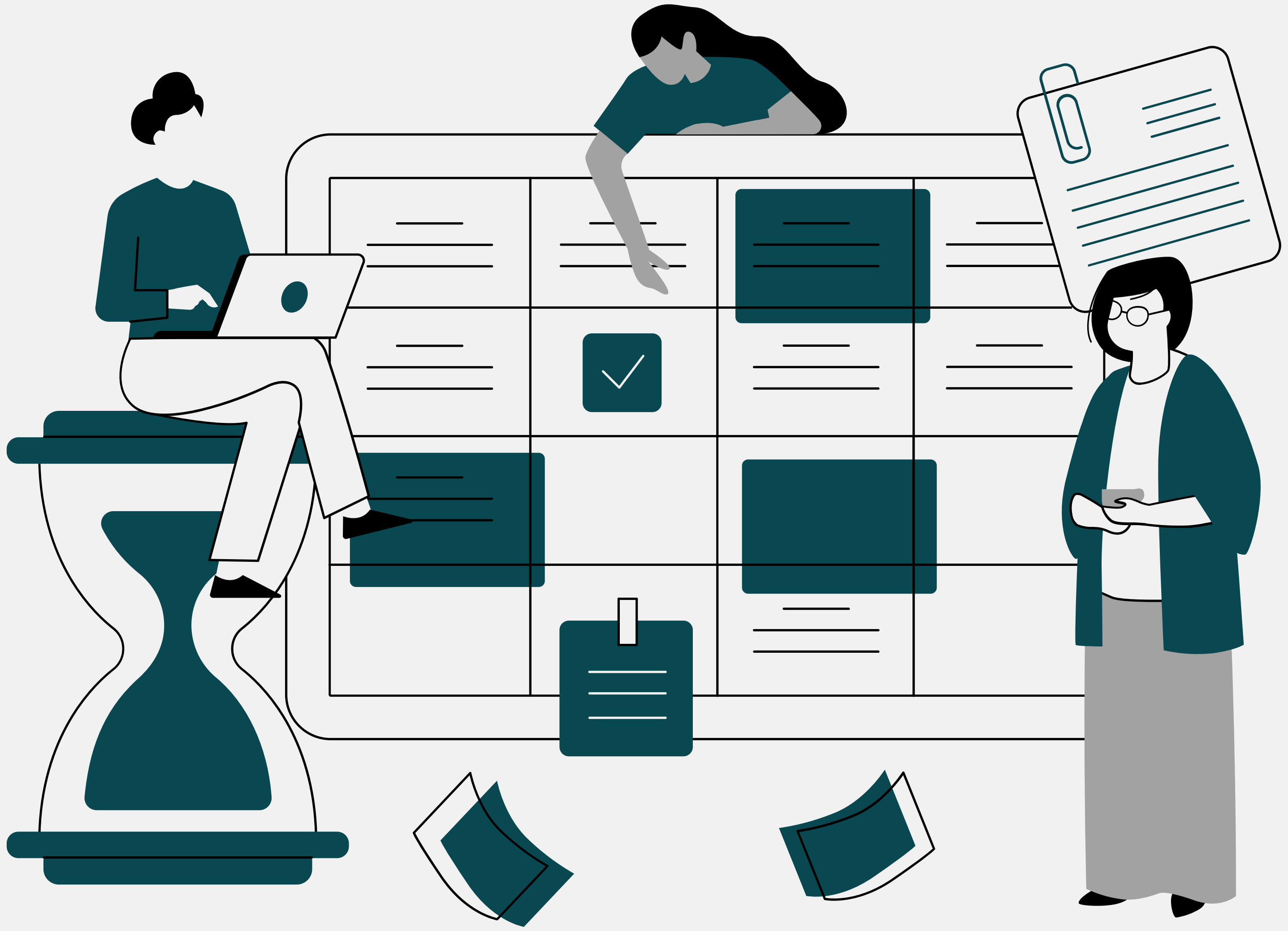
Correlation between people who receive loans

Prepared by
Bektursunov Adil
Khassen Meiirman
Tleubergen Zhuldyzay

Report for the Endterm

Content

A brief look at what we will discuss on this report



Annotation

People need the loan, they get permission to get loans from trustable sources such as Bank. In this case, Home Credit. They provide the data set of loan applicants with their informations about what type of document they have, about their properties, or even families. There are 307511 number of observations with 122 features. We need the programming tools to visualize and manipulate with such big data, so we use here Python with libraries pandas, numpy, matplotlib and seaborn to do all that things.

Data Gathering

```
application_train = pd.read_csv('D:/SDU/Python/Data Science/HomeCredit Competition/home-credit-default-risk/application_train.cs
```

Data of applications with target value of loan repayment look like the following:

```
application_train
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_
0	100002	1	Cash loans	M	N	Y	0	202500.0	
1	100003	0	Cash loans	F	N	N	0	270000.0	1
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	
4	100007	0	Cash loans	M	N	Y	0	121500.0	
...	...	...	...	...	...	...	...	...	
307506	456251	0	Cash loans	M	N	N	0	157500.0	
307507	456252	0	Cash loans	F	N	Y	0	72000.0	
307508	456253	0	Cash loans	F	N	Y	0	153000.0	
307509	456254	1	Cash loans	F	N	Y	0	171000.0	
307510	456255	0	Cash loans	F	N	N	0	157500.0	

307511 rows × 122 columns

Data Cleaning

```
for i in application_train.columns:
    print(i)
```

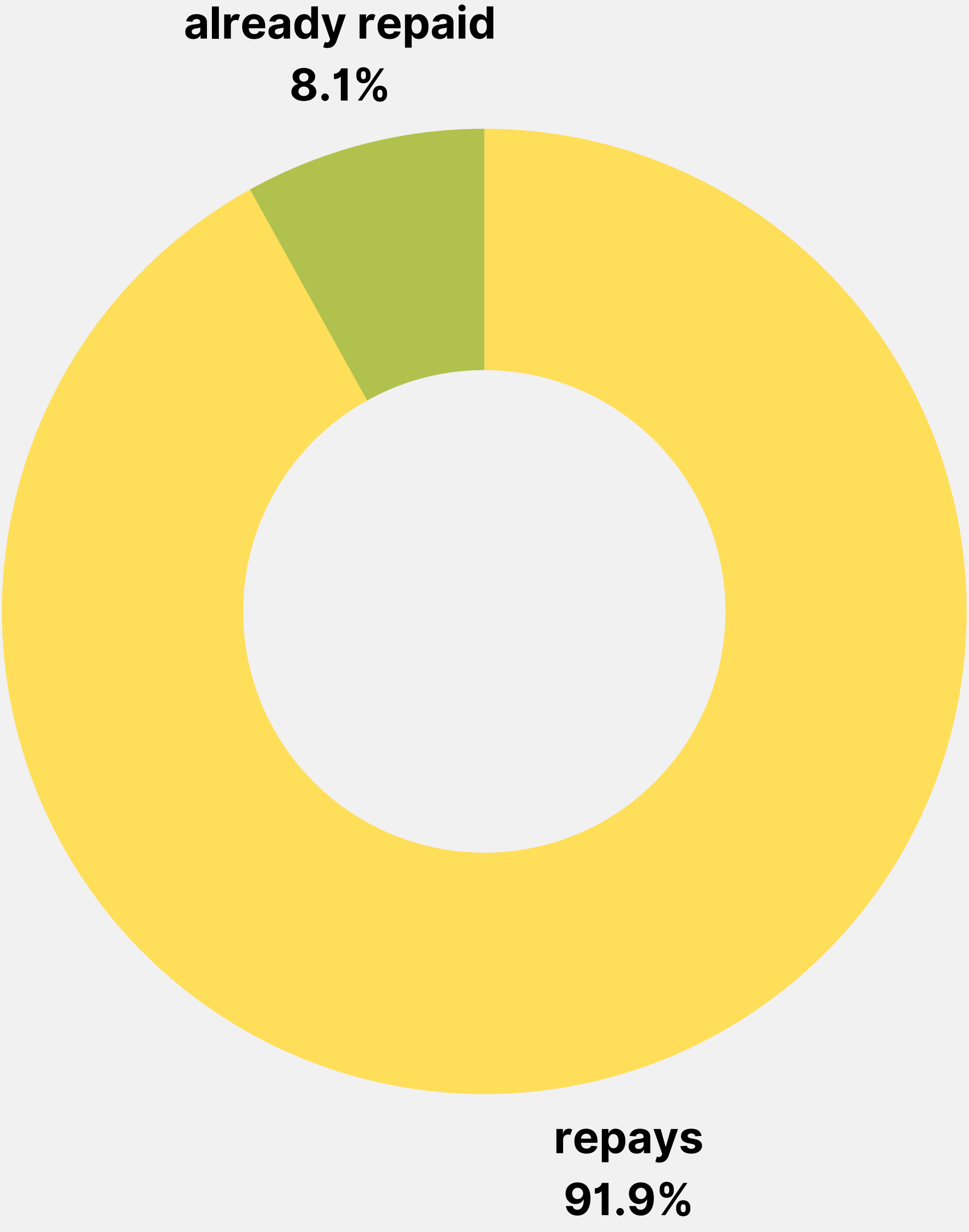
```
SK_ID_CURR
TARGET
NAME_CONTRACT_TYPE
CODE_GENDER
FLAG_OWN_CAR
FLAG_OWN_REALTY
CNT_CHILDREN
AMT_INCOME_TOTAL
AMT_CREDIT
AMT_ANNUITY
AMT_GOODS_PRICE
NAME_TYPE_SUITE
NAME_INCOME_TYPE
NAME_EDUCATION_TYPE
NAME_FAMILY_STATUS
NAME_HOUSING_TYPE
REGION_POPULATION_RELATIVE
DAYS_BIRTH
DAYS_EMPLOYED
DAYS_REGISTRATION
```

Loan repayment

Number of ppl, who repays their loan = 282686 (91.9271%)

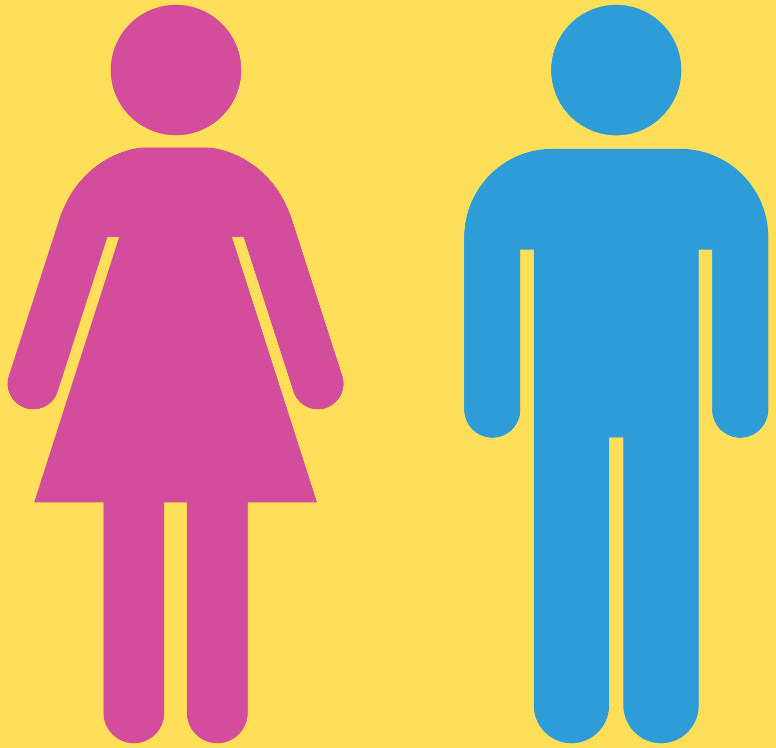
Number of ppl, who already repaid their loans = 24825 (8.0729%)

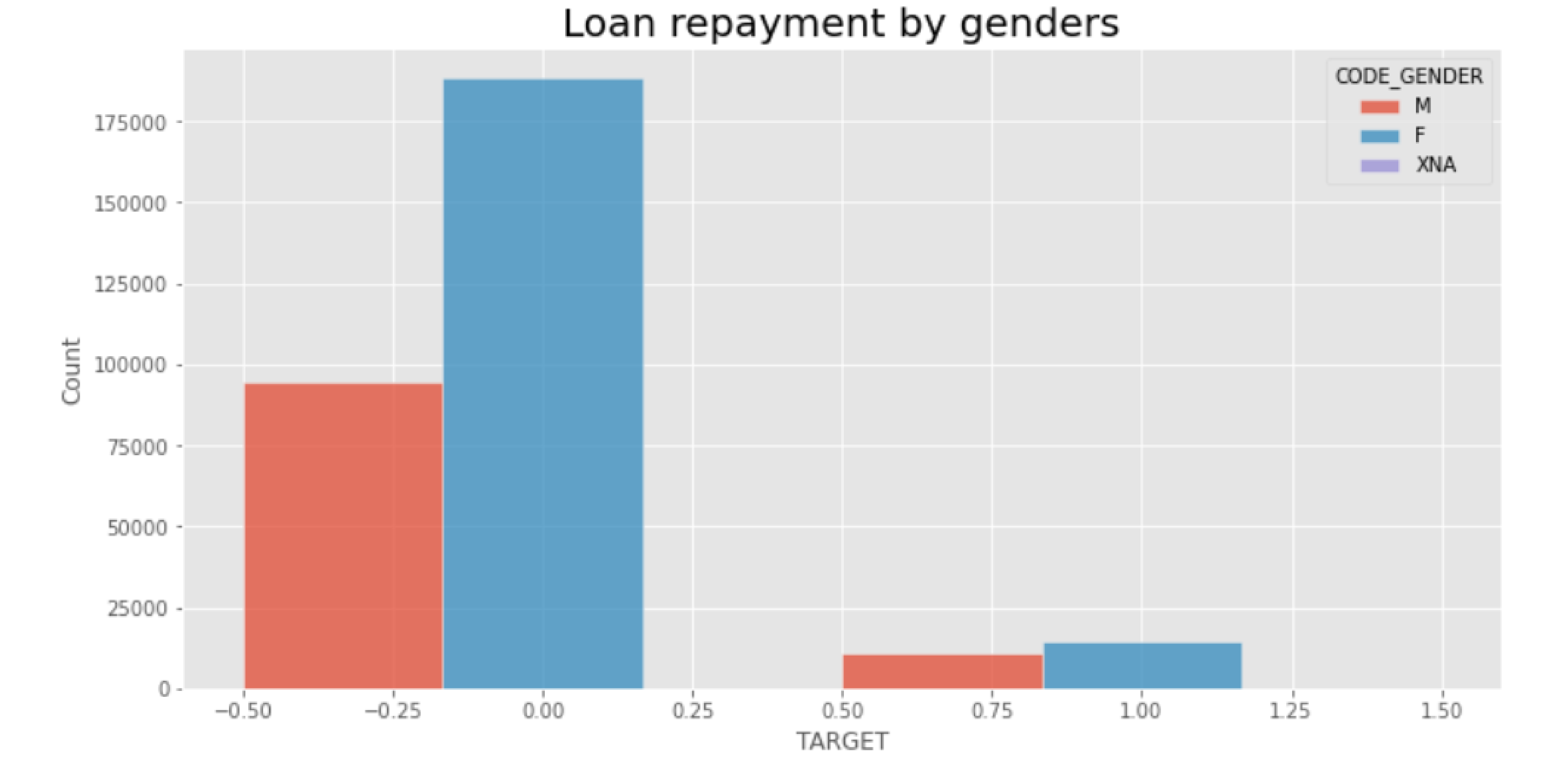
from dataset



We see high imbalance of target value, where 91.93% of loans were repayed, whenever 8.07% not.

CODE_GENDER	TARGET	
F	0	188278
	1	14170
M	0	94404
	1	10655
XNA	0	4





1

By looking at the graph it seems that females repay loan as twice as men. And we tested this hypothesis

Genders:

F	202448
M	105059
XNA	4

Genders by percentage:

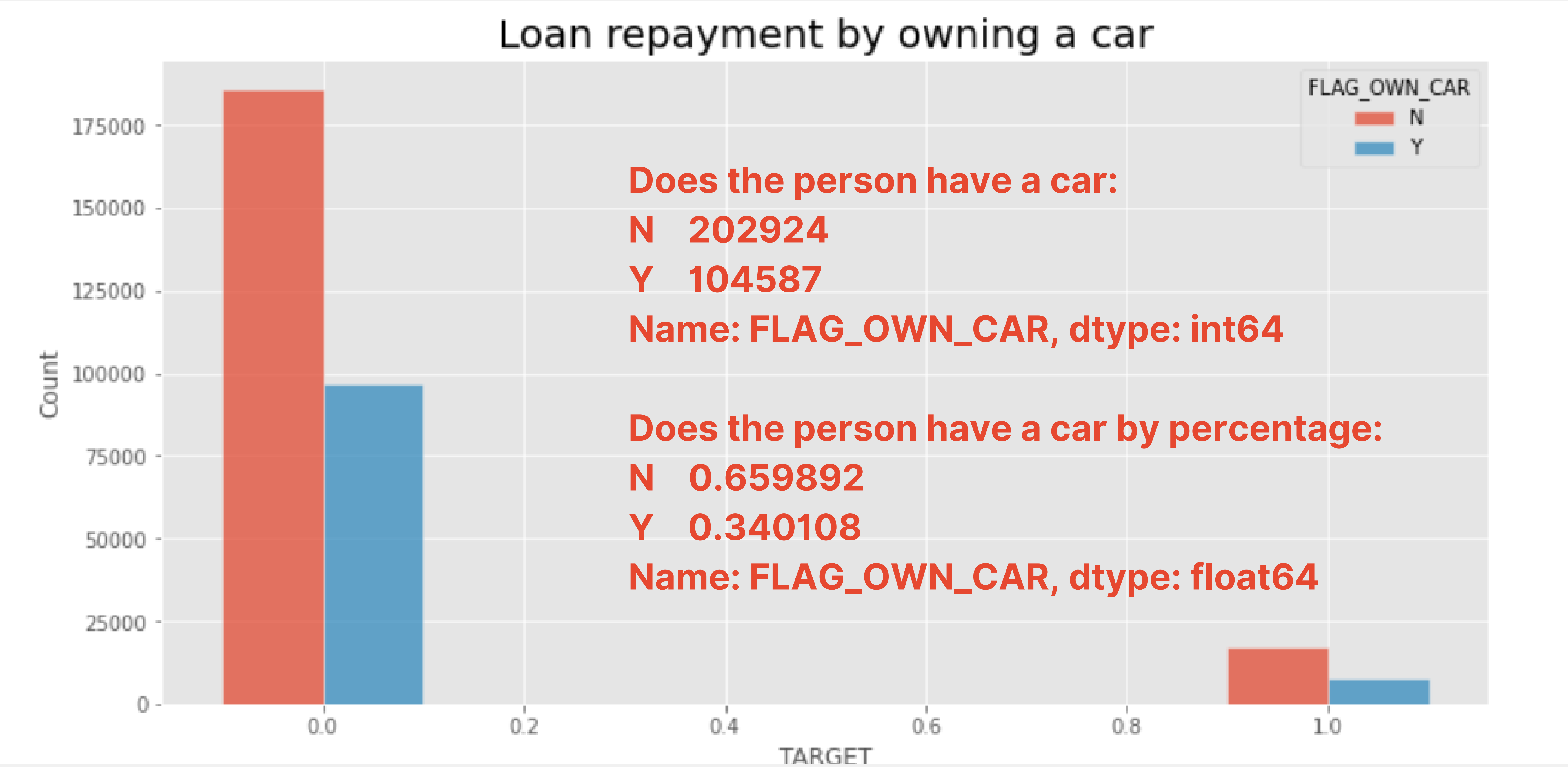
F	0.658344
M	0.341643
XNA	0.000013

Now, we can easily see that females repay loan as twice as men, because females are twice more than men

2

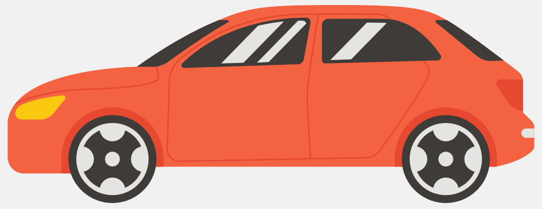
Loan payment by owning car

```
FLAG_OWN_CAR  TARGET
N             0         185675
              1          17249
Y             0          97011
              1           7576
Name: TARGET, dtype: int64
```



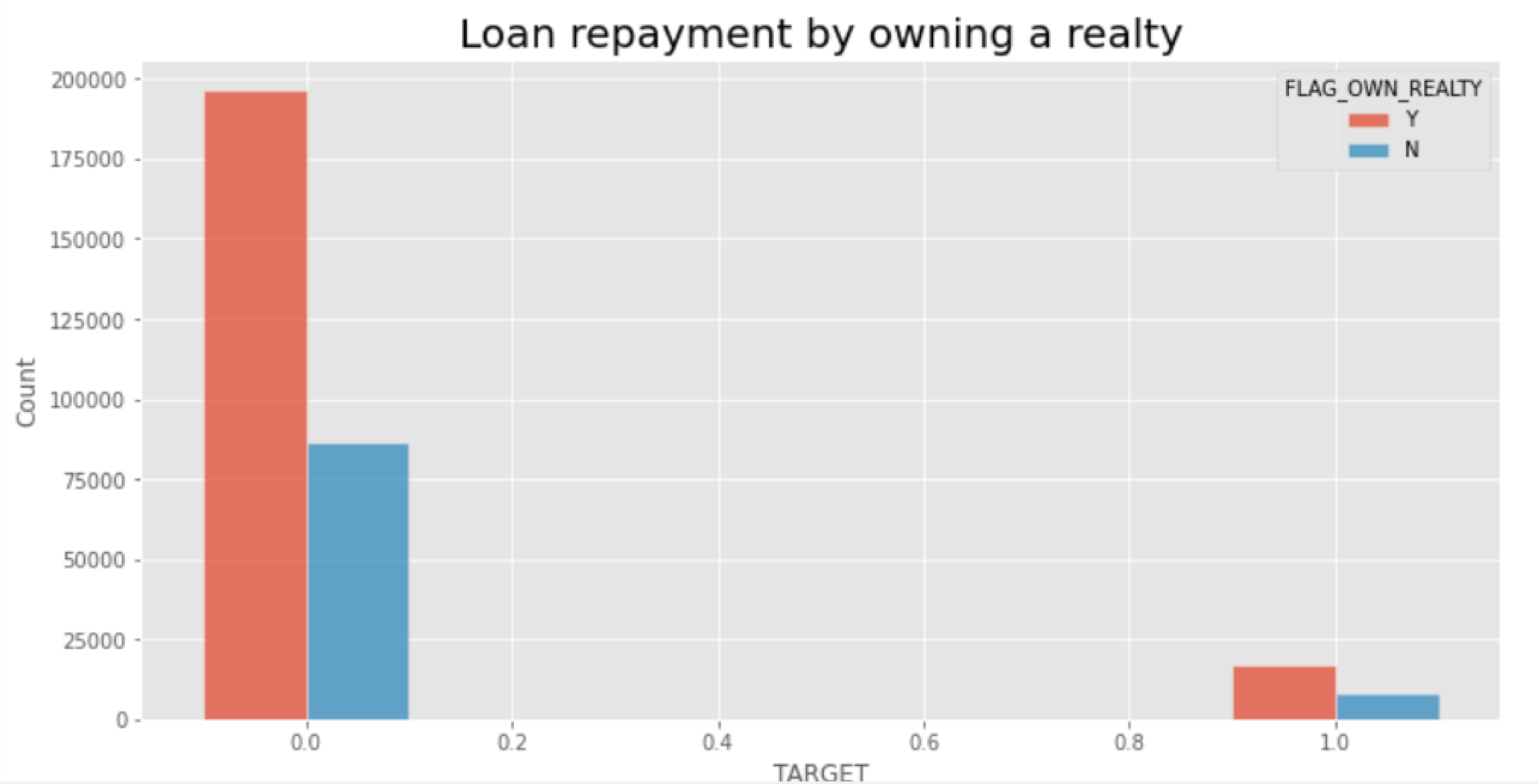
Does the person have a car:
N   202924
Y   104587
Name: FLAG_OWN_CAR, dtype: int64

Does the person have a car by percentage:
N   0.659892
Y   0.340108
Name: FLAG_OWN_CAR, dtype: float64

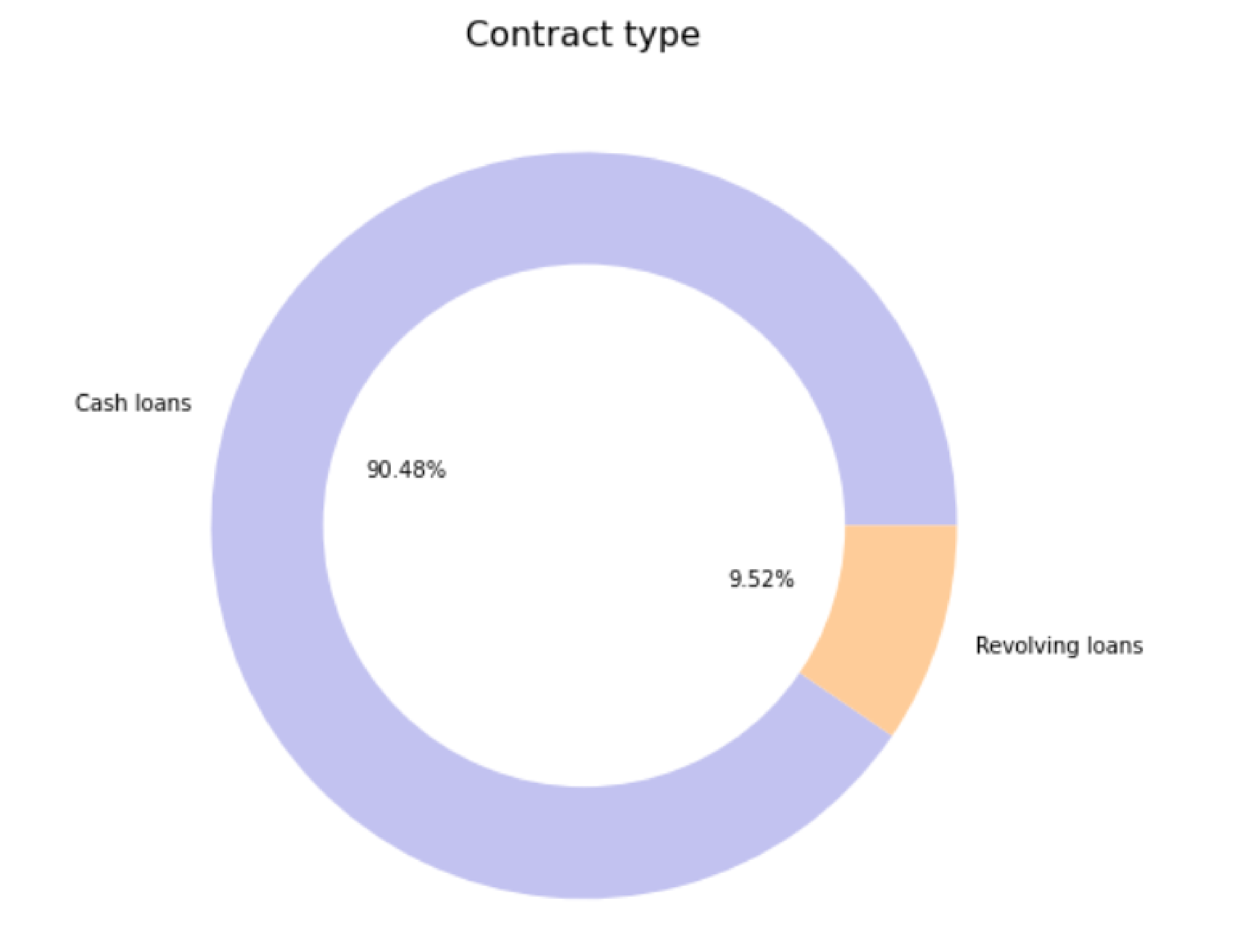


Loan payment by owning realty

```
FLAG_OWN_REALTY  TARGET
N                0          86357
                 1           7842
Y                0         196329
                 1          16983
Name: TARGET, dtype: int64
```



Y 213312
N 94199

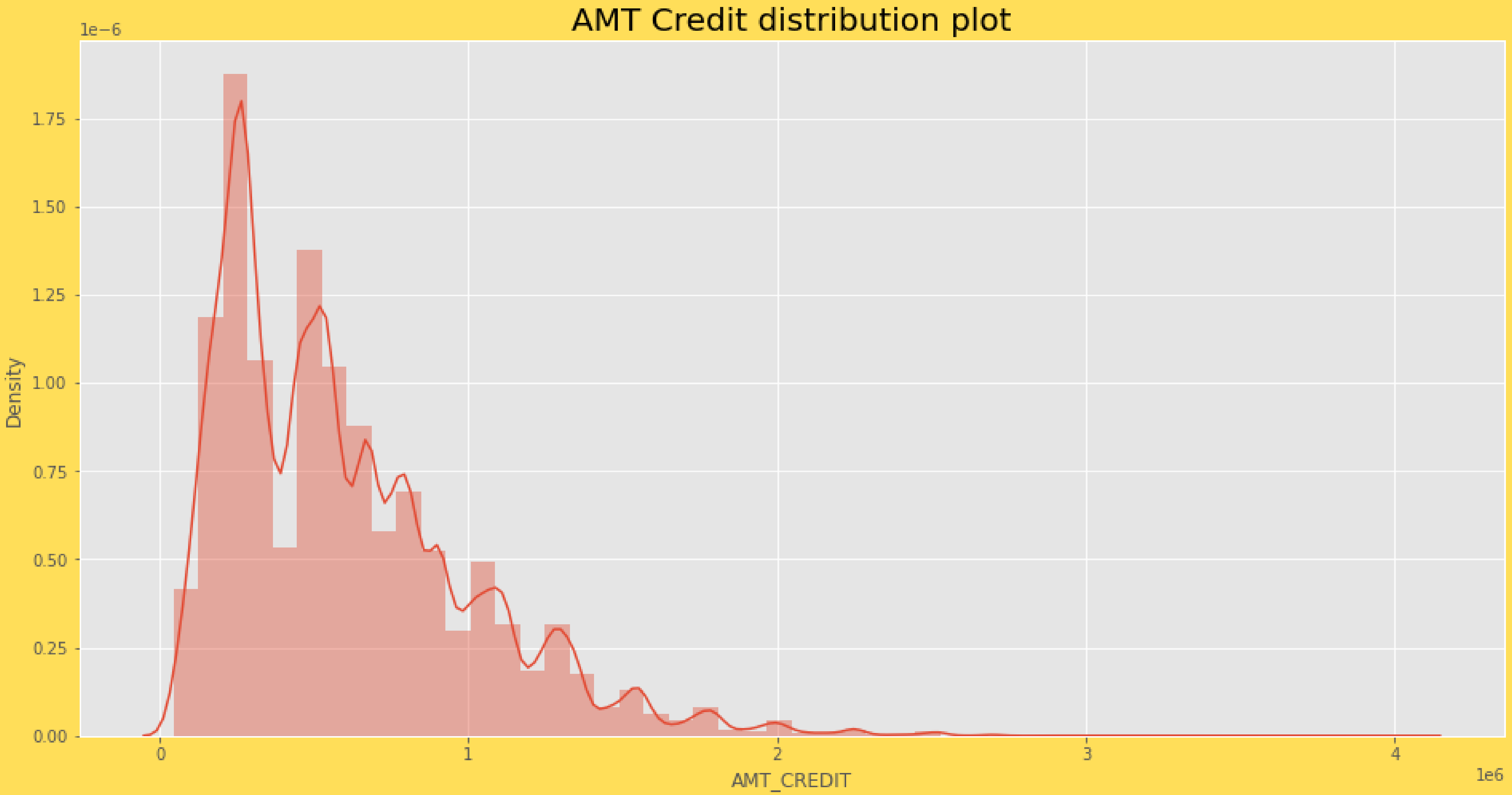


The most of the loans are cash loans with 90% frequency, while other 10% are revolving loans

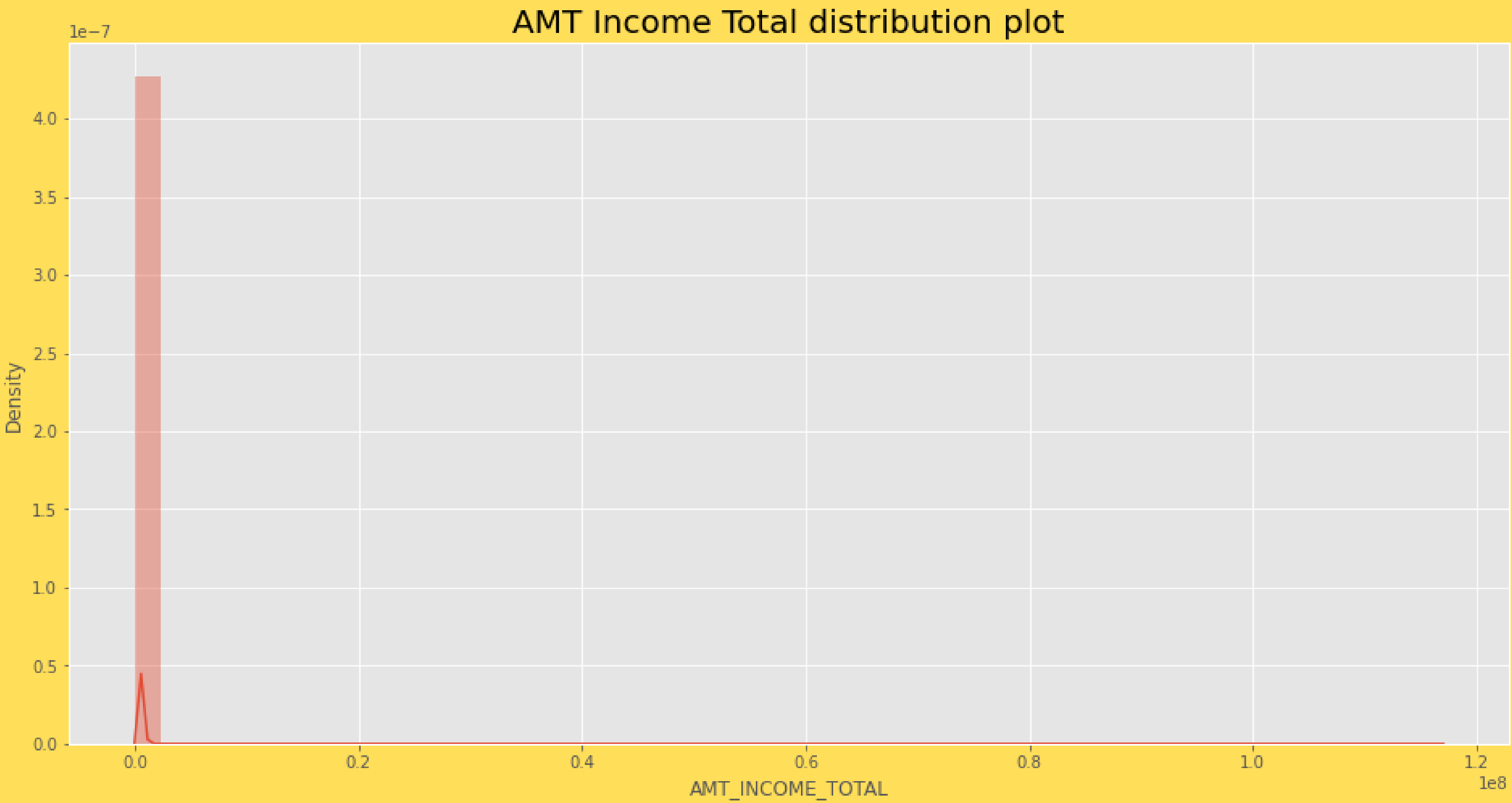
```
vals = application_train['NAME_CONTRACT_TYPE'].value_counts().values
inds = application_train['NAME_CONTRACT_TYPE'].value_counts().index

plt.figure(figsize = (16, 8))
plt.pie(x=vals, autopct="%.2f%%", labels = inds, colors = ['#c2c2f0','#ffcc99'], pctdistance = 0.5)
plt.title('Contract type', fontdict = {'fontsize': 16})

centre_circle = plt.Circle((0,0), 0.7, fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```

The AMT (Alternative Minimum Tax) Credit, Total Income, Annuity and Goods Price distribution plots

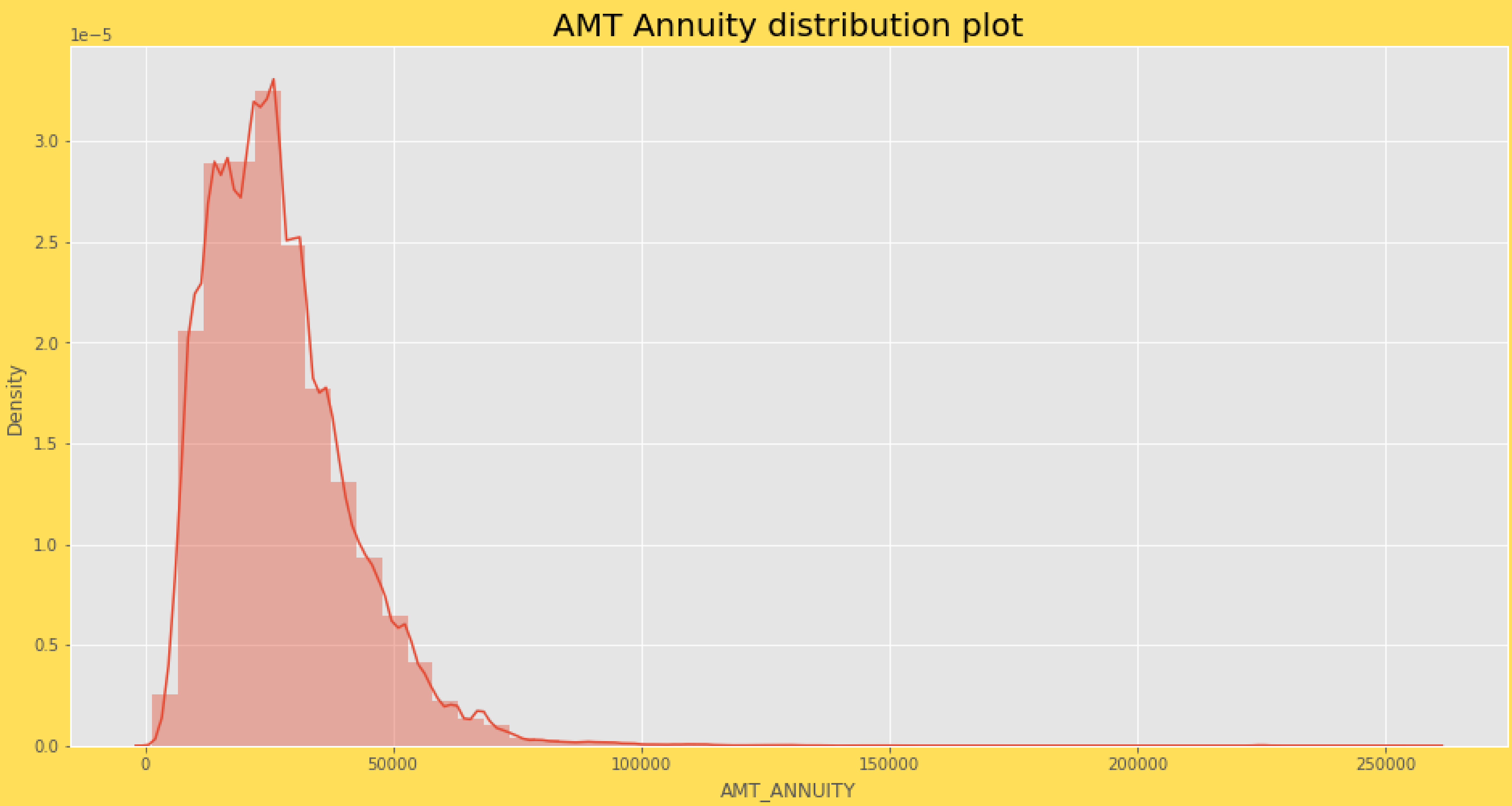
```python
def dist_plot(col, title):
    plt.figure(figsize = (16, 8))
    plt.title(title, fontdict = {'fontsize': 20})
    sns.distplot(application_train[col])
    plt.show()
```



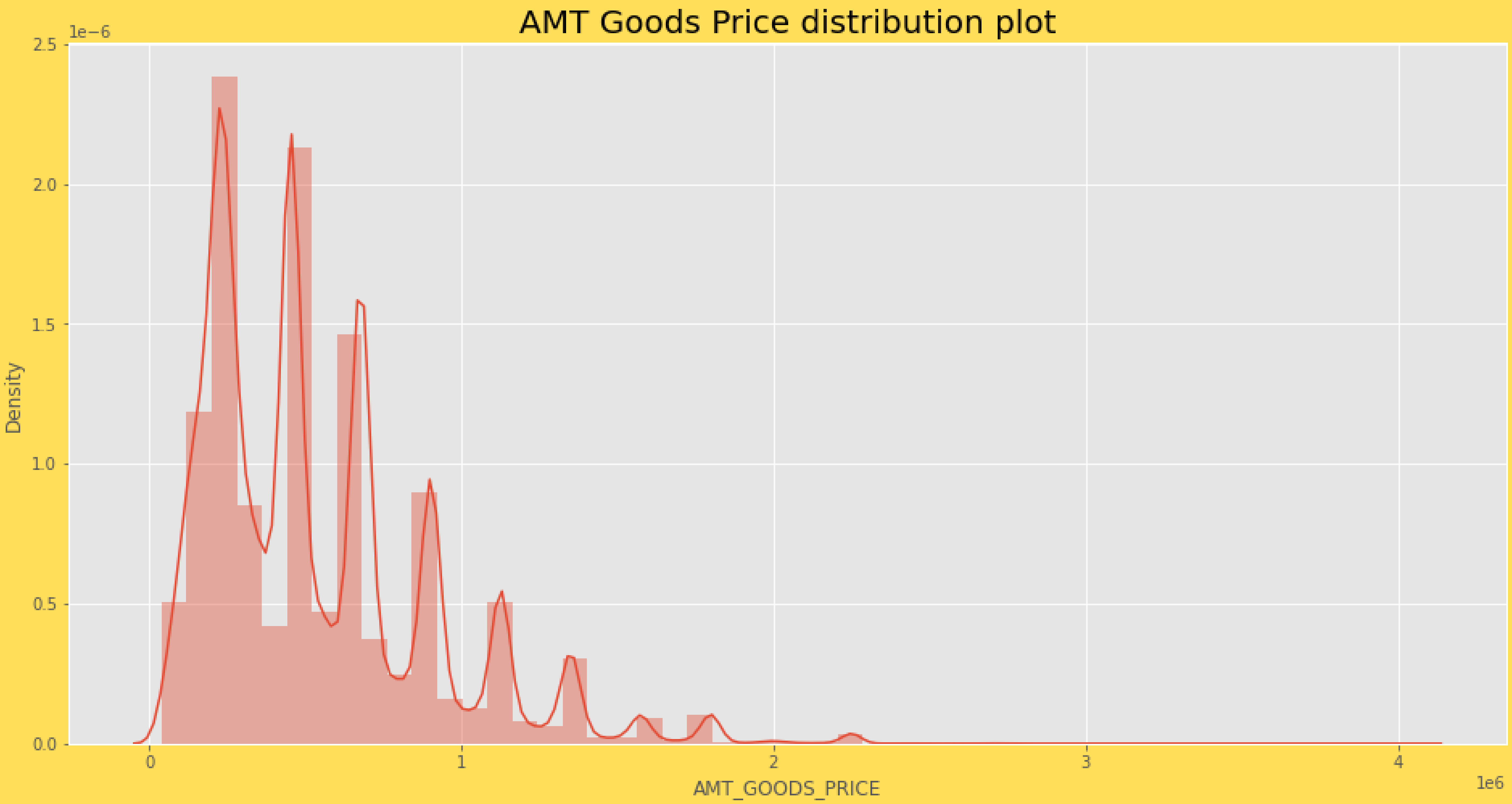
In [599]:
```
dist_plot('AMT_CREDIT', 'AMT Credit distribution plot')
```




```
In [600]: dist_plot('AMT_INCOME_TOTAL', 'AMT Income Total distribution plot')
```



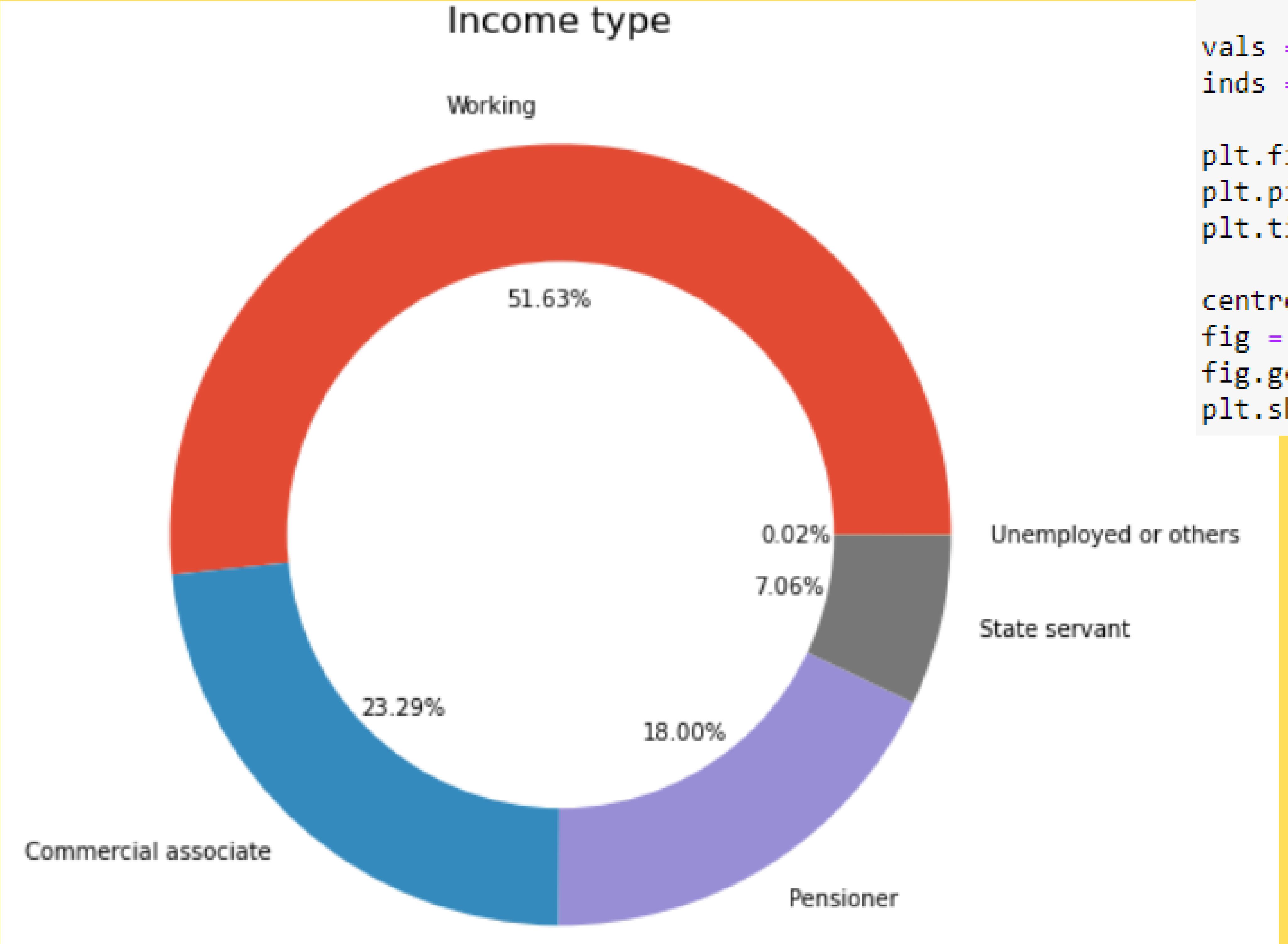
[601]:
```
dist_plot('AMT_ANNUITY', 'AMT Annuity distribution plot')
```

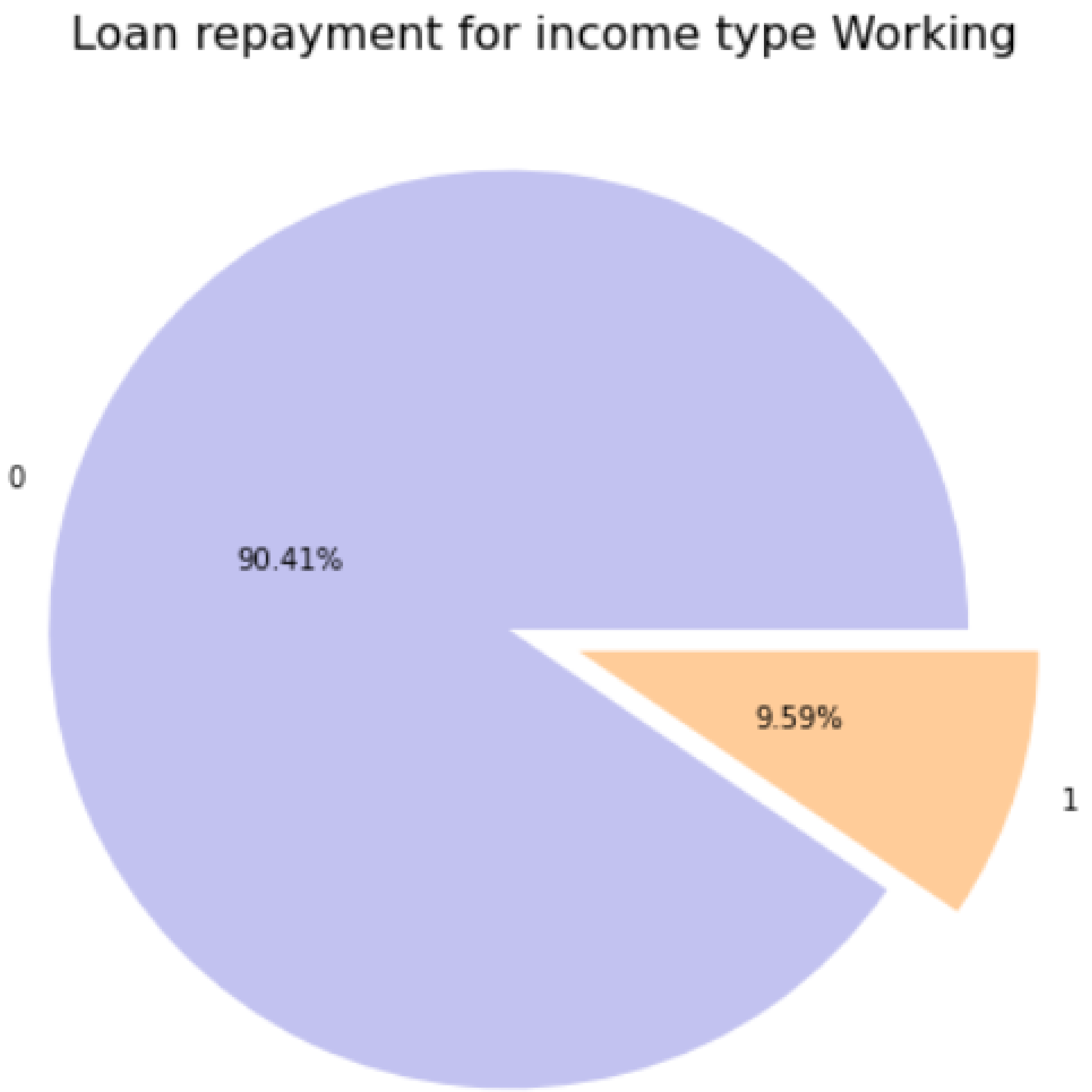


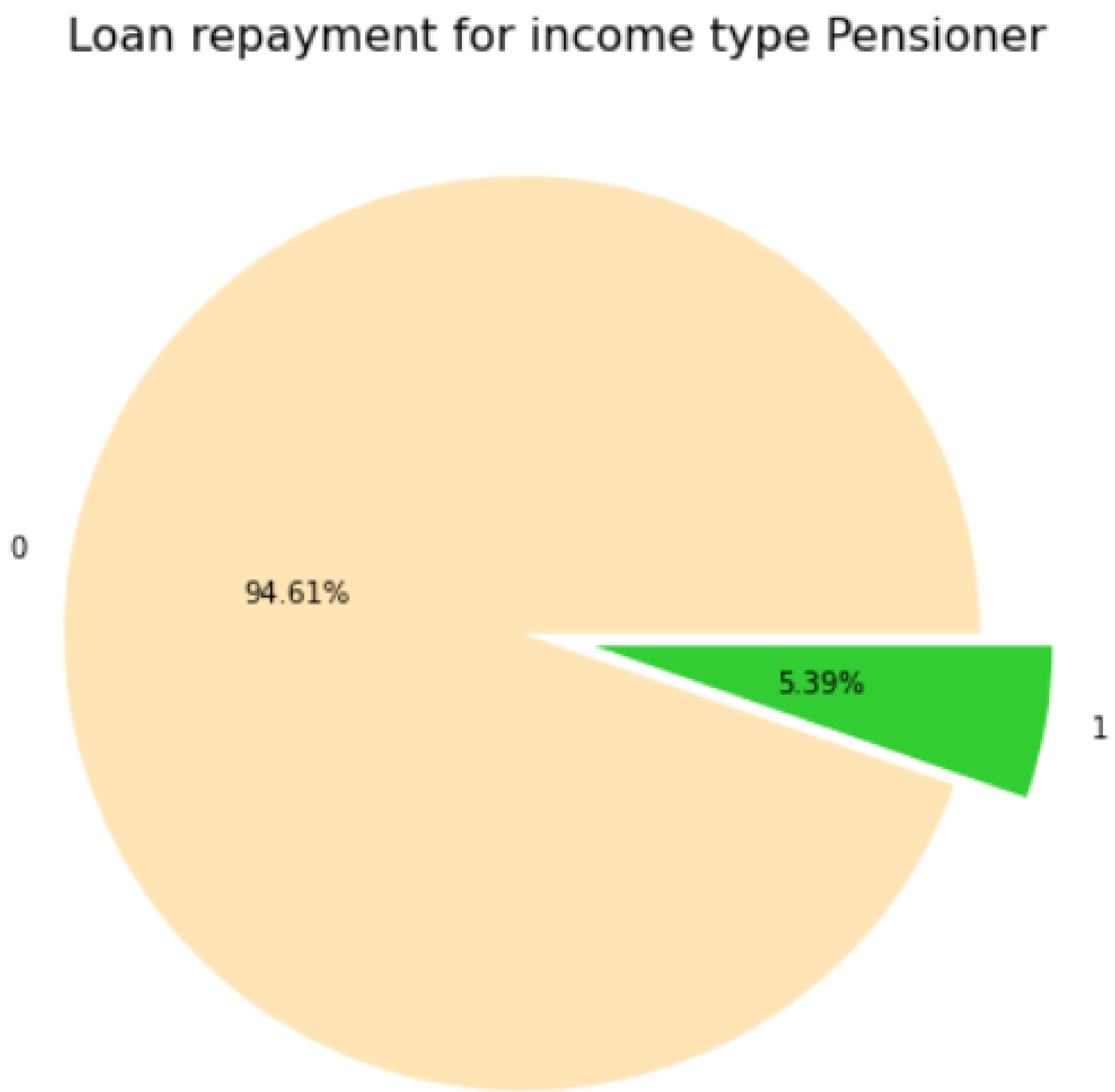
In [602]:
```
dist_plot('AMT_GOODS_PRICE', 'AMT Goods Price distribution plot')
```

From where do the applications get money?

```
vals = application_train['NAME_INCOME_TYPE'].value_counts().values
inds = application_train['NAME_INCOME_TYPE'].value_counts().index

other_vals = vals[4:].sum()
vals = vals[:4]
inds = inds[:4]
other_inds = pd.Index(['Unemployed or others'])

vals = np.append(vals, other_vals)
inds = inds.append(other_inds)

plt.figure(figsize = (16, 8))
plt.pie(vals, autopct="%.2f%%", labels = inds)
plt.title('Income type', fontdict = {'fontsize': 16})

centre_circle = plt.Circle((0,0), 0.7, fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```







90.41%

vs

9.59%

94.61%

vs

5.39%

Hypothesis

Where p_x is proportion for workers 90.41%, p_y for pensioners 94.61%

$$* \{H_0 : p_y <= p_x$$

$$* \{H_1 : p_y > p_x$$

$$p_0 = \frac{n_x \cdot \hat{p}_x + n_y \cdot \hat{p}_y}{n_x + n_y} \text{ or } p_0 = \frac{x_1 + y_1}{n_x + n_y}.$$

For large sample sizes $\left(n \cdot \hat{p} \cdot \hat{q} > 9\right)$ the value of the test statistic z for is computed as

$$T.S. = z = \frac{(\hat{p}_x - \hat{p}_y) - (p_x - p_y)}{\sqrt{\frac{p_0(1-p_0)}{n_x} + \frac{p_0(1-p_0)}{n_y}}}$$

2. To test either null hypothesis

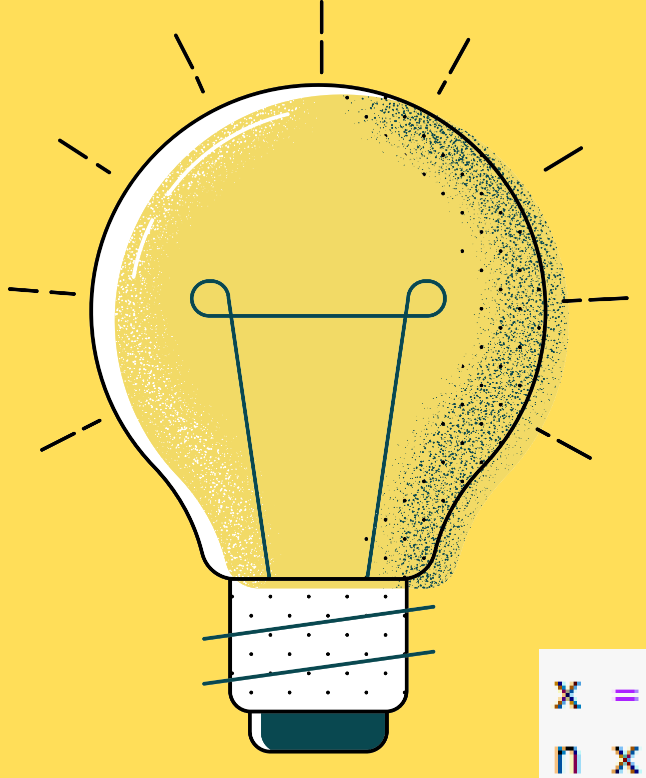
$$H_0 : p_x - p_y = 0 \qquad \text{or} \qquad H_0 : p_x - p_y \geq 0$$

against the alternative

$$H_1 : p_x - p_y < 0$$

the decision rule is

Reject H_0 if $T.S. < -z_\alpha$



Hypothesis test

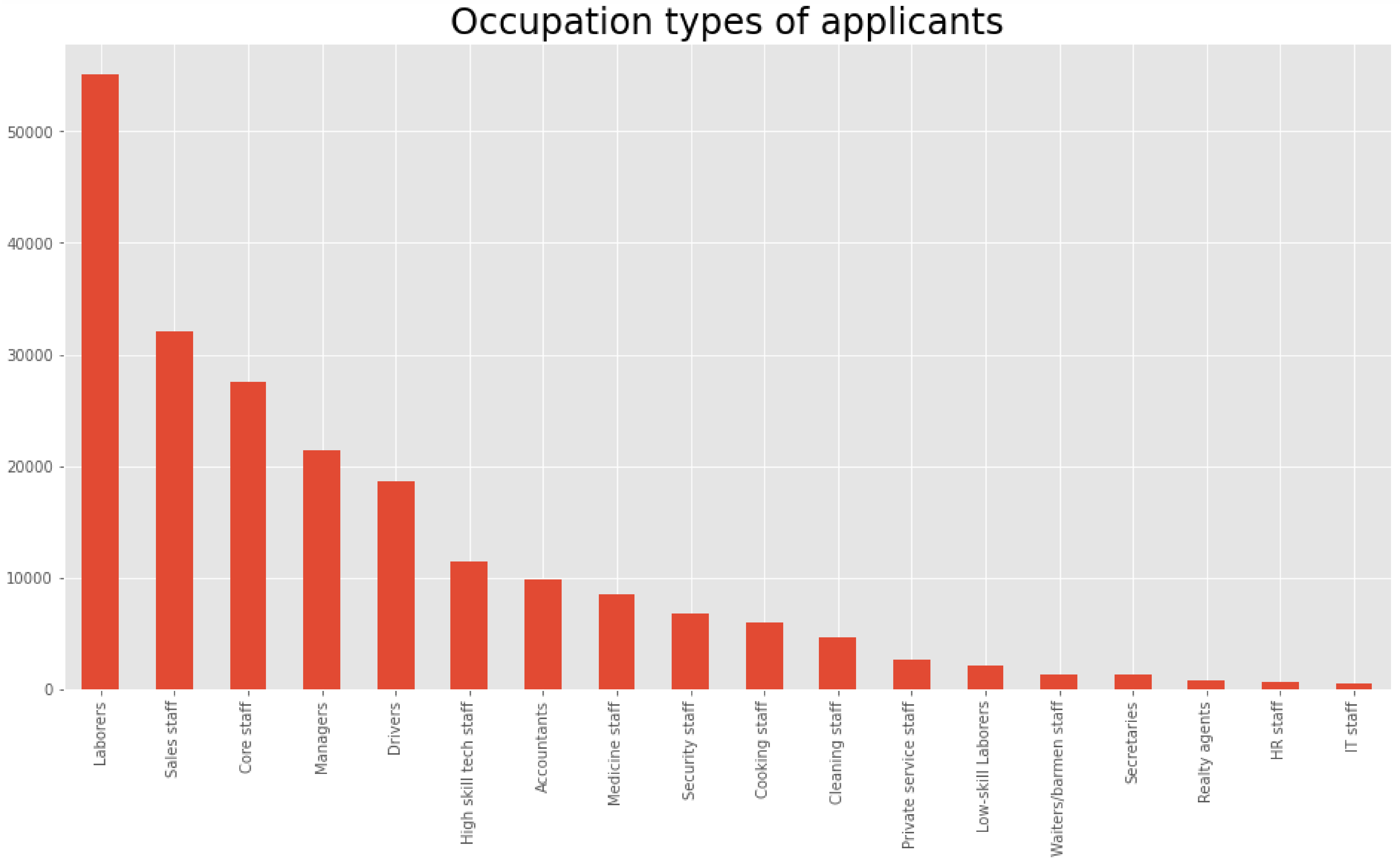
By hypothesis testing, it turns out that pensioners indeed tend to repay loans more than working people.

```
x = application_train.query("NAME_INCOME_TYPE == 'Working'").TARGET
n_x = len(x)
p_x = x.value_counts()[0] / n_x
q_x = x.value_counts()[1] / n_x

y = application_train.query("NAME_INCOME_TYPE == 'Pensioner'").TARGET
n_y = len(y)
p_y = y.value_counts()[0] / n_y
q_y = y.value_counts()[1] / n_y
print('Sample size npq for x:', n_x*p_x*q_x)
print('Sample size npq for y:', n_y*p_y*q_y)

p_0 = (n_x * p_x + n_y * p_y) / (n_x + n_y)
p_0_var = p_0 * (1 - p_0)

test_statistic = (p_x - p_y) / np.sqrt(p_0_var/n_x + p_0_var/n_y)
z_001 = norm.ppf(0.01)
print('Reject H_0 if T.S. < -z_alpha. Do we reject our null hypothesis? :', test_statistic < z_001)
```

```
Sample size npq for x: 13764.251073853402
Sample size npq for y: 2821.37856291319
Reject H_0 if T.S. < -z_alpha. Do we reject our null hypothesis? : True
```

The most of the occupation types are laborers, sales and core staff

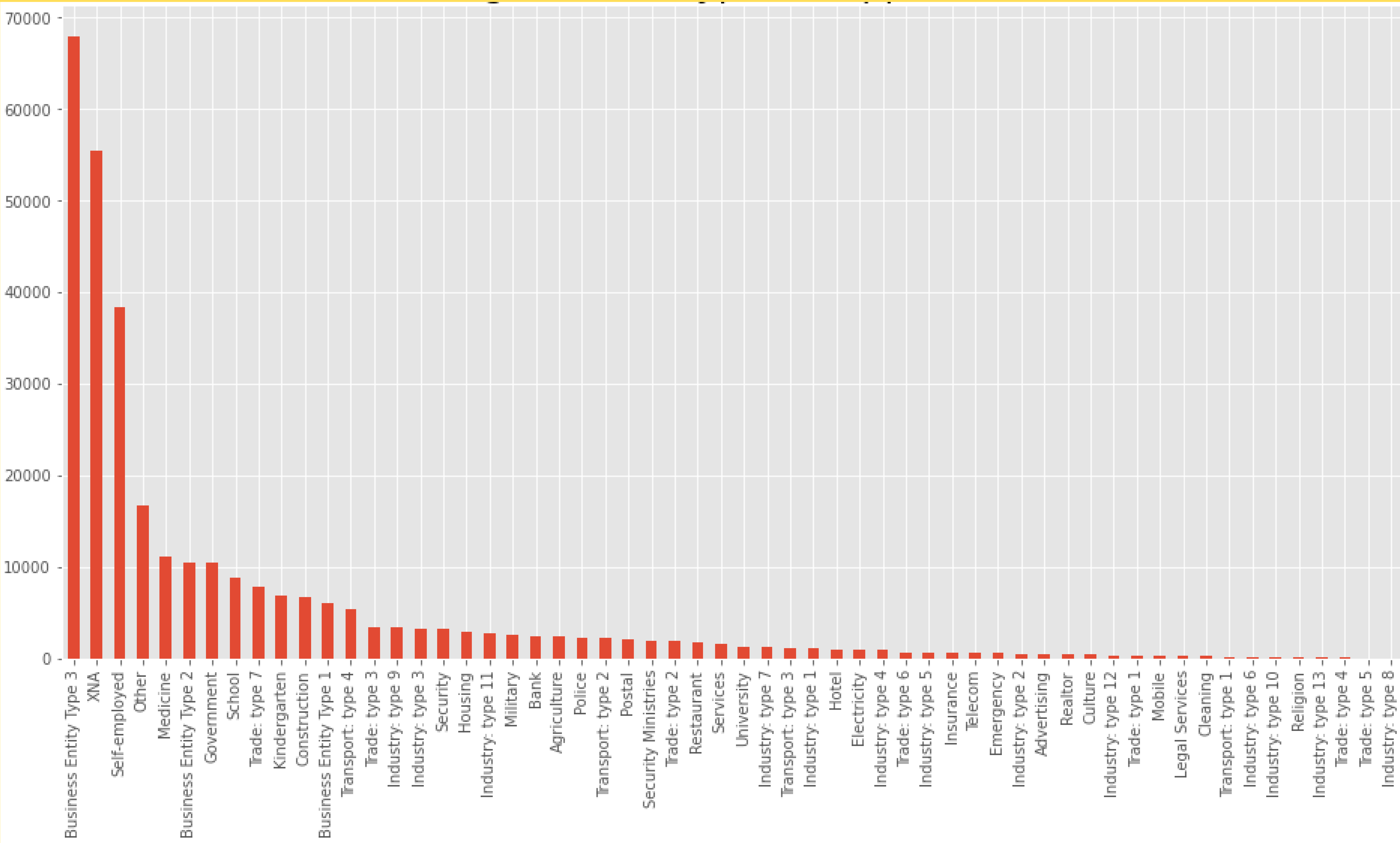
```
plt.figure(figsize = (16, 8))
application_train['OCCUPATION_TYPE'].value_counts().plot(kind = 'bar')
plt.title('Occupation types of applicants', fontdict = {'fontsize': 24})
plt.show()
```



Organization types of applicants

The most frequent organization types are business entity and self-employed

```
plt.figure(figsize = (16, 8))
application_train['ORGANIZATION_TYPE'].value_counts().plot(kind = 'bar')
plt.title('Organization types of applicants', fontdict = {'fontsize': 24})
plt.show()
```



70000
60000
50000
40000
30000
20000
10000
0
Business Entity Type 3
XNA
Self-employed
Other
Medicine
Business Entity Type 2
Government
School
Trade: type 7
Kindergarten
Construction
Business Entity Type 1
Transport: type 4
Trade: type 3
Industry: type 9
Industry: type 3
Security
Housing
Industry: type 11
Military
Bank
Agriculture
Police
Transport: type 2
Postal
Security Ministries
Trade: type 2
Restaurant
Services
University
Industry: type 7
Transport: type 3
Industry: type 1
Hotel
Electricity
Industry: type 4
Trade: type 6
Industry: type 5
Insurance
Telecom
Emergency
Industry: type 2
Advertising
Realtor
Culture
Industry: type 12
Trade: type 1
Mobile
Legal Services
Cleaning
Transport: type 1
Industry: type 6
Industry: type 10
Religion
Industry: type 13
Trade: type 4
Trade: type 5
Industry: type 8

Conclusion

Processing, cleaning and manipulation of data to work in further stages of analysis

Testing hypotheses, understanding the main factors when loans

Data visualization to understand the basic concept of data for a loan in a bank