In [121]:

```
##Import Libraries.

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [122]:

```
##Remove Warnings In Kernel While Running A Cell

import warnings
warnings.filterwarnings('ignore')
```

Dataset 1 - "application_data.csv "

1. Reading And Understanding The Dataset.

In [3]:

```
## To Display All The Rows And Column

pd.set_option('display.max_rows', 300)
pd.set_option('display.max_columns', 300)

## Load The Dataset

df_application = pd.read_csv('data1/application_data.csv')
df_application.head()
```

Out[3]:

	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
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	

	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
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	

In [4]:

```
## Checking Number Of Rows And Column In Data Sheet

df_application.shape
```

Out[4]:

```
(307511, 122)
```

In [5]:

```
## Checking The Information In Each Row And Column For Analysis.

df_application.info("all")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 122 columns):
SK_ID_CURR                    int64
TARGET                        int64
NAME_CONTRACT_TYPE            object
CODE_GENDER                   object
FLAG_OWN_CAR                  object
FLAG_OWN_REALTY               object
CNT_CHILDREN                  int64
AMT_INCOME_TOTAL              float64
AMT_CREDIT                    float64
AMT_ANNUITY                   float64
AMT_GOODS_PRICE               float64
NAME_TYPE_SUITE               object
NAME_INCOME_TYPE              object
NAME_EDUCATION_TYPE           object
NAME_FAMILY_STATUS            object
NAME_HOUSING_TYPE             object
REGION_POPULATION_RELATIVE    float64
DAYS_BIRTH                    int64
DAYS_EMPLOYED                 int64
DAYS_REGISTRATION             float64
DAYS_ID_PUBLISH               int64
OWN_CAR_AGE                   float64
FLAG_MOBIL                    int64
FLAG_EMP_PHONE                int64
FLAG_WORK_PHONE               int64
FLAG_CONT_MOBILE              int64
FLAG_PHONE                    int64
```

```
FLAG_EMAIL                    int64
OCCUPATION_TYPE               object
CNT_FAM_MEMBERS               float64
REGION_RATING_CLIENT          int64
REGION_RATING_CLIENT_W_CITY   int64
WEEKDAY_APPR_PROCESS_START    object
HOUR_APPR_PROCESS_START       int64
REG_REGION_NOT_LIVE_REGION    int64
REG_REGION_NOT_WORK_REGION    int64
LIVE_REGION_NOT_WORK_REGION   int64
REG_CITY_NOT_LIVE_CITY        int64
REG_CITY_NOT_WORK_CITY        int64
LIVE_CITY_NOT_WORK_CITY       int64
ORGANIZATION_TYPE             object
EXT_SOURCE_1                  float64
EXT_SOURCE_2                  float64
EXT_SOURCE_3                  float64
APARTMENTS_AVG                float64
BASEMENTAREA_AVG              float64
YEARS_BEGINEXPLUATATION_AVG   float64
YEARS_BUILD_AVG               float64
COMMONAREA_AVG                float64
ELEVATORS_AVG                 float64
ENTRANCES_AVG                 float64
FLOORSMAX_AVG                 float64
FLOORSMIN_AVG                 float64
LANDAREA_AVG                  float64
LIVINGAPARTMENTS_AVG          float64
LIVINGAREA_AVG                float64
NONLIVINGAPARTMENTS_AVG       float64
NONLIVINGAREA_AVG             float64
APARTMENTS_MODE               float64
BASEMENTAREA_MODE             float64
YEARS_BEGINEXPLUATATION_MODE  float64
YEARS_BUILD_MODE              float64
COMMONAREA_MODE               float64
ELEVATORS_MODE                float64
ENTRANCES_MODE                float64
FLOORSMAX_MODE                float64
FLOORSMIN_MODE                float64
LANDAREA_MODE                 float64
LIVINGAPARTMENTS_MODE         float64
LIVINGAREA_MODE               float64
NONLIVINGAPARTMENTS_MODE      float64
NONLIVINGAREA_MODE            float64
APARTMENTS_MEDI               float64
BASEMENTAREA_MEDI             float64
YEARS_BEGINEXPLUATATION_MEDI  float64
YEARS_BUILD_MEDI              float64
COMMONAREA_MEDI               float64
ELEVATORS_MEDI                float64
```

```
ELEVATORS_MEDI                float64
ENTRANCES_MEDI                float64
FLOORSMAX_MEDI                float64
FLOORSMIN_MEDI                float64
LANDAREA_MEDI                 float64
LIVINGAPARTMENTS_MEDI         float64
LIVINGAREA_MEDI               float64
NONLIVINGAPARTMENTS_MEDI      float64
NONLIVINGAREA_MEDI            float64
FONDKAPREMONT_MODE             object
HOUSETYPE_MODE                 object
TOTALAREA_MODE                float64
WALLSMATERIAL_MODE             object
EMERGENCYSTATE_MODE            object
OBS_30_CNT_SOCIAL_CIRCLE      float64
DEF_30_CNT_SOCIAL_CIRCLE      float64
OBS_60_CNT_SOCIAL_CIRCLE      float64
DEF_60_CNT_SOCIAL_CIRCLE      float64
DAYS_LAST_PHONE_CHANGE        float64
FLAG_DOCUMENT_2                 int64
FLAG_DOCUMENT_3                 int64
FLAG_DOCUMENT_4                 int64
FLAG_DOCUMENT_5                 int64
FLAG_DOCUMENT_6                 int64
FLAG_DOCUMENT_7                 int64
FLAG_DOCUMENT_8                 int64
FLAG_DOCUMENT_9                 int64
FLAG_DOCUMENT_10                int64
FLAG_DOCUMENT_11                int64
FLAG_DOCUMENT_12                int64
FLAG_DOCUMENT_13                int64
FLAG_DOCUMENT_14                int64
FLAG_DOCUMENT_15                int64
FLAG_DOCUMENT_16                int64
FLAG_DOCUMENT_17                int64
FLAG_DOCUMENT_18                int64
FLAG_DOCUMENT_19                int64
FLAG_DOCUMENT_20                int64
FLAG_DOCUMENT_21                int64
AMT_REQ_CREDIT_BUREAU_HOUR    float64
AMT_REQ_CREDIT_BUREAU_DAY     float64
AMT_REQ_CREDIT_BUREAU_WEEK    float64
AMT_REQ_CREDIT_BUREAU_MON     float64
AMT_REQ_CREDIT_BUREAU_QRT     float64
AMT_REQ_CREDIT_BUREAU_YEAR    float64
dtypes: float64(65), int64(41), object(16)
memory usage: 267.5+ MB
```

In [6]:

```
## Checking The Numaric Values Of Data Frame
```

```
df_application.describe()
```

Out[6]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_POPULATION_RELATIVE	DAYS_BIRTH	DAY
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e+05	307511.000000	307511.000000	
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e+05	0.020868	-16036.995067	
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e+05	0.013831	4363.988632	
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e+04	0.000290	-25229.000000	
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e+05	0.010006	-19682.000000	
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e+05	0.018850	-15750.000000	
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e+05	0.028663	-12413.000000	
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e+06	0.072508	-7489.000000	

Dataset 2 - "previous_application.csv"

2.1 Reading And Understanding The Dataset.

In [7]:

```
## To Display All The Rows And Column

pd.set_option('display.max_rows', 300)
pd.set_option('display.max_columns', 300)

## Load The Dataset

df_previous_application = pd.read_csv('data1/previous_application.csv')
df_previous_application.head()
```

Out[7]:

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_ST
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	0.0	17145.0	SATUI
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	NaN	607500.0	THURS
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	NaN	112500.0	TUES

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_S
3	2315218	175704	Cash loans	47041.335	450000.0	470790.0	NaN	450000.0	MON
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	NaN	337500.0	THURS

In [8]:

```
## Checking Number Of Rows And Column In Data Sheet

df_previous_application.shape
```

Out[8]:

```
(1670214, 37)
```

In [10]:

```
## Checking The Information In Each Row And Column For Analysis.

df_previous_application.info("all")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1670214 entries, 0 to 1670213
Data columns (total 37 columns):
SK_ID_PREV                    1670214 non-null int64
SK_ID_CURR                    1670214 non-null int64
NAME_CONTRACT_TYPE            1670214 non-null object
AMT_ANNUITY                   1297979 non-null float64
AMT_APPLICATION               1670214 non-null float64
AMT_CREDIT                    1670213 non-null float64
AMT_DOWN_PAYMENT              774370 non-null float64
AMT_GOODS_PRICE               1284699 non-null float64
WEEKDAY_APPR_PROCESS_START    1670214 non-null object
HOUR_APPR_PROCESS_START       1670214 non-null int64
FLAG_LAST_APPL_PER_CONTRACT   1670214 non-null object
NFLAG_LAST_APPL_IN_DAY        1670214 non-null int64
RATE_DOWN_PAYMENT             774370 non-null float64
RATE_INTEREST_PRIMARY         5951 non-null float64
RATE_INTEREST_PRIVILEGED      5951 non-null float64
NAME_CASH_LOAN_PURPOSE        1670214 non-null object
NAME_CONTRACT_STATUS          1670214 non-null object
DAYS_DECISION                 1670214 non-null int64
NAME_PAYMENT_TYPE             1670214 non-null object
CODE_REJECT_REASON            1670214 non-null object
NAME_TYPE_SUITE               849809 non-null object
NAME_CLIENT_TYPE              1670214 non-null object
NAME_GOODS_CATEGORY           1670214 non-null object
NAME_PORTFOLIO                1670214 non-null object
NAME_PRODUCT_TYPE             1670214 non-null object
CHANNEL_TYPE                  1670214 non-null object
```

```
CHANNEL_TYPE                 1670214 non-null object
SELLERPLACE_AREA             1670214 non-null int64
NAME_SELLER_INDUSTRY         1670214 non-null object
CNT_PAYMENT                  1297984 non-null float64
NAME_YIELD_GROUP             1670214 non-null object
PRODUCT_COMBINATION          1669868 non-null object
DAYS_FIRST_DRAWING           997149 non-null float64
DAYS_FIRST_DUE               997149 non-null float64
DAYS_LAST_DUE_1ST_VERSION    997149 non-null float64
DAYS_LAST_DUE                997149 non-null float64
DAYS_TERMINATION             997149 non-null float64
NFLAG_INSURED_ON_APPROVAL    997149 non-null float64
dtypes: float64(15), int64(6), object(16)
memory usage: 369.5+ MB
```

In [9]:

```
## Checking The Numaric Values Of Data Frame

df_previous_application.describe()
```

Out[9]:

	SK_ID_PREV	SK_ID_CURR	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	HOUR_APPR_PROCESS_START	NFLAG_LAST_APPI
count	1.670214e+06	1.670214e+06	1.297979e+06	1.670214e+06	1.670213e+06	7.743700e+05	1.284699e+06	1.670214e+06	1.67
mean	1.923089e+06	2.783572e+05	1.595512e+04	1.752339e+05	1.961140e+05	6.697402e+03	2.278473e+05	1.248418e+01	9.96
std	5.325980e+05	1.028148e+05	1.478214e+04	2.927798e+05	3.185746e+05	2.092150e+04	3.153966e+05	3.334028e+00	5.93
min	1.000001e+06	1.000010e+05	0.000000e+00	0.000000e+00	0.000000e+00	-9.000000e-01	0.000000e+00	0.000000e+00	0.00
25%	1.461857e+06	1.893290e+05	6.321780e+03	1.872000e+04	2.416050e+04	0.000000e+00	5.084100e+04	1.000000e+01	1.00
50%	1.923110e+06	2.787145e+05	1.125000e+04	7.104600e+04	8.054100e+04	1.638000e+03	1.123200e+05	1.200000e+01	1.00
75%	2.384280e+06	3.675140e+05	2.065842e+04	1.803600e+05	2.164185e+05	7.740000e+03	2.340000e+05	1.500000e+01	1.00
max	2.845382e+06	4.562550e+05	4.180581e+05	6.905160e+06	6.905160e+06	3.060045e+06	6.905160e+06	2.300000e+01	1.00

3 Cleaning, Manipulation, Outliers Treatment

3.1 Checking Null Values. Dataset "application_data.csv"

In [7]:

```
## Checking Null Values In Dataset
```

```
## Load The Dataset

df_application = pd.read_csv('data1/application_data.csv')
df_application.head()
```

Out[7]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...
0	**100002**	**1**	**Cash loans**	**M**	**N**	**Y**	**0**	**202500.0**	**406597.5**	**24700.5**	**...**
1	**100003**	**0**	**Cash loans**	**F**	**N**	**N**	**0**	**270000.0**	**1293502.5**	**35698.5**	**...**
2	**100004**	**0**	**Revolving loans**	**M**	**Y**	**Y**	**0**	**67500.0**	**135000.0**	**6750.0**	**...**
3	**100006**	**0**	**Cash loans**	**F**	**N**	**Y**	**0**	**135000.0**	**312682.5**	**29686.5**	**...**
4	**100007**	**0**	**Cash loans**	**M**	**N**	**Y**	**0**	**121500.0**	**513000.0**	**21865.5**	**...**

5 rows × 122 columns

In [8]:

```
df_application.isnull().sum()
```

Out[8]:

```
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
                                 ...
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
Length: 122, dtype: int64
```

Cleaning The Missing Data More Then 30% Null Values

In [10]:

```
emptycol=df_application.isnull().sum()
emptycol=emptycol[emptycol.values>(0.3*len(emptycol))]
len(emptycol)
```

Out[10]:

```
64
```

Removing Those 64 Columns Having Null Values Greater Then 30%

In [11]:

```
emptycol = list(emptycol[emptycol.values>=0.3].index)
df_application.drop(labels=emptycol,axis=1,inplace=True)
print(len(emptycol))
```

64

Now Checking The Column Having Lesser Null Percentage Values

In [12]:

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

Out[12]:

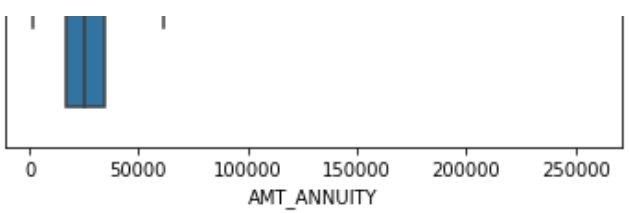
```
SK_ID_CURR                    0.000000
TARGET                        0.000000
NAME_CONTRACT_TYPE            0.000000
CODE_GENDER                   0.000000
FLAG_OWN_CAR                  0.000000
FLAG_OWN_REALTY               0.000000
CNT_CHILDREN                  0.000000
AMT_INCOME_TOTAL              0.000000
AMT_CREDIT                    0.000000
AMT_ANNUITY                   0.003902
NAME_INCOME_TYPE              0.000000
NAME_EDUCATION_TYPE           0.000000
NAME_FAMILY_STATUS            0.000000
NAME_HOUSING_TYPE             0.000000
REGION_POPULATION_RELATIVE    0.000000
DAYS_BIRTH                    0.000000
DAYS_EMPLOYED                 0.000000
DAYS_REGISTRATION             0.000000
DAYS_ID_PUBLISH               0.000000
FLAG_MOBIL                    0.000000
FLAG_EMP_PHONE                0.000000
FLAG_WORK_PHONE               0.000000
FLAG_CONT_MOBILE              0.000000
FLAG_PHONE                    0.000000
FLAG_EMAIL                    0.000000
CNT_FAM_MEMBERS               0.000650
REGION_RATING_CLIENT          0.000000
REGION_RATING_CLIENT_W_CITY   0.000000
WEEKDAY_APPR_PROCESS_START    0.000000
```

```
HOUR_APPR_PROCESS_START         0.000000
REG_REGION_NOT_LIVE_REGION      0.000000
REG_REGION_NOT_WORK_REGION      0.000000
LIVE_REGION_NOT_WORK_REGION     0.000000
REG_CITY_NOT_LIVE_CITY          0.000000
REG_CITY_NOT_WORK_CITY          0.000000
LIVE_CITY_NOT_WORK_CITY         0.000000
ORGANIZATION_TYPE               0.000000
DAYS_LAST_PHONE_CHANGE          0.000325
FLAG_DOCUMENT_2                 0.000000
FLAG_DOCUMENT_3                 0.000000
FLAG_DOCUMENT_4                 0.000000
FLAG_DOCUMENT_5                 0.000000
FLAG_DOCUMENT_6                 0.000000
FLAG_DOCUMENT_7                 0.000000
FLAG_DOCUMENT_8                 0.000000
FLAG_DOCUMENT_9                 0.000000
FLAG_DOCUMENT_10                0.000000
FLAG_DOCUMENT_11                0.000000
FLAG_DOCUMENT_12                0.000000
FLAG_DOCUMENT_13                0.000000
FLAG_DOCUMENT_14                0.000000
FLAG_DOCUMENT_15                0.000000
FLAG_DOCUMENT_16                0.000000
FLAG_DOCUMENT_17                0.000000
FLAG_DOCUMENT_18                0.000000
FLAG_DOCUMENT_19                0.000000
FLAG_DOCUMENT_20                0.000000
FLAG_DOCUMENT_21                0.000000
dtype: float64
```

Imputing AMT_ANNUITY Column Missing Values & Outliers With Median Value

In [13]:

```
### Ploting a Box plot for AMT_ANNUITY to check the outliers

sns.boxplot(df_application.AMT_ANNUITY)
plt.show()
```



In [14]:

```
#### Filling Missing Values With Median.

values =df_application['AMT_ANNUITY'].median()

df_application.loc[df_application['AMT_ANNUITY'].isnull(),'AMT_ANNUITY']=values
```

Serching Column For Null Values

In [15]:

```
df_application.isnull().sum()
```

Out[15]:

```
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                    0
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
FLAG_MOBIL                     0
FLAG_EMP_PHONE                 0
FLAG_WORK_PHONE                0
FLAG_CONT_MOBILE               0
FLAG_PHONE                     0
FLAG_EMAIL                     0
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
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
dtype: int64
```

Removing rows having null values greater than or equal to 30%

In [16]:

```
emptyrow=df_application.isnull().sum(axis=1)
emptyrow=list(emptyrow[emptyrow.values>=0.3*len(df_application)].index)
df_application.drop(labels=emptyrow,axis=0,inplace=True)
print(len(emptyrow))
```

```
0
```

Finding Categorical Columns Having 'XNA' values

In [17]:

In [17]:

```
df_application[df_application['CODE_GENDER']=='XNA'].shape
```

Out[17]:

```
(4, 58)
```

For Organization Column

In [18]:

```
df_application[df_application['ORGANIZATION_TYPE']=='XNA'].shape
```

Out[18]:

```
(55374, 58)
```

Describing Gender Column To Check Male & Females

In [19]:

```
df_application['CODE_GENDER'].value_counts()
```

Out[19]:

```
F      202448
M      105059
XNA         4
Name: CODE_GENDER, dtype: int64
```

Describing Organization Column

In [20]:

```
df_application['ORGANIZATION_TYPE'].describe()
```

Out[20]:

```
count                    307511
unique                       58
top       Business Entity Type 3
freq                      67992
Name: ORGANIZATION_TYPE, dtype: object
```

Dropping Total Number Of Rows In A Organization Column

In [21]:

```
df_application=df_application.drop(df_application.loc[df_application['ORGANIZATION_TYPE']=='XNA'].index)
df_application[df_application['ORGANIZATION_TYPE']=='XNA'].shape
```

Out[21]:

```
(0, 58)
```

Dividing The Dataset Into Target=1 & Target=0

In [22]:

```
target0_df_application=df_application.loc[df_application["TARGET"]==0]
target1_df_application=df_application.loc[df_application["TARGET"]==1]
```

Calculating Imbalence Percentage

In [23]:

```
round(len(target0_df_application)/len(target1_df_application),2)
```

Out[23]:

```
10.55
```

4 Analysis Of Variables

Performing Univariate Analysis

4.1 Dataset for "application_data.csv"

In [33]:

```
df_application.head()
```

Out[33]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	...
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	...
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	...
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	...

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	...

5 rows × 58 columns

In [17]:

```
Numarical_Data = ['AMT_ANNUITY','AMT_GOODS_PRICE','AGE_IN_YEARS','EMPLOYMENT_YEARS','AMT_INCOME_TOTAL_in_lakhs',
                  'AMT_CREDIT_in_lakhs','CNT_FAM_MEMBERS','Credit_Ratio']
```

In [18]:

```
Categorical_Data = ['FLAG_OWN_CAR','FLAG_OWN_REALTY','NAME_TYPE_SUITE','NAME_INCOME_TYPE','NAME_EDUCATION_TYPE',
                    'NAME_FAMILY_STATUS','NAME_HOUSING_TYPE','OCCUPATION_TYPE','AGE_IN_YEARS_RANGE','EMPLOYMENT_YEARS_RANGE','AMT
_CREDIT_in_lakhs_Range','AMT_INCOME_TOTAL_RANGE']
```

In [19]:

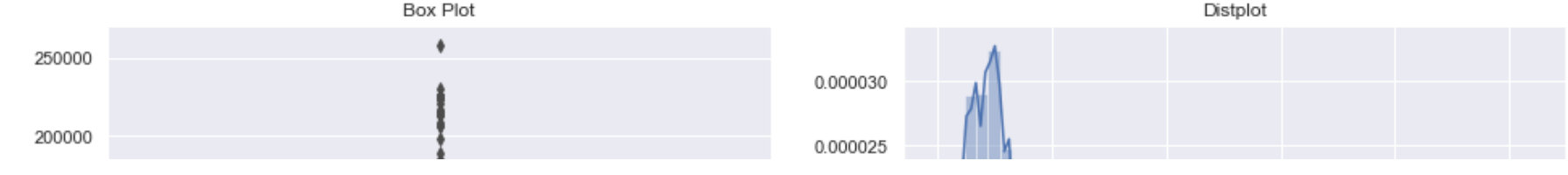
```
def Uni_Analysis_Numarical(dataframe, column):
    sns.set(style='darkgrid')
    plt.figure(figsize=(25,5))

    plt.subplot(1, 3, 1)
    sns.boxplot(data=dataframe, x=column, orient='v').set(title='Box Plot')

    plt.subplot(1,3,2)
    sns.distplot(dataframe[column].dropna()).set(title='Distplot')
    plt.show()
```

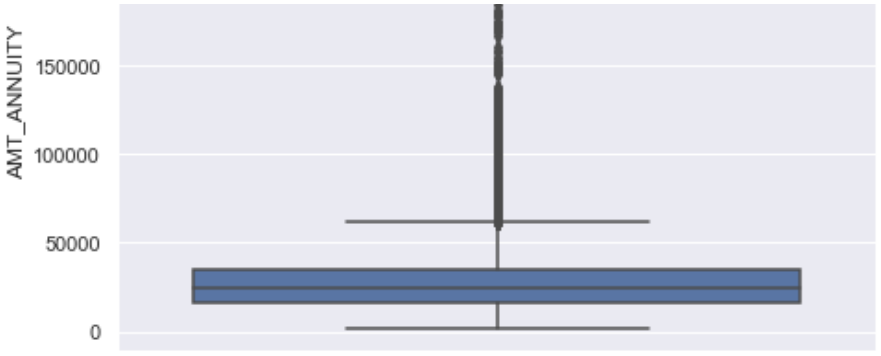
In [20]:

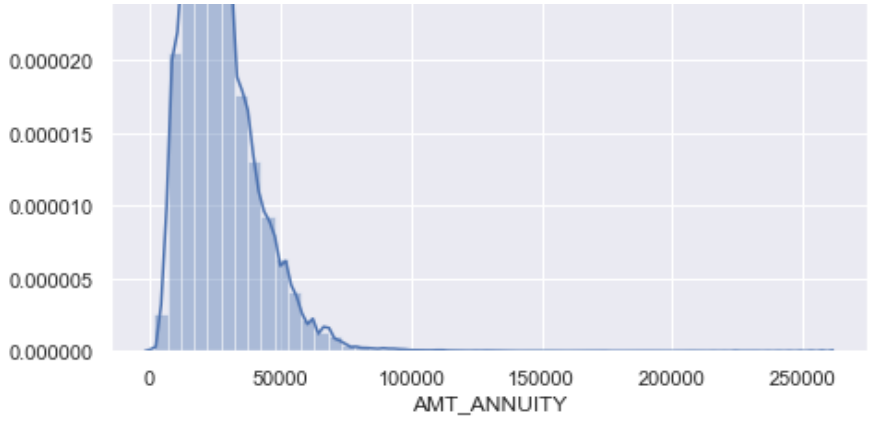
```
def Uni_Analysis_Categorcal(dataframe, column):
    sns.set(style='darkgrid')
    plt.figure(figsize = [12,5])
    dataframe[column].value_counts().plot.barh(width = 0.8)
    plt.title(column)
    plt.show()
```

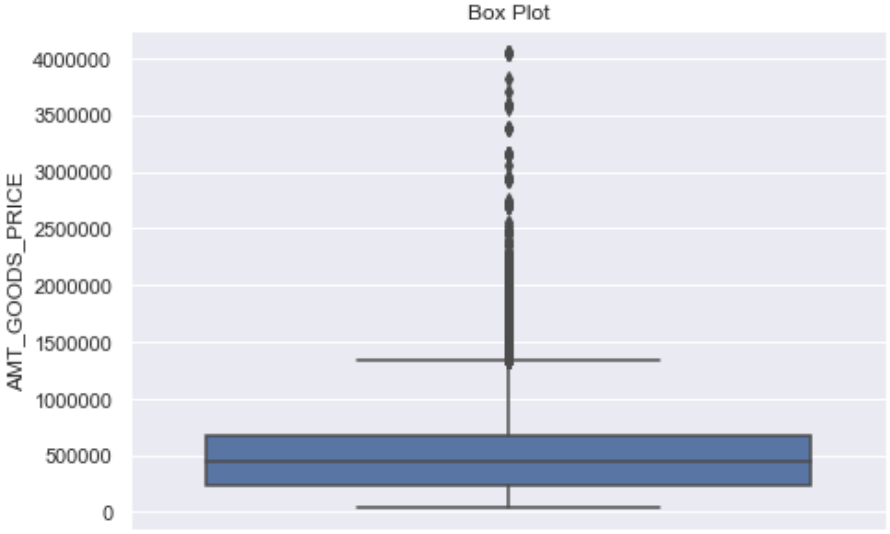
In [21]:

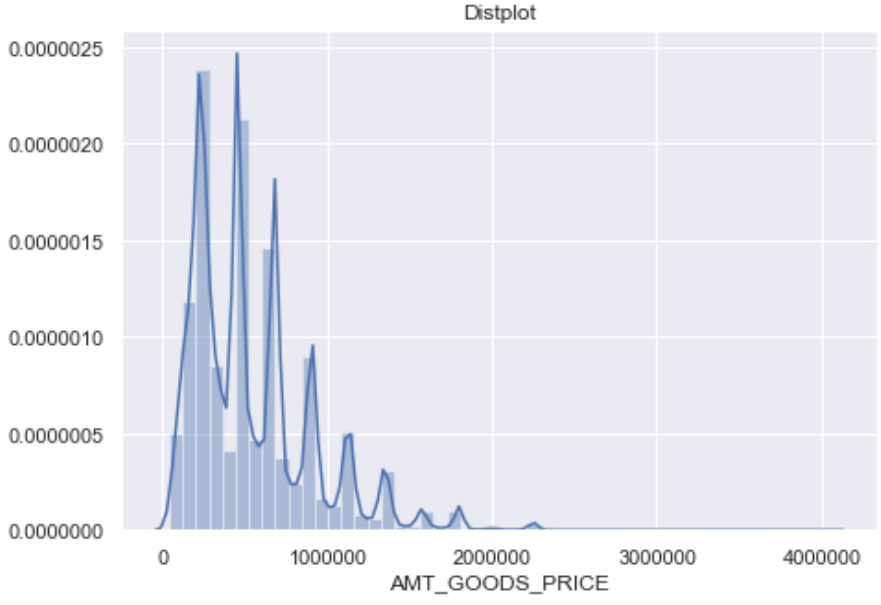
```
for i in Numarical_Data:
    Uni_Analysis_Numarical(df_application,i)
```





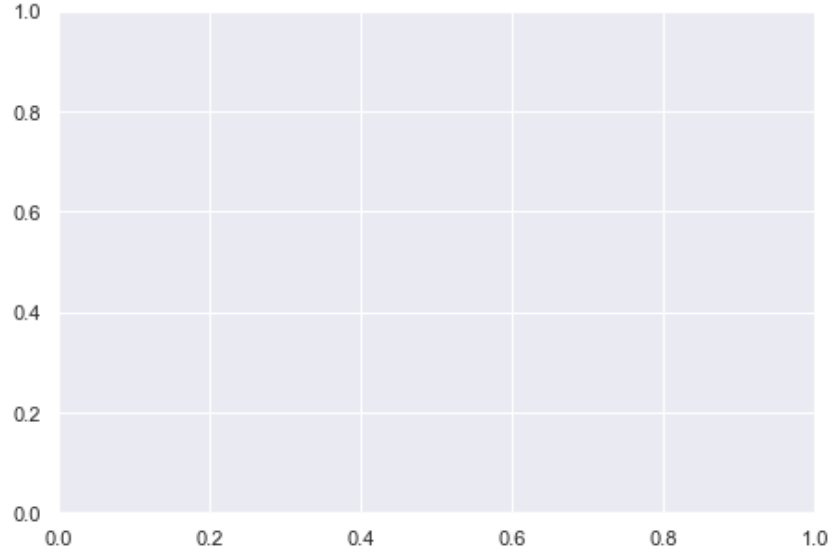






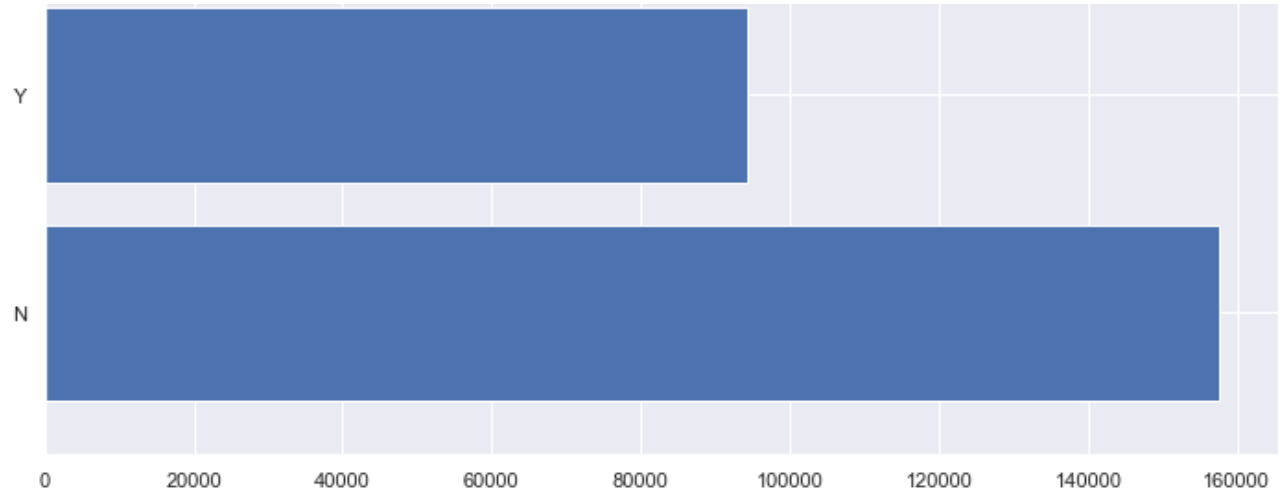

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-21-fcda5c95df8b> in <module>
      1 for i in Numarical_Data:
----> 2     Uni_Analysis_Numarical(df_application,i)

<ipython-input-19-85eab18cd2c5> in Uni_Analysis_Numarical(dataframe, column)
      4
      5     plt.subplot(1, 3, 1)
----> 6     sns.boxplot(data=dataframe, x=column, orient='v').set(title='Box Plot')
      7
      8     plt.subplot(1,3,2)

~\Anaconda3\lib\site-packages\seaborn\categorical.py in boxplot(x, y, hue, data, order, hue_order, orient, color, palette, saturat
ion, width, dodge, fliersize, linewidth, whis, notch, ax, **kwargs)
```

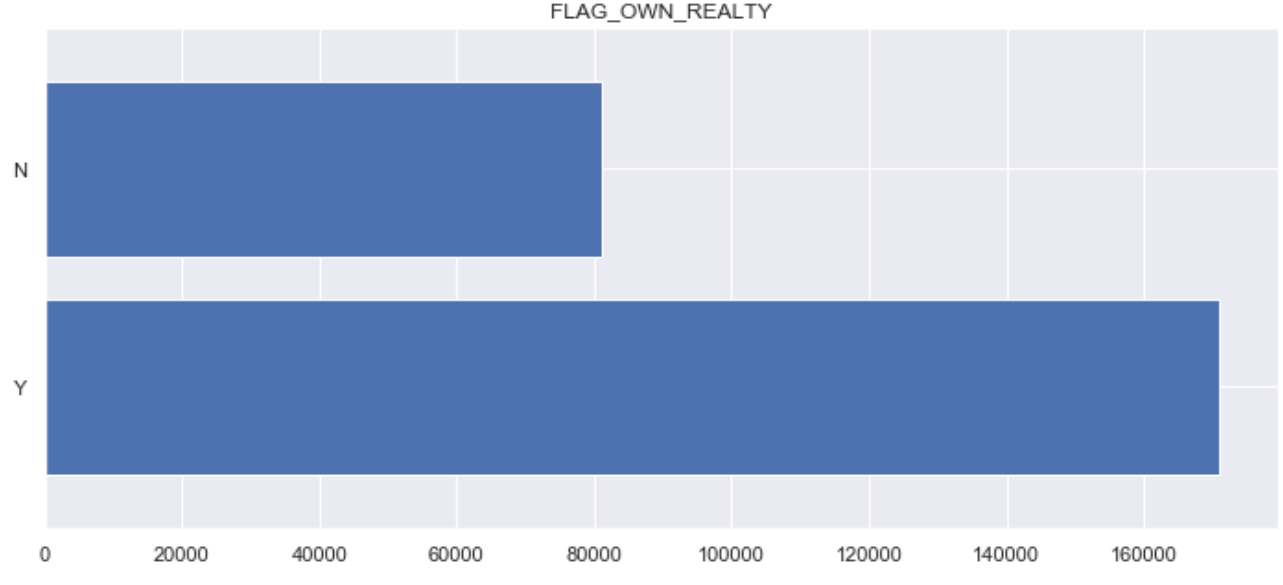
```
   2229     plotter = _BoxPlotter(x, y, hue, data, order, hue_order,
   2230                           orient, color, palette, saturation,
-> 2231                           width, dodge, fliersize, linewidth)
   2232
   2233     if ax is None:

~\Anaconda3\lib\site-packages\seaborn\categorical.py in __init__(self, x, y, hue, data, order, hue_order, orient, color, palette,
saturation, width, dodge, fliersize, linewidth)
    444                  width, dodge, fliersize, linewidth):
    445
--> 446         self.establish_variables(x, y, hue, data, orient, order, hue_order)
    447         self.establish_colors(color, palette, saturation)
    448

~\Anaconda3\lib\site-packages\seaborn\categorical.py in establish_variables(self, x, y, hue, data, orient, order, hue_order, units
)
    153                 if isinstance(input, string_types):
    154                     err = "Could not interpret input '{}'".format(input)
--> 155                     raise ValueError(err)
    156
    157             # Figure out the plotting orientation

ValueError: Could not interpret input 'AGE_IN_YEARS'
```



In [39]:

```
for i in Categorical_Data:
    Uni_Analysis_Categorcal(df_application,i)
```



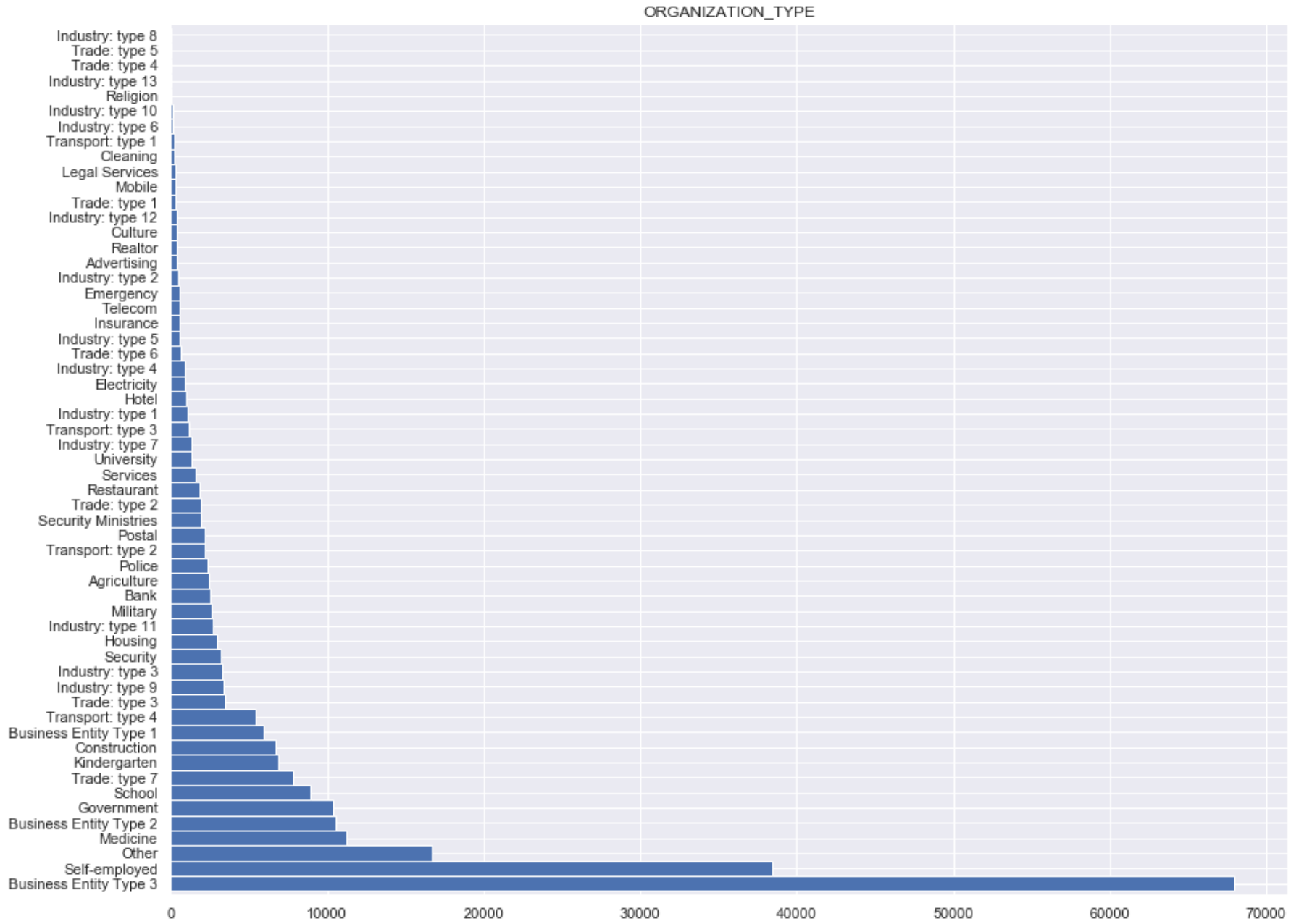



```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2656             try:
-> 2657                 return self._engine.get_loc(key)
   2658             except KeyError:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()
```

```
KeyError: 'NAME_TYPE_SUITE'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
<ipython-input-39-aafa93f961aa> in <module>
      1 for i in Categorical_Data:
----> 2     Uni_Analysis_Categorcal(df_application,i)

<ipython-input-37-6c981fbf8f56> in Uni_Analysis_Categorcal(dataframe, column)
      2     sns.set(style='darkgrid')
      3     plt.figure(figsize = [12,5])
----> 4     dataframe[column].value_counts().plot.barh(width = 0.8)
      5     plt.title(column)
      6     plt.show()

~\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
   2925             if self.columns.nlevels > 1:
   2926                 return self._getitem_multilevel(key)
-> 2927             indexer = self.columns.get_loc(key)
   2928             if is_integer(indexer):
   2929                 indexer = [indexer]

~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2657                 return self._engine.get_loc(key)
   2658             except KeyError:
-> 2659                 return self._engine.get_loc(self._maybe_cast_indexer(key))
   2660         indexer = self.get_indexer([key], method=method, tolerance=tolerance)
   2661         if indexer.ndim > 1 or indexer.size > 1:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'NAME_TYPE_SUITE'

<Figure size 864x360 with 0 Axes>
```

In [45]:

```
sns.set(style='darkgrid')
    plt.figure(figsize = [15,12])
    df_application['ORGANIZATION_TYPE'].value_counts().plot.barh(width = 1)
    plt.title('ORGANIZATION_TYPE')
    plt.show()
```



5.2 Dataset for "previous_application.csv"

In [8]:

```
df_previous_application.head()
```

Out[8]:

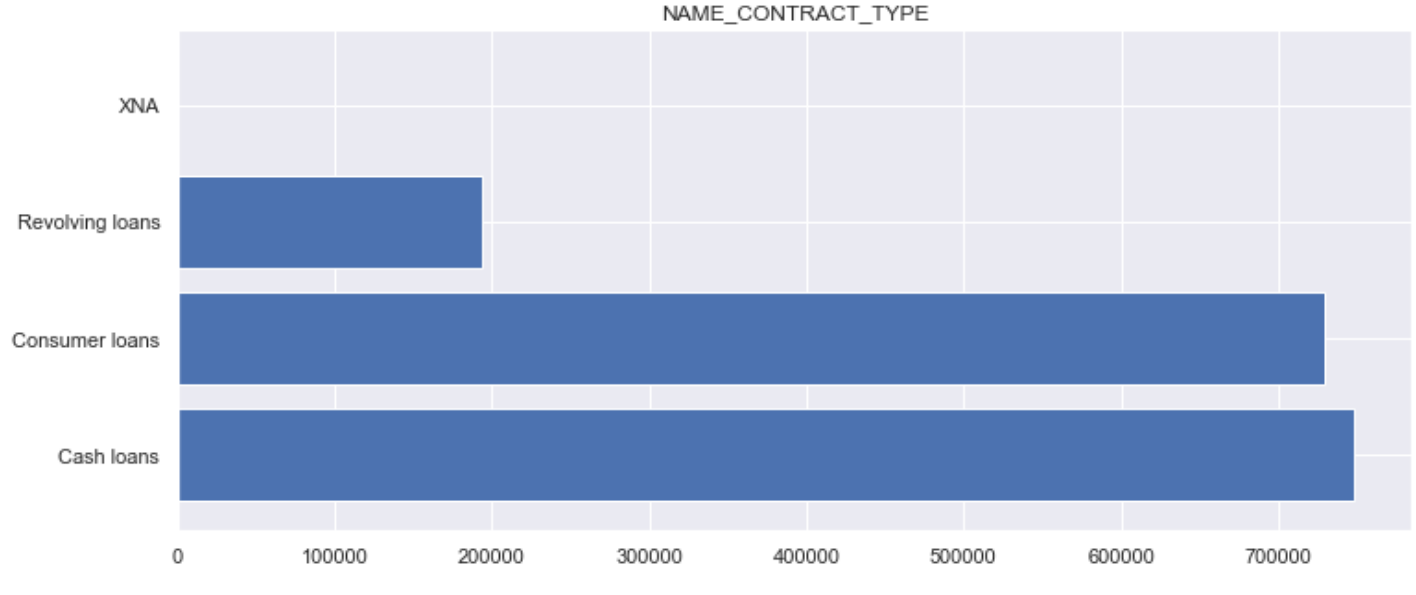
	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_ST
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	0.0	17145.0	SATUI
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	NaN	607500.0	THURS
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	NaN	112500.0	TUES
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	NaN	450000.0	MON
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	NaN	337500.0	THURS

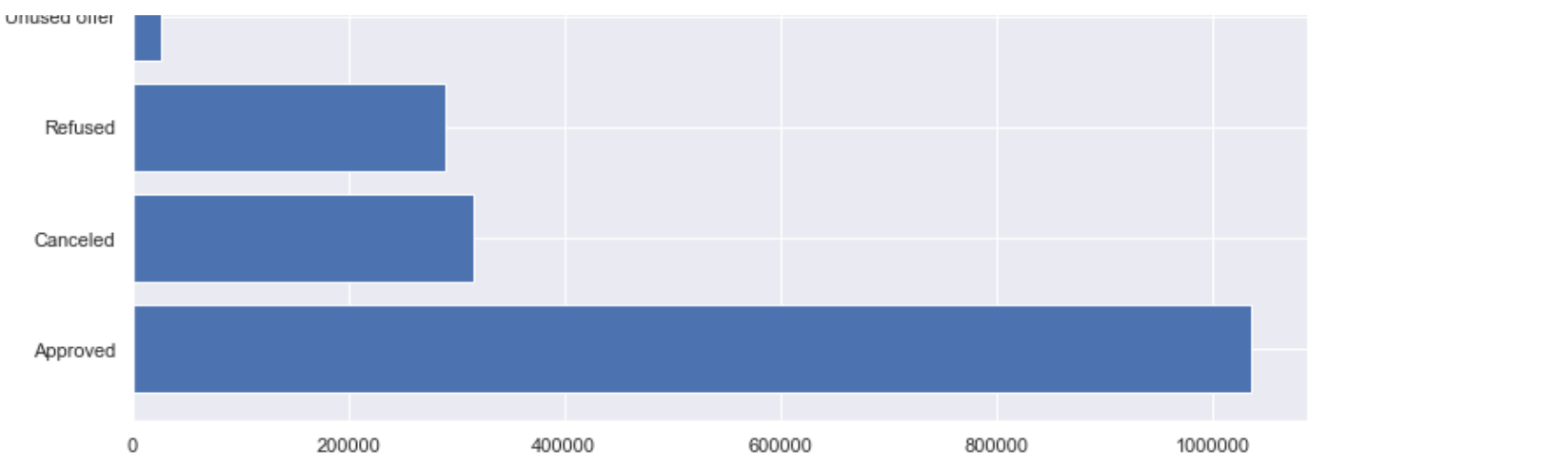
In [25]:

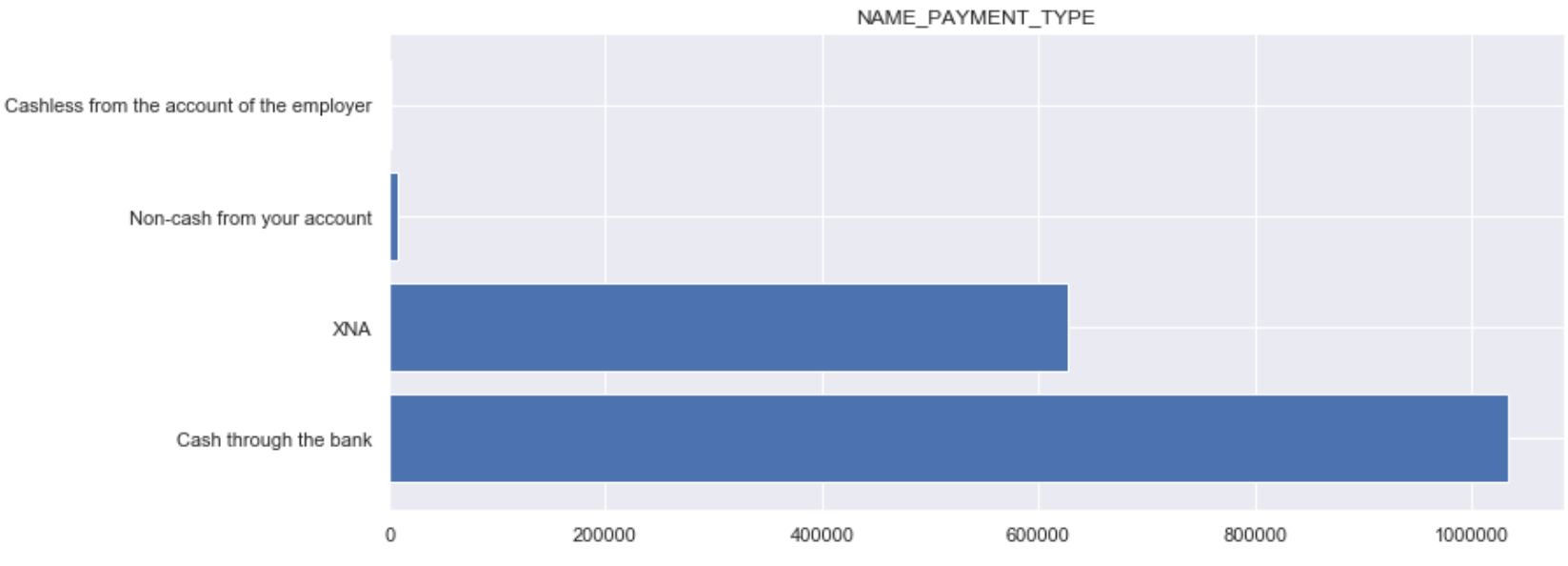
```
Categorical_Data_for_previous = ['NAME_CONTRACT_TYPE','NAME_CONTRACT_STATUS','NAME_PAYMENT_TYPE','NAME_TYPE_SUITE',
                                 'NAME_CLIENT_TYPE','AMT_CREDIT_LAKHS_Range','AMT_APPLICATION_LAKHS_Range',]
```

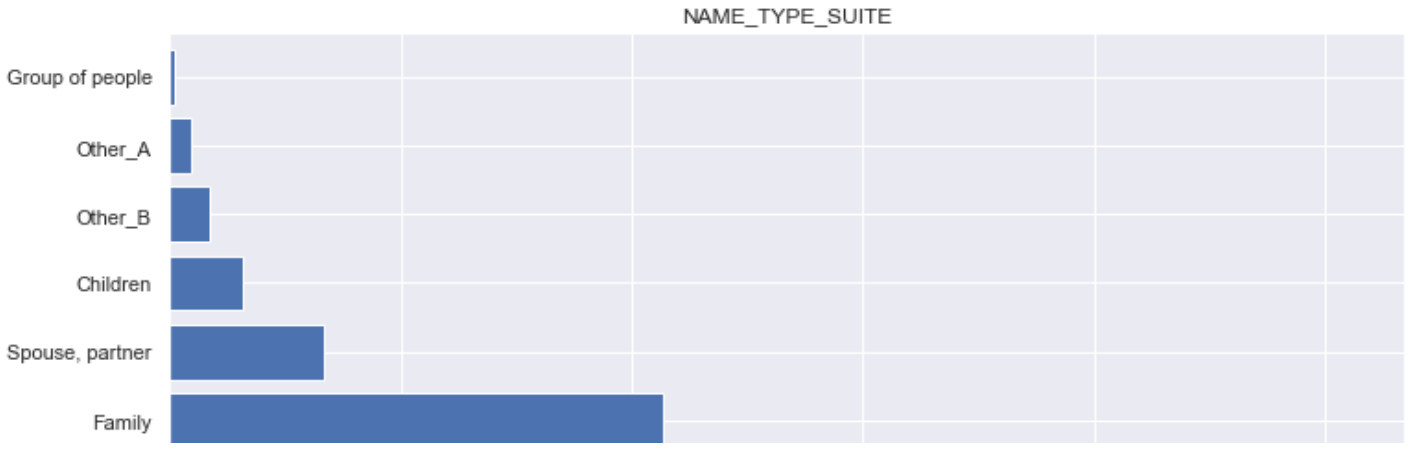
In [30]:

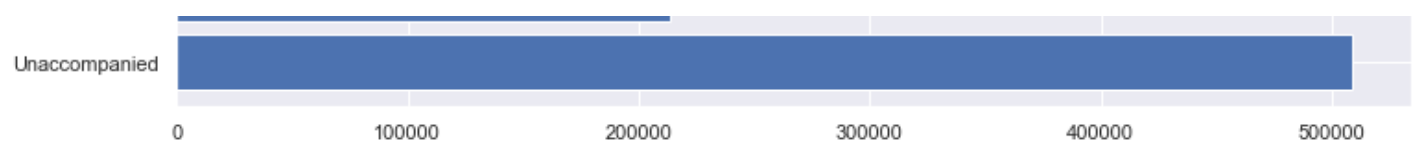
```
for i in Categorical_Data_for_previous:
        Uni_Analysis_Categorcal(df_previous_application,i)
```

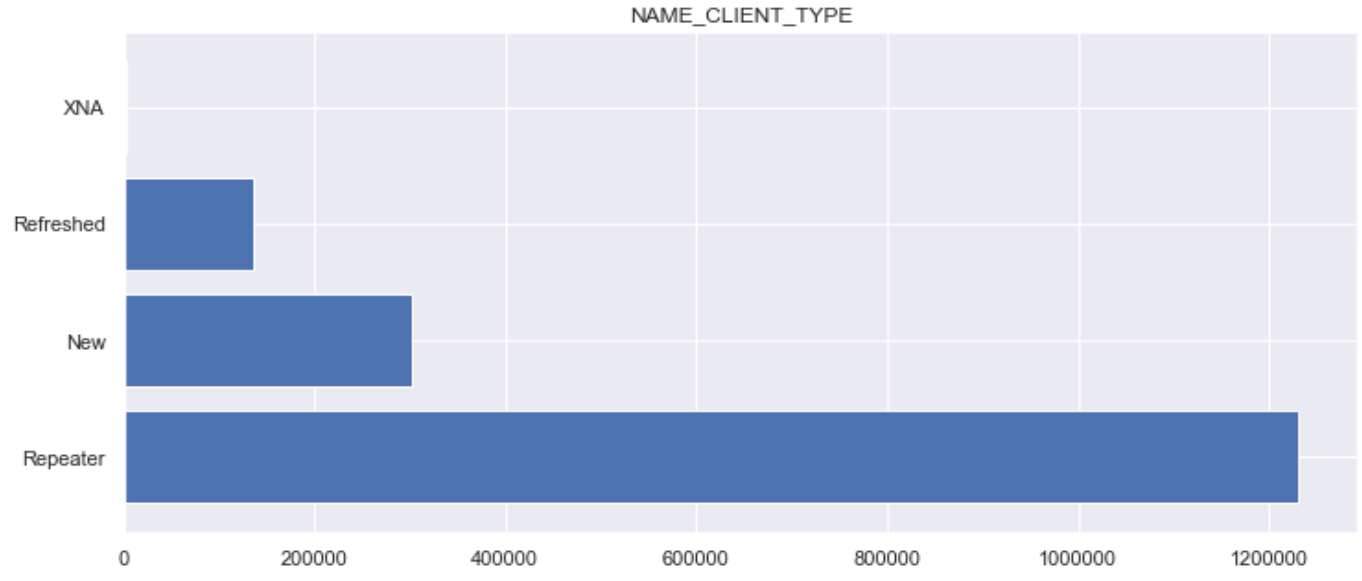



Unused offer
Refused
Canceled
Approved
0
200000
400000
600000
800000
1000000


NAME_PAYMENT_TYPE
Cashless from the account of the employer
Non-cash from your account
XNA
Cash through the bank
0
200000
400000
600000
800000
1000000


NAME_TYPE_SUITE
Group of people
Other_A
Other_B
Children
Spouse, partner
Family






```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2656             try:
-> 2657                 return self._engine.get_loc(key)
   2658             except KeyError:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'AMT_CREDIT_LAKHS_Range'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
<ipython-input-30-95042033bc3e> in <module>
      1 for i in Categorical_Data_for_previous:
----> 2         Uni_Analysis_Categorcal(df_previous_application,i)

<ipython-input-20-6c981fbf8f56> in Uni_Analysis_Categorcal(dataframe, column)
```

```
      2     sns.set(style= "darkgrid")
      3     plt.figure(figsize = [12,5])
----> 4     dataframe[column].value_counts().plot.barh(width = 0.8)
      5     plt.title(column)
      6     plt.show()

~\Anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
   2925             if self.columns.nlevels > 1:
   2926                 return self._getitem_multilevel(key)
-> 2927             indexer = self.columns.get_loc(key)
   2928             if is_integer(indexer):
   2929                 indexer = [indexer]

~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2657                 return self._engine.get_loc(key)
   2658             except KeyError:
-> 2659                 return self._engine.get_loc(self._maybe_cast_indexer(key))
   2660         indexer = self.get_indexer([key], method=method, tolerance=tolerance)
   2661         if indexer.ndim > 1 or indexer.size > 1:

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'AMT_CREDIT_LAKHS_Range'
```

```
<Figure size 864x360 with 0 Axes>
```

5 Bivariate Analysis For Target

In [116]:

```
df_previous_application.head()
```

Out[116]:

	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_ST
0	2030495	271877	Consumer loans	1730.430	17145.0	17145.0	0.0	17145.0	SATUI
1	2802425	108129	Cash loans	25188.615	607500.0	679671.0	NaN	607500.0	THURS
2	2523466	122040	Cash loans	15060.735	112500.0	136444.5	NaN	112500.0	TUES
3	2819243	176158	Cash loans	47041.335	450000.0	470790.0	NaN	450000.0	MON

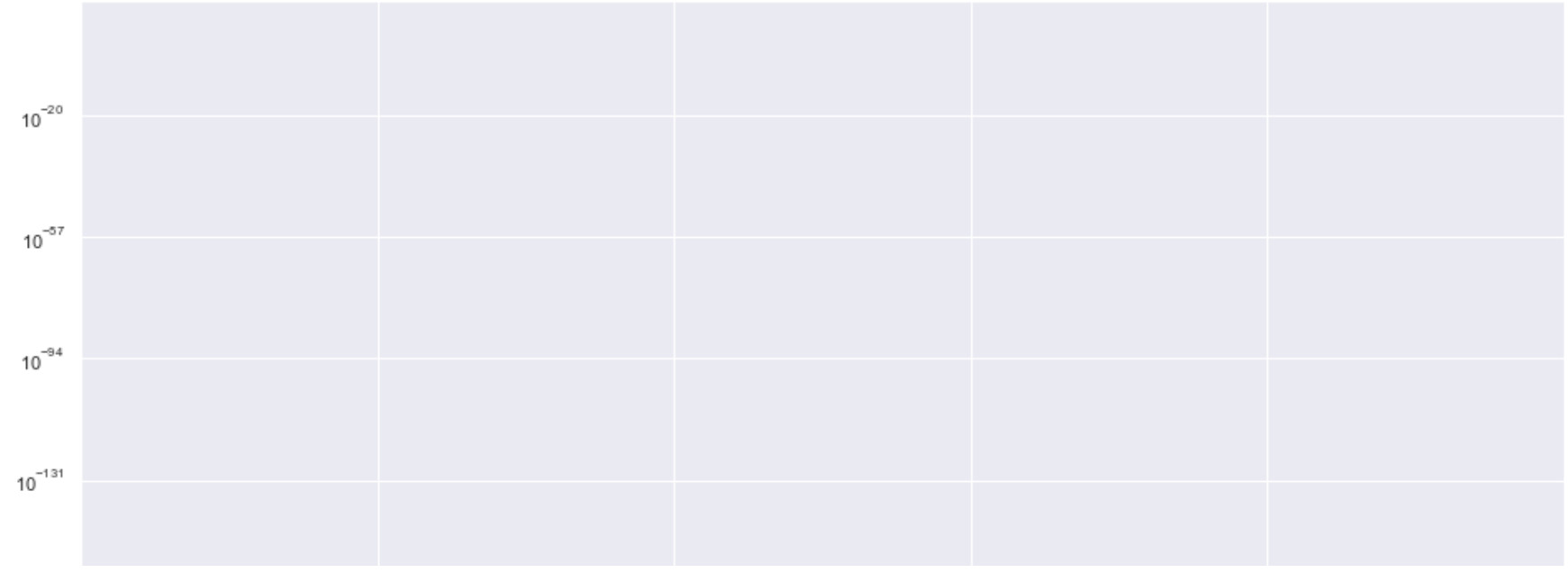
	SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	WEEKDAY_APPR_PROCESS_ST
4	1784265	202054	Cash loans	31924.395	337500.0	404055.0	NaN	337500.0	THURS

Bivariate Analysis For Categories

In [126]:

```
plt.figure(figsize=(16,12))
plt.xticks(rotation=45)
plt.yscale('log')
sns.boxplot(data =target0_df_previous_application, x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL', hue ='NAME_FAMILY_STATUS',orient=
'v')
plt.title('Income amount vs Education Status')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-126-ff9bba4aa047> in <module>
      2 plt.xticks(rotation=45)
      3 plt.yscale('log')
----> 4 sns.boxplot(data =target0_df_previous_application, x='NAME_EDUCATION_TYPE',y='AMT_INCOME_TOTAL', hue ='NAME_FAMILY_STATUS'
,orient='v')
      5 plt.title('Income amount vs Education Status')
      6 plt.show()

NameError: name 'target0_df_previous_application' is not defined
```



In [ ]: