Bank Loan Case Study

By Sneha Vora

Project Description

The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history. Because of that, some consumers use it as their advantage by becoming a defaulter. Suppose you work for a consumer finance company which specialises in lending various types of loans to urban customers. You have to use EDA to analyse the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.

Approach

- Understood the given data set
- Converted given data set in data frames
- Used python libraries pandas and numpy for data analysis
- Used matplotlib and seaborn libraries for visualization
- Performed Exploratory Data Analysis

Tech-Stack used

- Jupyter Notebook (Anaconda)

Insights

- Importing all the required libraries

```
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
%matplotlib inline
```

- Loading the dataset

```
Loanapp = pd.read_csv(r'C:\Users\SNEHA\Downloads\input\bank-loan-application-data\application_data.csv')
loanapp.head()
```

Out[3]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_
0	100002	1	Cash loans	M	N	Y	0	2025
1	100003	0	Cash loans	F	N	N	0	2700
2	100004	0	Revolving loans	M	Y	Y	0	6750
3	100006	0	Cash loans	F	N	Y	0	1350
4	100007	0	Cash loans	M	N	Y	0	1215

5 rows × 122 columns

- Checking null value percentage of the columns

```
loanapp.iloc[:,0:122].isnull().sum()/len(loanapp)*100
```

```
SK_ID_CURR                     0.000000
TARGET                         0.000000
NAME_CONTRACT_TYPE             0.000000
CODE_GENDER                    0.000000
FLAG_OWN_CAR                   0.000000
FLAG_OWN_REALTY                0.000000
CNT_CHILDREN                   0.000000
AMT_INCOME_TOTAL               0.000000
AMT_CREDIT                     0.000000
AMT_ANNUITY                    0.003902
AMT_GOODS_PRICE                0.090403
NAME_TYPE_SUITE                0.420148
NAME_INCOME_TYPE               0.000000
NAME_EDUCATION_TYPE            0.000000
NAME_FAMILY_STATUS             0.000000
NAME_HOUSING_TYPE              0.000000
REGION_POPULATION_RELATIVE     0.000000
DAYS_BIRTH                     0.000000
DAYS_EMPLOYED                  0.000000
DAYS_REGISTRATION              0.000000
DAYS_ID_PUBLISH                0.000000
OWN_CAR_AGE                   65.990810
FLAG_MOBIL                     0.000000
```

```
FLAG_EMP_PHONE                  0.000000
FLAG_WORK_PHONE                 0.000000
FLAG_CONT_MOBILE                0.000000
FLAG_PHONE                      0.000000
FLAG_EMAIL                      0.000000
OCCUPATION_TYPE                31.345545
CNT_FAM_MEMBERS                 0.000650
REGION_RATING_CLIENT            0.000000
REGION_RATING_CLIENT_W_CITY     0.000000
WEEKDAY_APPR_PROCESS_START      0.000000
HOUR_APPR_PROCESS_START         0.000000
REG_REGION_NOT_LIVE_REGION      0.000000
REG_REGION_NOT_WORK_REGION      0.000000
LIVE_REGION_NOT_WORK_REGION     0.000000
REG_CITY_NOT_LIVE_CITY          0.000000
REG_CITY_NOT_WORK_CITY          0.000000
LIVE_CITY_NOT_WORK_CITY         0.000000
ORGANIZATION_TYPE               0.000000
EXT_SOURCE_1                   56.381073
EXT_SOURCE_2                    0.214626
EXT_SOURCE_3                   19.825307
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
OBS_30_CNT_SOCIAL_CIRCLE	0.332021
DEF_30_CNT_SOCIAL_CIRCLE	0.332021
OBS_60_CNT_SOCIAL_CIRCLE	0.332021
DEF_60_CNT_SOCIAL_CIRCLE	0.332021
DAYS_LAST_PHONE_CHANGE	0.000325
FLAG_DOCUMENT_2	0.000000
FLAG_DOCUMENT_3	0.000000
FLAG_DOCUMENT_4	0.000000
FLAG_DOCUMENT_5	0.000000
FLAG_DOCUMENT_6	0.000000
FLAG_DOCUMENT_7	0.000000
FLAG_DOCUMENT_8	0.000000
FLAG_DOCUMENT_9	0.000000
FLAG_DOCUMENT_10	0.000000
FLAG_DOCUMENT_11	0.000000
FLAG_DOCUMENT_12	0.000000
FLAG_DOCUMENT_13	0.000000
FLAG_DOCUMENT_14	0.000000
FLAG_DOCUMENT_15	0.000000
FLAG_DOCUMENT_16	0.000000
FLAG_DOCUMENT_17	0.000000
FLAG_DOCUMENT_18	0.000000
FLAG_DOCUMENT_19	0.000000
FLAG_DOCUMENT_20	0.000000
FLAG_DOCUMENT_21	0.000000
AMT_REQ_CREDIT_BUREAU_HOUR	13.501631
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
dtype: float64
```

- Null value columns more than 30%

```
appempty=loanapp.isnull().sum()
appempty=appempty[appempty.values>(0.3*len(loanapp))]
len(appempty)
```

```
Out[11]:
        50
```

- Dropping columns with more than 30% null values

- Dropping the non relevant columns

```
nonrelevant=['FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBILE','FLAG_PHONE','FLAG_EMAIL','REGION_RATING_CLIENT','CNT_FAM_MEMBERS','REGION_RATING_CLIENT_W_CITY','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']

loanapp.drop(labels=nonrelevant,axis=1,inplace=True)
loanapp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 42 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_CURR                   307511 non-null  int64
 1   TARGET                       307511 non-null  int64
 2   NAME_CONTRACT_TYPE           307511 non-null  object
 3   CODE_GENDER                  307511 non-null  object
 4   FLAG_OWN_CAR                 307511 non-null  object
 5   FLAG_OWN_REALTY              307511 non-null  object
 6   CNT_CHILDREN                 307511 non-null  int64
 7   AMT_INCOME_TOTAL             307511 non-null  float64
```

```
 8   AMT_CREDIT                   307511 non-null  float64
 9   AMT_ANNUITY                  307511 non-null  float64
 10  AMT_GOODS_PRICE              307233 non-null  float64
 11  NAME_TYPE_SUITE              306219 non-null  object
 12  NAME_INCOME_TYPE             307511 non-null  object
 13  NAME_EDUCATION_TYPE          307511 non-null  object
 14  NAME_FAMILY_STATUS           307511 non-null  object
 15  NAME_HOUSING_TYPE            307511 non-null  object
 16  REGION_POPULATION_RELATIVE   307511 non-null  float64
 17  DAYS_BIRTH                   307511 non-null  int64
 18  DAYS_EMPLOYED                307511 non-null  int64
 19  DAYS_REGISTRATION            307511 non-null  float64
 20  DAYS_ID_PUBLISH              307511 non-null  int64
 21  WEEKDAY_APPR_PROCESS_START   307511 non-null  object
 22  HOUR_APPR_PROCESS_START      307511 non-null  int64
 23  REG_REGION_NOT_LIVE_REGION   307511 non-null  int64
 24  REG_REGION_NOT_WORK_REGION   307511 non-null  int64
 25  LIVE_REGION_NOT_WORK_REGION  307511 non-null  int64
 26  REG_CITY_NOT_LIVE_CITY       307511 non-null  int64
 27  REG_CITY_NOT_WORK_CITY       307511 non-null  int64
 28  LIVE_CITY_NOT_WORK_CITY      307511 non-null  int64
 29  ORGANIZATION_TYPE            307511 non-null  object
 30  EXT_SOURCE_2                 306851 non-null  float64
 31  EXT_SOURCE_3                 246546 non-null  float64
 32  OBS_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 33  DEF_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 34  OBS_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 35  DEF_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 36  AMT_REQ_CREDIT_BUREAU_HOUR   265992 non-null  float64
 37  AMT_REQ_CREDIT_BUREAU_DAY    265992 non-null  float64
 38  AMT_REQ_CREDIT_BUREAU_WEEK   265992 non-null  float64
 39  AMT_REQ_CREDIT_BUREAU_MON    265992 non-null  float64
 40  AMT_REQ_CREDIT_BUREAU_QRT    265992 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_YEAR   265992 non-null  float64
dtypes: float64(18), int64(13), object(11)
memory usage: 98.5+ MB
```

- Converting categorical columns to numerical data

```
ncs=['TARGET','CNT_CHILDREN','AMT_INCOME_TOTAL','AMT_CREDIT','AMT_ANNUITY',
'REGION_POPULATION_RELATIVE','DAYS_BIRTH','DAYS_EMPLOYED','DAYS_REGISTRATIO
N','DAYS_ID_PUBLISH','HOUR_APPR_PROCESS_START','LIVE_REGION_NOT_WORK_REGION
', 'REG_CITY_NOT_LIVE_CITY','REG_CITY_NOT_WORK_CITY', 'LIVE_CITY_NOT_WORK_C
ITY']
loanapp[ncs]=loanapp[ncs].apply(pd.to_numeric)
loanapp.head()
```

-
-
-
-

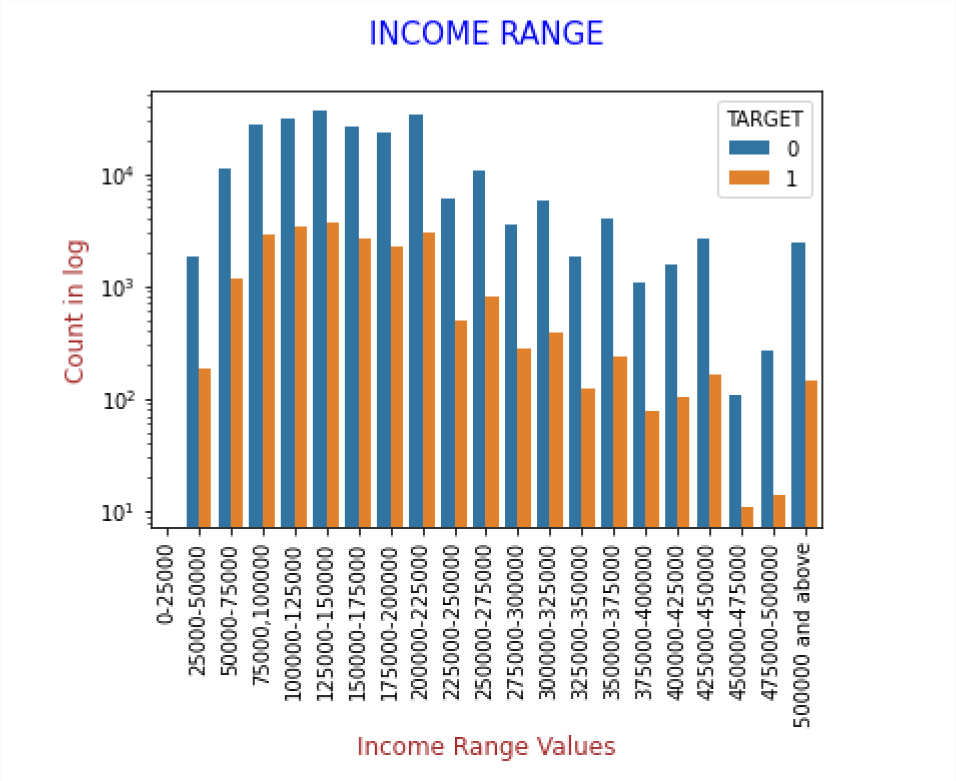
Out[38]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_
0	100002	1	Cash loans	M	N	Y	0	2025
1	100003	0	Cash loans	F	N	N	0	2700
2	100004	0	Revolving loans	M	Y	Y	0	6750
3	100006	0	Cash loans	F	N	Y	0	1350
4	100007	0	Cash loans	M	N	Y	0	1215

5 rows × 42 columns

- **Univariate Analysis**

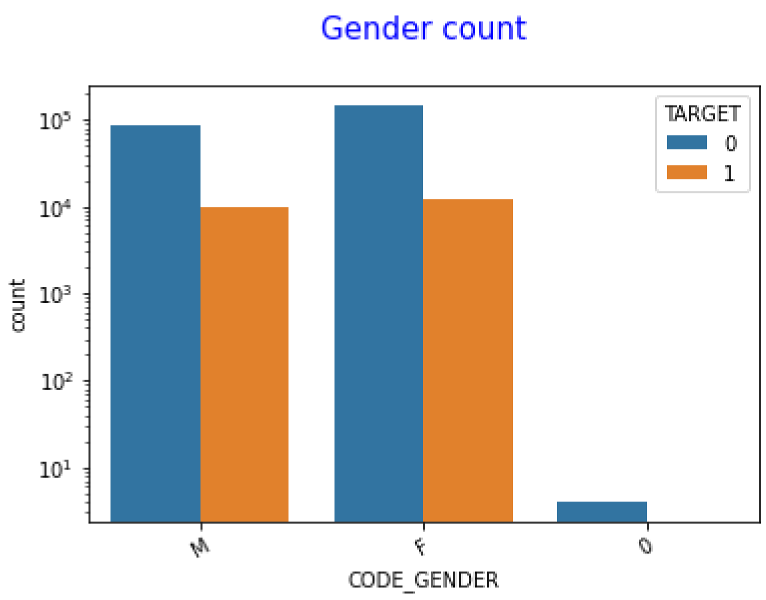
```
sns.countplot(loanapp.Income_range, hue=loanapp.TARGET)
plt.yscale('log')
plt.ylabel("Count in log", fontdict={'fontsize': 12, 'fontweight' : 5, 'col
or' : 'Brown'})
plt.xlabel("Income Range Values", fontdict={'fontsize': 12, 'fontweight' :
5, 'color' : 'Brown'})
plt.xticks(rotation=90)
plt.title('INCOME RANGE\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'col
or' : 'Blue'})
plt.show()
```



- From the above plot we can infer that the maximum number of clients without payment difficulties lie in the income range 1.2lac-1.5lac and immediate next income range is 2lac-.25 lac
- Most clients wiht payment difficulties lie in the income range is 4.5lac-4.75lac

- Plotting code gender values

```
sns.countplot(loanapp.CODE_GENDER,hue=loanapp.TARGET)
plt.yscale('log')
plt.xticks(rotation=30)
plt.title('Gender count\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'col
or' : 'Blue'})
plt.show()
```

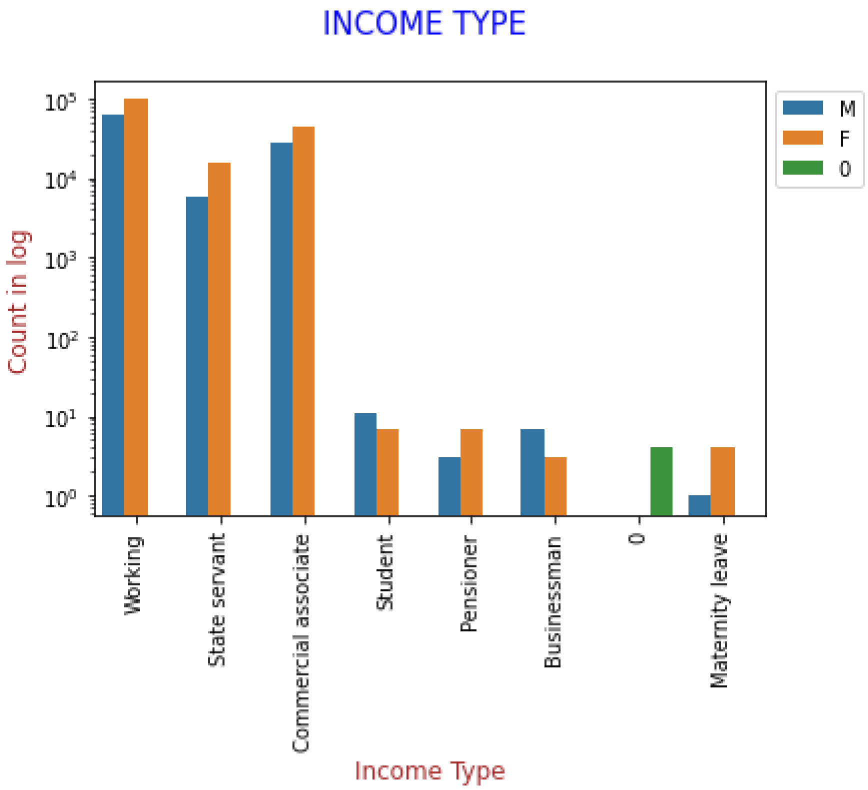


- It can be seen that both number of males and females is same for having payment difficulties

- It can also be seen that number of females is more than males for not having payment dificulties

- Plotting *NAME_INCOME_TYPE values*

```
sns.countplot(loanapp.NAME_INCOME_TYPE,hue=loanapp.CODE_GENDER)
plt.yscale('log')
plt.ylabel("Count in log", fontdict={'fontsize': 12, 'fontweight' : 5, 'col
or' : 'Brown'})
plt.xticks(rotation=90)
plt.xlabel("Income Type", fontdict={'fontsize': 12, 'fontweight' : 5, 'colo
r' : 'Brown'})
plt.title('INCOME TYPE \n',fontdict={'fontsize': 15, 'fontweight' : 7, 'col
or' : 'Blue'})
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

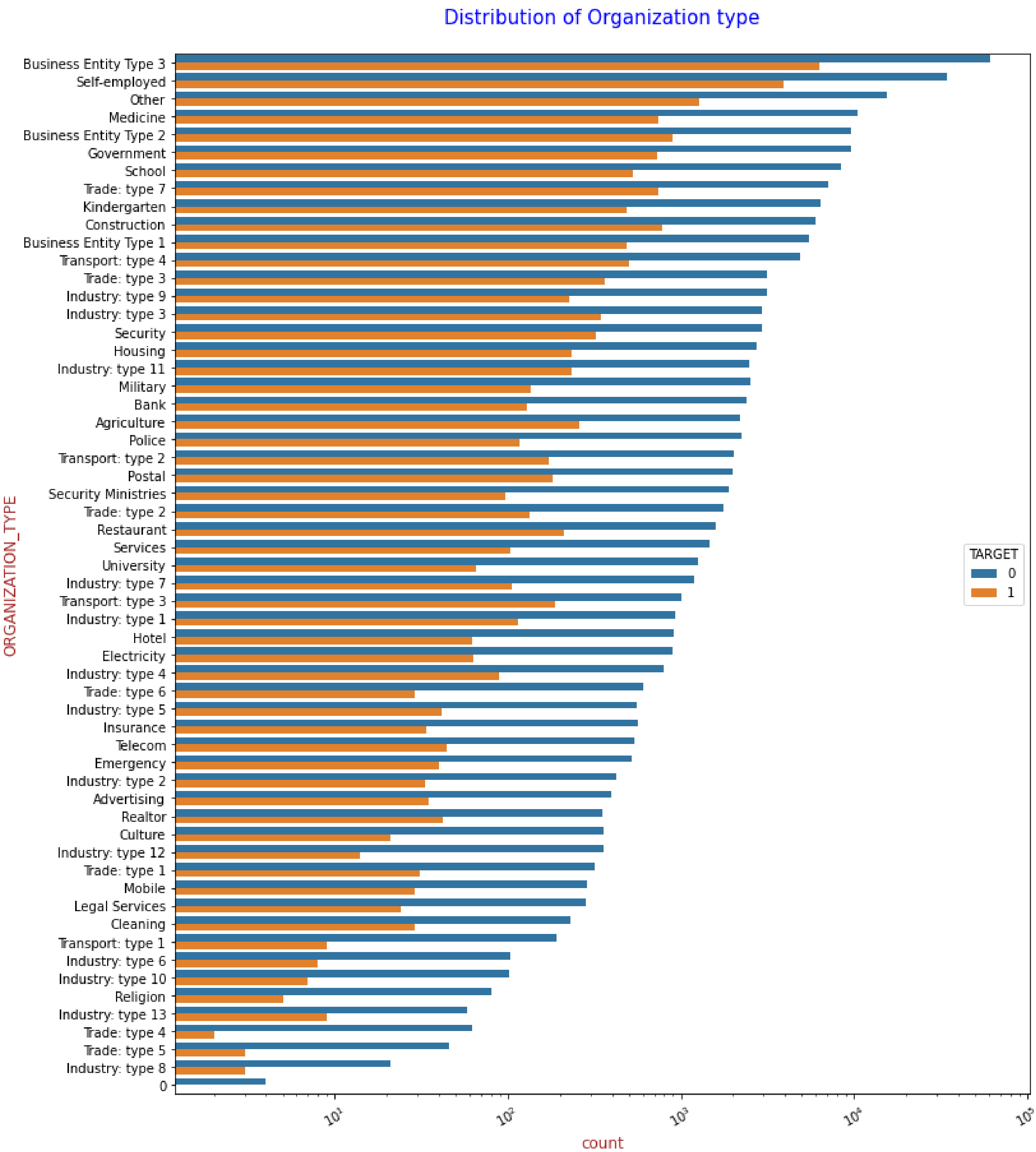


- We can conclude that most clients who fall in working type income category are applying for loans

- Checking what organisation type people are applying for loans and how many of them actually get it

```
plt.figure(figsize=(12,15))
plt.xticks(rotation=30)
plt.xscale('log')
```

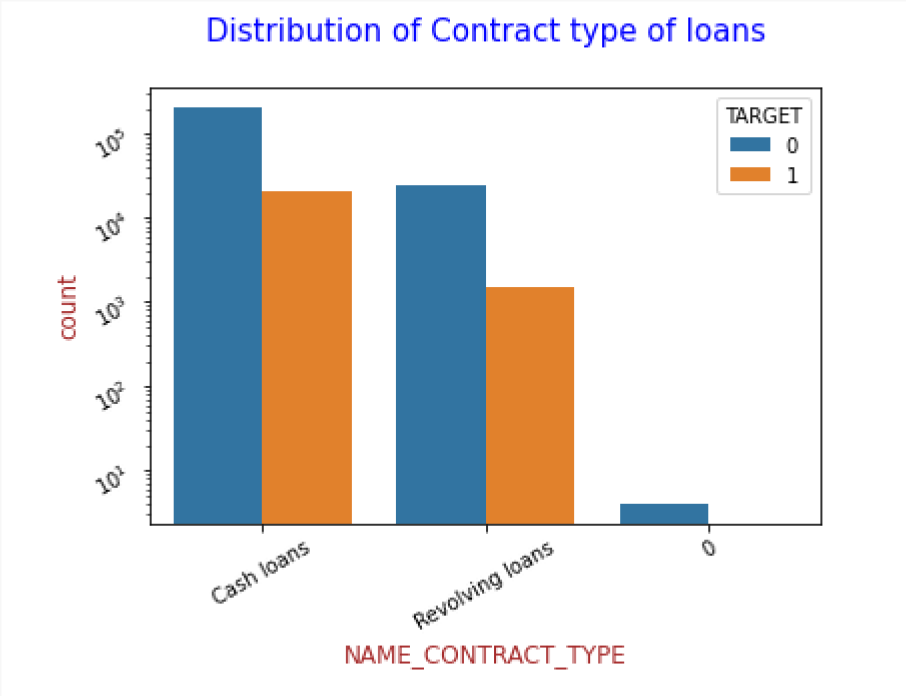
```
plt.ylabel("ORGANIZATION_TYPE", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.xlabel('count', fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.title('Distribution of Organization type\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
sns.countplot(data=loanapp,y='ORGANIZATION_TYPE',order=loanapp['ORGANIZATION_TYPE'].value_counts().index,hue='TARGET')
plt.show()
```



- It can be seen that Trade: type 4, organisation type have least count of payment difficulty clients
- Most clients with no payment difficulties lie in organsation type named Business Entity Type 3

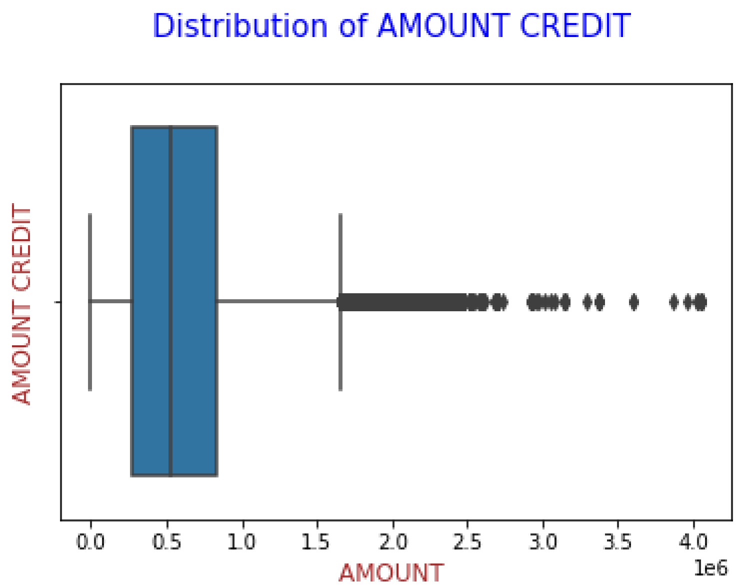
- Which type of contract loan is receiving more applications

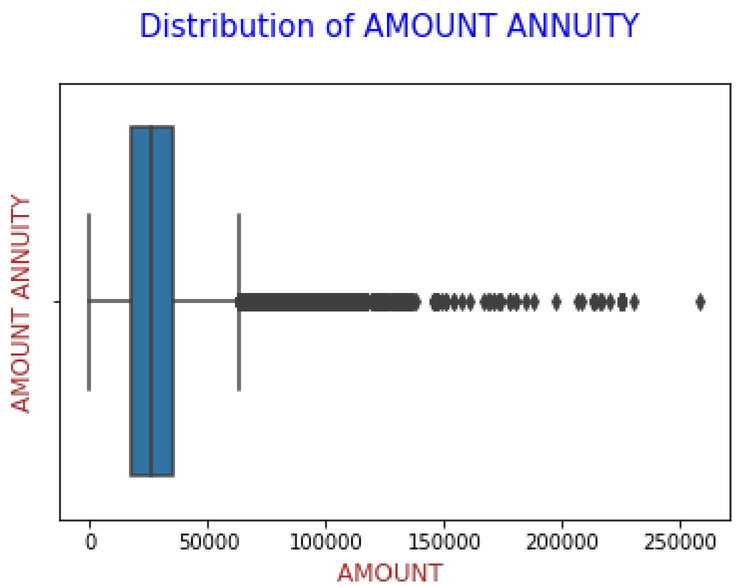
```
plt.xticks(rotation=90)
plt.yscale('log')
plt.xlabel("CONTRACT_TYPE", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.yticks(rotation=30)
plt.ylabel("Count in log", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.xticks(rotation=30)
plt.title('Distribution of Contract type of loans\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
sns.countplot(data=loanapp,x='NAME_CONTRACT_TYPE',order=loanapp.NAME_CONTRACT_TYPE.value_counts().index, hue=loanapp.TARGET)
plt.show()
```



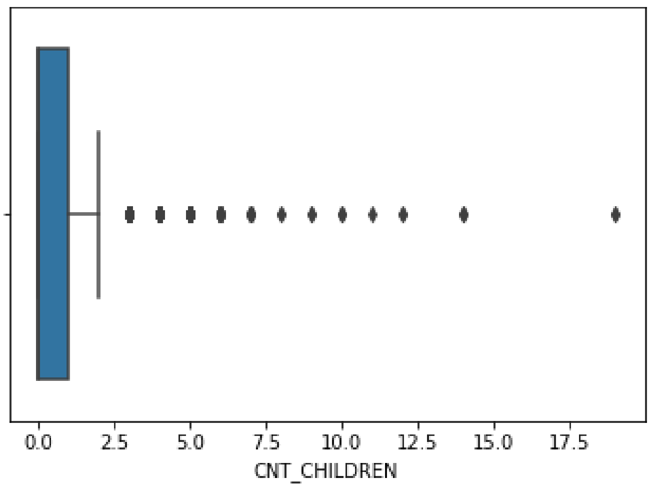
- Most cash loans applicants don't have payment difficulties
- The same type of loans also have the most applicants with payment difficulties

- Checking for outliers

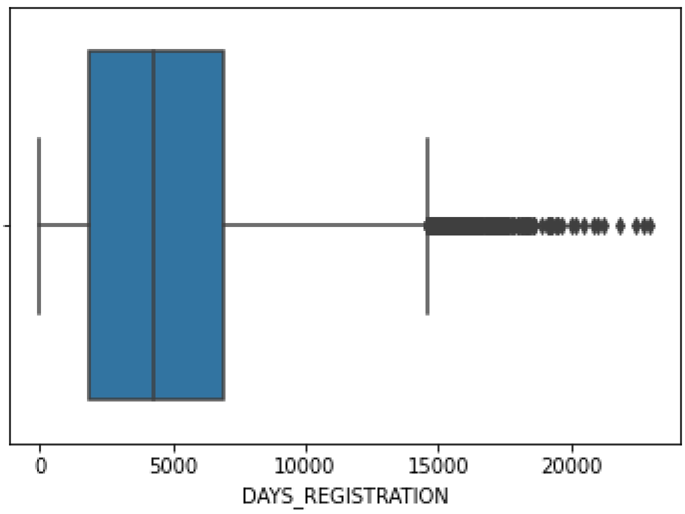




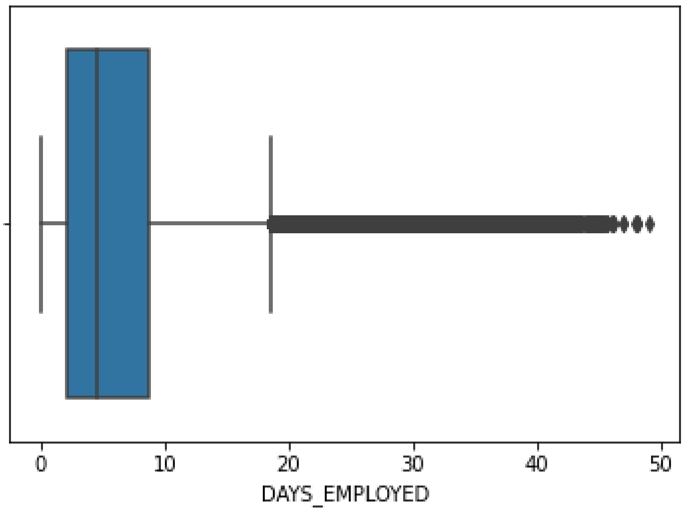
Count of children in a family



Checking for Range of Client's Number of Registered Days



Checking the Range of Number of Days Employed

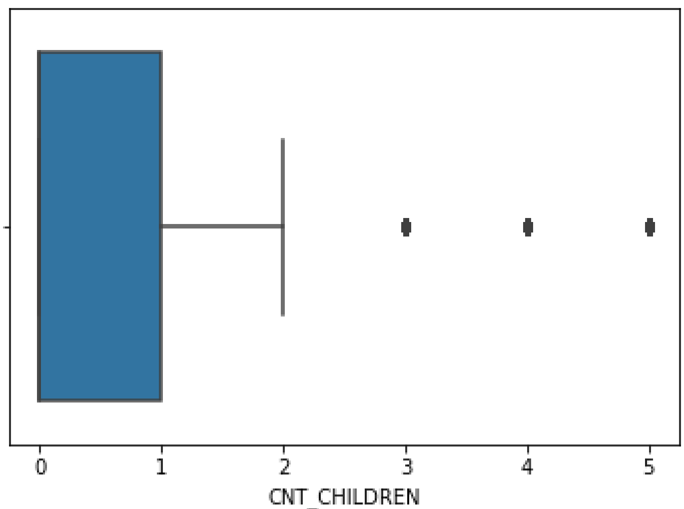


- Dropping rows with unrealistic data for children count

```
loanapp=loanapp[~(loanapp.CNT_CHILDREN>=6)]
```

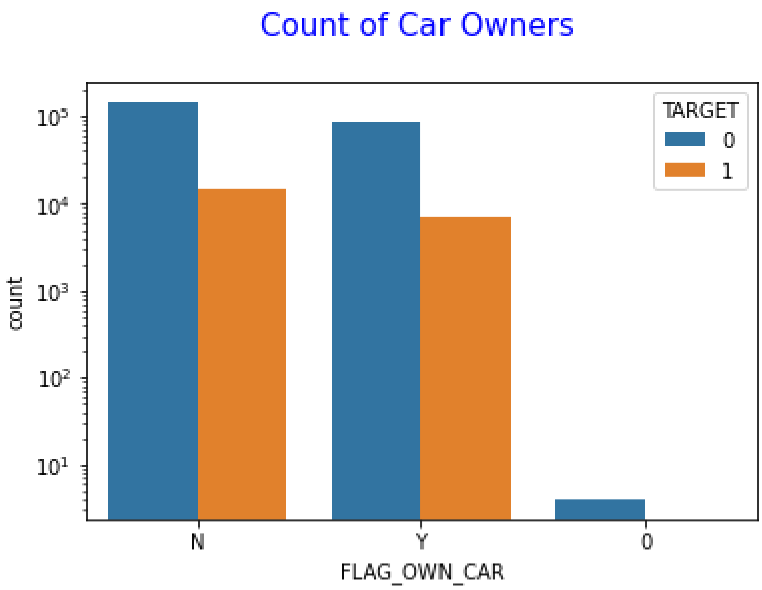
```
In [56]:
sns.boxplot(loanapp.CNT_CHILDREN)
plt.title('Count of children in a family after dropping some rows \n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
plt.show()
```

Count of children in a family after dropping some rows



- Checking count of car owners with their capabilities to make a payment

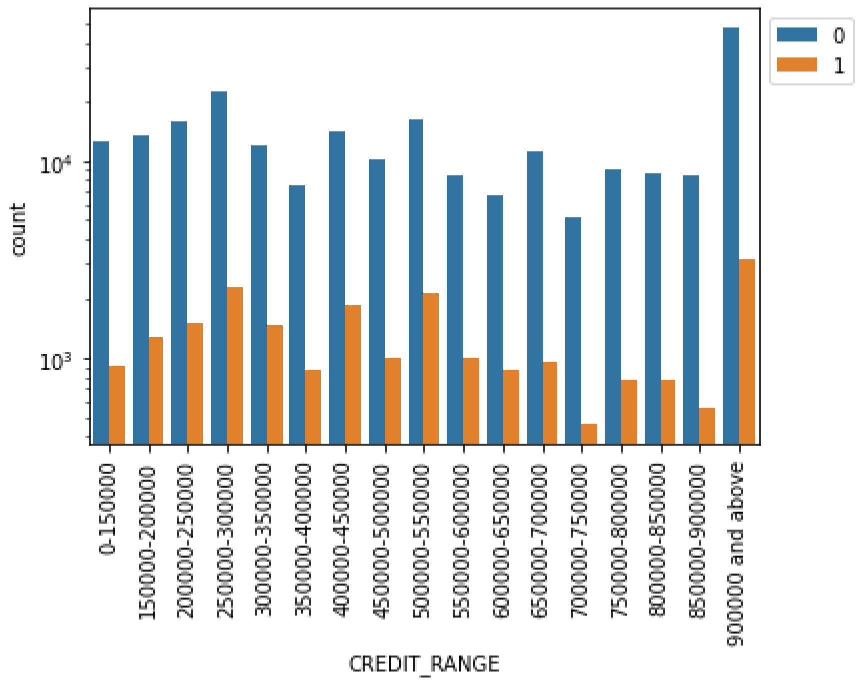
```
sns.countplot(loanapp.FLAG_OWN_CAR, hue=loanapp.TARGET)
plt.yscale('log')
plt.title('Count of Car Owners \n',fontdict={'fontsize': 15, 'fontweight' :
7, 'color' : 'Blue'})
plt.show()
```



- Dropped rows for more than 40 years of employed

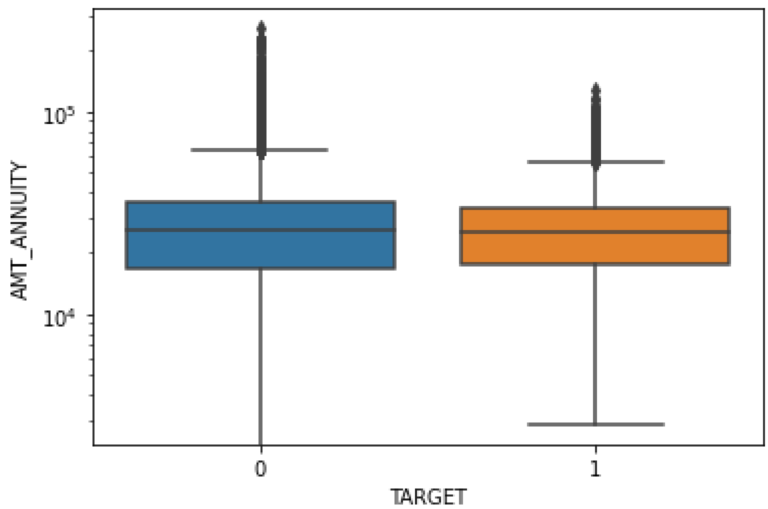
- Credit range that clients are getting and if they are likely to pay or not

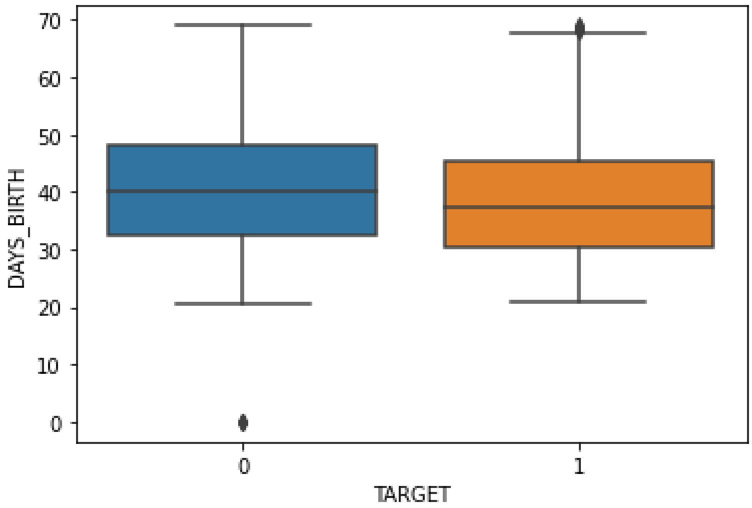
```
sns.countplot(loanapp.CREDIT_RANGE, hue=loanapp.TARGET)
plt.xticks(rotation=90)
plt.yscale('log')
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

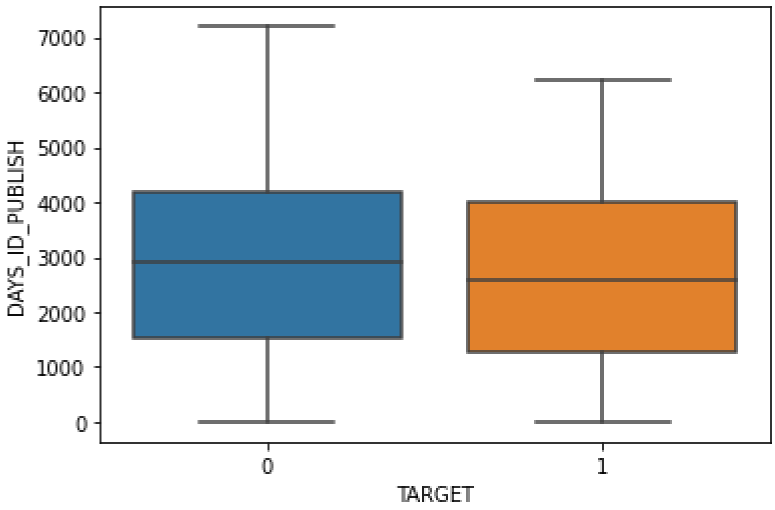


- Clients with credit range lying in 900000 and above are the ones who are capable of paying the loans back
- Least number of clients lying in income range 7lac- 7.5 lac are not capable of paying

- Checking the annuity amount of target variables

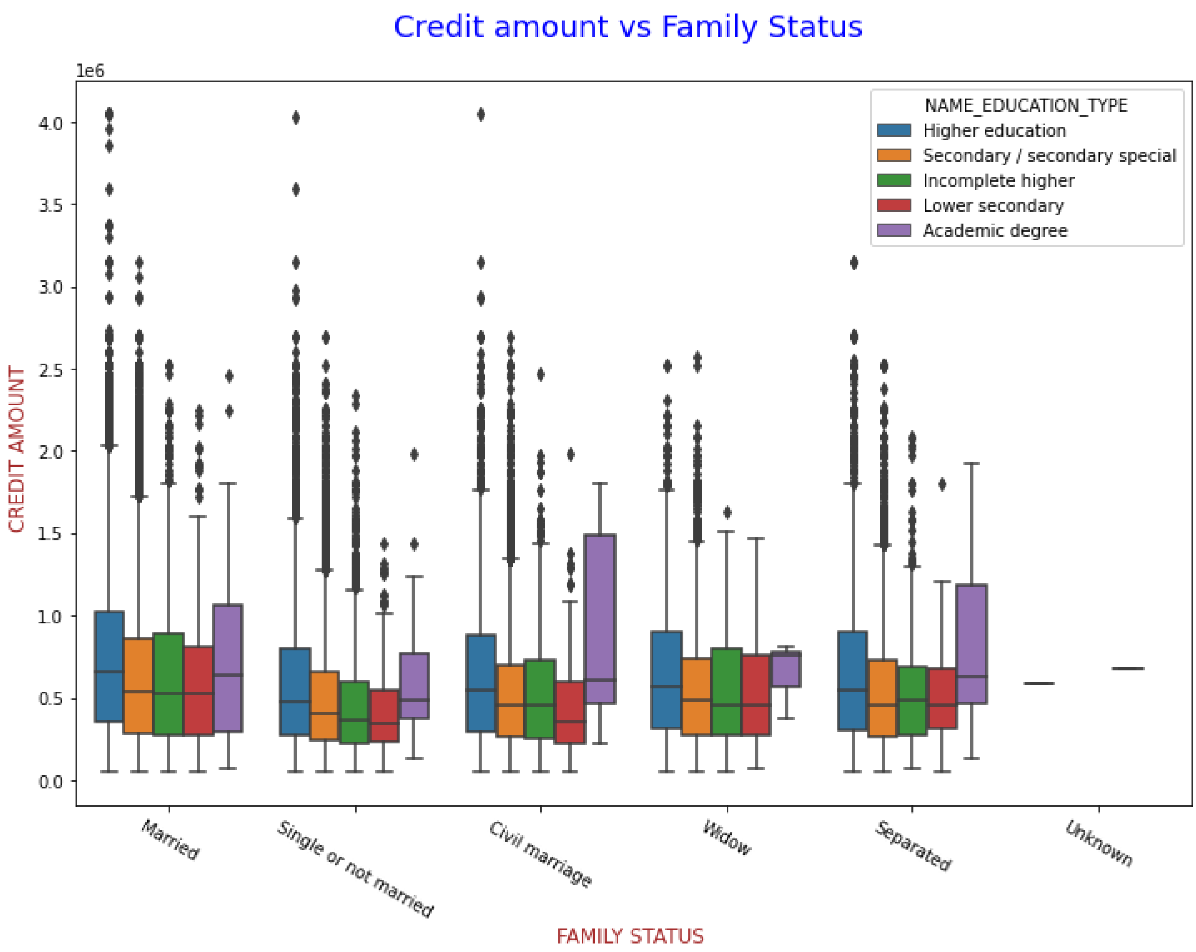






- **Bivariate Analysis**

```
target0=loanapp.loc[loanapp["TARGET"]==0]
target1=loanapp.loc[loanapp["TARGET"]==1]
plt.figure(figsize=(12,8))
scale_factor=5
sns.boxplot(data=target0,x=target0.NAME_FAMILY_STATUS,y=target0.AMT_CREDIT,
hue=target0.NAME_EDUCATION_TYPE)
plt.xticks(rotation=-30)
plt.xlabel("FAMILY STATUS ", fontdict={'fontsize': 12, 'fontweight' : 5, 'c
olor' : 'Brown'})
plt.ylabel("CREDIT AMOUNT ", fontdict={'fontsize': 12, 'fontweight' : 5, 'c
olor' : 'Brown'})
plt.title('Credit amount vs Family Status \n',fontdict={'fontsize': 18, 'fo
ntweight' : 10, 'color' : 'Blue'})
plt.show()
```

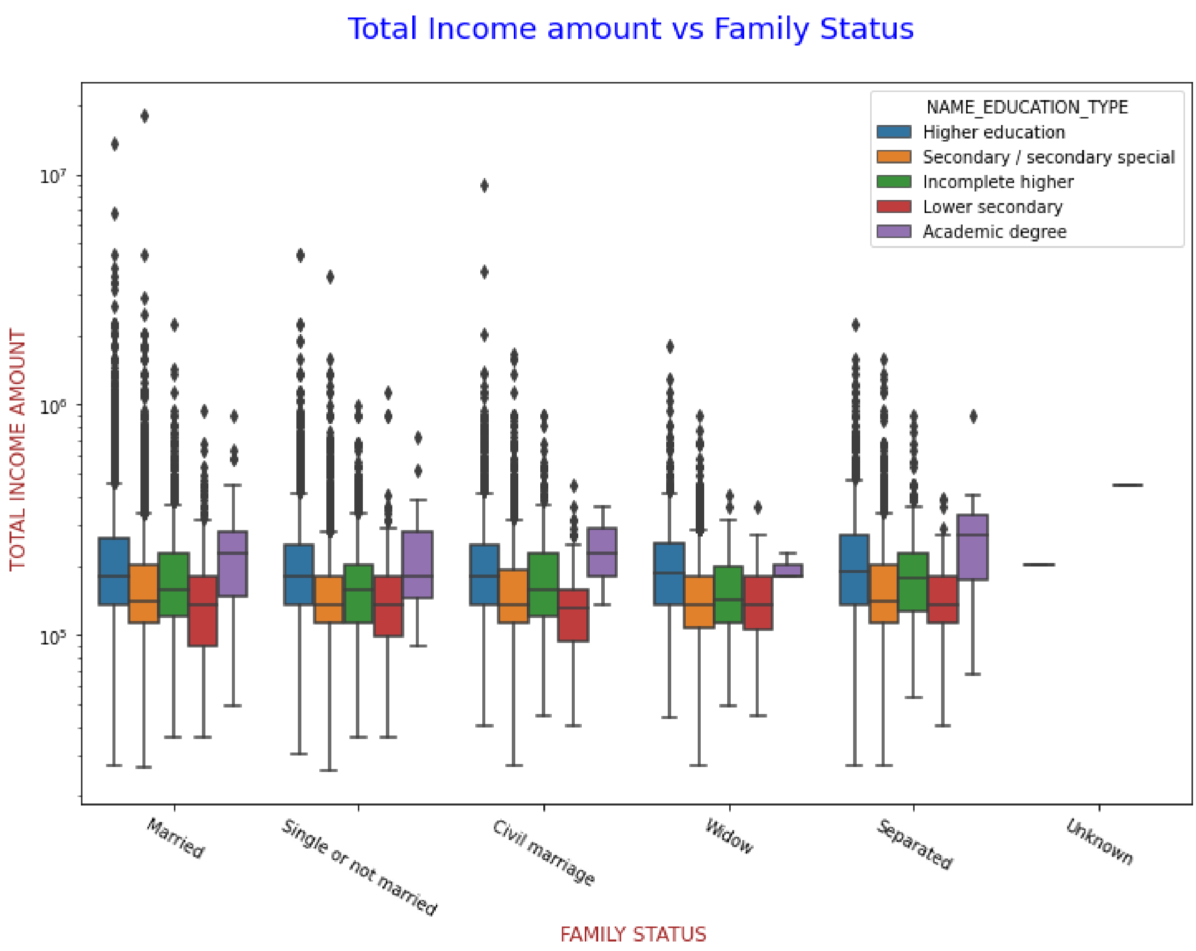


- Family status of 'civil marriage', 'marriage' and 'separated' of Academic degree education are having higher number of credits than others.
- Also, higher education of family status of 'marriage', 'single or not' and 'civil marriage' are having more outliers. Civil marriage for Academic degree is having most of the credits in the third quartile.

- Income of customers based on their family types and education status

```
plt.figure(figsize=(12,8))
scale_factor=5
sns.boxplot(data=target0,x=target0.NAME_FAMILY_STATUS,y=target0.AMT_INCOME_
TOTAL,hue=target0.NAME_EDUCATION_TYPE)
plt.xticks(rotation=-30)
plt.xlabel("FAMILY STATUS ", fontdict={'fontsize': 12, 'fontweight' : 5, 'c
olor' : 'Brown'})
plt.ylabel("TOTAL INCOME AMOUNT ", fontdict={'fontsize': 12, 'fontweight' :
5, 'color' : 'Brown'})
plt.yscale('log')
```

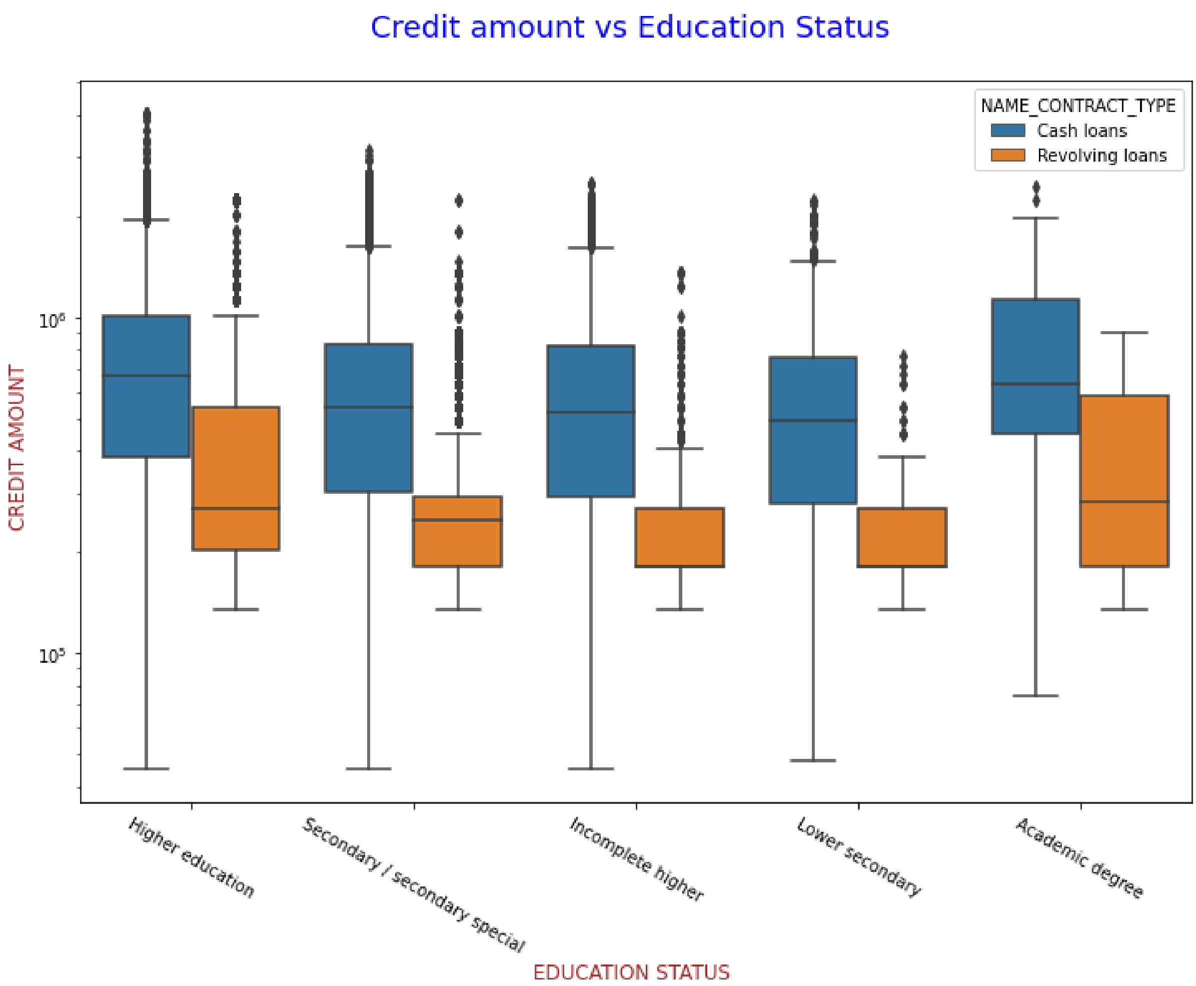
```
plt.title('Total Income amount vs Family Status \n',fontdict={'fontsize': 1
8, 'fontweight' : 10, 'color' : 'Blue'})
plt.show()
```



- Family status of 'civil marriage', 'marriage' and 'separated' of Higher education are having higher number of income than others.
- Also, higher education and secondary/second special education statuses with family status of 'marriage', 'single or not' and 'civil marriage' are having more outliers. Married for Higher education is having most of the incomes in the lower bound

- Credit amount vs education status

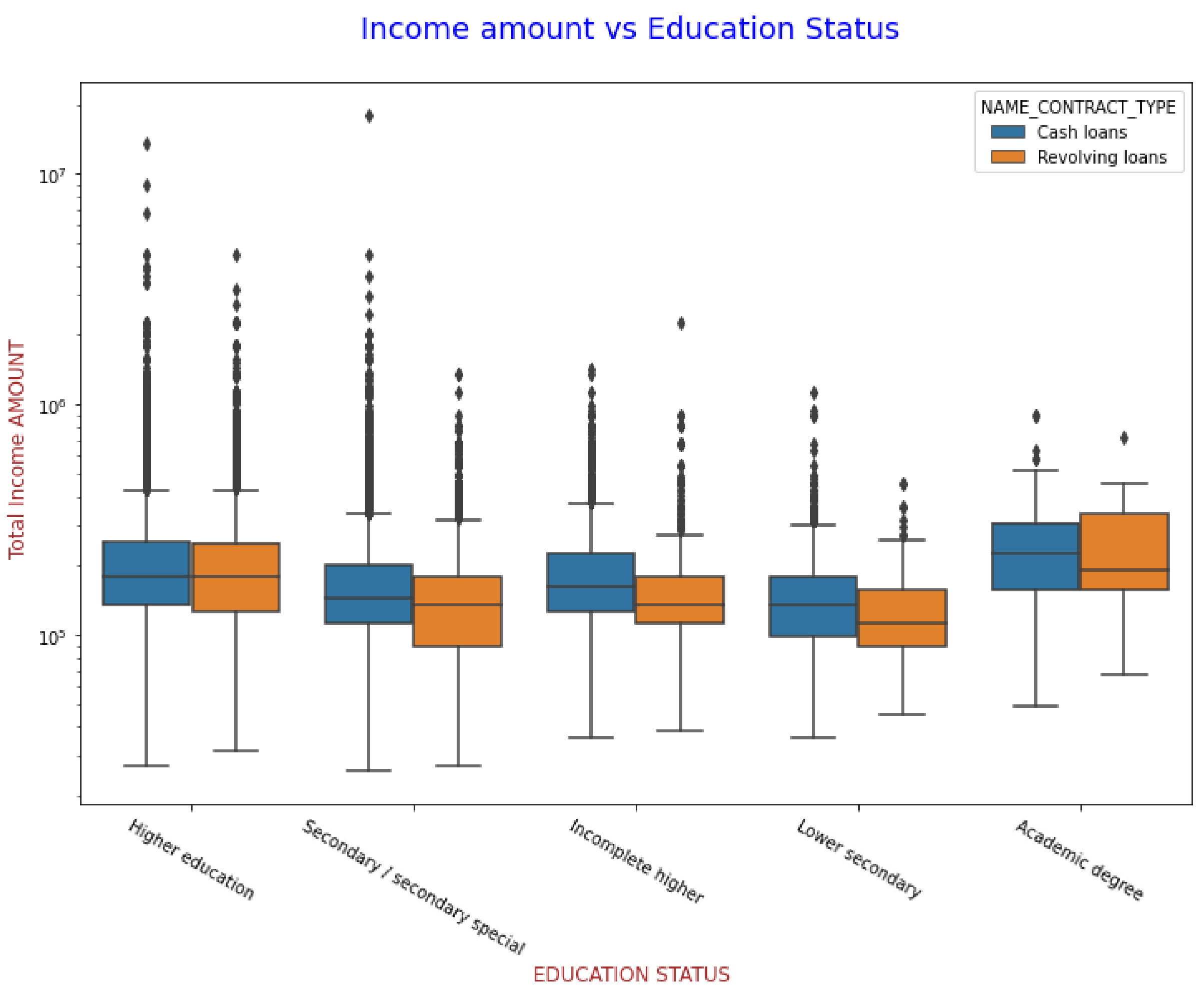
```
plt.figure(figsize=(12,8))
scale_factor=5
sns.boxplot(data=target0,x=target0.NAME_EDUCATION_TYPE,y=target0.AMT_CREDIT
,hue=target0.NAME_CONTRACT_TYPE)
plt.xticks(rotation=-30)
plt.xlabel("EDUCATION STATUS ", fontdict={'fontsize': 12, 'fontweight' : 5,
'color' : 'Brown'})
plt.ylabel("CREDIT AMOUNT ", fontdict={'fontsize': 12, 'fontweight' : 5, 'c
olor' : 'Brown'})
plt.yscale('log')
plt.title('Credit amount vs Education Status \n',fontdict={'fontsize': 18,
'fontweight' : 10, 'color' : 'Blue'})
plt.show()
```



- Education status of higher education and secondary/secondary special have most clients for contract type cash loans
- Most number of clients applying for revolving loans are in education status Acadamic degree and higher education

- Income amount vs education status

```
plt.figure(figsize=(12,8))
scale_factor=5
sns.boxplot(data=target0,x=target0.NAME_EDUCATION_TYPE,y=target0.AMT_INCOME
_TOTAL,hue=target0.NAME_CONTRACT_TYPE)
plt.xticks(rotation=-30)
plt.xlabel("EDUCATION STATUS ", fontdict={'fontsize': 12, 'fontweight' : 5,
'color' : 'Brown'})
plt.ylabel("Total Income AMOUNT ", fontdict={'fontsize': 12, 'fontweight' :
5, 'color' : 'Brown'})
plt.yscale('log')
plt.title('Income amount vs Education Status \n',fontdict={'fontsize': 18,
'fontweight' : 10, 'color' : 'Blue'})
plt.show()
```

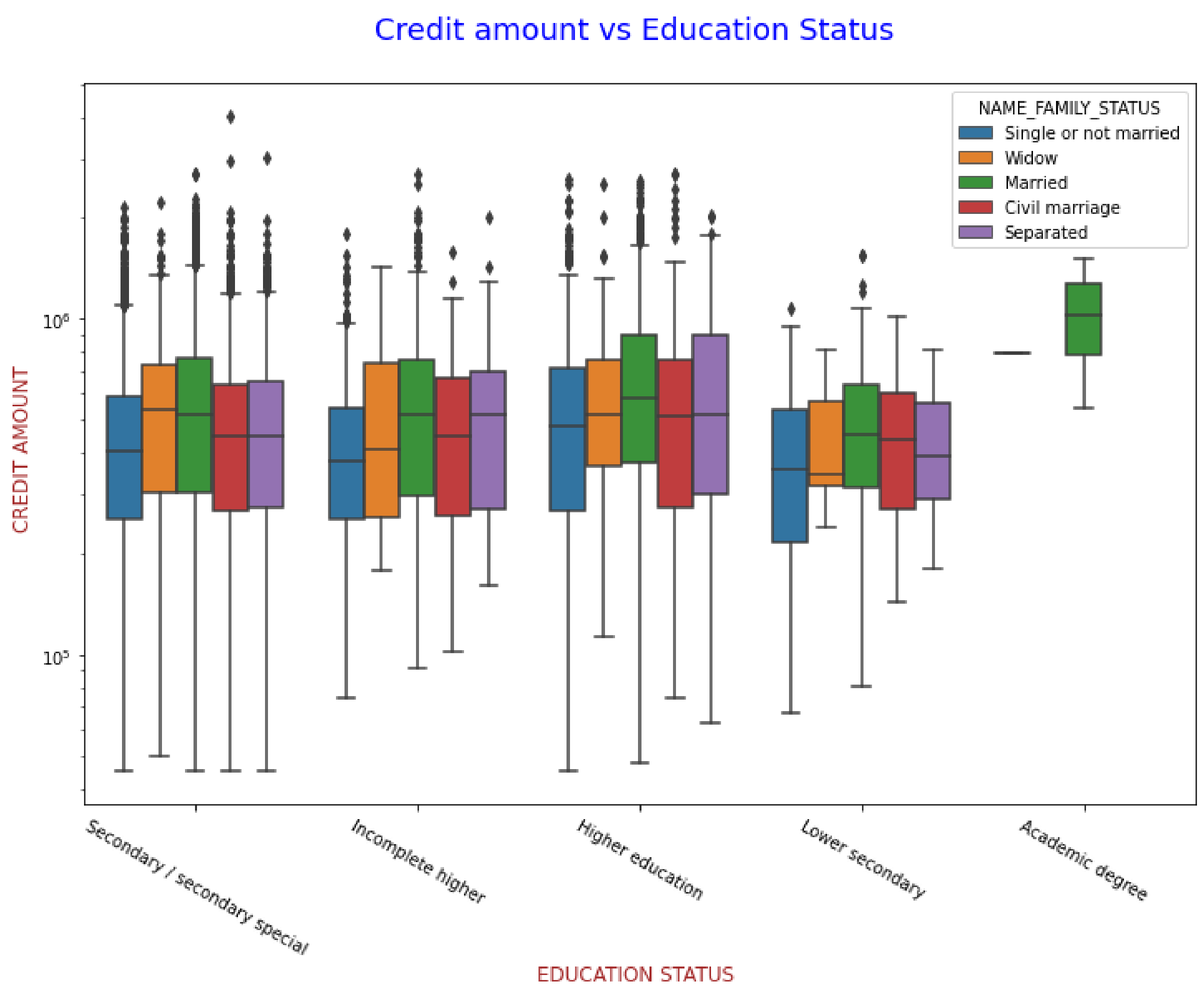


- It can be seen that the clients with education status of Higher education are the maximum credit seekers with highest in terms of contract type of cash loans of contract type

- It can also be observed that the contract type revolving loans issued maximum credit amount holders education status Higher education
- The highest credit amount in cash loans is given to a client with education level secondary/secondary special. Basically education level or type is not playing much role as of who gets what amount of credit

- Credit amount vs education status

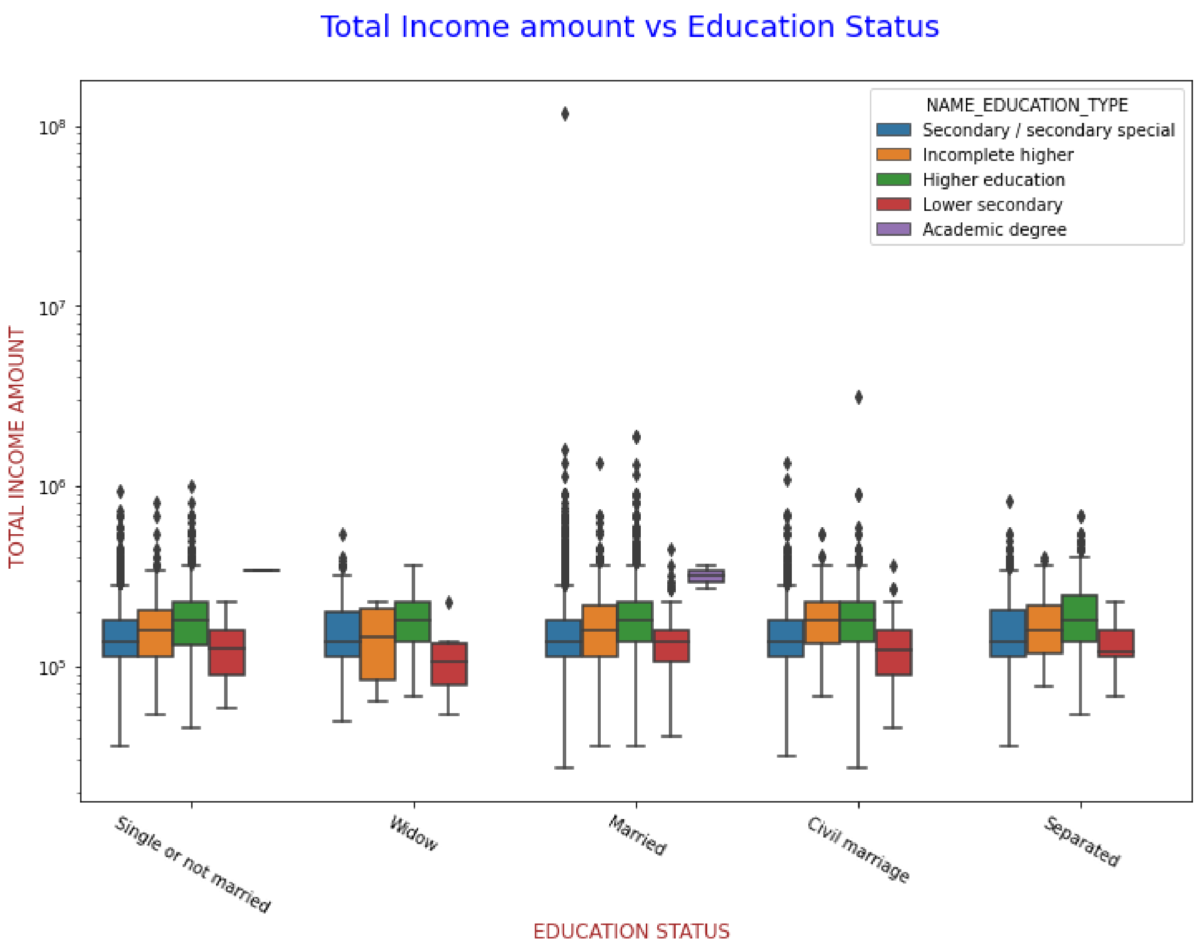
```
plt.figure(figsize=(12,8))
scale_factor=5
sns.boxplot(data=target1,x=target1.NAME_EDUCATION_TYPE,y=target1.AMT_CREDIT
,hue=target1.NAME_FAMILY_STATUS)
plt.xticks(rotation=-30)
plt.xlabel("EDUCATION STATUS ", fontdict={'fontsize': 12, 'fontweight' : 5,
'color' : 'Brown'})
plt.ylabel("CREDIT AMOUNT ", fontdict={'fontsize': 12, 'fontweight' : 5, 'c
olor' : 'Brown'})
plt.yscale('log')
plt.title('Credit amount vs Education Status \n',fontdict={'fontsize': 18,
'fontweight' : 10, 'color' : 'Blue'})
plt.show()
```



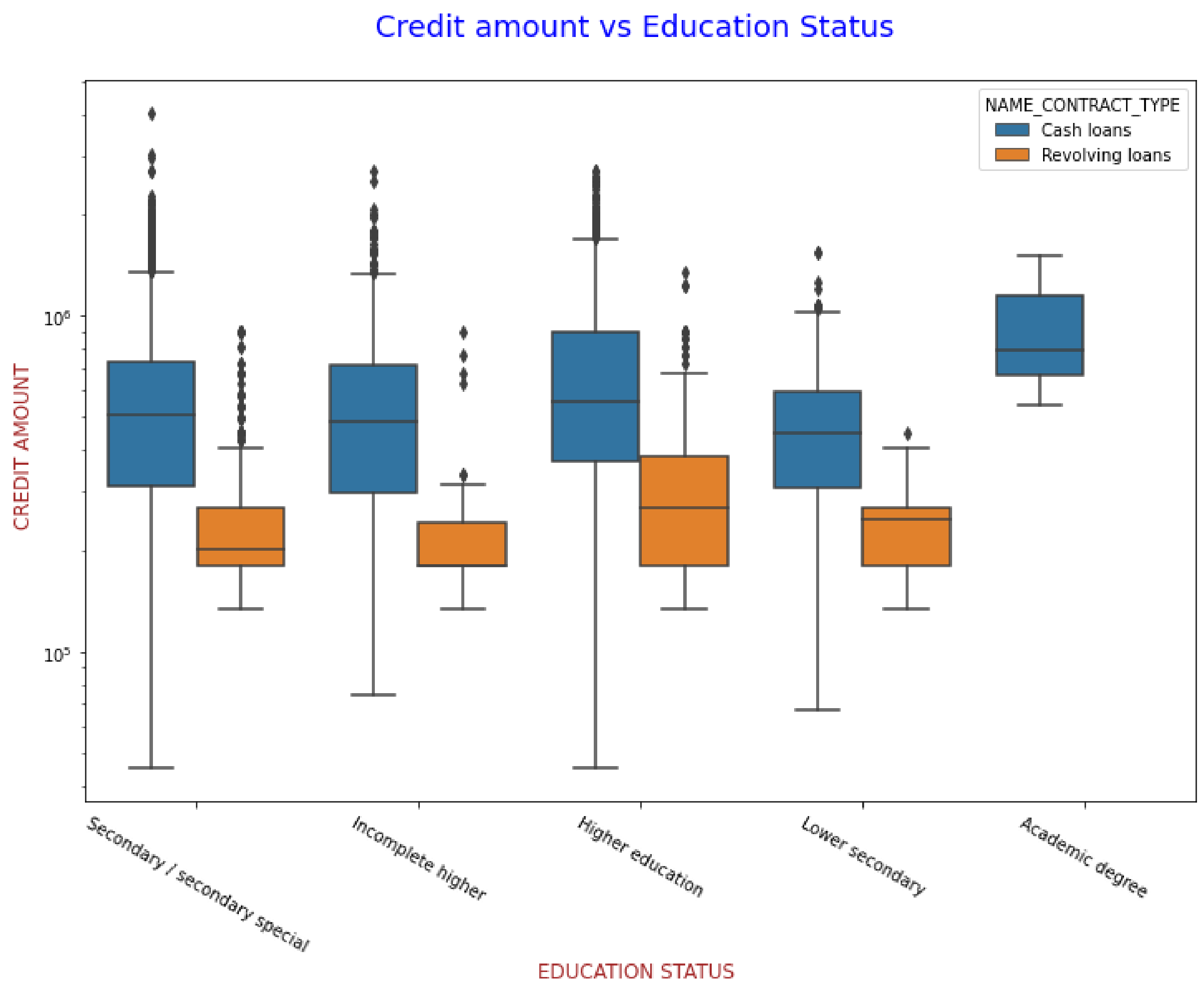
- Family status of 'civil marriage', 'marriage' and 'separated' of Academic degree education are --having less number of credits than others.
- Most of the outliers are from Education type 'Higher education' and 'Secondary'.
- Most number of all types of education as well as family lie in lower bound

- Total amount vs education status

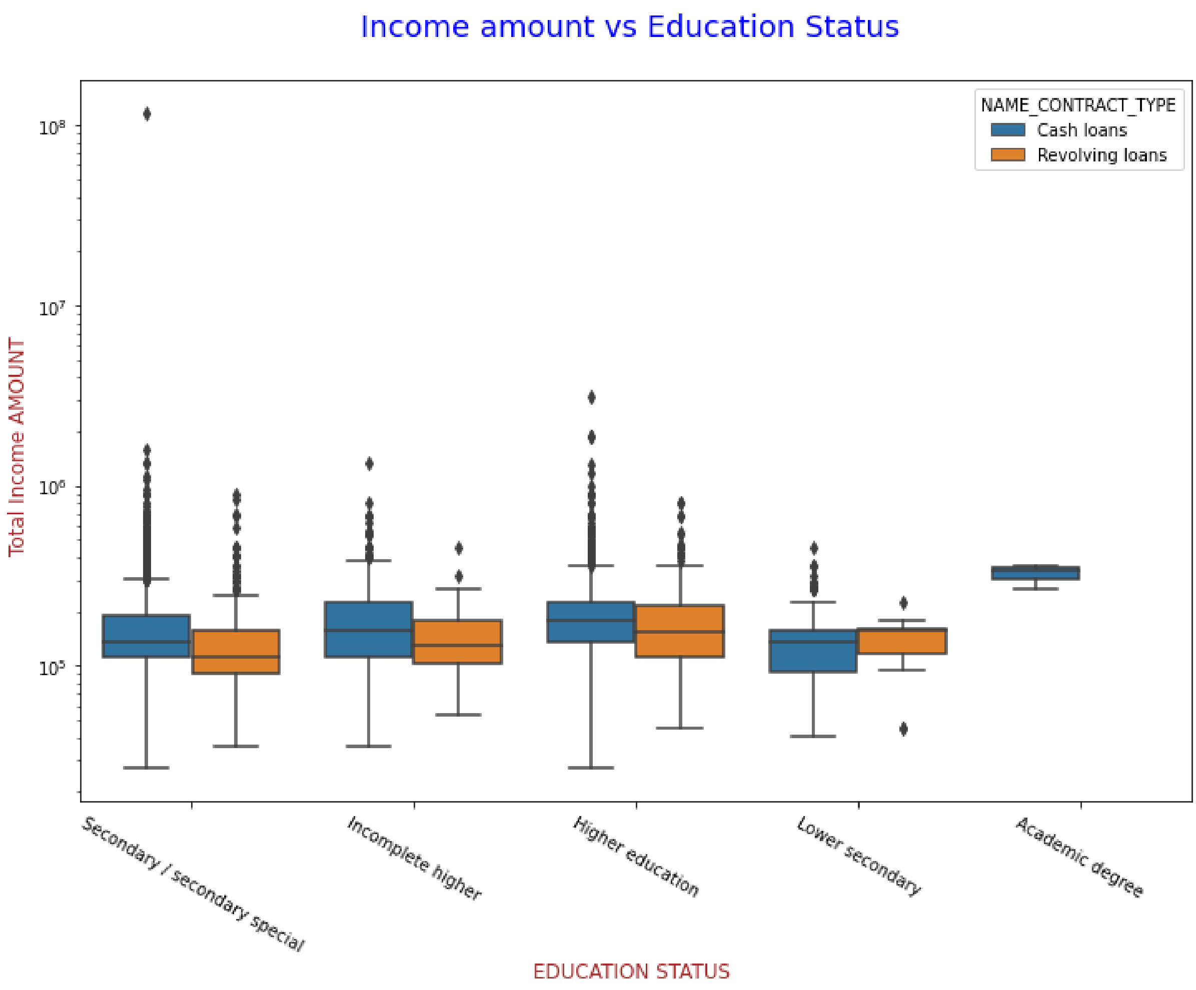
```
plt.figure(figsize=(12,8))
scale_factor=5
sns.boxplot(data=target1,x=target1.NAME_FAMILY_STATUS,y=target1.AMT_INC
OME_TOTAL,hue=target1.NAME_EDUCATION_TYPE)
plt.xticks(rotation=-30)
plt.xlabel("EDUCATION STATUS ", fontdict={'fontsize': 12, 'fontweight'
: 5, 'color' : 'Brown'})
plt.ylabel("TOTAL INCOME AMOUNT ", fontdict={'fontsize': 12, 'fontweigh
t' : 5, 'color' : 'Brown'})
plt.yscale('log')
plt.title('Total Income amount vs Education Status \n',fontdict={'fonts
ize': 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.show()
```



- Almost all Family types and education types have the income amount mostly equal.
- Least outlies are for Lower scondary and their income amount is also little lesser than that of all other education types.
- Academic degree are very less number of people in payments with diffuclt dataframe named target1



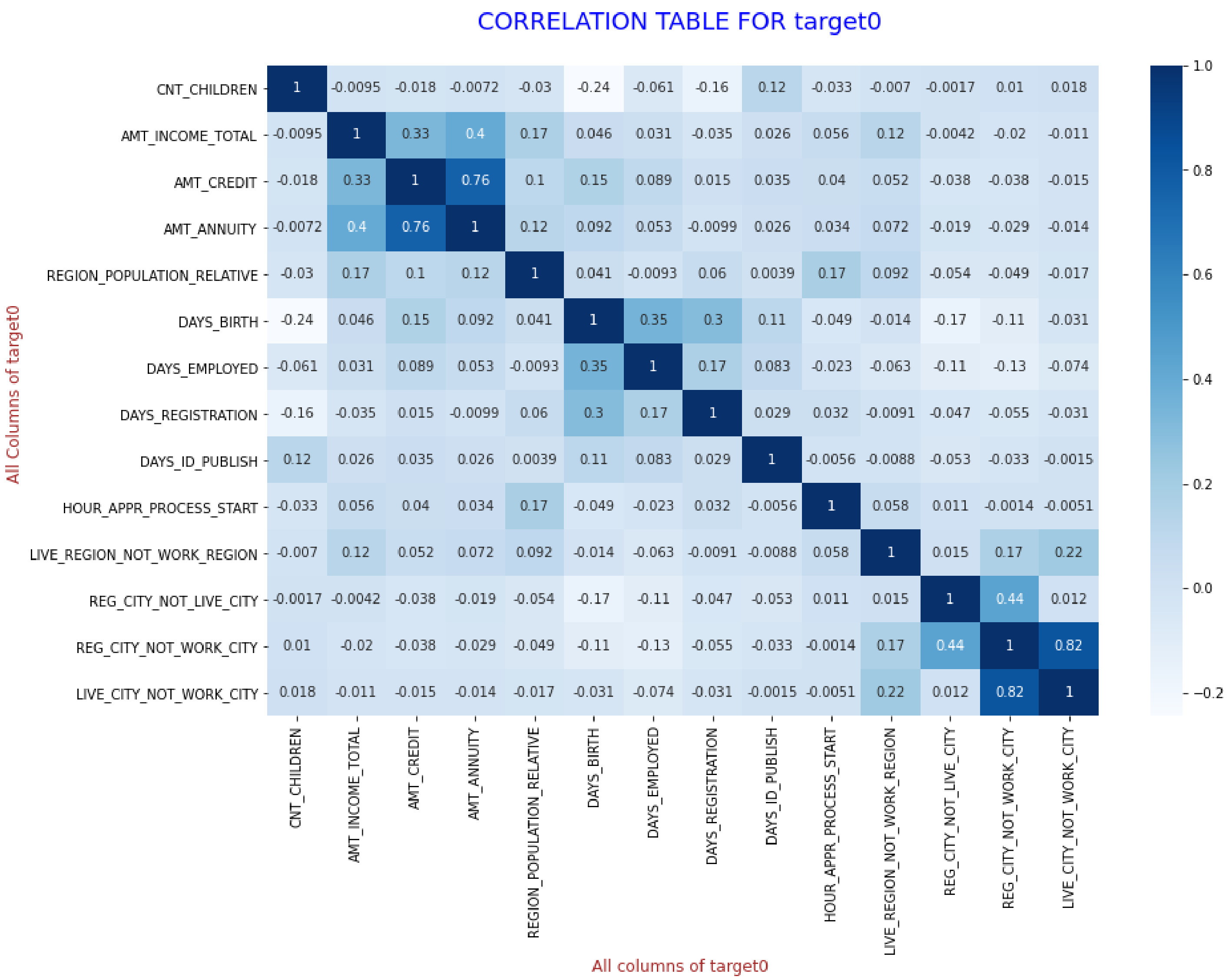
Credit amount vs Education Status
NAME_CONTRACT_TYPE
Cash loans
Revolving loans
CREDIT AMOUNT
10^6
10^5
Secondary / secondary special
Incomplete higher
Higher education
Lower secondary
Academic degree
EDUCATION STATUS



- It can be seen that the maximum income amount holders are with education levels secondary and higher educatin levels
- its strange that, there are no revolving amount loans are demanded by clients with educatin level acadamic degree

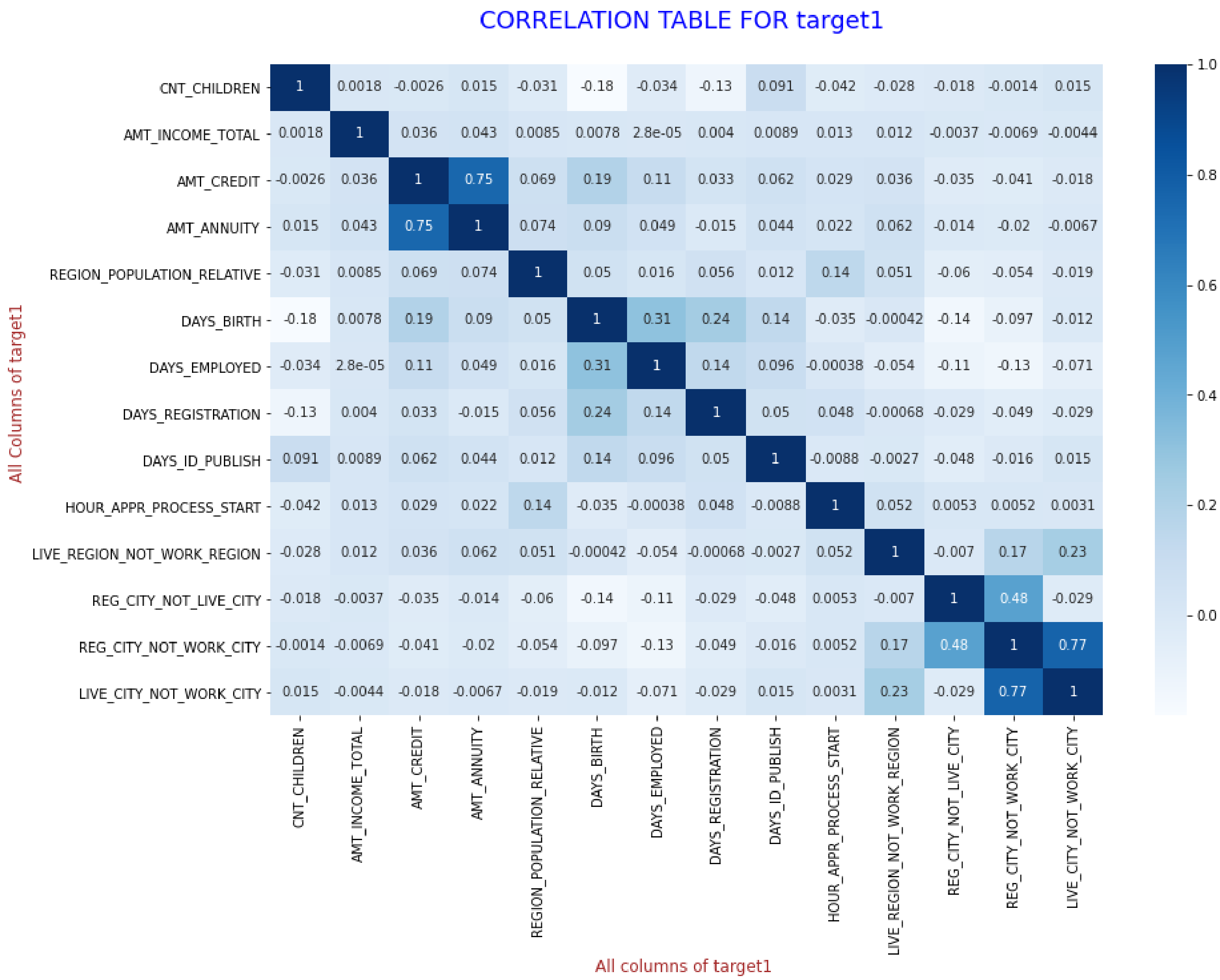
- **Correlation Analysis**

```
f, ax = plt.subplots(figsize=(14, 9))
sns.heatmap(target0_correlation,cmap='Blues',annot=True)
plt.title('CORRELATION TABLE FOR target0 \n',fontdict={'fontsize':18, '
fontweight' : 10, 'color' : 'Blue'})
plt.xlabel("All columns of target0 ", fontdict={'fontsize': 12, 'fontwe
ight' : 5, 'color' : 'Brown'})
plt.ylabel("All Columns of target0 ", fontdict={'fontsize': 12, 'fontwe
ight' : 5, 'color' : 'Brown'})
plt.show()
```



- It is observed that the maximum correlation is between the variables listed below in pairs
    - AMT_CREDIT to AMT_ANNUITY
    - AMT_CREDIT to AMT_TOTAL_INCOME
    - AMT_ANNUITY to AMT_INCOME_TOTAL
    - REG_CITY_NOT_WORK_CITY to REG_CITY_NOT_LIVE_CITY
    - DAYS_EMPLOYED to DAYS_BIRTH
    - DAYS_BIRTH to DAYS_REGISTRATION
- Least or negative correlation is between the variables listed below in pairs
    - DAYS_BIRTH to CNT_CHILDREN

```python
f, ax = plt.subplots(figsize=(14, 9))
sns.heatmap(target1_correlation, cmap='Blues',annot=True)
plt.title('CORRELATION TABLE FOR target1 \n',fontdict={'fontsize': 18,
'fontweight' : 10, 'color' : 'Blue'})
plt.xlabel("All columns of target1 ", fontdict={'fontsize': 12, 'fontwe
ight' : 5, 'color' : 'Brown'})
plt.ylabel("All Columns of target1 ", fontdict={'fontsize': 12, 'fontwe
ight' : 5, 'color' : 'Brown'})
plt.show()
```



- It is observed that the maximum correlation is between the variables listed below in pairs
  - AMT_CREDIT to AMT_ANNUITY
  - AMT_CREDIT to AMT_TOTAL_INCOME

    - AMT_ANNUITY to AMT_INCOME_TOTAL
    - REG_CITY_NOT_WORK_CITY to REG_CITY_NOT_LIVE_CITY
    - DAYS_EMPLOYED to DAYS_BIRTH
    - DAYS_BIRTH to DAYS_REGISTRATION
- Least or negative correlation is between the variables listed below in pairs
    - DAYS_BIRTH to CNT_CHILDREN

- **Loading file previous application and finding null values**

```
preloanapp.isnull().sum()/len(preloanapp)*100
```

```
SK_ID_PREV                      0.000000
SK_ID_CURR                      0.000000
NAME_CONTRACT_TYPE              0.000000
AMT_ANNUITY                    22.286665
AMT_APPLICATION                 0.000000
AMT_CREDIT                      0.000060
AMT_DOWN_PAYMENT               53.636480
AMT_GOODS_PRICE                23.081773
WEEKDAY_APPR_PROCESS_START      0.000000
HOUR_APPR_PROCESS_START         0.000000
FLAG_LAST_APPL_PER_CONTRACT     0.000000
NFLAG_LAST_APPL_IN_DAY          0.000000
RATE_DOWN_PAYMENT              53.636480
RATE_INTEREST_PRIMARY          99.643698
RATE_INTEREST_PRIVILEGED       99.643698
NAME_CASH_LOAN_PURPOSE          0.000000
NAME_CONTRACT_STATUS            0.000000
DAYS_DECISION                   0.000000
NAME_PAYMENT_TYPE               0.000000
CODE_REJECT_REASON              0.000000
NAME_TYPE_SUITE                49.119754
NAME_CLIENT_TYPE                0.000000
NAME_GOODS_CATEGORY             0.000000
NAME_PORTFOLIO                  0.000000
NAME_PRODUCT_TYPE               0.000000
CHANNEL_TYPE                    0.000000
SELLERPLACE_AREA                0.000000
NAME_SELLER_INDUSTRY            0.000000
CNT_PAYMENT                    22.286366
NAME_YIELD_GROUP                0.000000
PRODUCT_COMBINATION             0.020716
DAYS_FIRST_DRAWING             40.298129
DAYS_FIRST_DUE                 40.298129
DAYS_LAST_DUE_1ST_VERSION      40.298129
DAYS_LAST_DUE                  40.298129
DAYS_TERMINATION               40.298129
NFLAG_INSURED_ON_APPROVAL      40.298129
dtype: float64
```

- Dropping columns with more than 30% null values

- Merging the 2 dataframes for further analysis

```
mergedloandf=loanapp.merge(preloanapp,on='SK_ID_CURR')
```

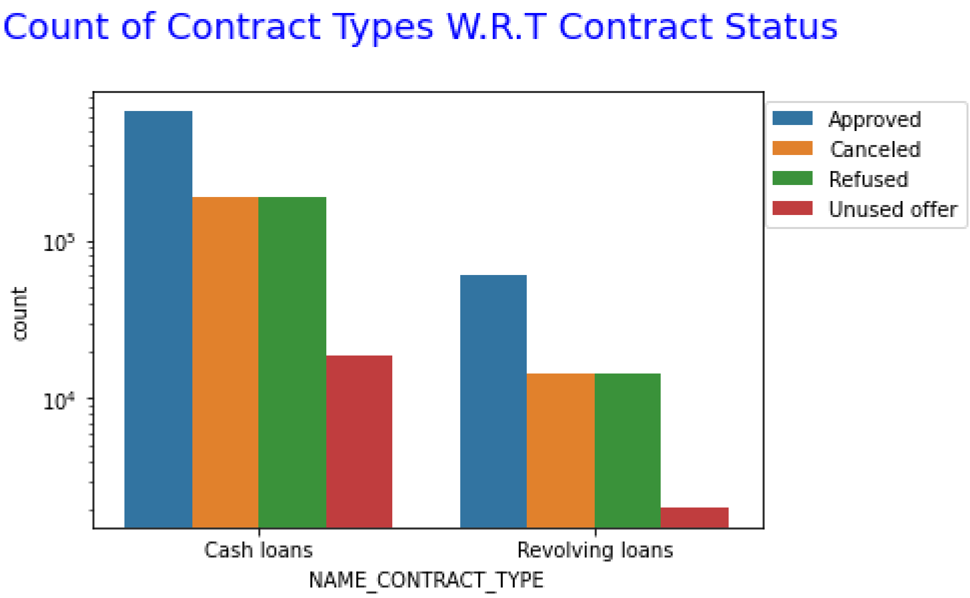
Out[91]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AM
0	100002	1	Cash loans	M	N	Y	0	202
1	100003	0	Cash loans	F	N	N	0	270
2	100003	0	Cash loans	F	N	N	0	270
3	100003	0	Cash loans	F	N	N	0	270
4	100004	0	Revolving loans	M	Y	Y	0	675

5 rows × 69 columns

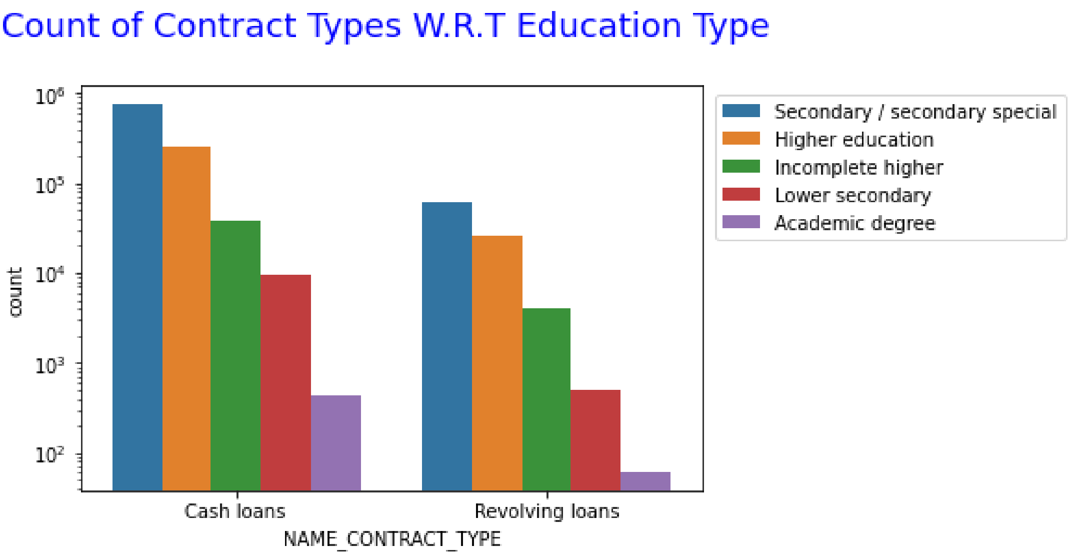
- **Univariate analsis on merged dataframe**

```
sns.countplot(mergedloandf.NAME_CONTRACT_TYPE,hue=mergedloandf.NAME_CONTRAC
T_STATUS)
plt.title('Count of Contract Types W.R.T Contract Status \n',fontdict={'fon
tsize': 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.yscale('log')
plt.legend(bbox_to_anchor=(1.32,1))
plt.show()
```



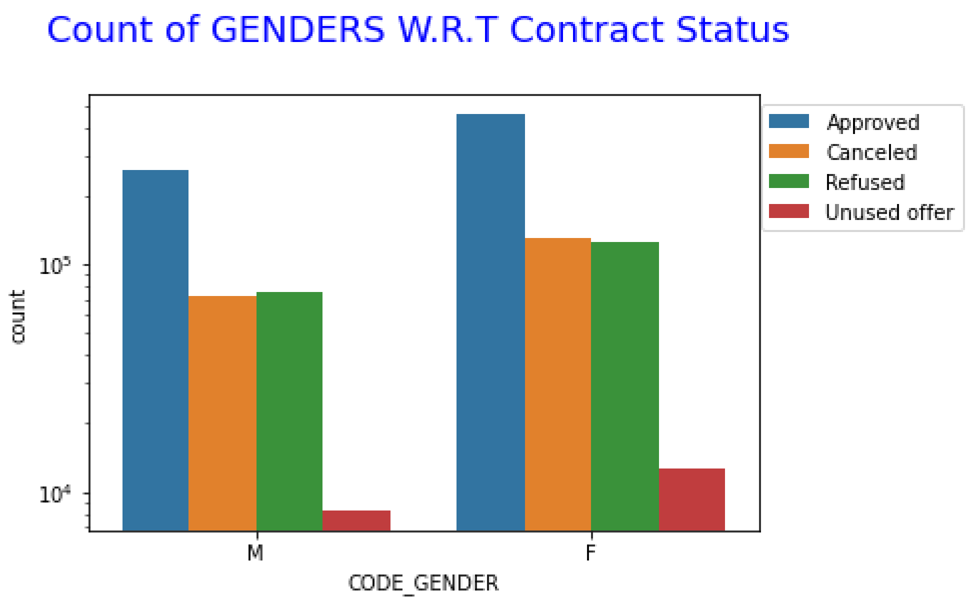
- Contract type cash loans are maximum in number where all kinds of contract statuses are more than contract type revolving loans

```
sns.countplot(mergedloandf.NAME_CONTRACT_TYPE,hue=mergedloandf.NAME_EDUCATION_TYPE)
plt.title('Count of Contract Types W.R.T Education Type \n',fontdict={'fontsize': 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.yscale('log')
plt.legend(bbox_to_anchor=(1.6,1))
plt.show()
```



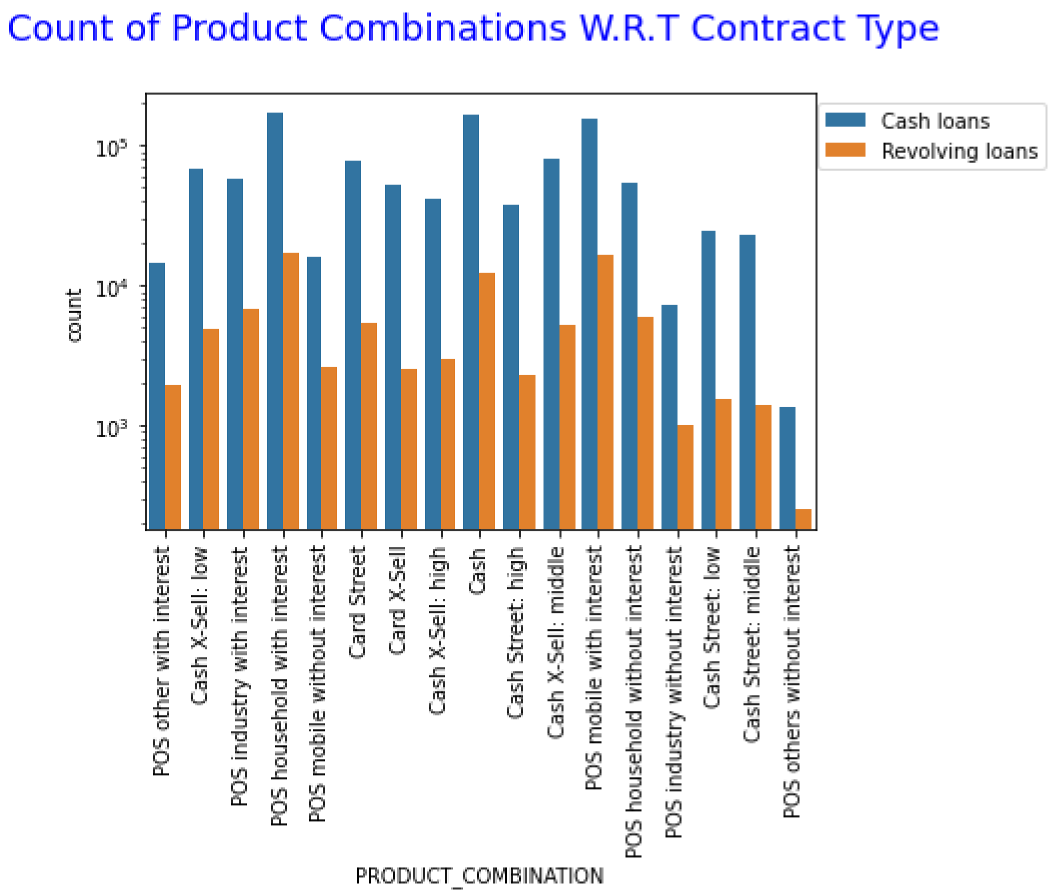
- Most clients with all kinds of education types apply mostly for cash loans rather than revolving loans

```
sns.countplot(mergedloandf.CODE_GENDER,hue=mergedloandf.NAME_CONTRACT_STATUS)
plt.yscale('log')
plt.title('Count of GENDERS W.R.T Contract Status \n',fontdict={'fontsize': 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.legend(bbox_to_anchor=(1.32,1))
plt.show()
```



- Maximum approved loans are for female clients
- It can also be observed that male clients use most of the offers of loans as unused offers are very less for male clients than that of female clients

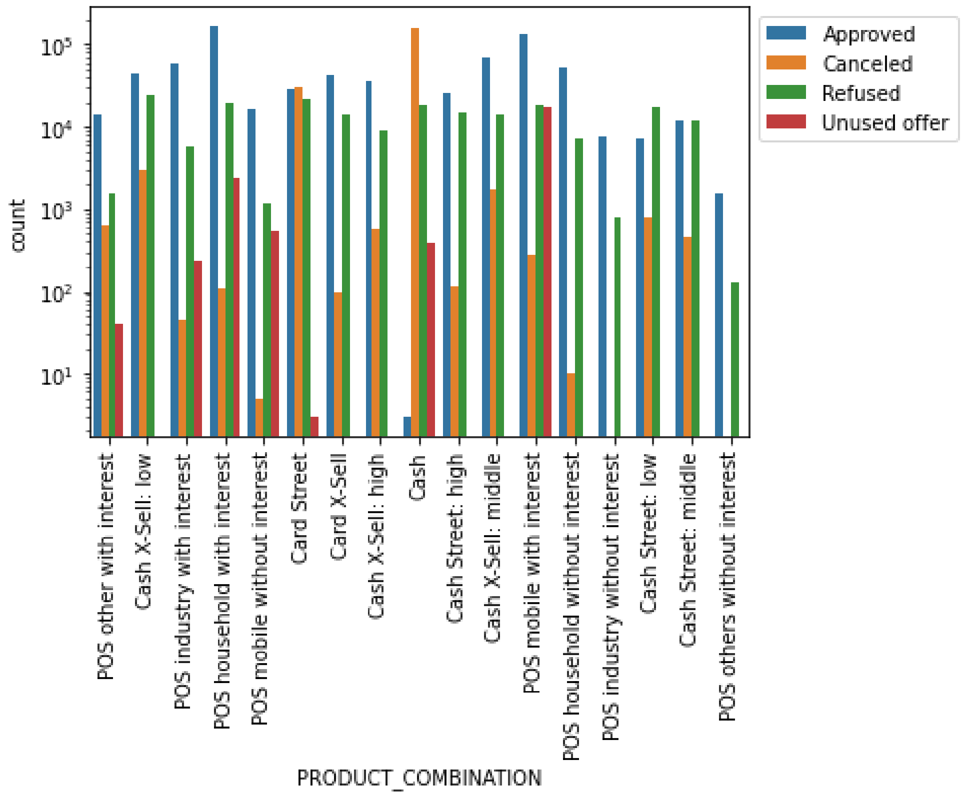
```
sns.countplot(mergedloandf.PRODUCT_COMBINATION,hue=mergedloandf.NAME_CONTRACT_TYPE)
plt.yscale('log')
plt.title('Count of Product Combinations W.R.T Contract Type \n',fontdict={'fontsize': 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.legend(bbox_to_anchor=(1.36,1))
plt.xticks(rotation=90)
plt.show()
```



- Maximum number of contract types are of Product combination of POS household with interest followed by Cash and then followed by POS mobile with interest
- Least number of clients opt for product combination of POS others without interest

```
sns.countplot(mergedloandf.PRODUCT_COMBINATION,hue=mergedloandf.NAME_CONTRACT_
STATUS)
plt.yscale('log')
plt.title('Count of Product Combinations W.R.T Contract Status \n',fontdict={'
fontsize': 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.legend(bbox_to_anchor=(1,1))
plt.xticks(rotation=90)
plt.show()
```

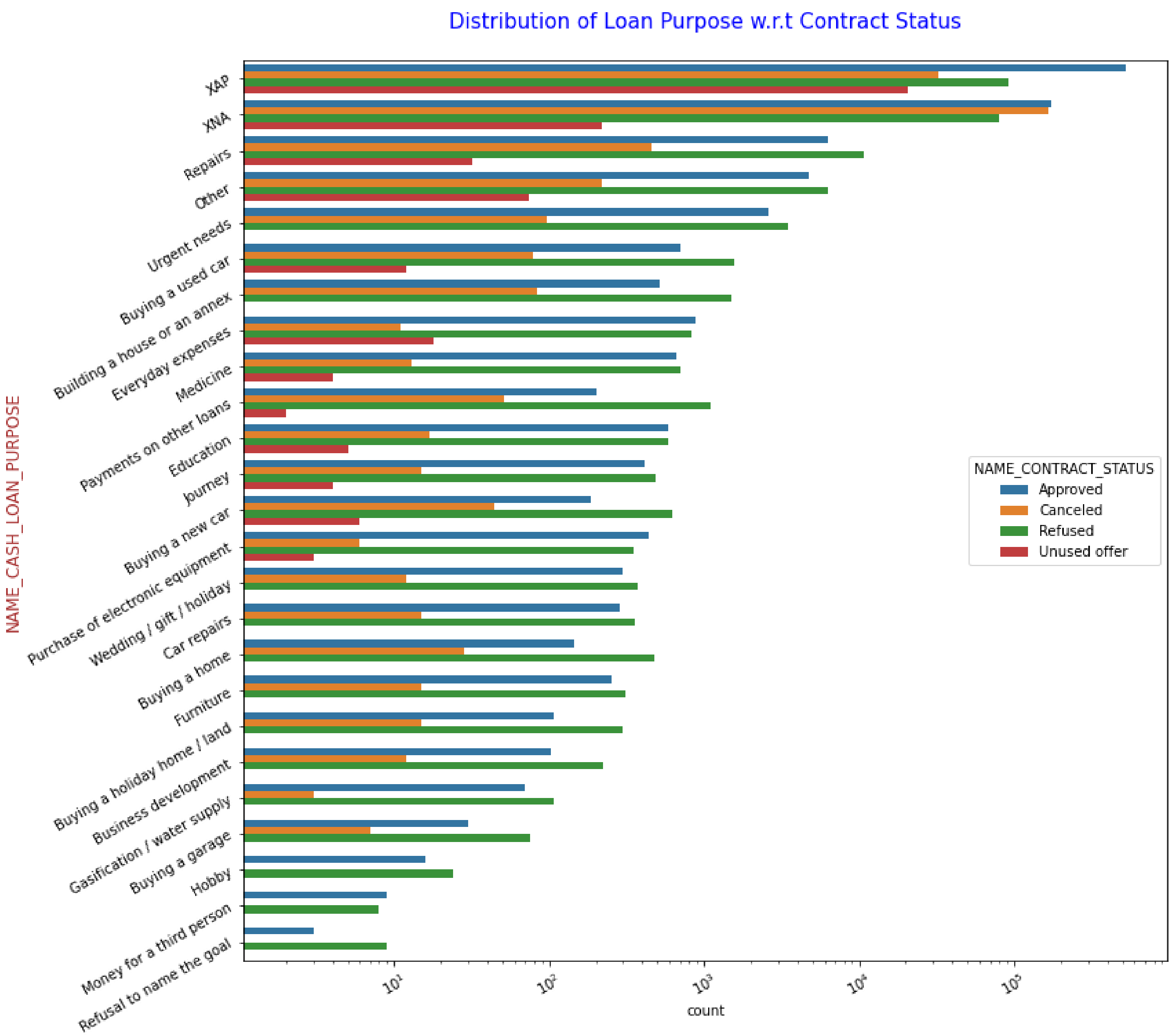
Count of Product Combinations W.R.T Contract Status



- Most canceled laons are of the product combination, Cash.
- Most refused loans are of product combination Cash X-Sell: low
- There are almost nill unused offers in product combinations listed below
    - Cash X-Sell: low
    - Card X-Sell
    - Card X-Sell: high
    - Cash Street: high
    - POS household with interest
    - Cash Street: middle
    - Cash X-Sell: middle
    - Cash Street: low
- Some product combiations are nill incase of unused offers as well as cancelled loans
    - POS industry without interest
    - POS others without interest

- Purpose for applying loan

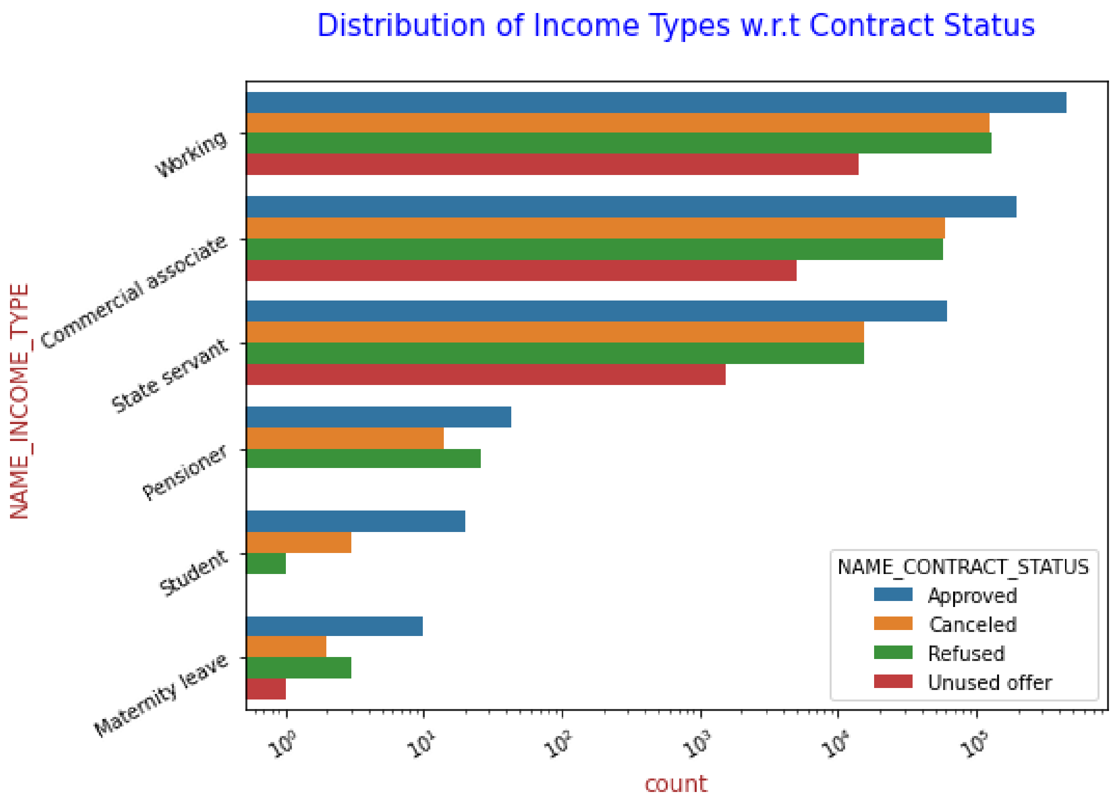
```
plt.figure(figsize=(12,12))
plt.xticks(rotation=30)
plt.xscale('log')
plt.ylabel("NAME_CONTRACT_STATUS", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.yticks(rotation=30)
plt.title('Distribution of Loan Purpose w.r.t Contract Status\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
sns.countplot(data=mergedloandf,y='NAME_CASH_LOAN_PURPOSE',order=mergedloandf['NAME_CASH_LOAN_PURPOSE'].value_counts().index,hue='NAME_CONTRACT_STATUS')
plt.show()
```



- Most rejection of loans came from purpose 'repairs'.
- For education purposes we have equal number of approves
- Rejection for paying other loans and buying a new car are having significant higher rejection than approves.

- Which type of income people are applying for loans

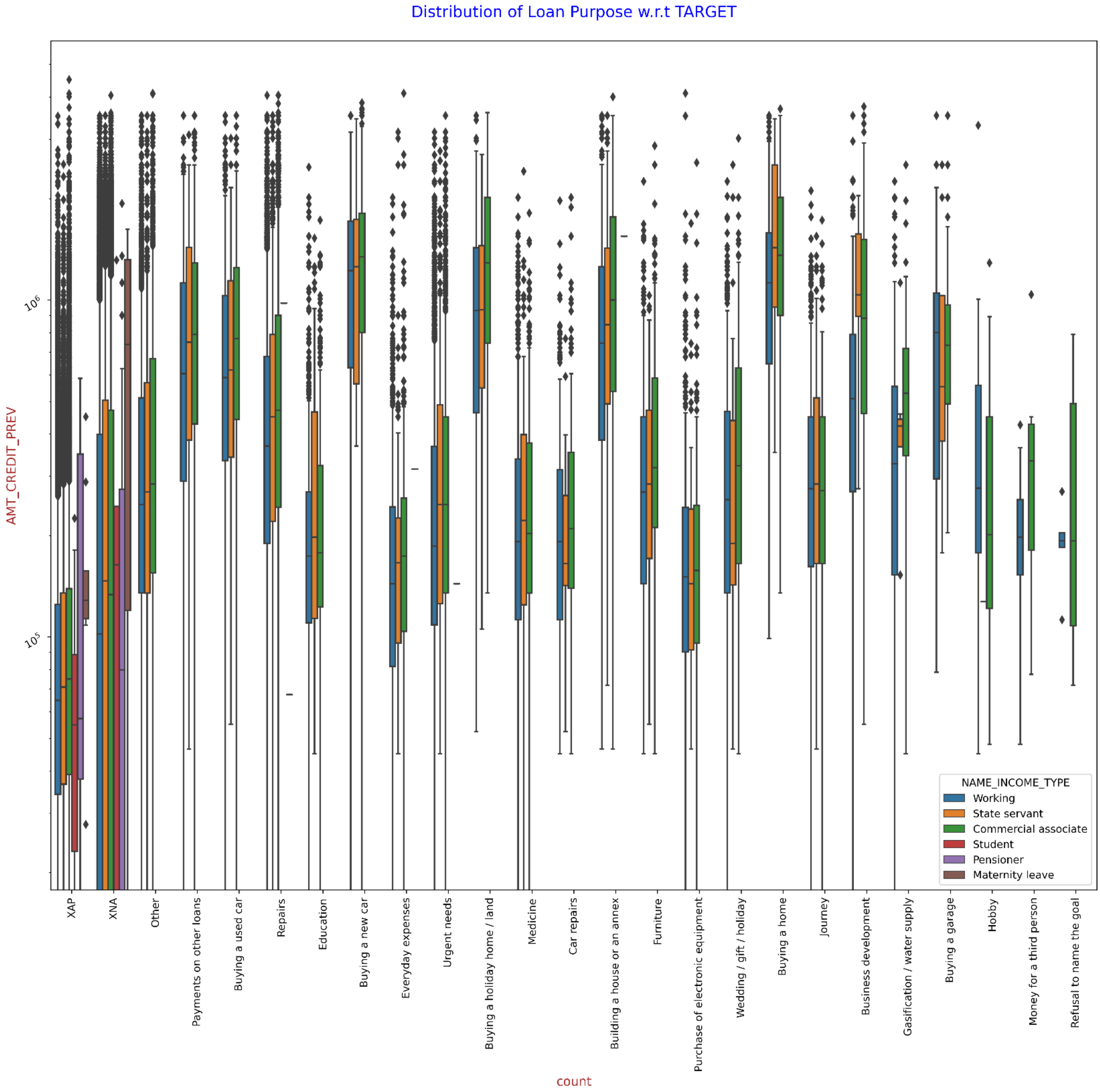
```
plt.figure(figsize=(8,6))
plt.xticks(rotation=30)
plt.xscale('log')
plt.ylabel("NAME_INCOME_TYPE", fontdict={'fontsize': 12, 'fontweight' :
5, 'color' : 'Brown'})
plt.xlabel("count", fontdict={'fontsize': 12, 'fontweight' : 5, 'color'
: 'Brown'})
plt.yticks(rotation=30)
plt.title('Distribution of Income Types w.r.t Contract Status\n',fontdi
ct={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
sns.countplot(data=mergedloandf,y='NAME_INCOME_TYPE',order=mergedloandf
['NAME_INCOME_TYPE'].value_counts().index,hue='NAME_CONTRACT_STATUS')
plt.show()
```



- There are no used offers for students and pensioners
- The number of approved loans for state servants is almost equal to the refusal or cancelled loans for Commercial associates
- Maximum unused offers is by working clients

- **Bivariate analysis**

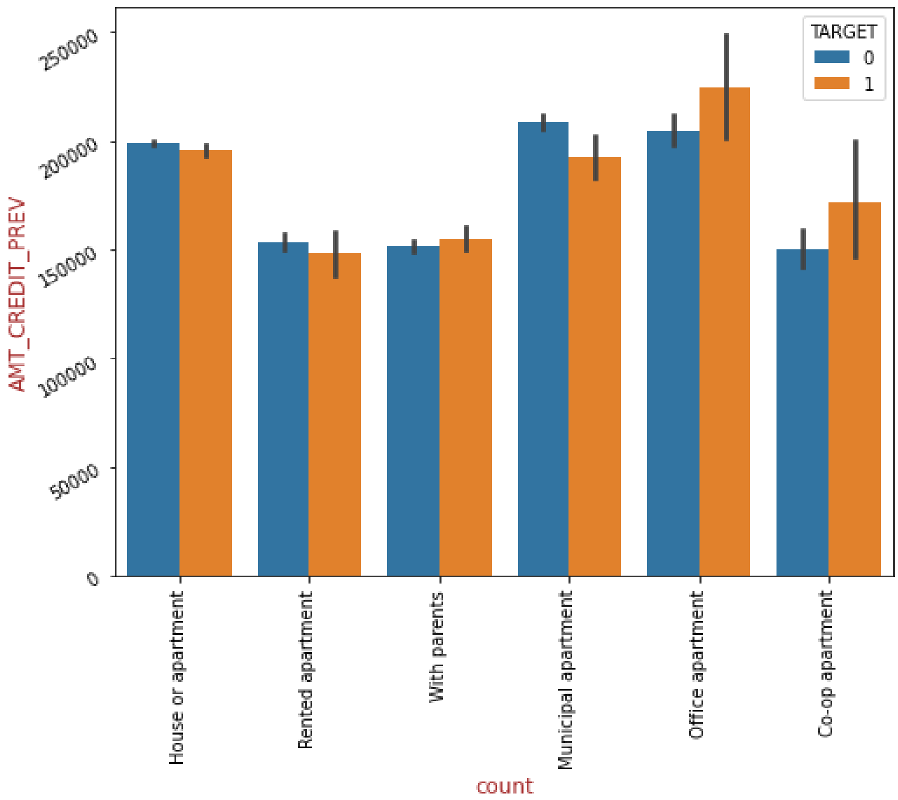
```
plt.figure(figsize=(18,15),dpi=400)
sns.boxplot(data=mergedloandf,y='AMT_CREDIT_PREV',x='NAME_CASH_LOAN_PURPOSE',hue='NAME_INCOME_TYPE')
plt.xticks(rotation=90)
plt.yscale('log')
plt.ylabel("AMT_CREDIT_PREV", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.xlabel("count", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Brown'})
plt.yticks(rotation=30)
plt.title('Distribution of Loan Purpose w.r.t TARGET\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
plt.show()
```



- The credit amount of Loan purposes like 'Buying a home', 'Buying a land', 'Buying a new car' and 'Building a house' is higher being a 'commercial associate' and even in general irrespective of income type.
- Income type of state servant's having no amount of credit applied for purpose of Money for third person or a Hobby.

- Most of the commercial associates have refused to provide purpose of credit

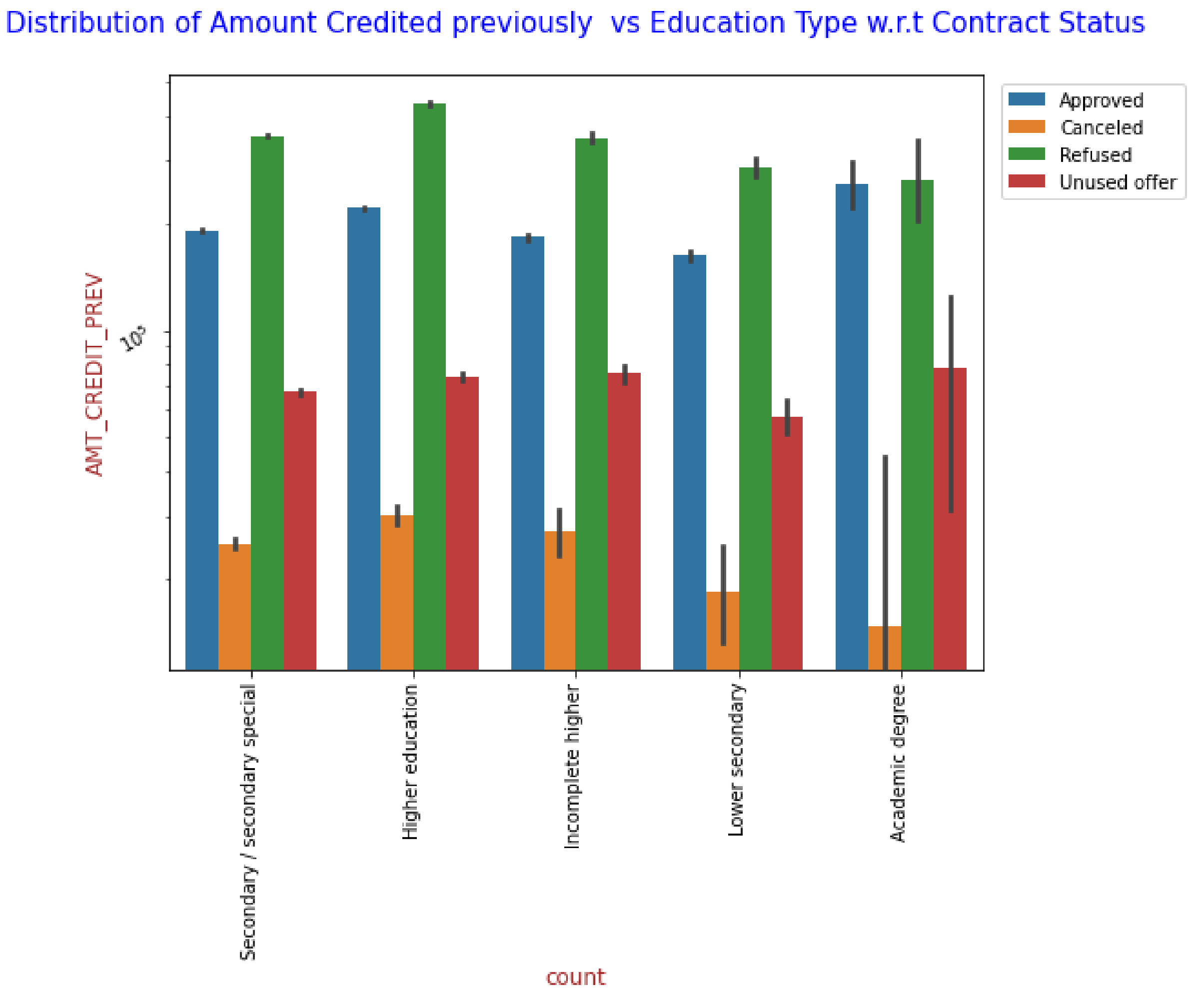
```
plt.figure(figsize=(8,6))
sns.barplot(data=mergedloandf,y='AMT_CREDIT_PREV',hue='TARGET',x='NAME_HOUSING
_TYPE')
plt.xticks(rotation=90)
plt.ylabel("AMT_CREDIT_PREV", fontdict={'fontsize': 12, 'fontweight' : 5, 'col
or' : 'Brown'})
plt.xlabel("count", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : 'Bro
wn'})
plt.yticks(rotation=30)
plt.title('Distribution of Amount Credited previously  vs Housing Type w.r.t T
ARGET\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color' : 'Blue'})
plt.show()
```



- Here for Housing type, office apartment and co-op apartment are having higher credit of target 1.
- So, we can conclude that bank should avoid giving loans to the housing type of office apartment and co-op apartment as they are having difficulties in payment.

- Bank can focus mostly on housing type with parents or House or apartment or municipal apartment for successful payments.

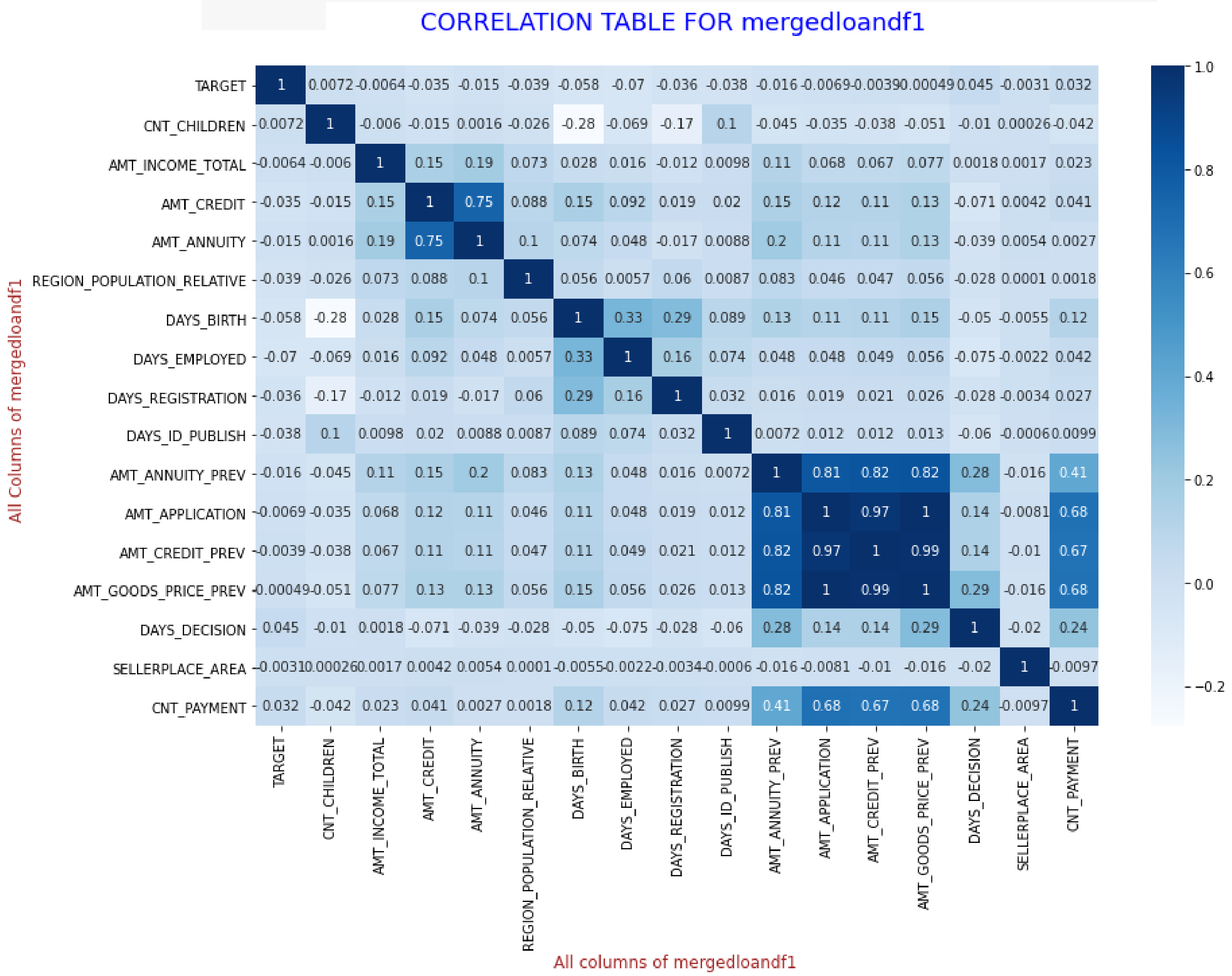
```
plt.figure(figsize=(8,6))
sns.barplot(data=mergedloandf,y='AMT_CREDIT_PREV',hue='NAME_CONTRACT_STATUS
',x='NAME_EDUCATION_TYPE')
plt.xticks(rotation=90)
plt.ylabel("AMT_CREDIT_PREV", fontdict={'fontsize': 12, 'fontweight' : 5, '
color' : 'Brown'})
plt.xlabel("count", fontdict={'fontsize': 12, 'fontweight' : 5, 'color' : '
Brown'})
plt.yticks(rotation=30)
plt.yscale('log')
plt.legend(bbox_to_anchor=(1.26,1))
plt.title('Distribution of Amount Credited previously  vs Education Type w.
r.t Contract Status\n',fontdict={'fontsize': 15, 'fontweight' : 7, 'color'
: 'Blue'})
plt.show()
```



- It can be seen that number of refusals are for clients has nothing to do with their education levels
- Almost all education level clients have equal unused offers.

- **Multivariate analysis**

```
f, ax = plt.subplots(figsize=(14, 9))
sns.heatmap(mergedloandfcorrelation, cmap='Blues',annot=True)
plt.title('CORRELATION TABLE FOR mergedloandf1 \n',fontdict={'fontsize'
: 18, 'fontweight' : 10, 'color' : 'Blue'})
plt.xlabel("All columns of mergedloandf1 ", fontdict={'fontsize': 12, '
fontweight' : 5, 'color' : 'Brown'})
plt.ylabel("All Columns of mergedloandf1 ", fontdict={'fontsize': 12, '
fontweight' : 5, 'color' : 'Brown'})
plt.show()
```



Result

- Banks should focus more on contract type ‘Student’ ,’pensioner’ and ‘Businessman’ with housing ‘type other than ‘Co-op apartment’ and 'office apartment' for successful payments.
- Banks should focus less on income types maternity leave and working as they have most number of unsuccessful payments
- Although having higher number of rejection in loan purposes with 'Repairs' we can observe difficulties in payment. There are few places where loan payment difficulty is significantly high. Bank should continue to be cautious while giving loan for this purpose.
- Bank can focus mostly on housing type with parents, House or apartment and municipal apartment with purpose of education, buying land, buying a garage, purchase of electronic equipment and some other purposes with target0 significantly more than target1 for successful payments.
- Banks can offer more offers to clients who are students and pensioners as they take all offers and are more likely to pay back
- This project helped me in understanding the tables at a much-detailed manner and helped to improve my strength in extracting data fromtables in a more efficient manner.