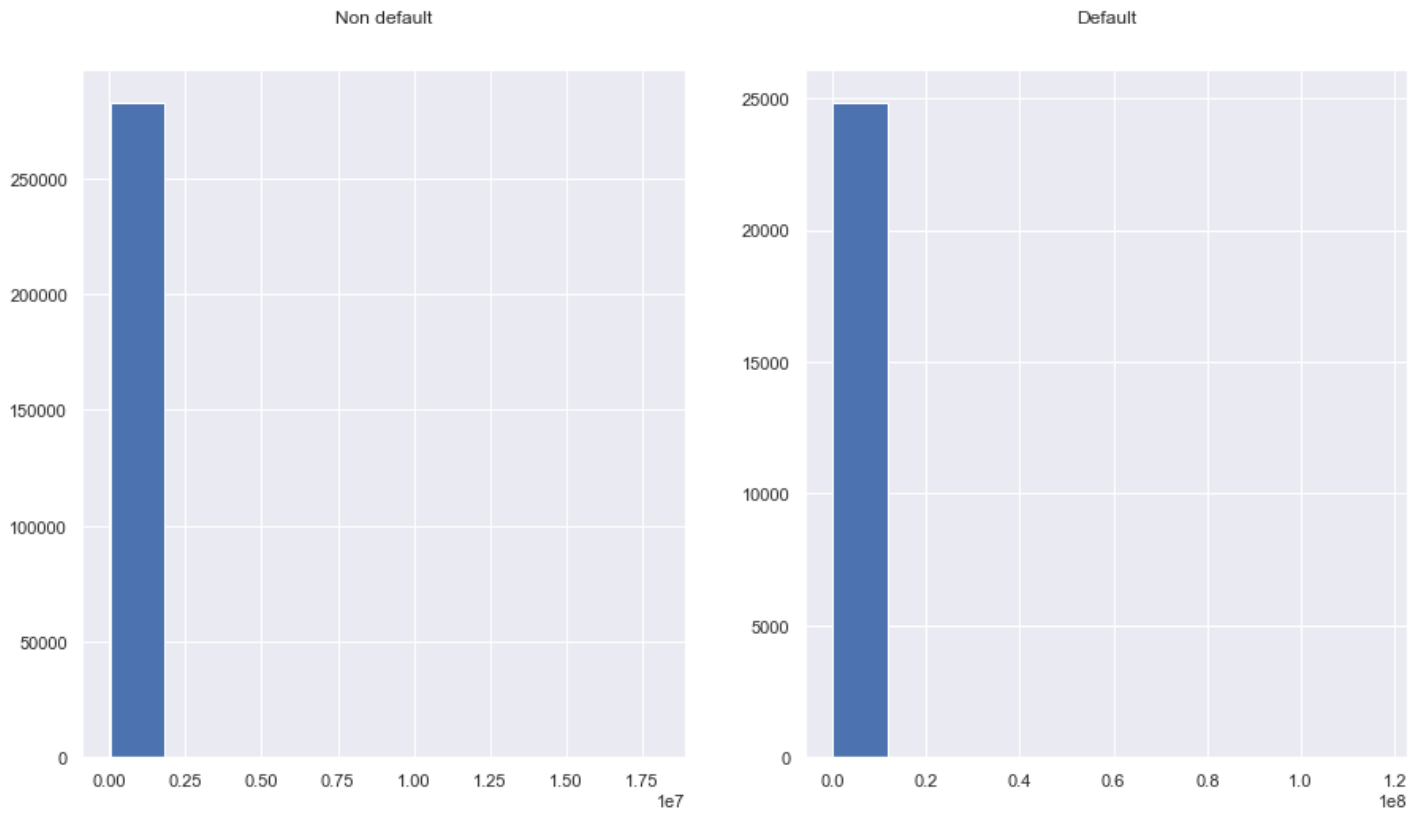
```
df1['AMT_INCOME_RANGE'].head()
```

Out[149…

```
0    200000-225000
1    250000-275000
2      50000-75000
3    125000-150000
4    100000-125000
Name: AMT_INCOME_RANGE, dtype: category
Categories (21, object): ['0-25000' < '25000-50000' < '50000-75000' < '75000-100000' ...
'425000-450000' < '450000-475000' < '475000-500000' < '500000 and above']
```

In [150…

```
plt.figure(figsize=(15,8))
plt.subplot(121)
plt.hist(x='AMT_INCOME_TOTAL',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='AMT_INCOME_TOTAL',data=df2)
plt.title("Default")
plt.show()
```

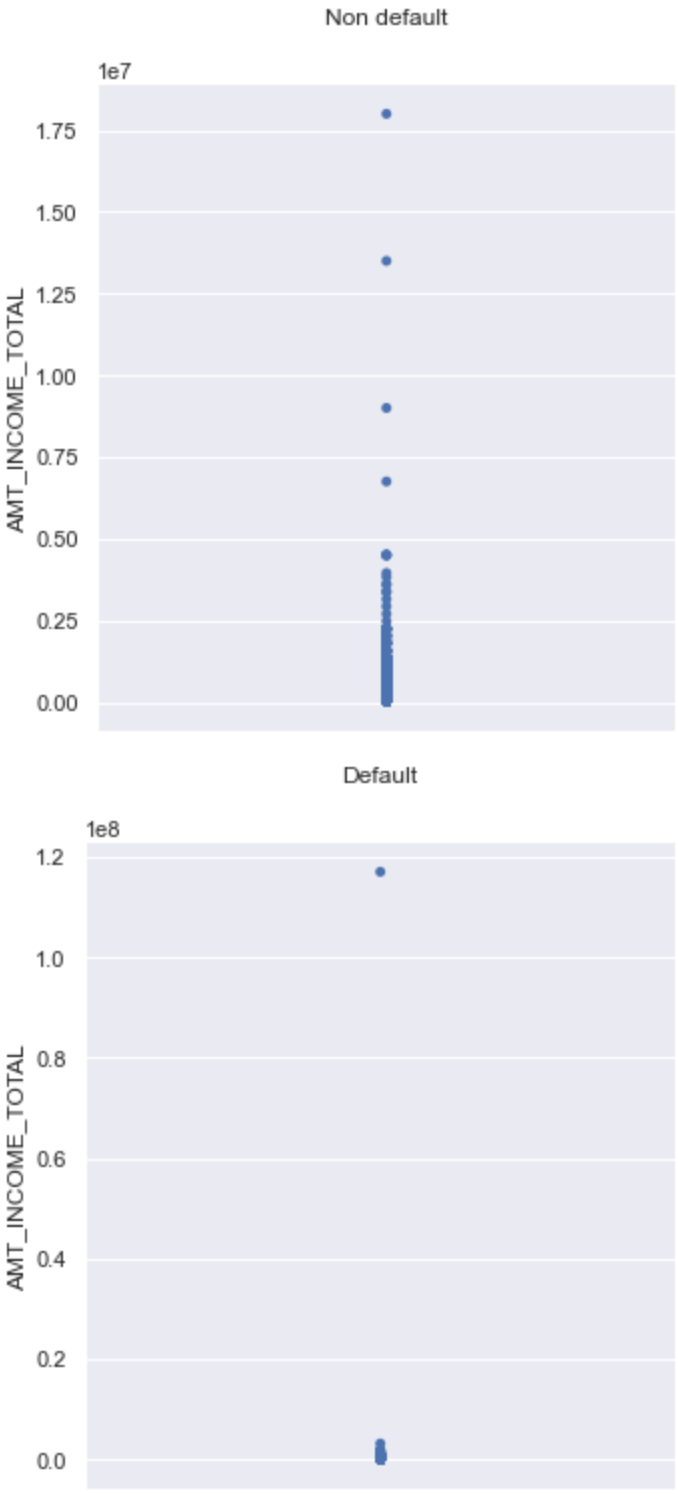


From the above graph we can infer that non defaulters are in the highest income group when compared to defaulters.

In [151…

```
plt.figure(figsize=(15,8))
sns.catplot(y="AMT_INCOME_TOTAL",jitter=False,data=df0)
plt.title('Non default')
plt.show()
sns.catplot(y="AMT_INCOME_TOTAL",jitter=False,data=df2)
plt.title('Default')
```

```
<Figure size 1080x576 with 0 Axes>
```

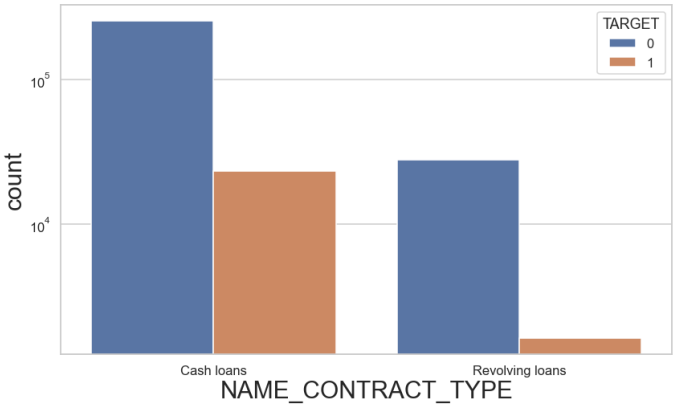


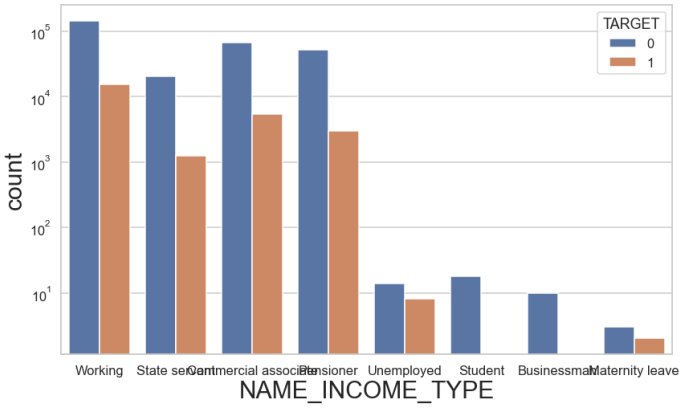
Non defaulters are having high income when compared to defaulters, they are having low income.

Categorical variables

In [244…

```
cat_columns=['NAME_CONTRACT_TYPE', 'NAME_INCOME_TYPE']

plt.figure(figsize=(30,8))
for i in (enumerate(cat_columns)):
    plt.subplot(len(cat_columns)//2,2,i[0]+1)
    sns.countplot(x=i[1],hue='TARGET',data=df1)
    plt.yscale('log')

plt.show()
```





'NAME_INCOME_TYPE'--

From graph for non defaulters working group is higher than in defaulters

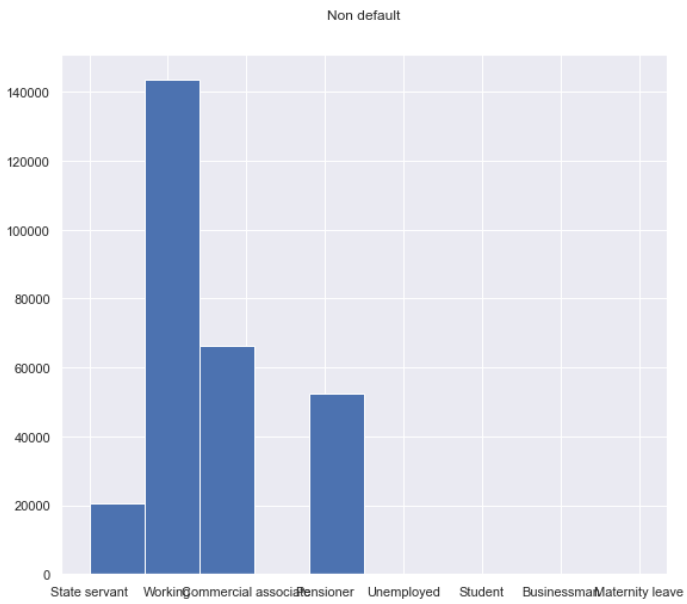
In most of the income type non defaulters are in high number for eg state servant,Pensioner etc

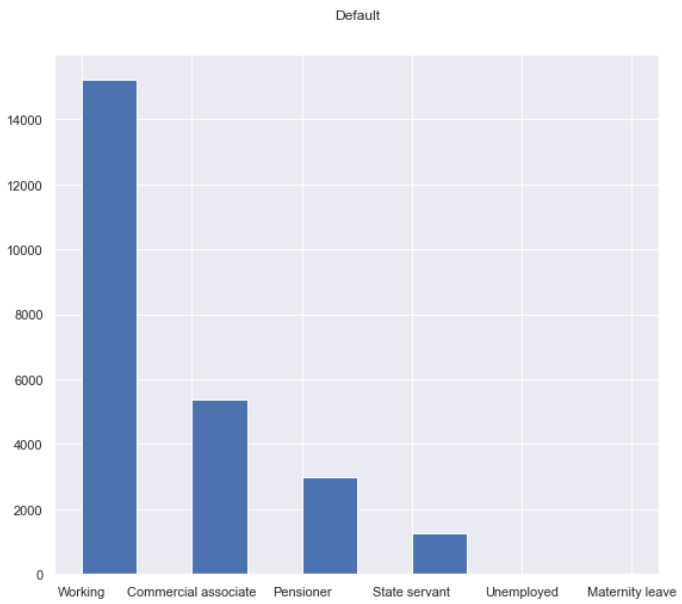
For businessman and student defaulters are very less when compared to non defaulters.

NAME_INCOME_TYPE

In [158…

```
plt.figure(figsize=(20,8))
plt.subplot(121)
plt.hist(x='NAME_INCOME_TYPE',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='NAME_INCOME_TYPE',data=df2)
plt.title("Default")
plt.show()
```



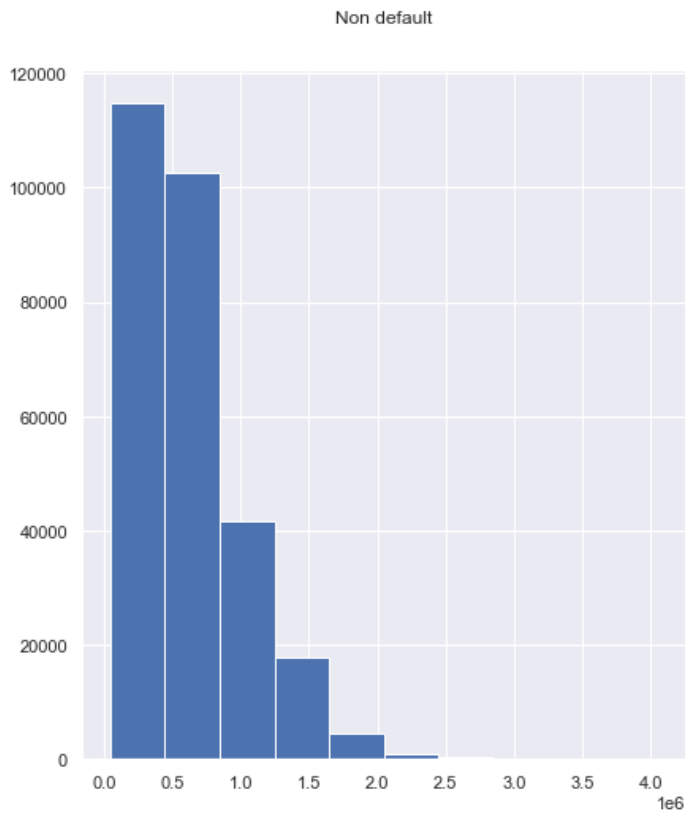


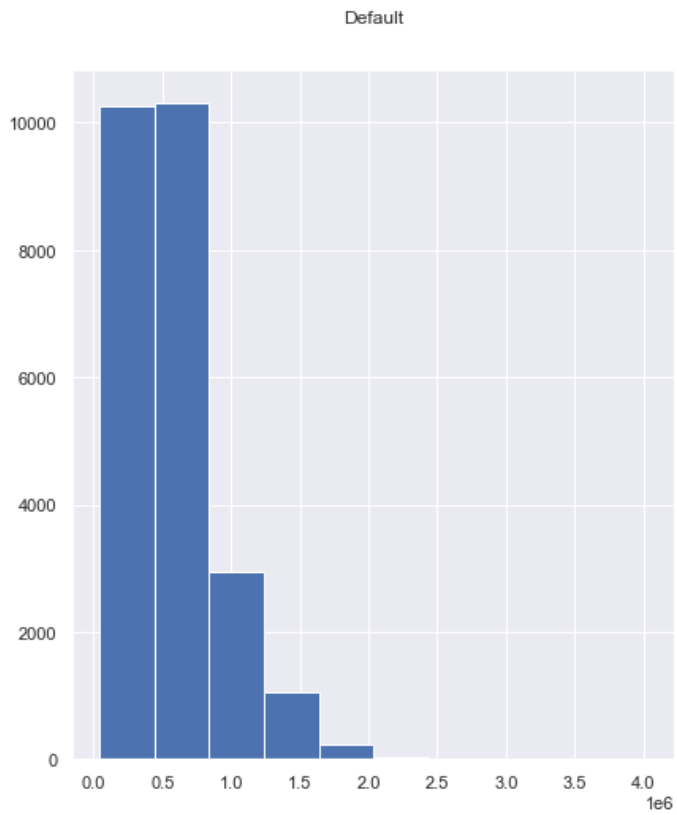
Highest non defaulters are working group and also same for defaulters.

AMT_CREDIT

In [161…

```
plt.figure(figsize=(15,8))
plt.subplot(121)
plt.hist(x='AMT_CREDIT',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='AMT_CREDIT',data=df2)
plt.title("Default")
plt.show()
```

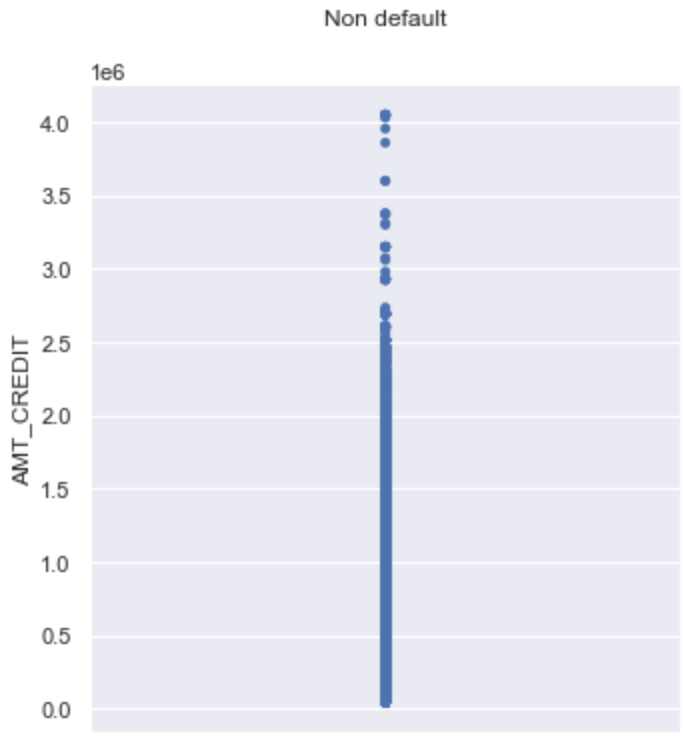


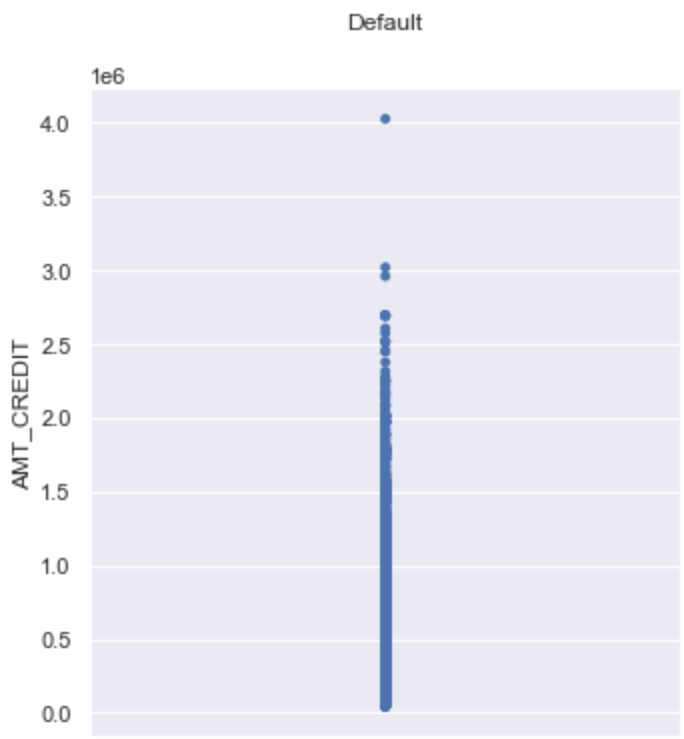


In [162…

```
plt.figure(figsize=(15,8))
sns.catplot(y="AMT_CREDIT",jitter=False,data=df0)
plt.title('Non default')
plt.show()
sns.catplot(y="AMT_CREDIT",jitter=False,data=df2)
plt.title('Default')
plt.show()
```

```
<Figure size 1080x576 with 0 Axes>
```



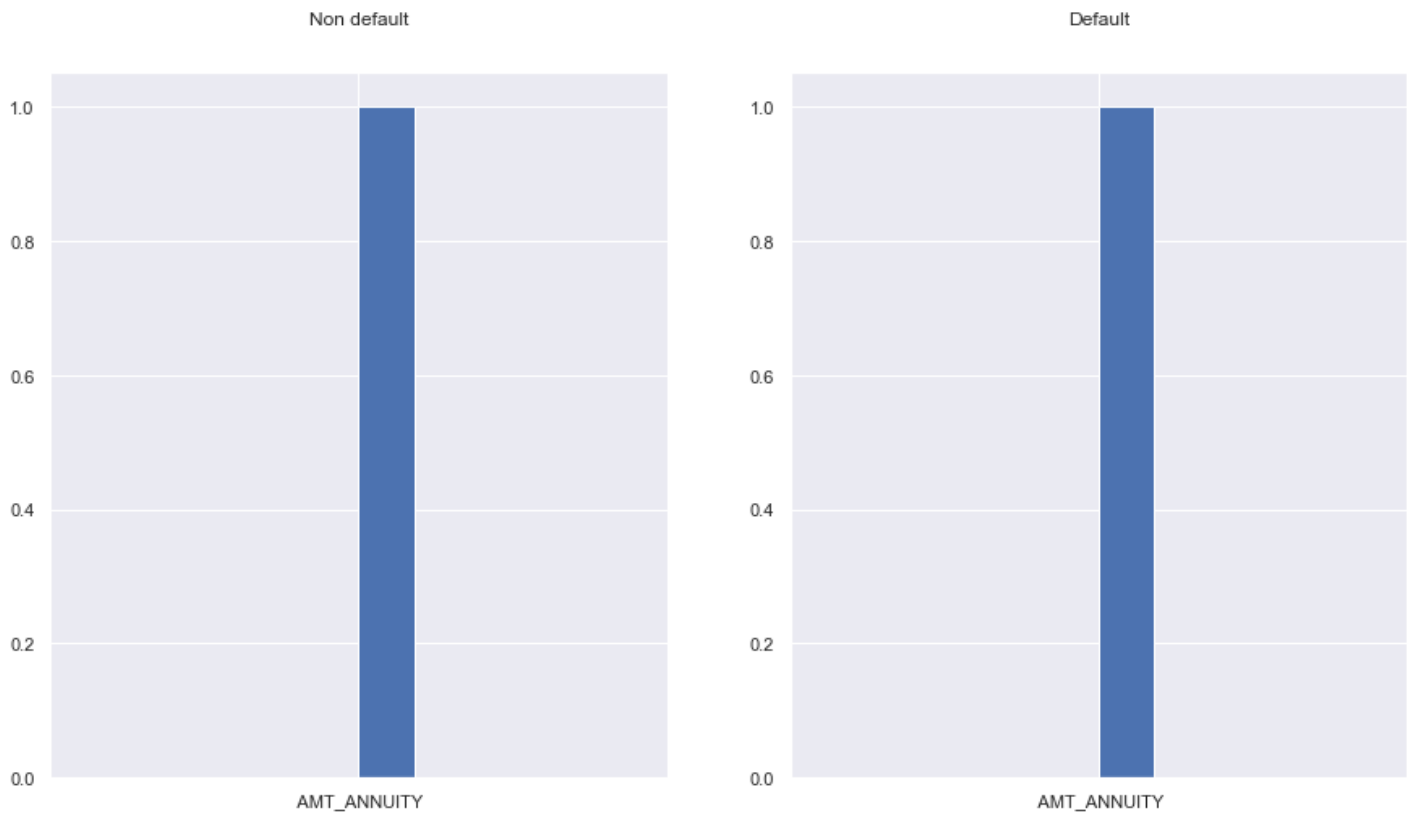


Amount of loan credit given to non defaulters is high when compared to defaulters.

AMT_ANNUITY

In [163…

```
plt.figure(figsize=(15,8))
plt.subplot(121)
plt.hist(x=' AMT_ANNUITY',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x=' AMT_ANNUITY',data=df2)
plt.title("Default")
plt.show()
```

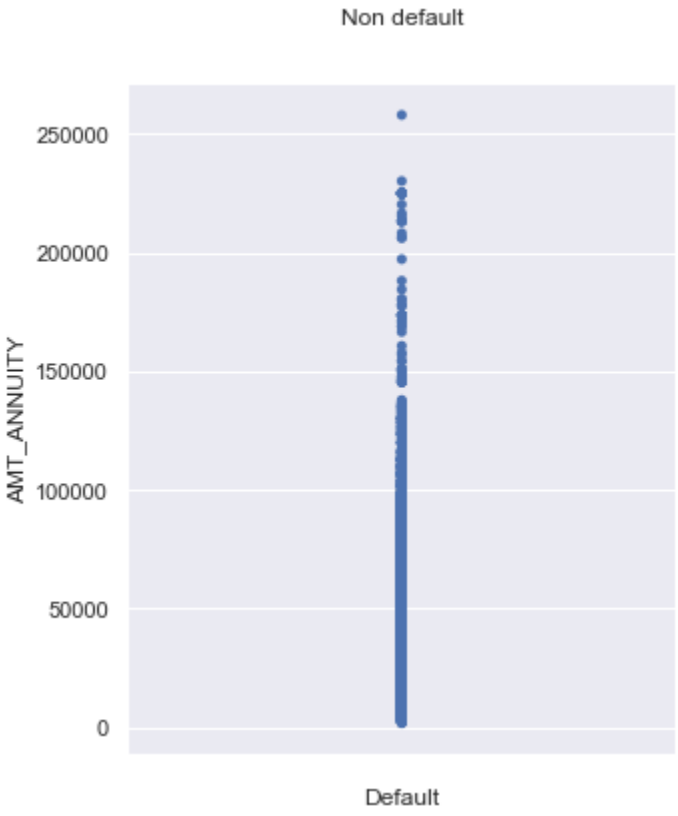


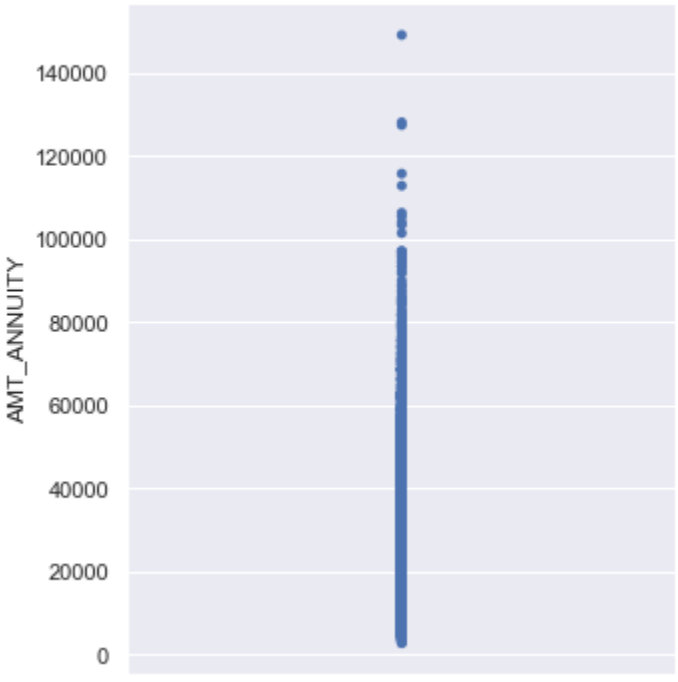
In [ ]:

In [164…

```
plt.figure(figsize=(15,8))
sns.catplot(y="AMT_ANNUITY",jitter=False,data=df0)
plt.title('Non default')
plt.show()
sns.catplot(y="AMT_ANNUITY",jitter=False,data=df2)
plt.title('Default')
plt.show()
```

```
<Figure size 1080x576 with 0 Axes>
```

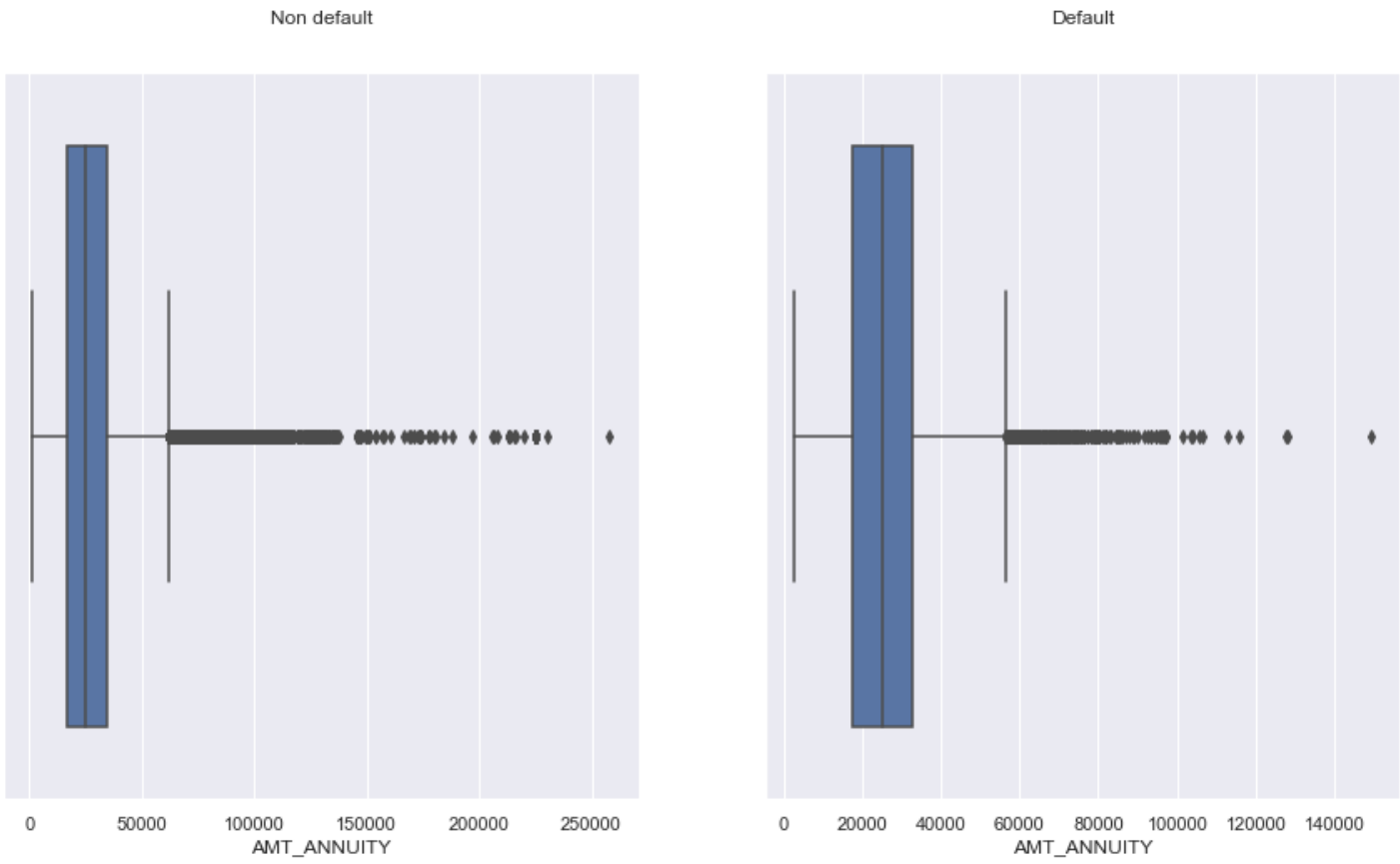




Loan annuity is more for non defaulters when compared to defaulters

In [ ]:

In [165…
```
plt.figure(figsize=(15,8))
plt.subplot(121)
sns.boxplot(x='AMT_ANNUITY',data=df0)
plt.title('Non default')
plt.subplot(122)
sns.boxplot(x='AMT_ANNUITY',data=df2)
plt.title("Default")
plt.show()
```



Maximum value of loan annuity for non defaulters is more when compared to that of defaulters.

Taking next 5 features from dataframe

In [166…
```
df1.columns.to_list()
```

Out[166…
```
['SK_ID_CURR',
 'TARGET',
 'NAME_CONTRACT_TYPE',
 'CODE_GENDER',
 'FLAG_OWN_CAR',
 'FLAG_OWN_REALTY',
 'CNT_CHILDREN',
 'AMT_INCOME_TOTAL',
 'AMT_CREDIT',
 'AMT_ANNUITY',
 'AMT_GOODS_PRICE',
 'NAME_TYPE_SUITE',
 'NAME_INCOME_TYPE',
 'NAME_EDUCATION_TYPE',
 'NAME_FAMILY_STATUS',
 '_TYPE',
```

```
'REGION_POPULATION_RELATIVE',
'DAYS_BIRTH',
'DAYS_EMPLOYED',
'DAYS_REGISTRATION',
'DAYS_ID_PUBLISH',
'FLAG_MOBIL',
'FLAG_EMP_PHONE',
'FLAG_WORK_PHONE',
'FLAG_CONT_MOBILE',
'FLAG_PHONE',
'FLAG_EMAIL',
'CNT_FAM_MEMBERS',
'REGION_RATING_CLIENT',
'REGION_RATING_CLIENT_W_CITY',
'WEEKDAY_APPR_PROCESS_START',
'HOUR_APPR_PROCESS_START',
'REG_REGION_NOT_LIVE_REGION',
'REG_REGION_NOT_WORK_REGION',
'LIVE_REGION_NOT_WORK_REGION',
'REG_CITY_NOT_LIVE_CITY',
'REG_CITY_NOT_WORK_CITY',
'LIVE_CITY_NOT_WORK_CITY',
'ORGANIZATION_TYPE',
'EXT_SOURCE_2',
'EXT_SOURCE_3',
'LIVINGAREA_AVG',
'YEARS_BEGINEXPLUATATION_MODE',
'YEARS_BUILD_MODE',
'COMMONAREA_MODE',
'ELEVATORS_MODE',
'ENTRANCES_MODE',
'FLOORSMAX_MODE',
'FLOORSMIN_MODE',
'LANDAREA_MODE',
'LIVINGAPARTMENTS_MODE',
'LIVINGAREA_MODE',
'NONLIVINGAPARTMENTS_MODE',
'NONLIVINGAREA_MODE',
'APARTMENTS_MEDI',
'BASEMENTAREA_MEDI',
'FLOORSMIN_MEDI',
'LIVINGAPARTMENTS_MEDI',
'LIVINGAREA_MEDI',
'OBS_30_CNT_SOCIAL_CIRCLE',
'DEF_30_CNT_SOCIAL_CIRCLE',
'OBS_60_CNT_SOCIAL_CIRCLE',
'DEF_60_CNT_SOCIAL_CIRCLE',
'DAYS_LAST_PHONE_CHANGE',
'FLAG_DOCUMENT_2',
'FLAG_DOCUMENT_3',
'FLAG_DOCUMENT_4',
'FLAG_DOCUMENT_5',
'FLAG_DOCUMENT_6',
'FLAG_DOCUMENT_7',
'FLAG_DOCUMENT_8',
'FLAG_DOCUMENT_9',
'FLAG_DOCUMENT_10',
'FLAG_DOCUMENT_11',
'FLAG_DOCUMENT_12',
'FLAG_DOCUMENT_13',
'FLAG_DOCUMENT_14',
'FLAG_DOCUMENT_15',
'FLAG_DOCUMENT_16',
'FLAG_DOCUMENT_17',
```

```
'FLAG_DOCUMENT_18',
'FLAG_DOCUMENT_19',
'FLAG_DOCUMENT_20',
'FLAG_DOCUMENT_21',
'AMT_REQ_CREDIT_BUREAU_HOUR',
'AMT_REQ_CREDIT_BUREAU_DAY',
'AMT_REQ_CREDIT_BUREAU_WEEK',
'AMT_REQ_CREDIT_BUREAU_MON',
'AMT_REQ_CREDIT_BUREAU_QRT',
'AMT_REQ_CREDIT_BUREAU_YEAR',
'AMT_INCOME_RANGE']
```

In [167…

```
df4=df1[['AMT_GOODS_PRICE','NAME_TYPE_SUITE','NAME_EDUCATION_TYPE','NAME_FAMILY_STATUS','N
df4.head()
```

Out[167…

	AMT_GOODS_PRICE	NAME_TYPE_SUITE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_T
0	351000.0	Unaccompanied	Secondary / secondary special	Single / not married	House / apart
1	1129500.0	Family	Higher education	Married	House / apart
2	135000.0	Unaccompanied	Secondary / secondary special	Single / not married	House / apart
3	297000.0	Unaccompanied	Secondary / secondary special	Civil marriage	House / apart
4	513000.0	Unaccompanied	Secondary / secondary special	Single / not married	House / apart

In [168…

```
[i for i in df4.columns  if i in number]
```

Out[168…

```
['AMT_GOODS_PRICE', 'TARGET']
```

In [169…

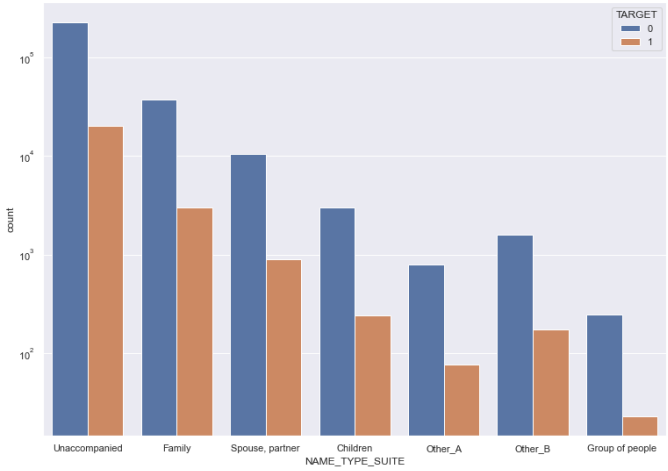
```
[i for i in df4.columns  if i in category]
```

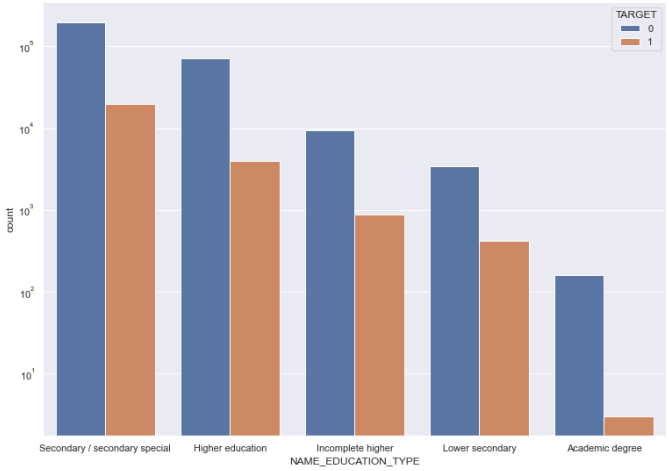
Out[169…

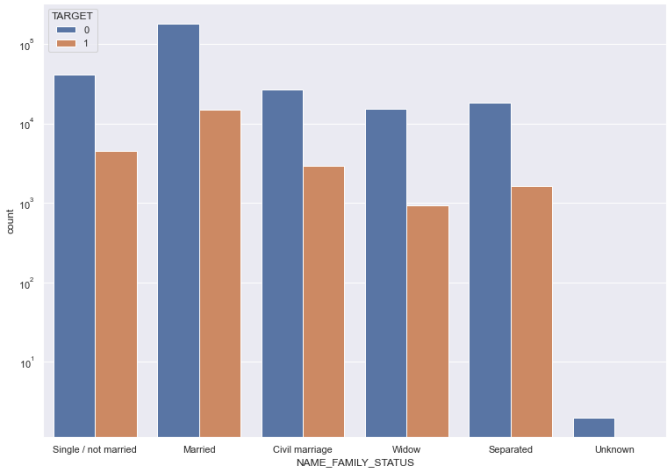
```
['NAME_TYPE_SUITE',
 'NAME_EDUCATION_TYPE',
 'NAME_FAMILY_STATUS',
 'NAME_HOUSING_TYPE']
```

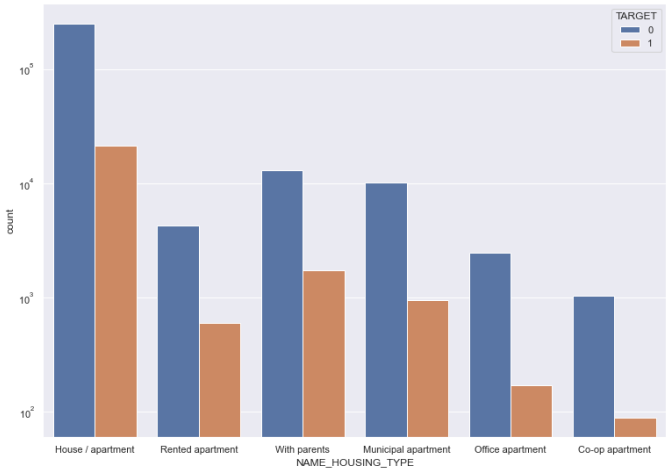
In [170…

```
cat_columns1=['NAME_TYPE_SUITE','NAME_EDUCATION_TYPE','NAME_FAMILY_STATUS','NAME_HOUSING_T

plt.figure(figsize=(28,20))
for i in (enumerate(cat_columns1)):
    plt.subplot(len(cat_columns1)//2,2,i[0]+1)
    sns.countplot(x=i[1],hue='TARGET',data=df1)
    plt.yscale('log')

plt.show()
```









In [ ]:

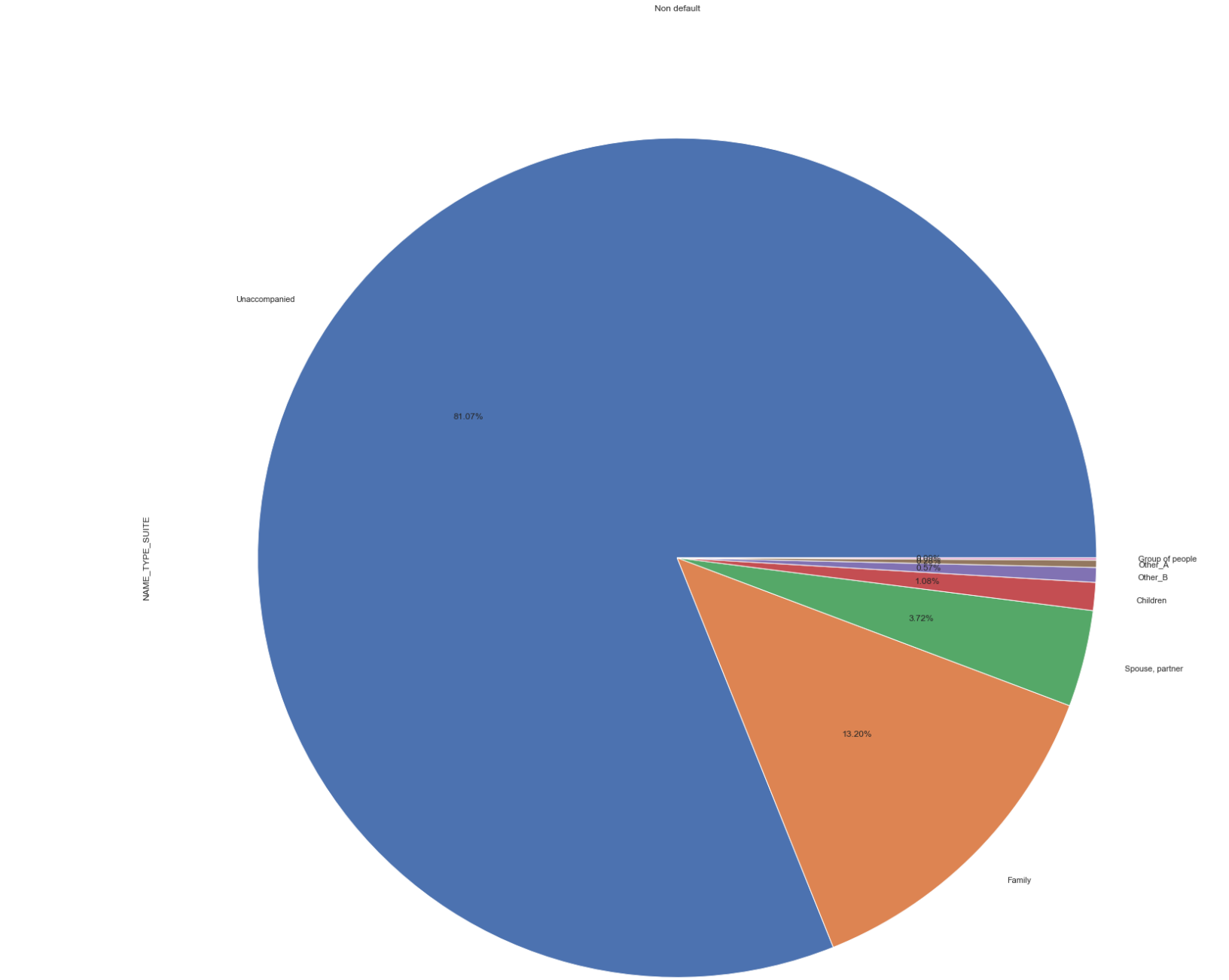
```
##### NAME_TYPE_SUITE - For non defaulters and defaulters client is unaccompanied in most
##### NAME_EDUCATION_TYPE-For non defaulters and defaulters most of the clients are having
#####                                                    For defaulters academic degree holders
##### NAME_FAMILY_STATUS-Highest non defaulters and defaulters are in the married category
##### NAME_HOUSING_TYPE-Highest non defaulters and defaulters are having house/apartment.
```

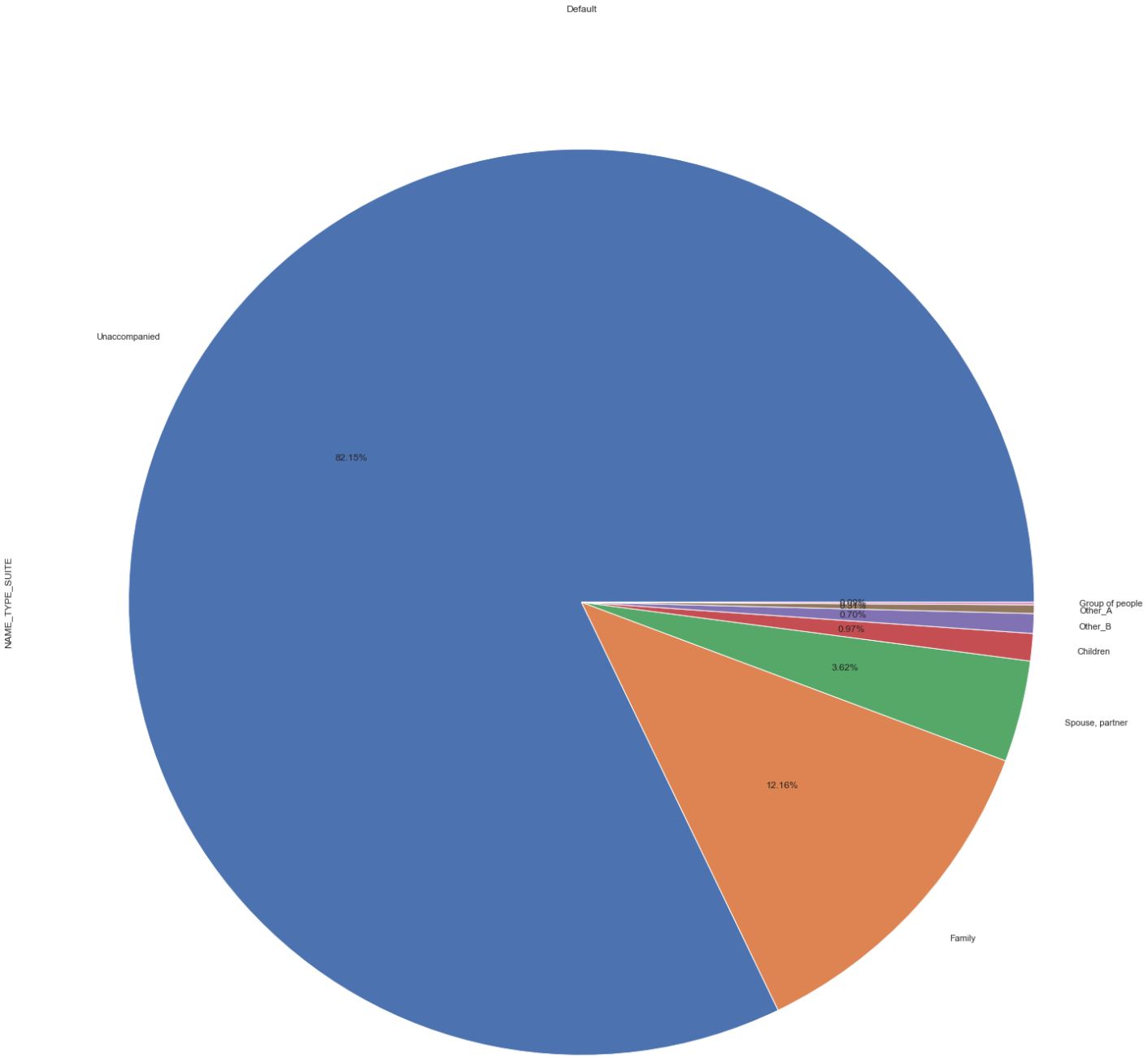
NAME_TYPE_SUITE

In [ ]:

In [171…

```
df0.NAME_TYPE_SUITE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Non default')
plt.show()
df2.NAME_TYPE_SUITE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Default')
plt.show()
```

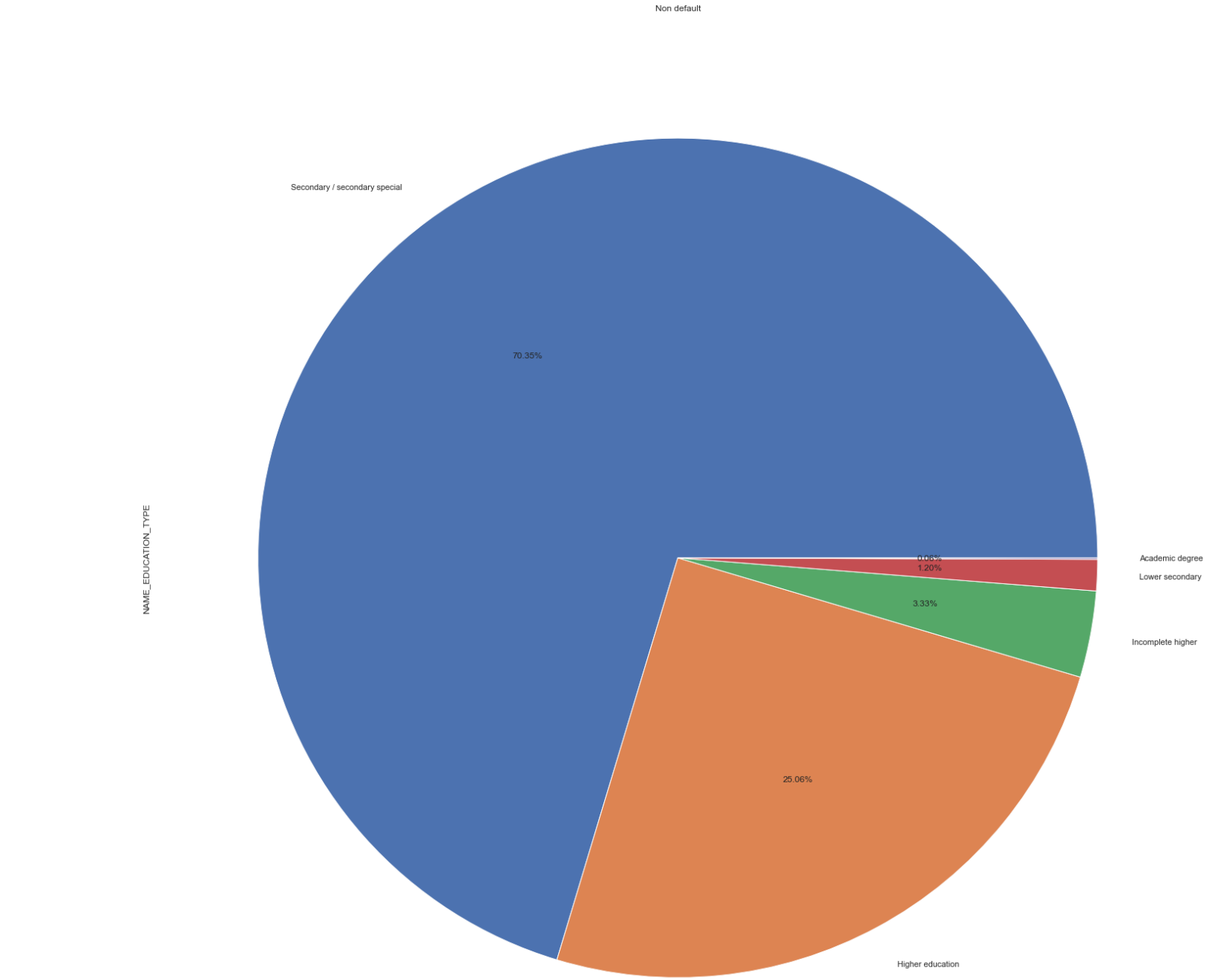


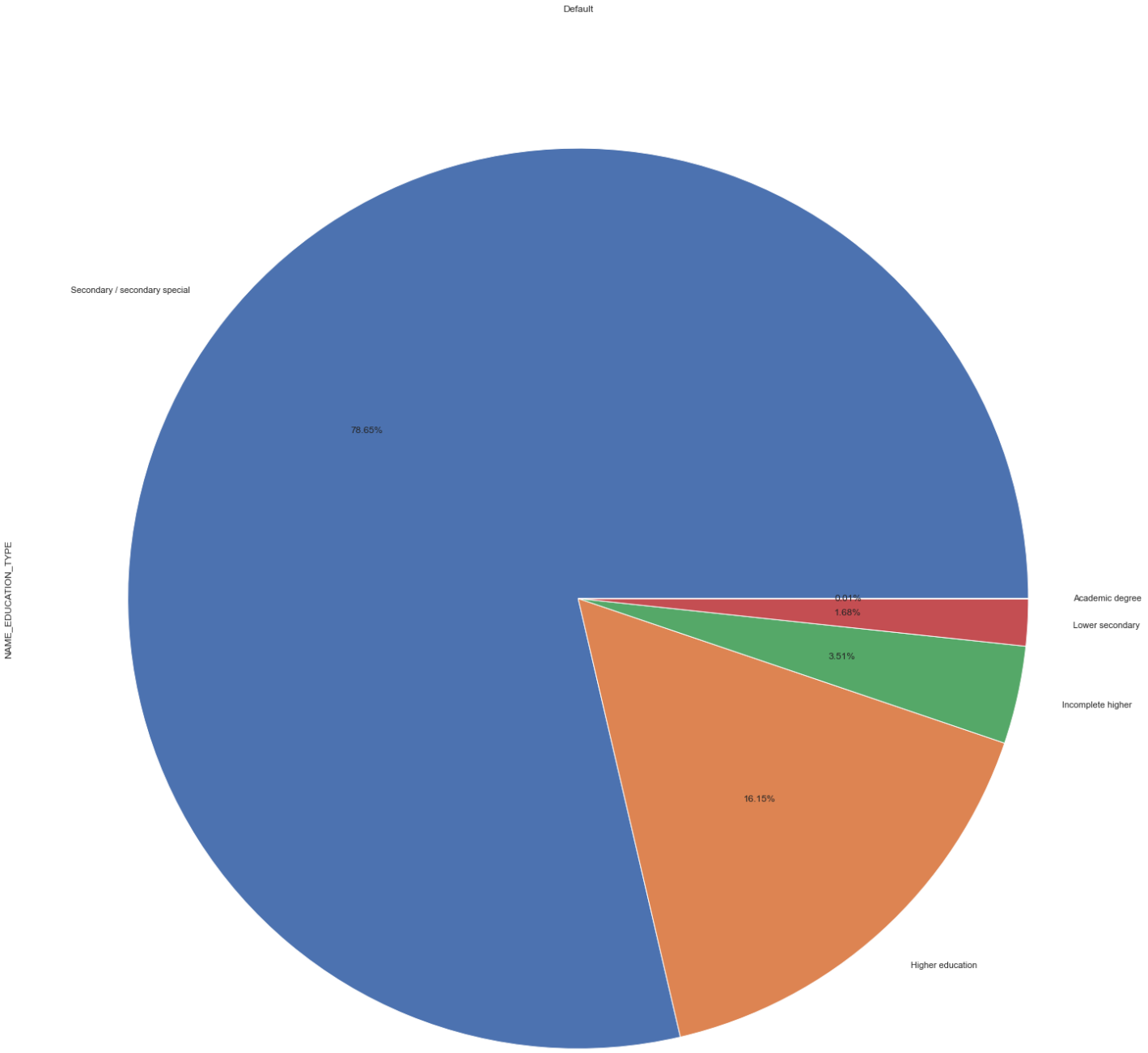


NAME_EDUCATION_TYPE

In [173…

```
df0.NAME_EDUCATION_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Non default')
plt.show()
df2.NAME_EDUCATION_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Default')
plt.show()
```



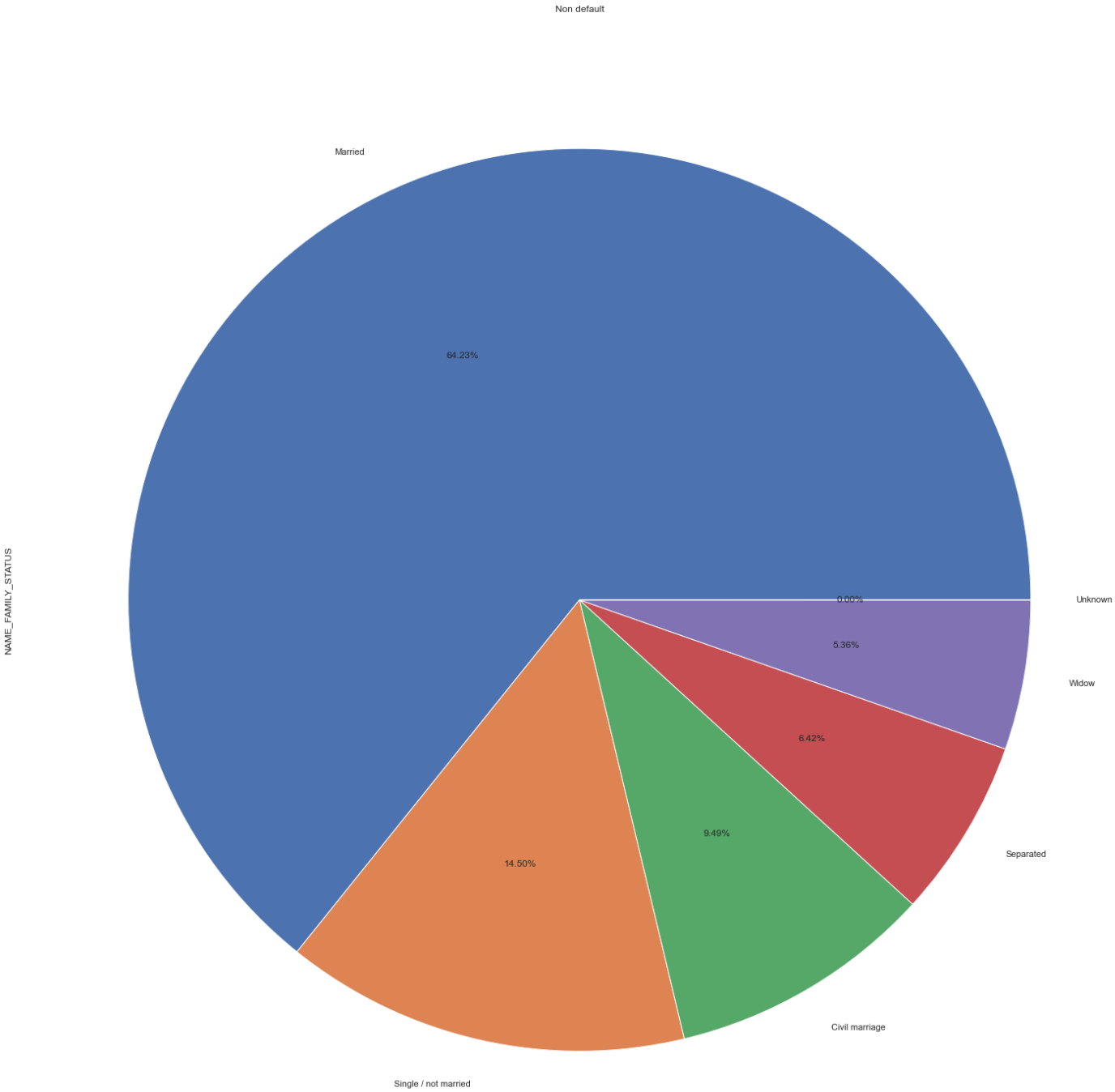


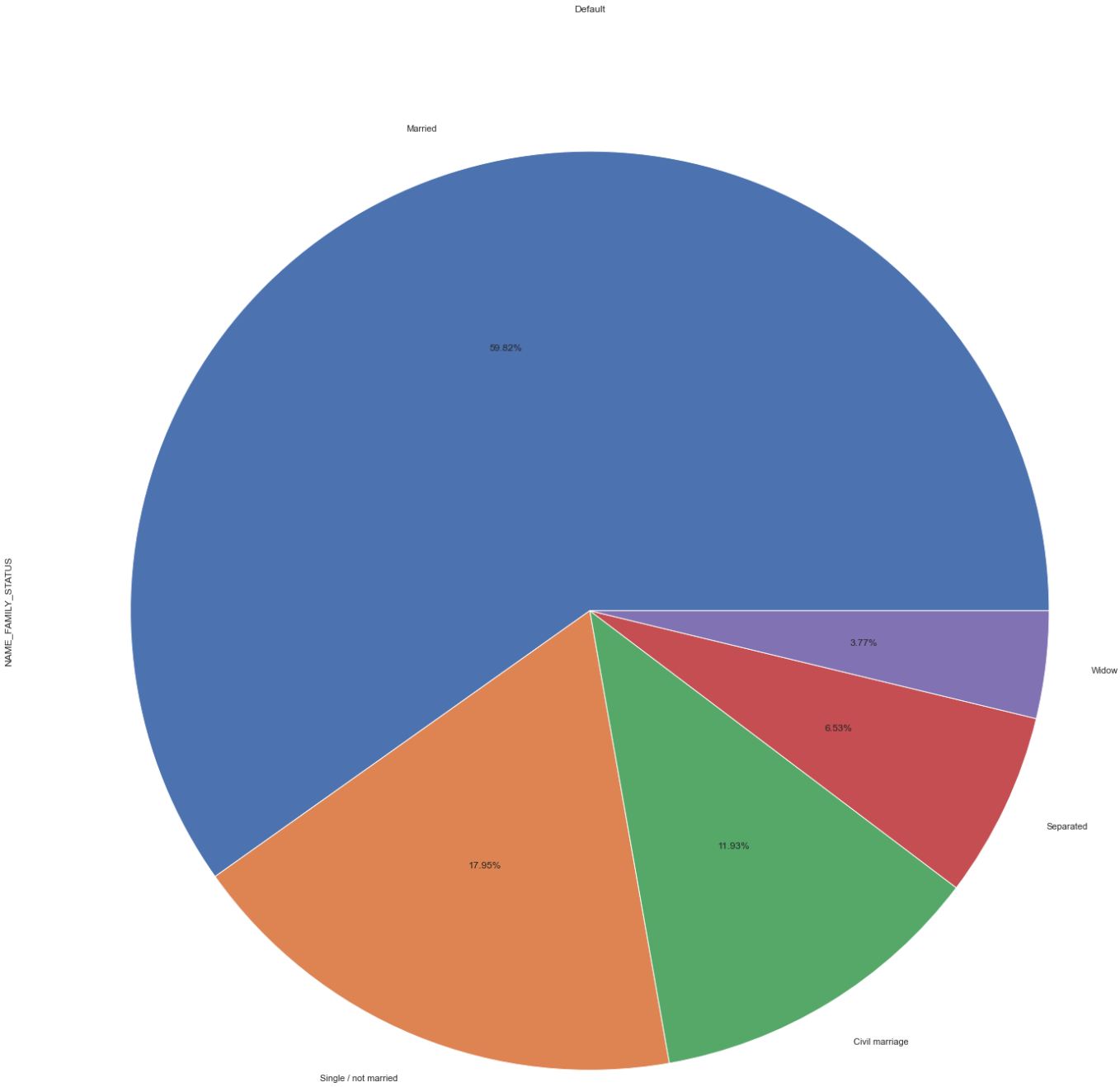
NAME_FAMILY_STATUS

In [ ]:

In [175…

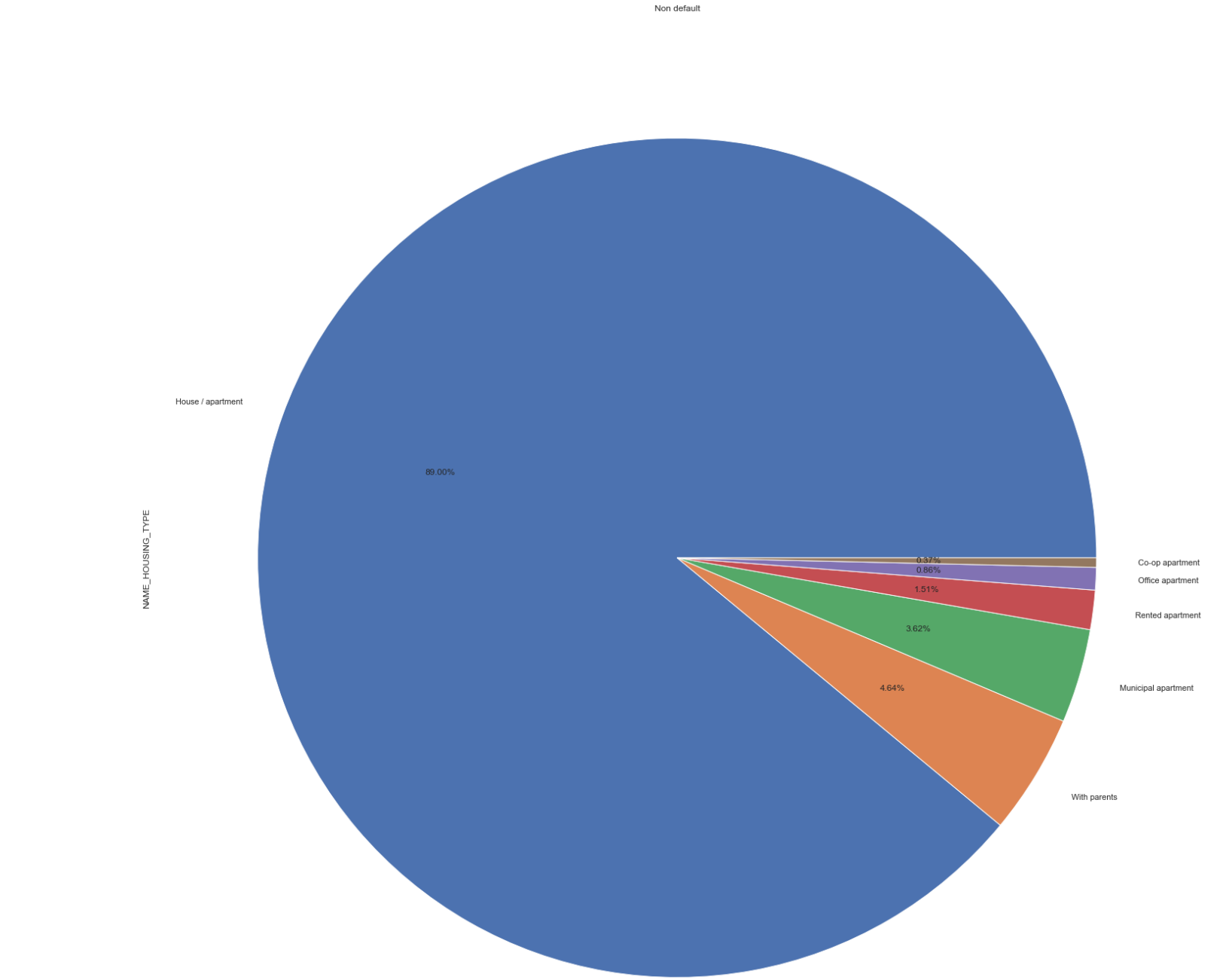
```
df0.NAME_FAMILY_STATUS.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Non default')
plt.show()
df2.NAME_FAMILY_STATUS.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Default')
plt.show()
```

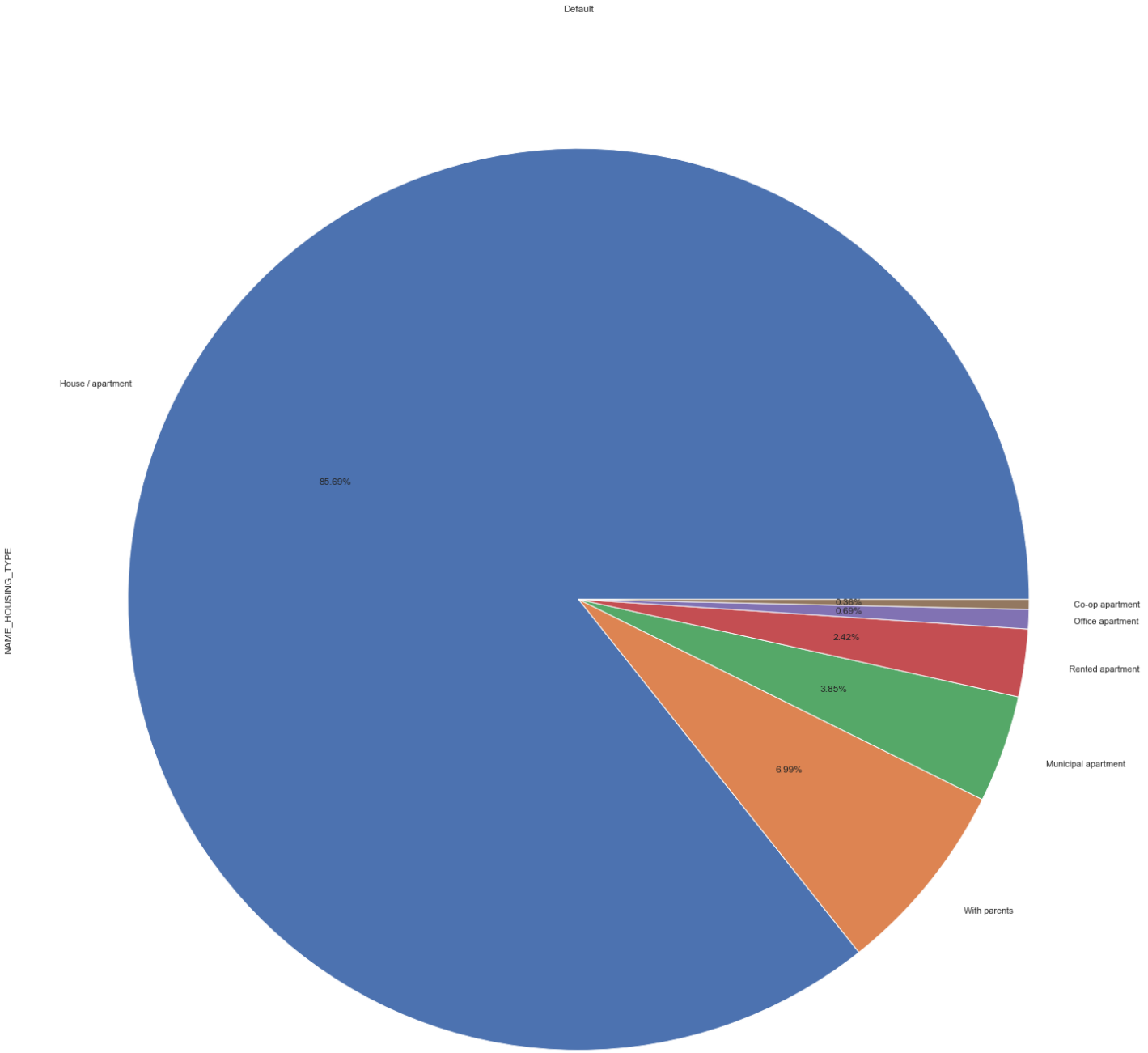




In [176…

```
df0.NAME_HOUSING_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Non default')
plt.show()
df2.NAME_HOUSING_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Default')
plt.show()
```





In [ ]:

In [ ]:
```
##### NAME_TYPE_SUITE - For non defaulters and defaulters client is unaccompanied in most
##### NAME_EDUCATION_TYPE-For non defaulters and defaulters most of the clients are having
#####                                                  For defaulters academic degree holders
##### NAME_FAMILY_STATUS-Highest non defaulters and defaulters are in the married category
##### NAME_HOUSING_TYPE-Highest non defaulters and defaulters are having house/apartment.
```

NAME_TYPE_SUITE - For non defaulters and defaulters client is unaccompanied in most of the cases.

NAME_EDUCATION_TYPE-For non defaulters and defaulters most of the clients are having secondary education.

For defaulters academic degree holders are less.

NAME_FAMILY_STATUS-Highest non defaulters and defaulters are in the married category.

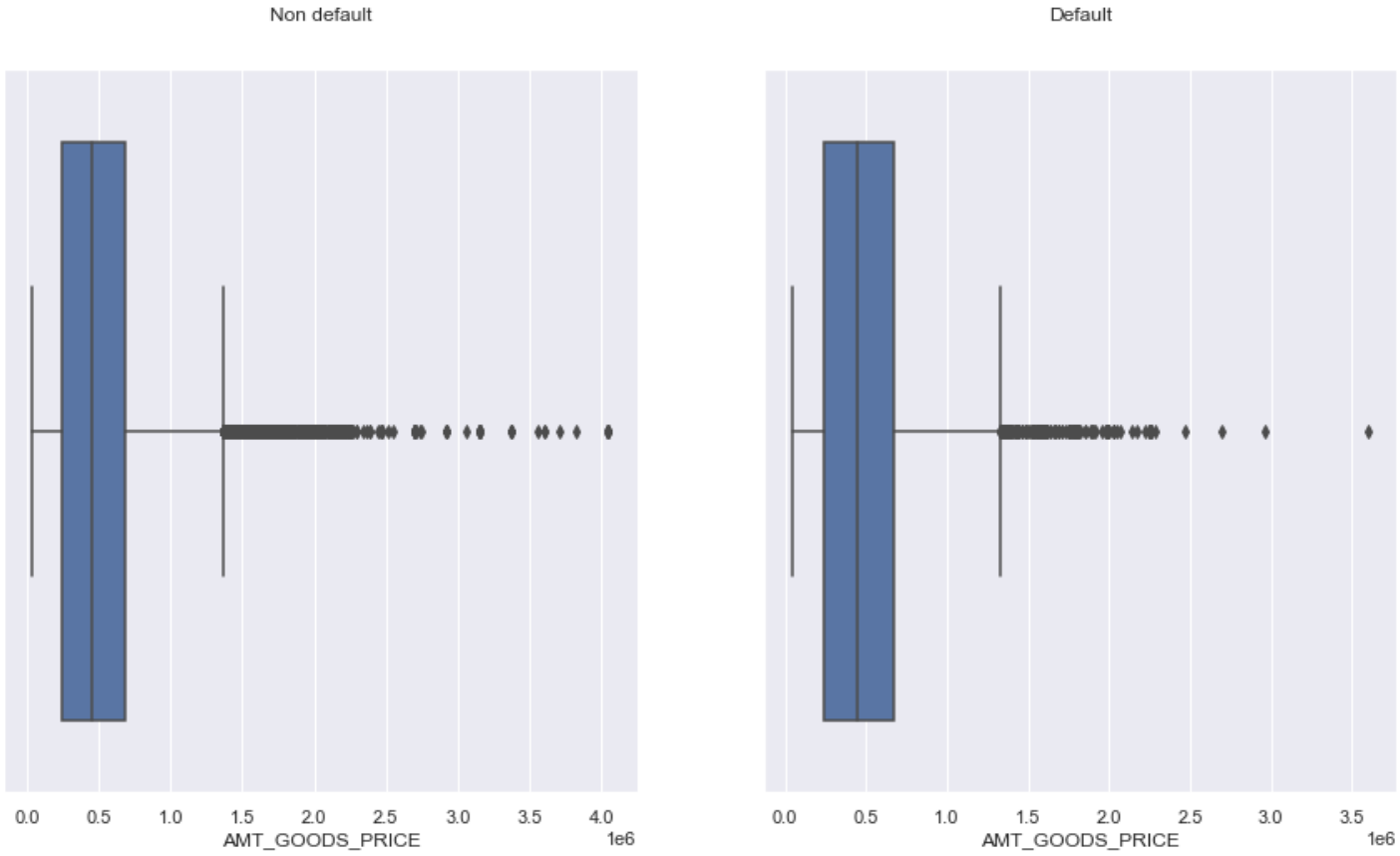
NAME_HOUSING_TYPE-Highest non defaulters and defaulters are having house/apartment.

NAME_HOUSING_TYPE

AMT_GOODS_PRICE

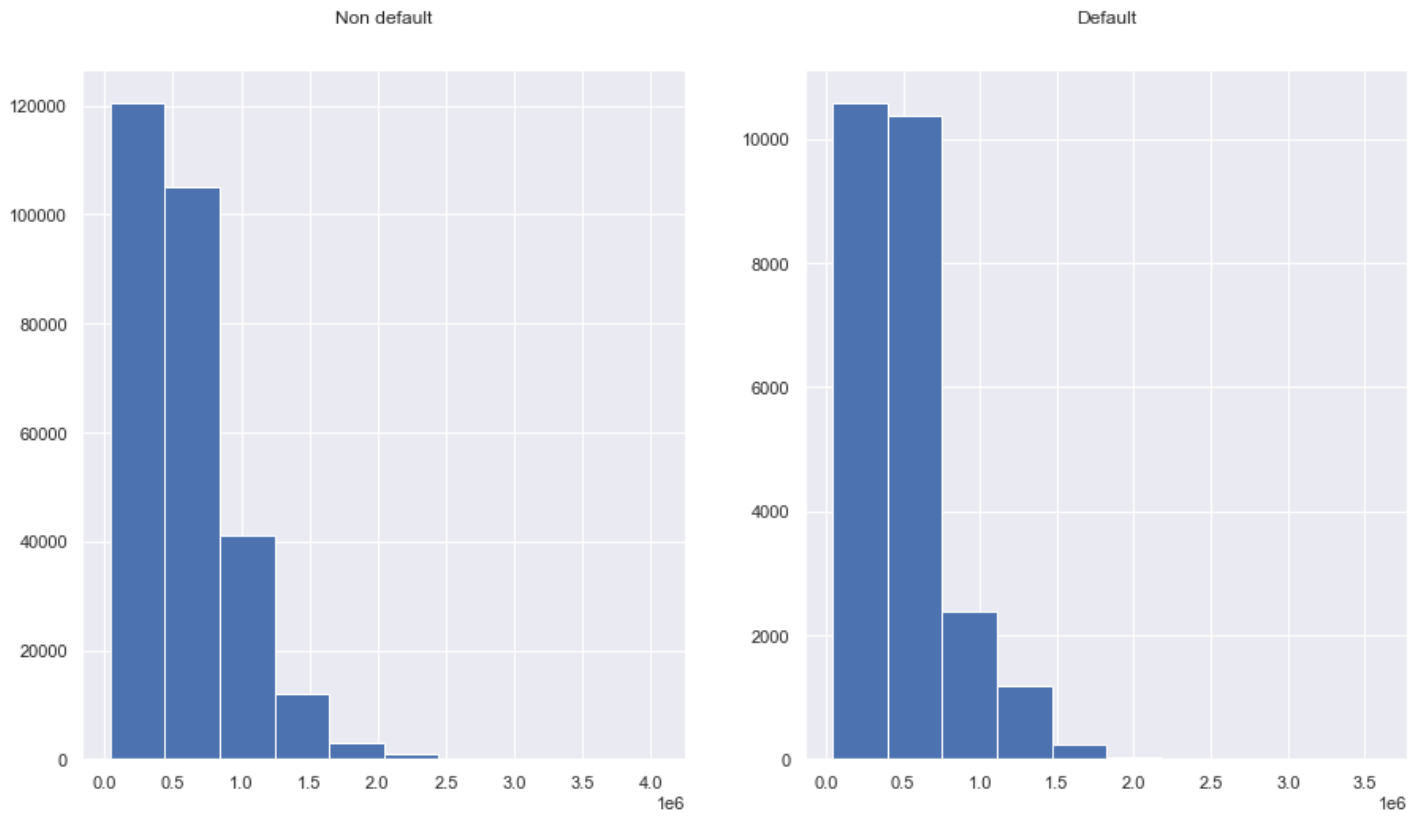
In [177…

```
plt.figure(figsize=(15,8))
plt.subplot(121)
sns.boxplot(x='AMT_GOODS_PRICE',data=df0)
plt.title('Non default')
plt.subplot(122)
sns.boxplot(x='AMT_GOODS_PRICE',data=df2)
plt.title("Default")
plt.show()
```



In [179…

```
plt.figure(figsize=(15,8))
plt.subplot(121)
plt.hist(x='AMT_GOODS_PRICE',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='AMT_GOODS_PRICE',data=df2)
plt.title("Default")
plt.show()
```



The price of the goods for which the loan is given is more for non defaulters

In [180…
```
df5=df1[['DAYS_BIRTH','DAYS_EMPLOYED','DAYS_ID_PUBLISH','CNT_FAM_MEMBERS','REGION_RATING_C
```

In [181…
```
df5.head()
```

Out[181…

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_ID_PUBLISH	CNT_FAM_MEMBERS	REGION_RATING_CLIENT	ORGANIZ
0	-9461	-637	-2120	1.0	2	Business
1	-16765	-1188	-291	2.0	1	
2	-19046	-225	-2531	1.0	2	
3	-19005	-3039	-2437	2.0	2	Business
4	-19932	-3038	-3458	1.0	2	

In [182…
```
[i for i in df5.columns  if i in category]
```

Out[182…
```
['ORGANIZATION_TYPE']
```

In [183…
```
[i for i in df5.columns  if i in number]
```

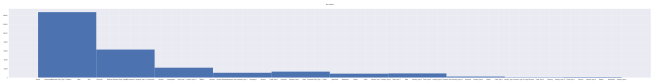
Out[183…
```
['DAYS_BIRTH',
 'DAYS_EMPLOYED',
 'DAYS_ID_PUBLISH',
 'CNT_FAM_MEMBERS',
 'REGION_RATING_CLIENT']
```

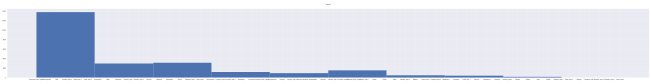
In [ ]:
```

```

ORGANIZATION_TYPE

In [184…

```
plt.figure(figsize=(200,10))
plt.subplot(121)
plt.hist(x='ORGANIZATION_TYPE',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='ORGANIZATION_TYPE',data=df2)
plt.title("Default")
plt.show()
```

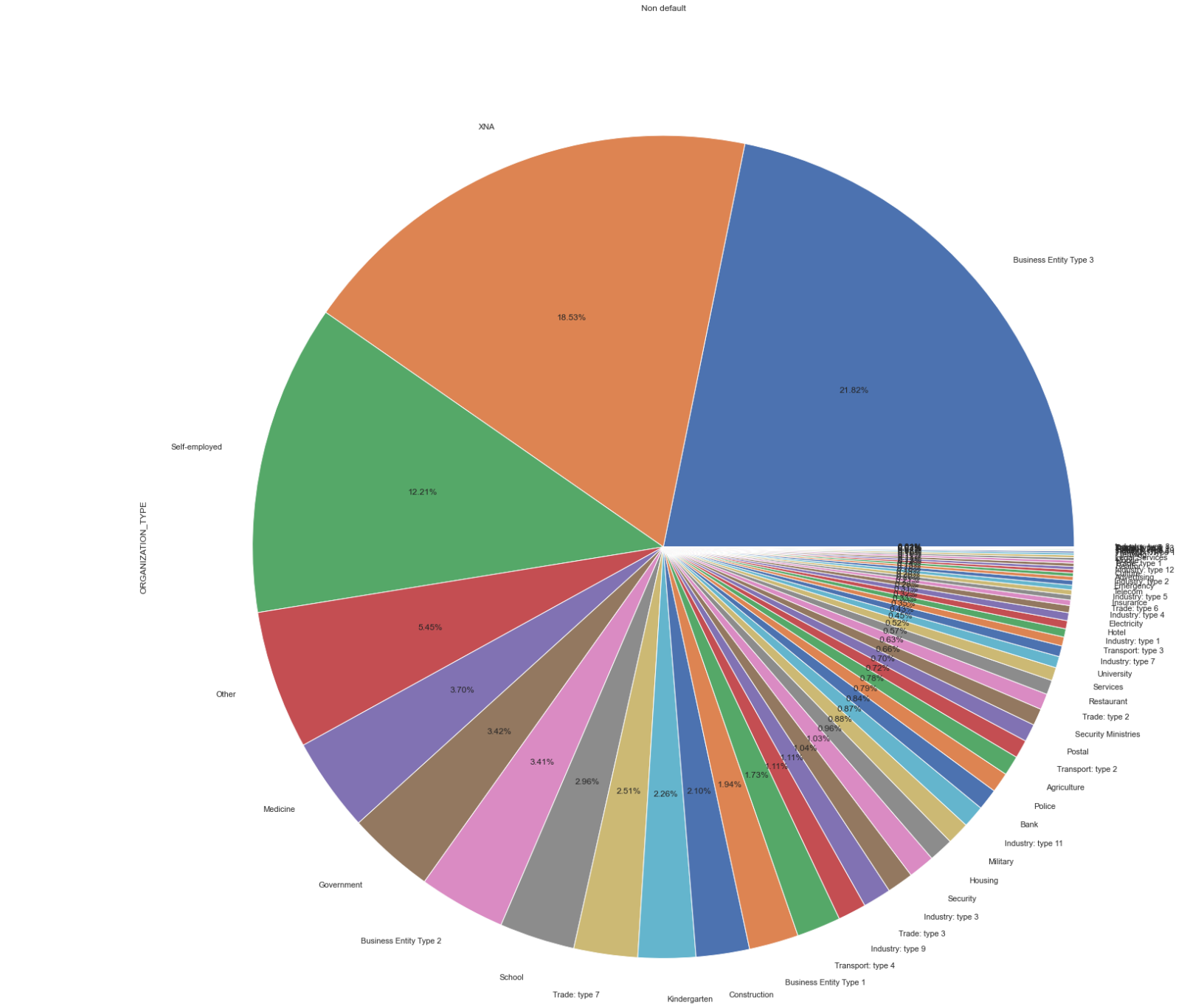


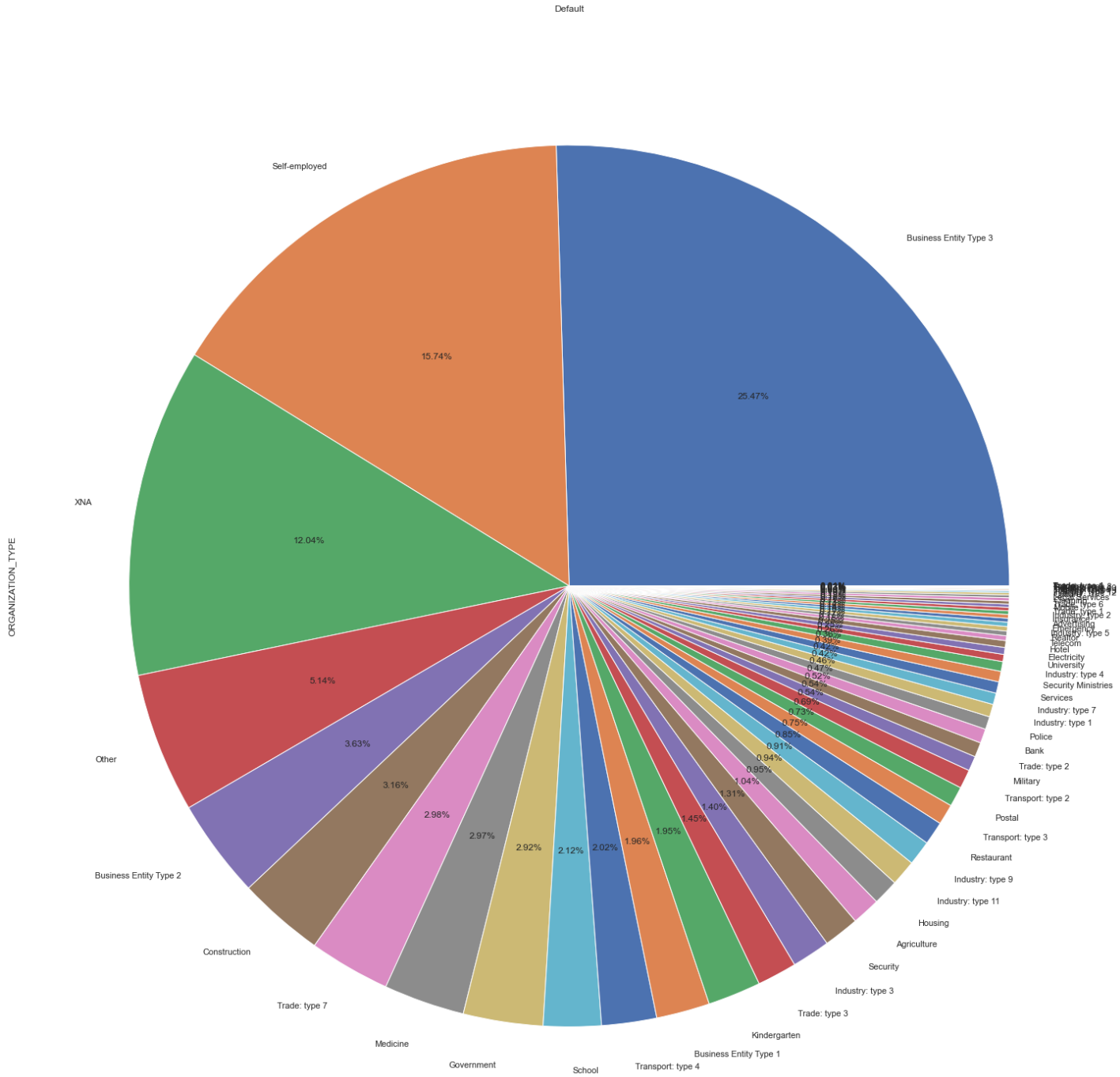


In [ ]:

In [186…

```
df0.ORGANIZATION_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Non default')
plt.show()
df2.ORGANIZATION_TYPE.value_counts(normalize=True).plot.pie(autopct='%1.2f%%')
plt.tight_layout()
plt.title('Default')
plt.show()
```





Both non defaulters and defaulters are high in business entity type 3

In [ ]:

DAYS_BIRTH

In [188…

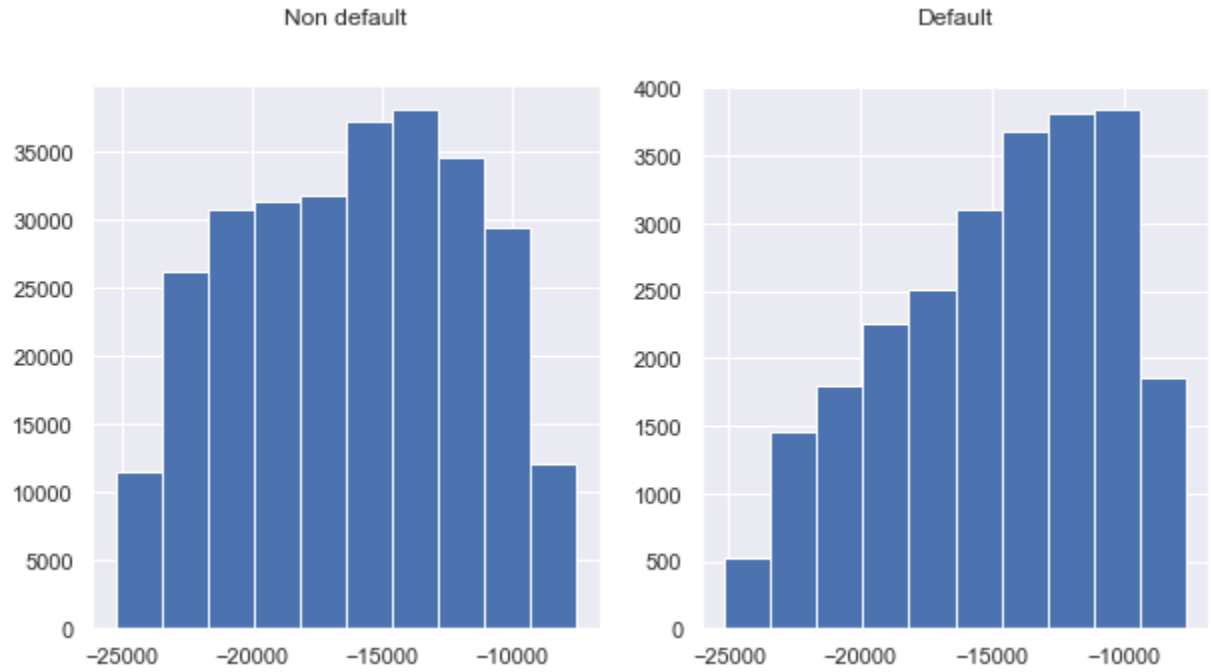
```
df5.head()
```

Out[188…

	DAYS_BIRTH	DAYS_EMPLOYED	DAYS_ID_PUBLISH	CNT_FAM_MEMBERS	REGION_RATING_CLIENT	ORGANIZ
0	-9461	-637	-2120	1.0	2	Business
1	-16765	-1188	-291	2.0	1	
2	-19046	-225	-2531	1.0	2	
3	-19005	-3039	-2437	2.0	2	Business
		-3038	-3458	1.0	2	

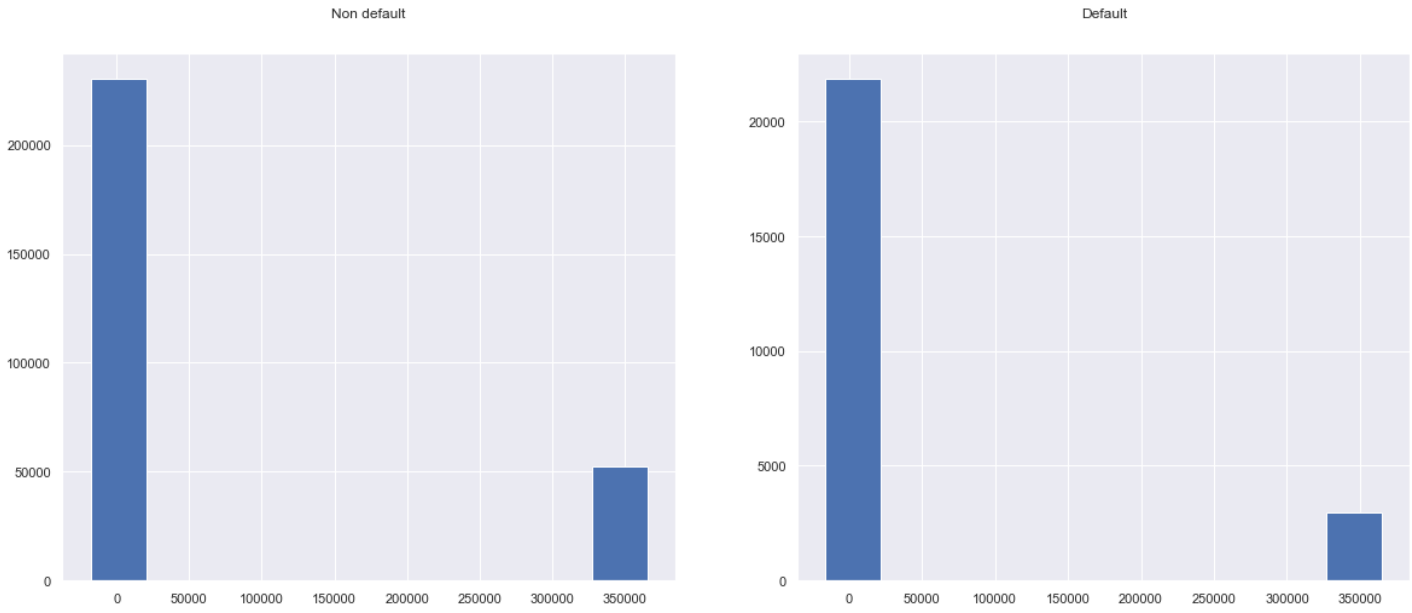
In [189…

```
plt.figure(figsize=(10,5))
plt.subplot(121)
plt.hist(x='DAYS_BIRTH',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='DAYS_BIRTH',data=df2)
plt.title("Default")
plt.show()
```



In [190…

```
plt.figure(figsize=(20,8))
plt.subplot(121)
plt.hist(x='DAYS_EMPLOYED',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='DAYS_EMPLOYED',data=df2)
plt.title("Default")
plt.show()
```

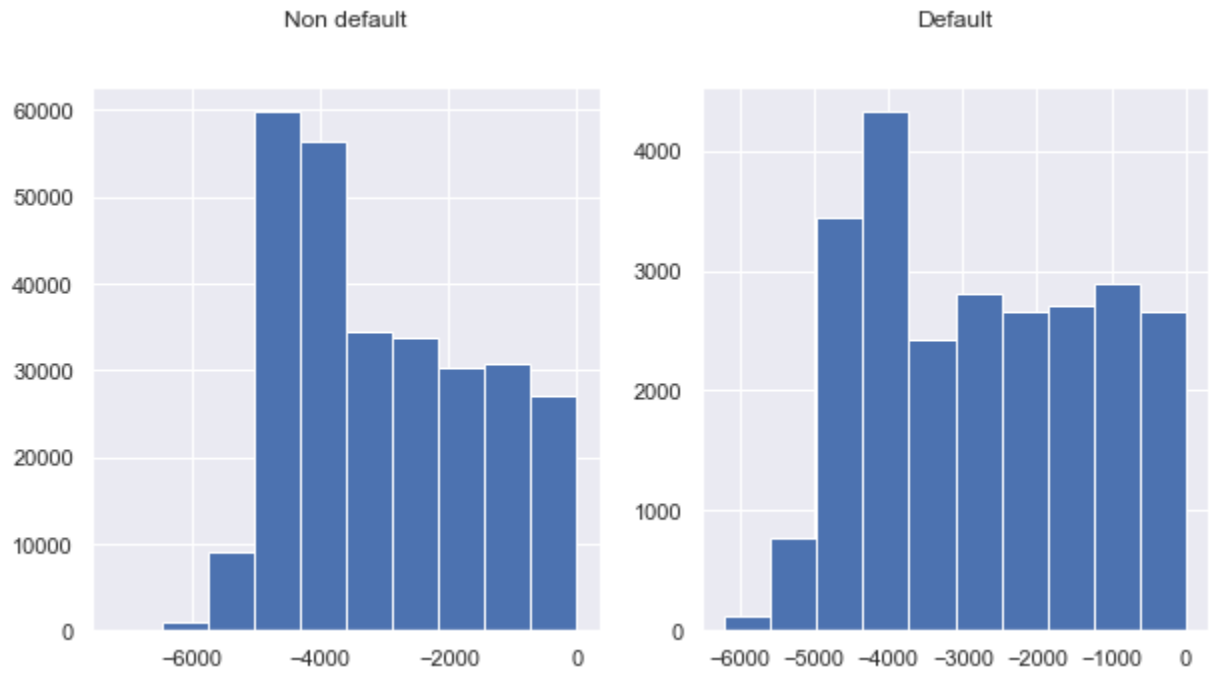


In [ ]:

DAYS_ID_PUBLISH

In [191…

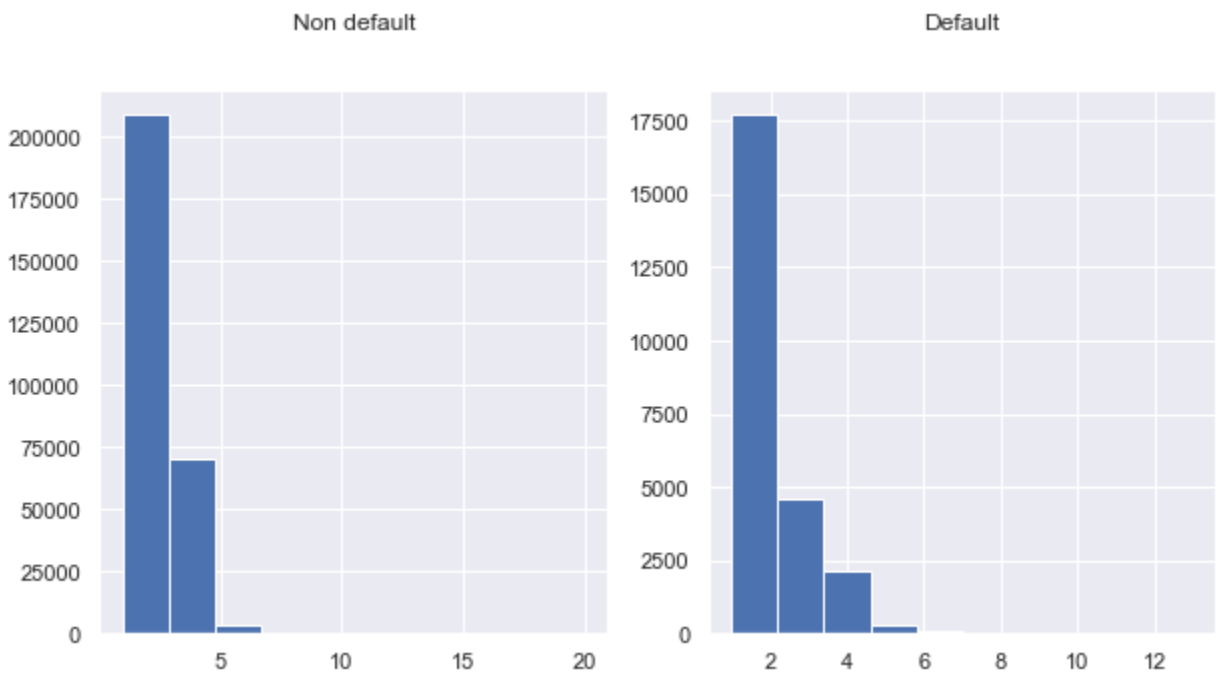
```
plt.figure(figsize=(10,5))
plt.subplot(121)
plt.hist(x='DAYS_ID_PUBLISH',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='DAYS_ID_PUBLISH',data=df2)
plt.title("Default")
plt.show()
```



CNT_FAM_MEMBERS

In [192…

```
plt.figure(figsize=(10,5))
plt.subplot(121)
plt.hist(x='CNT_FAM_MEMBERS',data=df0)
plt.title('Non default')
plt.subplot(122)
plt.hist(x='CNT_FAM_MEMBERS',data=df2)
plt.title("Default")
plt.show()
```



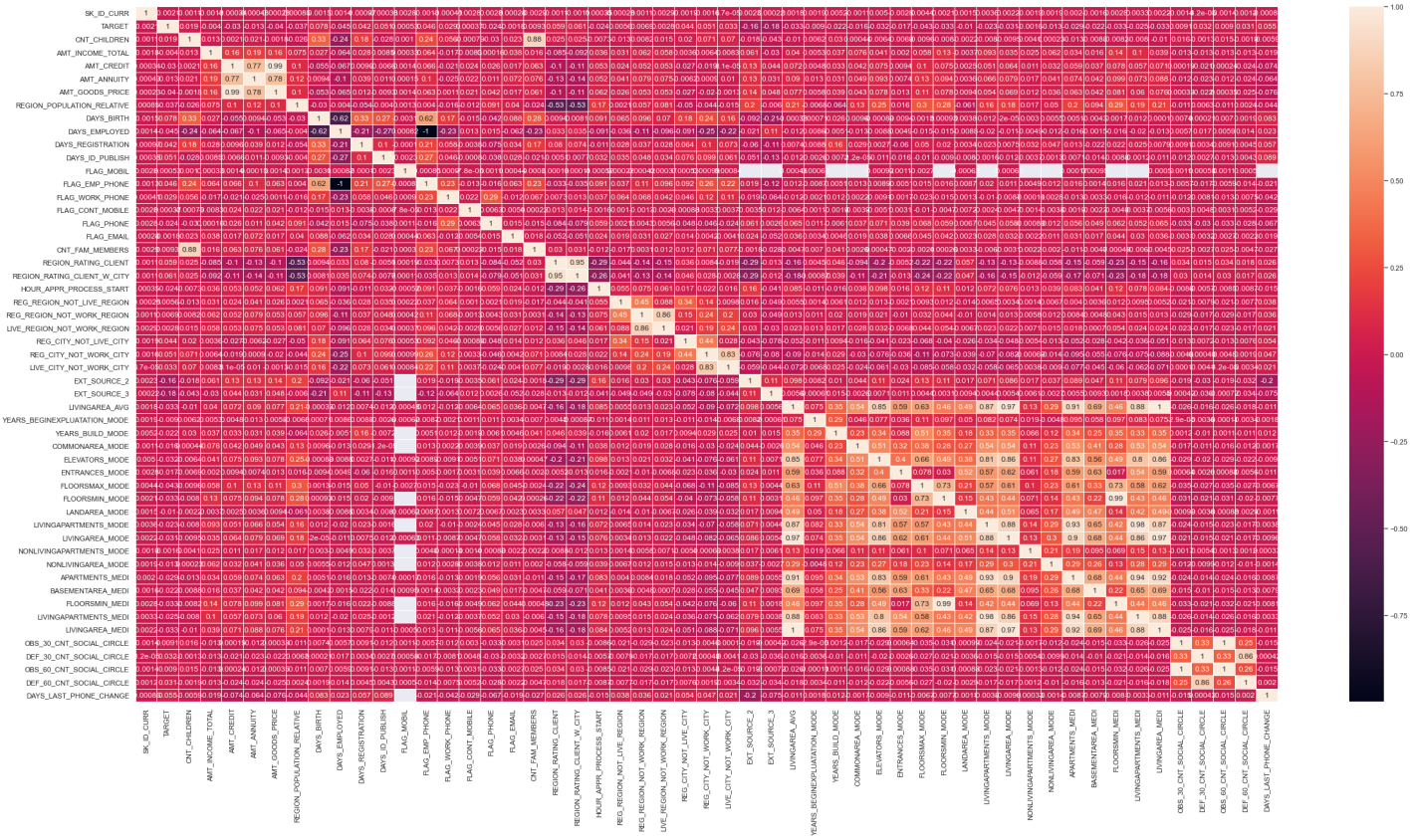
In [193…

```
sns.set(rc = {'figure.figsize':(40,20)})
sns.heatmap(df1[['SK_ID_CURR',
 'TARGET',
 'CNT_CHILDREN',
 'AMT_INCOME_TOTAL',
 'AMT_CREDIT',
 'AMT_ANNUITY',
 'AMT_GOODS_PRICE',
 'REGION_POPULATION_RELATIVE',
 'DAYS_BIRTH',
 'DAYS_EMPLOYED',
 'DAYS_REGISTRATION',
 'DAYS_ID_PUBLISH',
 'FLAG_MOBIL',
 'FLAG_EMP_PHONE',
 'FLAG_WORK_PHONE',
 'FLAG_CONT_MOBILE',
 'FLAG_PHONE',
 'FLAG_EMAIL',
 'CNT_FAM_MEMBERS',
 'REGION_RATING_CLIENT',
 'REGION_RATING_CLIENT_W_CITY',
 'HOUR_APPR_PROCESS_START',
 'REG_REGION_NOT_LIVE_REGION',
 'REG_REGION_NOT_WORK_REGION',
 'LIVE_REGION_NOT_WORK_REGION',
 'REG_CITY_NOT_LIVE_CITY',
 'REG_CITY_NOT_WORK_CITY',
 'LIVE_CITY_NOT_WORK_CITY',
 'EXT_SOURCE_2',
 'EXT_SOURCE_3',
 'LIVINGAREA_AVG',
 'YEARS_BEGINEXPLUATATION_MODE',
 'YEARS_BUILD_MODE',
 'COMMONAREA_MODE',
 'ELEVATORS_MODE',
 'ENTRANCES_MODE',
 'FLOORSMAX_MODE',
 'FLOORSMIN_MODE',
 'LANDAREA_MODE',
 'LIVINGAPARTMENTS_MODE',
 'MODE',
```

```
 'NONLIVINGAPARTMENTS_MODE',
 'NONLIVINGAREA_MODE',
 'APARTMENTS_MEDI',
 'BASEMENTAREA_MEDI',
 'FLOORSMIN_MEDI',
 'LIVINGAPARTMENTS_MEDI',
 'LIVINGAREA_MEDI',
 'OBS_30_CNT_SOCIAL_CIRCLE',
 'DEF_30_CNT_SOCIAL_CIRCLE',
 'OBS_60_CNT_SOCIAL_CIRCLE',
 'DEF_60_CNT_SOCIAL_CIRCLE',
 'DAYS_LAST_PHONE_CHANGE','NAME_CONTRACT_TYPE',
 'CODE_GENDER',
 'FLAG_OWN_CAR',
 'FLAG_OWN_REALTY',
 'NAME_TYPE_SUITE',
 'NAME_INCOME_TYPE',
 'NAME_EDUCATION_TYPE',
 'NAME_FAMILY_STATUS',
 'NAME_HOUSING_TYPE',
 'WEEKDAY_APPR_PROCESS_START',
 'ORGANIZATION_TYPE']].corr(),annot=True, linewidths=.5)
```

Out[193…

```
<AxesSubplot:>
```



REGION_RATING_CLIENT and REGION_RATING_CLIENT_W_CITY are highly correlated

LIVINGAPARTMENTS_MEDI and APARTMENTS_MEDI are highly correlated

AMT_CREDIT and AMT_GOODS_PRICE are highly correlated

AMT_CREDIT and ANNUITY are highly correlated

APARTMENTS_MEDI and LIVINGAREA_AVG are highly correlated

LIVINGAREA_MODE and ELEVATORS_MODE are highly correlated

----previous_application----

In [ ]:

```
```

In [194…

```
dfp=pd.read_csv(r"D:\Python\EDA  ASSIGNMENT\previous_application.csv")
dfp.head()
```

Out[194…

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	

5 rows × 37 columns

In [195…

```
dfp.columns.to_list()
```

Out[195…

```
['SK_ID_PREV',
 'SK_ID_CURR',
 'NAME_CONTRACT_TYPE',
 'AMT_ANNUITY',
 'AMT_APPLICATION',
 'AMT_CREDIT',
 'AMT_DOWN_PAYMENT',
 'AMT_GOODS_PRICE',
 'WEEKDAY_APPR_PROCESS_START',
 'HOUR_APPR_PROCESS_START',
 'FLAG_LAST_APPL_PER_CONTRACT',
 'NFLAG_LAST_APPL_IN_DAY',
 'RATE_DOWN_PAYMENT',
 'RATE_INTEREST_PRIMARY',
 'RATE_INTEREST_PRIVILEGED',
 'NAME_CASH_LOAN_PURPOSE',
 'NAME_CONTRACT_STATUS',
 'DAYS_DECISION',
 'NAME_PAYMENT_TYPE',
 'CODE_REJECT_REASON',
 'NAME_TYPE_SUITE',
 'NAME_CLIENT_TYPE',
 'NAME_GOODS_CATEGORY',
 'NAME_PORTFOLIO',
 'NAME_PRODUCT_TYPE',
 'CHANNEL_TYPE',
 'SELLERPLACE_AREA',
 'NAME_SELLER_INDUSTRY',
 'CNT_PAYMENT',
 'NAME_YIELD_GROUP',
 'PRODUCT_COMBINATION',
 'DAYS_FIRST_DRAWING',
 'DAYS_FIRST_DUE',
 'DAYS_LAST_DUE_1ST_VERSION',
 'DAYS_LAST_DUE',
 'DAYS_TERMINATION',
 'NFLAG_INSURED_ON_APPROVAL']
```

In [196…
```
df.describe()
```

Out[196…

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	

8 rows × 106 columns

In [197…
```
missing=dfp.isnull().sum()*100/len(dfp)
missing[missing>50]
```

Out[197…
```
AMT_DOWN_PAYMENT            53.636480
RATE_DOWN_PAYMENT           53.636480
RATE_INTEREST_PRIMARY       99.643698
RATE_INTEREST_PRIVILEGED    99.643698
dtype: float64
```

In [198…
```
dfp1=dfp.drop(['AMT_DOWN_PAYMENT','RATE_DOWN_PAYMENT','RATE_INTEREST_PRIMARY','RATE_INTERE
```

In [199…
```
dfp1.head()
```

Out[199…

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	

5 rows × 33 columns

In [200…
```
dfp1.columns.to_list()
```

Out[200…
```
['SK_ID_PREV',
 'SK_ID_CURR',
 'NAME_CONTRACT_TYPE',
 'AMT_ANNUITY',
 'AMT_APPLICATION',
 'AMT_CREDIT',
 'AMT_GOODS_PRICE',
 'WEEKDAY_APPR_PROCESS_START',
 'HOUR_APPR_PROCESS_START',
 'FLAG_LAST_APPL_PER_CONTRACT',
 'NFLAG_LAST_APPL_IN_DAY',
 'AN_PURPOSE',
```

```
'NAME_CONTRACT_STATUS',
'DAYS_DECISION',
'NAME_PAYMENT_TYPE',
'CODE_REJECT_REASON',
'NAME_TYPE_SUITE',
'NAME_CLIENT_TYPE',
'NAME_GOODS_CATEGORY',
'NAME_PORTFOLIO',
'NAME_PRODUCT_TYPE',
'CHANNEL_TYPE',
'SELLERPLACE_AREA',
'NAME_SELLER_INDUSTRY',
'CNT_PAYMENT',
'NAME_YIELD_GROUP',
'PRODUCT_COMBINATION',
'DAYS_FIRST_DRAWING',
'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
'DAYS_LAST_DUE',
'DAYS_TERMINATION',
'NFLAG_INSURED_ON_APPROVAL']
```

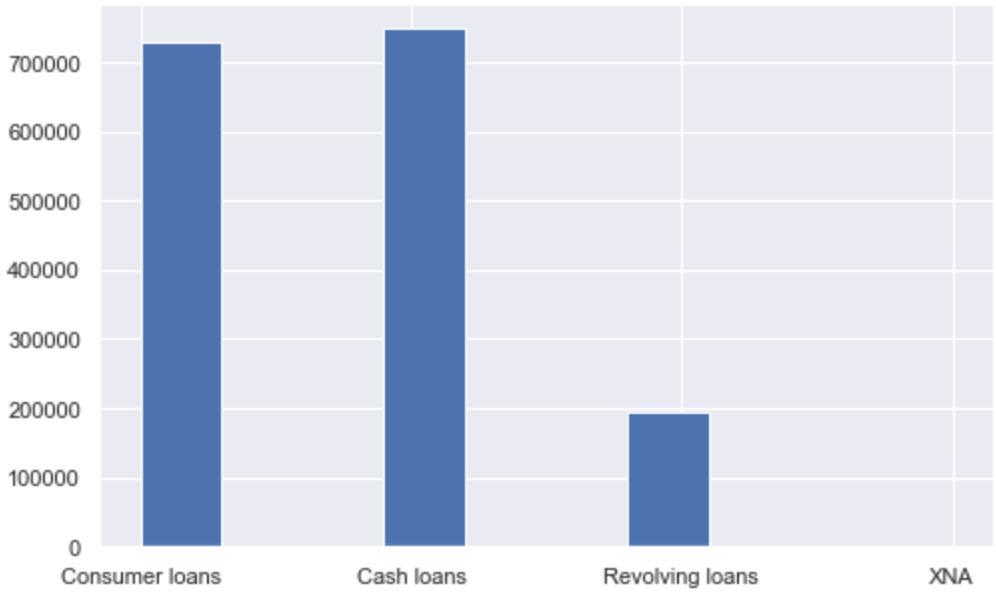
In [201…
```
dfp2=dfp1[['NAME_CONTRACT_TYPE','AMT_ANNUITY','AMT_APPLICATION','AMT_CREDIT','AMT_GOODS_PR
```

In [202…
```
dfp2.columns
```

Out[202…
```
Index(['NAME_CONTRACT_TYPE', 'AMT_ANNUITY', 'AMT_APPLICATION', 'AMT_CREDIT',
       'AMT_GOODS_PRICE'],
      dtype='object')
```

Univariate analysis

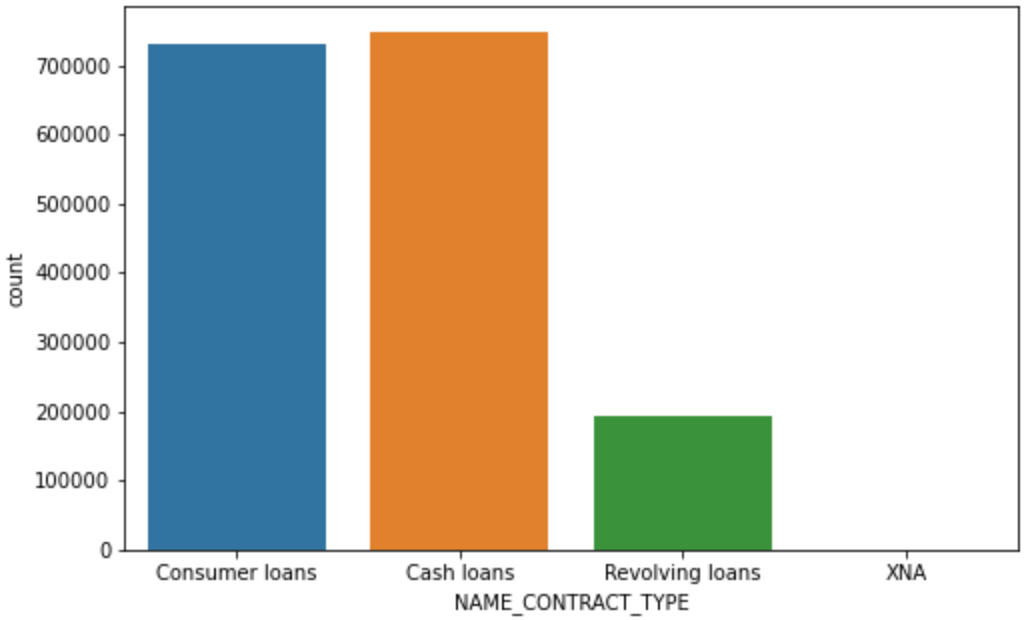
In [203…
```
plt.figure(figsize=(8,5))
plt.hist(dfp2['NAME_CONTRACT_TYPE'])
plt.show()
```



In [ ]:
```
```

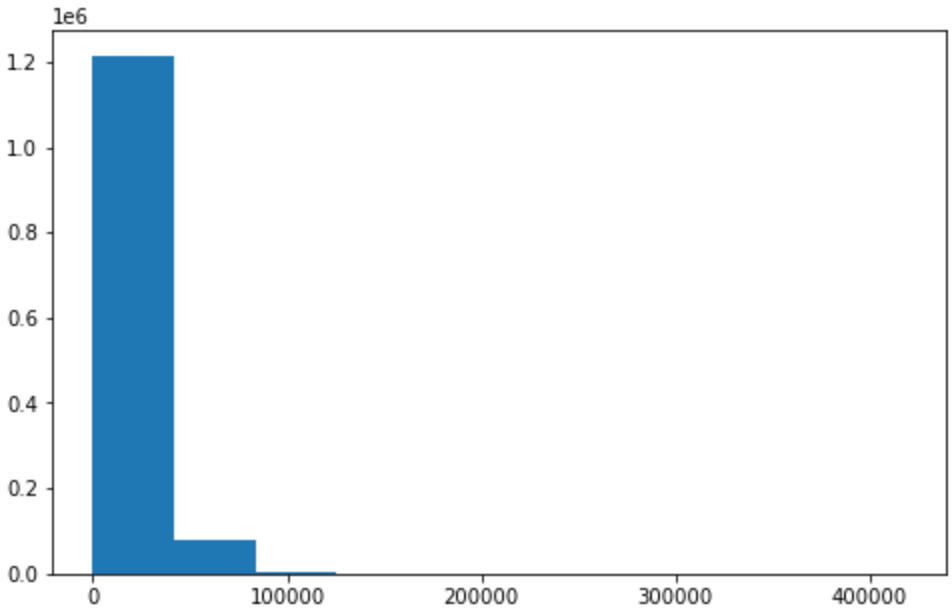
In [34]:
```
plt.figure(figsize=(8,5))
sns.countplot(x="NAME_CONTRACT_TYPE",data=dfp2)
```

Out[34]:
```
<AxesSubplot:xlabel='NAME_CONTRACT_TYPE', ylabel='count'>
```



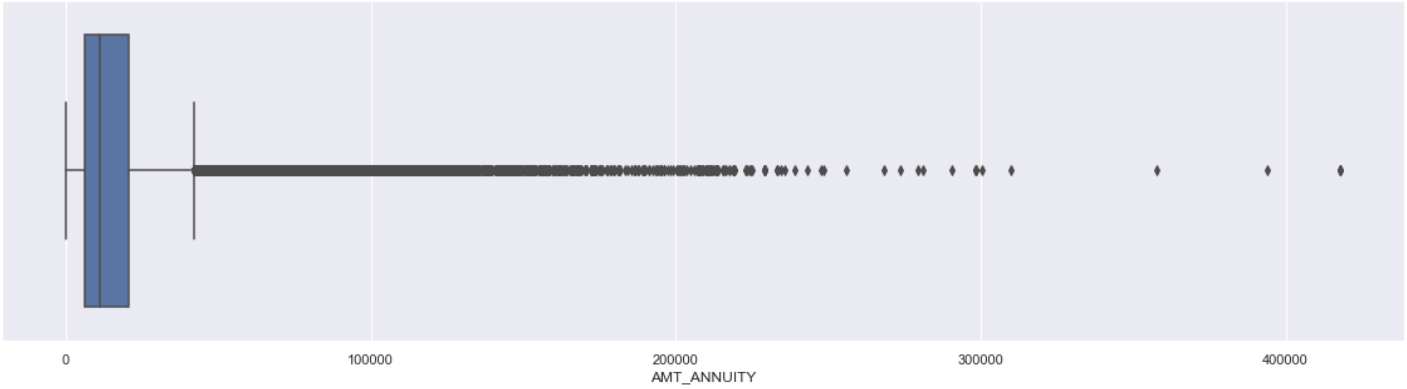
Cash loan is high when compared to consumer loan and revolving loan

In [35]:
```
plt.figure(figsize=(8,5))
plt.hist(dfp2['AMT_ANNUITY'])
plt.show()
```

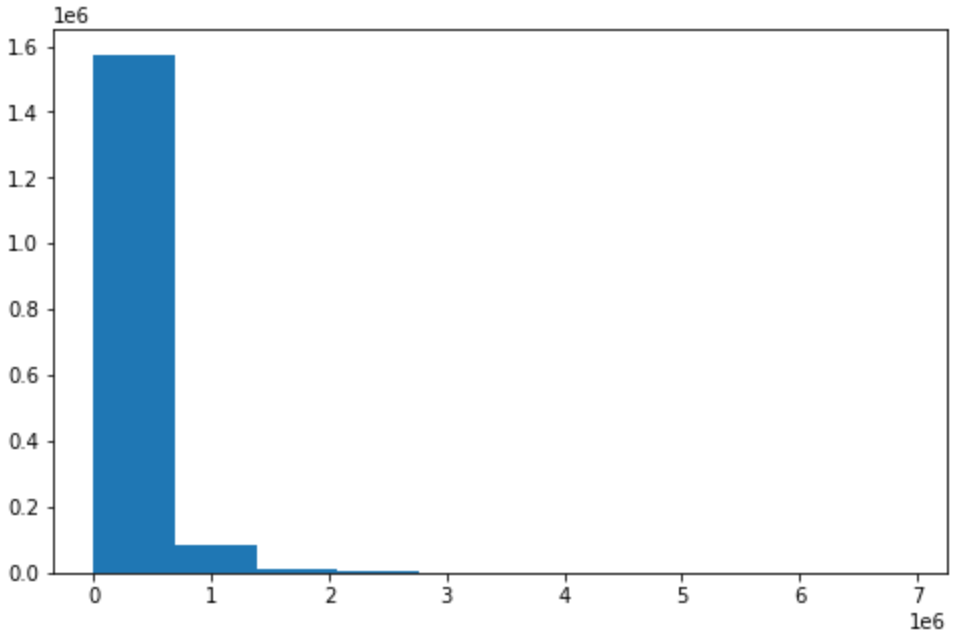


In [254…
```
sns.set(rc={'figure.figsize':(20,5)})
sns.boxplot(dfp2["AMT_ANNUITY"])
```

Out[254…
```
<AxesSubplot:xlabel='AMT_ANNUITY'>
```

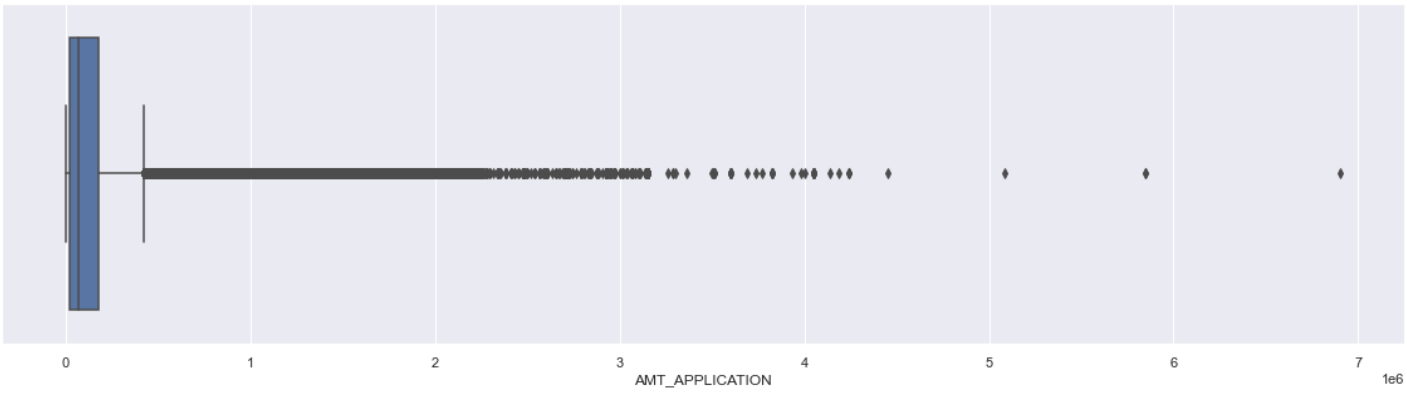


In [37]:
```
plt.figure(figsize=(8,5))
plt.hist(dfp2['AMT_APPLICATION'])
plt.show()
```

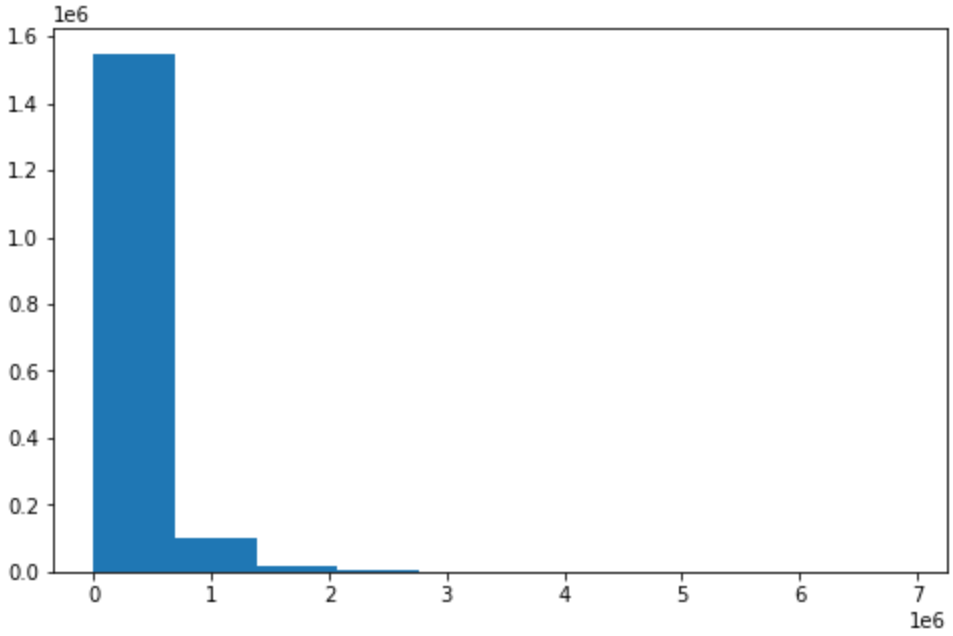


In [251…
```
sns.set(rc={'figure.figsize':(20,5)})
sns.boxplot(dfp2["AMT_APPLICATION"])
```

Out[251…
```
<AxesSubplot:xlabel='AMT_APPLICATION'>
```



In [39]:
```
plt.figure(figsize=(8,5))
plt.hist(dfp2['AMT_CREDIT'])
plt.show()
```

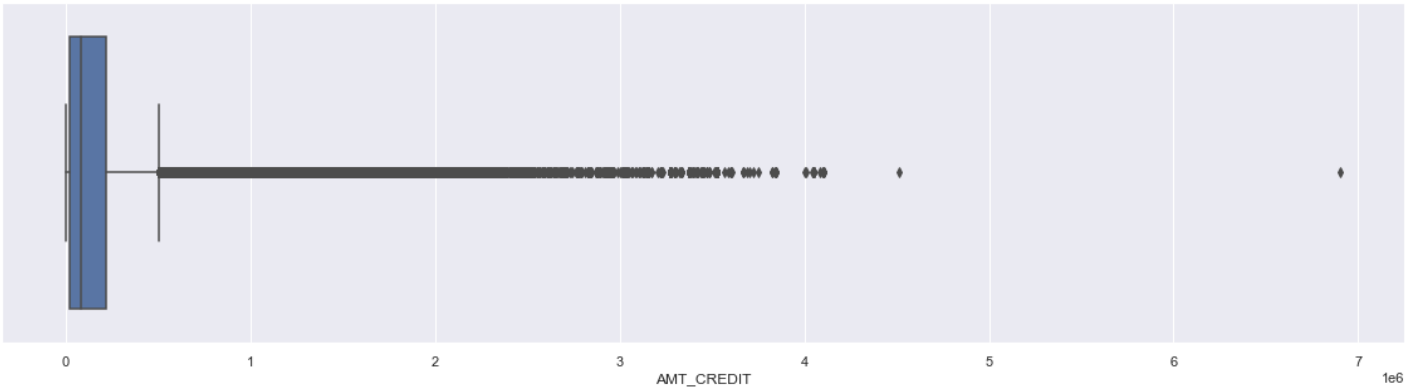


In [252…

```
sns.set(rc={'figure.figsize':(20,5)})
sns.boxplot(dfp2["AMT_CREDIT"])
```

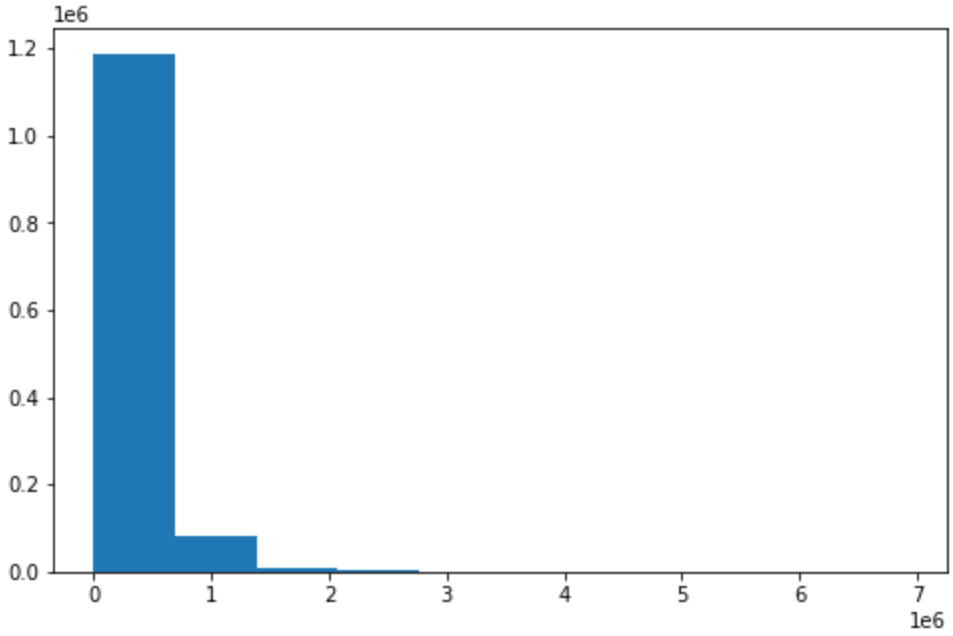
Out[252…

```
<AxesSubplot:xlabel='AMT_CREDIT'>
```



In [41]:

```
plt.figure(figsize=(8,5))
plt.hist(dfp2['AMT_GOODS_PRICE'])
plt.show()
```

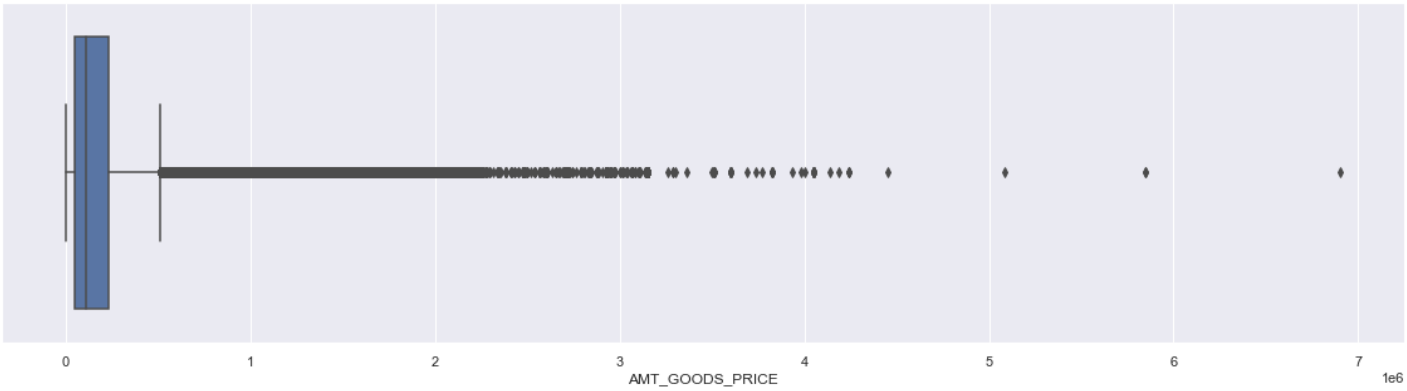


In [253…

```
sns.set(rc={'figure.figsize':(20,5)})
sns.boxplot(dfp2["AMT_GOODS_PRICE"])
```

Out[253…

```
<AxesSubplot:xlabel='AMT_GOODS_PRICE'>
```

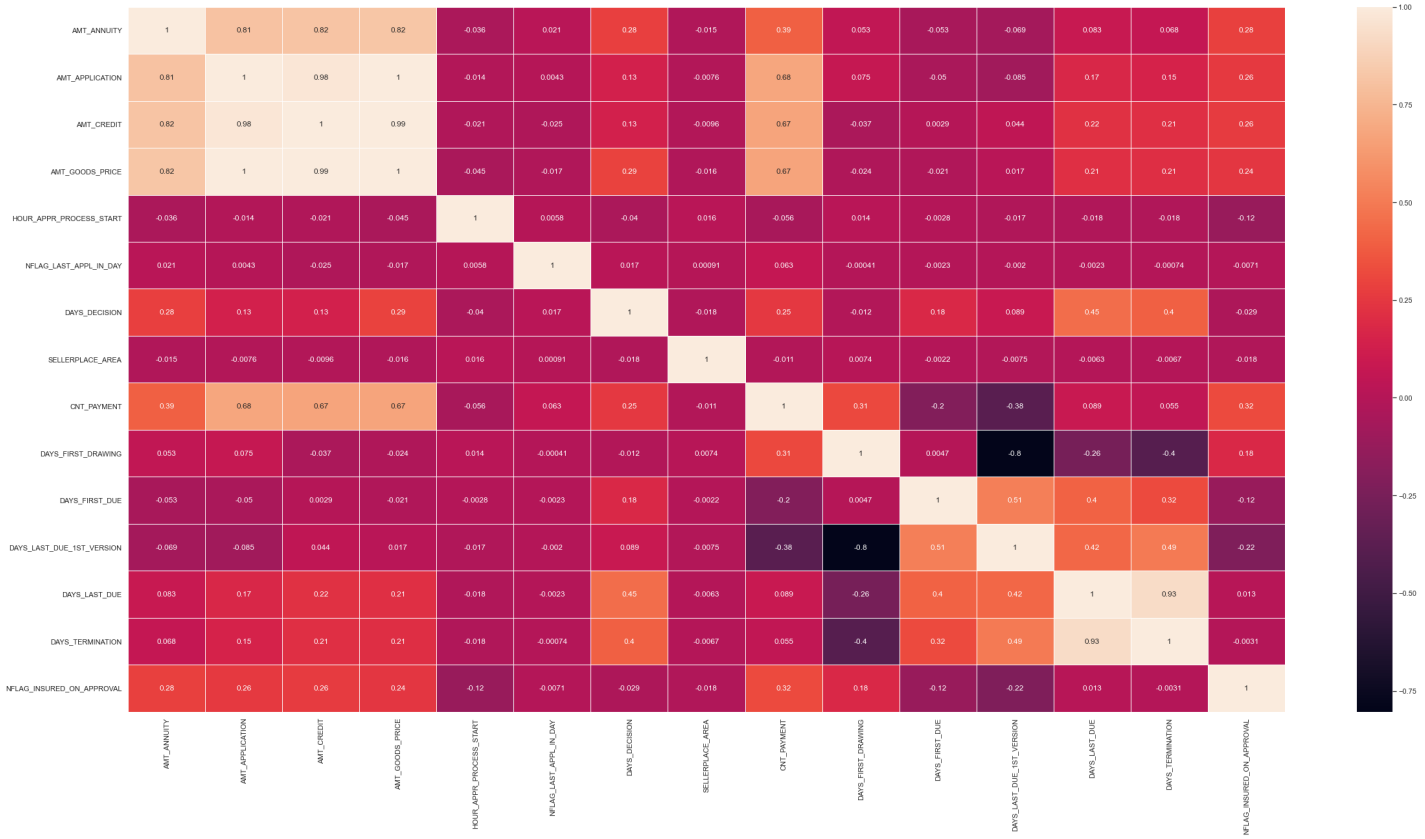


In [43]:

```
sns.set(rc = {'figure.figsize':(40,20)})
sns.heatmap(dfp1[['NAME_CONTRACT_TYPE',
 'AMT_ANNUITY',
 'AMT_APPLICATION',
 'AMT_CREDIT',
 'AMT_GOODS_PRICE',
 'WEEKDAY_APPR_PROCESS_START',
 'HOUR_APPR_PROCESS_START',
 'FLAG_LAST_APPL_PER_CONTRACT',
 'NFLAG_LAST_APPL_IN_DAY',
 'NAME_CASH_LOAN_PURPOSE',
 'NAME_CONTRACT_STATUS',
 'DAYS_DECISION',
 'NAME_PAYMENT_TYPE',
 'CODE_REJECT_REASON',
 'NAME_TYPE_SUITE',
 'NAME_CLIENT_TYPE',
 'NAME_GOODS_CATEGORY',
 'NAME_PORTFOLIO',
 'NAME_PRODUCT_TYPE',
 'CHANNEL_TYPE',
 'SELLERPLACE_AREA',
 'NAME_SELLER_INDUSTRY',
 ',
```

```
'NAME_YIELD_GROUP',
'PRODUCT_COMBINATION',
'DAYS_FIRST_DRAWING',
'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
'DAYS_LAST_DUE',
'DAYS_TERMINATION',
'NFLAG_INSURED_ON_APPROVAL']].corr(),annot=True, linewidths=.5)
```

Out[43]: <AxesSubplot:>



Top correlated variables

AMT_ANNUITY

AMT_APPLICATION

AMT_CREDIT

AMT_GOODS_PRICE

DAYS_DECISION

CNT_PAYMENT

DAYS_FIRST_DUE

DAYS_LAST_DUE

DAYS_TERMINATION

DAYS_FIRST_DRAWING

In [ ]:

In [ ]:

In [225…
```
dfp=dfp.drop(dfp[dfp['NAME_CASH_LOAN_PURPOSE']=='XNA'].index)
dfp=dfp.drop(dfp[dfp['NAME_CASH_LOAN_PURPOSE']=='XAP'].index)
```

In [ ]:
```
```

In [226…
```
merged=pd.merge(df,dfp, on='SK_ID_CURR',how='inner')
```

In [227…
```
merged.head()
```

Out[227…

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	C
0	100034	0	Revolving loans	M	N	Y	
1	100035	0	Cash loans	F	N	Y	
2	100039	0	Cash loans	M	Y	N	
3	100046	0	Revolving loans	M	Y	Y	
4	100046	0	Revolving loans	M	Y	Y	

5 rows × 158 columns

In [207…
```
merged.columns.to_list()
```

Out[207…
```
['SK_ID_CURR',
 'TARGET',
 'NAME_CONTRACT_TYPE_x',
 'CODE_GENDER',
 'FLAG_OWN_CAR',
 'FLAG_OWN_REALTY',
 'CNT_CHILDREN',
 'AMT_INCOME_TOTAL',
 'AMT_CREDIT_x',
 'AMT_ANNUITY_x',
 'AMT_GOODS_PRICE_x',
 'NAME_TYPE_SUITE_x',
 'NAME_INCOME_TYPE',
 'NAME_EDUCATION_TYPE',
 'NAME_FAMILY_STATUS',
 'NAME_HOUSING_TYPE',
 'REGION_POPULATION_RELATIVE',
 'DAYS_BIRTH',
 'DAYS_EMPLOYED',
 'DAYS_REGISTRATION',
 'DAYS_ID_PUBLISH',
 'OWN_CAR_AGE',
 'FLAG_MOBIL',
 'FLAG_EMP_PHONE',
 'FLAG_WORK_PHONE',
 'FLAG_CONT_MOBILE',
 'FLAG_PHONE',
 'FLAG_EMAIL',
 'OCCUPATION_TYPE',
 'CNT_FAM_MEMBERS',
 'REGION_RATING_CLIENT',
 'REGION_RATING_CLIENT_W_CITY',
 'WEEKDAY_APPR_PROCESS_START_x',
 'HOUR_APPR_PROCESS_START_x',
```

```
'REG_REGION_NOT_LIVE_REGION',
'REG_REGION_NOT_WORK_REGION',
'LIVE_REGION_NOT_WORK_REGION',
'REG_CITY_NOT_LIVE_CITY',
'REG_CITY_NOT_WORK_CITY',
'LIVE_CITY_NOT_WORK_CITY',
'ORGANIZATION_TYPE',
'EXT_SOURCE_1',
'EXT_SOURCE_2',
'EXT_SOURCE_3',
'APARTMENTS_AVG',
'BASEMENTAREA_AVG',
'YEARS_BEGINEXPLUATATION_AVG',
'YEARS_BUILD_AVG',
'COMMONAREA_AVG',
'ELEVATORS_AVG',
'ENTRANCES_AVG',
'FLOORSMAX_AVG',
'FLOORSMIN_AVG',
'LANDAREA_AVG',
'LIVINGAPARTMENTS_AVG',
'LIVINGAREA_AVG',
'NONLIVINGAPARTMENTS_AVG',
'NONLIVINGAREA_AVG',
'APARTMENTS_MODE',
'BASEMENTAREA_MODE',
'YEARS_BEGINEXPLUATATION_MODE',
'YEARS_BUILD_MODE',
'COMMONAREA_MODE',
'ELEVATORS_MODE',
'ENTRANCES_MODE',
'FLOORSMAX_MODE',
'FLOORSMIN_MODE',
'LANDAREA_MODE',
'LIVINGAPARTMENTS_MODE',
'LIVINGAREA_MODE',
'NONLIVINGAPARTMENTS_MODE',
'NONLIVINGAREA_MODE',
'APARTMENTS_MEDI',
'BASEMENTAREA_MEDI',
'YEARS_BEGINEXPLUATATION_MEDI',
'YEARS_BUILD_MEDI',
'COMMONAREA_MEDI',
'ELEVATORS_MEDI',
'ENTRANCES_MEDI',
'FLOORSMAX_MEDI',
'FLOORSMIN_MEDI',
'LANDAREA_MEDI',
'LIVINGAPARTMENTS_MEDI',
'LIVINGAREA_MEDI',
'NONLIVINGAPARTMENTS_MEDI',
'NONLIVINGAREA_MEDI',
'FONDKAPREMONT_MODE',
'HOUSETYPE_MODE',
'TOTALAREA_MODE',
'WALLSMATERIAL_MODE',
'EMERGENCYSTATE_MODE',
'OBS_30_CNT_SOCIAL_CIRCLE',
'DEF_30_CNT_SOCIAL_CIRCLE',
'OBS_60_CNT_SOCIAL_CIRCLE',
'DEF_60_CNT_SOCIAL_CIRCLE',
'DAYS_LAST_PHONE_CHANGE',
'FLAG_DOCUMENT_2',
'FLAG_DOCUMENT_3',
```

```
'FLAG_DOCUMENT_4',
'FLAG_DOCUMENT_5',
'FLAG_DOCUMENT_6',
'FLAG_DOCUMENT_7',
'FLAG_DOCUMENT_8',
'FLAG_DOCUMENT_9',
'FLAG_DOCUMENT_10',
'FLAG_DOCUMENT_11',
'FLAG_DOCUMENT_12',
'FLAG_DOCUMENT_13',
'FLAG_DOCUMENT_14',
'FLAG_DOCUMENT_15',
'FLAG_DOCUMENT_16',
'FLAG_DOCUMENT_17',
'FLAG_DOCUMENT_18',
'FLAG_DOCUMENT_19',
'FLAG_DOCUMENT_20',
'FLAG_DOCUMENT_21',
'AMT_REQ_CREDIT_BUREAU_HOUR',
'AMT_REQ_CREDIT_BUREAU_DAY',
'AMT_REQ_CREDIT_BUREAU_WEEK',
'AMT_REQ_CREDIT_BUREAU_MON',
'AMT_REQ_CREDIT_BUREAU_QRT',
'AMT_REQ_CREDIT_BUREAU_YEAR',
'SK_ID_PREV',
'NAME_CONTRACT_TYPE_y',
'AMT_ANNUITY_y',
'AMT_APPLICATION',
'AMT_CREDIT_y',
'AMT_DOWN_PAYMENT',
'AMT_GOODS_PRICE_y',
'WEEKDAY_APPR_PROCESS_START_y',
'HOUR_APPR_PROCESS_START_y',
'FLAG_LAST_APPL_PER_CONTRACT',
'NFLAG_LAST_APPL_IN_DAY',
'RATE_DOWN_PAYMENT',
'RATE_INTEREST_PRIMARY',
'RATE_INTEREST_PRIVILEGED',
'NAME_CASH_LOAN_PURPOSE',
'NAME_CONTRACT_STATUS',
'DAYS_DECISION',
'NAME_PAYMENT_TYPE',
'CODE_REJECT_REASON',
'NAME_TYPE_SUITE_y',
'NAME_CLIENT_TYPE',
'NAME_GOODS_CATEGORY',
'NAME_PORTFOLIO',
'NAME_PRODUCT_TYPE',
'CHANNEL_TYPE',
'SELLERPLACE_AREA',
'NAME_SELLER_INDUSTRY',
'CNT_PAYMENT',
'NAME_YIELD_GROUP',
'PRODUCT_COMBINATION',
'DAYS_FIRST_DRAWING',
'DAYS_FIRST_DUE',
'DAYS_LAST_DUE_1ST_VERSION',
'DAYS_LAST_DUE',
'DAYS_TERMINATION',
'NFLAG_INSURED_ON_APPROVAL']
```

In [208…

```
merged['TARGET'].value_counts(normalize=True)*100
```

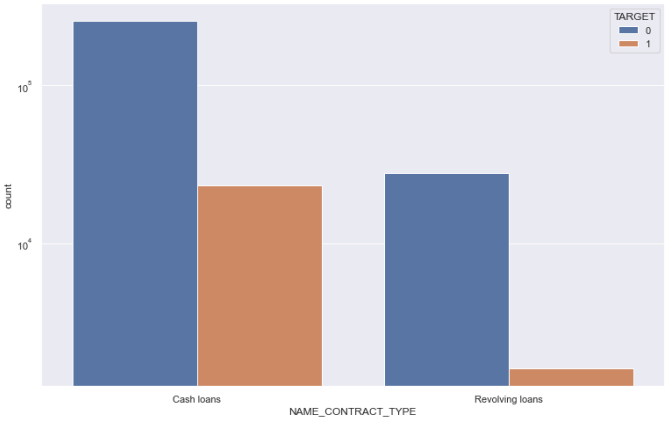
Out[208…
```
1     8.613694
Name: TARGET, dtype: float64
```

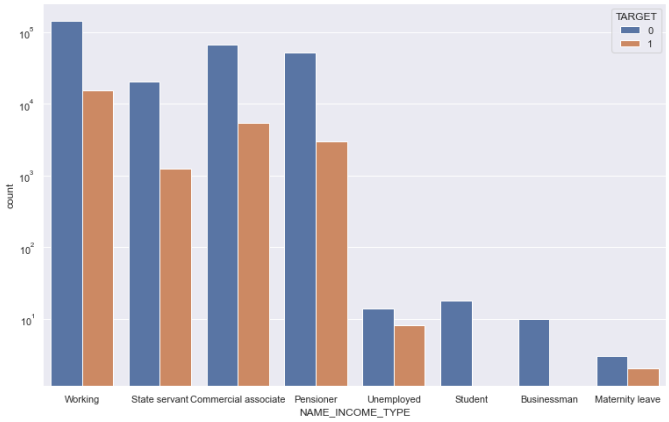
In [222…
```
dfN=df[merged.TARGET==0]
dfN1=df[merged.TARGET==1]
```

In [210…
```
plt.figure(figsize=(10,8))
sns.scatterplot(merged["AMT_APPLICATION"],merged["CNT_PAYMENT"])
```

Out[210…
```
<AxesSubplot:xlabel='AMT_APPLICATION', ylabel='CNT_PAYMENT'>
```

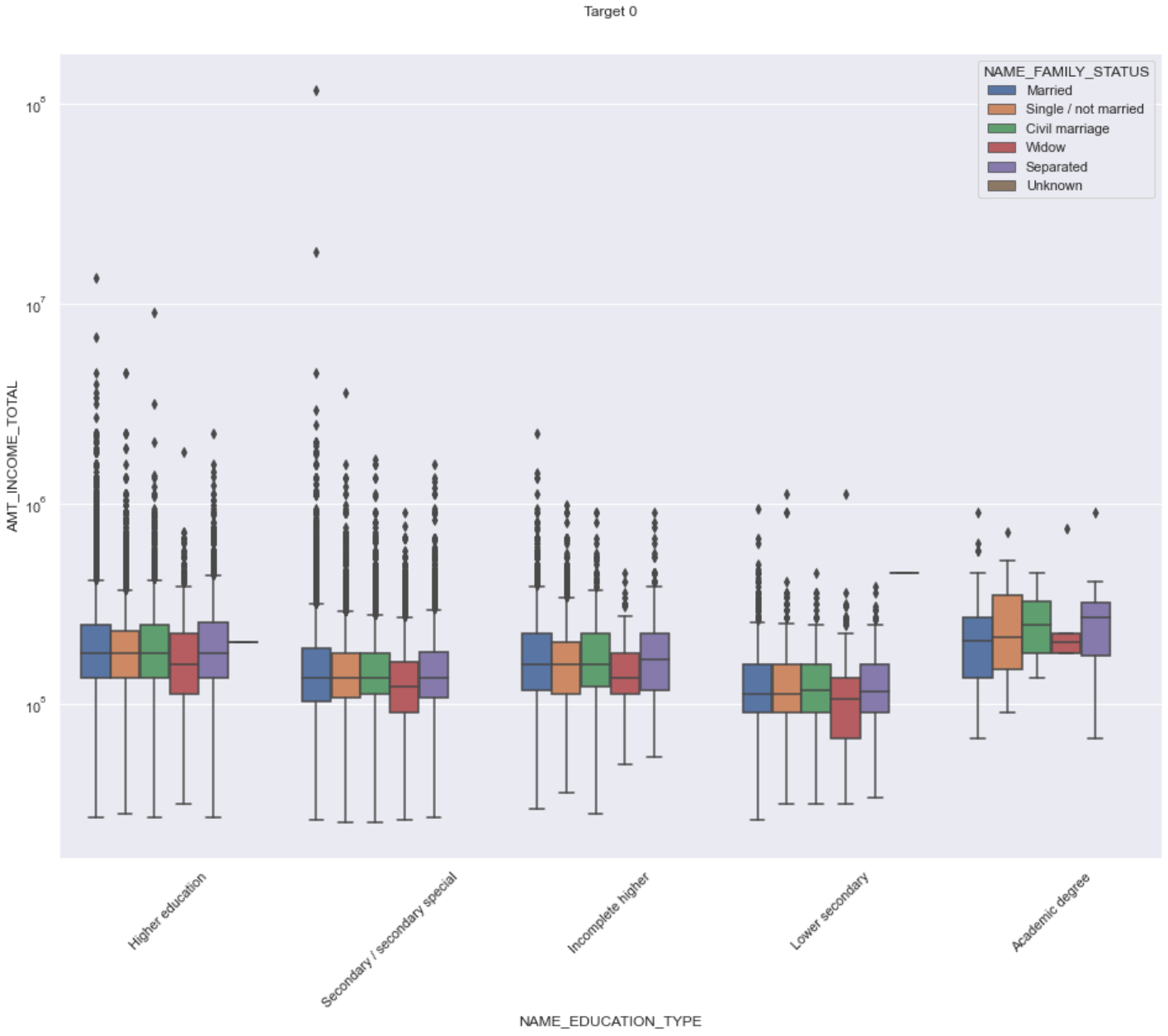
In [215…
```
cat_columns=['NAME_CONTRACT_TYPE', 'NAME_INCOME_TYPE']

plt.figure(figsize=(28,8))
for i in (enumerate(cat_columns)):
    plt.subplot(len(cat_columns)//2,2,i[0]+1)
    sns.countplot(x=i[1],hue='TARGET',data=df1)
    plt.yscale('log')

plt.show()
```





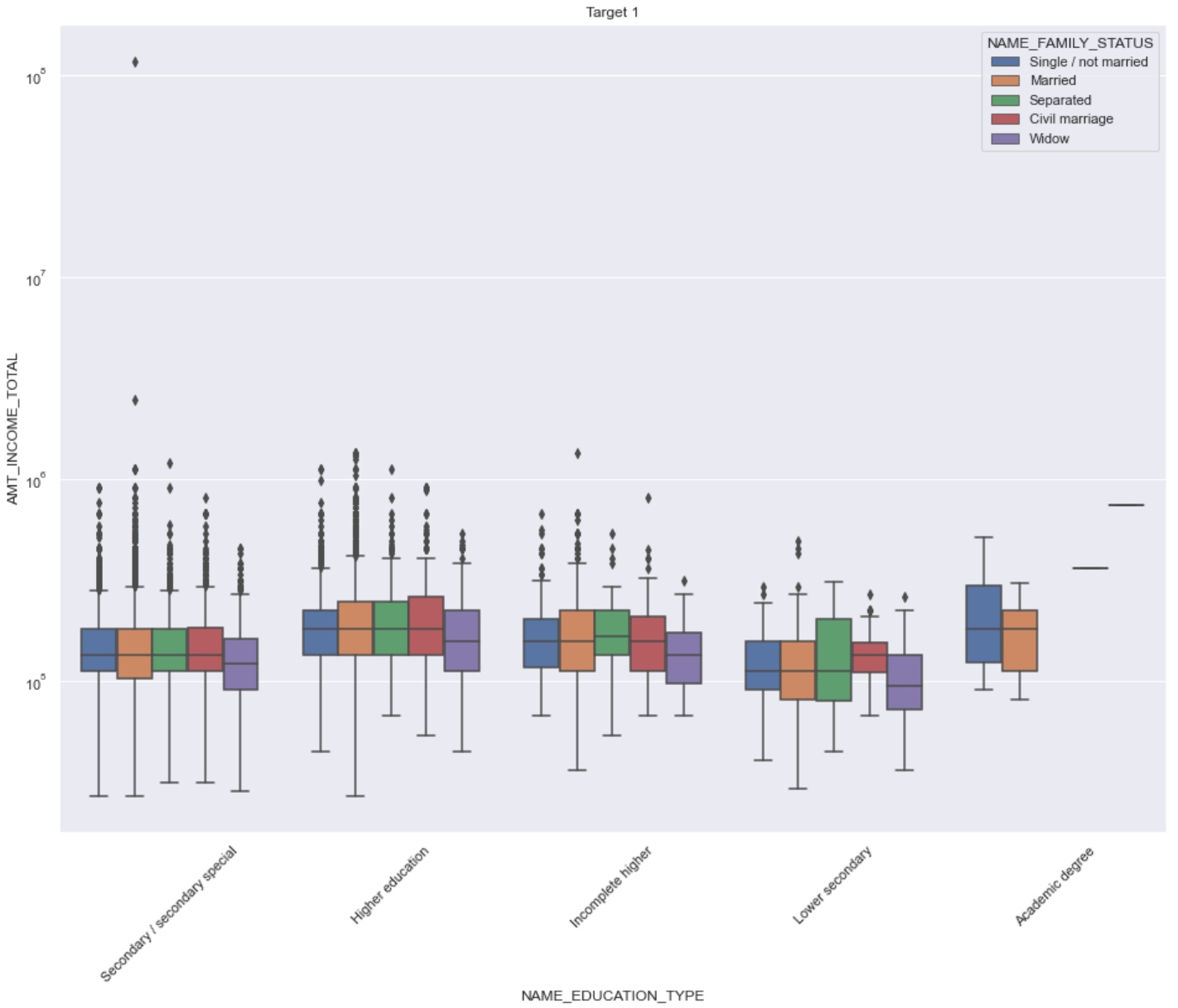
In [212…

```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data=dfN,x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL',hue='NAME_FAMILY_STATUS'
plt.title('Target 0')
plt.show()
```



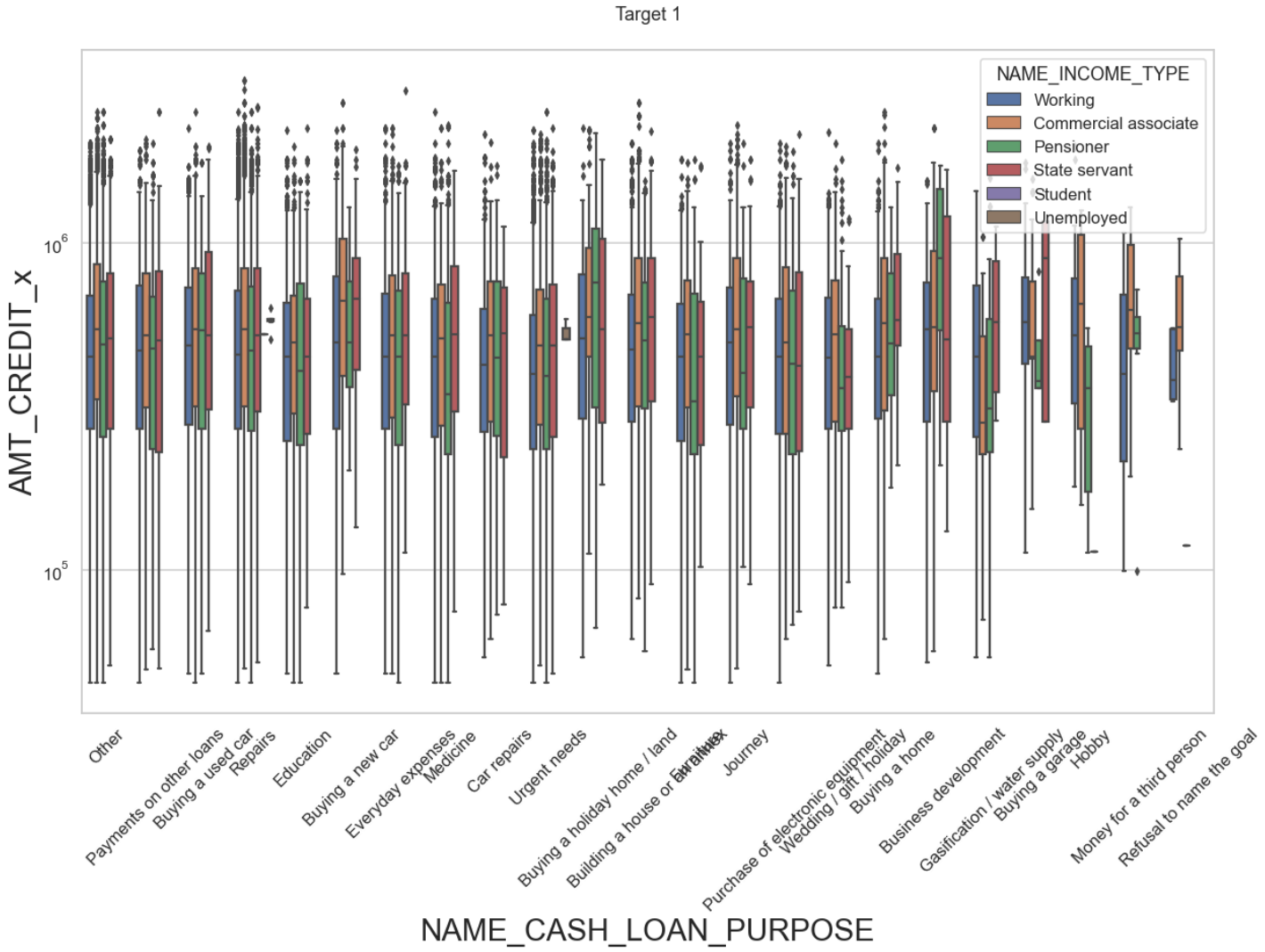
Under secondary/secondary special married clients are in the highest income group.

```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data=dfN1,x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL',hue='NAME_FAMILY_STATUS
plt.title('Target 1')
plt.show()
```



In [230…

```
plt.figure(figsize=(1,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data=merged,x='NAME_CASH_LOAN_PURPOSE',y='AMT_CREDIT_x',hue='NAME_INCOME_TYPE'
plt.title('Target 1')
plt.show()
```



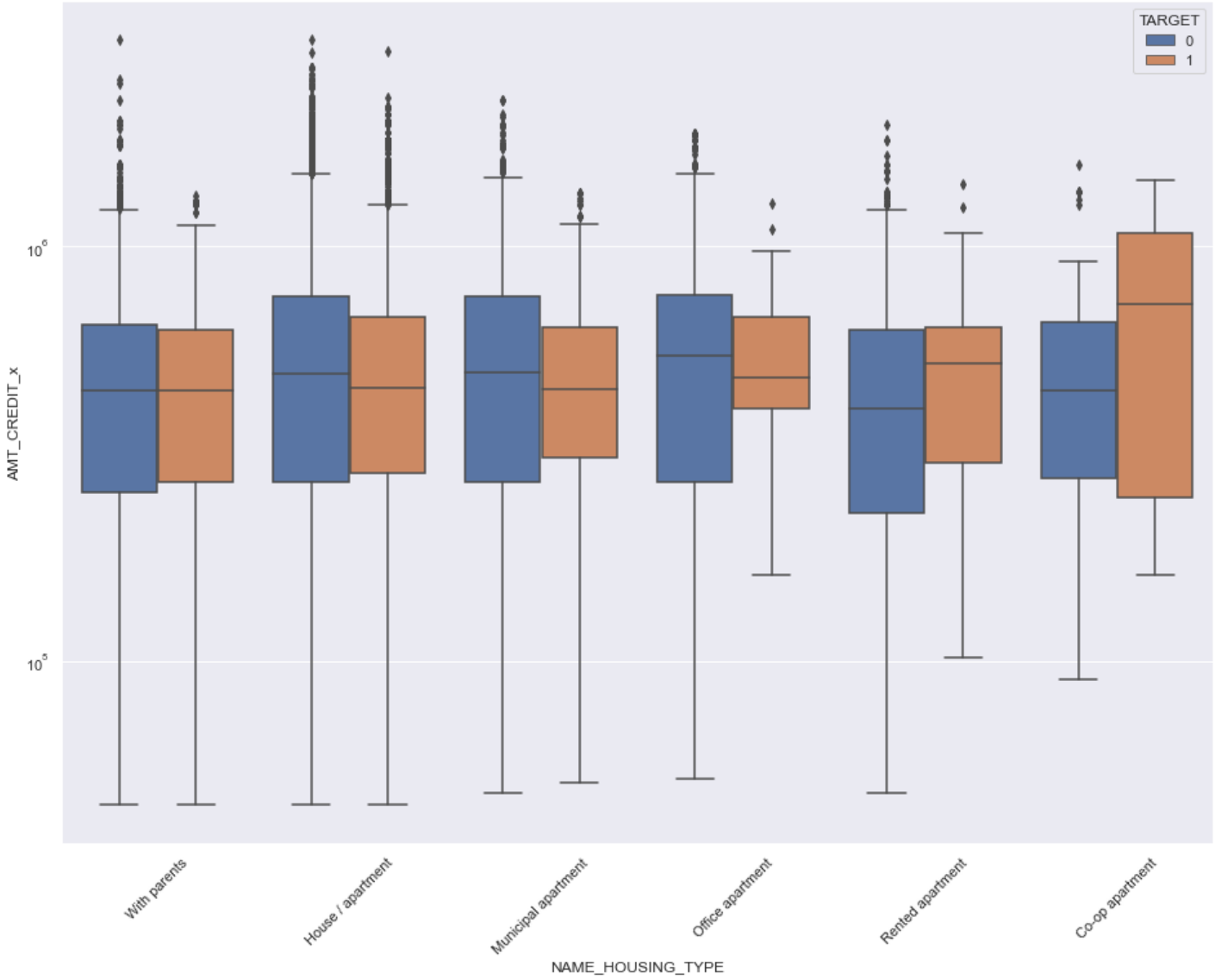
Hobby and Money for a third person are having less credit amount

Pensioners have high credit amount

The purpose for loan for buying a new car,buying a used car,buying a land,building a home is hight

In [250…

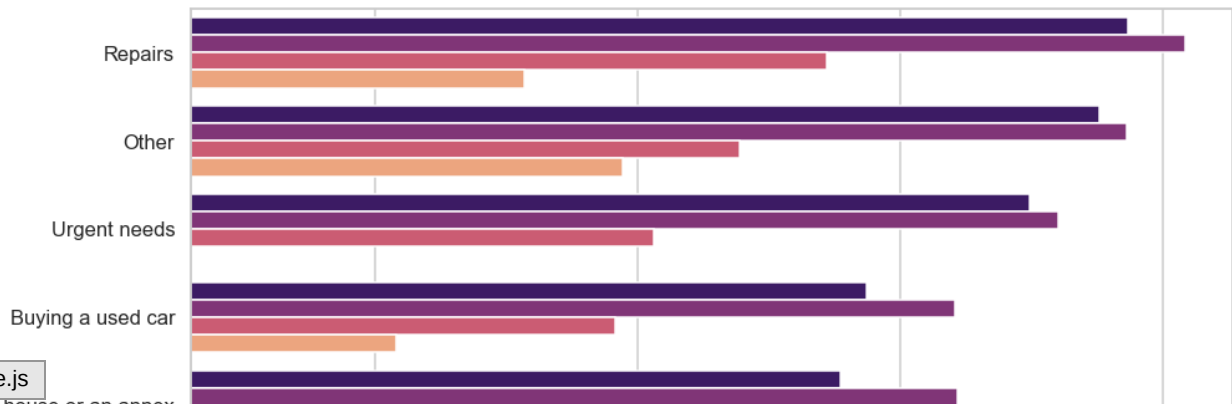
```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data=merged,x='NAME_HOUSING_TYPE',y='AMT_CREDIT_x',hue='TARGET',orient='v')
plt.show()
```

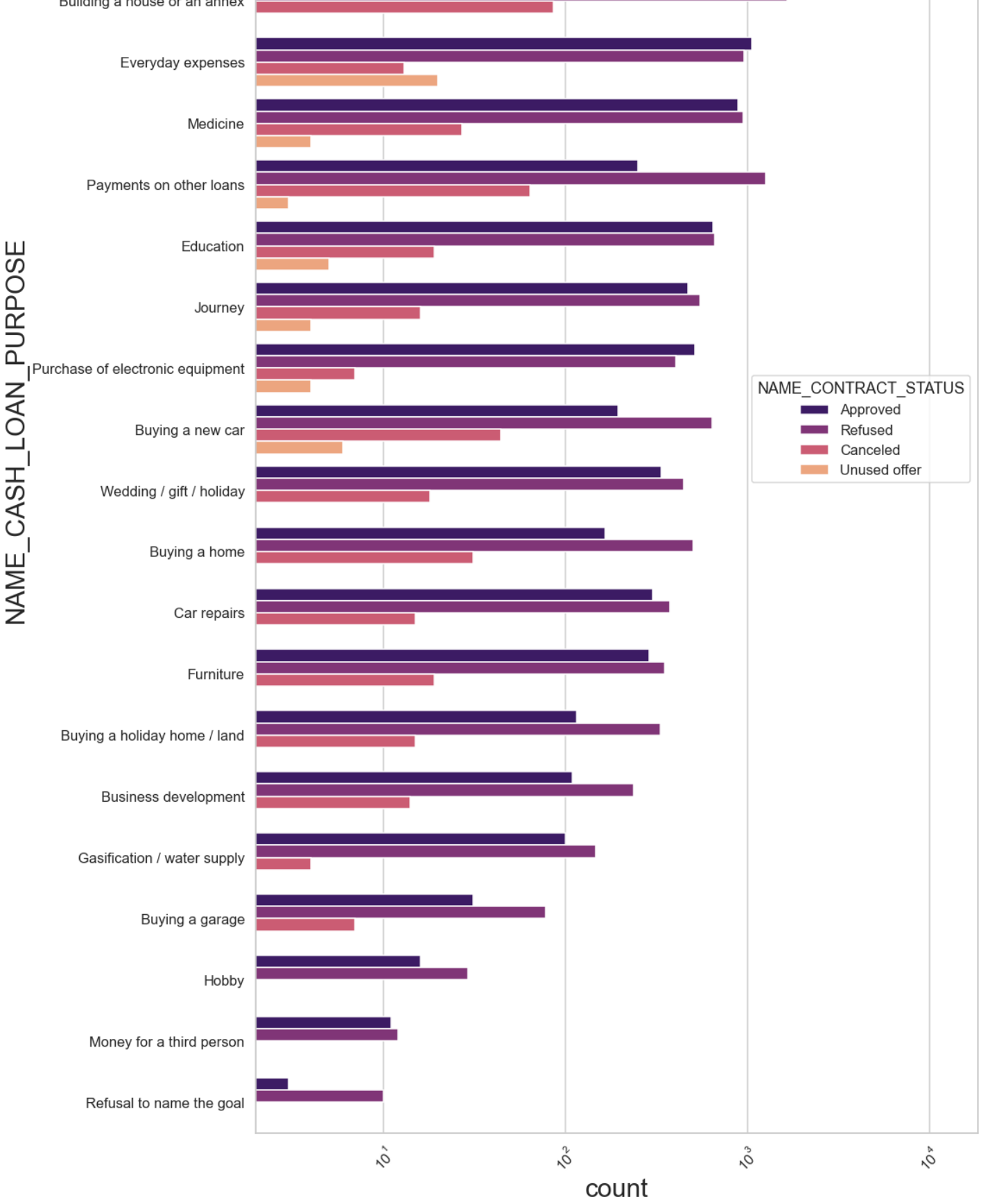


In [ ]:

In [229…

```
sns.set_style('whitegrid')
sns.set_context('talk')

plt.figure(figsize=(15,30))
plt.rcParams["axes.labelsize"]=20
plt.rcParams["axes.labelsize"]=22
plt.rcParams["axes.labelsize"]=30
plt.xticks(rotation=45)
plt.xscale('log')
ax=sns.countplot(data=merged,y='NAME_CASH_LOAN_PURPOSE',
                 order=merged['NAME_CASH_LOAN_PURPOSE'].value_counts().index,hue='NAME_CON
```





For buying a house,buying a used care,repairs loans are refused than approved.

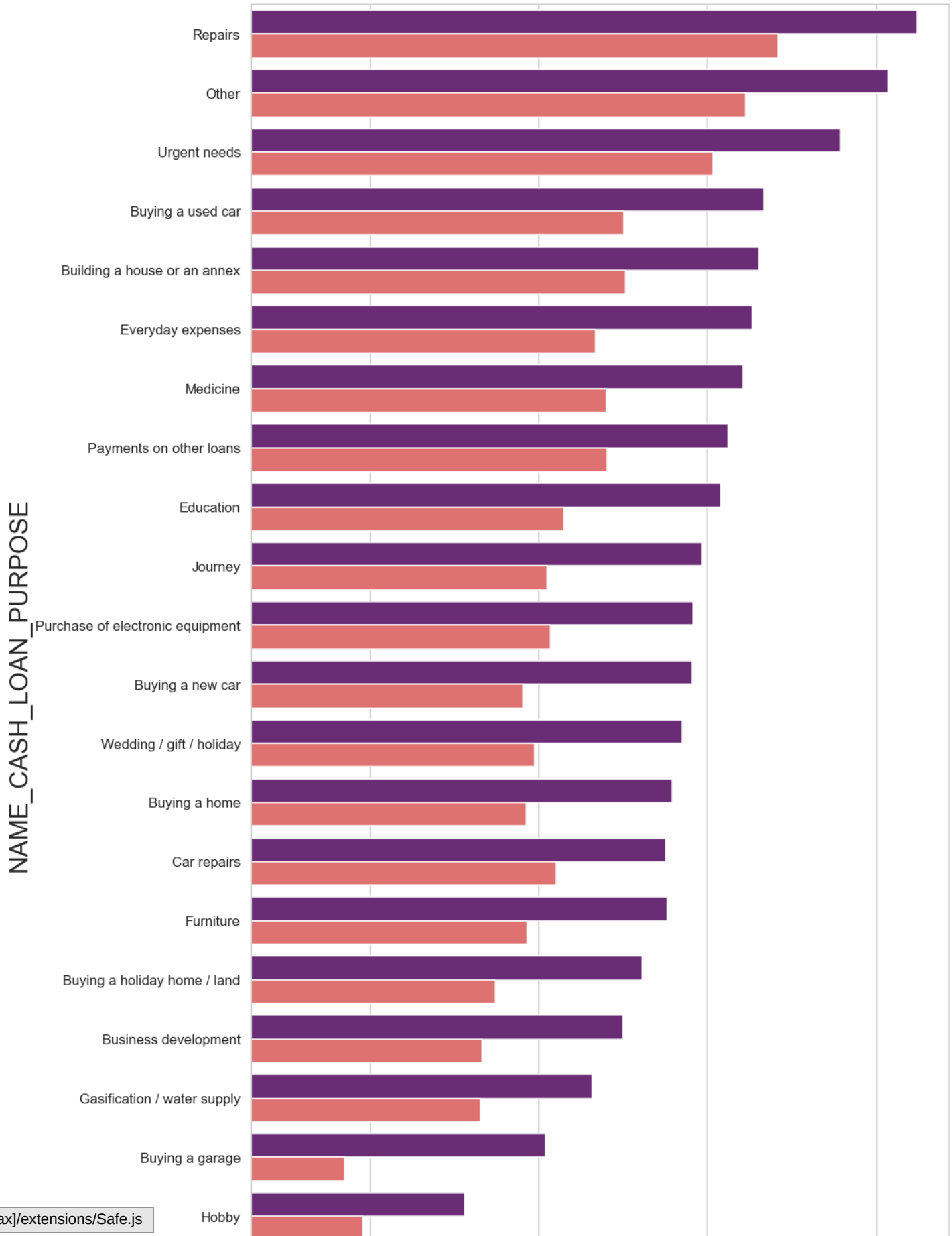
Higher rejections are in the repair category.

For education loan approved and refused are same

In [114…

```
sns.set_style('whitegrid')
sns.set_context('talk')

                gsize=(15,30))
```

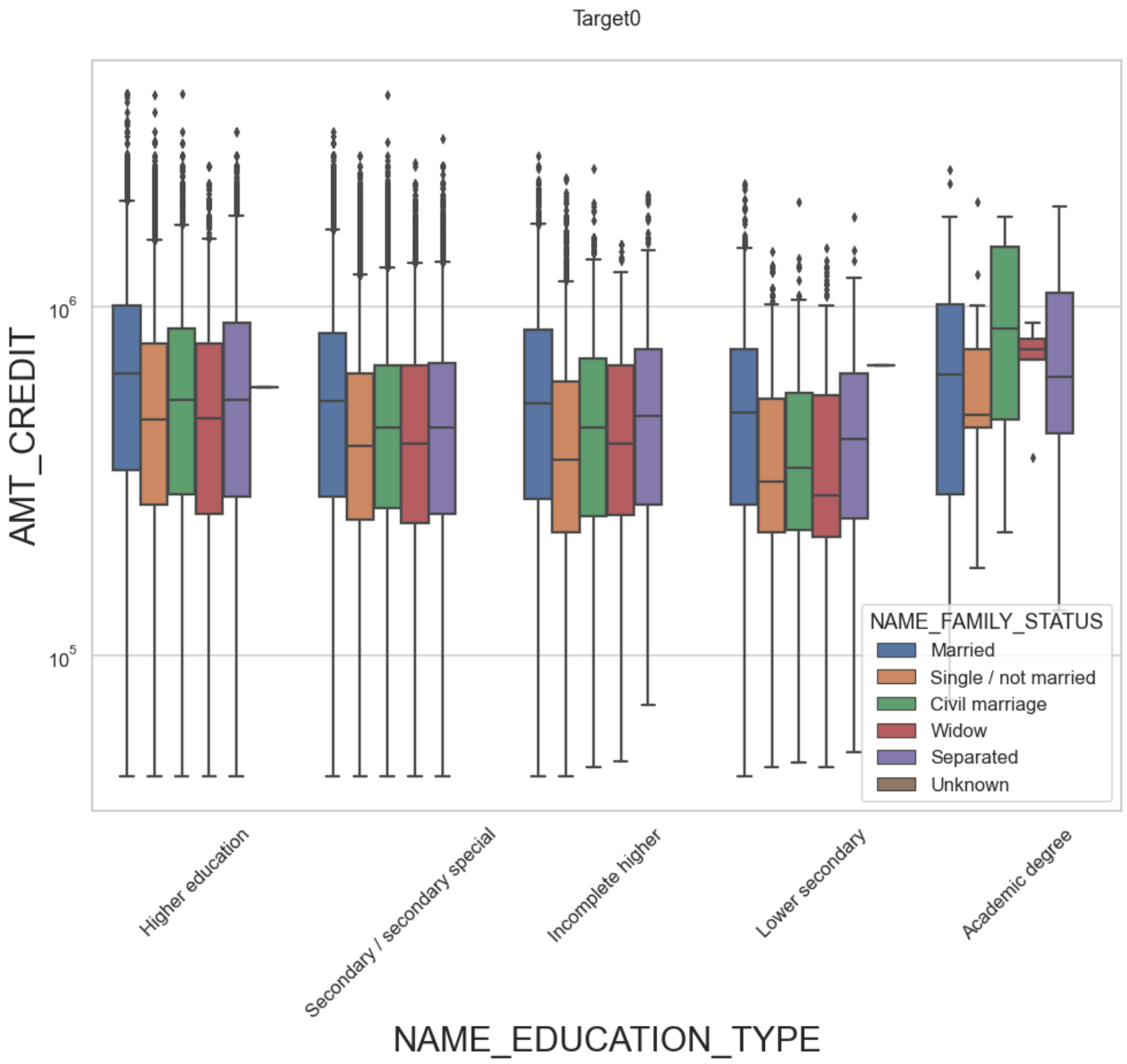
```
plt.rcParams["axes.labelsize"]=20
plt.rcParams["axes.labelsize"]=22
plt.rcParams["axes.labelsize"]=30
plt.xticks(rotation=45)
plt.xscale('log')
ax=sns.countplot(data=merged,y='NAME_CASH_LOAN_PURPOSE',
                 order=merged['NAME_CASH_LOAN_PURPOSE'].value_counts().index,hue='TARGET',
```



The purpose of the loan is highest for repairs

In [238…

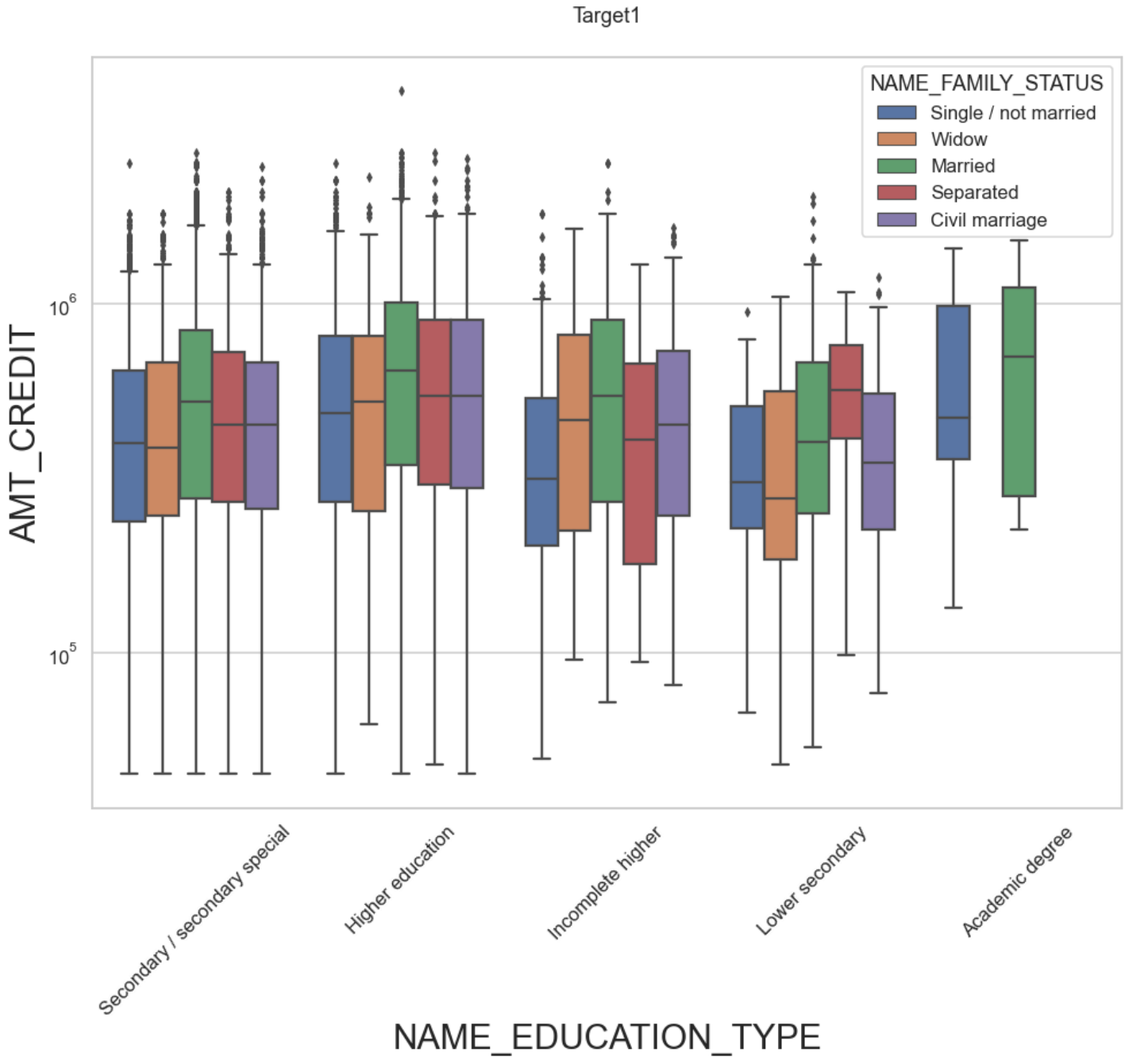
```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data=dfN,x='NAME_EDUCATION_TYPE',y='AMT_CREDIT',hue='NAME_FAMILY_STATUS',orier
plt.title('Target0')
plt.show()
```



For non defaulters married clients from higher education is having higher credit amount

In [239…

```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data=dfN1,x='NAME_EDUCATION_TYPE',y='AMT_CREDIT',hue='NAME_FAMILY_STATUS',orie
plt.title('Target1')
plt.show()
```



For defaulters married clients from secondary/secondary special is having higher credit amount

Conclusion

Banks should focus on repairs as they are having highest number of non defaulters.

Banks should not focus on hobby or money to third person as they are having least number of non defaulters.

Banks should not consider education type alone as criteria for loan.

Banks should focus on approving loans for buying a used car as the history says it has fair number of non defaulters.

In income type banks should focus on commercial associate as it has good history of non defaulters.

In income type banks should not be focusing on maternity leave as it has least number of non defaulters.

In [ ]: