In [4]:

```
# Importing All Libraries.

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

In [5]:

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 200)
```

In [6]:

```
# Read the data set.
applications = pd.read_csv('application_data.csv')
applications.head()
```

Out[6]:

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

In [7]:

```
# describing the data
applications.describe()
```

Out[7]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRI
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e-
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e-
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e-
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e-
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e-
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e-
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e-
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e-

In [8]:

```
# shape of the data
applications.shape
```

Out[8]:

Out[8]:

```
(307511, 122)
```

Data Cleaning (Fix columns, Handle missing values, Handle outliers, Standardize values)

Fixing coulmns

In [9]:

```
applications.info(null_counts=True, verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 122 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                   307499 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   OWN_CAR_AGE                   104582 non-null  float64
 22   FLAG_MOBIL                    307511 non-null  int64
 23   FLAG_EMP_PHONE                307511 non-null  int64
 24   FLAG_WORK_PHONE               307511 non-null  int64
 25   FLAG_CONT_MOBILE              307511 non-null  int64
 26   FLAG_PHONE                    307511 non-null  int64
 27   FLAG_EMAIL                    307511 non-null  int64
 28   OCCUPATION_TYPE               211120 non-null  object
 29   CNT_FAM_MEMBERS               307509 non-null  float64
 30   REGION_RATING_CLIENT          307511 non-null  int64
 31   REGION_RATING_CLIENT_W_CITY   307511 non-null  int64
 32   WEEKDAY_APPR_PROCESS_START    307511 non-null  object
 33   HOUR_APPR_PROCESS_START       307511 non-null  int64
 34   REG_REGION_NOT_LIVE_REGION    307511 non-null  int64
 35   REG_REGION_NOT_WORK_REGION    307511 non-null  int64
 36   LIVE_REGION_NOT_WORK_REGION   307511 non-null  int64
 37   REG_CITY_NOT_LIVE_CITY        307511 non-null  int64
 38   REG_CITY_NOT_WORK_CITY        307511 non-null  int64
 39   LIVE_CITY_NOT_WORK_CITY       307511 non-null  int64
 40   ORGANIZATION_TYPE             307511 non-null  object
 41   EXT_SOURCE_1                  134133 non-null  float64
 42   EXT_SOURCE_2                  306851 non-null  float64
 43   EXT_SOURCE_3                  246546 non-null  float64
 44   APARTMENTS_AVG                151450 non-null  float64
 45   BASEMENTAREA_AVG              127568 non-null  float64
 46   YEARS_BEGINEXPLUATATION_AVG   157504 non-null  float64
 47   YEARS_BUILD_AVG               103023 non-null  float64
 48   COMMONAREA_AVG                92646 non-null   float64
 49   ELEVATORS_AVG                 143620 non-null  float64
 50   ENTRANCES_AVG                 152683 non-null  float64
```

```
 51   FLOORSMAX_AVG                 154491 non-null  float64
 52   FLOORSMIN_AVG                 98869 non-null   float64
 53   LANDAREA_AVG                  124921 non-null  float64
 54   LIVINGAPARTMENTS_AVG          97312 non-null   float64
 55   LIVINGAREA_AVG                153161 non-null  float64
 56   NONLIVINGAPARTMENTS_AVG       93997 non-null   float64
 57   NONLIVINGAREA_AVG             137829 non-null  float64
 58   APARTMENTS_MODE               151450 non-null  float64
 59   BASEMENTAREA_MODE             127568 non-null  float64
 60   YEARS_BEGINEXPLUATATION_MODE  157504 non-null  float64
 61   YEARS_BUILD_MODE              103023 non-null  float64
 62   COMMONAREA_MODE               92646 non-null   float64
 63   ELEVATORS_MODE                143620 non-null  float64
 64   ENTRANCES_MODE                152683 non-null  float64
 65   FLOORSMAX_MODE                154491 non-null  float64
 66   FLOORSMIN_MODE                98869 non-null   float64
 67   LANDAREA_MODE                 124921 non-null  float64
 68   LIVINGAPARTMENTS_MODE         97312 non-null   float64
 69   LIVINGAREA_MODE               153161 non-null  float64
 70   NONLIVINGAPARTMENTS_MODE      93997 non-null   float64
 71   NONLIVINGAREA_MODE            137829 non-null  float64
 72   APARTMENTS_MEDI               151450 non-null  float64
 73   BASEMENTAREA_MEDI             127568 non-null  float64
 74   YEARS_BEGINEXPLUATATION_MEDI  157504 non-null  float64
 75   YEARS_BUILD_MEDI              103023 non-null  float64
 76   COMMONAREA_MEDI               92646 non-null   float64
 77   ELEVATORS_MEDI                143620 non-null  float64
 78   ENTRANCES_MEDI                152683 non-null  float64
 79   FLOORSMAX_MEDI                154491 non-null  float64
 80   FLOORSMIN_MEDI                98869 non-null   float64
 81   LANDAREA_MEDI                 124921 non-null  float64
 82   LIVINGAPARTMENTS_MEDI         97312 non-null   float64
 83   LIVINGAREA_MEDI               153161 non-null  float64
 84   NONLIVINGAPARTMENTS_MEDI      93997 non-null   float64
 85   NONLIVINGAREA_MEDI            137829 non-null  float64
 86   FONDKAPREMONT_MODE            97216 non-null   object
 87   HOUSETYPE_MODE                153214 non-null  object
 88   TOTALAREA_MODE                159080 non-null  float64
 89   WALLSMATERIAL_MODE            151170 non-null  object
 90   EMERGENCYSTATE_MODE           161756 non-null  object
 91   OBS_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 92   DEF_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 93   OBS_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 94   DEF_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 95   DAYS_LAST_PHONE_CHANGE        307510 non-null  float64
 96   FLAG_DOCUMENT_2               307511 non-null  int64
 97   FLAG_DOCUMENT_3               307511 non-null  int64
 98   FLAG_DOCUMENT_4               307511 non-null  int64
 99   FLAG_DOCUMENT_5               307511 non-null  int64
 100  FLAG_DOCUMENT_6               307511 non-null  int64
 101  FLAG_DOCUMENT_7               307511 non-null  int64
 102  FLAG_DOCUMENT_8               307511 non-null  int64
 103  FLAG_DOCUMENT_9               307511 non-null  int64
 104  FLAG_DOCUMENT_10              307511 non-null  int64
 105  FLAG_DOCUMENT_11              307511 non-null  int64
 106  FLAG_DOCUMENT_12              307511 non-null  int64
 107  FLAG_DOCUMENT_13              307511 non-null  int64
 108  FLAG_DOCUMENT_14              307511 non-null  int64
 109  FLAG_DOCUMENT_15              307511 non-null  int64
 110  FLAG_DOCUMENT_16              307511 non-null  int64
 111  FLAG_DOCUMENT_17              307511 non-null  int64
 112  FLAG_DOCUMENT_18              307511 non-null  int64
 113  FLAG_DOCUMENT_19              307511 non-null  int64
 114  FLAG_DOCUMENT_20              307511 non-null  int64
 115  FLAG_DOCUMENT_21              307511 non-null  int64
 116  AMT_REQ_CREDIT_BUREAU_HOUR    265992 non-null  float64
 117  AMT_REQ_CREDIT_BUREAU_DAY     265992 non-null  float64
 118  AMT_REQ_CREDIT_BUREAU_WEEK    265992 non-null  float64
 119  AMT_REQ_CREDIT_BUREAU_MON     265992 non-null  float64
 120  AMT_REQ_CREDIT_BUREAU_QRT     265992 non-null  float64
 121  AMT_REQ_CREDIT_BUREAU_YEAR    265992 non-null  float64
dtypes: float64(65), int64(41), object(16)
```

memory usage: 286.2+ MB

In [10]:

```
# Sum of null values
applications.isnull().sum()
```

Out[10]:

```
SK_ID_CURR                          0
TARGET                              0
NAME_CONTRACT_TYPE                  0
CODE_GENDER                         0
FLAG_OWN_CAR                        0
FLAG_OWN_REALTY                     0
CNT_CHILDREN                        0
AMT_INCOME_TOTAL                    0
AMT_CREDIT                          0
AMT_ANNUITY                        12
AMT_GOODS_PRICE                   278
NAME_TYPE_SUITE                  1292
NAME_INCOME_TYPE                    0
NAME_EDUCATION_TYPE                 0
NAME_FAMILY_STATUS                  0
NAME_HOUSING_TYPE                   0
REGION_POPULATION_RELATIVE          0
DAYS_BIRTH                          0
DAYS_EMPLOYED                       0
DAYS_REGISTRATION                   0
DAYS_ID_PUBLISH                     0
OWN_CAR_AGE                    202929
FLAG_MOBIL                          0
FLAG_EMP_PHONE                      0
FLAG_WORK_PHONE                     0
FLAG_CONT_MOBILE                    0
FLAG_PHONE                          0
FLAG_EMAIL                          0
OCCUPATION_TYPE                 96391
CNT_FAM_MEMBERS                     2
REGION_RATING_CLIENT                0
REGION_RATING_CLIENT_W_CITY         0
WEEKDAY_APPR_PROCESS_START          0
HOUR_APPR_PROCESS_START             0
REG_REGION_NOT_LIVE_REGION          0
REG_REGION_NOT_WORK_REGION          0
LIVE_REGION_NOT_WORK_REGION         0
REG_CITY_NOT_LIVE_CITY              0
REG_CITY_NOT_WORK_CITY              0
LIVE_CITY_NOT_WORK_CITY             0
ORGANIZATION_TYPE                   0
EXT_SOURCE_1                   173378
EXT_SOURCE_2                      660
EXT_SOURCE_3                    60965
APARTMENTS_AVG                 156061
BASEMENTAREA_AVG               179943
YEARS_BEGINEXPLUATATION_AVG    150007
YEARS_BUILD_AVG                204488
COMMONAREA_AVG                 214865
ELEVATORS_AVG                  163891
ENTRANCES_AVG                  154828
FLOORSMAX_AVG                  153020
FLOORSMIN_AVG                  208642
LANDAREA_AVG                   182590
LIVINGAPARTMENTS_AVG           210199
LIVINGAREA_AVG                 154350
NONLIVINGAPARTMENTS_AVG        213514
NONLIVINGAREA_AVG              169682
APARTMENTS_MODE                156061
BASEMENTAREA_MODE              179943
YEARS_BEGINEXPLUATATION_MODE   150007
YEARS_BUILD_MODE               204488
COMMONAREA_MODE                214865
```

```
ELEVATORS_MODE                  163891
ENTRANCES_MODE                  154828
FLOORSMAX_MODE                  153020
FLOORSMIN_MODE                  208642
LANDAREA_MODE                   182590
LIVINGAPARTMENTS_MODE           210199
LIVINGAREA_MODE                 154350
NONLIVINGAPARTMENTS_MODE        213514
NONLIVINGAREA_MODE              169682
APARTMENTS_MEDI                 156061
BASEMENTAREA_MEDI               179943
YEARS_BEGINEXPLUATATION_MEDI    150007
YEARS_BUILD_MEDI                204488
COMMONAREA_MEDI                 214865
ELEVATORS_MEDI                  163891
ENTRANCES_MEDI                  154828
FLOORSMAX_MEDI                  153020
FLOORSMIN_MEDI                  208642
LANDAREA_MEDI                   182590
LIVINGAPARTMENTS_MEDI           210199
LIVINGAREA_MEDI                 154350
NONLIVINGAPARTMENTS_MEDI        213514
NONLIVINGAREA_MEDI              169682
FONDKAPREMONT_MODE              210295
HOUSETYPE_MODE                  154297
TOTALAREA_MODE                  148431
WALLSMATERIAL_MODE              156341
EMERGENCYSTATE_MODE             145755
OBS_30_CNT_SOCIAL_CIRCLE          1021
DEF_30_CNT_SOCIAL_CIRCLE          1021
OBS_60_CNT_SOCIAL_CIRCLE          1021
DEF_60_CNT_SOCIAL_CIRCLE          1021
DAYS_LAST_PHONE_CHANGE               1
FLAG_DOCUMENT_2                      0
FLAG_DOCUMENT_3                      0
FLAG_DOCUMENT_4                      0
FLAG_DOCUMENT_5                      0
FLAG_DOCUMENT_6                      0
FLAG_DOCUMENT_7                      0
FLAG_DOCUMENT_8                      0
FLAG_DOCUMENT_9                      0
FLAG_DOCUMENT_10                     0
FLAG_DOCUMENT_11                     0
FLAG_DOCUMENT_12                     0
FLAG_DOCUMENT_13                     0
FLAG_DOCUMENT_14                     0
FLAG_DOCUMENT_15                     0
FLAG_DOCUMENT_16                     0
FLAG_DOCUMENT_17                     0
FLAG_DOCUMENT_18                     0
FLAG_DOCUMENT_19                     0
FLAG_DOCUMENT_20                     0
FLAG_DOCUMENT_21                     0
AMT_REQ_CREDIT_BUREAU_HOUR       41519
AMT_REQ_CREDIT_BUREAU_DAY        41519
AMT_REQ_CREDIT_BUREAU_WEEK       41519
AMT_REQ_CREDIT_BUREAU_MON        41519
AMT_REQ_CREDIT_BUREAU_QRT        41519
AMT_REQ_CREDIT_BUREAU_YEAR       41519
dtype: int64
```

In [15]:

```
# Percentage of null values
applications.isna().mean().round(5)*100
```

Out[15]:

```
SK_ID_CURR                       0.000
TARGET                           0.000
NAME_CONTRACT_TYPE               0.000
CODE_GENDER                      0.000
FLAG_OWN_CAR                     0.000
```

```
FLAG_OWN_CAR                    0.000
FLAG_OWN_REALTY                 0.000
CNT_CHILDREN                    0.000
AMT_INCOME_TOTAL                0.000
AMT_CREDIT                      0.000
AMT_ANNUITY                     0.004
AMT_GOODS_PRICE                 0.090
NAME_TYPE_SUITE                 0.420
NAME_INCOME_TYPE                0.000
NAME_EDUCATION_TYPE             0.000
NAME_FAMILY_STATUS              0.000
NAME_HOUSING_TYPE               0.000
REGION_POPULATION_RELATIVE      0.000
DAYS_BIRTH                      0.000
DAYS_EMPLOYED                   0.000
DAYS_REGISTRATION               0.000
DAYS_ID_PUBLISH                 0.000
OWN_CAR_AGE                    65.991
FLAG_MOBIL                      0.000
FLAG_EMP_PHONE                  0.000
FLAG_WORK_PHONE                 0.000
FLAG_CONT_MOBILE                0.000
FLAG_PHONE                      0.000
FLAG_EMAIL                      0.000
OCCUPATION_TYPE                31.346
CNT_FAM_MEMBERS                 0.001
REGION_RATING_CLIENT            0.000
REGION_RATING_CLIENT_W_CITY     0.000
WEEKDAY_APPR_PROCESS_START      0.000
HOUR_APPR_PROCESS_START         0.000
REG_REGION_NOT_LIVE_REGION      0.000
REG_REGION_NOT_WORK_REGION      0.000
LIVE_REGION_NOT_WORK_REGION     0.000
REG_CITY_NOT_LIVE_CITY          0.000
REG_CITY_NOT_WORK_CITY          0.000
LIVE_CITY_NOT_WORK_CITY         0.000
ORGANIZATION_TYPE               0.000
EXT_SOURCE_1                   56.381
EXT_SOURCE_2                    0.215
EXT_SOURCE_3                   19.825
APARTMENTS_AVG                 50.750
BASEMENTAREA_AVG               58.516
YEARS_BEGINEXPLUATATION_AVG    48.781
YEARS_BUILD_AVG                66.498
COMMONAREA_AVG                 69.872
ELEVATORS_AVG                  53.296
ENTRANCES_AVG                  50.349
FLOORSMAX_AVG                  49.761
FLOORSMIN_AVG                  67.849
LANDAREA_AVG                   59.377
LIVINGAPARTMENTS_AVG           68.355
LIVINGAREA_AVG                 50.193
NONLIVINGAPARTMENTS_AVG        69.433
NONLIVINGAREA_AVG              55.179
APARTMENTS_MODE                50.750
BASEMENTAREA_MODE              58.516
YEARS_BEGINEXPLUATATION_MODE   48.781
YEARS_BUILD_MODE               66.498
COMMONAREA_MODE                69.872
ELEVATORS_MODE                 53.296
ENTRANCES_MODE                 50.349
FLOORSMAX_MODE                 49.761
FLOORSMIN_MODE                 67.849
LANDAREA_MODE                  59.377
LIVINGAPARTMENTS_MODE          68.355
LIVINGAREA_MODE                50.193
NONLIVINGAPARTMENTS_MODE       69.433
NONLIVINGAREA_MODE             55.179
APARTMENTS_MEDI                50.750
BASEMENTAREA_MEDI              58.516
YEARS_BEGINEXPLUATATION_MEDI   48.781
YEARS_BUILD_MEDI               66.498
COMMONAREA_MEDI                69.872
```

```
COMMONAREA_MEDI                 69.872
ELEVATORS_MEDI                  53.296
ENTRANCES_MEDI                  50.349
FLOORSMAX_MEDI                  49.761
FLOORSMIN_MEDI                  67.849
LANDAREA_MEDI                   59.377
LIVINGAPARTMENTS_MEDI           68.355
LIVINGAREA_MEDI                 50.193
NONLIVINGAPARTMENTS_MEDI        69.433
NONLIVINGAREA_MEDI              55.179
FONDKAPREMONT_MODE              68.386
HOUSETYPE_MODE                  50.176
TOTALAREA_MODE                  48.269
WALLSMATERIAL_MODE              50.841
EMERGENCYSTATE_MODE             47.398
OBS_30_CNT_SOCIAL_CIRCLE         0.332
DEF_30_CNT_SOCIAL_CIRCLE         0.332
OBS_60_CNT_SOCIAL_CIRCLE         0.332
DEF_60_CNT_SOCIAL_CIRCLE         0.332
DAYS_LAST_PHONE_CHANGE           0.000
FLAG_DOCUMENT_2                  0.000
FLAG_DOCUMENT_3                  0.000
FLAG_DOCUMENT_4                  0.000
FLAG_DOCUMENT_5                  0.000
FLAG_DOCUMENT_6                  0.000
FLAG_DOCUMENT_7                  0.000
FLAG_DOCUMENT_8                  0.000
FLAG_DOCUMENT_9                  0.000
FLAG_DOCUMENT_10                 0.000
FLAG_DOCUMENT_11                 0.000
FLAG_DOCUMENT_12                 0.000
FLAG_DOCUMENT_13                 0.000
FLAG_DOCUMENT_14                 0.000
FLAG_DOCUMENT_15                 0.000
FLAG_DOCUMENT_16                 0.000
FLAG_DOCUMENT_17                 0.000
FLAG_DOCUMENT_18                 0.000
FLAG_DOCUMENT_19                 0.000
FLAG_DOCUMENT_20                 0.000
FLAG_DOCUMENT_21                 0.000
AMT_REQ_CREDIT_BUREAU_HOUR      13.502
AMT_REQ_CREDIT_BUREAU_DAY       13.502
AMT_REQ_CREDIT_BUREAU_WEEK      13.502
AMT_REQ_CREDIT_BUREAU_MON       13.502
AMT_REQ_CREDIT_BUREAU_QRT       13.502
AMT_REQ_CREDIT_BUREAU_YEAR      13.502
dtype: float64
```

Some columns consists more than 40% of null values can be drop. Certain columns like apartments area in a density region can give some insights to living condition of the individuals. Although we can keep some apartments columns, rest can be drop.

In [16]:

```
dropped_columns = applications.loc[:, 'BASEMENTAREA_AVG':'EMERGENCYSTATE_MODE'].columns
applications.drop(dropped_columns, inplace=True, axis=1)
applications.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 76 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
```

```
 8   AMT_CREDIT                   307511 non-null  float64
 9   AMT_ANNUITY                  307499 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  OWN_CAR_AGE                  104582 non-null  float64
 22  FLAG_MOBIL                   307511 non-null  int64
 23  FLAG_EMP_PHONE               307511 non-null  int64
 24  FLAG_WORK_PHONE              307511 non-null  int64
 25  FLAG_CONT_MOBILE             307511 non-null  int64
 26  FLAG_PHONE                   307511 non-null  int64
 27  FLAG_EMAIL                   307511 non-null  int64
 28  OCCUPATION_TYPE              211120 non-null  object
 29  CNT_FAM_MEMBERS              307509 non-null  float64
 30  REGION_RATING_CLIENT         307511 non-null  int64
 31  REGION_RATING_CLIENT_W_CITY  307511 non-null  int64
 32  WEEKDAY_APPR_PROCESS_START   307511 non-null  object
 33  HOUR_APPR_PROCESS_START      307511 non-null  int64
 34  REG_REGION_NOT_LIVE_REGION   307511 non-null  int64
 35  REG_REGION_NOT_WORK_REGION   307511 non-null  int64
 36  LIVE_REGION_NOT_WORK_REGION  307511 non-null  int64
 37  REG_CITY_NOT_LIVE_CITY       307511 non-null  int64
 38  REG_CITY_NOT_WORK_CITY       307511 non-null  int64
 39  LIVE_CITY_NOT_WORK_CITY      307511 non-null  int64
 40  ORGANIZATION_TYPE            307511 non-null  object
 41  EXT_SOURCE_1                 134133 non-null  float64
 42  EXT_SOURCE_2                 306851 non-null  float64
 43  EXT_SOURCE_3                 246546 non-null  float64
 44  APARTMENTS_AVG               151450 non-null  float64
 45  OBS_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 46  DEF_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 47  OBS_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 48  DEF_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 49  DAYS_LAST_PHONE_CHANGE       307510 non-null  float64
 50  FLAG_DOCUMENT_2              307511 non-null  int64
 51  FLAG_DOCUMENT_3              307511 non-null  int64
 52  FLAG_DOCUMENT_4              307511 non-null  int64
 53  FLAG_DOCUMENT_5              307511 non-null  int64
 54  FLAG_DOCUMENT_6              307511 non-null  int64
 55  FLAG_DOCUMENT_7              307511 non-null  int64
 56  FLAG_DOCUMENT_8              307511 non-null  int64
 57  FLAG_DOCUMENT_9              307511 non-null  int64
 58  FLAG_DOCUMENT_10             307511 non-null  int64
 59  FLAG_DOCUMENT_11             307511 non-null  int64
 60  FLAG_DOCUMENT_12             307511 non-null  int64
 61  FLAG_DOCUMENT_13             307511 non-null  int64
 62  FLAG_DOCUMENT_14             307511 non-null  int64
 63  FLAG_DOCUMENT_15             307511 non-null  int64
 64  FLAG_DOCUMENT_16             307511 non-null  int64
 65  FLAG_DOCUMENT_17             307511 non-null  int64
 66  FLAG_DOCUMENT_18             307511 non-null  int64
 67  FLAG_DOCUMENT_19             307511 non-null  int64
 68  FLAG_DOCUMENT_20             307511 non-null  int64
 69  FLAG_DOCUMENT_21             307511 non-null  int64
 70  AMT_REQ_CREDIT_BUREAU_HOUR   265992 non-null  float64
 71  AMT_REQ_CREDIT_BUREAU_DAY    265992 non-null  float64
 72  AMT_REQ_CREDIT_BUREAU_WEEK   265992 non-null  float64
 73  AMT_REQ_CREDIT_BUREAU_MON    265992 non-null  float64
 74  AMT_REQ_CREDIT_BUREAU_QRT    265992 non-null  float64
 75  AMT_REQ_CREDIT_BUREAU_YEAR   265992 non-null  float64
dtypes: float64(23), int64(41), object(12)
memory usage: 178.3+ MB
```

The Column Document flags doesn't have much relavent data as perpectives of our analysis. We can drop the

column because this information is not enough to analyse that what these documents are.

Handle missing values

In [17]:

```
# Find percentage of null values for each columns
applications.isna().mean().round(5)*100
```

Out[17]:

```
SK_ID_CURR                       0.000
TARGET                           0.000
NAME_CONTRACT_TYPE               0.000
CODE_GENDER                      0.000
FLAG_OWN_CAR                     0.000
FLAG_OWN_REALTY                  0.000
CNT_CHILDREN                     0.000
AMT_INCOME_TOTAL                 0.000
AMT_CREDIT                       0.000
AMT_ANNUITY                      0.004
AMT_GOODS_PRICE                  0.090
NAME_TYPE_SUITE                  0.420
NAME_INCOME_TYPE                 0.000
NAME_EDUCATION_TYPE              0.000
NAME_FAMILY_STATUS               0.000
NAME_HOUSING_TYPE                0.000
REGION_POPULATION_RELATIVE       0.000
DAYS_BIRTH                       0.000
DAYS_EMPLOYED                    0.000
DAYS_REGISTRATION                0.000
DAYS_ID_PUBLISH                  0.000
OWN_CAR_AGE                     65.991
FLAG_MOBIL                       0.000
FLAG_EMP_PHONE                   0.000
FLAG_WORK_PHONE                  0.000
FLAG_CONT_MOBILE                 0.000
FLAG_PHONE                       0.000
FLAG_EMAIL                       0.000
OCCUPATION_TYPE                 31.346
CNT_FAM_MEMBERS                  0.001
REGION_RATING_CLIENT             0.000
REGION_RATING_CLIENT_W_CITY      0.000
WEEKDAY_APPR_PROCESS_START       0.000
HOUR_APPR_PROCESS_START          0.000
REG_REGION_NOT_LIVE_REGION       0.000
REG_REGION_NOT_WORK_REGION       0.000
LIVE_REGION_NOT_WORK_REGION      0.000
REG_CITY_NOT_LIVE_CITY           0.000
REG_CITY_NOT_WORK_CITY           0.000
LIVE_CITY_NOT_WORK_CITY          0.000
ORGANIZATION_TYPE                0.000
EXT_SOURCE_1                    56.381
EXT_SOURCE_2                     0.215
EXT_SOURCE_3                    19.825
APARTMENTS_AVG                  50.750
OBS_30_CNT_SOCIAL_CIRCLE         0.332
DEF_30_CNT_SOCIAL_CIRCLE         0.332
OBS_60_CNT_SOCIAL_CIRCLE         0.332
DEF_60_CNT_SOCIAL_CIRCLE         0.332
DAYS_LAST_PHONE_CHANGE           0.000
FLAG_DOCUMENT_2                  0.000
FLAG_DOCUMENT_3                  0.000
FLAG_DOCUMENT_4                  0.000
FLAG_DOCUMENT_5                  0.000
FLAG_DOCUMENT_6                  0.000
FLAG_DOCUMENT_7                  0.000
FLAG_DOCUMENT_8                  0.000
FLAG_DOCUMENT_9                  0.000
FLAG_DOCUMENT_10                 0.000
```

```
FLAG_DOCUMENT_11                 0.000
FLAG_DOCUMENT_12                 0.000
FLAG_DOCUMENT_13                 0.000
FLAG_DOCUMENT_14                 0.000
FLAG_DOCUMENT_15                 0.000
FLAG_DOCUMENT_16                 0.000
FLAG_DOCUMENT_17                 0.000
FLAG_DOCUMENT_18                 0.000
FLAG_DOCUMENT_19                 0.000
FLAG_DOCUMENT_20                 0.000
FLAG_DOCUMENT_21                 0.000
AMT_REQ_CREDIT_BUREAU_HOUR      13.502
AMT_REQ_CREDIT_BUREAU_DAY       13.502
AMT_REQ_CREDIT_BUREAU_WEEK      13.502
AMT_REQ_CREDIT_BUREAU_MON       13.502
AMT_REQ_CREDIT_BUREAU_QRT       13.502
AMT_REQ_CREDIT_BUREAU_YEAR      13.502
dtype: float64
```

Column - OCCUPATION_TYPE has 31% null values and can be impute with a occupation category'Others'

In [19]:

```
applications.OCCUPATION_TYPE.fillna('Others', inplace=True)
applications.OCCUPATION_TYPE.value_counts(normalize=True)*100
```

Out[19]:

```
Others                   31.345545
Laborers                 17.946025
Sales staff              10.439301
Core staff                8.965533
Managers                  6.949670
Drivers                   6.049540
High skill tech staff     3.700681
Accountants               3.191105
Medicine staff            2.776161
Security staff            2.185613
Cooking staff             1.933589
Cleaning staff            1.513117
Private service staff     0.862408
Low-skill Laborers        0.680626
Waiters/barmen staff      0.438358
Secretaries               0.424375
Realty agents             0.244219
HR staff                  0.183083
IT staff                  0.171051
Name: OCCUPATION_TYPE, dtype: float64
```

The column 'EXT_SOURCE' have nulls(EXT_SOURCE_1-56.381%,EXT_SOURCE_2-0.215%,EXT_SOURCE_3-19.825%). These columns denoting scores given by external agencies and all appications doesn't have all the values filled in it. It will be meaningfull if we take mean of these three for analysis and can add a new column with average of the scores.

In [20]:

```
applications['EXT_SOURCE_AVG']= applications.loc[:,['EXT_SOURCE_1','EXT_SOURCE_2','EXT_SO
URCE_3']].mean(axis=1)
```

In [21]:

```
applications['EXT_SOURCE_AVG']
```

Out[21]:

```
0          0.161787
1          0.466757
2          0.642739
3          0.650442
4          0.322738
```

```
...
307506    0.413601
307507    0.115992
307508    0.499536
307509    0.587593
307510    0.518984
Name: EXT_SOURCE_AVG, Length: 307511, dtype: float64
```

Handle Outliers

In [22]:

```
applications.describe()
```

Out[22]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRI
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e-
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e-
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e-
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e-
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e-
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e-
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e-
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e-

Analysing column 'AMT_INCOME_TOTAL' for Outliers

In [26]:

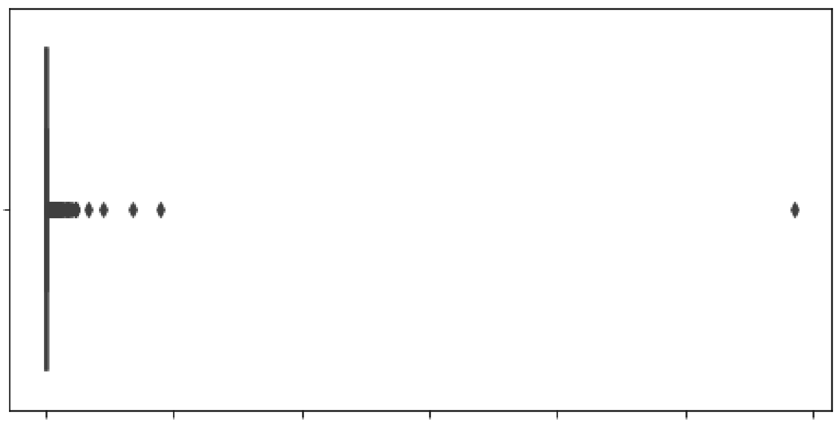
```
applications['AMT_INCOME_TOTAL'].describe()
```

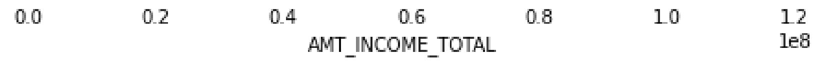
Out[26]:

```
count    3.075110e+05
mean     1.687979e+05
std      2.371231e+05
min      2.565000e+04
25%      1.125000e+05
50%      1.471500e+05
75%      2.025000e+05
max      1.170000e+08
Name: AMT_INCOME_TOTAL, dtype: float64
```

In [27]:

```
plt.figure(figsize=[8,4])
sns.boxplot(applications['AMT_INCOME_TOTAL'])
plt.show()
```





In [30]:

```
# Analysing Quantiles like(50%,75%,90% and so on)
applications.AMT_INCOME_TOTAL.quantile([0.5,0.7,0.90,00.95,00.99,0.999,0.9999])
```

Out[30]:

```
0.5000     147150.0
0.7000     180000.0
0.9000     270000.0
0.9500     337500.0
0.9900     472500.0
0.9990     900000.0
0.9999    2250000.0
Name: AMT_INCOME_TOTAL, dtype: float64
```

In [32]:

```
# values lying outside 0.9999 quantile which is outlier.
applications[applications.AMT_INCOME_TOTAL>0.2*10**8]
```

Out[32]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDI
12840	114967	1	Cash loans	F	N	Y	

Outliers in column 'AMT_INCOME_TOTAL' are continuous except one and we can retain them.

One variable s.no(12840) have one single value which is too high and the loan applied is a normal amount, so it can be drop.

In [35]:

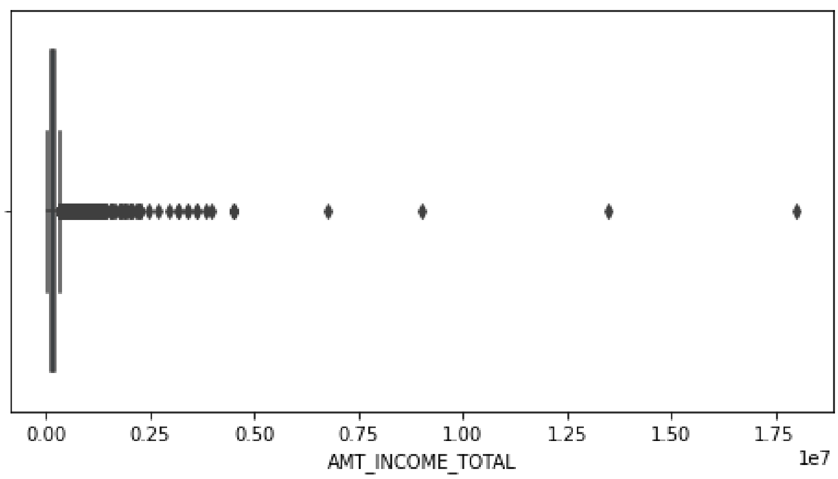
```
applications = applications[~(applications.AMT_INCOME_TOTAL > 0.2*10**8)]
applications.shape
```

Out[35]:

```
(307510, 77)
```

In [38]:

```
# Plot 'AMT_INCOME_TOTAL' again to check
plt.figure(figsize=[8,4])
sns.boxplot(applications['AMT_INCOME_TOTAL'])
plt.show()
```



There are more outliers in the columns 'AMT_INCOME_TOTAL', but these values are meaningful as it is spread

There are more outliers in the columns 'AMT_INCOME_TOTAL', but these values are meaningful as it is spread more or less evenly. So, we can remain as it is.

Analyse column 'AMT_CREDIT' for outliers

In [39]:

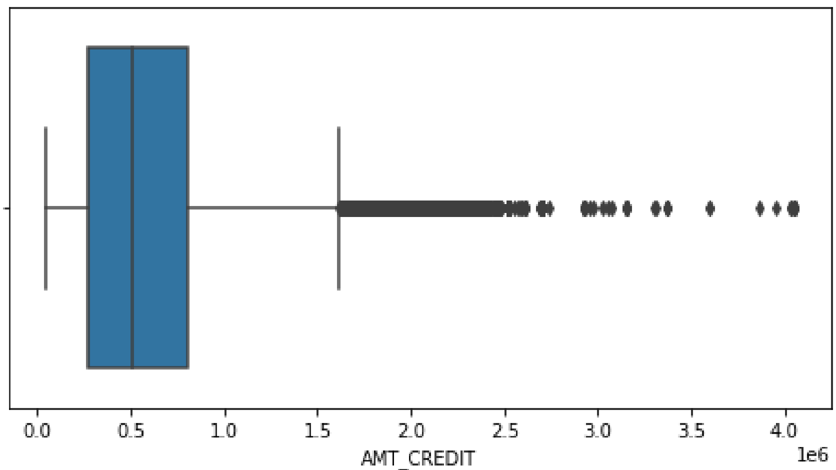
```
applications['AMT_CREDIT'].describe()
```

Out[39]:

```
count    3.075100e+05
mean     5.990261e+05
std      4.024914e+05
min      4.500000e+04
25%      2.700000e+05
50%      5.135310e+05
75%      8.086500e+05
max      4.050000e+06
Name: AMT_CREDIT, dtype: float64
```

In [45]:

```
plt.figure(figsize=[8,4])
sns.boxplot(applications['AMT_CREDIT'])
plt.show()
```



In [47]:

```
applications['AMT_CREDIT'].quantile([0.5,0.7,0.9,0.95,0.99])
```

Out[47]:

```
0.50     513531.0
0.70     755190.0
0.90    1133748.0
0.95    1350000.0
0.99    1854000.0
Name: AMT_CREDIT, dtype: float64
```

In [48]:

```
# In rows, values lying outside 0.99%.(more than 1854000)
applications[applications['AMT_CREDIT']>1854000]
```

Out[48]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILI
189	100219	0	Cash loans	M	N	Y	
337	100389	0	Cash loans	M	Y	Y	
341	100393	0	Cash loans	M	Y	Y	

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILD
441	100506	0	Cash loans	F	Y	Y	
485	100559	0	Cash loans	F	Y	Y	
...	...	...	...	...	...	...	
307055	455739	0	Cash loans	F	N	Y	
307095	455785	0	Cash loans	F	Y	Y	
307165	455868	0	Cash loans	F	Y	Y	
307214	455922	0	Cash loans	M	Y	N	
307422	456155	0	Cash loans	F	N	Y	

3075 rows × 77 columns

In the column 'AMT_GOODS_PRICE' have relatively higher credits and by observing many of them good rpice also close to credit. We can kept like that as it is.

Analysing the column 'AMT_GOODS_PRICE' column for outliers

In [49]:

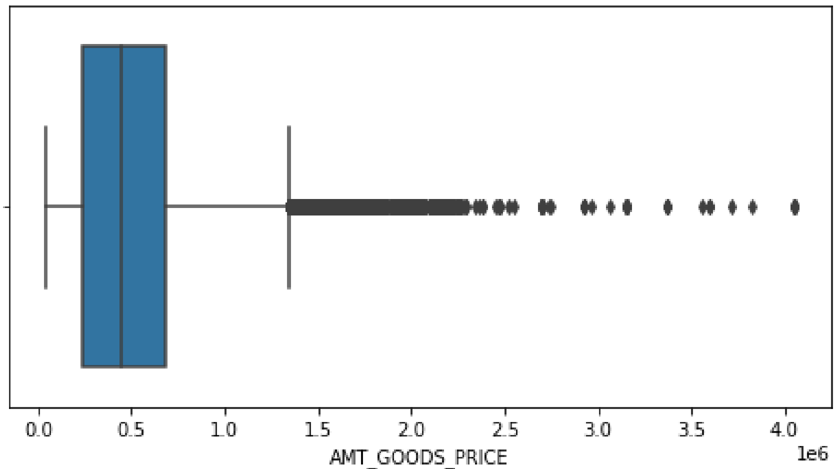
```
applications['AMT_GOODS_PRICE'].describe()
```

Out[49]:

```
count    3.072320e+05
mean     5.383965e+05
std      3.694470e+05
min      4.050000e+04
25%      2.385000e+05
50%      4.500000e+05
75%      6.795000e+05
max      4.050000e+06
Name: AMT_GOODS_PRICE, dtype: float64
```

In [50]:

```
plt.figure(figsize=[8,4])
sns.boxplot(applications['AMT_GOODS_PRICE'])
plt.show()
```



It seems very close to the credit distribution which is rela life scenario and hence can be left as it is.

Analysing column 'DAYS_BIRTH' for outliers

In [54]:

```
applications['DAYS_BIRTH'].describe()
```

Out[54]:

```
count    307510.000000
mean     -16037.006195
std        4363.991364
min      -25229.000000
25%      -19682.000000
50%      -15750.000000
75%      -12413.000000
max       -7489.000000
Name: DAYS_BIRTH, dtype: float64
```

In this column values are filled in the form of a number format and the data isn't in the readable format. By converting this values, we can store this data in the new column.

In [55]:

```
applications['AGE'] = np.ceil(applications['DAYS_BIRTH']/-365) # New column - 'AGE'
```

In [56]:

```
applications['AGE'].describe()  # Min age - 21 and Max age - 70
```

Out[56]:

```
count    307510.000000
mean         44.433121
std          11.954500
min          21.000000
25%          35.000000
50%          44.000000
75%          54.000000
max          70.000000
Name: AGE, dtype: float64
```

Standardize Values

Columns like 'DAYS_EMPLOYED' is in days, we need to convert it in years as already did for 'AGE'.

In [58]:

```
applications['EXPERIENCE'] = np.round(applications['DAYS_EMPLOYED']/-365,1)
applications['EXPERIENCE']
```

Out[58]:

```
0            1.7
1            3.3
2            0.6
3            8.3
4            8.3
           ...
307506       0.6
307507   -1000.7
307508      21.7
307509      13.1
307510       3.5
Name: EXPERIENCE, Length: 307510, dtype: float64
```

In [60]:

```
# (-)ve Values in EXPERIENCE
applications[applications['EXPERIENCE']<0]['EXPERIENCE'].value_counts()
```

Out[60]:

```
-1000.7    55374
Name: EXPERIENCE, dtype: int64
```

from above code, we can see that all 55374 values are same can be standardized by considering it as NaN.

Analysing column 'NAME_FAMILY_STATUS'.

To verify that column consist any unknown/null values or same values.

In [61]:

```
applications['NAME_FAMILY_STATUS'].value_counts()
```

Out[61]:

```
Married                 196431
Single / not married     45444
Civil marriage           29775
Separated                19770
Widow                    16088
Unknown                      2
Name: NAME_FAMILY_STATUS, dtype: int64
```

From above data we can say that 'marriage' and 'civil marriage' are same columns, can be merged into a single column. Single/Not married' can be convert to single. Data consists the 2 unknown values in it, can be drop.

In [64]:

```
# Coverting value names.
applications.NAME_FAMILY_STATUS.replace({'Civil marriage':'Married','Single / not married
':'Single'}, inplace=True)
```

In [65]:

```
# Dropping the unknown values
applications = applications[~(applications['NAME_FAMILY_STATUS']=='Unknown')]
applications[(applications['NAME_FAMILY_STATUS']=='Unknown')]
```

Out[65]:

SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN

Analyze column 'CODE_GENDER'

In [66]:

```
applications['CODE_GENDER'].value_counts(normalize=True)*100
```

Out[66]:

```
F      65.834385
M      34.164314
XNA     0.001301
Name: CODE_GENDER, dtype: float64
```

Analyze columns 'NAME_CONTRACT_TYPE' for loans.

In [67]:

```
applications['NAME_CONTRACT_TYPE'].value_counts()
```

Out[67]:

```
Cash loans         278231
Revolving loans     29277
Name: NAME_CONTRACT_TYPE, dtype: int64
```

Adding column 'Credit_Bureau_Total'to put the combining of total credits to each individuals.

In [71]:

```
applications['Credit_Bureau_Total'] = applications.iloc[:,-9:-3].sum(axis=1)
applications.head()
```

Out[71]:

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

In [ ]:

Read 'Previous Applications' CSV file

From this data, we can find the previous applications to know about the applicant got approved successfully or got rejected and reason behind that approval or rejection. If successful, is the loan over or not?, is there any due or not? This data could be more useful to us by applying EDA and merging to current data(i.e applications.csv)

In [74]:

```
previous_applications = pd.read_csv('previous_application.csv')
previous_applications.head()
```

Out[74]:

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYM
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	

In [75]:

```
previous_applications.shape
```

Out[75]:

(1670214, 37)

In [77]:

```
previous_applications.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
 #   Column                       Non-Null Count    Dtype  
---  ------                       --------------    -----  
 0   SK_ID_PREV                   1670214 non-null  int64  
 1   SK_ID_CURR                   1670214 non-null  int64  
 2   NAME_CONTRACT_TYPE           1670214 non-null  object 
 3   AMT_ANNUITY                  1297979 non-null  float64
 4   AMT_APPLICATION              1670214 non-null  float64
 5   AMT_CREDIT                   1670213 non-null  float64
 6   AMT_DOWN_PAYMENT             774370 non-null   float64
 7   AMT_GOODS_PRICE              1284699 non-null  float64
 8   WEEKDAY_APPR_PROCESS_START   1670214 non-null  object 
 9   HOUR_APPR_PROCESS_START      1670214 non-null  int64  
 10  FLAG_LAST_APPL_PER_CONTRACT  1670214 non-null  object 
 11  NFLAG_LAST_APPL_IN_DAY       1670214 non-null  int64  
 12  RATE_DOWN_PAYMENT            774370 non-null   float64
 13  RATE_INTEREST_PRIMARY        5951 non-null     float64
 14  RATE_INTEREST_PRIVILEGED     5951 non-null     float64
 15  NAME_CASH_LOAN_PURPOSE       1670214 non-null  object 
 16  NAME_CONTRACT_STATUS         1670214 non-null  object 
 17  DAYS_DECISION                1670214 non-null  int64  
 18  NAME_PAYMENT_TYPE            1670214 non-null  object 
 19  CODE_REJECT_REASON           1670214 non-null  object 
 20  NAME_TYPE_SUITE              849809 non-null   object 
 21  NAME_CLIENT_TYPE             1670214 non-null  object 
 22  NAME_GOODS_CATEGORY          1670214 non-null  object 
 23  NAME_PORTFOLIO               1670214 non-null  object 
 24  NAME_PRODUCT_TYPE            1670214 non-null  object 
 25  CHANNEL_TYPE                 1670214 non-null  object 
 26  SELLERPLACE_AREA             1670214 non-null  int64  
 27  NAME_SELLER_INDUSTRY         1670214 non-null  object 
 28  CNT_PAYMENT                  1297984 non-null  float64
 29  NAME_YIELD_GROUP             1670214 non-null  object 
 30  PRODUCT_COMBINATION          1669868 non-null  object 
 31  DAYS_FIRST_DRAWING           997149 non-null   float64
 32  DAYS_FIRST_DUE               997149 non-null   float64
 33  DAYS_LAST_DUE_1ST_VERSION    997149 non-null   float64
 34  DAYS_LAST_DUE                997149 non-null   float64
 35  DAYS_TERMINATION             997149 non-null   float64
 36  NFLAG_INSURED_ON_APPROVAL    997149 non-null   float64
dtypes: float64(15), int64(6), object(16)
memory usage: 471.5+ MB
```

In [79]:

```
previous_applications.isna().mean()*100
```

Out[79]:

```
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
```

In [80]:

```
previous_applications['NAME_CONTRACT_STATUS'].value_counts(normalize=True)*100
```

Out[80]:

```
Approved        62.074740
Canceled        18.938831
Refused         17.403638
Unused offer     1.582791
Name: NAME_CONTRACT_STATUS, dtype: float64
```

In [81]:

```
previous_applications['FLAG_LAST_APPL_PER_CONTRACT'].value_counts(normalize=True)
```

Out[81]:

```
Y    0.994926
N    0.005074
Name: FLAG_LAST_APPL_PER_CONTRACT, dtype: float64
```

Keeping only the last application of all previous applications and dropping rest of the entries in the column 'FLAG_LAST_APPL_PER_CONTRACT'

In [82]:

```
previous_applications[previous_applications['FLAG_LAST_APPL_PER_CONTRACT'] == 'Y']
previous_applications['FLAG_LAST_APPL_PER_CONTRACT'].value_counts(normalize=True)
```

Out[82]:

```
Y    0.994926
N    0.005074
Name: FLAG_LAST_APPL_PER_CONTRACT, dtype: float64
```

In [83]:

```
previous_applications['NFLAG_LAST_APPL_IN_DAY'].value_counts(normalize=True)
```

Out[83]:

```
1    0.996468
0    0.003532
Name: NFLAG_LAST_APPL_IN_DAY, dtype: float64
```

Sorting previous application based on application id and dropping duplicates

In [86]:

```
previous_applications = previous_applications.sort_values('SK_ID_PREV', ascending=False)
.drop_duplicates('SK_ID_CURR')
```

In [87]:

```
previous_applications[previous_applications['DAYS_TERMINATION']>0].head()
```

Out[87]:

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWI
888701	2843497	451578	Cash loans	9175.185	132482.97	149969.97	
1345642	2843496	425374	Revolving loans	31500.000	630000.00	630000.00	
298226	2843493	337804	Revolving loans	2250.000	45000.00	45000.00	
1489940	2843491	107385	Cash loans	25421.985	841500.00	963684.00	
728908	2843487	424008	Consumer loans	7179.795	78402.87	78399.00	

In [88]:

```
previous_applications.shape
```

Out[88]:

```
(338857, 37)
```

The column of interests are

'SK_ID_CURR', 'AMT_CREDIT', 'NAME_CONTRACT_STATUS', 'CODE_REJECT_REASON', 'NAME_YIELD_GROUP' and 'DAYS_TERMINATION'.

Keeping the columns of interests and dropping rest columns to get better insight.

In [92]:

```
previous_applications_1 = previous_applications[['SK_ID_CURR','AMT_CREDIT','NAME_CONTRACT_STATUS','CODE_REJECT_REASON','NAME_YIELD_GROUP','DAYS_TERMINATION']]
```

In [93]:

```
previous_applications_1.head()
```

Out[93]:

	SK_ID_CURR	AMT_CREDIT	NAME_CONTRACT_STATUS	CODE_REJECT_REASON	NAME_YIELD_GROUP	DAYS_TERM
205485	406596	30912.75	Unused offer	CLIENT	XNA	
717142	140761	41499.00	Unused offer	CLIENT	XNA	
886179	237546	60673.50	Refused	LIMIT	middle	
359118	100125	59503.50	Refused	SCO	middle	
70058	250234	108180.00	Refused	SCO	low_action	

Rename the column names into meaning contect of current application

In [96]:

```
renames = {'AMT_CREDIT':'PREV_AMT_CREDIT','NAME_CONTRACT_STATUS':'PREV_CONTRACT_STATUS','DAYS_TERMINATION':'PREV_DAYS_TERMINATION','CODE_REJECT_REASON':'PREV_REJECT_REASON','NAME_YIELD_GROUP':'PREV_YIELD_GROUP'}
```

```
previous_applications_1 = previous_applications_1.rename(columns=renames)
previous_applications_1.head()
```

Out[96]:

	SK_ID_CURR	PREV_AMT_CREDIT	PREV_CONTRACT_STATUS	PREV_REJECT_REASON	PREV_YIELD_GROUP	PREV_I
205485	406596	30912.75	Unused offer	CLIENT	XNA	
717142	140761	41499.00	Unused offer	CLIENT	XNA	
886179	237546	60673.50	Refused	LIMIT	middle	
359118	100125	59503.50	Refused	SCO	middle	
70058	250234	108180.00	Refused	SCO	low_action	

Fixing anomalies in coulmn 'PREV_DAYS_TERMINATION'.

In [97]:

```
previous_applications_1['PREV_DAYS_TERMINATION'].value_counts(normalize=True)
```

Out[97]:

```
 365243.0    0.232769
-9.0         0.000909
-15.0        0.000909
-144.0       0.000905
-17.0        0.000901
               ...   
-2774.0      0.000004
-2709.0      0.000004
-2777.0      0.000004
-2783.0      0.000004
-2733.0      0.000004
Name: PREV_DAYS_TERMINATION, Length: 2785, dtype: float64
```

In [99]:

```
previous_applications_1.PREV_DAYS_TERMINATION[previous_applications_1.PREV_DAYS_TERMINATI
ON >0].value_counts()   # value 365243.0 seems impossible value, we'll replace it by NaN.
```

Out[99]:

```
365243.0    56079
Name: PREV_DAYS_TERMINATION, dtype: int64
```

In [100]:

```
# Replacing value by NaN
previous_applications_1.PREV_DAYS_TERMINATION.replace({365243.0:np.NaN}, inplace=True)
previous_applications_1.PREV_DAYS_TERMINATION[previous_applications_1.PREV_DAYS_TERMINATI
ON >0].value_counts()
```

Out[100]:

```
Series([], Name: PREV_DAYS_TERMINATION, dtype: int64)
```

Merge both the datasets ('Previous applications' and 'Current applications')

Using joins -:

1. left join

2. right join

3. left_on

4. right_on

In [103]:

```
applications = pd.merge(left=applications, right=previous_applications_1, how='left', le
ft_on='SK_ID_CURR', right_on='SK_ID_CURR')
applications.head()
```

Out[103]:

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

In [104]:

```
applications['PREV_CONTRACT_STATUS'].isna().mean()
```

Out[104]:

```
0.05350104712722921
```

In [105]:

```
applications.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 307508 entries, 0 to 307507
Data columns (total 85 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   SK_ID_CURR                   307508 non-null  int64
 1   TARGET                       307508 non-null  int64
 2   NAME_CONTRACT_TYPE           307508 non-null  object
 3   CODE_GENDER                  307508 non-null  object
 4   FLAG_OWN_CAR                 307508 non-null  object
 5   FLAG_OWN_REALTY              307508 non-null  object
 6   CNT_CHILDREN                 307508 non-null  int64
 7   AMT_INCOME_TOTAL             307508 non-null  float64
 8   AMT_CREDIT                   307508 non-null  float64
 9   AMT_ANNUITY                  307496 non-null  float64
 10  AMT_GOODS_PRICE              307232 non-null  float64
 11  NAME_TYPE_SUITE              306218 non-null  object
 12  NAME_INCOME_TYPE             307508 non-null  object
 13  NAME_EDUCATION_TYPE          307508 non-null  object
 14  NAME_FAMILY_STATUS           307508 non-null  object
 15  NAME_HOUSING_TYPE            307508 non-null  object
 16  REGION_POPULATION_RELATIVE   307508 non-null  float64
 17  DAYS_BIRTH                   307508 non-null  int64
 18  DAYS_EMPLOYED                307508 non-null  int64
 19  DAYS_REGISTRATION            307508 non-null  float64
 20  DAYS_ID_PUBLISH              307508 non-null  int64
 21  OWN_CAR_AGE                  104582 non-null  float64
 22  FLAG_MOBIL                   307508 non-null  int64
 23  FLAG_EMP_PHONE               307508 non-null  int64
 24  FLAG_WORK_PHONE              307508 non-null  int64
 25  FLAG_CONT_MOBILE             307508 non-null  int64
 26  FLAG_PHONE                   307508 non-null  int64
```

```
 26  FLAG_PHONE                    307508 non-null  int64
 27  FLAG_EMAIL                    307508 non-null  int64
 28  OCCUPATION_TYPE               307508 non-null  object
 29  CNT_FAM_MEMBERS               307508 non-null  float64
 30  REGION_RATING_CLIENT          307508 non-null  int64
 31  REGION_RATING_CLIENT_W_CITY   307508 non-null  int64
 32  WEEKDAY_APPR_PROCESS_START    307508 non-null  object
 33  HOUR_APPR_PROCESS_START       307508 non-null  int64
 34  REG_REGION_NOT_LIVE_REGION    307508 non-null  int64
 35  REG_REGION_NOT_WORK_REGION    307508 non-null  int64
 36  LIVE_REGION_NOT_WORK_REGION   307508 non-null  int64
 37  REG_CITY_NOT_LIVE_CITY        307508 non-null  int64
 38  REG_CITY_NOT_WORK_CITY        307508 non-null  int64
 39  LIVE_CITY_NOT_WORK_CITY       307508 non-null  int64
 40  ORGANIZATION_TYPE             307508 non-null  object
 41  EXT_SOURCE_1                  134131 non-null  float64
 42  EXT_SOURCE_2                  306848 non-null  float64
 43  EXT_SOURCE_3                  246544 non-null  float64
 44  APARTMENTS_AVG                151448 non-null  float64
 45  OBS_30_CNT_SOCIAL_CIRCLE      306487 non-null  float64
 46  DEF_30_CNT_SOCIAL_CIRCLE      306487 non-null  float64
 47  OBS_60_CNT_SOCIAL_CIRCLE      306487 non-null  float64
 48  DEF_60_CNT_SOCIAL_CIRCLE      306487 non-null  float64
 49  DAYS_LAST_PHONE_CHANGE        307507 non-null  float64
 50  FLAG_DOCUMENT_2               307508 non-null  int64
 51  FLAG_DOCUMENT_3               307508 non-null  int64
 52  FLAG_DOCUMENT_4               307508 non-null  int64
 53  FLAG_DOCUMENT_5               307508 non-null  int64
 54  FLAG_DOCUMENT_6               307508 non-null  int64
 55  FLAG_DOCUMENT_7               307508 non-null  int64
 56  FLAG_DOCUMENT_8               307508 non-null  int64
 57  FLAG_DOCUMENT_9               307508 non-null  int64
 58  FLAG_DOCUMENT_10              307508 non-null  int64
 59  FLAG_DOCUMENT_11              307508 non-null  int64
 60  FLAG_DOCUMENT_12              307508 non-null  int64
 61  FLAG_DOCUMENT_13              307508 non-null  int64
 62  FLAG_DOCUMENT_14              307508 non-null  int64
 63  FLAG_DOCUMENT_15              307508 non-null  int64
 64  FLAG_DOCUMENT_16              307508 non-null  int64
 65  FLAG_DOCUMENT_17              307508 non-null  int64
 66  FLAG_DOCUMENT_18              307508 non-null  int64
 67  FLAG_DOCUMENT_19              307508 non-null  int64
 68  FLAG_DOCUMENT_20              307508 non-null  int64
 69  FLAG_DOCUMENT_21              307508 non-null  int64
 70  AMT_REQ_CREDIT_BUREAU_HOUR    265990 non-null  float64
 71  AMT_REQ_CREDIT_BUREAU_DAY     265990 non-null  float64
 72  AMT_REQ_CREDIT_BUREAU_WEEK    265990 non-null  float64
 73  AMT_REQ_CREDIT_BUREAU_MON     265990 non-null  float64
 74  AMT_REQ_CREDIT_BUREAU_QRT     265990 non-null  float64
 75  AMT_REQ_CREDIT_BUREAU_YEAR    265990 non-null  float64
 76  EXT_SOURCE_AVG                307336 non-null  float64
 77  AGE                           307508 non-null  float64
 78  EXPERIENCE                    307508 non-null  float64
 79  Credit_Bureau_Total           307508 non-null  float64
 80  PREV_AMT_CREDIT               291056 non-null  float64
 81  PREV_CONTRACT_STATUS          291056 non-null  object
 82  PREV_REJECT_REASON            291056 non-null  object
 83  PREV_YIELD_GROUP              291056 non-null  object
 84  PREV_DAYS_TERMINATION         159689 non-null  float64
dtypes: float64(29), int64(41), object(15)
memory usage: 201.8+ MB
```

In [ ]:

```

```

Univariate Analysis

Analyzing Target variable

In [107]:

```
applications['TARGET'].value_counts(normalize=True)
```

Out[107]:

```
0    0.919274
1    0.080726
Name: TARGET, dtype: float64
```

In [109]:

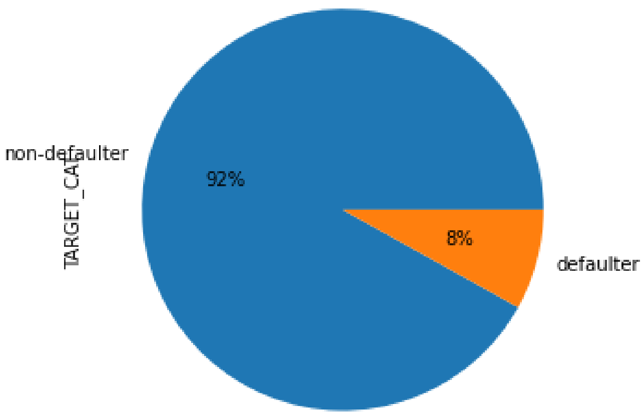
```
# Adding new column from column 'TARGET'
applications['TARGET_CAT']=applications['TARGET'].apply(lambda x: 'defaulter' if x==1 el
se 'non-defaulter')
applications['TARGET_CAT'].value_counts(normalize=True)*100
```

Out[109]:

```
non-defaulter    91.927364
defaulter         8.072636
Name: TARGET_CAT, dtype: float64
```

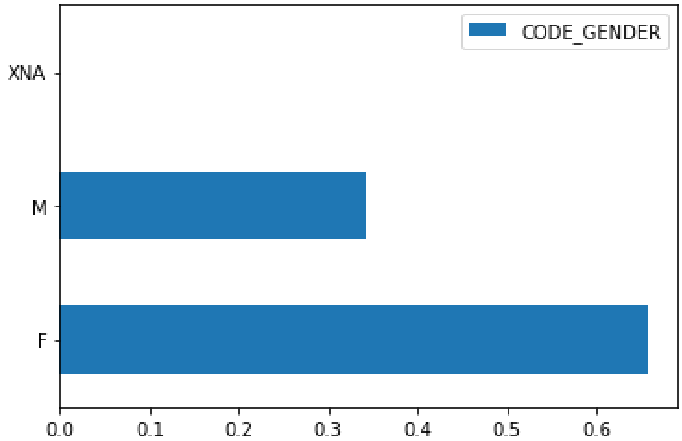
In [129]:

```
plt.figure(figsize=(5,5))
applications['TARGET_CAT'].value_counts(normalize=True).plot.pie(autopct='%1.0f%%')
plt.show()
```



In [132]:

```
# gerder distribution in data
applications['CODE_GENDER'].value_counts(normalize=True).plot.barh()
plt.legend()
plt.show()
```



Education Type

In [133]:

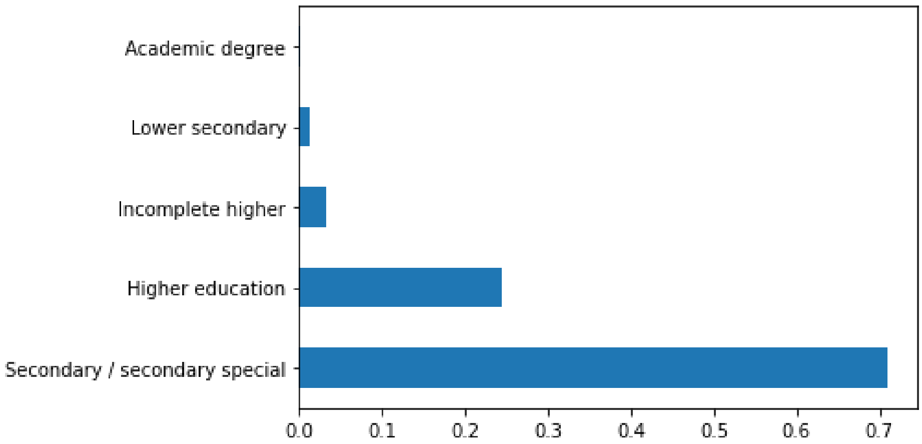
```
applications['NAME_EDUCATION_TYPE'].value_counts()
```

Out[133]:

```
Secondary / secondary special    218390
Higher education                  74862
Incomplete higher                 10277
Lower secondary                    3815
Academic degree                     164
Name: NAME_EDUCATION_TYPE, dtype: int64
```

In [137]:

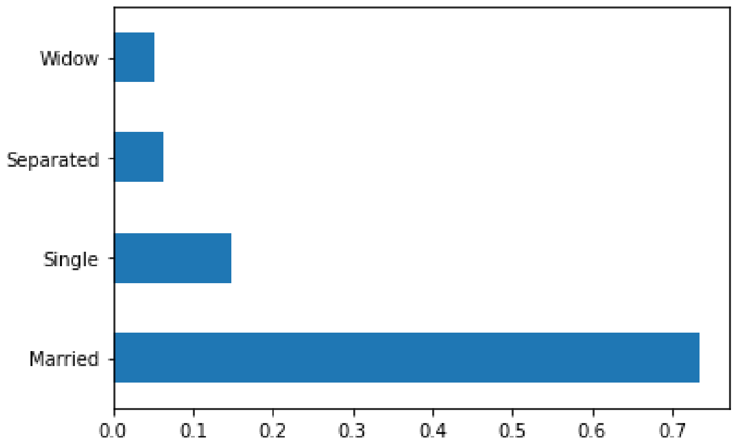
```
applications['NAME_EDUCATION_TYPE'].value_counts(normalize=True).plot.barh()
plt.show()
```



Family Status

In [141]:

```
applications['NAME_FAMILY_STATUS'].value_counts(normalize=True).plot.barh()
plt.show()
```



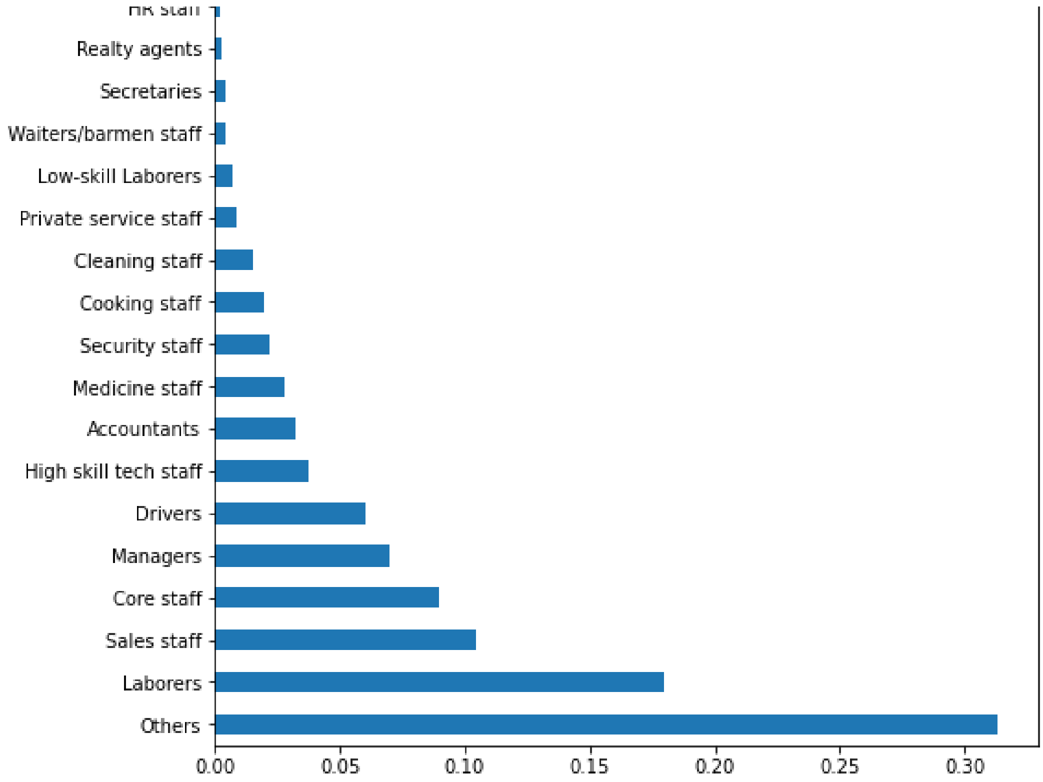
Occupation Type

In [148]:

```
plt.figure(figsize=[8,8])
applications['OCCUPATION_TYPE'].value_counts(normalize=True).plot.barh()
```

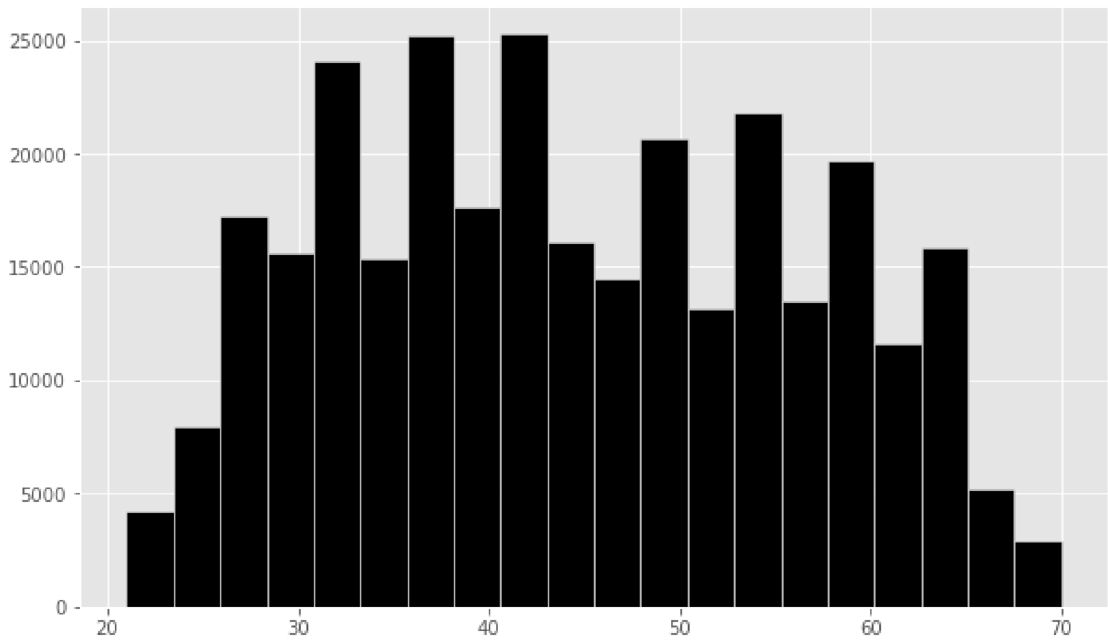
Out[148]:

```
<AxesSubplot:>
```



In [160]:

```
plt.style.use('ggplot')
plt.figure(figsize=[10,6])
plt.hist(applications['AGE'], bins=20, color='black', edgecolor='white')
plt.show()
```



In [162]:

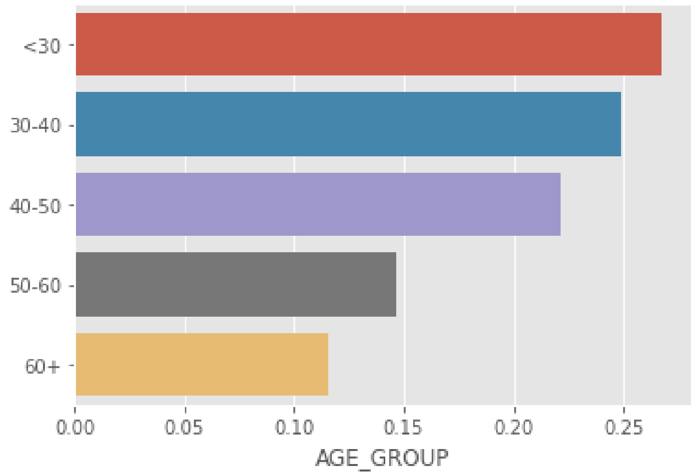
```
# AGE Groups
age_buckets = ['<30','30-40','40-50','50-60','60+']
applications['AGE_GROUP']= pd.cut(applications.AGE, [0,30,40,50,60,999], labels=age_buck
ets)
applications['AGE_GROUP'].value_counts(normalize=True)*100
```

Out[162]:

```
30-40    26.765157
40-50    24.890735
50-60    22.133408
<30      14.640595
60+      11.570105
Name: AGE_GROUP, dtype: float64
```

In [163]:

```
sns.barplot(applications['AGE_GROUP'].value_counts(normalize=True), age_buckets)
plt.show()
```



Previous applications status

In [164]:

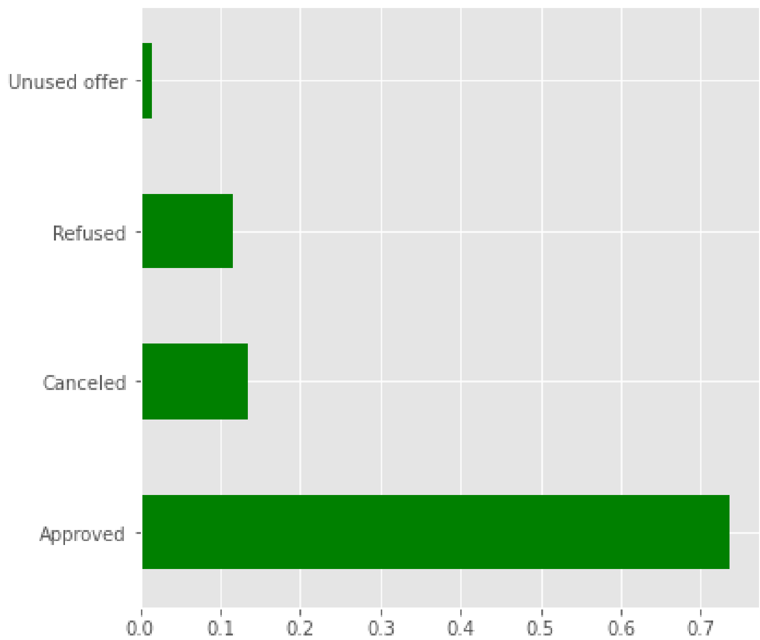
```
applications['PREV_CONTRACT_STATUS'].value_counts(normalize=True)*100
```

Out[164]:

```
Approved        73.472459
Canceled        13.325614
Refused         11.637623
Unused offer     1.564304
Name: PREV_CONTRACT_STATUS, dtype: float64
```

In [174]:

```
plt.figure(figsize=[6,6])
applications['PREV_CONTRACT_STATUS'].value_counts(normalize=True).plot.barh(color='green
')
plt.show()
```

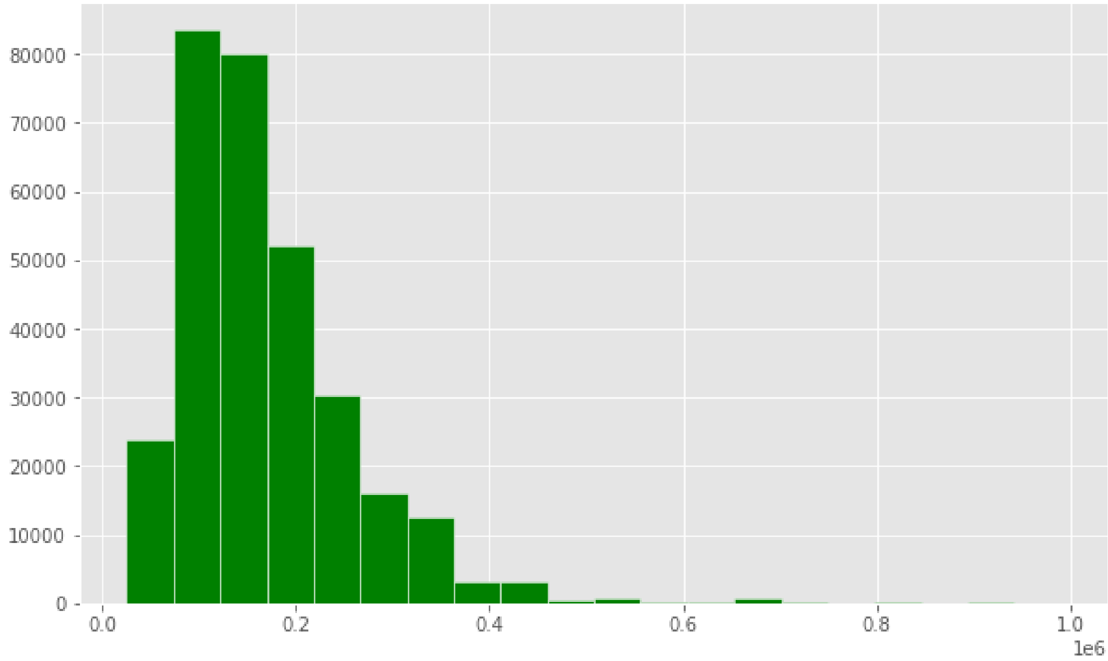


AMT_INCOME_TOTAL

In [175]:

```
plt.figure(figsize=[10,6])
```

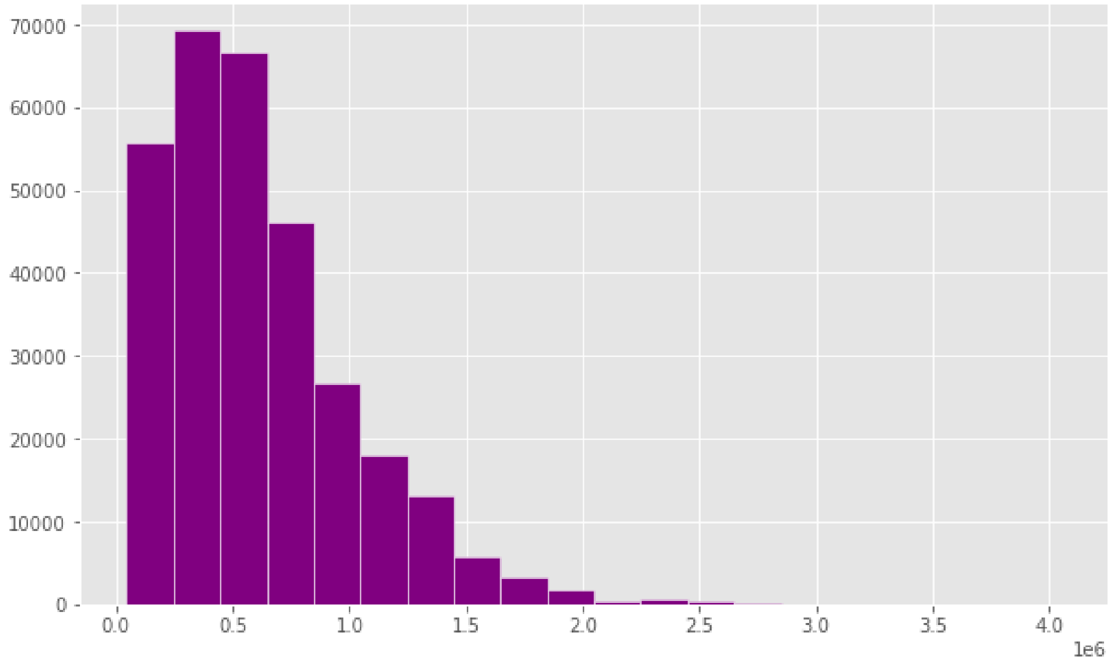
```
plt.figure(figsize=[10,6])
plt.hist(applications[applications['AMT_INCOME_TOTAL']<10**6].AMT_INCOME_TOTAL, bins=20,
color='green', edgecolor='white')
plt.show()
```



AMT_CREDIT

In [177]:

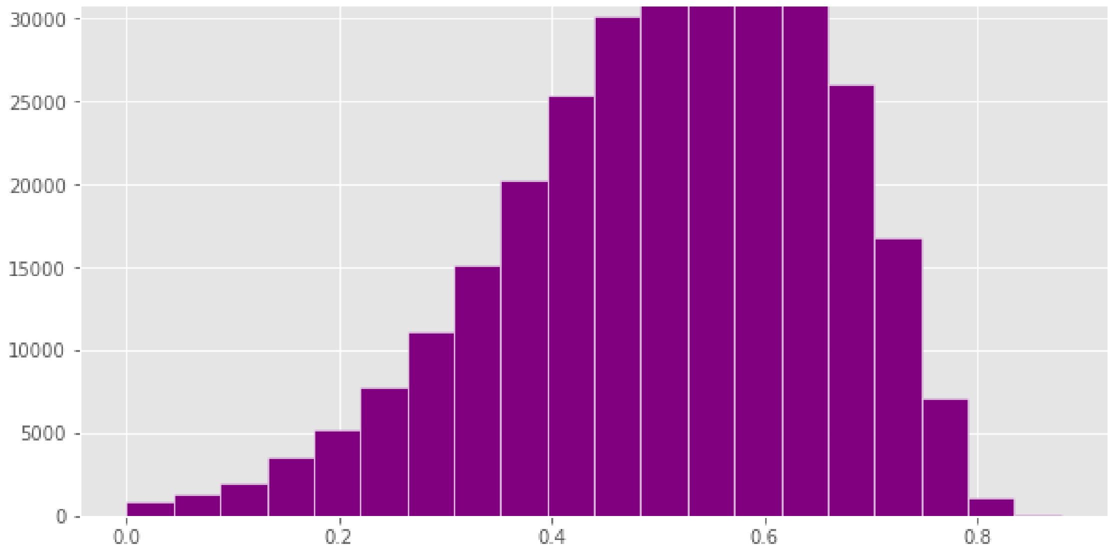
```
plt.figure(figsize=[10,6])
plt.hist(applications['AMT_CREDIT'], bins=20, color='purple',edgecolor='white')
plt.show()
```



EXT_SOURCE_AVG

In [179]:

```
plt.figure(figsize=[10,6])
plt.hist(applications['EXT_SOURCE_AVG'], bins=20, color='Purple',edgecolor='white')
plt.show()
```



FLAG_OWN_REALTY (Individual/owns Property)

In [180]:

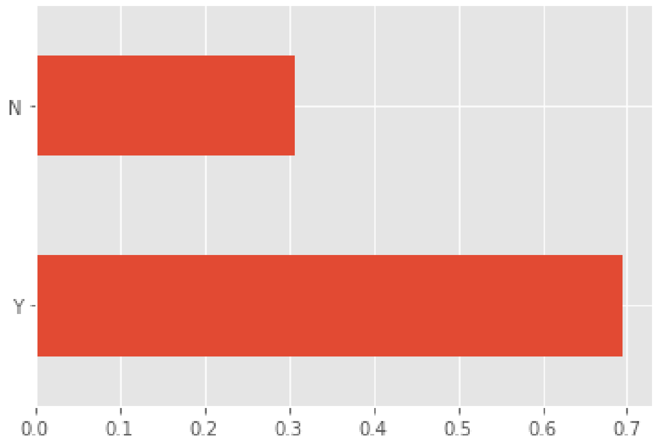
```
applications.FLAG_OWN_REALTY.value_counts(normalize=True)*100
```

Out[180]:

```
Y    69.366976
N    30.633024
Name: FLAG_OWN_REALTY, dtype: float64
```

In [184]:

```
applications['FLAG_OWN_REALTY'].value_counts(normalize=True).plot.barh()
plt.show()
```



In [ ]:

```

```

Bivariate Analysis

Numerical - Categorical

Education level vs Income.

In [187]:

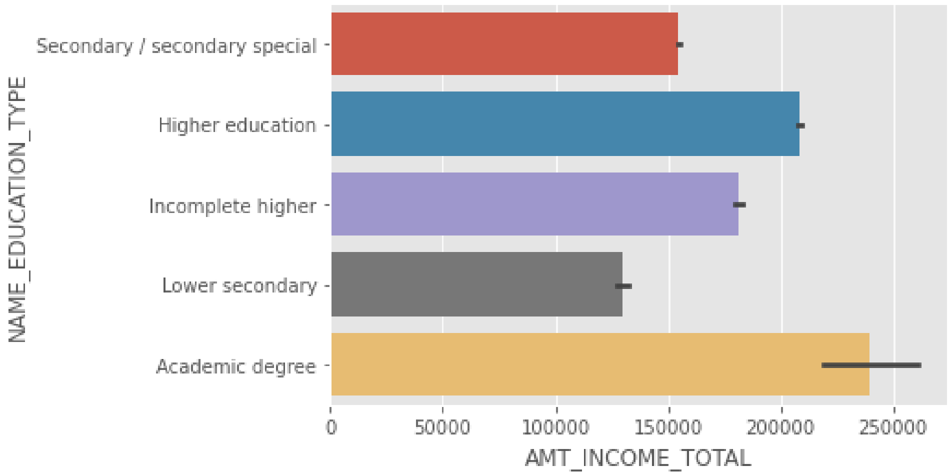
```
applications.groupby('NAME_EDUCATION_TYPE').AMT_INCOME_TOTAL.aggregate(['mean','median']
)
```

Out[187]:

NAME_EDUCATION_TYPE	mean	median
Academic degree	240009.146341	211500.0
Higher education	208652.135993	180000.0
Incomplete higher	181563.812397	157500.0
Lower secondary	129995.499869	112500.0
Secondary / secondary special	154623.483787	135000.0

In [188]:

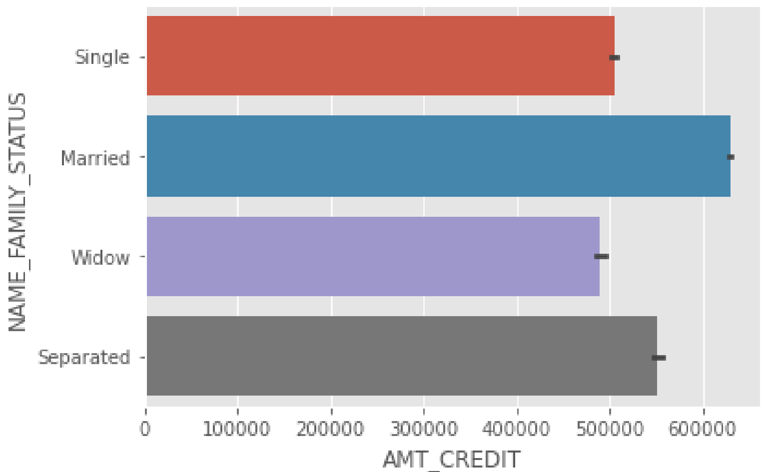
```
sns.barplot(applications['AMT_INCOME_TOTAL'], applications['NAME_EDUCATION_TYPE'])
plt.show()
```



Marital status vs Amount requested for loan

In [189]:

```
sns.barplot(applications['AMT_CREDIT'],applications['NAME_FAMILY_STATUS'])
plt.show()
```



Occuoation type vs Total Income

In [190]:

```
applications.groupby('OCCUPATION_TYPE').AMT_INCOME_TOTAL.aggregate(['mean','median'])
```

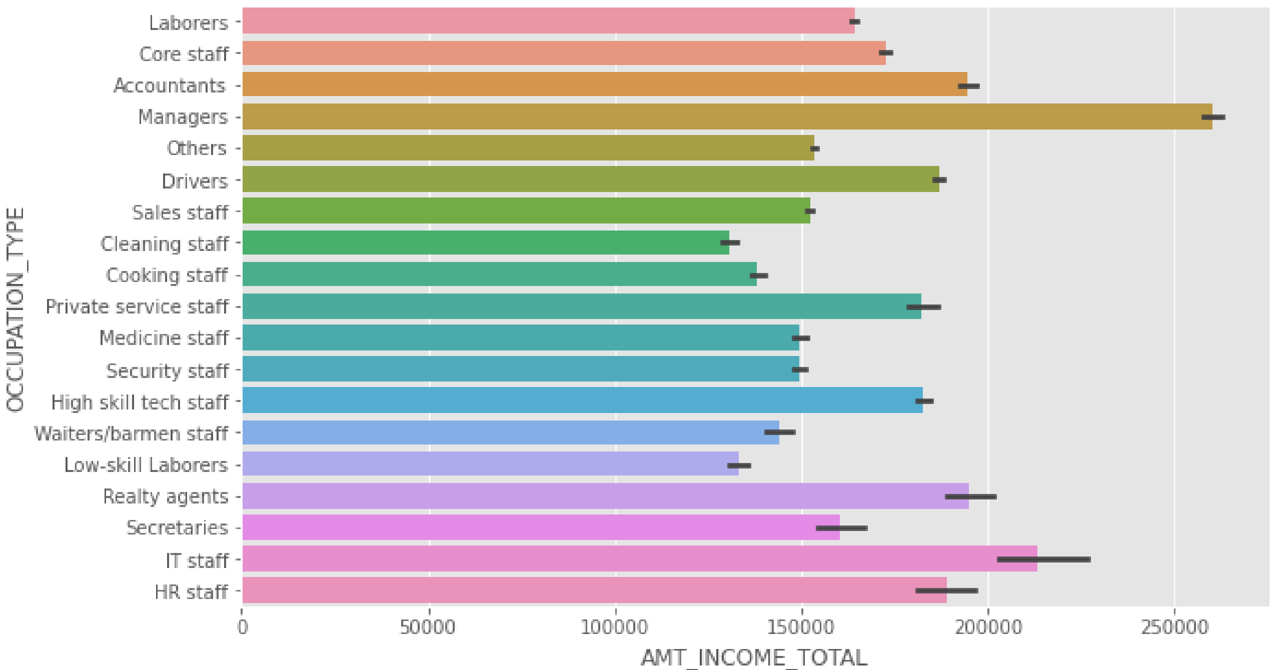
Out[190]:

	mean	median

OCCUPATION_TYPE	mean	median
OCCUPATION_TYPE		
Accountants	194577.550499	178218.0
Cleaning staff	130790.895551	112500.0
Cooking staff	138396.508176	126000.0
Core staff	172656.695254	157500.0
Drivers	187011.606413	180000.0
HR staff	188916.282416	158400.0
High skill tech staff	182842.045683	157500.0
IT staff	213465.601711	180000.0
Laborers	164240.355724	157500.0
Low-skill Laborers	133228.001911	121500.0
Managers	260327.806503	225000.0
Medicine staff	149709.643434	135000.0
Others	153516.031752	135000.0
Private service staff	182334.812783	157500.0
Realty agents	195003.994674	180000.0
Sales staff	152302.874710	135000.0
Secretaries	160541.662069	135000.0
Security staff	149662.695953	135000.0
Waiters/barmen staff	144272.583828	135000.0

In [193]:

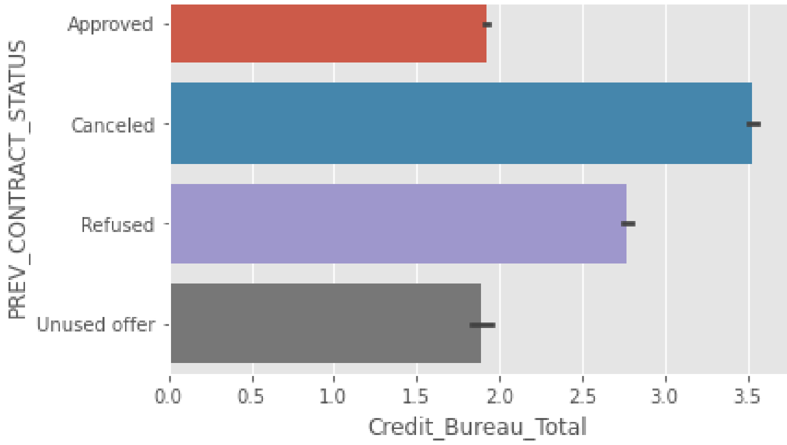
```
plt.figure(figsize=[10,6])
sns.barplot(applications['AMT_INCOME_TOTAL'],applications['OCCUPATION_TYPE'])
plt.show()
```



Total no of Credits Searches vs Status of previous Loan Application

In [194]:

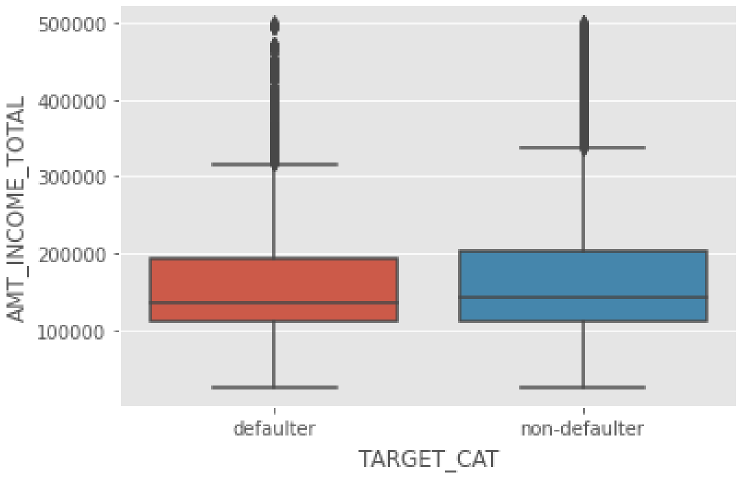
```
sns.barplot(applications['Credit_Bureau_Total'], applications['PREV_CONTRACT_STATUS'])
plt.show()
```



Income Amount vs Target

In [197]:

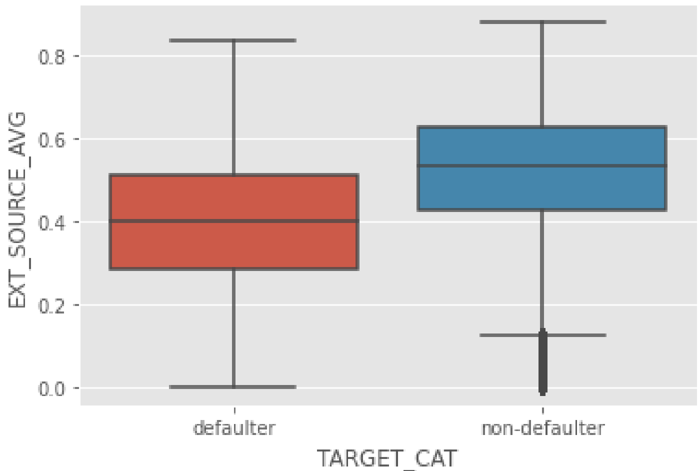
```
sns.boxplot(x=applications['TARGET_CAT'], y=applications[applications.AMT_INCOME_TOTAL <
0.5*10**6].AMT_INCOME_TOTAL)
plt.show()
```



Ext Source Score vs Target

In [200]:

```
sns.boxplot(x=applications['TARGET_CAT'], y=applications.EXT_SOURCE_AVG)
plt.show()
```

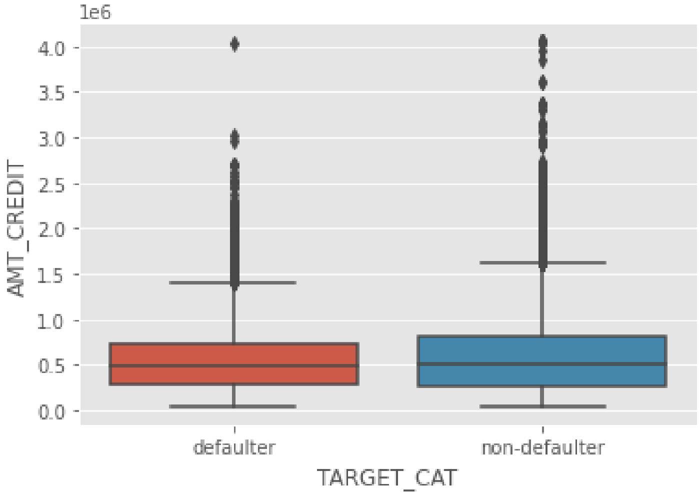


Amount of loan vs Target

In [201]:

```
sns.boxplot(x=applications['TARGET_CAT'], y=applications.AMT_CREDIT)
```

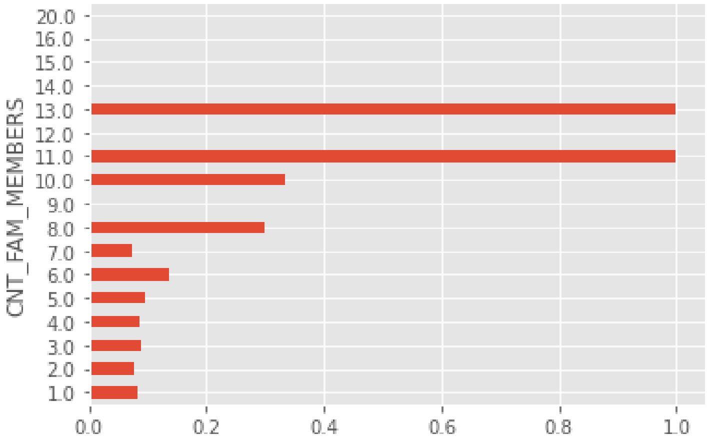
```
sns.boxplot(x=applications['TARGET_CAT'], y=applications.AMT_CREDIT)
plt.show()
```



Family member count vs Target

In [203]:

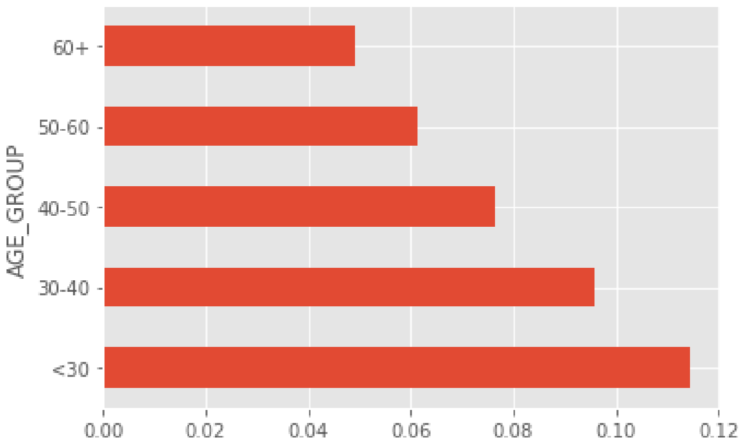
```
applications.groupby('CNT_FAM_MEMBERS').TARGET.mean().plot.barh()
plt.show()
```



Age Group vs Target

In [202]:

```
applications.groupby('AGE_GROUP').TARGET.mean().plot.barh()
plt.show()
```

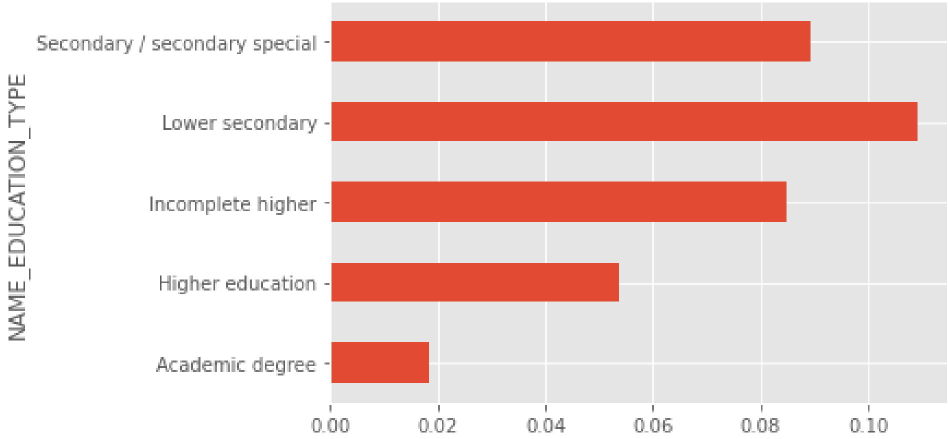


Categorical - Categorical

Education Type vs Target

In [206]:

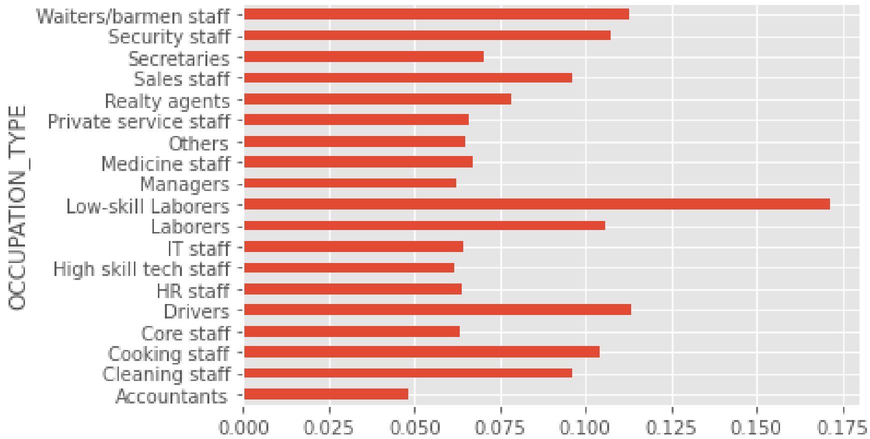
```
applications.groupby('NAME_EDUCATION_TYPE').TARGET.mean().plot.barh()
plt.show()
```



Occupation type vs Target

In [207]:

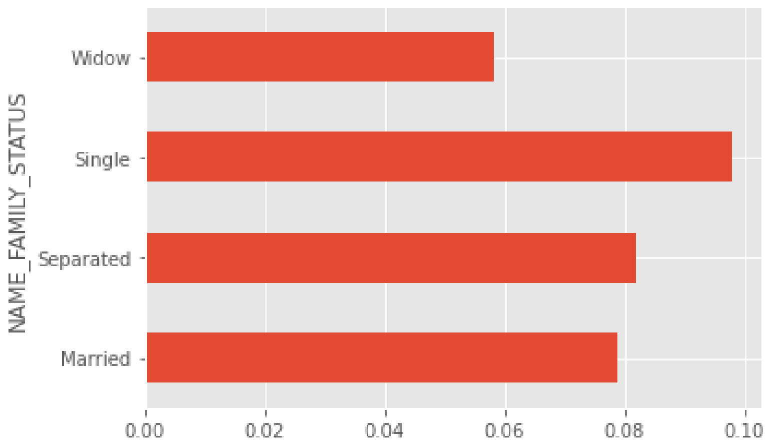
```
applications.groupby('OCCUPATION_TYPE').TARGET.mean().plot.barh()
plt.show()
```



Family Status vs Target

In [208]:

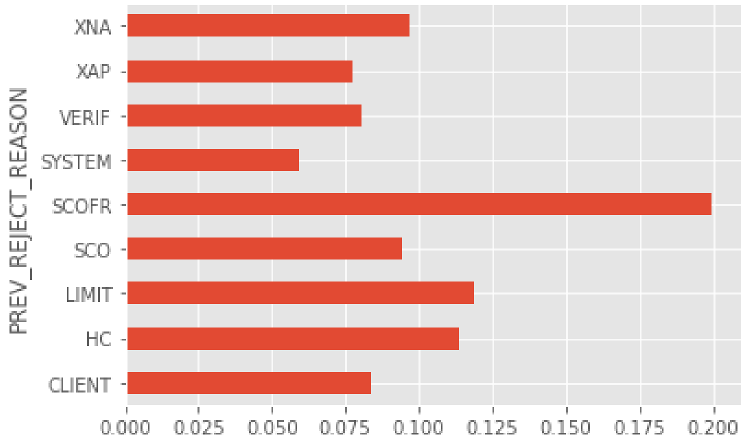
```
applications.groupby('NAME_FAMILY_STATUS').TARGET.mean().plot.barh()
plt.show()
```



Previous rejection reason vs Target

In [209]:

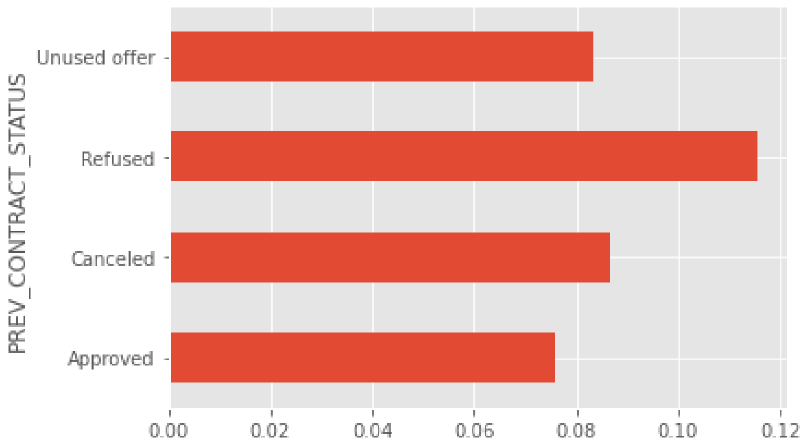
```
applications.groupby('PREV_REJECT_REASON').TARGET.mean().plot.barh()
plt.show()
```



Previous contract status vs Target

In [210]:

```
applications.groupby('PREV_CONTRACT_STATUS').TARGET.mean().plot.barh()
plt.show()
```

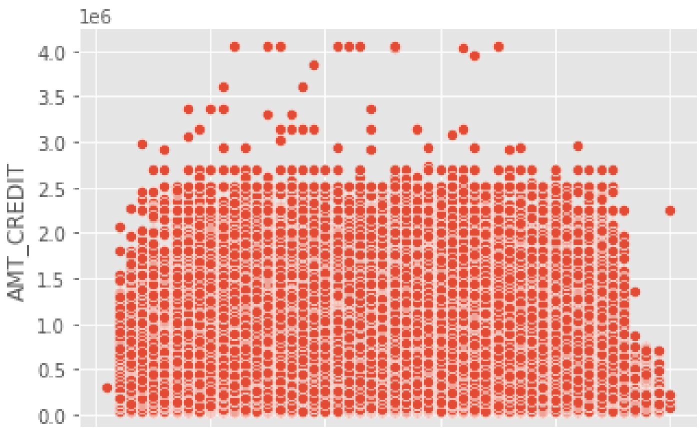


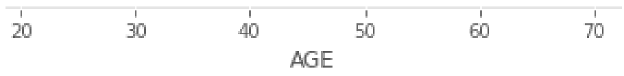
Numerical - Numerical

Age vs Requested loan amount

In [215]:

```
sns.scatterplot(applications['AGE'], applications['AMT_CREDIT'])
plt.show()
```

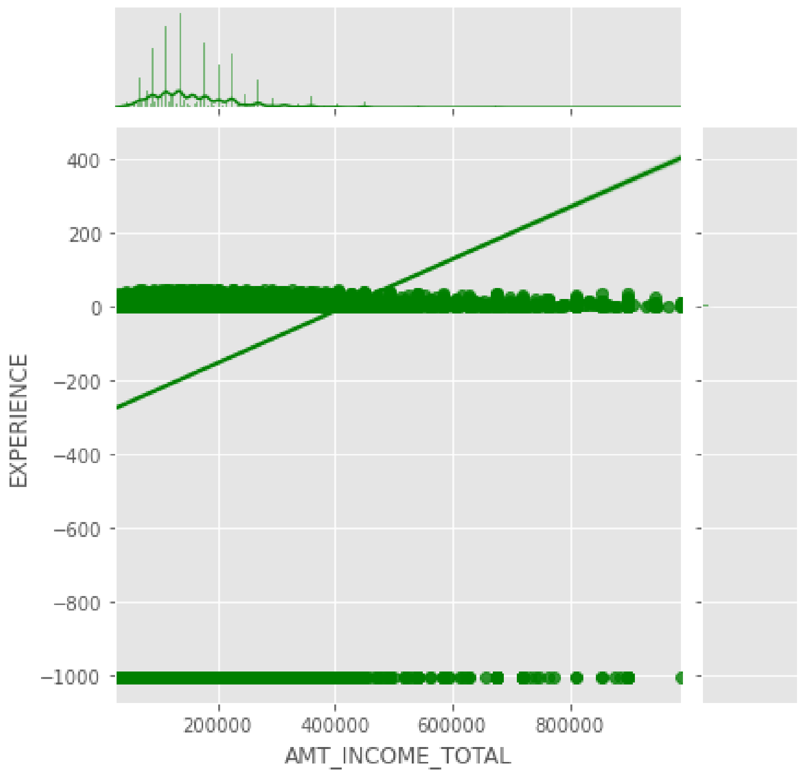




Total Income vs Experience in years

In [216]:

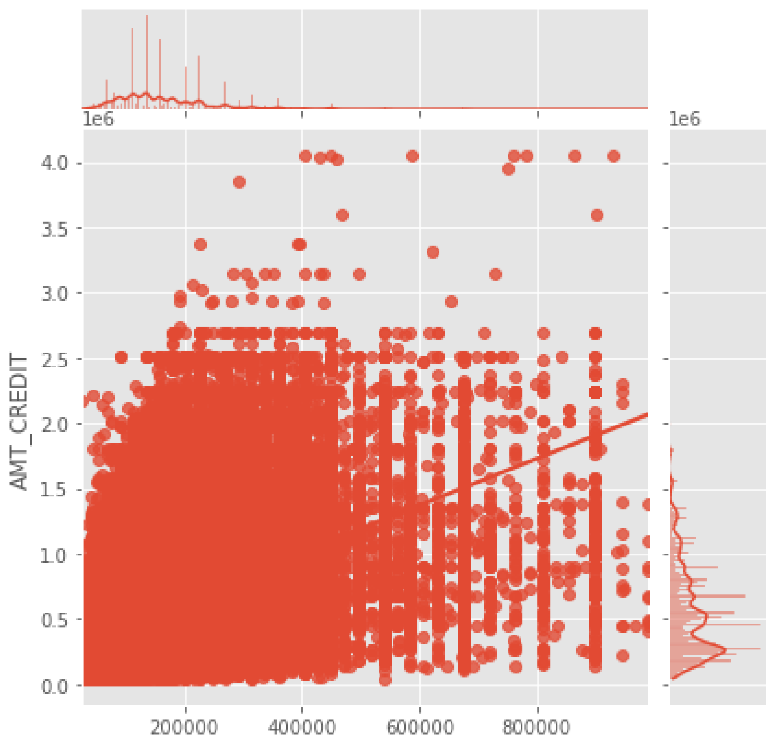
```
sns.jointplot(data=applications[applications.AMT_INCOME_TOTAL < 10**6], x = 'AMT_INCOME_
TOTAL', y='EXPERIENCE', kind='reg', color='green')
plt.show()
```



Total income vs Amount requested for loan

In [212]:

```
sns.jointplot(data=applications[applications.AMT_INCOME_TOTAL < 10**6], x = 'AMT_INCOME_
TOTAL', y='AMT_CREDIT', kind='reg')
plt.show()
```



AMT_INCOME_TOTAL

In [ ]:

```

```

Multivariate Analysis

Family status vs Occupation vs Target

In [218]:

```
data = pd.pivot_table(data=applications, index='OCCUPATION_TYPE',columns='NAME_FAMILY_ST
ATUS', values='TARGET')
data
```

Out[218]:

NAME_FAMILY_STATUS	Married	Separated	Single	Widow
OCCUPATION_TYPE				
Accountants	0.045846	0.053352	0.058704	0.044521
Cleaning staff	0.097342	0.078652	0.122066	0.067265
Cooking staff	0.101545	0.084135	0.144550	0.059859
Core staff	0.060456	0.057576	0.080491	0.038880
Drivers	0.106101	0.145655	0.147986	0.153153
HR staff	0.056511	0.065217	0.066667	0.200000
High skill tech staff	0.059609	0.061252	0.072008	0.048327
IT staff	0.072674	0.000000	0.060000	0.000000
Laborers	0.101180	0.109462	0.132301	0.069250
Low-skill Laborers	0.164499	0.164835	0.204545	0.041667
Managers	0.059686	0.068750	0.077488	0.043290
Medicine staff	0.068426	0.060514	0.073257	0.038554
Others	0.063827	0.068443	0.077041	0.056668
Private service staff	0.062396	0.105263	0.061896	0.034483
Realty agents	0.073171	0.115385	0.093960	0.000000
Sales staff	0.095648	0.097902	0.101770	0.077991
Secretaries	0.079511	0.000000	0.059633	0.038462
Security staff	0.100552	0.118834	0.140472	0.086486
Waiters/barmen staff	0.113074	0.070000	0.131653	0.047619

In [219]:

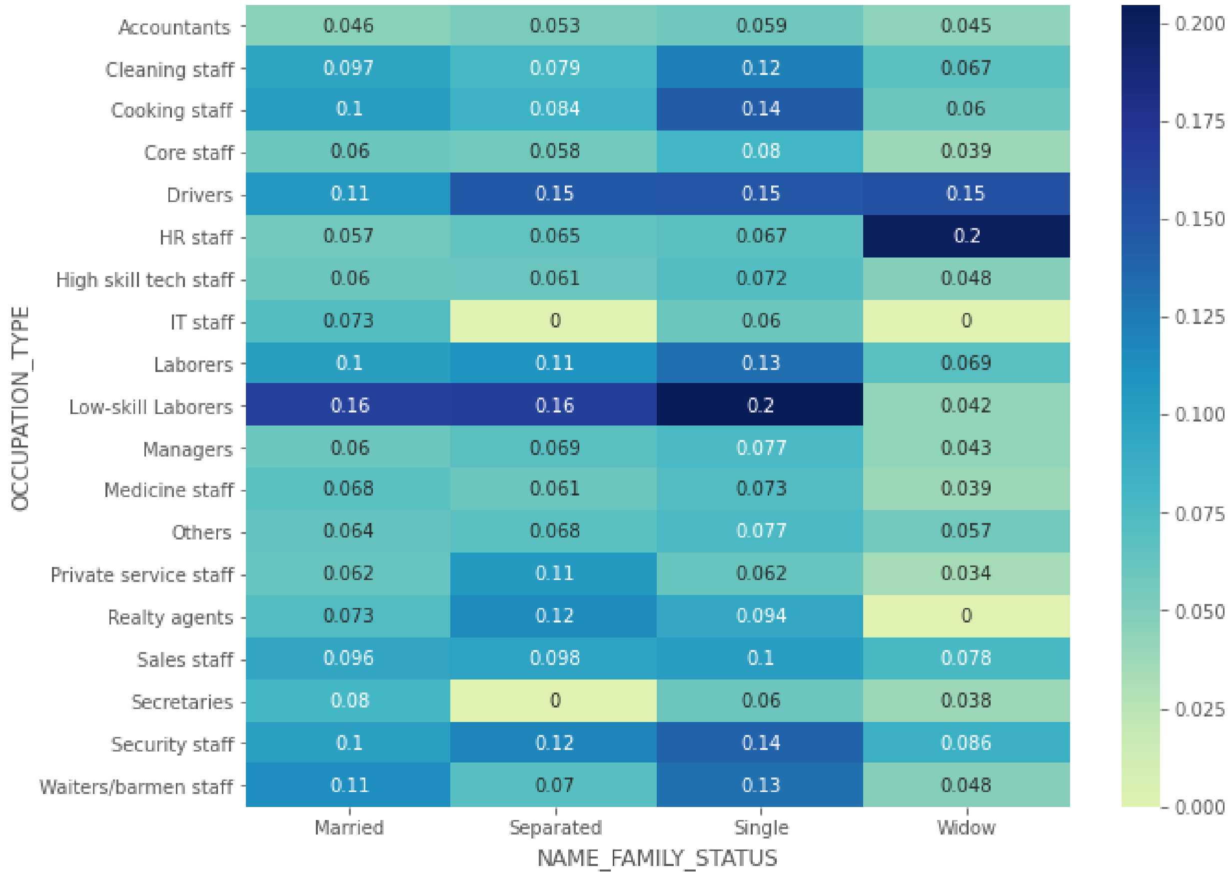
```
applications.TARGET.value_counts(normalize=True)
```

Out[219]:

```
0    0.919274
1    0.080726
Name: TARGET, dtype: float64
```

In [222]:

```
plt.figure(figsize=[10,8])
sns.heatmap(data, annot=True, cmap='YlGnBu', center=0.081)
plt.show()
```



Family status vs Education type vs Target

In [223]:

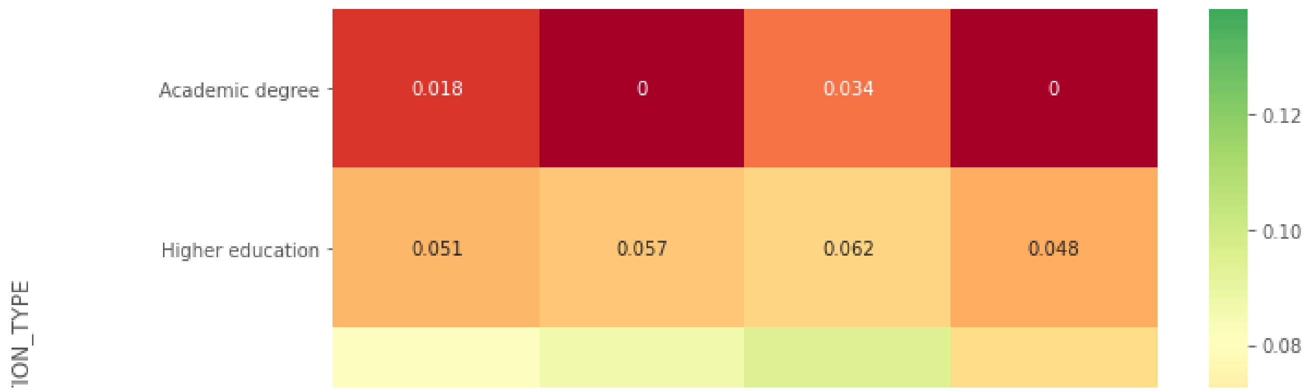
```
data_1 = pd.pivot_table(data=applications, index='NAME_EDUCATION_TYPE',columns='NAME_FAM
ILY_STATUS', values='TARGET')
data_1
```

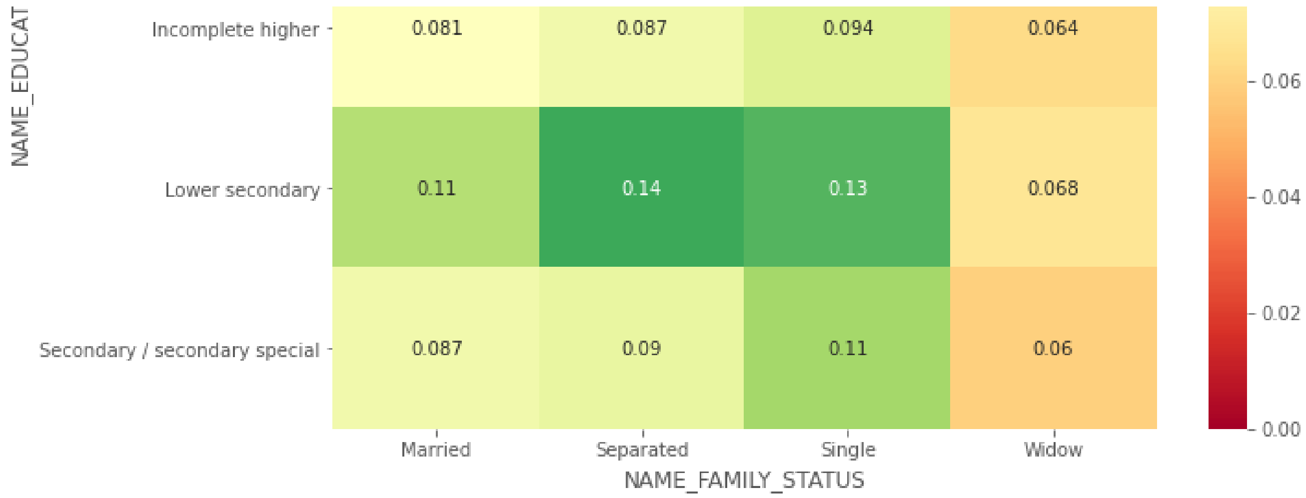
Out[223]:

NAME_FAMILY_STATUS	Married	Separated	Single	Widow
NAME_EDUCATION_TYPE				
Academic degree	0.017544	0.000000	0.034483	0.000000
Higher education	0.051481	0.057346	0.062049	0.048094
Incomplete higher	0.081425	0.086643	0.094143	0.063584
Lower secondary	0.108170	0.138249	0.132988	0.067961
Secondary / secondary special	0.087434	0.089904	0.113243	0.059666

In [226]:

```
plt.figure(figsize=[10,8])
sns.heatmap(data_1, annot=True, cmap='RdYlGn', center=0.081)
plt.show()
```





Correlation between target and prominent numeric variables

In [229]:

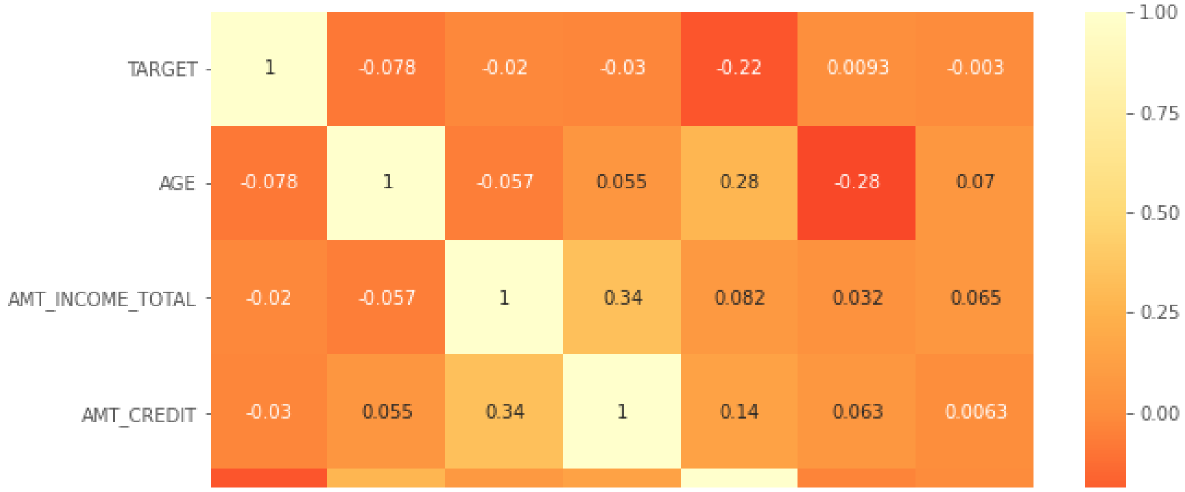
```
data_3 = applications[['TARGET','AGE','AMT_INCOME_TOTAL','AMT_CREDIT','EXT_SOURCE_AVG','
CNT_FAM_MEMBERS','Credit_Bureau_Total']].corr()
data_3
```

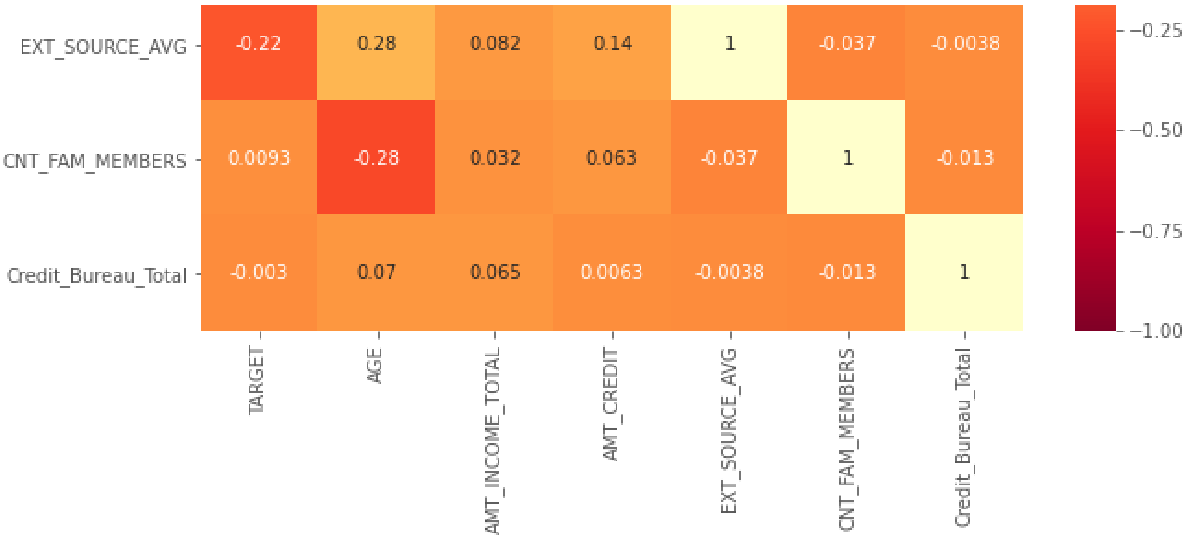
Out[229]:

	TARGET	AGE	AMT_INCOME_TOTAL	AMT_CREDIT	EXT_SOURCE_AVG	CNT_FAM_MEMBERS	Cre
TARGET	1.000000	-0.078232	-0.020457	-0.030369	-0.222036	0.009298	
AGE	-0.078232	1.000000	-0.056616	0.055392	0.279730	-0.278894	
AMT_INCOME_TOTAL	-0.020457	-0.056616	1.000000	0.342172	0.082098	0.032363	
AMT_CREDIT	-0.030369	0.055392	0.342172	1.000000	0.143684	0.063160	
EXT_SOURCE_AVG	-0.222036	0.279730	0.082098	0.143684	1.000000	-0.037363	
CNT_FAM_MEMBERS	0.009298	-0.278894	0.032363	0.063160	-0.037363	1.000000	
Credit_Bureau_Total	-0.002985	0.069799	0.065497	0.006282	-0.003752	-0.013251	

In [234]:

```
plt.figure(figsize=[10,8])
sns.heatmap(data_3, annot=True, cmap='YlOrRd_r', vmin=-1, vmax=1)
plt.show()
```





Occupation type vs Education type vs Target

In [235]:

```
data_2 = pd.pivot_table(data=applications, index='OCCUPATION_TYPE', columns='NAME_EDUCAT
ION_TYPE', values='TARGET')
data_2
```

Out[235]:

NAME_EDUCATION_TYPE	Academic degree	Higher education	Incomplete higher	Lower secondary	Secondary / secondary special
OCCUPATION_TYPE					
Accountants	0.125000	0.038813	0.056180	0.166667	0.062077
Cleaning staff	0.000000	0.084000	0.102041	0.138889	0.095664
Cooking staff	0.000000	0.088993	0.093220	0.171875	0.105135
Core staff	0.000000	0.045670	0.083902	0.105691	0.078621
Drivers	0.000000	0.083415	0.103870	0.128514	0.117219
HR staff	NaN	0.044818	0.083333	NaN	0.100000
High skill tech staff	0.000000	0.047261	0.070085	0.129032	0.071405
IT staff	NaN	0.049853	0.038462	0.000000	0.113636
Laborers	0.000000	0.076011	0.086326	0.160274	0.109829
Low-skill Laborers	0.000000	0.098765	0.200000	0.173913	0.174166
Managers	0.000000	0.048091	0.080559	0.089286	0.078558
Medicine staff	NaN	0.043043	0.020548	0.116883	0.070677
Others	0.017544	0.048779	0.076500	0.076560	0.069237
Private service staff	0.000000	0.065511	0.043860	0.000000	0.067797
Realty agents	0.000000	0.044053	0.191489	0.000000	0.084388
Sales staff	0.111111	0.075872	0.102861	0.135965	0.100141
Secretaries	NaN	0.062370	0.114583	0.000000	0.070442
Security staff	0.000000	0.080390	0.153846	0.144330	0.109516
Waiters/barmen staff	NaN	0.089385	0.095745	0.058824	0.119093

In [237]:

```
plt.figure(figsize=[10,8])
sns.heatmap(data_2, annot=True, cmap='RdYlGn', center=0.081)
```

Out[237]:

```
<AxesSubplot:xlabel='NAME_EDUCATION_TYPE', ylabel='OCCUPATION_TYPE'>
```

END ------