In [1]:

```
import warnings
warnings.simplefilter('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.model_selection import train_test_split  # sklearn.cross_validation in
import re
from time import time
from scipy import stats
import json

import numpy as np
import pandas as pd
```

Loading Data

In [2]:

```
application_test = pd.read_csv('application_test.csv')
application_train = pd.read_csv('application_train.csv')
# bureau = pd.read_csv('bureau.csv')
# bureau_balance = pd.read_csv('bureau_balance.csv')
# credit_card_balance = pd.read_csv('credit_card_balance.csv')
# # HomeCredit_columns_description = pd.read_csv('HomeCredit_columns_description.csv
# installments_payments = pd.read_csv('installments_payments.csv')
# POS_CASH_balance = pd.read_csv('POS_CASH_balance.csv')
# previous_application = pd.read_csv('previous_application.csv')
# sample_submission = pd.read_csv('sample_submission.csv')
```

EDA

In [86]:

```
def EDA(df,df_name):
    print("Test description; data type: {}".format(df_name))
    print(df.dtypes)
    print("\n###########################################################\n")
    print(" Dataset size (rows columns): {}".format(df_name))
    print(df.shape)
    print("\n###########################################################\n")
    print("Summary statistics: {}".format(df_name))
    print(df.describe())
    print("\n###########################################################\n")
    print("Correlation analysis: {}".format(df_name))
    print(df.corr())
    print("\n###########################################################\n")
    print("Other Analysis: {}".format(df_name))
    print("1. Checking for Null values: {}".format(df_name))
    print(df.isna().sum())
    print("\n2. Info")
    print(df.info())
```

In [87]:

```
EDA(application_train,'application_train_data')
```

```
Test description; data type: application_train_data
SK_ID_CURR                              int64
TARGET                                  int64
CODE_GENDER                            object
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
FLAG_OWN_CAR                  object
                                ...
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
Length: 122, dtype: object

#############################################################

 Dataset size (rows columns): application_train_data
(307511, 122)

#############################################################

Summary statistics: application_train_data
          SK_ID_CURR         TARGET   CNT_CHILDREN  AMT_INCOME_TOTAL  \
count  307511.000000  307511.000000  307511.000000      3.075110e+05
mean   278180.518577       0.080729       0.417052      1.687979e+05
std    102790.175348       0.272419       0.722121      2.371231e+05
min    100002.000000       0.000000       0.000000      2.565000e+04
25%    189145.500000       0.000000       0.000000      1.125000e+05
50%    278202.000000       0.000000       0.000000      1.471500e+05
75%    367142.500000       0.000000       1.000000      2.025000e+05
max    456255.000000       1.000000      19.000000      1.170000e+08

         AMT_CREDIT    AMT_ANNUITY  AMT_GOODS_PRICE  \
count  3.075110e+05  307499.000000     3.072330e+05
mean   5.990260e+05   27108.573909     5.383962e+05
std    4.024908e+05   14493.737315     3.694465e+05
min    4.500000e+04    1615.500000     4.050000e+04
25%    2.700000e+05   16524.000000     2.385000e+05
50%    5.135310e+05   24903.000000     4.500000e+05
75%    8.086500e+05   34596.000000     6.795000e+05
max    4.050000e+06  258025.500000     4.050000e+06

       REGION_POPULATION_RELATIVE     DAYS_BIRTH  DAYS_EMPLOYED  ...  \
count               307511.000000  307511.000000  307511.000000  ...
mean                     0.020868  -16036.995067   63815.045904  ...
std                      0.013831    4363.988632  141275.766519  ...
min                      0.000290  -25229.000000  -17912.000000  ...
25%                      0.010006  -19682.000000   -2760.000000  ...
50%                      0.018850  -15750.000000   -1213.000000  ...
75%                      0.028663  -12413.000000    -289.000000  ...
max                      0.072508   -7489.000000  365243.000000  ...

       FLAG_DOCUMENT_18  FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  FLAG_DOCUMENT_21  \
count     307511.000000     307511.000000     307511.000000     307511.000000
mean           0.008130          0.000595          0.000507          0.000335
std            0.089798          0.024387          0.022518          0.018299
min            0.000000          0.000000          0.000000          0.000000
25%            0.000000          0.000000          0.000000          0.000000
50%            0.000000          0.000000          0.000000          0.000000
75%            0.000000          0.000000          0.000000          0.000000
max            1.000000          1.000000          1.000000          1.000000

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

       AMT_REQ_CREDIT_BUREAU_WEEK  AMT_REQ_CREDIT_BUREAU_MON  \
count               265992.000000              265992.000000
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
mean                   0.034362                0.267395
std                    0.204685                0.916002
min                    0.000000                0.000000
25%                    0.000000                0.000000
50%                    0.000000                0.000000
75%                    0.000000                0.000000
max                    8.000000               27.000000

       AMT_REQ_CREDIT_BUREAU_QRT  AMT_REQ_CREDIT_BUREAU_YEAR
count              265992.000000               265992.000000
mean                    0.265474                    1.899974
std                     0.794056                    1.869295
min                     0.000000                    0.000000
25%                     0.000000                    0.000000
50%                     0.000000                    1.000000
75%                     0.000000                    3.000000
max                   261.000000                   25.000000

[8 rows x 106 columns]

##########################################################

Correlation analysis: application_train_data
                            SK_ID_CURR    TARGET  CNT_CHILDREN  \
SK_ID_CURR                    1.000000 -0.002108     -0.001129
TARGET                       -0.002108  1.000000      0.019187
CNT_CHILDREN                 -0.001129  0.019187      1.000000
AMT_INCOME_TOTAL             -0.001820 -0.003982      0.012882
AMT_CREDIT                   -0.000343 -0.030369      0.002145
...                                ...       ...           ...
AMT_REQ_CREDIT_BUREAU_DAY    -0.002193  0.002704     -0.000366
AMT_REQ_CREDIT_BUREAU_WEEK    0.002099  0.000788     -0.002436
AMT_REQ_CREDIT_BUREAU_MON     0.000485 -0.012462     -0.010808
AMT_REQ_CREDIT_BUREAU_QRT     0.001025 -0.002022     -0.007836
AMT_REQ_CREDIT_BUREAU_YEAR    0.004659  0.019930     -0.041550

                            AMT_INCOME_TOTAL  AMT_CREDIT  AMT_ANNUITY  \
SK_ID_CURR                         -0.001820   -0.000343    -0.000433
TARGET                             -0.003982   -0.030369    -0.012817
CNT_CHILDREN                        0.012882    0.002145     0.021374
AMT_INCOME_TOTAL                    1.000000    0.156870     0.191657
AMT_CREDIT                          0.156870    1.000000     0.770138
...                                      ...         ...          ...
AMT_REQ_CREDIT_BUREAU_DAY           0.002944    0.004238     0.002185
AMT_REQ_CREDIT_BUREAU_WEEK          0.002387   -0.001275     0.013881
AMT_REQ_CREDIT_BUREAU_MON           0.024700    0.054451     0.039148
AMT_REQ_CREDIT_BUREAU_QRT           0.004859    0.015925     0.010124
AMT_REQ_CREDIT_BUREAU_YEAR          0.011690   -0.048448    -0.011320

                            AMT_GOODS_PRICE  REGION_POPULATION_RELATIVE  \
SK_ID_CURR                        -0.000232                    0.000849
TARGET                            -0.039645                   -0.037227
CNT_CHILDREN                      -0.001827                   -0.025573
AMT_INCOME_TOTAL                   0.159610                    0.074796
AMT_CREDIT                         0.986968                    0.099738
...                                     ...                         ...
AMT_REQ_CREDIT_BUREAU_DAY          0.004677                    0.001399
AMT_REQ_CREDIT_BUREAU_WEEK        -0.001007                   -0.002149
AMT_REQ_CREDIT_BUREAU_MON          0.056422                    0.078607
AMT_REQ_CREDIT_BUREAU_QRT          0.016432                   -0.001279
AMT_REQ_CREDIT_BUREAU_YEAR        -0.050998                    0.001003

                            DAYS_BIRTH  DAYS_EMPLOYED  ...  FLAG_DOCUMENT_18  \
SK_ID_CURR                   -0.001500       0.001366  ...          0.000509
TARGET                        0.078239      -0.044932  ...         -0.007952
CNT_CHILDREN                  0.330938      -0.239818  ...          0.004031
AMT_INCOME_TOTAL              0.027261      -0.064223  ...          0.003130
[illegible]                   [illegible]6  -0.066838  ...          0.034329
...                                ...            ...  ...               ...
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
AMT_REQ_CREDIT_BUREAU_DAY     0.002255       0.000472  ...           0.013281
AMT_REQ_CREDIT_BUREAU_WEEK   -0.001336       0.003072  ...          -0.004640
AMT_REQ_CREDIT_BUREAU_MON     0.001372      -0.034457  ...          -0.001565
AMT_REQ_CREDIT_BUREAU_QRT    -0.011799       0.015345  ...          -0.005125
AMT_REQ_CREDIT_BUREAU_YEAR   -0.071983       0.049988  ...          -0.047432

                            FLAG_DOCUMENT_19  FLAG_DOCUMENT_20  \
SK_ID_CURR                          0.000167          0.001073   
TARGET                             -0.001358          0.000215   
CNT_CHILDREN                        0.000864          0.000988   
AMT_INCOME_TOTAL                    0.002408          0.000242   
AMT_CREDIT                          0.021082          0.031023   
...                                      ...               ...   
AMT_REQ_CREDIT_BUREAU_DAY           0.001126         -0.000120   
AMT_REQ_CREDIT_BUREAU_WEEK         -0.001275         -0.001770   
AMT_REQ_CREDIT_BUREAU_MON          -0.002729          0.001285   
AMT_REQ_CREDIT_BUREAU_QRT          -0.001575         -0.001010   
AMT_REQ_CREDIT_BUREAU_YEAR         -0.007009         -0.012126   

                            FLAG_DOCUMENT_21  AMT_REQ_CREDIT_BUREAU_HOUR  \
SK_ID_CURR                          0.000282                   -0.002672   
TARGET                              0.003709                    0.000930   
CNT_CHILDREN                       -0.002450                   -0.000410   
AMT_INCOME_TOTAL                   -0.000589                    0.000709   
AMT_CREDIT                         -0.016148                   -0.003906   
...                                      ...                         ...   
AMT_REQ_CREDIT_BUREAU_DAY          -0.001130                    0.230374   
AMT_REQ_CREDIT_BUREAU_WEEK          0.000081                    0.004706   
AMT_REQ_CREDIT_BUREAU_MON          -0.003612                   -0.000018   
AMT_REQ_CREDIT_BUREAU_QRT          -0.002004                   -0.002716   
AMT_REQ_CREDIT_BUREAU_YEAR         -0.005457                   -0.004597   

                            AMT_REQ_CREDIT_BUREAU_DAY  \
SK_ID_CURR                                  -0.002193   
TARGET                                       0.002704   
CNT_CHILDREN                                -0.000366   
AMT_INCOME_TOTAL                             0.002944   
AMT_CREDIT                                   0.004238   
...                                               ...   
AMT_REQ_CREDIT_BUREAU_DAY                    1.000000   
AMT_REQ_CREDIT_BUREAU_WEEK                   0.217412   
AMT_REQ_CREDIT_BUREAU_MON                   -0.005258   
AMT_REQ_CREDIT_BUREAU_QRT                   -0.004416   
AMT_REQ_CREDIT_BUREAU_YEAR                  -0.003355   

                            AMT_REQ_CREDIT_BUREAU_WEEK  \
SK_ID_CURR                                     0.002099   
TARGET                                         0.000788   
CNT_CHILDREN                                  -0.002436   
AMT_INCOME_TOTAL                               0.002387   
AMT_CREDIT                                    -0.001275   
...                                                ...   
AMT_REQ_CREDIT_BUREAU_DAY                      0.217412   
AMT_REQ_CREDIT_BUREAU_WEEK                     1.000000   
AMT_REQ_CREDIT_BUREAU_MON                     -0.014096   
AMT_REQ_CREDIT_BUREAU_QRT                     -0.015115   
AMT_REQ_CREDIT_BUREAU_YEAR                     0.018917   

                            AMT_REQ_CREDIT_BUREAU_MON  \
SK_ID_CURR                                   0.000485   
TARGET                                      -0.012462   
CNT_CHILDREN                                -0.010808   
AMT_INCOME_TOTAL                             0.024700   
AMT_CREDIT                                   0.054451   
...                                               ...   
AMT_REQ_CREDIT_BUREAU_DAY                   -0.005258   
AMT_REQ_CREDIT_BUREAU_WEEK                  -0.014096   
AMT_REQ_CREDIT_BUREAU_MON                    1.000000   
AMT_REQ_CREDIT_BUREAU_QRT                   -0.007789   
```

```
AMT_REQ_CREDIT_BUREAU_YEAR                      -0.004975

                            AMT_REQ_CREDIT_BUREAU_QRT  \
SK_ID_CURR                                   0.001025
TARGET                                      -0.002022
CNT_CHILDREN                                -0.007836
AMT_INCOME_TOTAL                             0.004859
AMT_CREDIT                                   0.015925
...                                               ...
AMT_REQ_CREDIT_BUREAU_DAY                   -0.004416
AMT_REQ_CREDIT_BUREAU_WEEK                  -0.015115
AMT_REQ_CREDIT_BUREAU_MON                   -0.007789
AMT_REQ_CREDIT_BUREAU_QRT                    1.000000
AMT_REQ_CREDIT_BUREAU_YEAR                   0.076208

                            AMT_REQ_CREDIT_BUREAU_YEAR
SK_ID_CURR                                    0.004659
TARGET                                        0.019930
CNT_CHILDREN                                 -0.041550
AMT_INCOME_TOTAL                              0.011690
AMT_CREDIT                                   -0.048448
...                                                ...
AMT_REQ_CREDIT_BUREAU_DAY                    -0.003355
AMT_REQ_CREDIT_BUREAU_WEEK                    0.018917
AMT_REQ_CREDIT_BUREAU_MON                    -0.004975
AMT_REQ_CREDIT_BUREAU_QRT                     0.076208
AMT_REQ_CREDIT_BUREAU_YEAR                    1.000000

[106 rows x 106 columns]

##########################################################

Other Analysis: application_train_data
1. Checking for Null values: application_train_data
SK_ID_CURR                        0
TARGET                            0
NAME_CONTRACT_TYPE                0
CODE_GENDER                       0
FLAG_OWN_CAR                      0
                              ...
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

2. Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
None
```

Target Vs borrowers based on gender

In [30]:

```
male_data = application_train[application_train['CODE_GENDER']=='M']['TARGET'].value
male_data['count_percent'] = male_data['user_count']/male_data['user_count'].sum()*1
male_data['CODE_GENDER'] = 'M'
female_data = application_train[application_train['CODE_GENDER']=='F']['TARGET'].val
female_data['count_percent'] = female_data['user_count']/female_data['user_count'].s
female_data['CODE_GENDER'] = 'F'
gender_data = male_data.append(female_data, ignore_index=True,sort=False)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

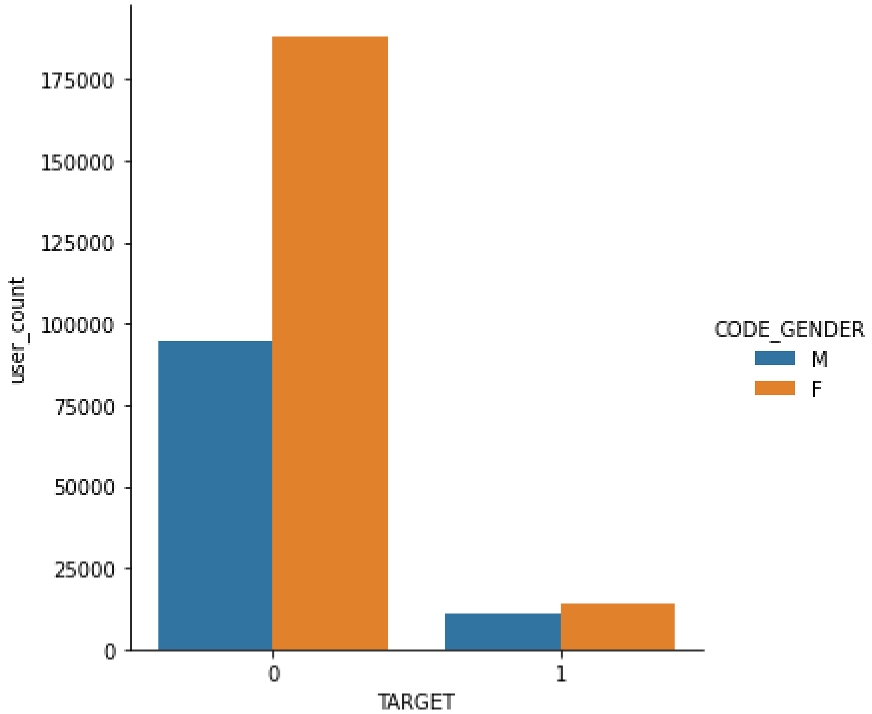
Out[30]:

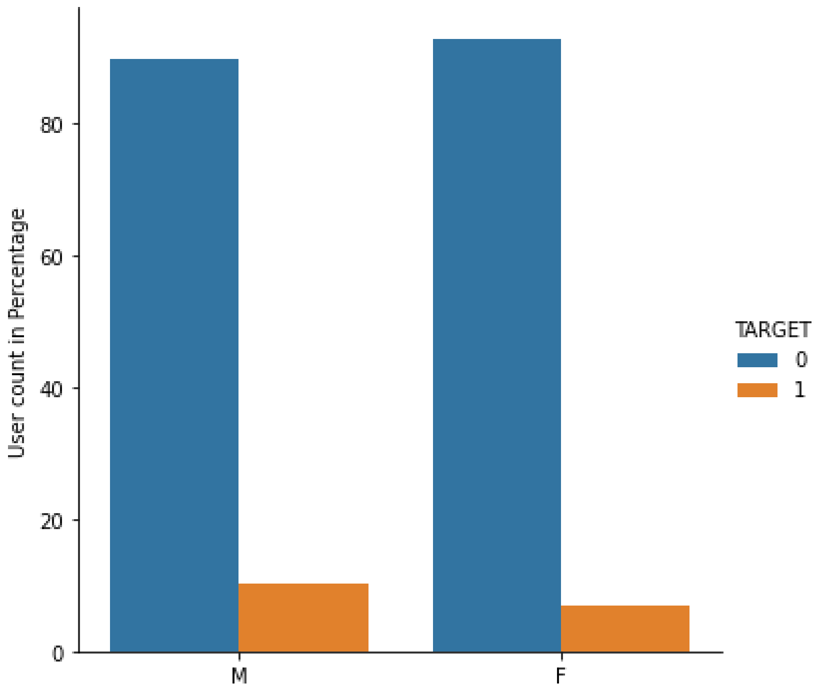
	TARGET	user_count	count_percent	CODE_GENDER
0	0	94404	89.858080	M
1	1	10655	10.141920	M
2	0	188278	93.000672	F
3	1	14170	6.999328	F

In [41]:

```
sns.catplot(data=gender_data, kind="bar", x="TARGET", y="user_count", hue="CODE_GEND
sns.catplot(data=gender_data, kind="bar", x="CODE_GENDER", y="count_percent", hue="T
plt.xlabel("Gender")
plt.ylabel('User count in Percentage')
```

Out[41]: Text(10.788472222222218, 0.5, 'User count in Percentage')





Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Male most likely to default than Female based on percentage of defaulter_count(Second Graph)

In [ ]:

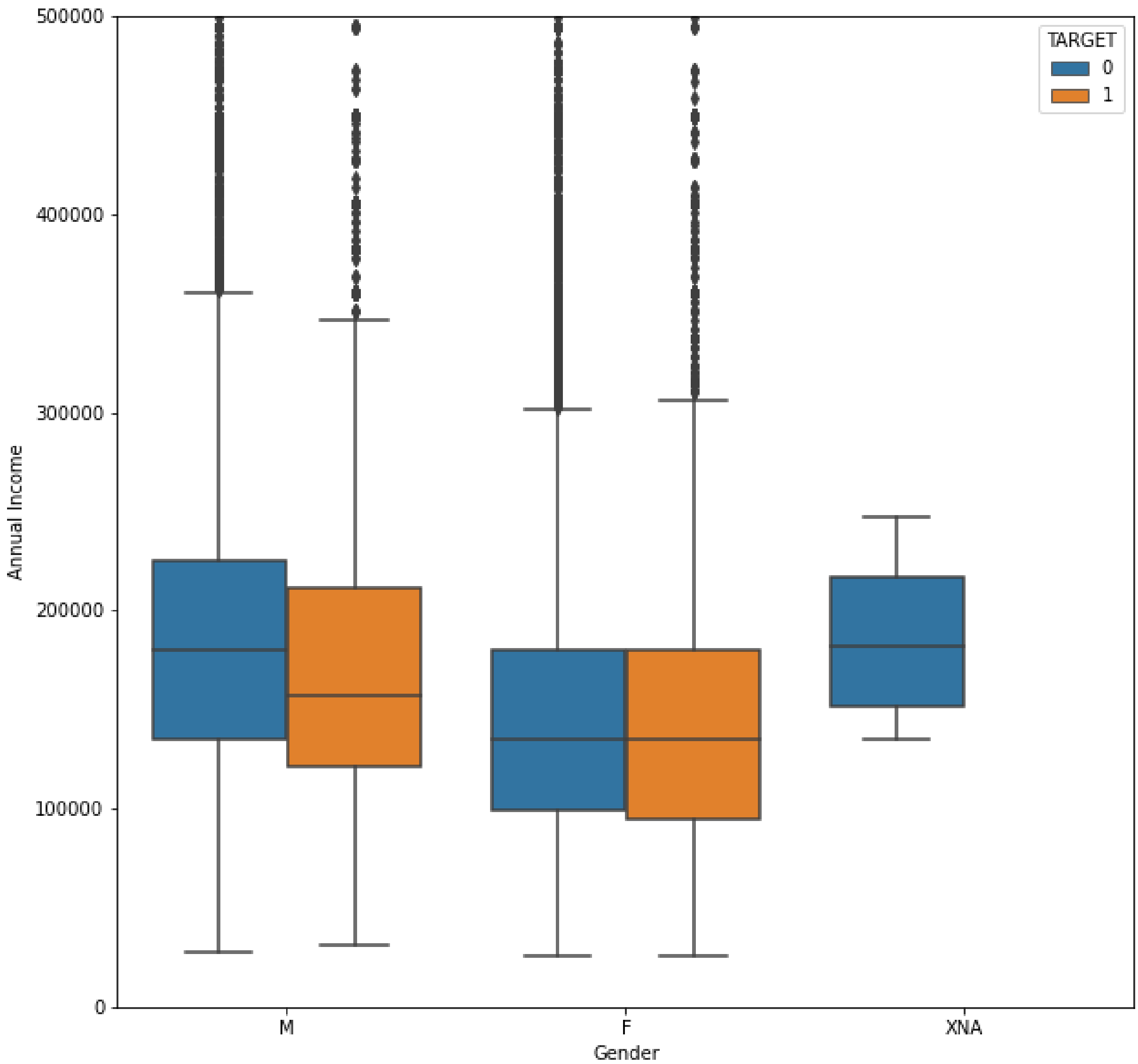
In [ ]:

Gender Vs Income based on Target

In [8]:

```
fig,ax = plt.subplots(figsize = (10,10))
sns.boxplot(x='CODE_GENDER',hue = 'TARGET',y='AMT_INCOME_TOTAL', data=application_tr
plt.ylim(0, 500000)
plt.xlabel("Gender")
plt.ylabel('Annual Income')
```

Out[8]: `Text(0, 0.5, 'Annual Income')`



Own House count based Target

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
own_house_data = application_train[application_train['FLAG_OWN_REALTY']=='Y']['TARGE
```

```
own_house_data['OWN_HOUSE'] = 'Y'
own_house_data['count_percent'] = own_house_data['user_count']/own_house_data['user_
not_own_house_data = application_train[application_train['FLAG_OWN_REALTY']=='N']['T
not_own_house_data['OWN_HOUSE'] = 'N'
not_own_house_data['count_percent'] = not_own_house_data['user_count']/not_own_house
own_house_data = own_house_data.append(not_own_house_data,ignore_index=True,sort=Fal
own_house_data
```

Out[46]:

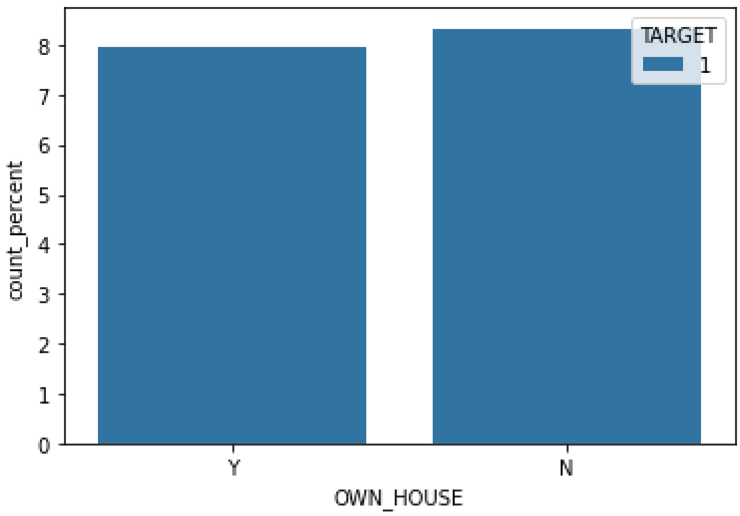
	TARGET	user_count	OWN_HOUSE	count_percent
0	0	196329	Y	92.038423
1	1	16983	Y	7.961577
2	0	86357	N	91.675071
3	1	7842	N	8.324929

In [100…

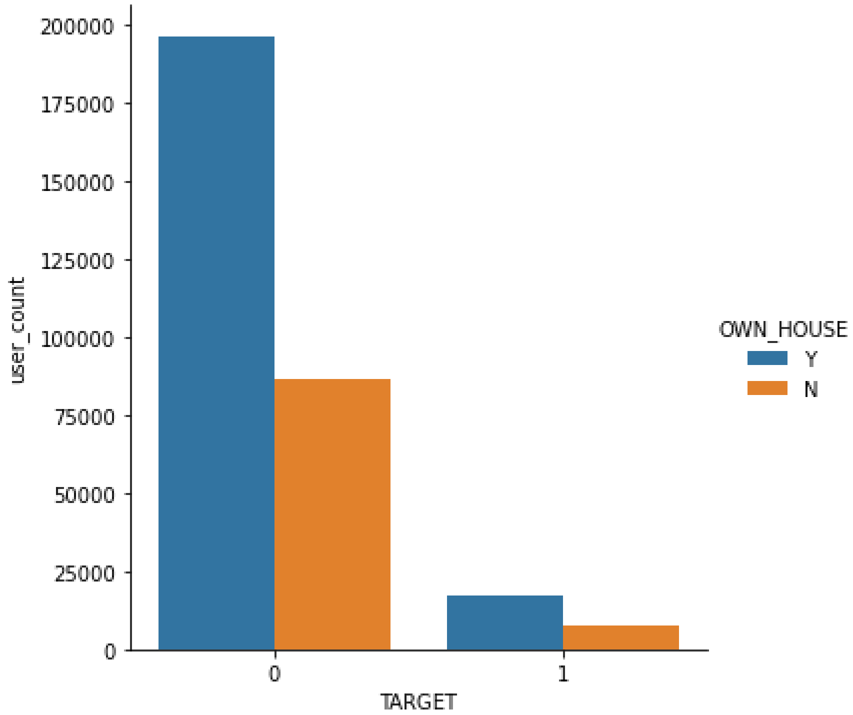
```
sns.barplot(x='OWN_HOUSE',y='count_percent',hue = 'TARGET',data=own_house_data[own_h

sns.catplot(data=own_house_data, kind="bar", x="TARGET", y="user_count", hue="OWN_HO

sns.catplot(data=own_house_data, kind="bar", x="OWN_HOUSE", y="count_percent", hue="
plt.xlabel("Own House")
plt.ylabel('User count in Percentage')
```

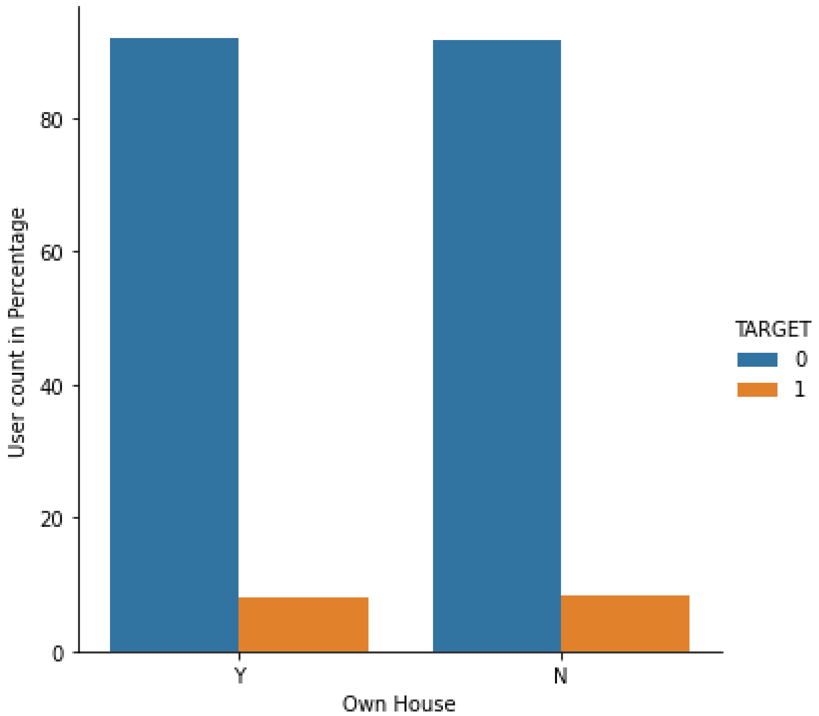
Out[100…

```
Text(10.788472222222218, 0.5, 'User count in Percentage')
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js





Not a significant difference, but borrowers who own a house are more likely to pay

In [ ]:

In [ ]:

Own car count based Target

In [88]:

```
own_car_data = application_train[application_train['FLAG_OWN_CAR']=='Y']['TARGET'].v
own_car_data['count_percent'] = own_car_data['user_count']/own_car_data['user_count'
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
not_own_car_data = application_train[application_train['FLAG_OWN_CAR']=='N']['TARGET
not_own_car_data['FLAG_OWN_CAR'] = 'N'
not_own_car_data['count_percent'] = not_own_car_data['user_count']/not_own_car_data[
own_car_data = own_car_data.append(not_own_car_data,ignore_index=True,sort=False)
own_car_data
```

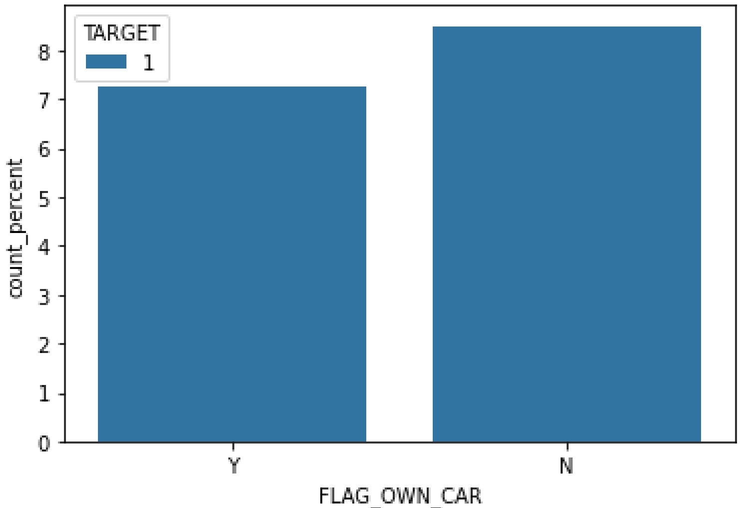
Out[88]:

	TARGET	user_count	FLAG_OWN_CAR	count_percent
0	0	97011	Y	92.756270
1	1	7576	Y	7.243730
2	0	185675	N	91.499773
3	1	17249	N	8.500227

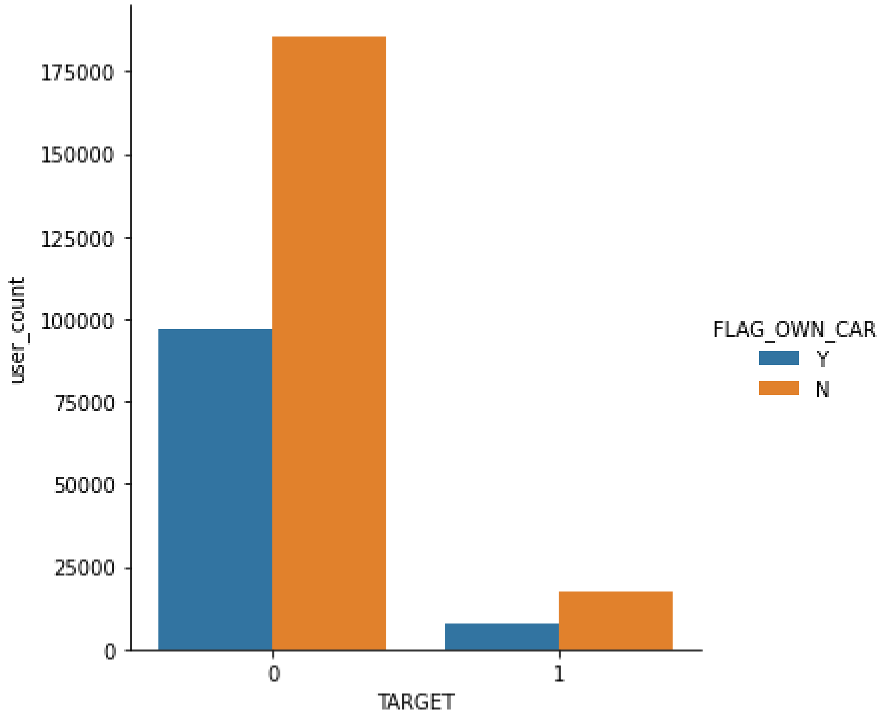
In [99]:

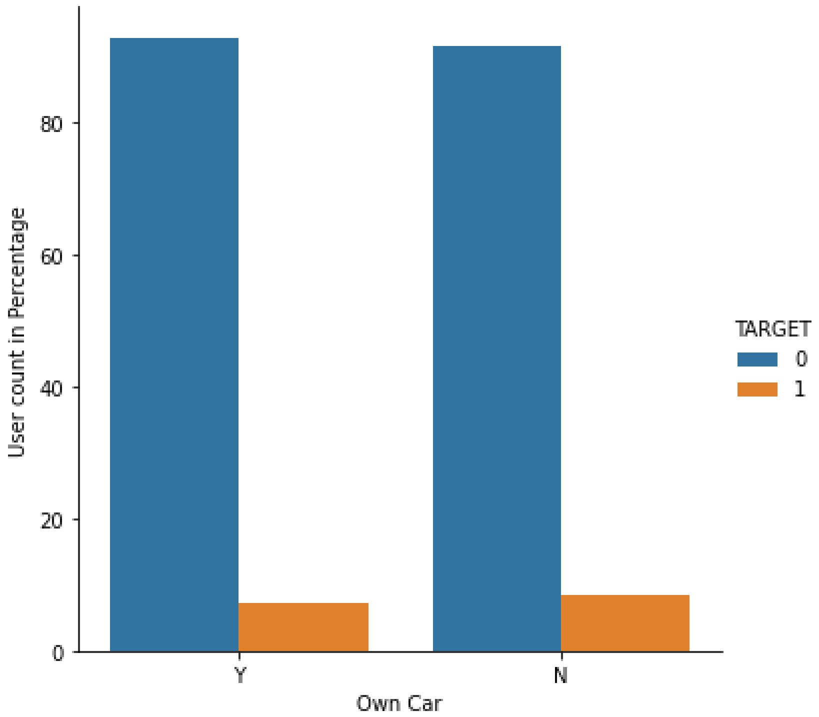
```
sns.barplot(x='FLAG_OWN_CAR',y='count_percent',hue = 'TARGET',data=own_car_data[own_

sns.catplot(data=own_car_data, kind="bar", x="TARGET", y="user_count", hue="FLAG_OWN

sns.catplot(data=own_car_data, kind="bar", x="FLAG_OWN_CAR", y="count_percent", hue=
plt.xlabel("Own Car")
plt.ylabel('User count in Percentage')
```

Out[99]: Text(10.788472222222218, 0.5, 'User count in Percentage')



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js





Borrowers owning a car are more likely to pay on time

In [ ]:

Occupation type count based on Target

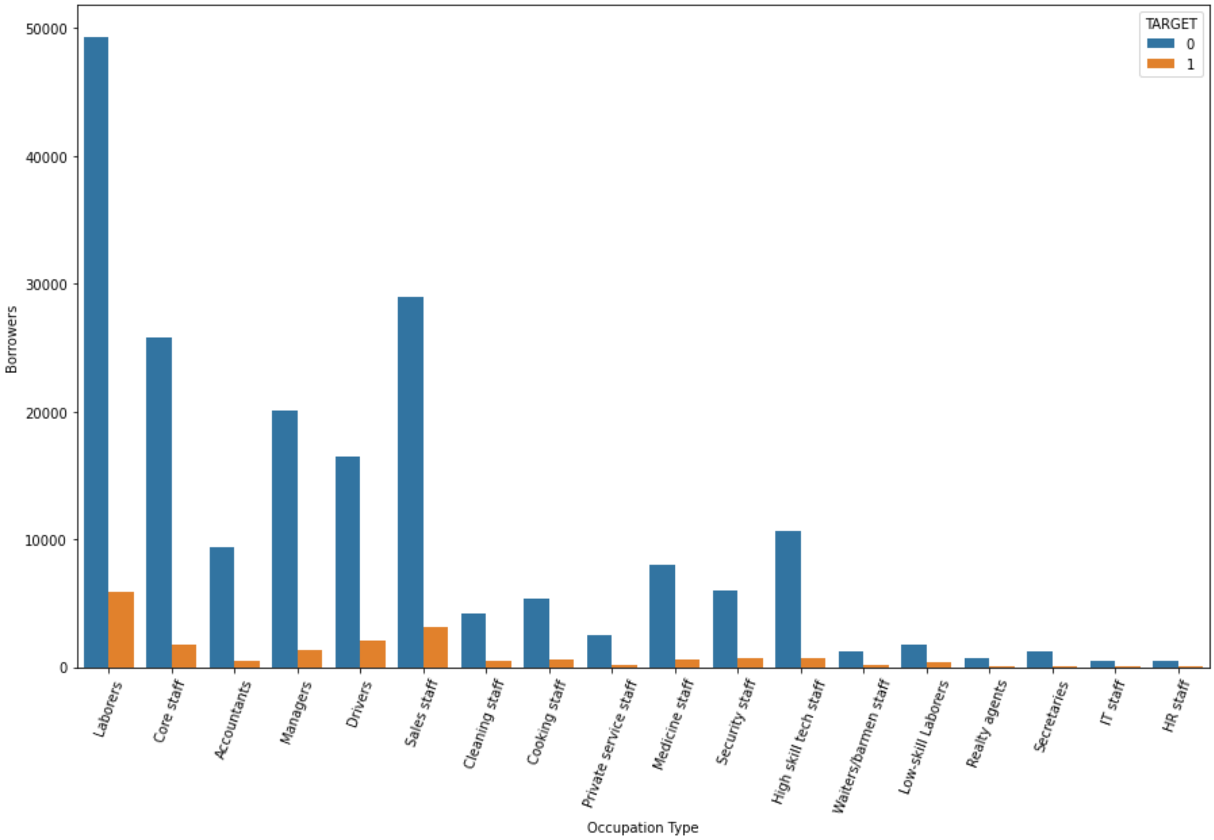
In [11]:

```
fig, ax = plt.subplots(figsize=(15,9))
sns.countplot(x='OCCUPATION_TYPE', hue = 'TARGET',data=application_train)
plt.xlabel("Occupation Type")
plt.ylabel('Borrowers')
plt.xticks(rotation=70)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[11]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
```

```
        17]),
 [Text(0, 0, 'Laborers'),
  Text(1, 0, 'Core staff'),
  Text(2, 0, 'Accountants'),
  Text(3, 0, 'Managers'),
  Text(4, 0, 'Drivers'),
  Text(5, 0, 'Sales staff'),
  Text(6, 0, 'Cleaning staff'),
  Text(7, 0, 'Cooking staff'),
  Text(8, 0, 'Private service staff'),
  Text(9, 0, 'Medicine staff'),
  Text(10, 0, 'Security staff'),
  Text(11, 0, 'High skill tech staff'),
  Text(12, 0, 'Waiters/barmen staff'),
  Text(13, 0, 'Low-skill Laborers'),
  Text(14, 0, 'Realty agents'),
  Text(15, 0, 'Secretaries'),
  Text(16, 0, 'IT staff'),
  Text(17, 0, 'HR staff')])
```



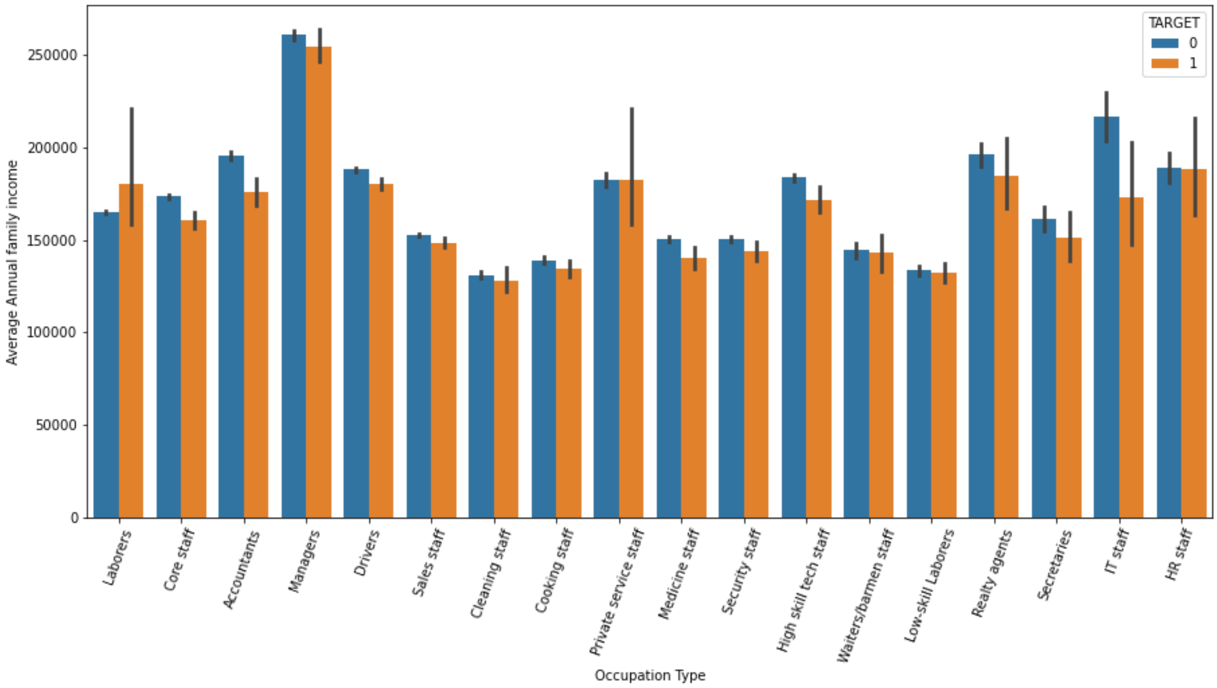
Occupation type vs income based on Target

In [12]:

```
fig, ax = plt.subplots(figsize=(15,7))
sns.barplot(x='OCCUPATION_TYPE',y='AMT_INCOME_TOTAL',hue = 'TARGET',data=application
plt.xticks(rotation=70)
plt.xlabel("Occupation Type")
plt.ylabel("Average Annual family income")
```

Out[12]: Text(0, 0.5, 'Average Annual family income')

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [76]:

```
income_credit_ratio_data = application_train[['AMT_INCOME_TOTAL','AMT_CREDIT','TARGE
income_credit_ratio_data['IC_ratio'] = income_credit_ratio_data['AMT_INCOME_TOTAL']/
income_credit_ratio_data['quantile'] = pd.qcut(income_credit_ratio_data['IC_ratio'],
income_credit_ratio_data
```

Out[76]:

	AMT_INCOME_TOTAL	AMT_CREDIT	TARGET	IC_ratio	quantile
0	202500.0	406597.5	1	0.498036	7
1	270000.0	1293502.5	0	0.208736	2
2	67500.0	135000.0	0	0.500000	7
3	135000.0	312682.5	0	0.431748	6
4	121500.0	513000.0	0	0.236842	3
...	...	...	...	...	...
307506	157500.0	254700.0	0	0.618375	8
307507	72000.0	269550.0	0	0.267112	4
307508	153000.0	677664.0	0	0.225776	3
307509	171000.0	370107.0	1	0.462029	7
307510	157500.0	675000.0	0	0.233333	3

307511 rows × 5 columns

In [77]:

```
income_credit_ratio_data = income_credit_ratio_data.groupby(['quantile','TARGET'])['
income_credit_ratio_data['count_percent'] = income_credit_ratio_data.apply(lambda x:
income_credit_ratio_data
```

Out[77]:

	quantile	TARGET	user_count	count_percent
0	0	0	28613	92.929523
1	0	1	2177	7.070477

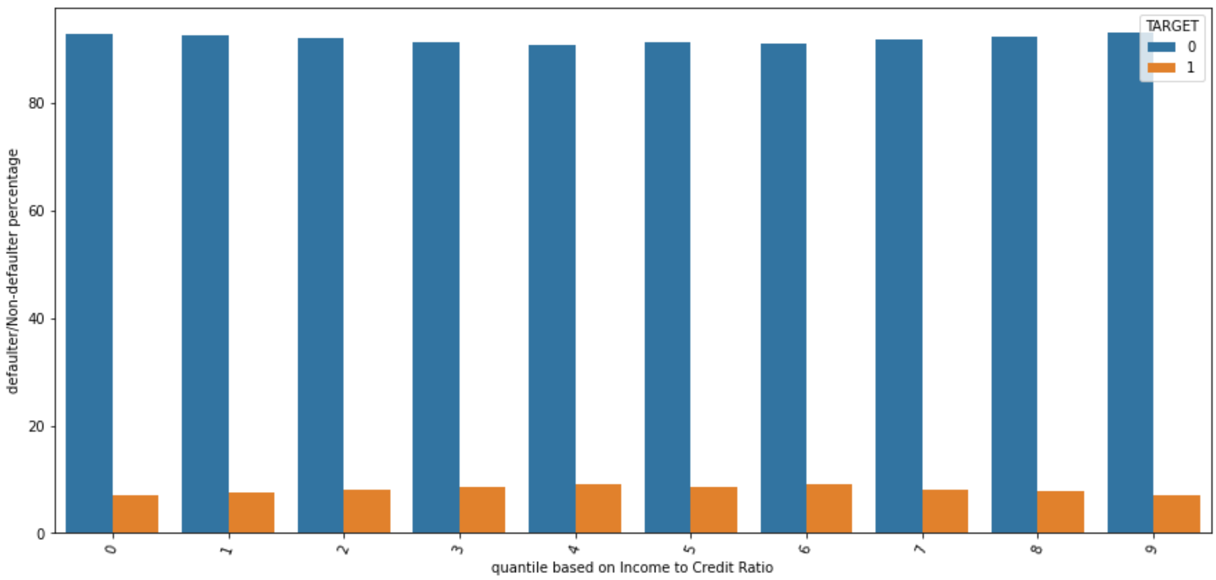
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	quantile	TARGET	user_count	count_percent
2	1	0	28499	92.577313
3	1	1	2285	7.422687
4	2	0	28241	92.035196
5	2	1	2444	7.964804
6	3	0	28128	91.375110
7	3	1	2655	8.624890
8	4	0	27899	90.805234
9	4	1	2825	9.194766
10	5	0	28298	91.307434
11	5	1	2694	8.692566
12	6	0	27764	91.023539
13	6	1	2738	8.976461
14	7	0	28498	91.863839
15	7	1	2524	8.136161
16	8	0	28126	92.264795
17	8	1	2358	7.735205
18	9	0	28620	93.088307
19	9	1	2125	6.911693

In [72]:

```
fig, ax = plt.subplots(figsize=(15,7))
sns.barplot(x='quantile',y='count_percent',hue = 'TARGET',data=income_credit_ratio_d
plt.xticks(rotation=70)
plt.xlabel("quantile based on Income to Credit Ratio")
plt.ylabel("defaulter/Non-defaulter percentage")
```

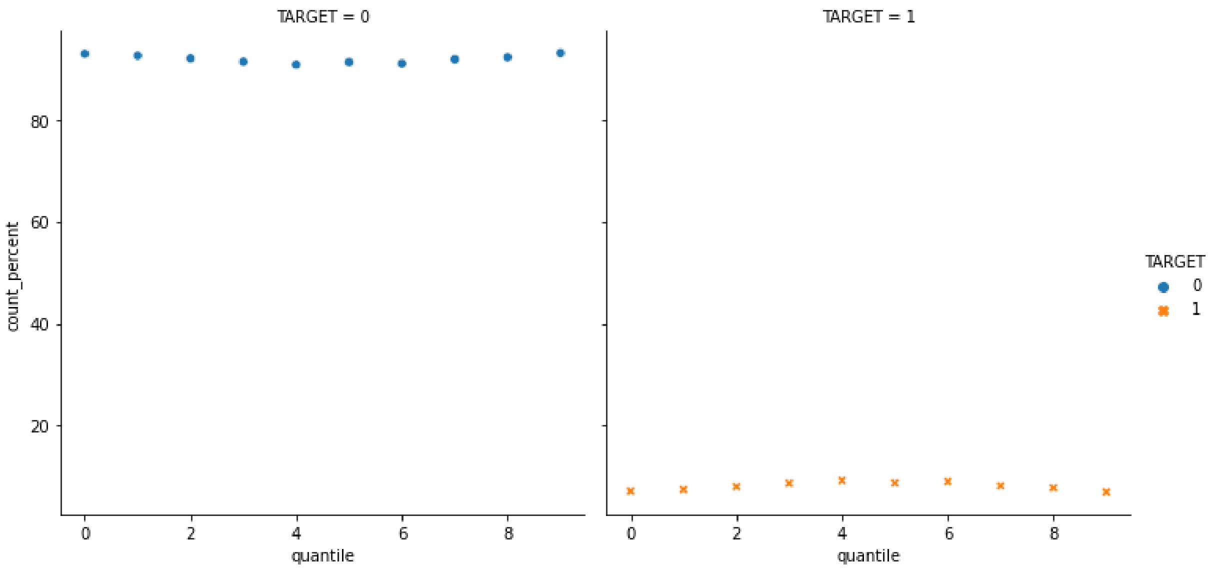
Out[72]: `Text(0, 0.5, 'defaulter/Non-defaulter percentage')`



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
    data=income_credit_ratio_data, x="quantile", y="count_percent",
    col="TARGET", hue="TARGET", style="TARGET",
    kind="scatter"
)
```

Out[84]: <seaborn.axisgrid.FacetGrid at 0x255b8154490>



Defaulters percentage is less when IC_ratio is either Low or High

In [ ]:

Repayers to Applicants Ratio

In [43]:

```
occ_data = pd.DataFrame(data=application_train.groupby(['OCCUPATION_TYPE','TARGET'])
occ_data = occ_data.reset_index()
value_counts = occ_data['SK_ID_CURR'].values
def repayers_to_applicants_ratio(values):
    flag = 1
    ratios = []
    for count in range(len(values)):
        if flag == 1:
            current_number = values[count]
            next_number = values[count+1]
            ratios.append(current_number/(current_number+next_number))
            ratios.append(current_number/(current_number+next_number))
        flag=flag*-1
    return ratios
occ_data['Ratio R/A'] = repayers_to_applicants_ratio(value_counts)
occ_ratio = occ_data.groupby(['OCCUPATION_TYPE','Ratio R/A']).count().drop(['TARGET'
occ_ratio = occ_ratio.reset_index()
occ_ratio = occ_ratio.sort_values(['Ratio R/A'],ascending=False)
occ_ratio
```

Out[43]:

	OCCUPATION_TYPE	Ratio R/A
0	Accountants	0.951697
6	High skill tech staff	0.938401
10	Managers	0.937860
3	Core staff	0.936960

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

	OCCUPATION_TYPE	Ratio R/A
5	HR staff	0.936057
7	IT staff	0.935361
12	Private service staff	0.934012
11	Medicine staff	0.932998
15	Secretaries	0.929502
13	Realty agents	0.921438
1	Cleaning staff	0.903933
14	Sales staff	0.903682
2	Cooking staff	0.895560
8	Laborers	0.894212
16	Security staff	0.892576
17	Waiters/barmen staff	0.887240
4	Drivers	0.886739
9	Low-skill Laborers	0.828476

Correlation of the positive days since birth and target

In [50]:

```
# Find the correlation of the positive days since birth and target
application_train['DAYS_BIRTH'] = abs(application_train['DAYS_BIRTH'])
-1*(application_train['DAYS_BIRTH'].corr(application_train['TARGET']))
```

Out[50]:

```
0.07823930830982712
```

Correlation of the positive days since employement and target

In [47]:

```
application_train['DAYS_EMPLOYED'] = abs(application_train['DAYS_EMPLOYED'])
-1*(application_train['DAYS_EMPLOYED'].corr(application_train['TARGET']))
```

Out[47]:

```
0.04704582521599294
```

Fetching important releavant features

In [110…

```
imp_features = ['FLOORSMAX_MEDI', 'ELEVATORS_MEDI', 'AMT_GOODS_PRICE', 'EMERGENCYSTA
imp_features = ['CODE_GENDER','FLAG_OWN_REALTY','FLAG_OWN_CAR','AMT_CREDIT','AMT_ANN
imp_features = list(set(imp_features))
```

In [111

```
Report
profile = ProfileReport(application_train[imp_features], title='HomeCredit Dataset P
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

In [112…

```
profile
```

Overview

Dataset statistics

Number of variables	30
Number of observations	307511
Missing cells	2447652
Missing cells (%)	26.5%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	168.5 MiB
Average record size in memory	574.6 B

Variable types

Categorical	8
Numeric	19
Boolean	3

Alerts

`BASEMENTAREA_MODE` is highly correlated with `ENTRANCES_MEDI` and 2 other fields (ENTRANCES_MEDI, APARTMENTS_MODE, BASEMENTAREA_MEDI) **High correlation**

`AMT_ANNUITY` is highly correlated with `AMT_GOODS_PRICE` and 1 **High correlation**

Out[112…

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js