

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.style as style
import seaborn as sns
import itertools
%matplotlib inline
```

```
In [2]: style.use('seaborn-poster')
style.use('fivethirtyeight')
```

```
C:\Users\DELL\i7\AppData\Local\Temp\ipykernel_12488\3592107732.py:1: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_8-<style>'. Alternatively, directly use the seaborn API instead.
style.use('seaborn-poster')
```

```
In [3]: import warnings
warnings.filterwarnings('ignore')
```

```
In [4]: pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
pd.set_option('display.expand_frame_repr', False)
```

```
In [5]: import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
In [6]: applicationDF =pd.read_csv(r'C:\Users\DELL i7\Desktop\11th_resume project\applicationDF.csv')
previousDF=pd.read_csv(r'C:\Users\DELL i7\Desktop\11th_resume project\previousDF.csv')
applicationDF.head()
```

Out[6]:

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

```
In [7]: previousDF.head()
```

Out[7]:

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_GOODS_PRICE
0	2030495	271877	Consumer loans	1730.430	17145.0	10000.0
1	2802425	108129	Cash loans	25188.615	607500.0	10000.0
2	2523466	122040	Cash loans	15060.735	112500.0	10000.0
3	2819243	176158	Cash loans	47041.335	450000.0	10000.0
4	1784265	202054	Cash loans	31924.395	337500.0	10000.0

```
In [8]: print("Database dimension - applicationDF      : ",applicationDF.shape)
print("Database dimension - previousDF           : ",previousDF.shape)
```

```
print("Database size - applicationDF            : ",applicationDF.size)
print("Database size - previousDF               : ",previousDF.size)
```

```
Database dimension - applicationDF      : (307511, 122)
Database dimension - previousDF           : (1670214, 37)
Database size - applicationDF            : 37516342
Database size - previousDF               : 61797918
```

In [9]: applicationDF.info(verbose=True)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 122 columns):
 #   Column           Dtype  
 --- 
 0   SK_ID_CURR       int64  
 1   TARGET           int64  
 2   NAME_CONTRACT_TYPE  object 
 3   CODE_GENDER      object 
 4   FLAG_OWN_CAR     object 
 5   FLAG_OWN_REALTY  object 
 6   CNT_CHILDREN     int64  
 7   AMT_INCOME_TOTAL float64 
 8   AMT_CREDIT        float64 
 9   AMT_ANNUITY       float64 
 10  AMT_GOODS_PRICE  float64 
 11  NAME_TYPE_SUITE  object 
 12  NAME_INCOME_TYPE object 
 13  NAME_EDUCATION_TYPE object 
 14  NAME_FAMILY_STATUS object 
 15  NAME_HOUSING_TYPE object 
```

In [10]: applicationDF.describe()

Out[10]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT...
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	3074
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	271
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	144
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	16
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	165
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	249
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	345
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	2580

In [11]: `previousDF.describe()`

Out[11]:

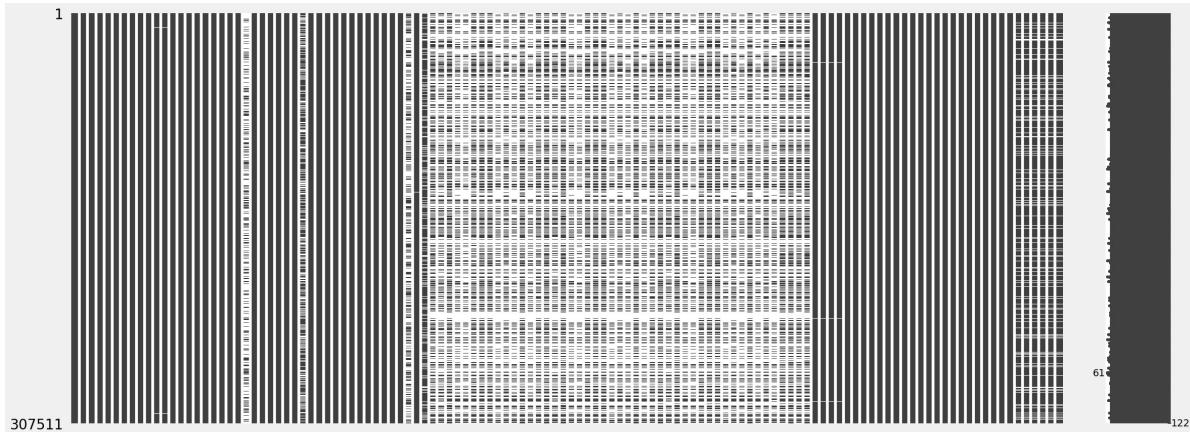
	SK_ID_PREV	SK_ID_CURR	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOW
count	1.670214e+06	1.670214e+06	1.297979e+06	1.670214e+06	1.670213e+06	1
mean	1.923089e+06	2.783572e+05	1.595512e+04	1.752339e+05	1.961140e+05	6
std	5.325980e+05	1.028148e+05	1.478214e+04	2.927798e+05	3.185746e+05	2
min	1.000001e+06	1.000010e+05	0.000000e+00	0.000000e+00	0.000000e+00	-
25%	1.461857e+06	1.893290e+05	6.321780e+03	1.872000e+04	2.416050e+04	1
50%	1.923110e+06	2.787145e+05	1.125000e+04	7.104600e+04	8.054100e+04	-
75%	2.384280e+06	3.675140e+05	2.065842e+04	1.803600e+05	2.164185e+05	1
max	2.845382e+06	4.562550e+05	4.180581e+05	6.905160e+06	6.905160e+06	1

In [12]: `pip install missingno`

```
Requirement already satisfied: missingno in c:\users\dell i7\anaconda3\lib\site-packages (0.5.2)
Requirement already satisfied: numpy in c:\users\dell i7\anaconda3\lib\site-packages (from missingno) (1.24.3)
Requirement already satisfied: matplotlib in c:\users\dell i7\anaconda3\lib\site-packages (from missingno) (3.7.1)
Requirement already satisfied: scipy in c:\users\dell i7\anaconda3\lib\site-packages (from missingno) (1.10.1)
Requirement already satisfied: seaborn in c:\users\dell i7\anaconda3\lib\site-packages (from missingno) (0.12.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell i7\anaconda3\lib\site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: pandas>=0.25 in c:\users\dell i7\anaconda3\lib\site-packages (from seaborn->missingno) (1.5.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell i7\anaconda3\lib\site-packages (from pandas>=0.25->seaborn->missingno) (2022.7)
Requirement already satisfied: six>=1.5 in c:\users\dell i7\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib->missingno) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
In [13]: import missingno as mn  
mn.matrix(applicationDF)
```

Out[13]: <Axes: >

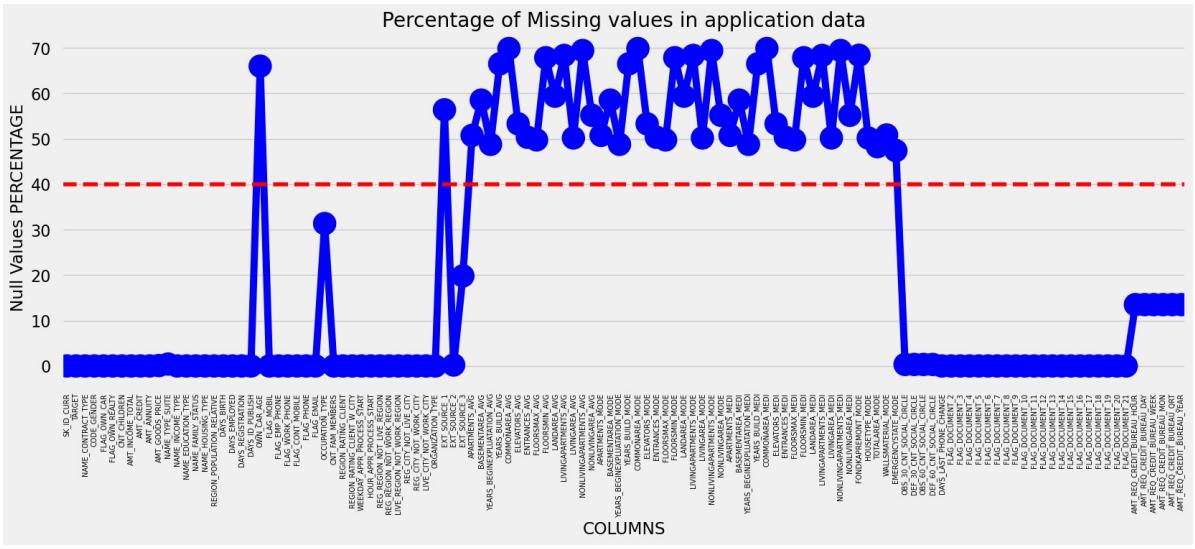


```
In [14]: round(applicationDF.isnull().sum() / applicationDF.shape[0] * 100.00,2)
```

Out[14]:

SK_ID_CURR	0.00
TARGET	0.00
NAME_CONTRACT_TYPE	0.00
CODE_GENDER	0.00
FLAG_OWN_CAR	0.00
FLAG_OWN_REALTY	0.00
CNT_CHILDREN	0.00
AMT_INCOME_TOTAL	0.00
AMT_CREDIT	0.00
AMT_ANNUITY	0.00
AMT_GOODS_PRICE	0.09
NAME_TYPE_SUITE	0.42
NAME_INCOME_TYPE	0.00
NAME_EDUCATION_TYPE	0.00
NAME_FAMILY_STATUS	0.00
NAME_HOUSING_TYPE	0.00
REGION_POPULATION_RELATIVE	0.00
DAYS_BIRTH	0.00
DAYS_EMPLOYED	0.00
DAYS_REGISTRATION	0.00

```
In [15]: null_applicationDF = pd.DataFrame((applicationDF.isnull().sum())*100/applicationDF.count())
null_applicationDF.columns = ['Column Name', 'Null Values Percentage']
fig = plt.figure(figsize=(18,6))
ax = sns.pointplot(x="Column Name",y="Null Values Percentage",data=null_applicationDF)
plt.xticks(rotation =90,fontsize =7)
ax.axhline(40, ls='--',color='red')
plt.title("Percentage of Missing values in application data")
plt.ylabel("Null Values PERCENTAGE")
plt.xlabel("COLUMNS")
plt.show()
```



```
In [16]: nullcol_40_application = null_applicationDF=null_applicationDF["Null Values Percentage"]
nullcol_40_application
```

Out[16]:

	Column Name	Null Values Percentage
21	OWN_CAR_AGE	65.990810
41	EXT_SOURCE_1	56.381073
44	APARTMENTS_AVG	50.749729
45	BASEMENTAREA_AVG	58.515956
46	YEARS_BEGINEXPLUATATION_AVG	48.781019
47	YEARS_BUILD_AVG	66.497784
48	COMMONAREA_AVG	69.872297
49	ELEVATORS_AVG	53.295980
50	ENTRANCES_AVG	50.348768
51	FLOORSMAX_AVG	49.760822
52	FLOORSMIN_AVG	67.848630
53	LANDAREA_AVG	59.376738
54	LIVINGAPARTMENTS_AVG	68.354953
55	LIVINGAREA_AVG	50.193326
56	NONLIVINGAPARTMENTS_AVG	69.432963
57	NONLIVINGAREA_AVG	55.179164
58	APARTMENTS_MODE	50.749729
59	BASEMENTAREA_MODE	58.515956
60	YEARS_BEGINEXPLUATATION_MODE	48.781019
61	YEARS_BUILD_MODE	66.497784
62	COMMONAREA_MODE	69.872297
63	ELEVATORS_MODE	53.295980
64	ENTRANCES_MODE	50.348768
65	FLOORSMAX_MODE	49.760822
66	FLOORSMIN_MODE	67.848630
67	LANDAREA_MODE	59.376738
68	LIVINGAPARTMENTS_MODE	68.354953
69	LIVINGAREA_MODE	50.193326
70	NONLIVINGAPARTMENTS_MODE	69.432963
71	NONLIVINGAREA_MODE	55.179164
72	APARTMENTS_MEDI	50.749729
73	BASEMENTAREA_MEDI	58.515956
74	YEARS_BEGINEXPLUATATION_MEDI	48.781019

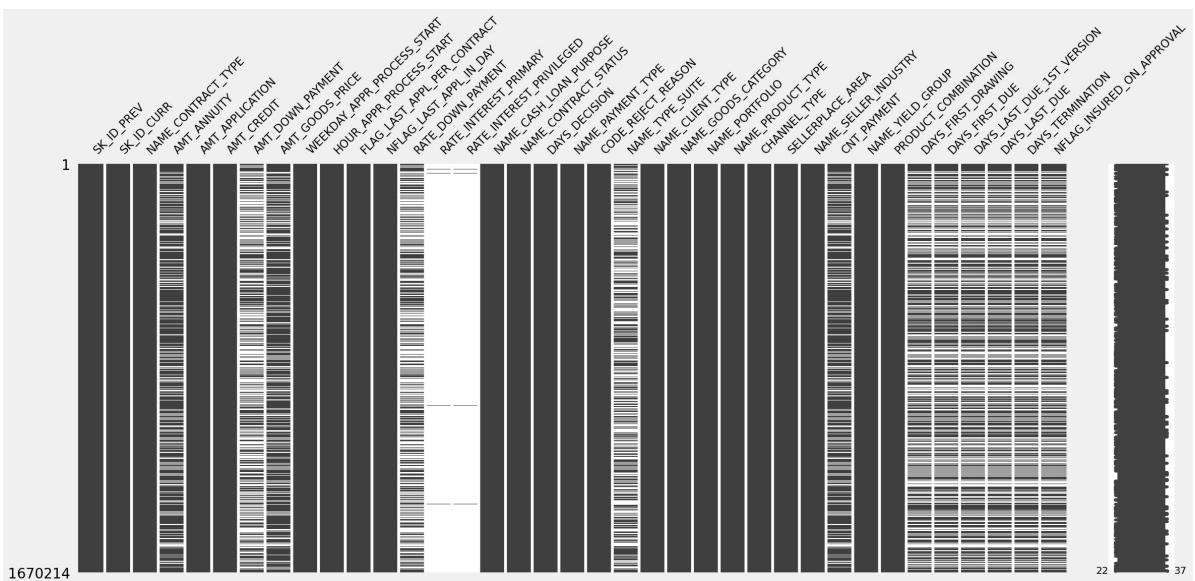
	Column Name	Null Values Percentage
75	YEARS_BUILD_MEDI	66.497784
76	COMMONAREA_MEDI	69.872297
77	ELEVATORS_MEDI	53.295980
78	ENTRANCES_MEDI	50.348768
79	FLOORSMAX_MEDI	49.760822
80	FLOORSMIN_MEDI	67.848630
81	LANDAREA_MEDI	59.376738
82	LIVINGAPARTMENTS_MEDI	68.354953
83	LIVINGAREA_MEDI	50.193326
84	NONLIVINGAPARTMENTS_MEDI	69.432963
85	NONLIVINGAREA_MEDI	55.179164
86	FONDKAPREMONT_MODE	68.386172
87	HOUSETYPE_MODE	50.176091
88	TOTALAREA_MODE	48.268517
89	WALLSMATERIAL_MODE	50.840783

In [17]: `len(nullcol_40_application)`

Out[17]: 49

In [18]: `mn.matrix(previousDF)`

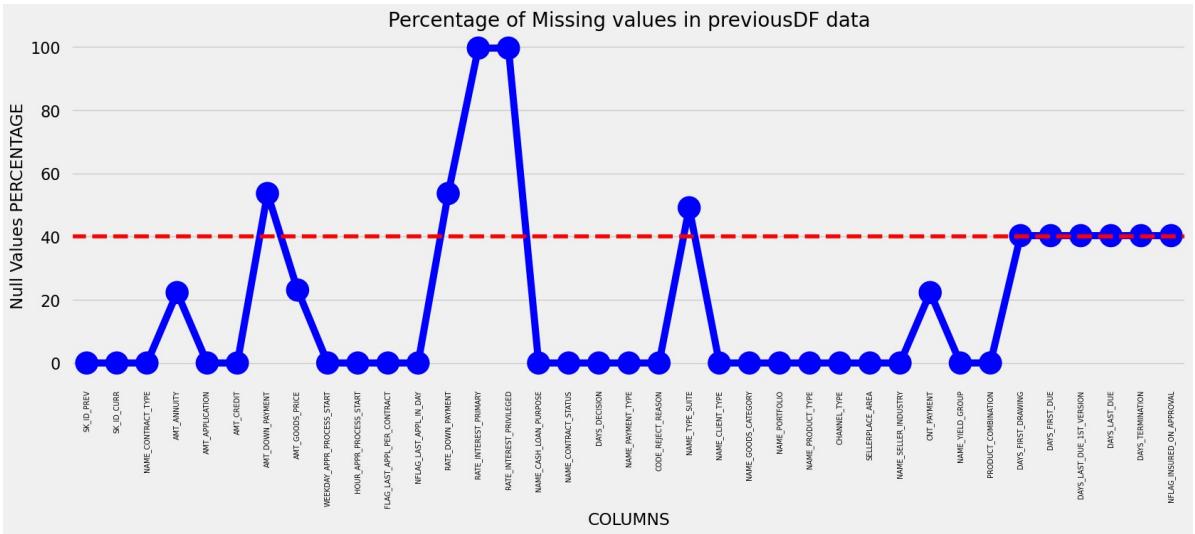
Out[18]: <Axes: >



```
In [19]: round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)
```

```
Out[19]: SK_ID_PREV           0.00  
SK_ID_CURR            0.00  
NAME_CONTRACT_TYPE    0.00  
AMT_ANNUITY           22.29  
AMT_APPLICATION       0.00  
AMT_CREDIT             0.00  
AMT_DOWN_PAYMENT      53.64  
AMT_GOODS_PRICE        23.08  
WEEKDAY_APPR_PROCESS_START 0.00  
HOUR_APPR_PROCESS_START 0.00  
FLAG_LAST_APPL_PER_CONTRACT 0.00  
NFLAG_LAST_APPL_IN_DAY 0.00  
RATE_DOWN_PAYMENT      53.64  
RATE_INTEREST_PRIMARY   99.64  
RATE_INTEREST_PRIVILEGED 99.64  
NAME_CASH_LOAN_PURPOSE 0.00  
NAME_CONTRACT_STATUS    0.00  
DAYS_DECISION          0.00  
NAME_PAYMENT_TYPE       0.00  
CODE_REJECT_REASON     0.00  
NAME_TYPE_SUITE         49.12  
NAME_CLIENT_TYPE        0.00  
NAME_GOODS_CATEGORY     0.00  
NAME_PORTFOLIO          0.00  
NAME_PRODUCT_TYPE       0.00  
CHANNEL_TYPE            0.00  
SELLERPLACE_AREA        0.00  
NAME_SELLER_INDUSTRY   0.00  
CNT_PAYMENT             22.29  
NAME_YIELD_GROUP        0.00  
PRODUCT_COMBINATION     0.02  
DAYS_FIRST_DRAWING     40.30  
DAYS_FIRST_DUE          40.30  
DAYS_LAST_DUE_1ST_VERSION 40.30  
DAYS_LAST_DUE           40.30  
DAYS_TERMINATION        40.30  
NFLAG_INSURED_ON_APPROVAL 40.30  
dtype: float64
```

```
In [20]: null_previousDF = pd.DataFrame((previousDF.isnull().sum())*100/previousDF.shape[0])
null_previousDF.columns = ['Column Name', 'Null Values Percentage']
fig = plt.figure(figsize=(18,6))
ax = sns.pointplot(x="Column Name",y="Null Values Percentage",data=null_previousDF)
plt.xticks(rotation =90,fontsize =7)
ax.axhline(40, ls='--',color='red')
plt.title("Percentage of Missing values in previousDF data")
plt.ylabel("Null Values PERCENTAGE")
plt.xlabel("COLUMNS")
plt.show()
```



```
In [21]: nullcol_40_previous = null_previousDF[null_previousDF["Null Values Percentage"] > 40]
```

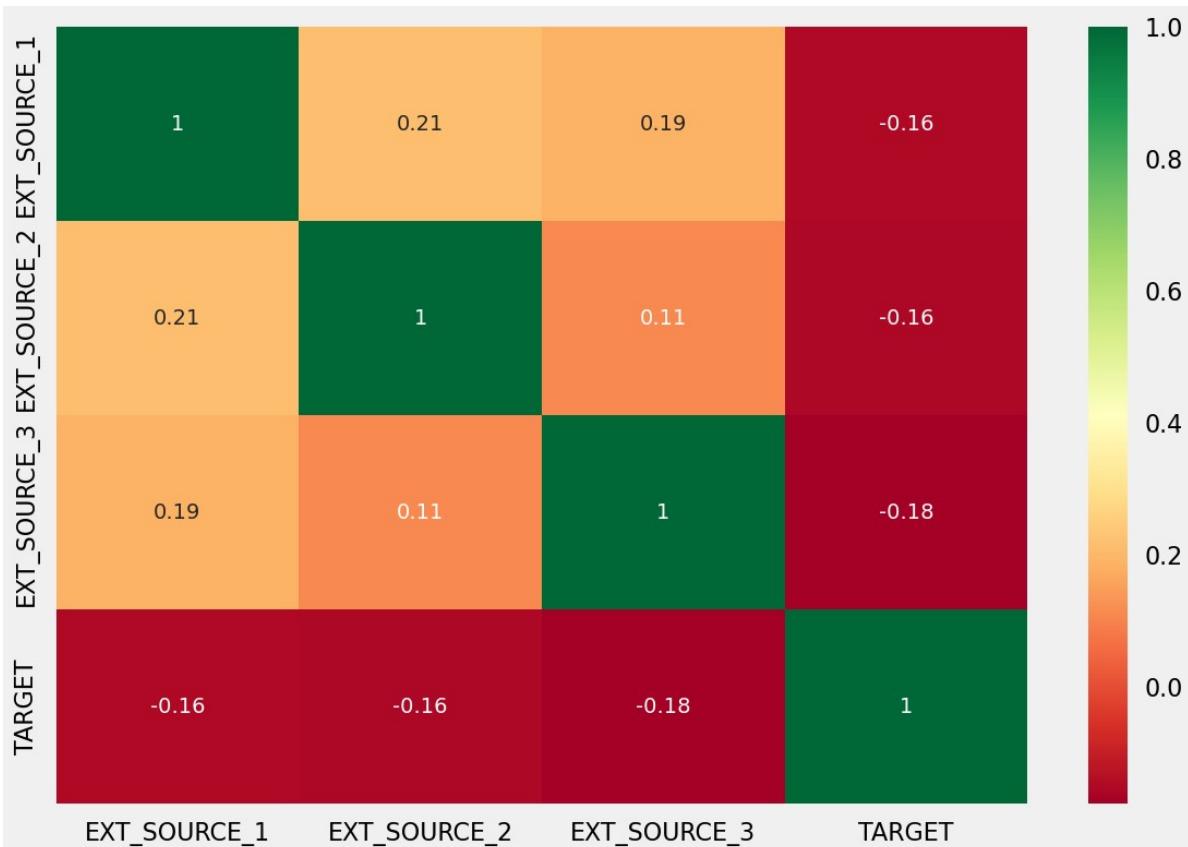
Out[21]:

	Column Name	Null Values Percentage
6	AMT_DOWN_PAYMENT	53.636480
12	RATE_DOWN_PAYMENT	53.636480
13	RATE_INTEREST_PRIMARY	99.643698
14	RATE_INTEREST_PRIVILEGED	99.643698
20	NAME_TYPE_SUITE	49.119754
31	DAYS_FIRST_DRAWING	40.298129
32	DAYS_FIRST_DUE	40.298129
33	DAYS_LAST_DUE_1ST_VERSION	40.298129
34	DAYS_LAST_DUE	40.298129
35	DAYS_TERMINATION	40.298129
36	NFLAG_INSURED_ON_APPROVAL	40.298129

```
In [22]: len(nullcol_40_previous)
```

Out[22]: 11

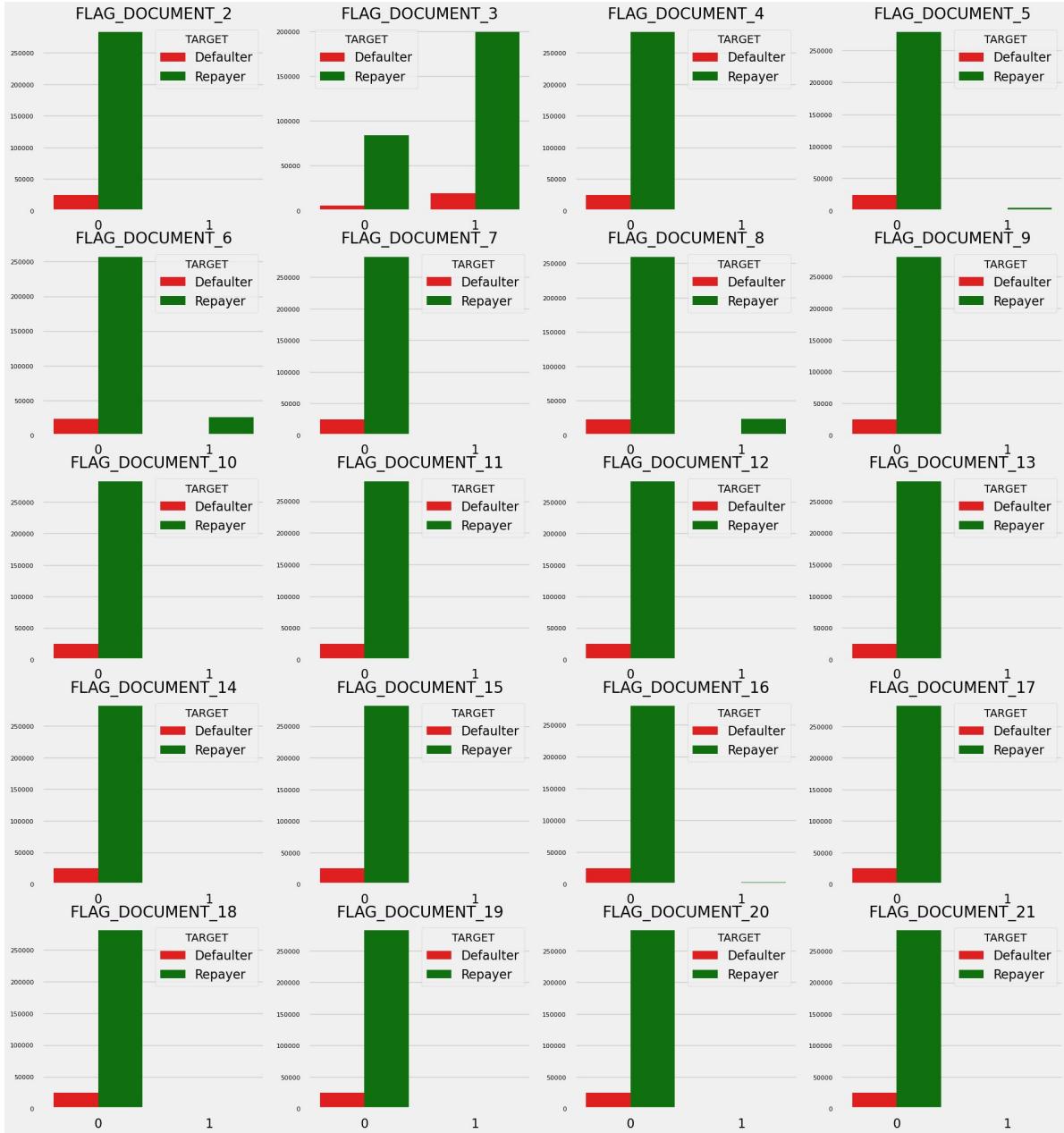
```
Source = applicationDF[["EXT_SOURCE_1", "EXT_SOURCE_2", "EXT_SOURCE_3", "TARGET"]]  
source_corr = Source.corr()  
ax = sns.heatmap(source_corr,  
                  xticklabels=source_corr.columns,  
                  yticklabels=source_corr.columns,  
                  annot = True,  
                  cmap = "RdYlGn")
```



```
In [24]: Unwanted_application = nullcol_40_application["Column Name"].tolist() + ['EXT_S  
len(Unwanted_application)
```

Out[24]: 51

```
In [25]: col_Doc = [ 'FLAG_DOCUMENT_2', 'FLAG_DOCUMENT_3', 'FLAG_DOCUMENT_4', 'FLAG_DOCU  
        'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9', 'FLAG_DOCUMENT_10', 'FLAG_DOCU  
        'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15', 'FLAG_DOCUMENT_16', 'FLAG_DO  
        'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21']  
df_flag = applicationDF[col_Doc+["TARGET"]]  
  
length = len(col_Doc)  
  
df_flag["TARGET"] = df_flag["TARGET"].replace({1:"Defaulter",0:"Repayer"})  
  
fig = plt.figure(figsize=(21,24))  
  
for i,j in itertools.zip_longest(col_Doc,range(length)):  
    plt.subplot(5,4,j+1)  
    ax = sns.countplot(x=df_flag[i],hue=df_flag["TARGET"],palette=["r","g"])  
    plt.yticks(fontsize=8)  
    plt.xlabel("")  
    plt.ylabel("")  
    plt.title(i)
```



```
In [26]: col_Doc.remove('FLAG_DOCUMENT_3')
Unwanted_application = Unwanted_application + col_Doc
len(Unwanted_application)
```

Out[26]: 70

```
In [27]: contact_col =['FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_CONT_MOBIL']
Contact_corr= applicationDF[contact_col].corr()
fig=plt.figure(figsize=(8,8))
ax=sns.heatmap(Contact_corr,
                  xticklabels=Contact_corr.columns,
                  yticklabels=Contact_corr.columns,
                  annot=True,
                  cmap="RdYlGn",
                  linewidth=1)
```



```
In [28]: contact_col.remove('TARGET')
Unwanted_application = Unwanted_application + contact_col
```

```
In [29]: len(Unwanted_application)
```

```
Out[29]: 76
```

```
In [30]: applicationDF.drop(labels=Unwanted_application, axis=1, inplace=True)
```

```
In [31]: applicationDF.shape
```

```
Out[31]: (307511, 46)
```

In [32]: applicationDF.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 46 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  OCCUPATION_TYPE    211120 non-null   object  
 22  CNT_FAM_MEMBERS   307509 non-null   float64 
 23  REGION_RATING_CLIENT 307511 non-null   int64  
 24  REGION_RATING_CLIENT_W_CITY 307511 non-null   int64  
 25  WEEKDAY_APPR_PROCESS_START 307511 non-null   object  
 26  HOUR_APPR_PROCESS_START 307511 non-null   int64  
 27  REG_REGION_NOT_LIVE_REGION 307511 non-null   int64  
 28  REG_REGION_NOT_WORK_REGION 307511 non-null   int64  
 29  LIVE_REGION_NOT_WORK_REGION 307511 non-null   int64  
 30  REG_CITY_NOT_LIVE_CITY 307511 non-null   int64  
 31  REG_CITY_NOT_WORK_CITY 307511 non-null   int64  
 32  LIVE_CITY_NOT_WORK_CITY 307511 non-null   int64  
 33  ORGANIZATION_TYPE   307511 non-null   object  
 34  OBS_30_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 35  DEF_30_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 36  OBS_60_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 37  DEF_60_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 38  DAYS_LAST_PHONE_CHANGE 307510 non-null   float64 
 39  FLAG_DOCUMENT_3     307511 non-null   int64  
 40  AMT_REQ_CREDIT_BUREAU_HOUR 265992 non-null   float64 
 41  AMT_REQ_CREDIT_BUREAU_DAY 265992 non-null   float64 
 42  AMT_REQ_CREDIT_BUREAU_WEEK 265992 non-null   float64 
 43  AMT_REQ_CREDIT_BUREAU_MON 265992 non-null   float64 
 44  AMT_REQ_CREDIT_BUREAU_QRT 265992 non-null   float64 
 45  AMT_REQ_CREDIT_BUREAU_YEAR 265992 non-null   float64 

dtypes: float64(18), int64(16), object(12)
memory usage: 107.9+ MB
```

```
In [33]: Unwanted_previous = nullcol_40_previous["Column Name"].tolist()
Unwanted_previous
```

```
Out[33]: ['AMT_DOWN_PAYMENT',
          'RATE_DOWN_PAYMENT',
          'RATE_INTEREST_PRIMARY',
          'RATE_INTEREST_PRIVILEGED',
          'NAME_TYPE_SUITE',
          'DAYS_FIRST_DRAWING',
          'DAYS_FIRST_DUE',
          'DAYS_LAST_DUE_1ST_VERSION',
          'DAYS_LAST_DUE',
          'DAYS_TERMINATION',
          'NFLAG_INSURED_ON_APPROVAL']
```

```
In [34]: Unnecessary_previous=['WEEKDAY_APPR_PROCESS_START','HOUR_APPR_PROCESS_START',
                                'FLAG_LAST_APPL_PER_CONTRACT',
                                'NFLAG_LAST_APPL_IN_DAY']
```

```
In [35]: Unwanted_previous = Unwanted_previous + Unnecessary_previous
```

```
In [36]: len(Unwanted_previous)
```

```
Out[36]: 15
```

```
In [37]: previousDF.drop(labels=Unwanted_previous, axis=1, inplace=True)
previousDF.shape
```

```
Out[37]: (1670214, 22)
```

```
In [38]: previousDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 22 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_GOODS_PRICE   1284699 non-null  float64 
 7   NAME_CASH_LOAN_PURPOSE  1670214 non-null  object  
 8   NAME_CONTRACT_STATUS  1670214 non-null  object  
 9   DAYS_DECISION     1670214 non-null  int64  
 10  NAME_PAYMENT_TYPE  1670214 non-null  object  
 11  CODE_REJECT_REASON  1670214 non-null  object  
 12  NAME_CLIENT_TYPE   1670214 non-null  object  
 13  NAME_GOODS_CATEGORY  1670214 non-null  object  
 14  NAME_PORTFOLIO     1670214 non-null  object  
 15  NAME_PRODUCT_TYPE   1670214 non-null  object  
 16  CHANNEL_TYPE       1670214 non-null  object  
 17  SELLERPLACE_AREA    1670214 non-null  int64  
 18  NAME_SELLER_INDUSTRY  1670214 non-null  object  
 19  CNT_PAYMENT        1297984 non-null  float64 
 20  NAME_YIELD_GROUP    1670214 non-null  object  
 21  PRODUCT_COMBINATION  1669868 non-null  object  
dtypes: float64(5), int64(4), object(13)
memory usage: 280.3+ MB
```

```
In [39]: date_col=['DAYS_BIRTH', 'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH']
for col in date_col:
    applicationDF[col]=abs(applicationDF[col])
```

```
In [40]: applicationDF['AMT_INCOME_TOTAL']=applicationDF['AMT_INCOME_TOTAL']/100000
bins=[0,1,2,3,4,5,6,7,8,9,10,11]
slot=['0-100K', '100L-200K', '200K-300K', '300K-400K', '400K-500K', '500K-600K', '600K-700K', '700K-800K', '800K-900K', '900K-1M', '1M-1.1M', '1.1M-1.2M', '1.2M-1.3M', '1.3M-1.4M', '1.4M-1.5M', '1.5M-1.6M', '1.6M-1.7M', '1.7M-1.8M', '1.8M-1.9M', '1.9M-2M', '2M-2.1M', '2.1M-2.2M', '2.2M-2.3M', '2.3M-2.4M', '2.4M-2.5M', '2.5M-2.6M', '2.6M-2.7M', '2.7M-2.8M', '2.8M-2.9M', '2.9M-3M', '3M-3.1M', '3.1M-3.2M', '3.2M-3.3M', '3.3M-3.4M', '3.4M-3.5M', '3.5M-3.6M', '3.6M-3.7M', '3.7M-3.8M', '3.8M-3.9M', '3.9M-4M', '4M-4.1M', '4.1M-4.2M', '4.2M-4.3M', '4.3M-4.4M', '4.4M-4.5M', '4.5M-4.6M', '4.6M-4.7M', '4.7M-4.8M', '4.8M-4.9M', '4.9M-5M', '5M-5.1M', '5.1M-5.2M', '5.2M-5.3M', '5.3M-5.4M', '5.4M-5.5M', '5.5M-5.6M', '5.6M-5.7M', '5.7M-5.8M', '5.8M-5.9M', '5.9M-6M', '6M-6.1M', '6.1M-6.2M', '6.2M-6.3M', '6.3M-6.4M', '6.4M-6.5M', '6.5M-6.6M', '6.6M-6.7M', '6.7M-6.8M', '6.8M-6.9M', '6.9M-7M', '7M-7.1M', '7.1M-7.2M', '7.2M-7.3M', '7.3M-7.4M', '7.4M-7.5M', '7.5M-7.6M', '7.6M-7.7M', '7.7M-7.8M', '7.8M-7.9M', '7.9M-8M', '8M-8.1M', '8.1M-8.2M', '8.2M-8.3M', '8.3M-8.4M', '8.4M-8.5M', '8.5M-8.6M', '8.6M-8.7M', '8.7M-8.8M', '8.8M-8.9M', '8.9M-9M', '9M-9.1M', '9.1M-9.2M', '9.2M-9.3M', '9.3M-9.4M', '9.4M-9.5M', '9.5M-9.6M', '9.6M-9.7M', '9.7M-9.8M', '9.8M-9.9M', '9.9M-10M', '10M-10.1M', '10.1M-10.2M', '10.2M-10.3M', '10.3M-10.4M', '10.4M-10.5M', '10.5M-10.6M', '10.6M-10.7M', '10.7M-10.8M', '10.8M-10.9M', '10.9M-11M', '11M-11.1M', '11.1M-11.2M', '11.2M-11.3M', '11.3M-11.4M', '11.4M-11.5M', '11.5M-11.6M', '11.6M-11.7M', '11.7M-11.8M', '11.8M-11.9M', '11.9M-12M', '12M-12.1M', '12.1M-12.2M', '12.2M-12.3M', '12.3M-12.4M', '12.4M-12.5M', '12.5M-12.6M', '12.6M-12.7M', '12.7M-12.8M', '12.8M-12.9M', '12.9M-13M', '13M-13.1M', '13.1M-13.2M', '13.2M-13.3M', '13.3M-13.4M', '13.4M-13.5M', '13.5M-13.6M', '13.6M-13.7M', '13.7M-13.8M', '13.8M-13.9M', '13.9M-14M', '14M-14.1M', '14.1M-14.2M', '14.2M-14.3M', '14.3M-14.4M', '14.4M-14.5M', '14.5M-14.6M', '14.6M-14.7M', '14.7M-14.8M', '14.8M-14.9M', '14.9M-15M', '15M-15.1M', '15.1M-15.2M', '15.2M-15.3M', '15.3M-15.4M', '15.4M-15.5M', '15.5M-15.6M', '15.6M-15.7M', '15.7M-15.8M', '15.8M-15.9M', '15.9M-16M', '16M-16.1M', '16.1M-16.2M', '16.2M-16.3M', '16.3M-16.4M', '16.4M-16.5M', '16.5M-16.6M', '16.6M-16.7M', '16.7M-16.8M', '16.8M-16.9M', '16.9M-17M', '17M-17.1M', '17.1M-17.2M', '17.2M-17.3M', '17.3M-17.4M', '17.4M-17.5M', '17.5M-17.6M', '17.6M-17.7M', '17.7M-17.8M', '17.8M-17.9M', '17.9M-18M', '18M-18.1M', '18.1M-18.2M', '18.2M-18.3M', '18.3M-18.4M', '18.4M-18.5M', '18.5M-18.6M', '18.6M-18.7M', '18.7M-18.8M', '18.8M-18.9M', '18.9M-19M', '19M-19.1M', '19.1M-19.2M', '19.2M-19.3M', '19.3M-19.4M', '19.4M-19.5M', '19.5M-19.6M', '19.6M-19.7M', '19.7M-19.8M', '19.8M-19.9M', '19.9M-20M', '20M-20.1M', '20.1M-20.2M', '20.2M-20.3M', '20.3M-20.4M', '20.4M-20.5M', '20.5M-20.6M', '20.6M-20.7M', '20.7M-20.8M', '20.8M-20.9M', '20.9M-21M', '21M-21.1M', '21.1M-21.2M', '21.2M-21.3M', '21.3M-21.4M', '21.4M-21.5M', '21.5M-21.6M', '21.6M-21.7M', '21.7M-21.8M', '21.8M-21.9M', '21.9M-22M', '22M-22.1M', '22.1M-22.2M', '22.2M-22.3M', '22.3M-22.4M', '22.4M-22.5M', '22.5M-22.6M', '22.6M-22.7M', '22.7M-22.8M', '22.8M-22.9M', '22.9M-23M', '23M-23.1M', '23.1M-23.2M', '23.2M-23.3M', '23.3M-23.4M', '23.4M-23.5M', '23.5M-23.6M', '23.6M-23.7M', '23.7M-23.8M', '23.8M-23.9M', '23.9M-24M', '24M-24.1M', '24.1M-24.2M', '24.2M-24.3M', '24.3M-24.4M', '24.4M-24.5M', '24.5M-24.6M', '24.6M-24.7M', '24.7M-24.8M', '24.8M-24.9M', '24.9M-25M', '25M-25.1M', '25.1M-25.2M', '25.2M-25.3M', '25.3M-25.4M', '25.4M-25.5M', '25.5M-25.6M', '25.6M-25.7M', '25.7M-25.8M', '25.8M-25.9M', '25.9M-26M', '26M-26.1M', '26.1M-26.2M', '26.2M-26.3M', '26.3M-26.4M', '26.4M-26.5M', '26.5M-26.6M', '26.6M-26.7M', '26.7M-26.8M', '26.8M-26.9M', '26.9M-27M', '27M-27.1M', '27.1M-27.2M', '27.2M-27.3M', '27.3M-27.4M', '27.4M-27.5M', '27.5M-27.6M', '27.6M-27.7M', '27.7M-27.8M', '27.8M-27.9M', '27.9M-28M', '28M-28.1M', '28.1M-28.2M', '28.2M-28.3M', '28.3M-28.4M', '28.4M-28.5M', '28.5M-28.6M', '28.6M-28.7M', '28.7M-28.8M', '28.8M-28.9M', '28.9M-29M', '29M-29.1M', '29.1M-29.2M', '29.2M-29.3M', '29.3M-29.4M', '29.4M-29.5M', '29.5M-29.6M', '29.6M-29.7M', '29.7M-29.8M', '29.8M-29.9M', '29.9M-30M', '30M-30.1M', '30.1M-30.2M', '30.2M-30.3M', '30.3M-30.4M', '30.4M-30.5M', '30.5M-30.6M', '30.6M-30.7M', '30.7M-30.8M', '30.8M-30.9M', '30.9M-31M', '31M-31.1M', '31.1M-31.2M', '31.2M-31.3M', '31.3M-31.4M', '31.4M-31.5M', '31.5M-31.6M', '31.6M-31.7M', '31.7M-31.8M', '31.8M-31.9M', '31.9M-32M', '32M-32.1M', '32.1M-32.2M', '32.2M-32.3M', '32.3M-32.4M', '32.4M-32.5M', '32.5M-32.6M', '32.6M-32.7M', '32.7M-32.8M', '32.8M-32.9M', '32.9M-33M', '33M-33.1M', '33.1M-33.2M', '33.2M-33.3M', '33.3M-33.4M', '33.4M-33.5M', '33.5M-33.6M', '33.6M-33.7M', '33.7M-33.8M', '33.8M-33.9M', '33.9M-34M', '34M-34.1M', '34.1M-34.2M', '34.2M-34.3M', '34.3M-34.4M', '34.4M-34.5M', '34.5M-34.6M', '34.6M-34.7M', '34.7M-34.8M', '34.8M-34.9M', '34.9M-35M', '35M-35.1M', '35.1M-35.2M', '35.2M-35.3M', '35.3M-35.4M', '35.4M-35.5M', '35.5M-35.6M', '35.6M-35.7M', '35.7M-35.8M', '35.8M-35.9M', '35.9M-36M', '36M-36.1M', '36.1M-36.2M', '36.2M-36.3M', '36.3M-36.4M', '36.4M-36.5M', '36.5M-36.6M', '36.6M-36.7M', '36.7M-36.8M', '36.8M-36.9M', '36.9M-37M', '37M-37.1M', '37.1M-37.2M', '37.2M-37.3M', '37.3M-37.4M', '37.4M-37.5M', '37.5M-37.6M', '37.6M-37.7M', '37.7M-37.8M', '37.8M-37.9M', '37.9M-38M', '38M-38.1M', '38.1M-38.2M', '38.2M-38.3M', '38.3M-38.4M', '38.4M-38.5M', '38.5M-38.6M', '38.6M-38.7M', '38.7M-38.8M', '38.8M-38.9M', '38.9M-39M', '39M-39.1M', '39.1M-39.2M', '39.2M-39.3M', '39.3M-39.4M', '39.4M-39.5M', '39.5M-39.6M', '39.6M-39.7M', '39.7M-39.8M', '39.8M-39.9M', '39.9M-40M', '40M-40.1M', '40.1M-40.2M', '40.2M-40.3M', '40.3M-40.4M', '40.4M-40.5M', '40.5M-40.6M', '40.6M-40.7M', '40.7M-40.8M', '40.8M-40.9M', '40.9M-41M', '41M-41.1M', '41.1M-41.2M', '41.2M-41.3M', '41.3M-41.4M', '41.4M-41.5M', '41.5M-41.6M', '41.6M-41.7M', '41.7M-41.8M', '41.8M-41.9M', '41.9M-42M', '42M-42.1M', '42.1M-42.2M', '42.2M-42.3M', '42.3M-42.4M', '42.4M-42.5M', '42.5M-42.6M', '42.6M-42.7M', '42.7M-42.8M', '42.8M-42.9M', '42.9M-43M', '43M-43.1M', '43.1M-43.2M', '43.2M-43.3M', '43.3M-43.4M', '43.4M-43.5M', '43.5M-43.6M', '43.6M-43.7M', '43.7M-43.8M', '43.8M-43.9M', '43.9M-44M', '44M-44.1M', '44.1M-44.2M', '44.2M-44.3M', '44.3M-44.4M', '44.4M-44.5M', '44.5M-44.6M', '44.6M-44.7M', '44.7M-44.8M', '44.8M-44.9M', '44.9M-45M', '45M-45.1M', '45.1M-45.2M', '45.2M-45.3M', '45.3M-45.4M', '45.4M-45.5M', '45.5M-45.6M', '45.6M-45.7M', '45.7M-45.8M', '45.8M-45.9M', '45.9M-46M', '46M-46.1M', '46.1M-46.2M', '46.2M-46.3M', '46.3M-46.4M', '46.4M-46.5M', '46.5M-46.6M', '46.6M-46.7M', '46.7M-46.8M', '46.8M-46.9M', '46.9M-47M', '47M-47.1M', '47.1M-47.2M', '47.2M-47.3M', '47.3M-47.4M', '47.4M-47.5M', '47.5M-47.6M', '47.6M-47.7M', '47.7M-47.8M', '47.8M-47.9M', '47.9M-48M', '48M-48.1M', '48.1M-48.2M', '48.2M-48.3M', '48.3M-48.4M', '48.4M-48.5M', '48.5M-48.6M', '48.6M-48.7M', '48.7M-48.8M', '48.8M-48.9M', '48.9M-49M', '49M-49.1M', '49.1M-49.2M', '49.2M-49.3M', '49.3M-49.4M', '49.4M-49.5M', '49.5M-49.6M', '49.6M-49.7M', '49.7M-49.8M', '49.8M-49.9M', '49.9M-50M', '50M-50.1M', '50.1M-50.2M', '50.2M-50.3M', '50.3M-50.4M', '50.4M-50.5M', '50.5M-50.6M', '50.6M-50.7M', '50.7M-50.8M', '50.8M-50.9M', '50.9M-51M', '51M-51.1M', '51.1M-51.2M', '51.2M-51.3M', '51.3M-51.4M', '51.4M-51.5M', '51.5M-51.6M', '51.6M-51.7M', '51.7M-51.8M', '51.8M-51.9M', '51.9M-52M', '52M-52.1M', '52.1M-52.2M', '52.2M-52.3M', '52.3M-52.4M', '52.4M-52.5M', '52.5M-52.6M', '52.6M-52.7M', '52.7M-52.8M', '52.8M-52.9M', '52.9M-53M', '53M-53.1M', '53.1M-53.2M', '53.2M-53.3M', '53.3M-53.4M', '53.4M-53.5M', '53.5M-53.6M', '53.6M-53.7M', '53.7M-53.8M', '53.8M-53.9M', '53.9M-54M', '54M-54.1M', '54.1M-54.2M', '54.2M-54.3M', '54.3M-54.4M', '54.4M-54.5M', '54.5M-54.6M', '54.6M-54.7M', '54.7M-54.8M', '54.8M-54.9M', '54.9M-55M', '55M-55.1M', '55.1M-55.2M', '55.2M-55.3M', '55.3M-55.4M', '55.4M-55.5M', '55.5M-55.6M', '55.6M-55.7M', '55.7M-55.8M', '55.8M-55.9M', '55.9M-56M', '56M-56.1M', '56.1M-56.2M', '56.2M-56.3M', '56.3M-56.4M', '56.4M-56.5M', '56.5M-56.6M', '56.6M-56.7M', '56.7M-56.8M', '56.8M-56.9M', '56.9M-57M', '57M-57.1M', '57.1M-57.2M', '57.2M-57.3M', '57.3M-57.4M', '57.4M-57.5M', '57.5M-57.6M', '57.6M-57.7M', '57.7M-57.8M', '57.8M-57.9M', '57.9M-58M', '58M-58.1M', '58.1M-58.2M', '58.2M-58.3M', '58.3M-58.4M', '58.4M-58.5M', '58.5M-58.6M', '58.6M-58.7M', '58.7M-58.8M', '58.8M-58.9M', '58.9M-59M', '59M-59.1M', '59.1M-59.2M', '59.2M-59.3M', '59.3M-59.4M', '59.4M-59.5M', '59.5M-59.6M', '59.6M-59.7M', '59.7M-59.8M', '59.8M-59.9M', '59.9M-60M', '60M-60.1M', '60.1M-60.2M', '60.2M-60.3M', '60.3M-60.4M', '60.4M-60.5M', '60.5M-60.6M', '60.6M-60.7M', '60.7M-60.8M', '60.8M-60.9M', '60.9M-61M', '61M-61.1M', '61.1M-61.2M', '61.2M-61.3M', '61.3M-61.4M', '61.4M-61.5M', '61.5M-61.6M', '61.6M-61.7M', '61.7M-61.8M', '61.8M-61.9M', '61.9M-62M', '62M-62.1M', '62.1M-62.2M', '62.2M-62.3M', '62.3M-62.4M', '62.4M-62.5M', '62.5M-62.6M', '62.6M-62.7M', '62.7M-62.8M', '62.8M-62.9M', '62.9M-63M', '63M-63.1M', '63.1M-63.2M', '63.2M-63.3M', '63.3M-63.4M', '63.4M-63.5M', '63.5M-63.6M', '63.6M-63.7M', '63.7M-63.8M', '63.8M-63.9M', '63.9M-64M', '64M-64.1M', '64.1M-64.2M', '64.2M-64.3M', '64.3M-64.4M', '64.4M-64.5M', '64.5M-64.6M', '64.6M-64.7M', '64.7M-64.8M', '64.8M-64.9M', '64.9M-65M', '65M-65.1M', '65.1M-65.2M', '65.2M-65.3M', '65.3M-65.4M', '65.4M-65.5M', '65.5M-65.6M', '65.6M-65.7M', '65.7M-65.8M', '65.8M-65.9M', '65.9M-66M', '66M-66.1M', '66.1M-66.2M', '66.2M-66.3M', '66.3M-66.4M', '66.4M-66.5M', '66.5M-66.6M', '66.6M-66.7M', '66.7M-66.8M', '66.8M-66.9M', '66.9M-67M', '67M-67.1M', '67.1M-67.2M', '67.2M-67.3M', '67.3M-67.4M', '67.4M-67.5M', '67.5M-67.6M', '67.6M-67.7M', '67.7M-67.8M', '67.8M-67.9M', '67.9M-68M', '68M-68.1M', '68.1M-68.2M', '68.2M-68.3M', '68.3M-68.4M', '68.4M-68.5M', '68.5M-68.6M', '68.6M-68.7M', '68.7M-68.8M', '68.8M-68.9M', '68.9M-69M', '69M-69.1M', '69.1M-69.2M', '69.2M-69.3M', '69.3M-69.4M', '69.4M-69.5M', '69.5M-69.6M', '69.6M-69.7M', '69.7M-69.8M', '69.8M-69.9M', '69.9M-70M', '70M-70.1M', '70.1M-70.2M', '70.2M-70.3M', '70.3M-70.4M', '70.4M-70.5M', '70.5M-70.6M', '70.6M-70.7M', '70.7M-70.8M', '70.8M-70.9M', '70.9M-71M', '71M-71.1M', '71.1M-71.2M', '71.2M-71.3M', '71.3M-71.4M', '71.4M-71.5M', '71.5M-71.6M', '71.6M-71.7M', '71.7M-71.8M', '71.8M-71.9M', '71.9M-72M', '72M-72.1M', '72.1M-72.2M', '72.2M-72.3M', '72.3M-72.4M', '72.4M-72.5M', '72.5M-72.6M', '72.6M-72.7M', '72.7M-72.8M', '72.8M-72.9M', '72.9M-73M', '73M-73.1M', '73.1M-73.2M', '73.2M-73.3M', '73.3M-73.4M', '73.4M-73.5M', '73.5M-73.6M', '73.6M-73.7M', '73.7M-73.8M', '73.8M-73.9M', '73.9M-74M', '74M-74.1M', '74.1M-74.2M', '74.2M-74.3M', '74.3M-74.4M', '74.4M-74.5M', '74.5M-74.6M', '74.6M-74.7M', '74.7M-74.8M', '74.8M-74.9M', '74.9M-75M', '75M-75.1M', '75.1M-75.2M', '75.2M-75.3M', '75.3M-75.4M', '75.4M-75.5M', '75.5M-75.6M', '75.6M-75.7M', '75.7M-75.8M', '75.8M-75.9M', '75.9M-76M', '76M-76.1M', '76.1M-76.2M', '76.2M-76.3M', '76.3M-76.4M', '76.4M-76.5M', '76.5M-76.6M', '76.6M-76.7M', '76.7M-76.8M', '76.8M-76.9M', '76.9M-77M', '77M-77.1M', '77.1M-77.2M', '77.2M-77.3M', '77.3M-77.4M', '77.4M-77.5M', '77.5M-77.6M', '77.6M-77.7M', '77.7M-77.8M', '77.8M-77.9M', '77.9M-78M', '78M-78.1M', '78.1M-78.2M', '78.2M-78.3M', '78.3M-78.4M', '78.4M-78.5M', '78.5M-78.6M', '78.6M-78.7M', '78.7M-78.8M', '78.8M-78.9M', '78.9M-79M', '79M-79.1M', '79.1M-79.2M', '79.2M-79.3M', '79.3M-79.4M', '79.4M-79.5M', '79.5M-79.6M', '79.6M-79.7M', '79.7M-79.8M', '79.8M-79.9M', '79.9M-80M', '80M-80.1M', '80.1M-80.2M', '80.2M-80.3M', '80.3M-80.4M', '80.4M-80.5M', '80.5M-80.6M', '80.6M-80.7M', '80.7M-80.8M', '80.8M-80.9M', '80.9M-81M', '81M-81.1M', '81.1M-81.2M', '81.2M-81.3M', '81.3M-81.4M', '81.4M-81.5M', '81.5M-81.6M', '81.6M-81.7M', '81.7M-81.8M', '81.8M-81.9M', '81.9M-82M', '82M-82.1M', '82.1M-82.2M', '82.2M-82.3M', '82.3M-82.4M', '82.4M-82.5M', '82.5M-82.6M', '82.6M-82.7M', '82.7M-82.8M', '82.8M-82.9M', '82.9M-83M', '83M-83.1M', '83.1M-83.2M', '83.2M-83.3M', '83.3M-83.4M', '83.4M-83.5M', '83.5M-83.6M', '83.6M-83.7M', '83.7M-83.8M', '83.8M-83.9M', '83.9M-84M', '84M-84.1M', '84.1M-84.2M', '84.2M-84.3M', '84.3M-84.4M', '84.4M-84.5M', '84.5M-84.6M', '84.6M-84.7M', '84.
```

In [41]:

```
applicationDF['AMT_INCOME_RANGE'].value_counts(normalize=True)*100
```

Out[41]:

100L-200K	50.735000
200K-300K	21.210691
0-100K	20.729695
300K-400K	4.776116
400K-500K	1.744669
500K-600K	0.356354
600K-700K	0.282805
800K-900K	0.096980
700K-800K	0.052721
900K-1M	0.009112
1Above	0.005858

Name: AMT\_INCOME\_RANGE, dtype: float64

In [42]:

```
applicationDF['AMT_CREDIT']=applicationDF['AMT_CREDIT']/100000  
  
bins = [0,1,2,3,4,5,6,7,8,9,10,100]  
slots = ['0-100K', '100K-200K', '200K-300K', '300K-400K', '400K-500K', '500K-600K'  
         '800K-900K', '900K-1M', '1M Above']  
  
applicationDF['AMT_CREDIT_RANGE']=pd.cut(applicationDF['AMT_CREDIT'], bins=bins)
```

In [43]:

```
applicationDF['AMT_CREDIT_RANGE'].value_counts(normalize=True)*100
```

Out[43]:

200K-300K	17.824728
1M Above	16.254703
500K-600K	11.131960
400K-500K	10.418489
100K-200K	9.801275
300K-400K	8.564897
600K-700K	7.820533
800K-900K	7.086576
700K-800K	6.241403
900K-1M	2.902986
0-100K	1.952450

Name: AMT\_CREDIT\_RANGE, dtype: float64

In [44]:

```
applicationDF['AGE'] = applicationDF['DAYS_BIRTH'] // 365  
bins = [0,20,30,40,50,100]  
slots = ['0-20', '20-30', '30-40', '40-50', '50 above']  
  
applicationDF['AGE_GROUP']=pd.cut(applicationDF['AGE'], bins=bins, labels=slots)
```

```
In [45]: applicationDF['AGE_GROUP'].value_counts(normalize=True)*100
```

```
Out[45]: 50 above    31.604398  
30-40      27.028952  
40-50      24.194582  
20-30      17.171743  
0-20       0.000325  
Name: AGE_GROUP, dtype: float64
```

```
In [46]: applicationDF['YEARS_EMPLOYED'] = applicationDF['DAYS_EMPLOYED'] // 365  
bins = [0, 5, 10, 20, 30, 40, 50, 60, 150]  
slots = ['0-5', '5-10', '10-20', '20-30', '30-40', '40-50', '50-60', '60 above']  
  
applicationDF['EMPLOYMENT_YEAR']=pd.cut(applicationDF['YEARS_EMPLOYED'],bins=b
```

```
In [47]: applicationDF['EMPLOYMENT_YEAR'].value_counts(normalize=True)*100
```

```
Out[47]: 0-5        55.582363  
5-10       24.966441  
10-20      14.564315  
20-30       3.750117  
30-40       1.058720  
40-50       0.078044  
50-60       0.000000  
60 above    0.000000  
Name: EMPLOYMENT_YEAR, dtype: float64
```

```
In [48]: applicationDF.nunique().sort_values()
```

```
Out[48]:
```

LIVE_CITY_NOT_WORK_CITY	2
TARGET	2
NAME_CONTRACT_TYPE	2
REG_REGION_NOT_LIVE_REGION	2
FLAG_OWN_CAR	2
FLAG_OWN_REALTY	2
REG_REGION_NOT_WORK_REGION	2
LIVE_REGION_NOT_WORK_REGION	2
FLAG_DOCUMENT_3	2
REG_CITY_NOT_LIVE_CITY	2
REG_CITY_NOT_WORK_CITY	2
REGION_RATING_CLIENT	3
CODE_GENDER	3
REGION_RATING_CLIENT_W_CITY	3
AMT_REQ_CREDIT_BUREAU_HOUR	5
NAME_EDUCATION_TYPE	5
AGE_GROUP	5
NAME_FAMILY_STATUS	6
NAME_HOUSING_TYPE	6
EMPLOYMENT_YEAR	6
WEEKDAY_APPR_PROCESS_START	7
NAME_TYPE_SUITE	7
NAME_INCOME_TYPE	8
AMT_REQ_CREDIT_BUREAU_WEEK	9
AMT_REQ_CREDIT_BUREAU_DAY	9
DEF_60_CNT_SOCIAL_CIRCLE	9
DEF_30_CNT_SOCIAL_CIRCLE	10
AMT_CREDIT_RANGE	11
AMT_INCOME_RANGE	11
AMT_REQ_CREDIT_BUREAU_QRT	11
CNT_CHILDREN	15
CNT_FAM_MEMBERS	17
OCCUPATION_TYPE	18
HOUR_APPR_PROCESS_START	24
AMT_REQ_CREDIT_BUREAU_MON	24
AMT_REQ_CREDIT_BUREAU_YEAR	25
OBS_60_CNT_SOCIAL_CIRCLE	33
OBS_30_CNT_SOCIAL_CIRCLE	33
AGE	50
YEARS_EMPLOYED	51
ORGANIZATION_TYPE	58
REGION_POPULATION_RELATIVE	81
AMT_GOODS_PRICE	1002
AMT_INCOME_TOTAL	2548
DAYS_LAST_PHONE_CHANGE	3773
AMT_CREDIT	5603
DAYS_ID_PUBLISH	6168
DAYS_EMPLOYED	12574
AMT_ANNUITY	13672
DAYS_REGISTRATION	15688
DAYS_BIRTH	17460
SK_ID_CURR	307511

dtype: int64

In [49]: applicationDF.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 52 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  OCCUPATION_TYPE    211120 non-null   object  
 22  CNT_FAM_MEMBERS   307509 non-null   float64 
 23  REGION_RATING_CLIENT 307511 non-null   int64  
 24  REGION_RATING_CLIENT_W_CITY 307511 non-null   int64  
 25  WEEKDAY_APPR_PROCESS_START 307511 non-null   object  
 26  HOUR_APPR_PROCESS_START 307511 non-null   int64  
 27  REG_REGION_NOT_LIVE_REGION 307511 non-null   int64  
 28  REG_REGION_NOT_WORK_REGION 307511 non-null   int64  
 29  LIVE_REGION_NOT_WORK_REGION 307511 non-null   int64  
 30  REG_CITY_NOT_LIVE_CITY 307511 non-null   int64  
 31  REG_CITY_NOT_WORK_CITY 307511 non-null   int64  
 32  LIVE_CITY_NOT_WORK_CITY 307511 non-null   int64  
 33  ORGANIZATION_TYPE   307511 non-null   object  
 34  OBS_30_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 35  DEF_30_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 36  OBS_60_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 37  DEF_60_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 38  DAYS_LAST_PHONE_CHANGE 307510 non-null   float64 
 39  FLAG_DOCUMENT_3     307511 non-null   int64  
 40  AMT_REQ_CREDIT_BUREAU_HOUR 265992 non-null   float64 
 41  AMT_REQ_CREDIT_BUREAU_DAY 265992 non-null   float64 
 42  AMT_REQ_CREDIT_BUREAU_WEEK 265992 non-null   float64 
 43  AMT_REQ_CREDIT_BUREAU_MON 265992 non-null   float64 
 44  AMT_REQ_CREDIT_BUREAU_QRT 265992 non-null   float64 
 45  AMT_REQ_CREDIT_BUREAU_YEAR 265992 non-null   float64 
 46  AMT_INCOME_RANGE     307279 non-null   category 
 47  AMT_CREDIT_RANGE      307511 non-null   category 
 48  AGE                  307511 non-null   int64
```

```
49  AGE_GROUP           307511 non-null  category
50  YEARS_EMPLOYED      307511 non-null  int64
51  EMPLOYMENT_YEAR     224233 non-null  category
dtypes: category(4), float64(18), int64(18), object(12)
memory usage: 113.8+ MB
```

```
In [50]: categorical_columns = ['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'NAME_TYPE_SUITE', 'N
'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'OCCUPATION_TY
'ORGANIZATION_TYPE', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'L
'REG_CITY_NOT_LIVE_CITY', 'REG_CITY_NOT_WORK_CITY', 'REG_
'LIVE_REGION_NOT_WORK_REGION', 'REGION_RATING_CLIENT', 'W
'REGION_RATING_CLIENT_W_CITY'
]
for col in categorical_columns:
    applicationDF[col] = pd.Categorical(applicationDF[col])
```

In [51]: applicationDF.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 52 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   category
 3   CODE_GENDER      307511 non-null   category
 4   FLAG_OWN_CAR     307511 non-null   category
 5   FLAG_OWN_REALTY  307511 non-null   category
 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   category
 12  NAME_INCOME_TYPE 307511 non-null   category
 13  NAME_EDUCATION_TYPE 307511 non-null   category
 14  NAME_FAMILY_STATUS 307511 non-null   category
 15  NAME_HOUSING_TYPE 307511 non-null   category
 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  OCCUPATION_TYPE   211120 non-null   category
 22  CNT_FAM_MEMBERS   307509 non-null   float64 
 23  REGION_RATING_CLIENT 307511 non-null   category
 24  REGION_RATING_CLIENT_W_CITY 307511 non-null   category
 25  WEEKDAY_APPR_PROCESS_START 307511 non-null   category
 26  HOUR_APPR_PROCESS_START 307511 non-null   int64  
 27  REG_REGION_NOT_LIVE_REGION 307511 non-null   int64  
 28  REG_REGION_NOT_WORK_REGION 307511 non-null   category
 29  LIVE_REGION_NOT_WORK_REGION 307511 non-null   category
 30  REG_CITY_NOT_LIVE_CITY 307511 non-null   category
 31  REG_CITY_NOT_WORK_CITY 307511 non-null   category
 32  LIVE_CITY_NOT_WORK_CITY 307511 non-null   category
 33  ORGANIZATION_TYPE   307511 non-null   category
 34  OBS_30_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 35  DEF_30_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 36  OBS_60_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 37  DEF_60_CNT_SOCIAL_CIRCLE 306490 non-null   float64 
 38  DAYS_LAST_PHONE_CHANGE 307510 non-null   float64 
 39  FLAG_DOCUMENT_3     307511 non-null   int64  
 40  AMT_REQ_CREDIT_BUREAU_HOUR 265992 non-null   float64 
 41  AMT_REQ_CREDIT_BUREAU_DAY 265992 non-null   float64 
 42  AMT_REQ_CREDIT_BUREAU_WEEK 265992 non-null   float64 
 43  AMT_REQ_CREDIT_BUREAU_MON 265992 non-null   float64 
 44  AMT_REQ_CREDIT_BUREAU_QRT 265992 non-null   float64 
 45  AMT_REQ_CREDIT_BUREAU_YEAR 265992 non-null   float64 
 46  AMT_INCOME_RANGE    307279 non-null   category
 47  AMT_CREDIT_RANGE    307511 non-null   category
 48  AGE                307511 non-null   int64 
```

```
49  AGE_GROUP           307511 non-null  category
50  YEARS_EMPLOYED      307511 non-null  int64
51  EMPLOYMENT_YEAR     224233 non-null  category
dtypes: category(23), float64(18), int64(11)
memory usage: 74.8 MB
```

```
In [52]: previousDF.nunique().sort_values()
```

```
Out[52]: NAME_PRODUCT_TYPE          3
NAME_PAYMENT_TYPE          4
NAME_CONTRACT_TYPE          4
NAME_CLIENT_TYPE          4
NAME_CONTRACT_STATUS         4
NAME_PORTFOLIO          5
NAME_YIELD_GROUP          5
CHANNEL_TYPE          8
CODE_REJECT_REASON         9
NAME_SELLER_INDUSTRY        11
PRODUCT_COMBINATION        17
NAME_CASH_LOAN_PURPOSE       25
NAME_GOODS_CATEGORY        28
CNT_PAYMENT          49
SELLERPLACE_AREA          2097
DAYS_DECISION          2922
AMT_CREDIT          86803
AMT_GOODS_PRICE          93885
AMT_APPLICATION          93885
SK_ID_CURR          338857
AMT_ANNUITY          357959
SK_ID_PREV          1670214
dtype: int64
```

```
In [53]: previousDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 22 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_GOODS_PRICE   1284699 non-null  float64 
 7   NAME_CASH_LOAN_PURPOSE 1670214 non-null  object  
 8   NAME_CONTRACT_STATUS 1670214 non-null  object  
 9   DAYS_DECISION    1670214 non-null  int64  
 10  NAME_PAYMENT_TYPE 1670214 non-null  object  
 11  CODE_REJECT_REASON 1670214 non-null  object  
 12  NAME_CLIENT_TYPE  1670214 non-null  object  
 13  NAME_GOODS_CATEGORY 1670214 non-null  object  
 14  NAME_PORTFOLIO    1670214 non-null  object  
 15  NAME_PRODUCT_TYPE 1670214 non-null  object  
 16  CHANNEL_TYPE     1670214 non-null  object  
 17  SELLERPLACE_AREA  1670214 non-null  int64  
 18  NAME_SELLER_INDUSTRY 1670214 non-null  object  
 19  CNT_PAYMENT      1297984 non-null  float64 
 20  NAME_YIELD_GROUP 1670214 non-null  object  
 21  PRODUCT_COMBINATION 1669868 non-null  object  
dtypes: float64(5), int64(4), object(13)
memory usage: 280.3+ MB
```

```
In [54]: previousDF['DAYS_DECISION'] = abs(previousDF['DAYS_DECISION'])
```

```
In [55]: previousDF['DAYS_DECISION_GROUP'] = (previousDF['DAYS_DECISION']-(previousDF['
```

```
In [56]: previousDF['DAYS_DECISION_GROUP'].value_counts(normalize=True)*100
```

```
Out[56]: 0-400          37.490525
400-800         22.944724
800-1200        12.444753
1200-1600        7.904556
2400-2800        6.297456
1600-2000        5.795784
2000-2400        5.684960
2800-3200        1.437241
Name: DAYS_DECISION_GROUP, dtype: float64
```

```
In [57]: Categorical_col_p = ['NAME_CASH_LOAN_PURPOSE', 'NAME_CONTRACT_STATUS', 'NAME_PAYM  
'CODE_REJECT_REASON', 'NAME_CLIENT_TYPE', 'NAME_GOODS_CATEGO  
'NAME_PRODUCT_TYPE', 'CHANNEL_TYPE', 'NAME_SELLER_INDUSTRY', '  
'NAME_CONTRACT_TYPE', 'DAYS_DECISION_GROUP']  
  
for col in Categorical_col_p:  
    previousDF[col] = pd.Categorical(previousDF[col])
```

```
In [58]: previousDF.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1670214 entries, 0 to 1670213  
Data columns (total 23 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  category  
 3   AMT_ANNUITY     1297979 non-null  float64  
 4   AMT_APPLICATION  1670214 non-null  float64  
 5   AMT_CREDIT       1670213 non-null  float64  
 6   AMT_GOODS_PRICE  1284699 non-null  float64  
 7   NAME_CASH_LOAN_PURPOSE  1670214 non-null  category  
 8   NAME_CONTRACT_STATUS  1670214 non-null  category  
 9   DAYS_DECISION    1670214 non-null  int64    
 10  NAME_PAYMENT_TYPE  1670214 non-null  category  
 11  CODE_REJECT_REASON  1670214 non-null  category  
 12  NAME_CLIENT_TYPE  1670214 non-null  category  
 13  NAME_GOODS_CATEGORY  1670214 non-null  category  
 14  NAME_PORTFOLIO    1670214 non-null  category  
 15  NAME_PRODUCT_TYPE  1670214 non-null  category  
 16  CHANNEL_TYPE      1670214 non-null  category  
 17  SELLERPLACE_AREA  1670214 non-null  int64    
 18  NAME_SELLER_INDUSTRY  1670214 non-null  category  
 19  CNT_PAYMENT      1297984 non-null  float64  
 20  NAME_YIELD_GROUP  1670214 non-null  category  
 21  PRODUCT_COMBINATION  1669868 non-null  category  
 22  DAYS_DECISION_GROUP  1670214 non-null  category  
dtypes: category(14), float64(5), int64(4)  
memory usage: 137.0 MB
```

```
In [59]: round(applicationDF.isnull().sum()/applicationDF.shape[0]*100.00,2)
```

```
Out[59]: SK_ID_CURR           0.00  
TARGET              0.00  
NAME_CONTRACT_TYPE 0.00  
CODE_GENDER          0.00  
FLAG_OWN_CAR         0.00  
FLAG_OWN_REALTY     0.00  
CNT_CHILDREN         0.00  
AMT_INCOME_TOTAL    0.00  
AMT_CREDIT            0.00  
AMT_ANNUITY           0.00  
AMT_GOODS_PRICE       0.09  
NAME_TYPE_SUITE       0.42  
NAME_INCOME_TYPE      0.00  
NAME_EDUCATION_TYPE   0.00  
NAME_FAMILY_STATUS     0.00  
NAME_HOUSING_TYPE     0.00  
REGION_POPULATION_RELATIVE 0.00  
DAYS_BIRTH             0.00  
DAYS_EMPLOYED          0.00  
DAYS_REGISTRATION      0.00  
DAYS_ID_PUBLISH        0.00  
OCCUPATION_TYPE        31.35  
CNT_FAM_MEMBERS         0.00  
REGION_RATING_CLIENT    0.00  
REGION_RATING_CLIENT_W_CITY 0.00  
WEEKDAY_APPR_PROCESS_START 0.00  
HOUR_APPR_PROCESS_START 0.00  
REG_REGION_NOT_LIVE_REGION 0.00  
REG_REGION_NOT_WORK_REGION 0.00  
LIVE_REGION_NOT_WORK_REGION 0.00  
REG_CITY_NOT_LIVE_CITY   0.00  
REG_CITY_NOT_WORK_CITY   0.00  
LIVE_CITY_NOT_WORK_CITY   0.00  
ORGANIZATION_TYPE        0.00  
OBS_30_CNT_SOCIAL_CIRCLE 0.33  
DEF_30_CNT_SOCIAL_CIRCLE 0.33  
OBS_60_CNT_SOCIAL_CIRCLE 0.33  
DEF_60_CNT_SOCIAL_CIRCLE 0.33  
DAYS_LAST_PHONE_CHANGE   0.00  
FLAG_DOCUMENT_3           0.00  
AMT_REQ_CREDIT_BUREAU_HOUR 13.50  
AMT_REQ_CREDIT_BUREAU_DAY 13.50  
AMT_REQ_CREDIT_BUREAU_WEEK 13.50  
AMT_REQ_CREDIT_BUREAU_MON 13.50  
AMT_REQ_CREDIT_BUREAU_QRT 13.50  
AMT_REQ_CREDIT_BUREAU_YEAR 13.50  
AMT_INCOME_RANGE          0.08  
AMT_CREDIT_RANGE           0.00  
AGE                      0.00  
AGE_GROUP                 0.00  
YEARS_EMPLOYED             0.00  
EMPLOYMENT_YEAR            27.08  
dtype: float64
```

```
In [60]: applicationDF['NAME_TYPE_SUITE'].describe()
```

```
Out[60]: count      306219
unique       7
top    Unaccompanied
freq      248526
Name: NAME_TYPE_SUITE, dtype: object
```

```
In [61]: applicationDF['NAME_TYPE_SUITE'].fillna((applicationDF['NAME_TYPE_SUITE'].mode
```

```
In [62]: applicationDF['OCCUPATION_TYPE'] = applicationDF['OCCUPATION_TYPE'].cat.add_categories(['Unknown'])
applicationDF['OCCUPATION_TYPE'].fillna('Unknown', inplace =True)
```

```
In [63]: applicationDF[['AMT_REQ_CREDIT_BUREAU_HOUR', 'AMT_REQ_CREDIT_BUREAU_DAY',
                           'AMT_REQ_CREDIT_BUREAU_WEEK', 'AMT_REQ_CREDIT_BUREAU_MON',
                           'AMT_REQ_CREDIT_BUREAU_QRT', 'AMT_REQ_CREDIT_BUREAU_YEAR']].desc
```

```
Out[63]:
```

	AMT_REQ_CREDIT_BUREAU_HOUR	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_CREDIT_BUREAU_WEEK	AMT_REQ_CREDIT_BUREAU_MON	AMT_REQ_CREDIT_BUREAU_QRT	AMT_REQ_CREDIT_BUREAU_YEAR
--	----------------------------	---------------------------	----------------------------	---------------------------	---------------------------	----------------------------

count	265992.000000	265992.000000	265992.000000	265992.000000	265992.000000	265992.000000
mean	0.006402	0.007000	0.007000	0.007000	0.007000	0.007000
std	0.083849	0.110757	0.110757	0.110757	0.110757	0.110757
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	4.000000	9.000000	9.000000	9.000000	9.000000	9.000000

```
In [64]: amount = ['AMT_REQ_CREDIT_BUREAU_HOUR', 'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_WEEK',
                  'AMT_REQ_CREDIT_BUREAU_MON', 'AMT_REQ_CREDIT_BUREAU_QRT', 'AMT_REQ_CREDIT_BUREAU_YEAR']
```

```
for col in amount:
    applicationDF[col].fillna(applicationDF[col].median(), inplace = True)
```

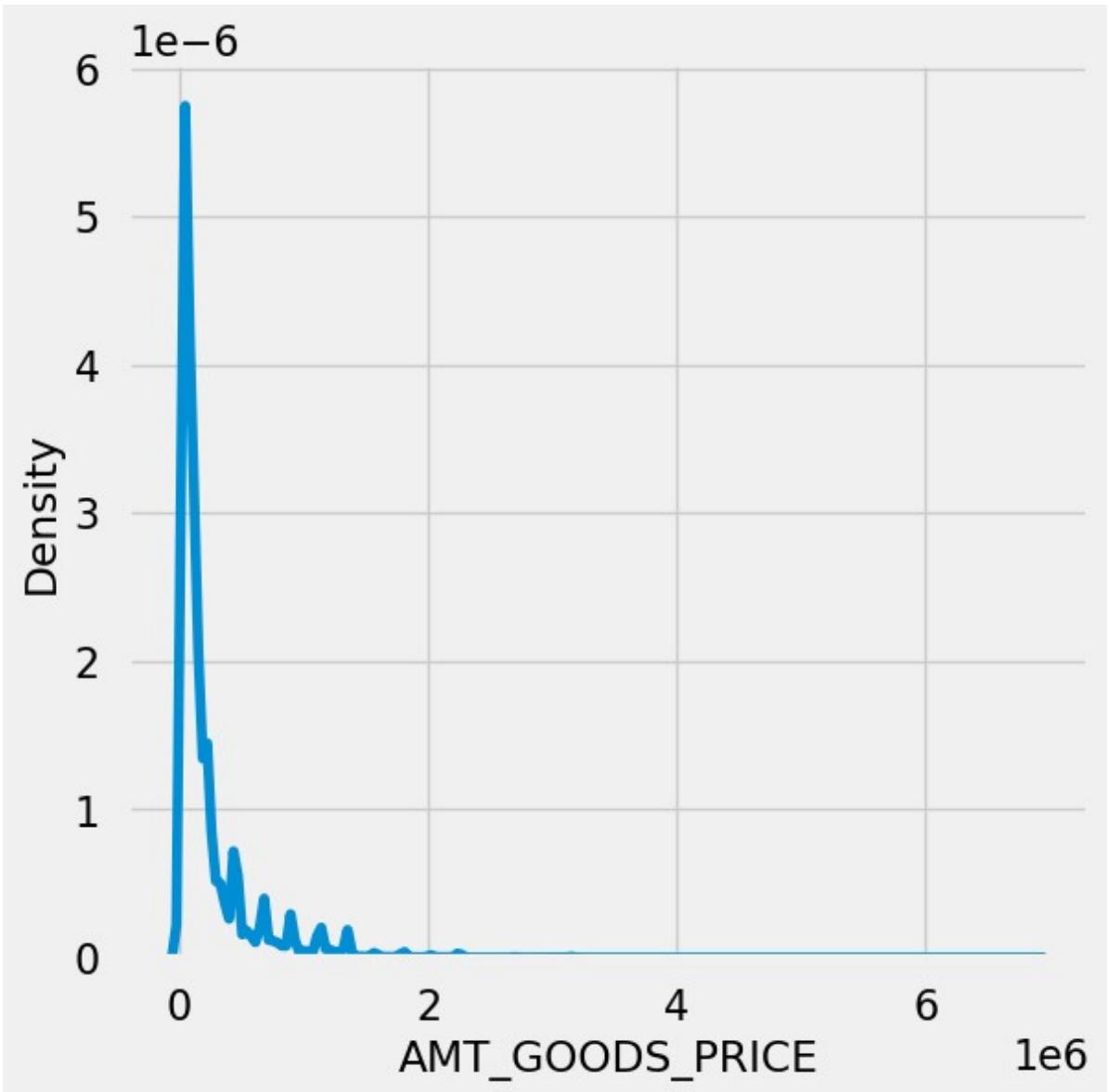
```
In [65]: round(applicationDF.isnull().sum()/previousDF.shape[0]* 100.00,2)
```

```
Out[65]: SK_ID_CURR           0.00  
TARGET              0.00  
NAME_CONTRACT_TYPE 0.00  
CODE_GENDER          0.00  
FLAG_OWN_CAR         0.00  
FLAG_OWN_REALTY     0.00  
CNT_CHILDREN         0.00  
AMT_INCOME_TOTAL    0.00  
AMT_CREDIT            0.00  
AMT_ANNUITY           0.00  
AMT_GOODS_PRICE       0.02  
NAME_TYPE_SUITE       0.00  
NAME_INCOME_TYPE     0.00  
NAME_EDUCATION_TYPE  0.00  
NAME_FAMILY_STATUS    0.00  
NAME_HOUSING_TYPE    0.00  
REGION_POPULATION_RELATIVE 0.00  
DAYS_BIRTH             0.00  
DAYS_EMPLOYED          0.00  
DAYS_REGISTRATION      0.00  
DAYS_ID_PUBLISH        0.00  
OCCUPATION_TYPE        0.00  
CNT_FAM_MEMBERS        0.00  
REGION_RATING_CLIENT   0.00  
REGION_RATING_CLIENT_W_CITY 0.00  
WEEKDAY_APPR_PROCESS_START 0.00  
HOUR_APPR_PROCESS_START 0.00  
REG_REGION_NOT_LIVE_REGION 0.00  
REG_REGION_NOT_WORK_REGION 0.00  
LIVE_REGION_NOT_WORK_REGION 0.00  
REG_CITY_NOT_LIVE_CITY 0.00  
REG_CITY_NOT_WORK_CITY 0.00  
LIVE_CITY_NOT_WORK_CITY 0.00  
ORGANIZATION_TYPE       0.00  
OBS_30_CNT_SOCIAL_CIRCLE 0.06  
DEF_30_CNT_SOCIAL_CIRCLE 0.06  
OBS_60_CNT_SOCIAL_CIRCLE 0.06  
DEF_60_CNT_SOCIAL_CIRCLE 0.06  
DAYS_LAST_PHONE_CHANGE 0.00  
FLAG_DOCUMENT_3          0.00  
AMT_REQ_CREDIT_BUREAU_HOUR 0.00  
AMT_REQ_CREDIT_BUREAU_DAY 0.00  
AMT_REQ_CREDIT_BUREAU_WEEK 0.00  
AMT_REQ_CREDIT_BUREAU_MON 0.00  
AMT_REQ_CREDIT_BUREAU_QRT 0.00  
AMT_REQ_CREDIT_BUREAU_YEAR 0.00  
AMT_INCOME_RANGE         0.01  
AMT_CREDIT_RANGE          0.00  
AGE                     0.00  
AGE_GROUP               0.00  
YEARS_EMPLOYED           0.00  
EMPLOYMENT_YEAR          4.99  
dtype: float64
```

```
In [66]: round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)
```

```
Out[66]: SK_ID_PREV          0.00  
SK_ID_CURR           0.00  
NAME_CONTRACT_TYPE   0.00  
AMT_ANNUITY          22.29  
AMT_APPLICATION      0.00  
AMT_CREDIT            0.00  
AMT_GOODS_PRICE       23.08  
NAME_CASH_LOAN_PURPOSE 0.00  
NAME_CONTRACT_STATUS  0.00  
DAYS_DECISION         0.00  
NAME_PAYMENT_TYPE     0.00  
CODE_REJECT_REASON    0.00  
NAME_CLIENT_TYPE      0.00  
NAME_GOODS_CATEGORY   0.00  
NAME_PORTFOLIO         0.00  
NAME_PRODUCT_TYPE     0.00  
CHANNEL_TYPE          0.00  
SELLERPLACE_AREA      0.00  
NAME_SELLER_INDUSTRY  0.00  
CNT_PAYMENT           22.29  
NAME_YIELD_GROUP      0.00  
PRODUCT_COMBINATION    0.02  
DAYS_DECISION_GROUP   0.00  
dtype: float64
```

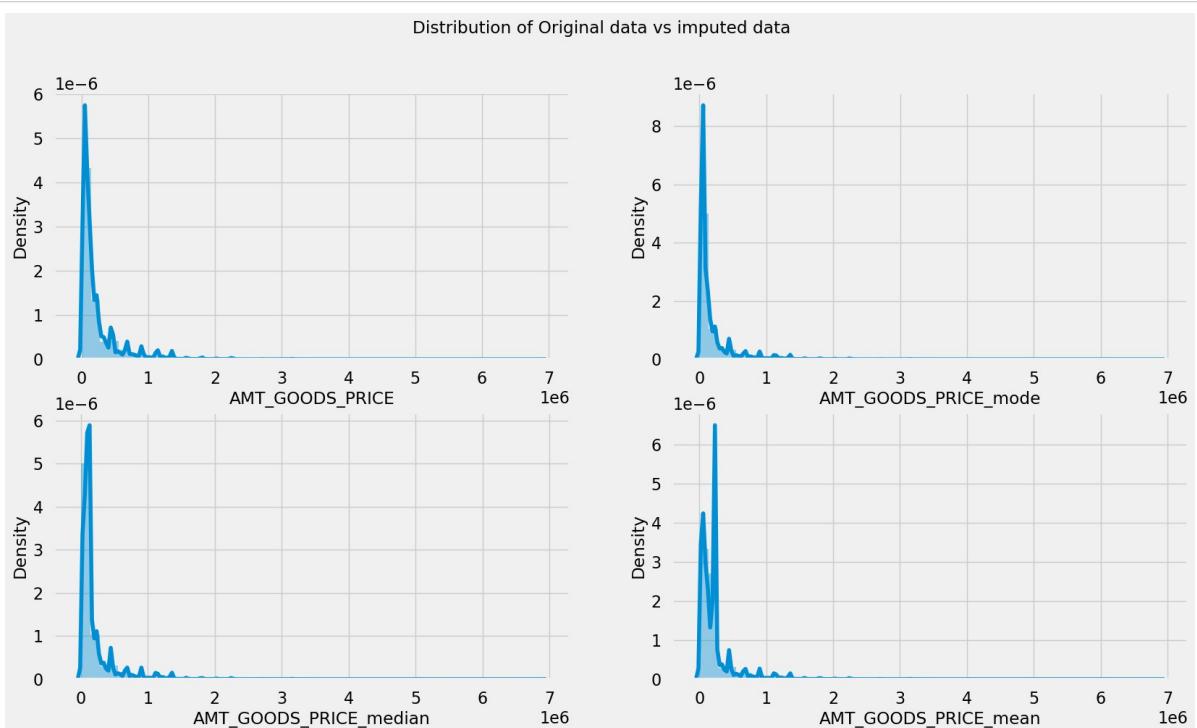
```
In [67]: plt.figure(figsize=(6,6))
sns.kdeplot(previousDF['AMT_GOODS_PRICE'][pd.notnull(previousDF['AMT_GOODS_PRICE'])])
plt.show()
```



```
In [68]: statsDF = pd.DataFrame()
statsDF['AMT_GOODS_PRICE_mode'] = previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].mode()[0])
statsDF['AMT_GOODS_PRICE_median'] = previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].median())
statsDF['AMT_GOODS_PRICE_mean'] = previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].mean())

cols = ['AMT_GOODS_PRICE_mode', 'AMT_GOODS_PRICE_median','AMT_GOODS_PRICE_mean']

plt.figure(figsize=(18,10))
plt.suptitle('Distribution of Original data vs imputed data')
plt.subplot(221)
sns.distplot(previousDF['AMT_GOODS_PRICE'][pd.notnull(previousDF['AMT_GOODS_PRICE'])])
for i in enumerate(cols):
    plt.subplot(2,2,i[0]+2)
    sns.distplot(statsDF[i[1]])
```



```
In [69]: previousDF['AMT_GOODS_PRICE'].fillna(previousDF['AMT_GOODS_PRICE'].mode()[0], inplace=True)
```

```
In [70]: previousDF.loc[previousDF['CNT_PAYMENT'].isnull(), 'NAME_CONTRACT_STATUS'].value_counts()
```

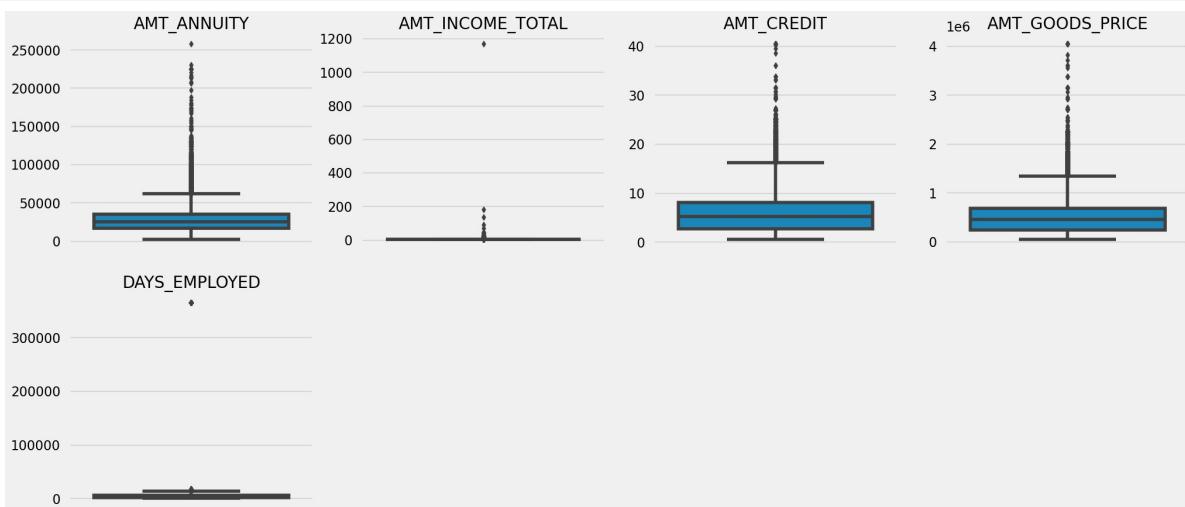
```
Out[70]: Canceled      305805
Refused       40897
Unused offer   25524
Approved        4
Name: NAME_CONTRACT_STATUS, dtype: int64
```

```
In [71]: previousDF['CNT_PAYMENT'].fillna(0,inplace = True)
```

```
In [72]: round(previousDF.isnull().sum() / previousDF.shape[0] * 100.00,2)
```

```
Out[72]: SK_ID_PREV          0.00
SK_ID_CURR           0.00
NAME_CONTRACT_TYPE   0.00
AMT_ANNUITY          22.29
AMT_APPLICATION      0.00
AMT_CREDIT           0.00
AMT_GOODS_PRICE       0.00
NAME_CASH_LOAN_PURPOSE 0.00
NAME_CONTRACT_STATUS  0.00
DAYS_DECISION        0.00
NAME_PAYMENT_TYPE     0.00
CODE_REJECT_REASON    0.00
NAME_CLIENT_TYPE      0.00
NAME_GOODS_CATEGORY   0.00
NAME_PORTFOLIO        0.00
NAME_PRODUCT_TYPE     0.00
CHANNEL_TYPE          0.00
SELLERPLACE_AREA      0.00
NAME_SELLER_INDUSTRY 0.00
CNT_PAYMENT           0.00
NAME_YIELD_GROUP      0.00
PRODUCT_COMBINATION   0.02
DAYS_DECISION_GROUP   0.00
dtype: float64
```

```
In [73]: plt.figure(figsize=(22,10))
app_outlier_col_1 = ['AMT_ANNUITY', 'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_GOODS_PRICE']
app_outlier_col_2 = ['CNT_CHILDREN', 'DAYS_BIRTH']
for i in enumerate(app_outlier_col_1):
    plt.subplot(2,4,i[0]+1)
    sns.boxplot(y=applicationDF[i[1]])
    plt.title(i[1])
    plt.ylabel("")
```

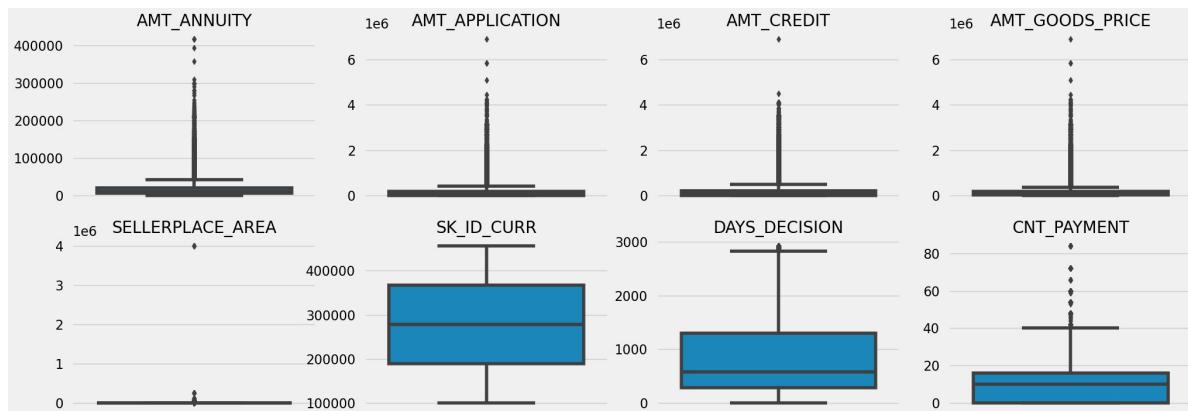


In [74]: applicationDF[['AMT\_ANNUITY', 'AMT\_INCOME\_TOTAL', 'AMT\_CREDIT', 'AMT\_GOODS\_PRI

Out[74]:

	AMT_ANNUITY	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_GOODS_PRICE	DAYS_BIRTH
<b>count</b>	307499.000000	307511.000000	307511.000000	3.072330e+05	307511.000000
<b>mean</b>	27108.573909	1.687979	5.990260	5.383962e+05	16036.995067
<b>std</b>	14493.737315	2.371231	4.024908	3.694465e+05	4363.988632
<b>min</b>	1615.500000	0.256500	0.450000	4.050000e+04	7489.000000
<b>25%</b>	16524.000000	1.125000	2.700000	2.385000e+05	12413.000000
<b>50%</b>	24903.000000	1.471500	5.135310	4.500000e+05	15750.000000
<b>75%</b>	34596.000000	2.025000	8.086500	6.795000e+05	19682.000000
<b>max</b>	258025.500000	1170.000000	40.500000	4.050000e+06	25229.000000

In [75]: plt.figure(figsize=(22,8))  
prev\_outlier\_col\_1 = ['AMT\_ANNUITY', 'AMT\_APPLICATION', 'AMT\_CREDIT', 'AMT\_GOODS\_PRICE']  
prev\_outlier\_col\_2 = ['SK\_ID\_CURR', 'DAYS\_DECISION', 'CNT\_PAYMENT']  
for i in enumerate(prev\_outlier\_col\_1):  
 plt.subplot(2,4,i[0]+1)  
 sns.boxplot(y=previousDF[i[1]])  
 plt.title(i[1])  
 plt.ylabel("")  
  
for i in enumerate(prev\_outlier\_col\_2):  
 plt.subplot(2,4,i[0]+6)  
 sns.boxplot(y=previousDF[i[1]])  
 plt.title(i[1])  
 plt.ylabel("")



```
In [76]: previousDF[['AMT_ANNUITY', 'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_GOODS_PRICE',
```

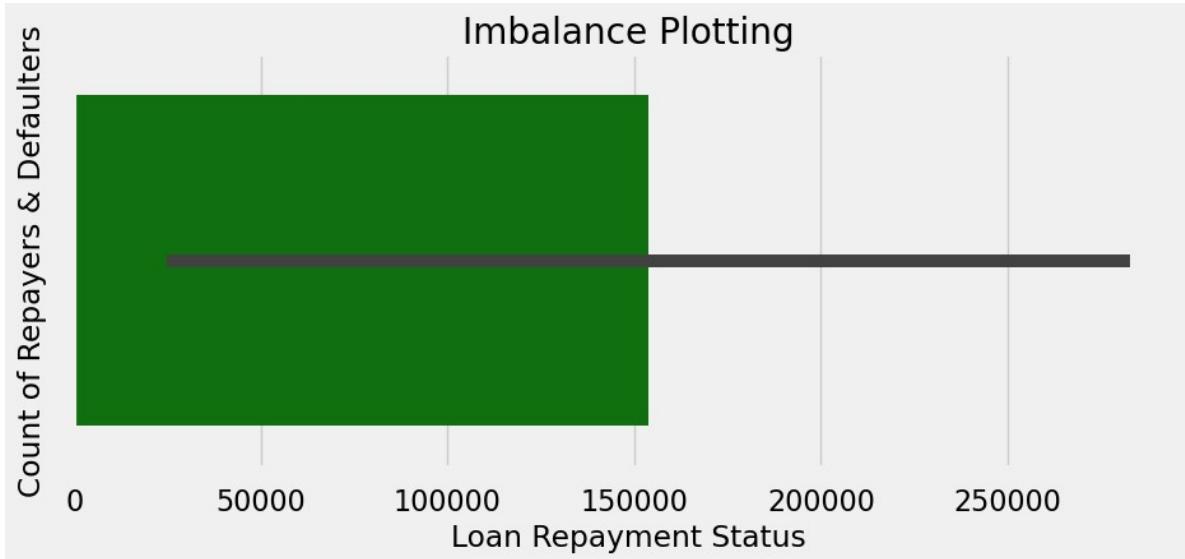
Out[76]:

	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_GOODS_PRICE	SELLERPLACE_AMT_GOODS_PRICE
count	1.297979e+06	1.670214e+06	1.670213e+06	1.670214e+06	1.670214e+06
mean	1.595512e+04	1.752339e+05	1.961140e+05	1.856429e+05	3.139511e+04
std	1.478214e+04	2.927798e+05	3.185746e+05	2.871413e+05	7.127443e+04
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	-1.000000e+00
25%	6.321780e+03	1.872000e+04	2.416050e+04	4.500000e+04	-1.000000e+00
50%	1.125000e+04	7.104600e+04	8.054100e+04	7.105050e+04	3.000000e+04
75%	2.065842e+04	1.803600e+05	2.164185e+05	1.804050e+05	8.200000e+04
max	4.180581e+05	6.905160e+06	6.905160e+06	6.905160e+06	4.000000e+06

```
In [77]: Imbalance = applicationDF["TARGET"].value_counts().reset_index()
```

```
plt.figure(figsize=(10,4))
x= ['Repayer', 'Defaulter']

sns.barplot(x="TARGET", data = Imbalance, palette= ['g', 'r'])
plt.xlabel("Loan Repayment Status")
plt.ylabel("Count of Repayers & Defaulters")
plt.title("Imbalance Plotting")
plt.show()
```



```
In [78]: count_0 = Imbalance.iloc[0]["TARGET"]
count_1 = Imbalance.iloc[1]["TARGET"]
count_0_perc = round(count_0/(count_0+count_1)*100,2)
count_1_perc = round(count_1/(count_0+count_1)*100,2)

print('Ratios of imbalance in percentage with respect to Repayer and Defaulter
      print('Ratios of imbalance in relative with respect to Repayer and Defaulter d
```

Ratios of imbalance in percentage with respect to Repayer and Defaulter  
data are: 91.93 and 8.07

Ratios of imbalance in relative with respect to Repayer and Defaulter data is  
11.39 : 1 (approx)

```
In [79]: def univariate_categorical(feature,ylog=False,label_rotation=False,horizontal_
temp = applicationDF[feature].value_counts()

df1 = pd.DataFrame({feature: temp.index, 'Number of contracts': temp.values
cat_perc = applicationDF[[feature, 'TARGET']].groupby([feature],as_index=False)
cat_perc["TARGET"] = cat_perc["TARGET"]*100
cat_perc.sort_values(by='TARGET', ascending=False, inplace=True)

if(horizontal_layout):
    fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,6))
else:
    fig, (ax1, ax2) = plt.subplots(nrows=2, figsize=(20,24))

s = sns.countplot(ax=ax1,
                  x = feature,
                  data=applicationDF,
                  hue ="TARGET",
                  order=cat_perc[feature],
                  palette=['g','r'])

ax1.set_title(feature, fontdict={'fontsize' : 10, 'fontweight' : 3, 'color' : 'black'})
ax1.legend(['Repayer','Defaulter'])

if ylog:
    ax1.set_yscale('log')
    ax1.set_ylabel("Count (log)",fontdict={'fontsize' : 10, 'fontweight' : 3, 'color' : 'black'})

if(label_rotation):
    s.set_xticklabels(s.get_xticklabels(),rotation=90)

s = sns.barplot(ax=ax2,
                 x = feature,
                 y='TARGET',
                 order=cat_perc[feature],
                 data=cat_perc,
                 palette='Set2')

if(label_rotation):
    s.set_xticklabels(s.get_xticklabels(),rotation=90)
plt.ylabel('Percent of Defaulters [%]', fontsize=10)
plt.tick_params(axis='both', which='major', labelsize=10)
ax2.set_title(feature + " Defaulter %", fontdict={'fontsize' : 15, 'fontweight' : 3, 'color' : 'black'})

plt.show();
```

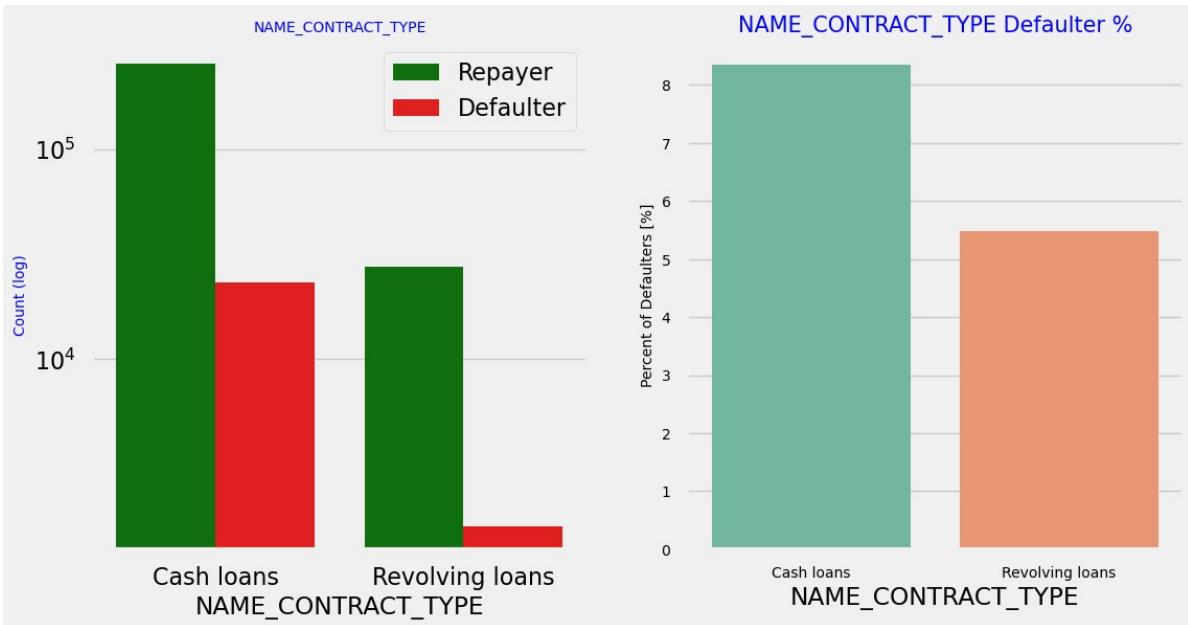
```
In [80]: def bivariate_bar(x,y,df,hue,figsize):  
  
    plt.figure(figsize=figsize)  
    sns.barplot(x=x,  
                y=y,  
                data=df,  
                hue=hue,  
                palette =['g','r'])  
  
    #  
    plt.xlabel(x,fontdict={'fontsize' : 10, 'fontweight' : 3, 'color' : 'Blue'}  
    plt.ylabel(y,fontdict={'fontsize' : 10, 'fontweight' : 3, 'color' : 'Blue'}  
    plt.title(col, fontdict={'fontsize' : 15, 'fontweight' : 5, 'color' : 'Blue'}  
    plt.xticks(rotation=90, ha='right')  
    plt.legend(labels = ['Repayer','Defaulter'])  
    plt.show()
```

```
In [81]: def bivariate_rel(x,y,data, hue, kind, palette, legend,figsize):  
  
    plt.figure(figsize=figsize)  
    sns.relplot(x=x,  
                y=y,  
                data=applicationDF,  
                hue="TARGET",  
                kind=kind,  
                palette = ['g','r'],  
                legend = False)  
    plt.legend(['Repayer','Defaulter'])  
    plt.xticks(rotation=90, ha='right')  
    plt.show()
```

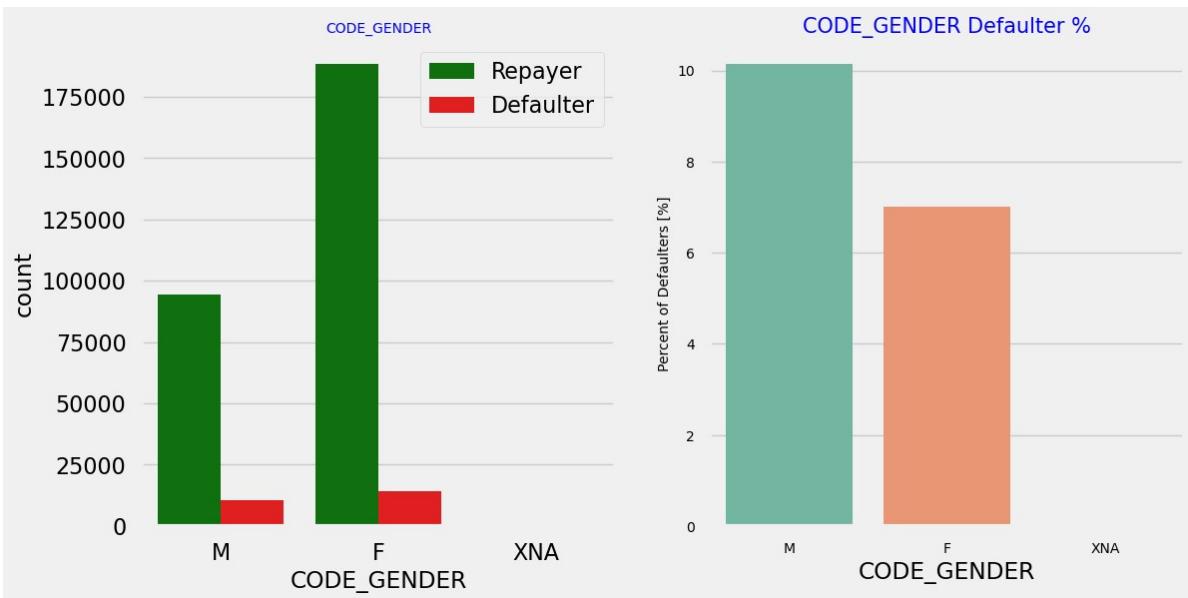
```
In [82]: def univariate_merged(col,df,hue,palette,ylog,figsize):  
    plt.figure(figsize=figsize)  
    ax=sns.countplot(x=col,  
                      data=df,  
                      hue= hue,  
                      palette= palette,  
                      order=df[col].value_counts().index)  
  
    if ylog:  
        plt.yscale('log')  
        plt.ylabel("Count (log)",fontdict={'fontsize' : 10, 'fontweight' : 3},  
    else:  
        plt.ylabel("Count",fontdict={'fontsize' : 10, 'fontweight' : 3, 'color' : 'Blue'})  
  
    plt.title(col , fontdict={'fontsize' : 15, 'fontweight' : 5, 'color' : 'Blue'}  
    plt.legend(loc = "upper right")  
    plt.xticks(rotation=90, ha='right')  
  
    plt.show()
```

```
In [83]: def merged_pointplot(x,y):
    plt.figure(figsize=(8,4))
    sns.pointplot(x=x,
                  y=y,
                  hue="TARGET",
                  data=loan_process_df,
                  palette =[ 'g' , 'r'])
```

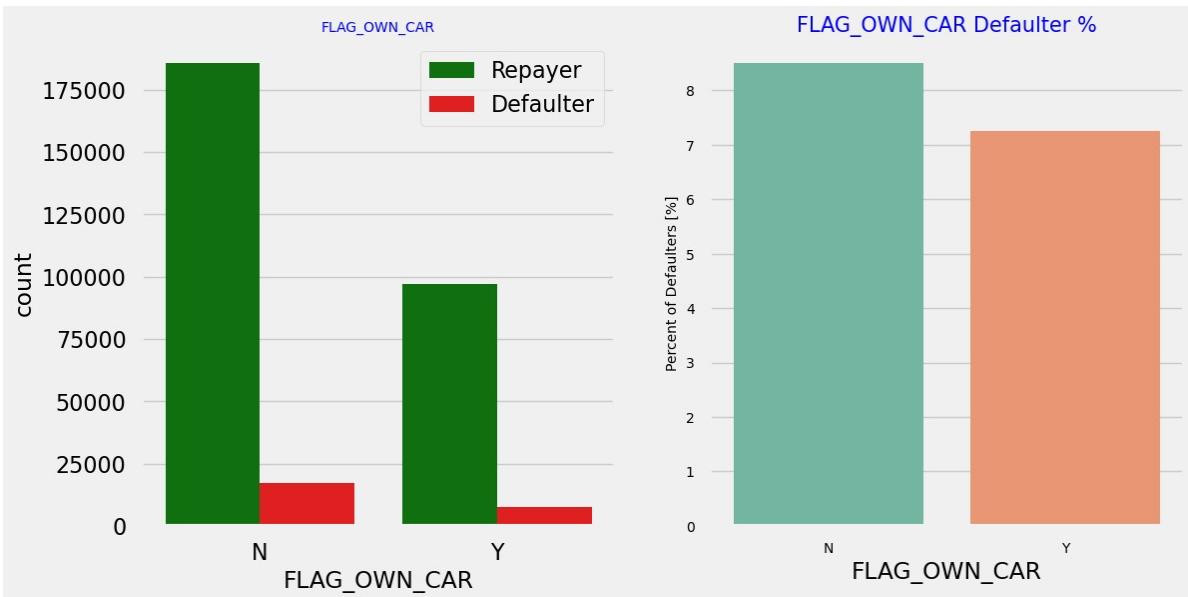
```
In [84]: univariate_categorical('NAME_CONTRACT_TYPE',True)
```



```
In [85]: univariate_categorical('CODE_GENDER')
```



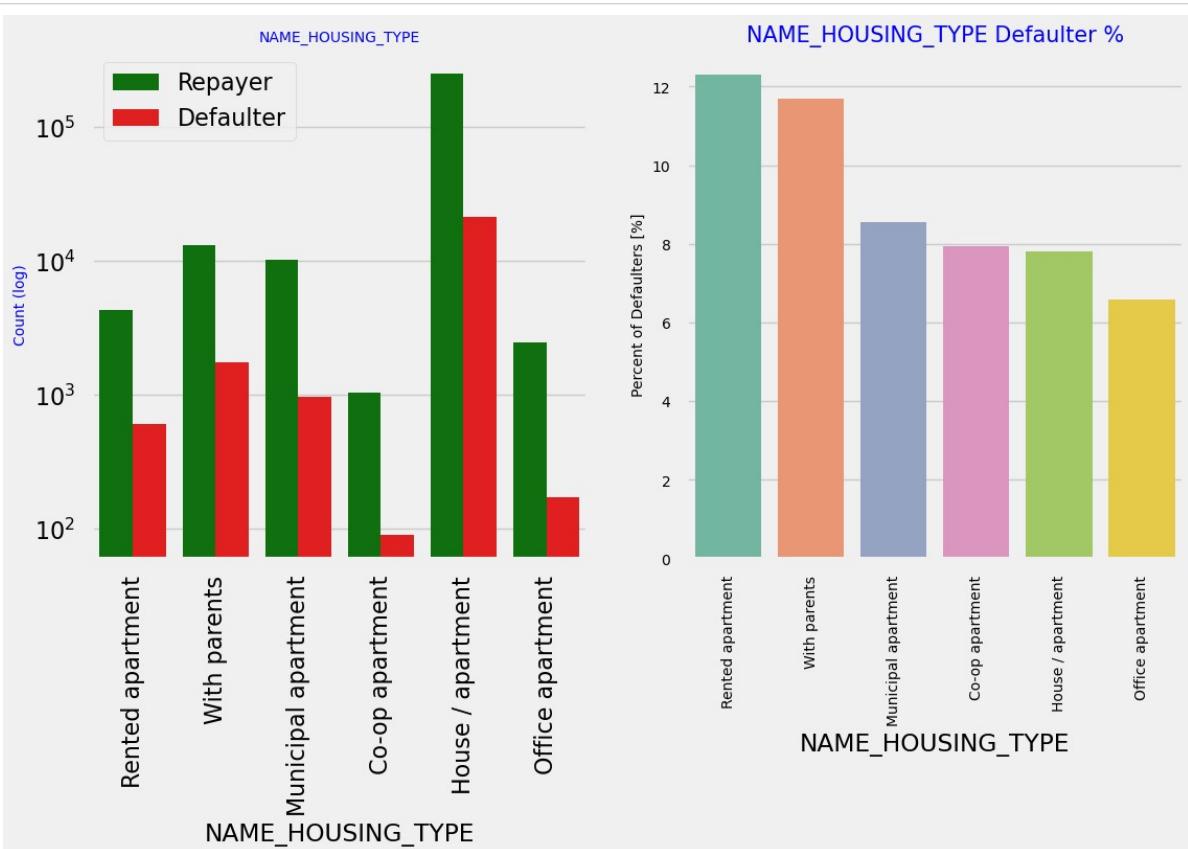
```
In [86]: univariate_categorical('FLAG_OWN_CAR')
```



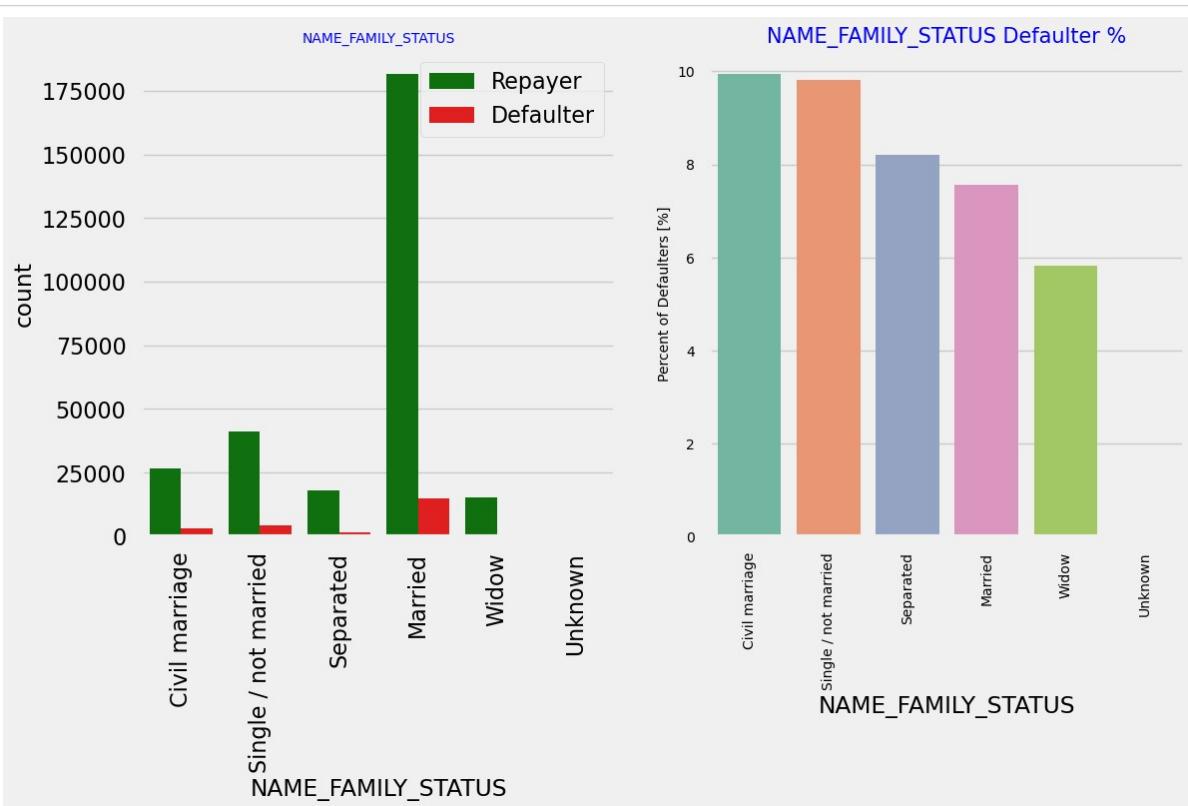
```
In [87]: univariate_categorical('FLAG_OWN_REALTY')
```



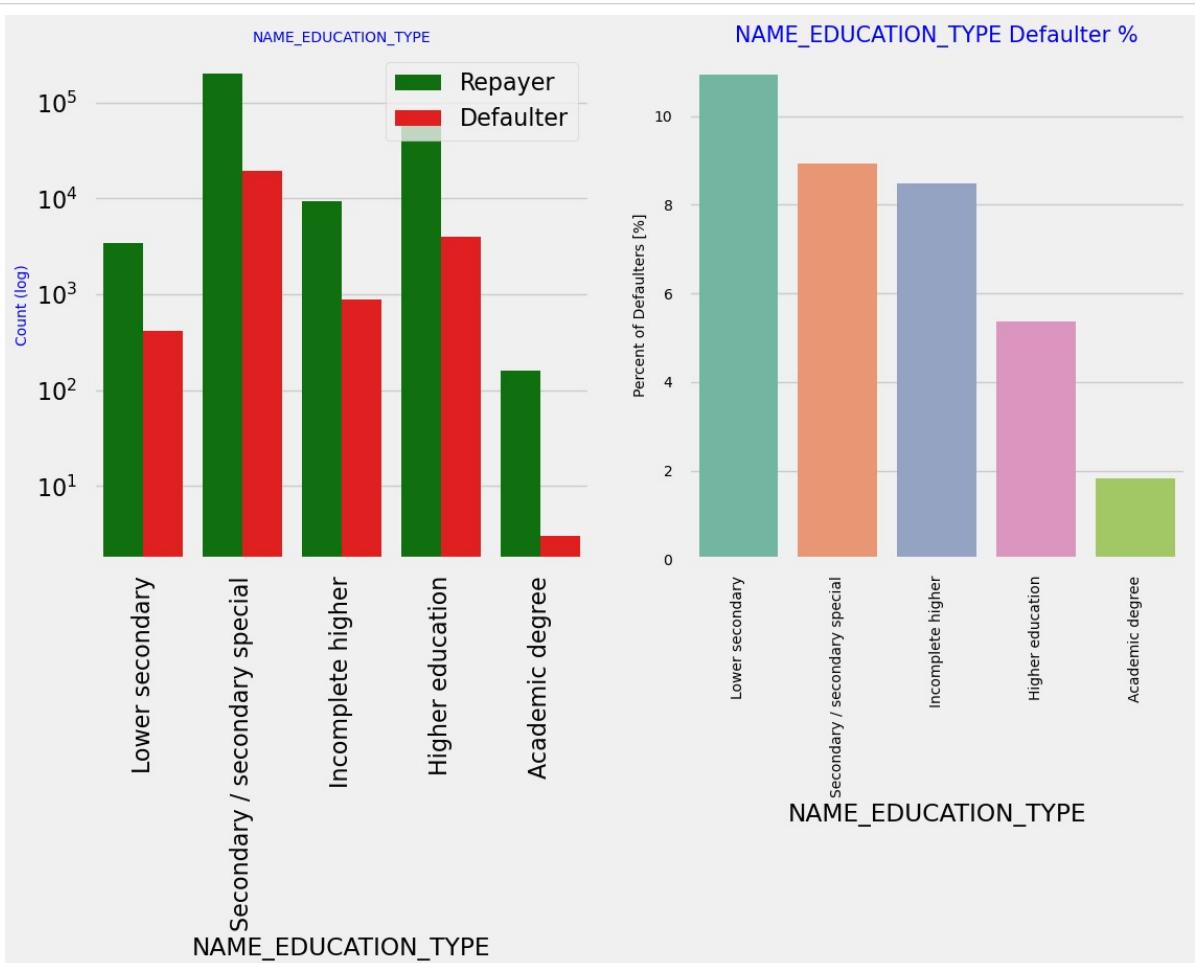
```
In [88]: univariate_categorical("NAME_HOUSING_TYPE",True,True,True)
```

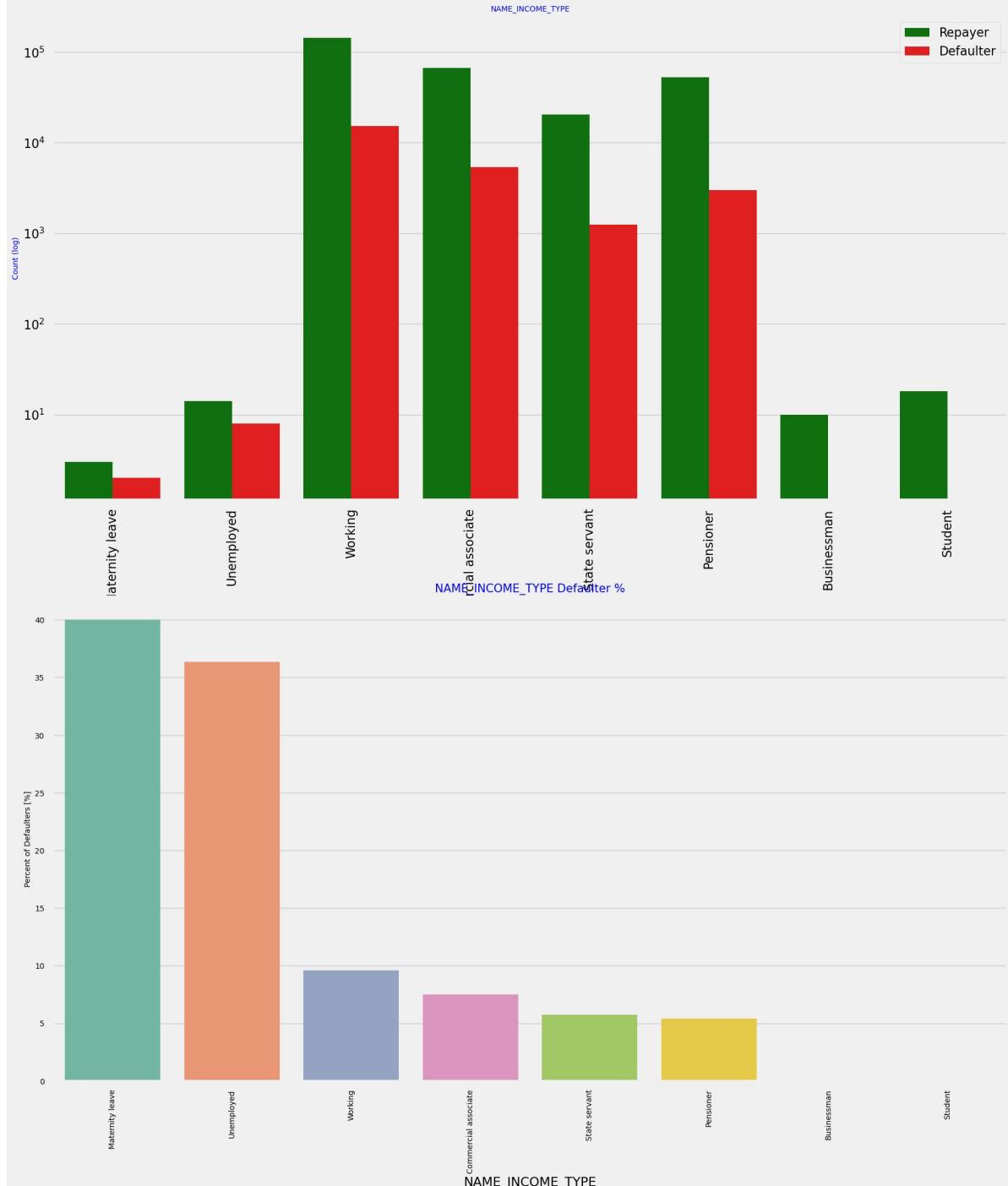


```
In [89]: univariate_categorical("NAME_FAMILY_STATUS",False,True,True)
```

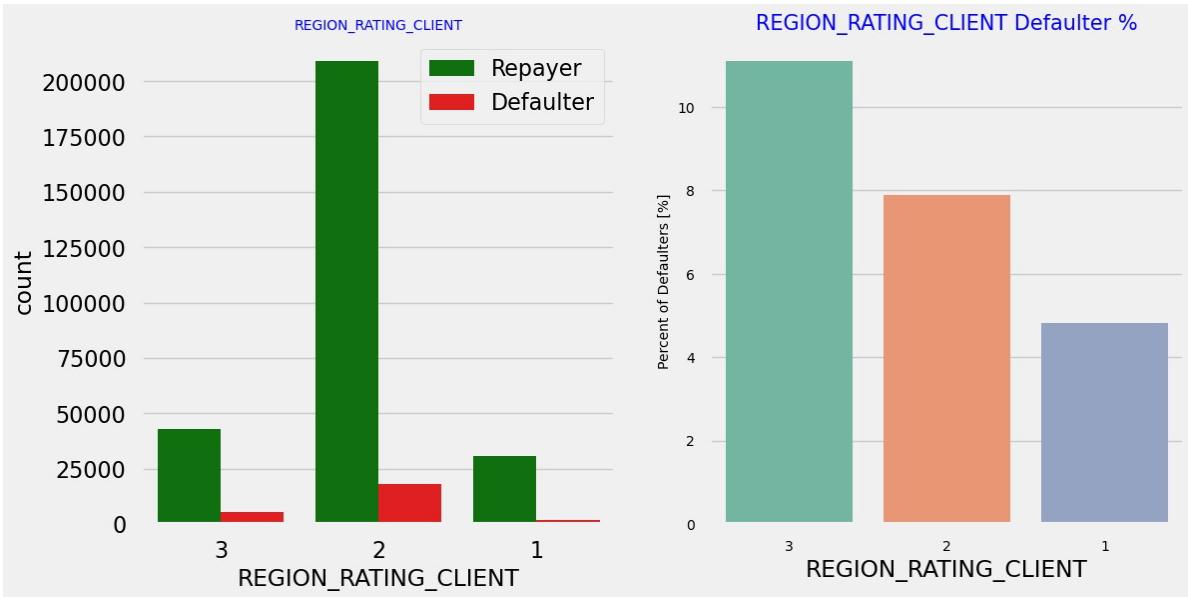


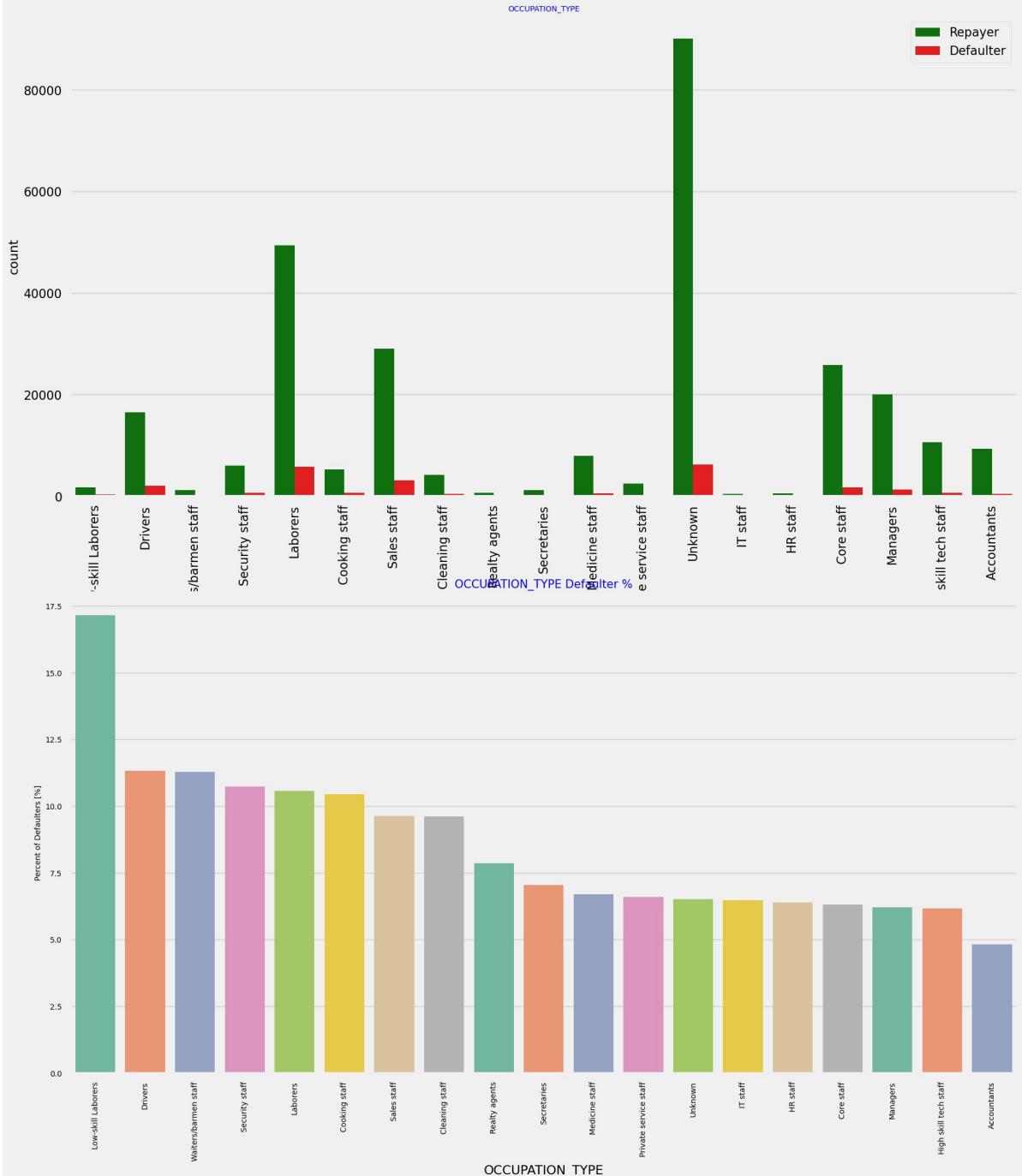
```
In [90]: univariate_categorical("NAME_EDUCATION_TYPE",True,True,True)
```



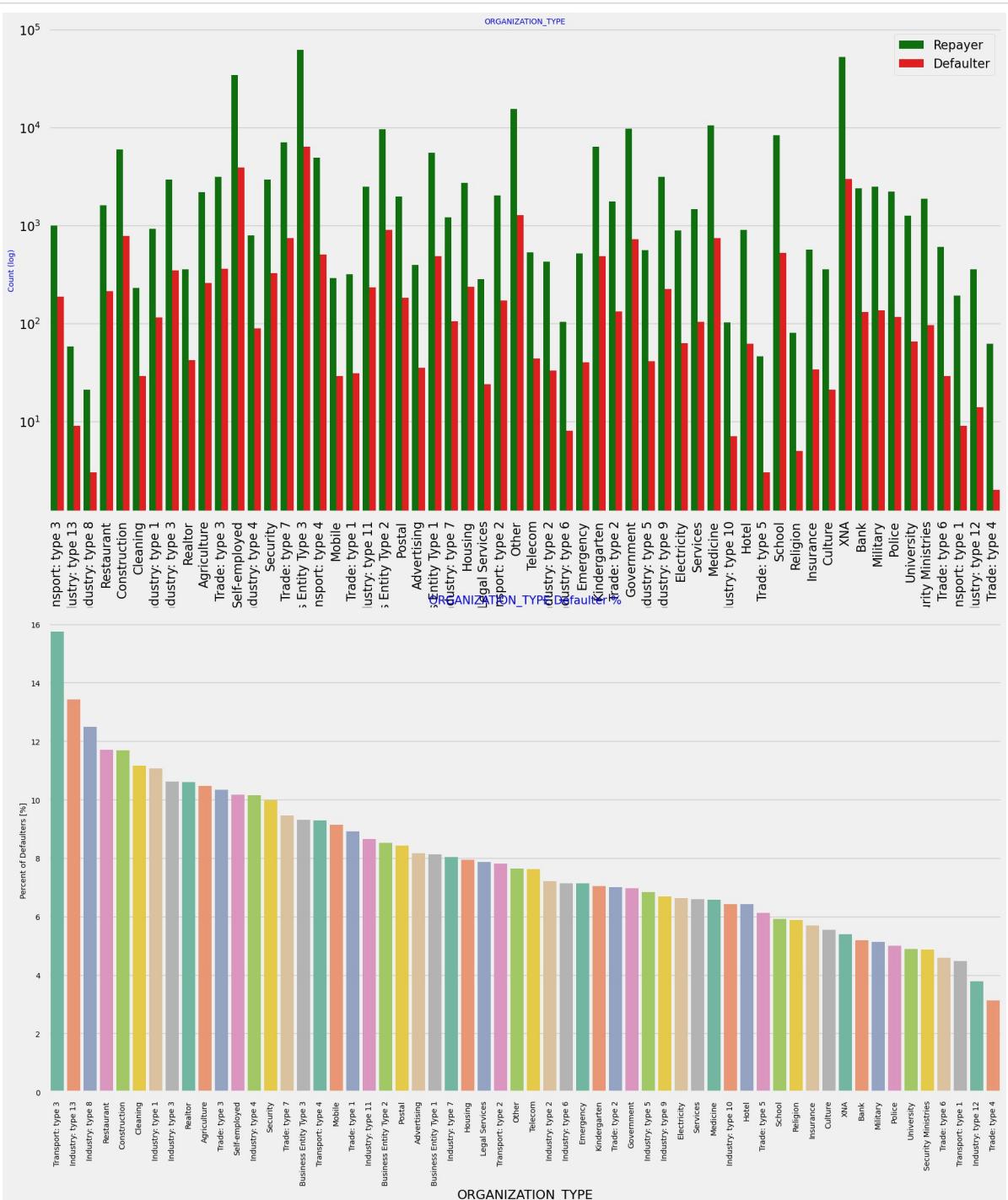


```
In [92]: univariate_categorical("REGION_RATING_CLIENT", False, False, True)
```

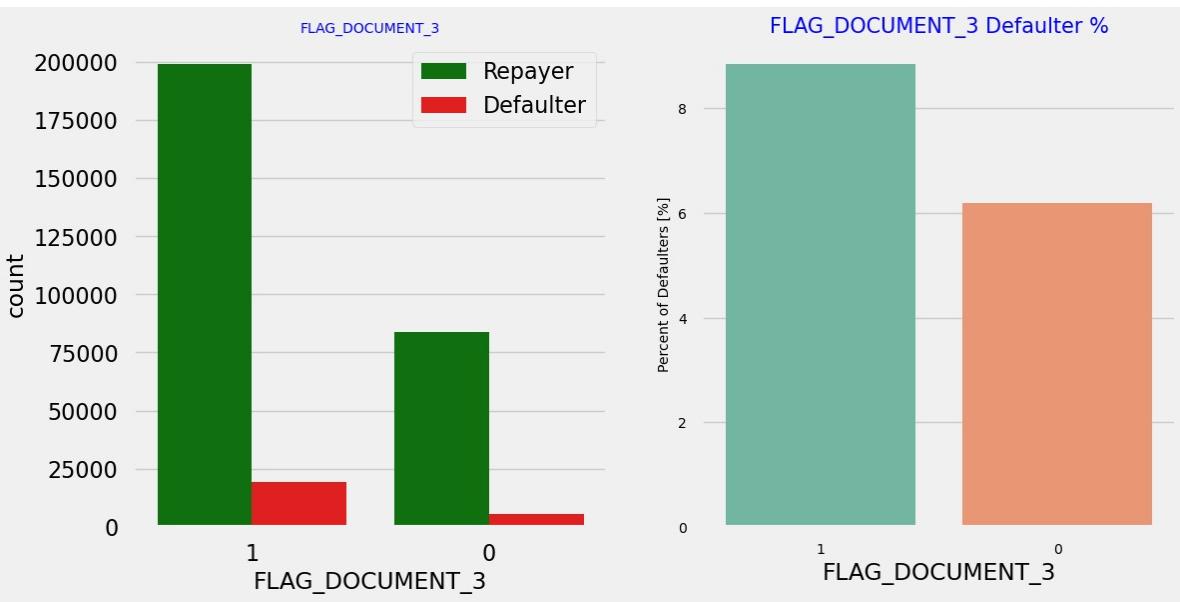




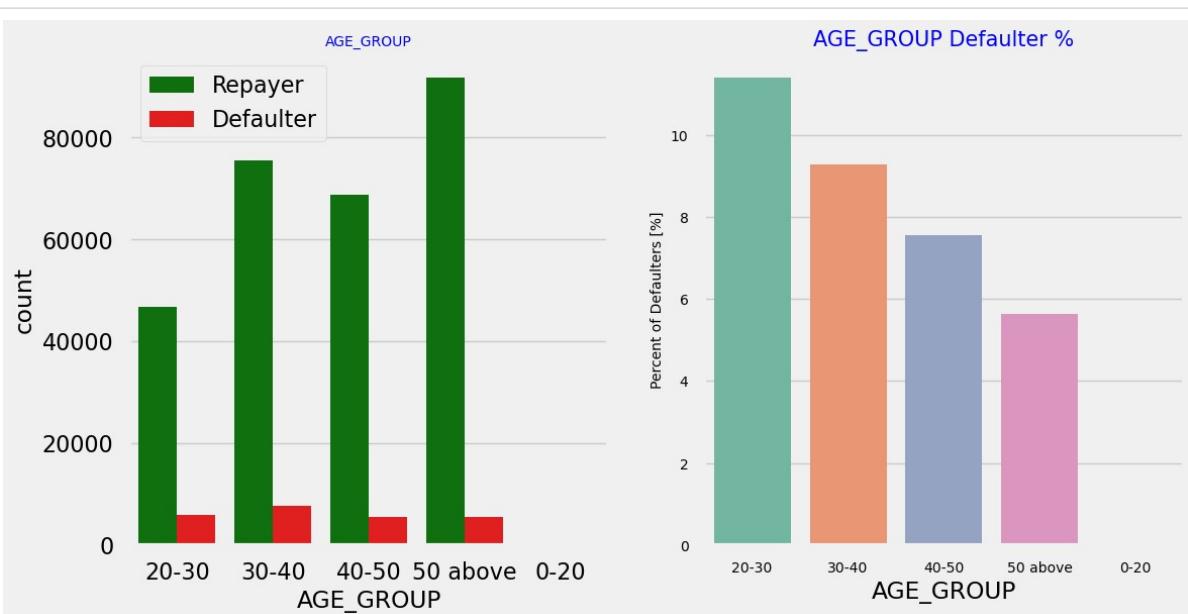
In [94]: `univariate_categorical("ORGANIZATION_TYPE", True, True, False)`

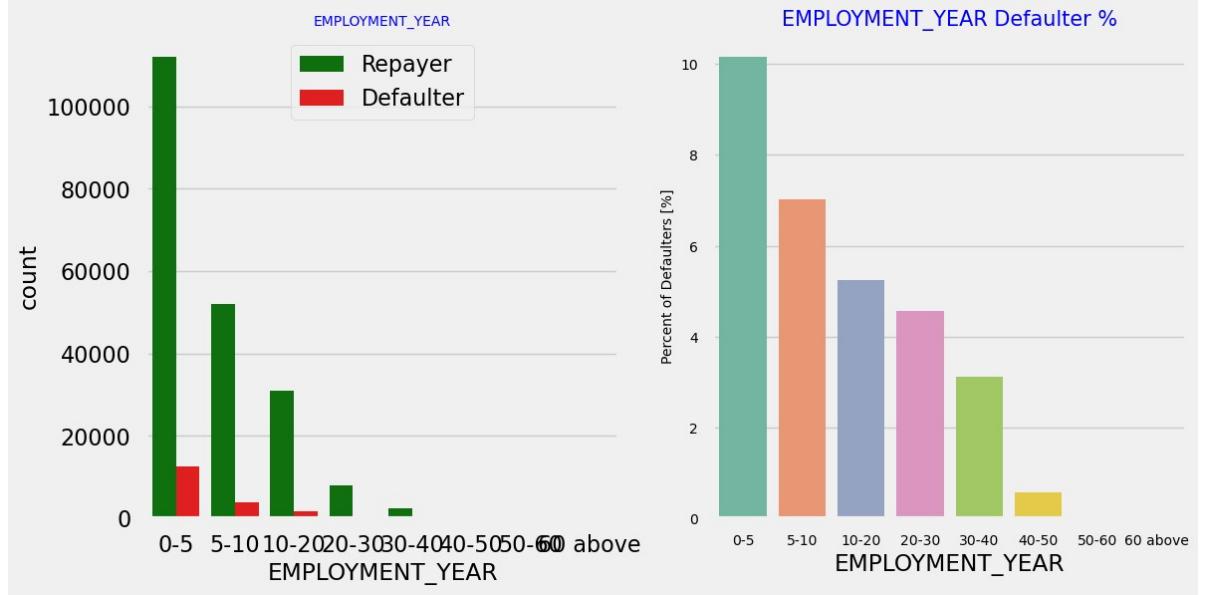


```
In [95]: univariate_categorical("FLAG_DOCUMENT_3", False, False, True)
```

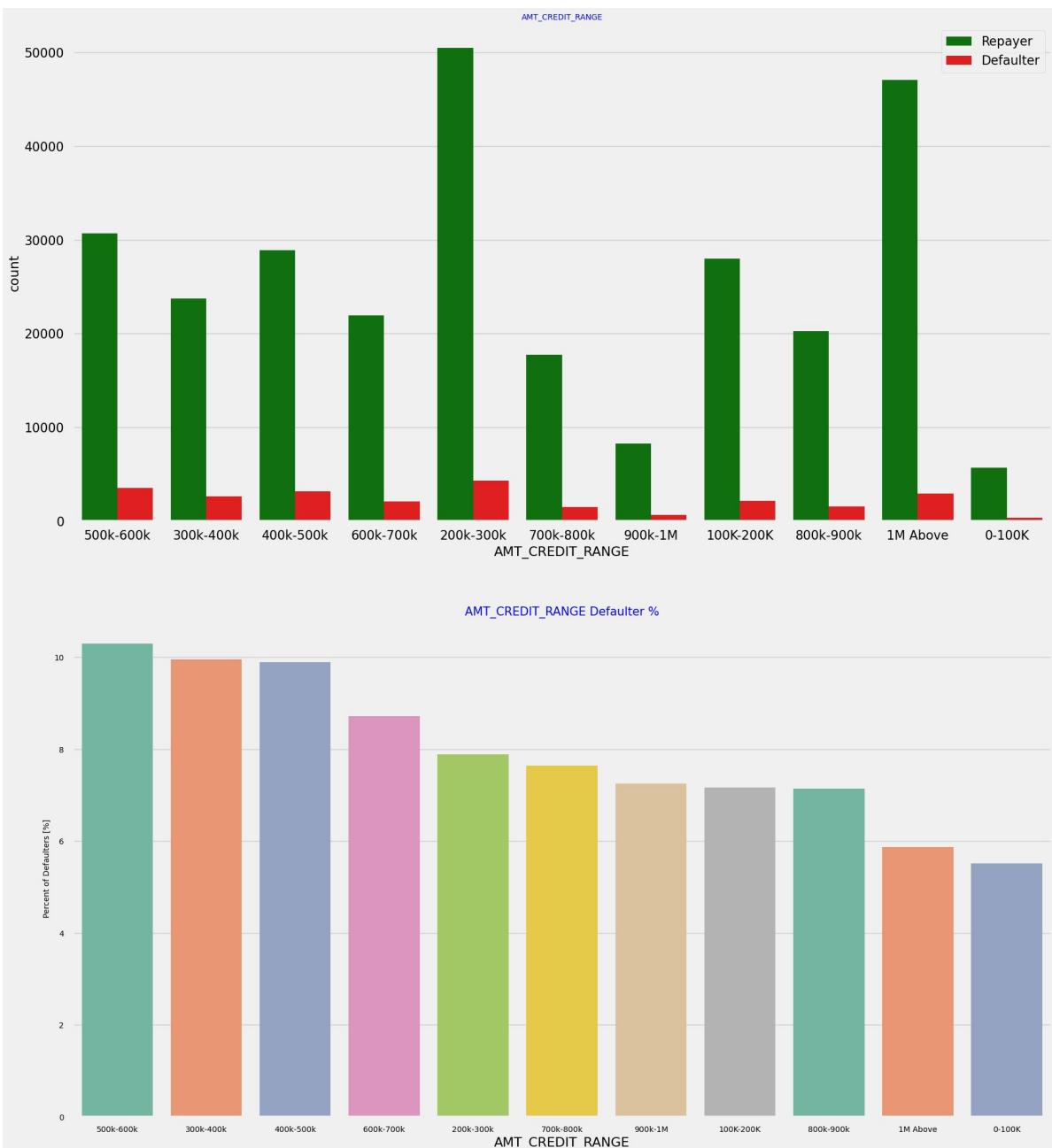


```
In [96]: univariate_categorical("AGE_GROUP", False, False, True)
```

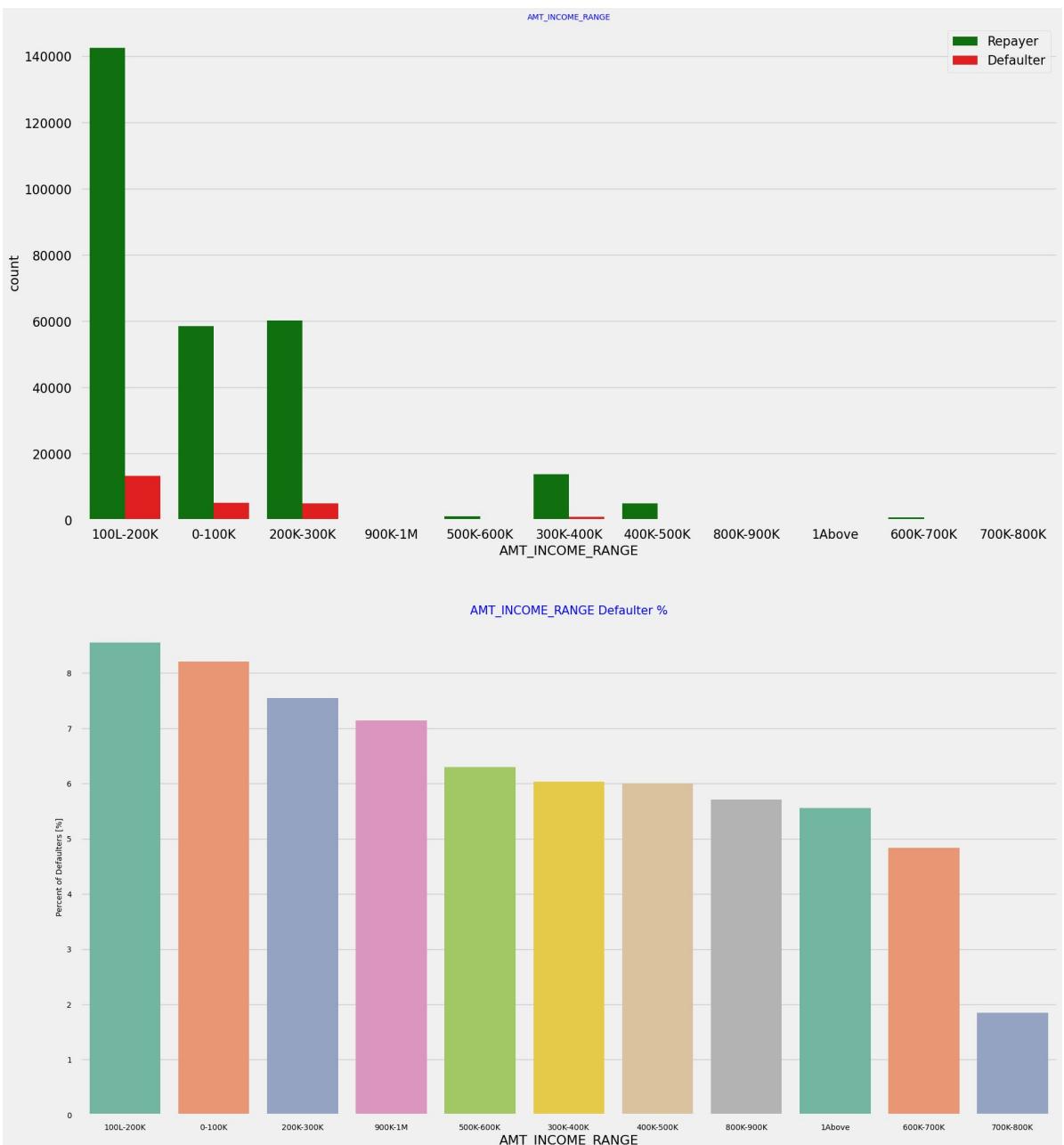




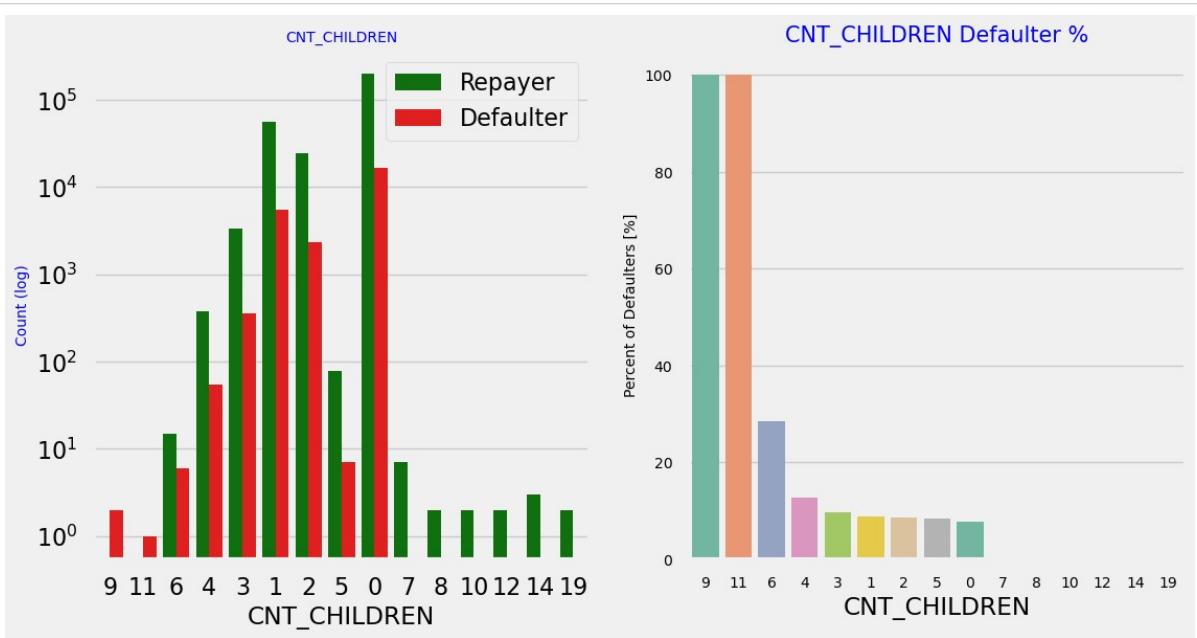
```
In [98]: univariate_categorical("AMT_CREDIT_RANGE", False, False, False)
```



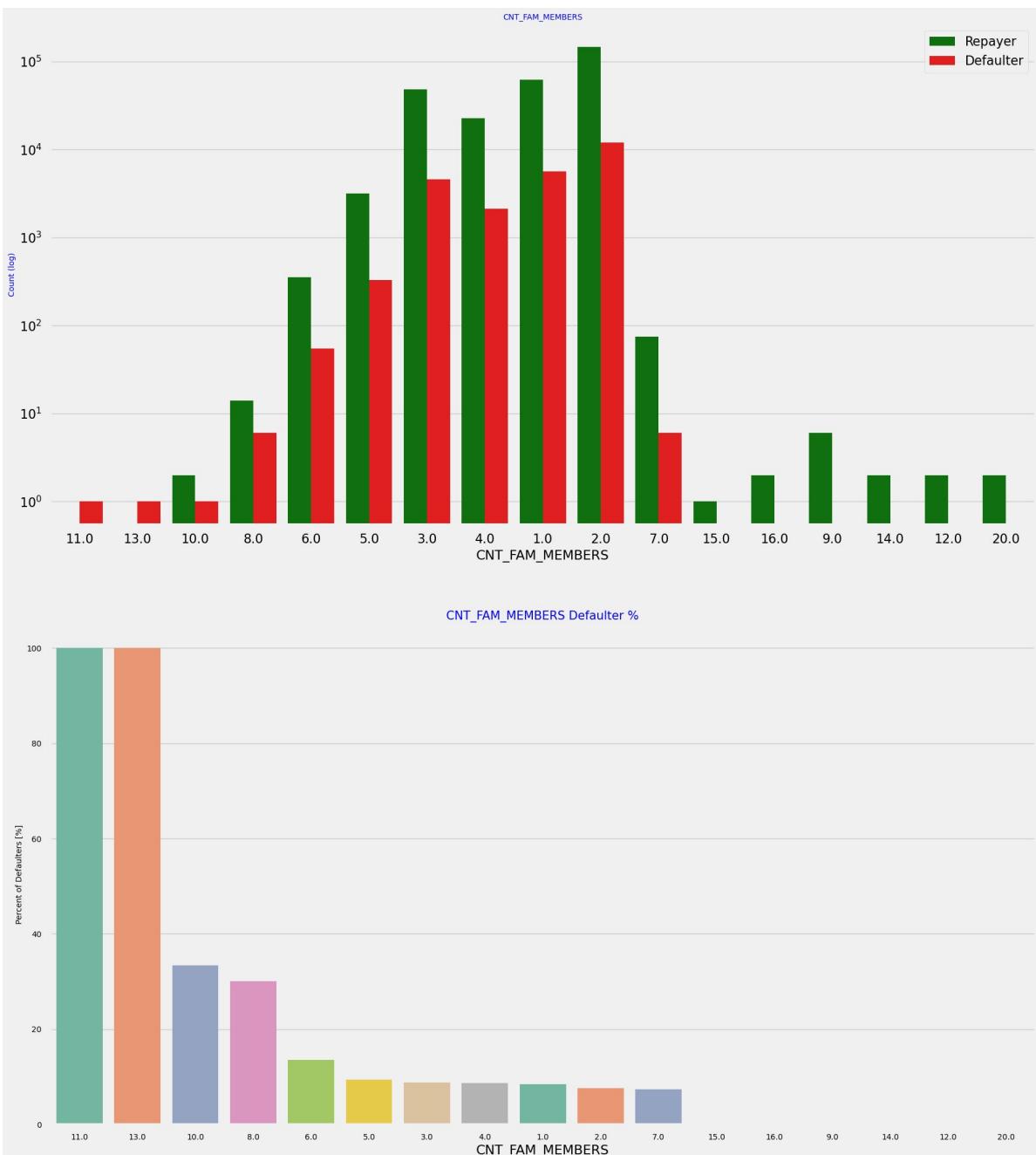
```
In [99]: univariate_categorical("AMT_INCOME_RANGE", False, False, False)
```



```
In [100]: univariate_categorical("CNT_CHILDREN",True)
```



```
In [101]: univariate_categorical("CNT_FAM_MEMBERS",True, False, False)
```

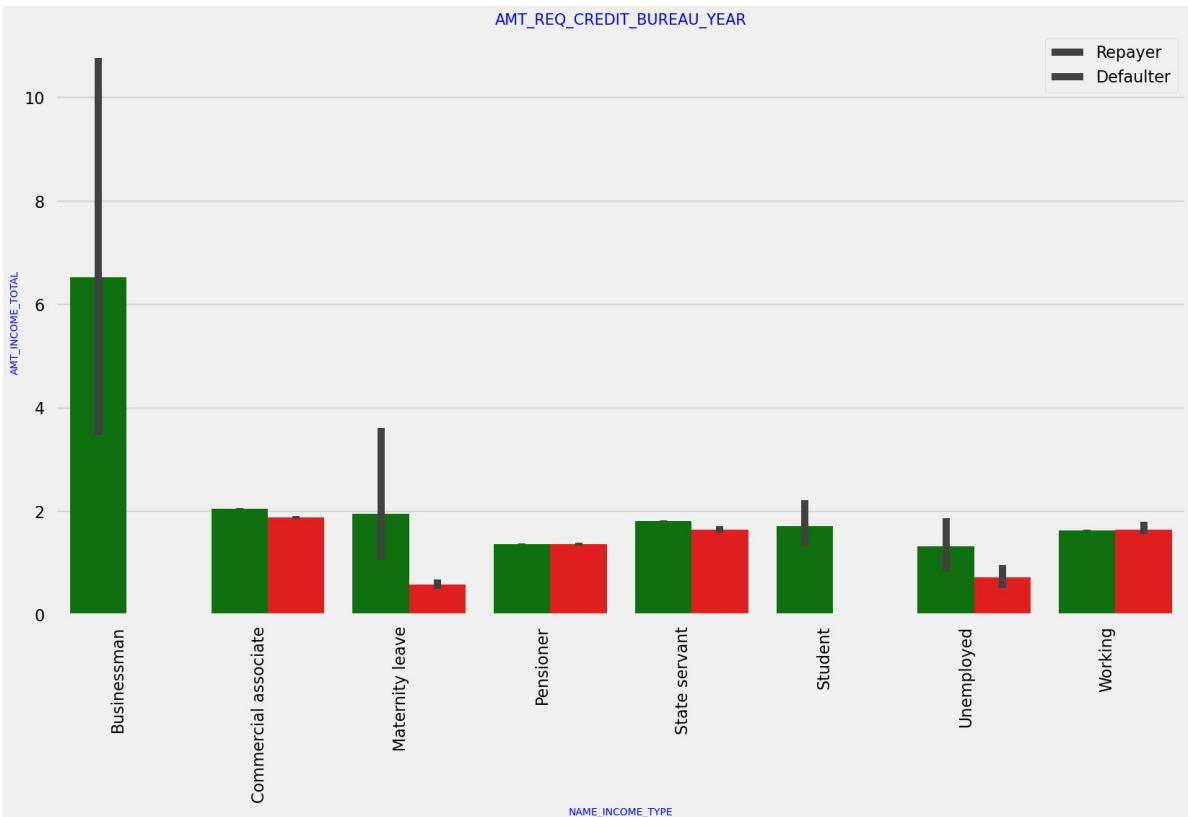


```
In [102]: applicationDF.groupby('NAME_INCOME_TYPE')['AMT_INCOME_TOTAL'].describe()
```

Out[102]:

NAME_INCOME_TYPE	count	mean	std	min	25%	50%	75%	max
<b>Businessman</b>	10.0	6.525000	6.272260	1.8000	2.250	4.9500	8.43750	22.5000
<b>Commercial associate</b>	71617.0	2.029553	1.479742	0.2655	1.350	1.8000	2.25000	180.00009
<b>Maternity leave</b>	5.0	1.404000	1.268569	0.4950	0.675	0.9000	1.35000	3.6000
<b>Pensioner</b>	55362.0	1.364013	0.766503	0.2565	0.900	1.1700	1.66500	22.5000
<b>State servant</b>	21703.0	1.797380	1.008806	0.2700	1.125	1.5750	2.25000	31.5000
<b>Student</b>	18.0	1.705000	1.066447	0.8100	1.125	1.5750	1.78875	5.6250
<b>Unemployed</b>	22.0	1.105364	0.880551	0.2655	0.540	0.7875	1.35000	3.3750
<b>Working</b>	158774.0	1.631699	3.075777	0.2565	1.125	1.3500	2.02500	1170.0000

```
In [103]: bivariate_bar("NAME_INCOME_TYPE", "AMT_INCOME_TOTAL", applicationDF, "TARGET", (18, 18))
```



```
In [104]: applicationDF.columns
```

```
Out[104]: Index(['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'OCCUPATION_TYPE', 'CNT_FAM_MEMBERS', 'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY', 'WEEKDAY_APPR_PROCESS_START', 'HOUR_APPR_PROCESS_START', 'REG_REGION_NOT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION', 'LIVE_REGION_NOT_WORK_REGION', 'REG_CITY_NOT_LIVE_CITY', 'REG_CITY_NOT_WORK_CITY', 'LIVE_CITY_NOT_WORK_CITY', 'ORGANIZATION_TYPE', 'OBS_30_CNT_SOCIAL_CIRCLE', 'DEF_30_CNT_SOCIAL_CIRCLE', 'OBS_60_CNT_SOCIAL_CIRCLE', 'DEF_60_CNT_SOCIAL_CIRCLE', 'DAYS_LAST_PHONE_CHANGE', 'FLAG_DOCUMENT_3', 'AMT_REQ_CREDIT_BUREAU_HOUR', 'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_WEEK', 'AMT_REQ_CREDIT_BUREAU_MON', 'AMT_REQ_CREDIT_BUREAU_QRT', 'AMT_REQ_CREDIT_BUREAU_YEAR', 'AMT_INCOME_RANGE', 'AMT_CREDIT_RANGE', 'AGE', 'AGE_GROUP', 'YEARS_EMPLOYED', 'EMPLOYMENT_YEAR'],  
      dtype='object')
```

```
In [105]: cols_for_correlation = ['NAME_CONTRACT_TYPE', 'CODE_GENDER', 'FLAG_OWN_CAR', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'OCCUPATION_TYPE', 'REGION_RATING_CLIENT_W_CITY', 'WEEKDAY_APPR_PROCESS_START', 'REG_REGION_NOT_LIVE_REGION', 'REG_REGION_NOT_WORK_REGION', 'REG_CITY_NOT_LIVE_CITY', 'REG_CITY_NOT_WORK_CITY', 'L', 'OBS_60_CNT_SOCIAL_CIRCLE', 'DEF_60_CNT_SOCIAL_CIRCLE', 'AMT_REQ_CREDIT_BUREAU_HOUR', 'AMT_REQ_CREDIT_BUREAU_DAY', 'AMT_REQ_CREDIT_BUREAU_MON', 'AMT_REQ_CREDIT_BUREAU_QRT', 'AMT_REQ_CREDIT_BUREAU_YEAR']
```

```
Repayer_df = applicationDF.loc[applicationDF['TARGET']==0, cols_for_correlation]
Defaulter_df = applicationDF.loc[applicationDF['TARGET']==1, cols_for_correlation]
```

```
In [106]: corr_repayer = Repayer_df.corr()

corr_repayer = corr_repayer.where(np.triu(np.ones(corr_repayer.shape), k=1).astype(bool))

corr_df_repayer = corr_repayer.unstack().reset_index()
corr_df_repayer.columns =[ 'VAR1', 'VAR2', 'Correlation']
corr_df_repayer.dropna(subset = ["Correlation"], inplace = True)

corr_df_repayer["Correlation"] = corr_df_repayer["Correlation"].abs()

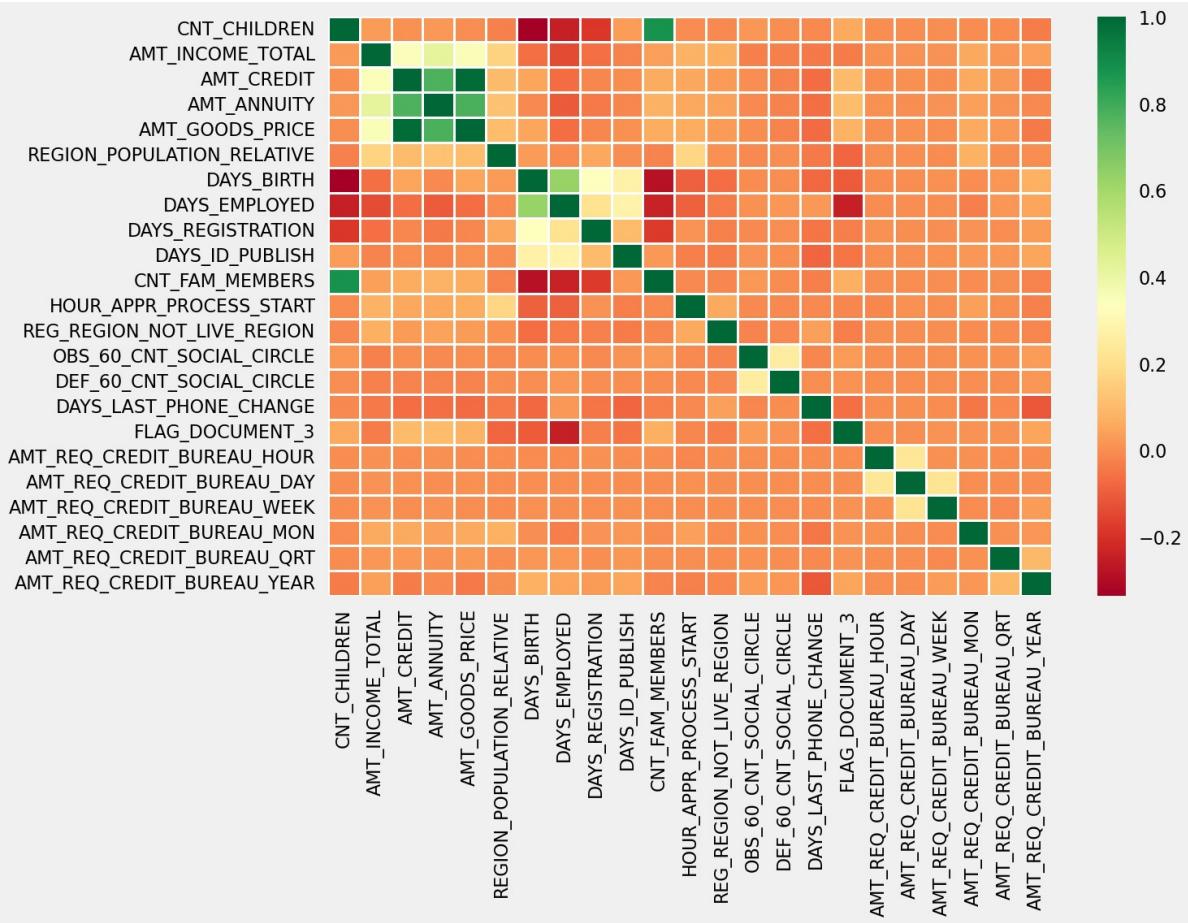
corr_df_repayer.sort_values(by='Correlation', ascending=False, inplace=True)

corr_df_repayer.head(10)
```

Out[106]:

	VAR1	VAR2	Correlation
94	AMT_GOODS_PRICE	AMT_CREDIT	0.987250
230	CNT_FAM_MEMBERS	CNT_CHILDREN	0.878571
95	AMT_GOODS_PRICE	AMT_ANNUITY	0.776686
71	AMT_ANNUITY	AMT_CREDIT	0.771309
167	DAYS_EMPLOYED	DAYS_BIRTH	0.626114
70	AMT_ANNUITY	AMT_INCOME_TOTAL	0.418953
93	AMT_GOODS_PRICE	AMT_INCOME_TOTAL	0.349462
47	AMT_CREDIT	AMT_INCOME_TOTAL	0.342799
138	DAYS_BIRTH	CNT_CHILDREN	0.336966
190	DAYS_REGISTRATION	DAYS_BIRTH	0.333151

```
In [107]: fig=plt.figure()
ax=sns.heatmap(Repayer_df.corr(), cmap ='RdYlGn', annot=False , linewidth =1)
```

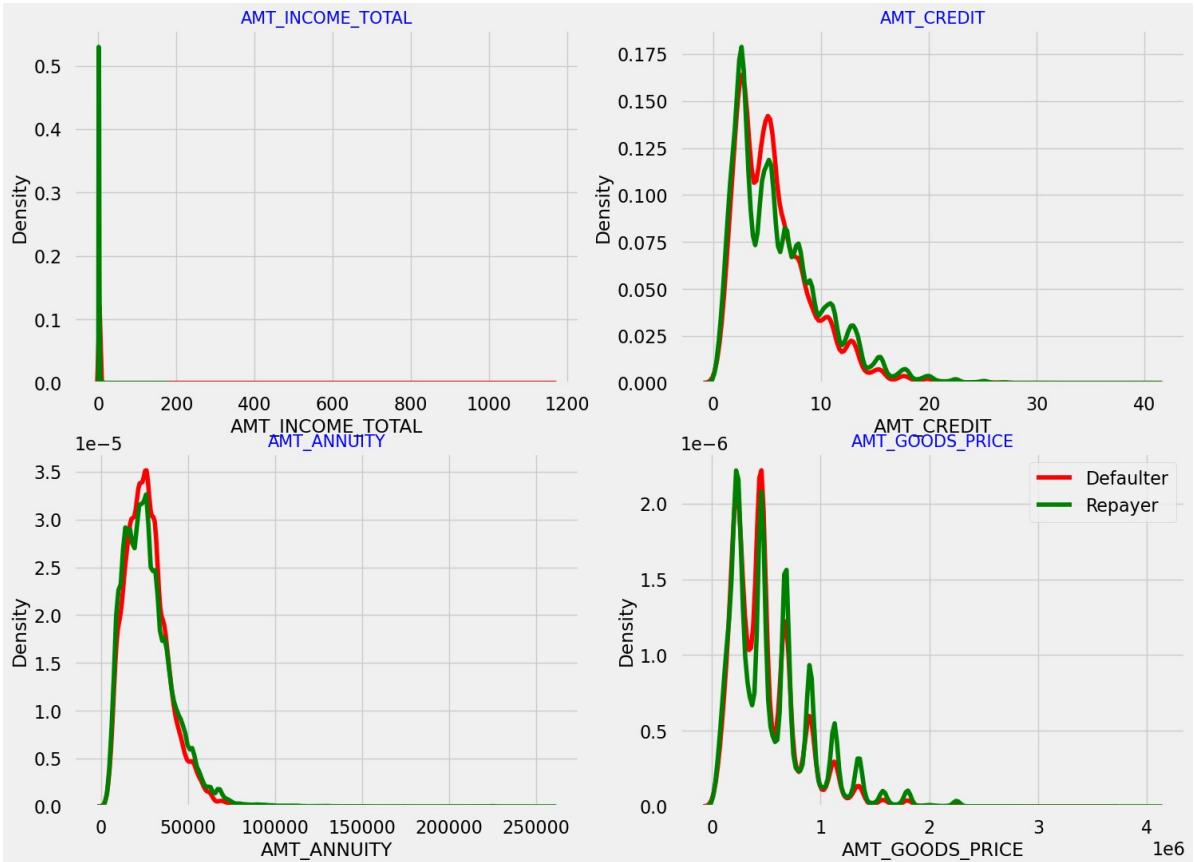


```
In [108]: amount = applicationDF[['AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE']]

fig = plt.figure(figsize=(16,12))

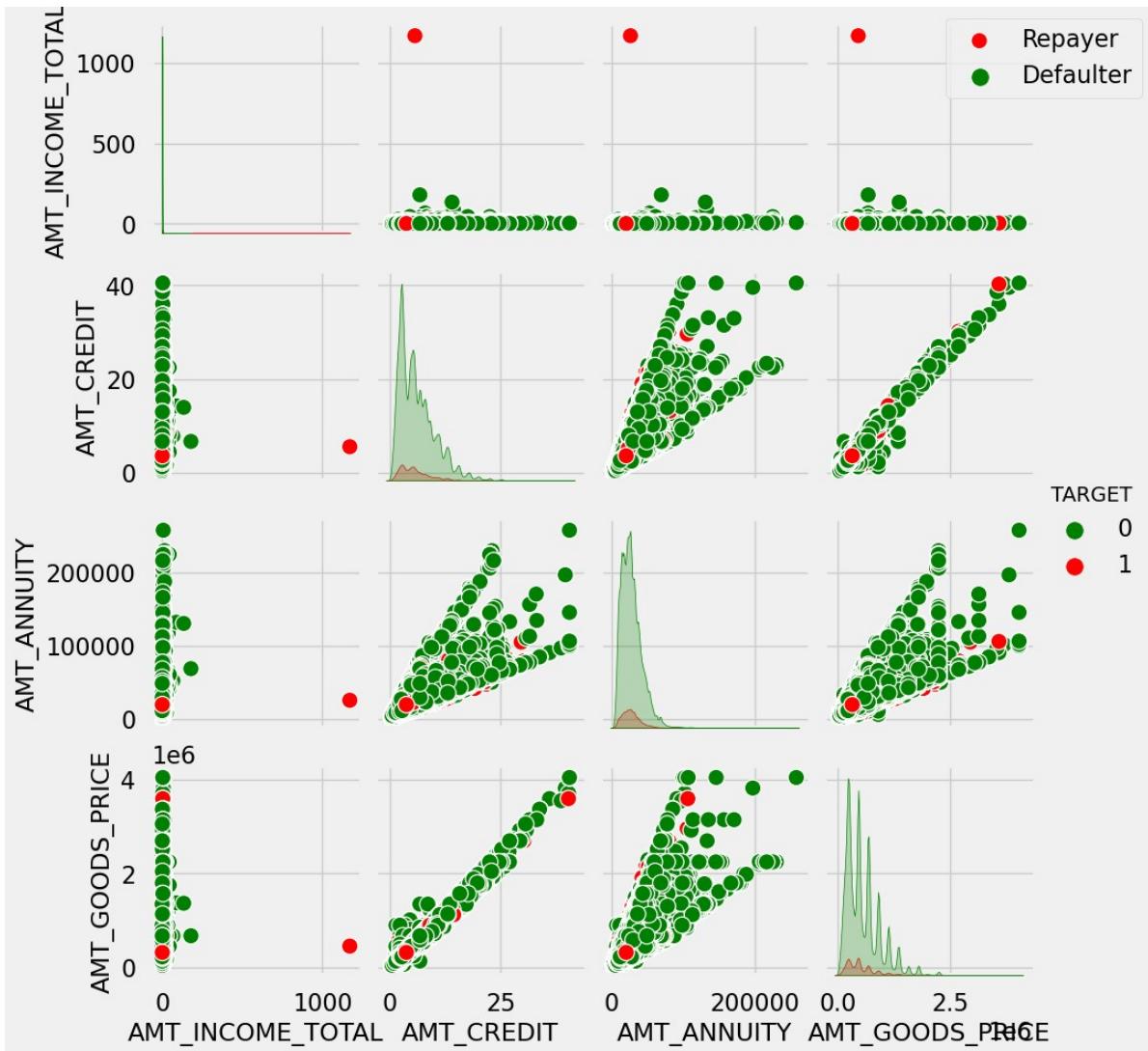
for i in enumerate(amount):
    plt.subplot(2,2,i[0]+1)
    sns.distplot(Defaulter_df[i[1]], hist=False, color='r', label = "Defaulter")
    sns.distplot(Repayer_df[i[1]], hist=False, color='g', label = "Repayer")
    plt.title(i[1], fontdict={'fontsize' : 15, 'fontweight' : 5, 'color' : 'Blue'})
    plt.legend()

plt.show()
```





```
In [110]: amount = applicationDF[['AMT_INCOME_TOTAL', 'AMT_CREDIT',  
                                'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'TARGET']]  
amount = amount[(amount["AMT_GOODS_PRICE"].notnull()) & (amount["AMT_ANNUITY"]  
ax = sns.pairplot(amount,hue="TARGET",palette=["g","r"])  
ax.fig.legend(labels=['Repayer', 'Defaulter'])  
plt.show()
```



```
In [111]: loan_process_df = pd.merge(applicationDF,previousDF,how = 'inner',on='SK_ID_CURR')
loan_process_df.head()
```

Out[111]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE_x	CODE_GENDER	FLAG_OWN_CAR	FLAG_
0	100002	1	Cash loans	M	N	
1	100003	0	Cash loans	F	N	
2	100003	0	Cash loans	F	N	
3	100003	0	Cash loans	F	N	
4	100004	0	Revolving loans	M		Y

```
In [112]: loan_process_df.shape
```

Out[112]: (1413701, 74)

```
In [113]: loan_process_df.size
```

Out[113]: 104613874

In [114]: loan\_process\_df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1413701 entries, 0 to 1413700
Data columns (total 74 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   SK_ID_CURR       1413701 non-null  int64  
 1   TARGET           1413701 non-null  int64  
 2   NAME_CONTRACT_TYPE_x  1413701 non-null  category
 3   CODE_GENDER      1413701 non-null  category
 4   FLAG_OWN_CAR    1413701 non-null  category
 5   FLAG_OWN_REALTY 1413701 non-null  category
 6   CNT_CHILDREN    1413701 non-null  int64  
 7   AMT_INCOME_TOTAL 1413701 non-null  float64
 8   AMT_CREDIT_x    1413701 non-null  float64
 9   AMT_ANNUITY_x   1413608 non-null  float64
 10  AMT_GOODS_PRICE_x 1412493 non-null  float64
 11  NAME_TYPE_SUITE 1413701 non-null  category
 12  NAME_INCOME_TYPE 1413701 non-null  category
 13  NAME_EDUCATION_TYPE 1413701 non-null  category
 14  NAME_FAMILY_STATUS 1413701 non-null  category
 15  NAME_HOUSING_TYPE 1413701 non-null  category
 16  REGION_POPULATION_RELATIVE 1413701 non-null  float64
 17  DAYS_BIRTH       1413701 non-null  int64  
 18  DAYS_EMPLOYED    1413701 non-null  int64  
 19  DAYS_REGISTRATION 1413701 non-null  float64
 20  DAYS_ID_PUBLISH 1413701 non-null  int64  
 21  OCCUPATION_TYPE 1413701 non-null  category
 22  CNT_FAM_MEMBERS 1413701 non-null  float64
 23  REGION_RATING_CLIENT 1413701 non-null  category
 24  REGION_RATING_CLIENT_W_CITY 1413701 non-null  category
 25  WEEKDAY_APPR_PROCESS_START 1413701 non-null  category
 26  HOUR_APPR_PROCESS_START 1413701 non-null  int64  
 27  REG_REGION_NOT_LIVE_REGION 1413701 non-null  int64  
 28  REG_REGION_NOT_WORK_REGION 1413701 non-null  category
 29  LIVE_REGION_NOT_WORK_REGION 1413701 non-null  category
 30  REG_CITY_NOT_LIVE_CITY 1413701 non-null  category
 31  REG_CITY_NOT_WORK_CITY 1413701 non-null  category
 32  LIVE_CITY_NOT_WORK_CITY 1413701 non-null  category
 33  ORGANIZATION_TYPE 1413701 non-null  category
 34  OBS_30_CNT_SOCIAL_CIRCLE 1410555 non-null  float64
 35  DEF_30_CNT_SOCIAL_CIRCLE 1410555 non-null  float64
 36  OBS_60_CNT_SOCIAL_CIRCLE 1410555 non-null  float64
 37  DEF_60_CNT_SOCIAL_CIRCLE 1410555 non-null  float64
 38  DAYS_LAST_PHONE_CHANGE 1413701 non-null  float64
 39  FLAG_DOCUMENT_3   1413701 non-null  int64  
 40  AMT_REQ_CREDIT_BUREAU_HOUR 1413701 non-null  float64
 41  AMT_REQ_CREDIT_BUREAU_DAY 1413701 non-null  float64
 42  AMT_REQ_CREDIT_BUREAU_WEEK 1413701 non-null  float64
 43  AMT_REQ_CREDIT_BUREAU_MON 1413701 non-null  float64
 44  AMT_REQ_CREDIT_BUREAU_QRT 1413701 non-null  float64
 45  AMT_REQ_CREDIT_BUREAU_YEAR 1413701 non-null  float64
 46  AMT_INCOME_RANGE   1413024 non-null  category
 47  AMT_CREDIT_RANGE   1413701 non-null  category
 48  AGE               1413701 non-null  int64
```

```

49 AGE_GROUP          1413701 non-null category
50 YEARS_EMPLOYED    1413701 non-null int64
51 EMPLOYMENT_YEAR   1032756 non-null category
52 SK_ID_PREV         1413701 non-null int64
53 NAME_CONTRACT_TYPE_y 1413701 non-null category
54 AMT_ANNUITY_y      1106483 non-null float64
55 AMT_APPLICATION    1413701 non-null float64
56 AMT_CREDIT_y        1413700 non-null float64
57 AMT_GOODS_PRICE_y   1413701 non-null float64
58 NAME_CASH_LOAN_PURPOSE 1413701 non-null category
59 NAME_CONTRACT_STATUS 1413701 non-null category
60 DAYS_DECISION       1413701 non-null int64
61 NAME_PAYMENT_TYPE   1413701 non-null category
62 CODE_REJECT_REASON  1413701 non-null category
63 NAME_CLIENT_TYPE    1413701 non-null category
64 NAME_GOODS_CATEGORY 1413701 non-null category
65 NAME_PORTFOLIO       1413701 non-null category
66 NAME_PRODUCT_TYPE   1413701 non-null category
67 CHANNEL_TYPE         1413701 non-null category
68 SELLERPLACE_AREA     1413701 non-null int64
69 NAME_SELLER_INDUSTRY 1413701 non-null category
70 CNT_PAYMENT          1413701 non-null float64
71 NAME_YIELD_GROUP     1413701 non-null category
72 PRODUCT_COMBINATION  1413388 non-null category
73 DAYS_DECISION_GROUP 1413701 non-null category
dtypes: category(37), float64(23), int64(14)
memory usage: 459.8 MB

```

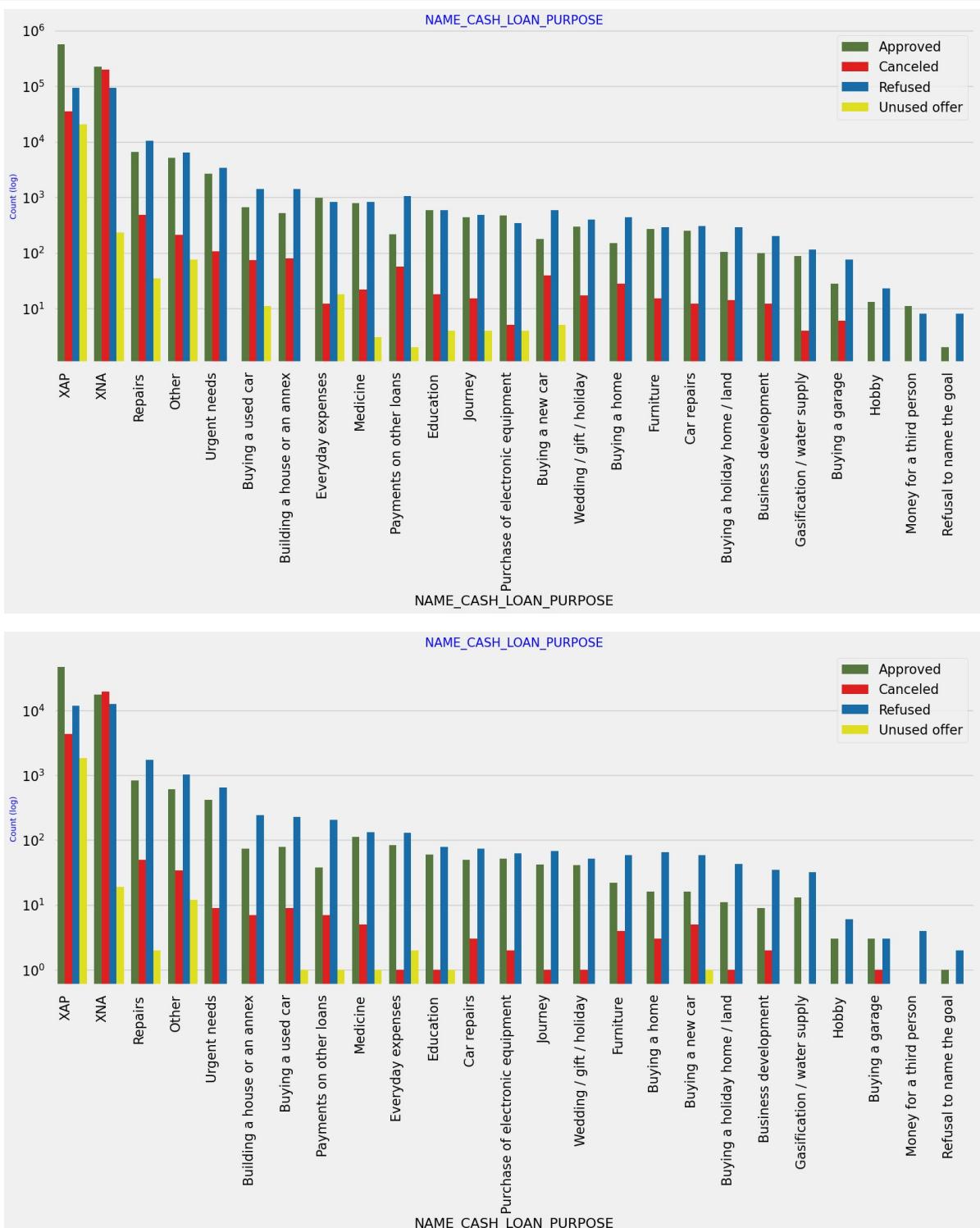
In [115]: `loan_process_df.describe()`

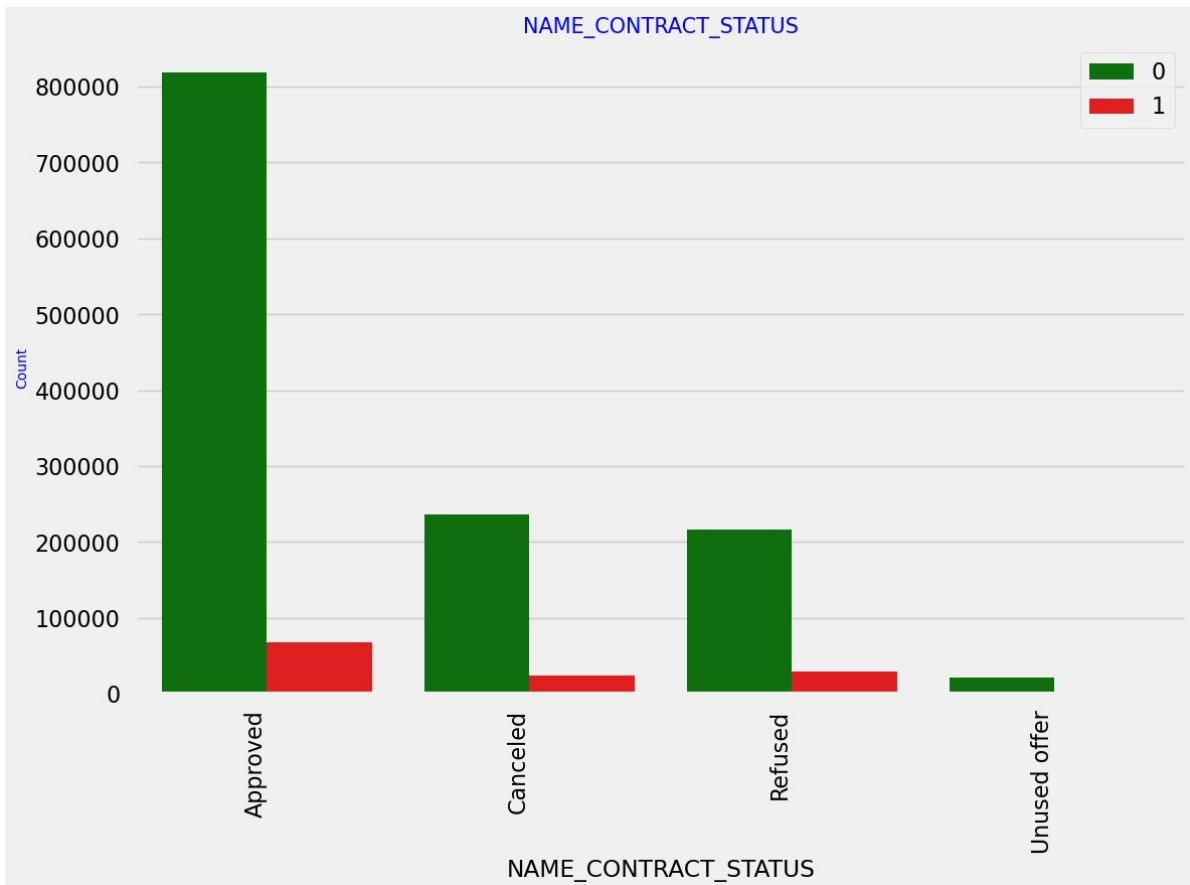
Out[115]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT_x	AMT
<b>count</b>	1.413701e+06	1.413701e+06	1.413701e+06	1.413701e+06	1.413701e+06	1
<b>mean</b>	2.784813e+05	8.655296e-02	4.048933e-01	1.733160e+00	5.875537e+00	2
<b>std</b>	1.028118e+05	2.811789e-01	7.173454e-01	1.985734e+00	3.849173e+00	1
<b>min</b>	1.000020e+05	0.000000e+00	0.000000e+00	2.565000e-01	4.500000e-01	1
<b>25%</b>	1.893640e+05	0.000000e+00	0.000000e+00	1.125000e+00	2.700000e+00	1
<b>50%</b>	2.789920e+05	0.000000e+00	0.000000e+00	1.575000e+00	5.084955e+00	2
<b>75%</b>	3.675560e+05	0.000000e+00	1.000000e+00	2.070000e+00	8.079840e+00	3
<b>max</b>	4.562550e+05	1.000000e+00	1.900000e+01	1.170000e+03	4.050000e+01	2

In [116]: `L0 = loan_process_df[loan_process_df['TARGET']==0]`  
`L1 = loan_process_df[loan_process_df['TARGET']==1]`

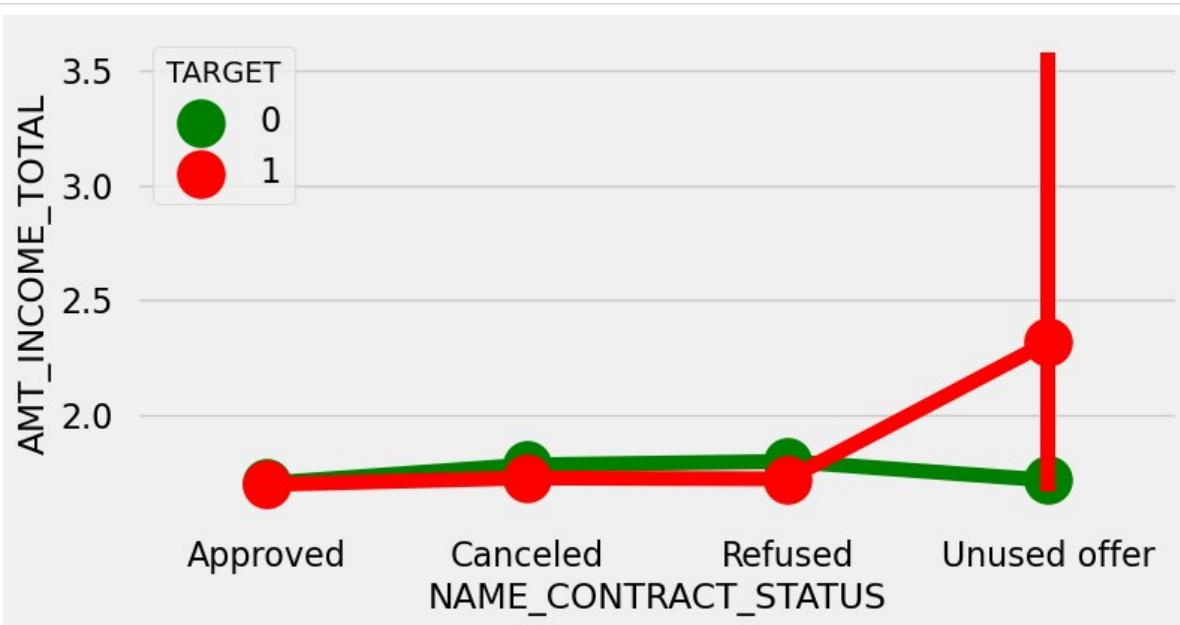
```
In [117]: univariate_merged("NAME_CASH_LOAN_PURPOSE", L0, "NAME_CONTRACT_STATUS", ["#548235"]
univariate_merged("NAME_CASH_LOAN_PURPOSE", L1, "NAME_CONTRACT_STATUS", ["#548235"]
```



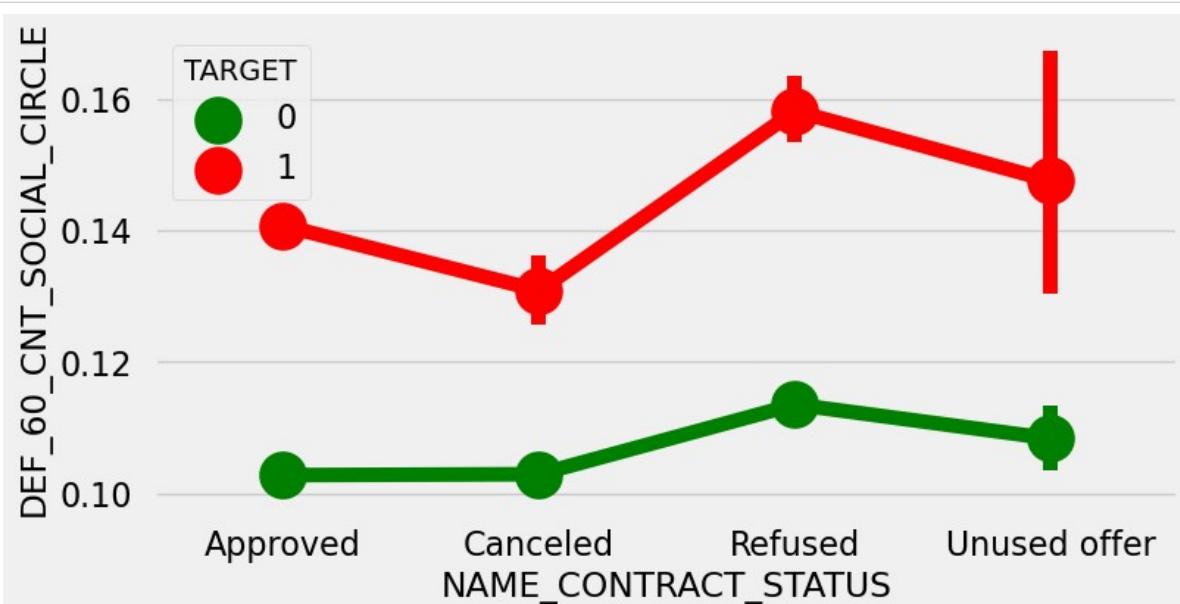


NAME_CONTRACT_STATUS	TARGET	Counts	Percentage
Approved	0	818856	92.41%
	1	67243	7.59%
Canceled	0	235641	90.83%
	1	23800	9.17%
Refused	0	215952	88.0%
	1	29438	12.0%
Unused offer	0	20892	91.75%
	1	1879	8.25%

```
In [119]: merged_pointplot("NAME_CONTRACT_STATUS", 'AMT_INCOME_TOTAL')
```



```
In [120]: merged_pointplot("NAME_CONTRACT_STATUS", 'DEF_60_CNT_SOCIAL_CIRCLE')
```



```
In [ ]:
```

In [ ]:

In [ ]:

In [ ]: