In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from matplotlib.pyplot import figure
warnings.filterwarnings(action="ignore")
pd.set_option("display.max_columns",500)
pd.set_option("display.max_rows",500)
```

In [4]:
```
df=pd.read_csv("application_data.csv")
df.head()
```

Out[4]:

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

In [4]:
```
df.shape
```

Out[4]:
```
(307511, 122)
```

In [6]:
```
df.info(verbose=True, show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 122 columns):
 #    Column                       Non-Null Count   Dtype
---   ------                       --------------   -----
 0    SK_ID_CURR                   307511 non-null  int64
 1    TARGET                       307511 non-null  int64
 2    NAME_CONTRACT_TYPE           307511 non-null  object
 3    CODE_GENDER                  307511 non-null  object
 4    FLAG_OWN_CAR                 307511 non-null  object
 5    FLAG_OWN_REALTY              307511 non-null  object
 6    CNT_CHILDREN                 307511 non-null  int64
 7    AMT_INCOME_TOTAL             307511 non-null  float64
 8    AMT_CREDIT                   307511 non-null  float64
 9    AMT_ANNUITY                  307499 non-null  float64
 10   AMT_GOODS_PRICE              307233 non-null  float64
 11   NAME_TYPE_SUITE              306219 non-null  object
 12   NAME_INCOME_TYPE             307511 non-null  object
 13   NAME_EDUCATION_TYPE          307511 non-null  object
 14   NAME_FAMILY_STATUS           307511 non-null  object
 15   NAME_HOUSING_TYPE            307511 non-null  object
 16   REGION_POPULATION_RELATIVE   307511 non-null  float64
 17   DAYS_BIRTH                   307511 non-null  int64
 18   DAYS_EMPLOYED                307511 non-null  int64
 19   DAYS_REGISTRATION            307511 non-null  float64
 20   DAYS_ID_PUBLISH              307511 non-null  int64
 21   OWN_CAR_AGE                  104582 non-null  float64
 22   FLAG_MOBIL                   307511 non-null  int64
 23   FLAG_EMP_PHONE               307511 non-null  int64
 24   FLAG_WORK_PHONE              307511 non-null  int64
 25   FLAG_CONT_MOBILE             307511 non-null  int64
 26   FLAG_PHONE                   307511 non-null  int64
 27   FLAG_EMAIL                   307511 non-null  int64
 28   OCCUPATION_TYPE              211120 non-null  object
 29   CNT_FAM_MEMBERS              307509 non-null  float64
 30   REGION_RATING_CLIENT         307511 non-null  int64
 31   REGION_RATING_CLIENT_W_CITY  307511 non-null  int64
 32   WEEKDAY_APPR_PROCESS_START   307511 non-null  object
 33   HOUR_APPR_PROCESS_START      307511 non-null  int64
 34   REG_REGION_NOT_LIVE_REGION   307511 non-null  int64
 35   REG_REGION_NOT_WORK_REGION   307511 non-null  int64
 36   LIVE_REGION_NOT_WORK_REGION  307511 non-null  int64
 37   REG_CITY_NOT_LIVE_CITY       307511 non-null  int64
 38   REG_CITY_NOT_WORK_CITY       307511 non-null  int64
 39   LIVE_CITY_NOT_WORK_CITY      307511 non-null  int64
 40   ORGANIZATION_TYPE            307511 non-null  object
 41   EXT_SOURCE_1                 134133 non-null  float64
 42   EXT_SOURCE_2                 306851 non-null  float64
 43   EXT_SOURCE_3                 246546 non-null  float64
 44   APARTMENTS_AVG               151450 non-null  float64
 45   BASEMENTAREA_AVG             127568 non-null  float64
```

```
 46   YEARS_BEGINEXPLUATATION_AVG   157504 non-null  float64
 47   YEARS_BUILD_AVG               103023 non-null  float64
 48   COMMONAREA_AVG                92646 non-null   float64
 49   ELEVATORS_AVG                 143620 non-null  float64
 50   ENTRANCES_AVG                 152683 non-null  float64
 51   FLOORSMAX_AVG                 154491 non-null  float64
 52   FLOORSMIN_AVG                 98869 non-null   float64
 53   LANDAREA_AVG                  124921 non-null  float64
 54   LIVINGAPARTMENTS_AVG          97312 non-null   float64
 55   LIVINGAREA_AVG                153161 non-null  float64
 56   NONLIVINGAPARTMENTS_AVG       93997 non-null   float64
 57   NONLIVINGAREA_AVG             137829 non-null  float64
 58   APARTMENTS_MODE               151450 non-null  float64
 59   BASEMENTAREA_MODE             127568 non-null  float64
 60   YEARS_BEGINEXPLUATATION_MODE  157504 non-null  float64
 61   YEARS_BUILD_MODE              103023 non-null  float64
 62   COMMONAREA_MODE               92646 non-null   float64
 63   ELEVATORS_MODE                143620 non-null  float64
 64   ENTRANCES_MODE                152683 non-null  float64
 65   FLOORSMAX_MODE                154491 non-null  float64
 66   FLOORSMIN_MODE                98869 non-null   float64
 67   LANDAREA_MODE                 124921 non-null  float64
 68   LIVINGAPARTMENTS_MODE         97312 non-null   float64
 69   LIVINGAREA_MODE               153161 non-null  float64
 70   NONLIVINGAPARTMENTS_MODE      93997 non-null   float64
 71   NONLIVINGAREA_MODE            137829 non-null  float64
 72   APARTMENTS_MEDI               151450 non-null  float64
 73   BASEMENTAREA_MEDI             127568 non-null  float64
 74   YEARS_BEGINEXPLUATATION_MEDI  157504 non-null  float64
 75   YEARS_BUILD_MEDI              103023 non-null  float64
 76   COMMONAREA_MEDI               92646 non-null   float64
 77   ELEVATORS_MEDI                143620 non-null  float64
 78   ENTRANCES_MEDI                152683 non-null  float64
 79   FLOORSMAX_MEDI                154491 non-null  float64
 80   FLOORSMIN_MEDI                98869 non-null   float64
 81   LANDAREA_MEDI                 124921 non-null  float64
 82   LIVINGAPARTMENTS_MEDI         97312 non-null   float64
 83   LIVINGAREA_MEDI               153161 non-null  float64
 84   NONLIVINGAPARTMENTS_MEDI      93997 non-null   float64
 85   NONLIVINGAREA_MEDI            137829 non-null  float64
 86   FONDKAPREMONT_MODE            97216 non-null   object
 87   HOUSETYPE_MODE                153214 non-null  object
 88   TOTALAREA_MODE                159080 non-null  float64
 89   WALLSMATERIAL_MODE            151170 non-null  object
 90   EMERGENCYSTATE_MODE           161756 non-null  object
 91   OBS_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 92   DEF_30_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 93   OBS_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 94   DEF_60_CNT_SOCIAL_CIRCLE      306490 non-null  float64
 95   DAYS_LAST_PHONE_CHANGE        307510 non-null  float64
 96   FLAG_DOCUMENT_2               307511 non-null  int64
 97   FLAG_DOCUMENT_3               307511 non-null  int64
 98   FLAG_DOCUMENT_4               307511 non-null  int64
 99   FLAG_DOCUMENT_5               307511 non-null  int64
 100  FLAG_DOCUMENT_6               307511 non-null  int64
 101  FLAG_DOCUMENT_7               307511 non-null  int64
 102  FLAG_DOCUMENT_8               307511 non-null  int64
 103  FLAG_DOCUMENT_9               307511 non-null  int64
 104  FLAG_DOCUMENT_10              307511 non-null  int64
 105  FLAG_DOCUMENT_11              307511 non-null  int64
 106  FLAG_DOCUMENT_12              307511 non-null  int64
 107  FLAG_DOCUMENT_13              307511 non-null  int64
 108  FLAG_DOCUMENT_14              307511 non-null  int64
 109  FLAG_DOCUMENT_15              307511 non-null  int64
 110  FLAG_DOCUMENT_16              307511 non-null  int64
 111  FLAG_DOCUMENT_17              307511 non-null  int64
 112  FLAG_DOCUMENT_18              307511 non-null  int64
 113  FLAG_DOCUMENT_19              307511 non-null  int64
 114  FLAG_DOCUMENT_20              307511 non-null  int64
 115  FLAG_DOCUMENT_21              307511 non-null  int64
 116  AMT_REQ_CREDIT_BUREAU_HOUR    265992 non-null  float64
 117  AMT_REQ_CREDIT_BUREAU_DAY     265992 non-null  float64
 118  AMT_REQ_CREDIT_BUREAU_WEEK    265992 non-null  float64
 119  AMT_REQ_CREDIT_BUREAU_MON     265992 non-null  float64
 120  AMT_REQ_CREDIT_BUREAU_QRT     265992 non-null  float64
 121  AMT_REQ_CREDIT_BUREAU_YEAR    265992 non-null  float64
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

In [7]:
```
df.dtypes
```

Out[7]:
```
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
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
dtype: object
```

Converting days_birth to age

In [8]:
```
df["AGE"]=df["DAYS_BIRTH"]/(-365)
```

In [11]:
```
## creating age groups
## create the bucket <30,30-40,40-50,50-60,60+
df["AGE_GROUP"]=pd.cut(df.AGE,bins=[0,30,40,50,60,9999],labels=["<30","30-40","40-50","50-60","60+"])
```

Converting Days Employed to Years

In [12]:
```
df["DAYS_EMPLOYED"].value_counts()
```

Out[12]:
```
DAYS_EMPLOYED
 365243    55374
-200         156
-224         152
-230         151
-199         151
           ...
-13961         1
-11827         1
-10176         1
-9459          1
-8694          1
Name: count, Length: 12574, dtype: int64
```

In [13]:
```
df["DAYS_EMPLOYED"]=df["DAYS_EMPLOYED"].replace(365243,np.NaN)
```

In [14]:
```
df["DAYS_EMPLOYED"].value_counts()
```

Out[14]:
```
DAYS_EMPLOYED
-200.0      156
-224.0      152
-199.0      151
-230.0      151
-212.0      150
           ...
-13961.0      1
-11827.0      1
-10176.0      1
-9459.0       1
-8694.0       1
Name: count, Length: 12573, dtype: int64
```

In [15]:
```
## Creating years bucket
df["YEARS_EMPLOYED_GRP"]=pd.cut(df["DAYS_EMPLOYED"],bins=[0,5,10,15,20,9999],labels=["0-5","5-10","10-15","15-2
```

In [16]:
```
## Converting days registration to years
df["DAYS_REGISTRATION"].value_counts()
```

Out[16]:
```
DAYS_REGISTRATION
-1.0        113
-7.0         98
-6.0         96
-4.0         92
-2.0         92
            ...
-15581.0      1
-15031.0      1
-14804.0      1
-15008.0      1
-14798.0      1
Name: count, Length: 15688, dtype: int64
```

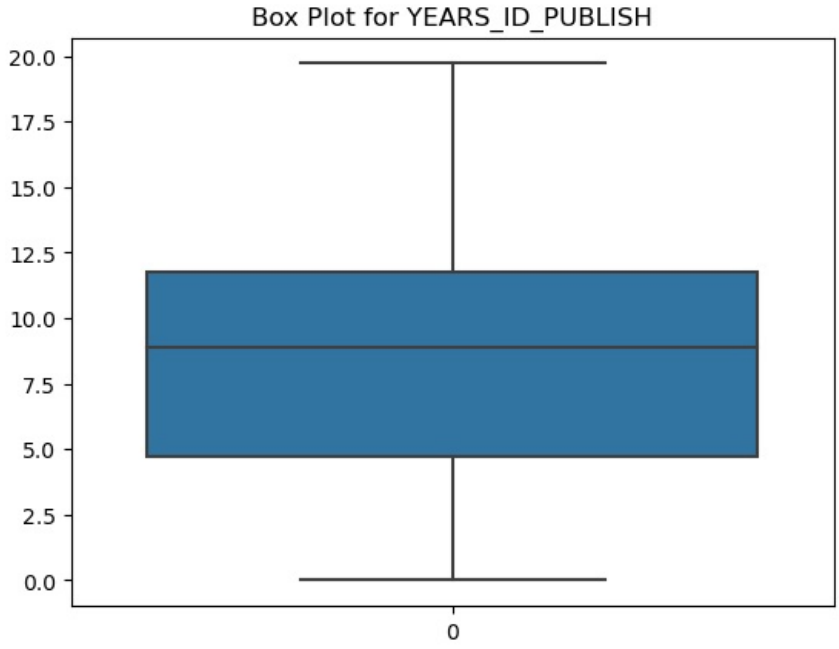
In [17]:
```
df["REGISTRATION_YEARS"]=df["DAYS_REGISTRATION"]/(-365)
```

In [18]:
```
df["REGISTRATION_YEARS"]
```

Out[18]:
```
0          9.994521
1          3.249315
2         11.671233
3         26.939726
4         11.810959
              ...
307506    23.167123
307507    12.021918
307508    18.457534
307509     7.019178
307510    14.049315
Name: REGISTRATION_YEARS, Length: 307511, dtype: float64
```

Converting days_id_publish to years

In [20]:
```
df["YEARS_ID_PUBLISH"]=df["DAYS_ID_PUBLISH"]/(-365)
# plotting the years_id_publish column
sns.boxplot(df["YEARS_ID_PUBLISH"])
plt.title("Box Plot for YEARS_ID_PUBLISH")
plt.show()
```



Splitting Amt_income_total into buckets for easy analysis

In [21]:
```
df["AMT_INCOME_GROUP"]=pd.cut(df["AMT_INCOME_TOTAL"],bins=[0,100000,200000,300000,400000,500000],labels=["V.LOW
```

In [23]:
```
df["BLN_OWN_CAR"]=df["FLAG_OWN_CAR"].apply(lambda x:1 if(x =="Y") else 0)
```

In [24]:
```
df["BLN_OWN_REALTY"]=df["FLAG_OWN_REALTY"].apply(lambda x:1 if(x =="Y") else 0)
```

Handling missing data

In [25]:
```
#Analysing organization type column
df["ORGANIZATION_TYPE"].value_counts()
```

Out[25]:
```
ORGANIZATION_TYPE
Business Entity Type 3    67992
XNA                       55374
Self-employed             38412
Other                     16683
Medicine                  11193
Business Entity Type 2    10553
Government                10404
School                     8893
Trade: type 7              7831
Kindergarten               6880
Construction               6721
Business Entity Type 1     5984
Transport: type 4          5398
Trade: type 3              3492
Industry: type 9           3368
Industry: type 3           3278
Security                   3247
Housing                    2958
Industry: type 11          2704
Military                   2634
Bank                       2507
Agriculture                2454
Police                     2341
Transport: type 2          2204
Postal                     2157
Security Ministries        1974
Trade: type 2              1900
Restaurant                 1811
Services                   1575
University                 1327
Industry: type 7           1307
Transport: type 3          1187
Industry: type 1           1039
Hotel                       966
Electricity                 950
Industry: type 4            877
Trade: type 6               631
Industry: type 5            599
Insurance                   597
Telecom                     577
Emergency                   560
Industry: type 2            458
Advertising                 429
Realtor                     396
Culture                     379
Industry: type 12           369
Trade: type 1               348
Mobile                      317
Legal Services              305
Cleaning                    260
Transport: type 1           201
Industry: type 6            112
Industry: type 10           109
Religion                     85
Industry: type 13            67
Trade: type 4                64
Trade: type 5                49
Industry: type 8             24
Name: count, dtype: int64
```

Large number of records with value XNA could denote missing values. Hence replacing it with NaN

In [26]:
```
df["ORGANIZATION_TYPE"]= df["ORGANIZATION_TYPE"].replace("XNA",np.NaN)
```

In [30]:
```
df.isnull().sum()/len(df)*100
```

Out[30]:
```
SK_ID_CURR                    0.000000
TARGET                        0.000000
NAME_CONTRACT_TYPE            0.000000
CODE_GENDER                   0.000000
FLAG_OWN_CAR                  0.000000
FLAG_OWN_REALTY               0.000000
CNT_CHILDREN                  0.000000
AMT_INCOME_TOTAL              0.000000
AMT_CREDIT                    0.000000
AMT_ANNUITY                   0.003902
AMT_GOODS_PRICE               0.090403
NAME_TYPE_SUITE               0.420148
NAME_INCOME_TYPE              0.000000
NAME_EDUCATION_TYPE           0.000000
NAME_FAMILY_STATUS            0.000000
NAME_HOUSING_TYPE             0.000000
REGION_POPULATION_RELATIVE    0.000000
DAYS_BIRTH                    0.000000
```

DAYS_EMPLOYED	18.007161
DAYS_REGISTRATION	0.000000
DAYS_ID_PUBLISH	0.000000
OWN_CAR_AGE	65.990810
FLAG_MOBIL	0.000000
FLAG_EMP_PHONE	0.000000
FLAG_WORK_PHONE	0.000000
FLAG_CONT_MOBILE	0.000000
FLAG_PHONE	0.000000
FLAG_EMAIL	0.000000
OCCUPATION_TYPE	31.345545
CNT_FAM_MEMBERS	0.000650
REGION_RATING_CLIENT	0.000000
REGION_RATING_CLIENT_W_CITY	0.000000
WEEKDAY_APPR_PROCESS_START	0.000000
HOUR_APPR_PROCESS_START	0.000000
REG_REGION_NOT_LIVE_REGION	0.000000
REG_REGION_NOT_WORK_REGION	0.000000
LIVE_REGION_NOT_WORK_REGION	0.000000
REG_CITY_NOT_LIVE_CITY	0.000000
REG_CITY_NOT_WORK_CITY	0.000000
LIVE_CITY_NOT_WORK_CITY	0.000000
ORGANIZATION_TYPE	18.007161
EXT_SOURCE_1	56.381073
EXT_SOURCE_2	0.214626
EXT_SOURCE_3	19.825307
APARTMENTS_AVG	50.749729
BASEMENTAREA_AVG	58.515956
YEARS_BEGINEXPLUATATION_AVG	48.781019
YEARS_BUILD_AVG	66.497784
COMMONAREA_AVG	69.872297
ELEVATORS_AVG	53.295980
ENTRANCES_AVG	50.348768
FLOORSMAX_AVG	49.760822
FLOORSMIN_AVG	67.848630
LANDAREA_AVG	59.376738
LIVINGAPARTMENTS_AVG	68.354953
LIVINGAREA_AVG	50.193326
NONLIVINGAPARTMENTS_AVG	69.432963
NONLIVINGAREA_AVG	55.179164
APARTMENTS_MODE	50.749729
BASEMENTAREA_MODE	58.515956
YEARS_BEGINEXPLUATATION_MODE	48.781019
YEARS_BUILD_MODE	66.497784
COMMONAREA_MODE	69.872297
ELEVATORS_MODE	53.295980
ENTRANCES_MODE	50.348768
FLOORSMAX_MODE	49.760822
FLOORSMIN_MODE	67.848630
LANDAREA_MODE	59.376738
LIVINGAPARTMENTS_MODE	68.354953
LIVINGAREA_MODE	50.193326
NONLIVINGAPARTMENTS_MODE	69.432963
NONLIVINGAREA_MODE	55.179164
APARTMENTS_MEDI	50.749729
BASEMENTAREA_MEDI	58.515956
YEARS_BEGINEXPLUATATION_MEDI	48.781019
YEARS_BUILD_MEDI	66.497784
COMMONAREA_MEDI	69.872297
ELEVATORS_MEDI	53.295980
ENTRANCES_MEDI	50.348768
FLOORSMAX_MEDI	49.760822
FLOORSMIN_MEDI	67.848630
LANDAREA_MEDI	59.376738
LIVINGAPARTMENTS_MEDI	68.354953
LIVINGAREA_MEDI	50.193326
NONLIVINGAPARTMENTS_MEDI	69.432963
NONLIVINGAREA_MEDI	55.179164
FONDKAPREMONT_MODE	68.386172
HOUSETYPE_MODE	50.176091
TOTALAREA_MODE	48.268517
WALLSMATERIAL_MODE	50.840783
EMERGENCYSTATE_MODE	47.398304
OBS_30_CNT_SOCIAL_CIRCLE	0.332021
DEF_30_CNT_SOCIAL_CIRCLE	0.332021
OBS_60_CNT_SOCIAL_CIRCLE	0.332021
DEF_60_CNT_SOCIAL_CIRCLE	0.332021
DAYS_LAST_PHONE_CHANGE	0.000325
FLAG_DOCUMENT_2	0.000000
FLAG_DOCUMENT_3	0.000000
FLAG_DOCUMENT_4	0.000000
FLAG_DOCUMENT_5	0.000000
FLAG_DOCUMENT_6	0.000000
FLAG_DOCUMENT_7	0.000000
FLAG_DOCUMENT_8	0.000000
FLAG_DOCUMENT_9	0.000000
FLAG_DOCUMENT_10	0.000000
FLAG_DOCUMENT_11	0.000000
FLAG_DOCUMENT_12	0.000000

```
FLAG_DOCUMENT_13                0.000000
FLAG_DOCUMENT_14                0.000000
FLAG_DOCUMENT_15                0.000000
FLAG_DOCUMENT_16                0.000000
FLAG_DOCUMENT_17                0.000000
FLAG_DOCUMENT_18                0.000000
FLAG_DOCUMENT_19                0.000000
FLAG_DOCUMENT_20                0.000000
FLAG_DOCUMENT_21                0.000000
AMT_REQ_CREDIT_BUREAU_HOUR     13.501631
AMT_REQ_CREDIT_BUREAU_DAY      13.501631
AMT_REQ_CREDIT_BUREAU_WEEK     13.501631
AMT_REQ_CREDIT_BUREAU_MON      13.501631
AMT_REQ_CREDIT_BUREAU_QRT      13.501631
AMT_REQ_CREDIT_BUREAU_YEAR     13.501631
AGE                             0.000000
AGE_GROUP                       0.000000
YEARS_EMPLOYED_GRP            100.000000
REGISTRATION_YEARS              0.000000
YEARS_ID_PUBLISH                0.000000
AMT_INCOME_GROUP                0.878668
BLN_OWN_CAR                     0.000000
BLN_OWN_REALTY                  0.000000
dtype: float64
```

Keeping a threshold of 40% missing data to remove columns

In [34]:
```
df = df[df.columns[df.isnull().mean()<.4]]
```

In [35]:
```
df.head(20)
```

Out[35]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME
0	100002	1	Cash loans	M	N	Y	0	20
1	100003	0	Cash loans	F	N	N	0	27
2	100004	0	Revolving loans	M	Y	Y	0	6
3	100006	0	Cash loans	F	N	Y	0	13
4	100007	0	Cash loans	M	N	Y	0	12
5	100008	0	Cash loans	M	N	Y	0	9
6	100009	0	Cash loans	F	Y	Y	1	17
7	100010	0	Cash loans	M	Y	Y	0	36
8	100011	0	Cash loans	F	N	Y	0	11
9	100012	0	Revolving loans	M	N	Y	0	13
10	100014	0	Cash loans	F	N	Y	1	11
11	100015	0	Cash loans	F	N	Y	0	3
12	100016	0	Cash loans	F	N	Y	0	6
13	100017	0	Cash loans	M	Y	N	1	22
14	100018	0	Cash loans	F	N	Y	0	18
15	100019	0	Cash loans	M	Y	Y	0	15
16	100020	0	Cash loans	M	N	N	0	10
17	100021	0	Revolving loans	F	N	Y	1	8
18	100022	0	Revolving loans	F	N	Y	0	11
19	100023	0	Cash loans	F	N	Y	1	9

In [36]:
```
df.shape
```

Out[36]:
```
(307511, 80)
```

In [37]:
```
#get info about existing columns
```

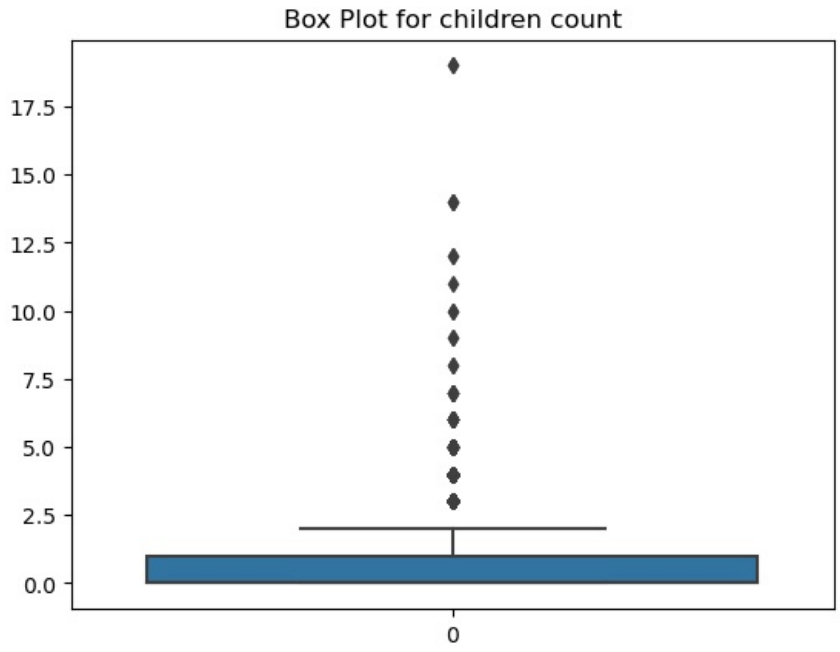
```
df.info(verbose=True, show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Data columns (total 80 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                252137 non-null  float64
 19  DAYS_REGISTRATION            307511 non-null  float64
 20  DAYS_ID_PUBLISH              307511 non-null  int64  
 21  FLAG_MOBIL                   307511 non-null  int64  
 22  FLAG_EMP_PHONE               307511 non-null  int64  
 23  FLAG_WORK_PHONE              307511 non-null  int64  
 24  FLAG_CONT_MOBILE             307511 non-null  int64  
 25  FLAG_PHONE                   307511 non-null  int64  
 26  FLAG_EMAIL                   307511 non-null  int64  
 27  OCCUPATION_TYPE              211120 non-null  object 
 28  CNT_FAM_MEMBERS              307509 non-null  float64
 29  REGION_RATING_CLIENT         307511 non-null  int64  
 30  REGION_RATING_CLIENT_W_CITY  307511 non-null  int64  
 31  WEEKDAY_APPR_PROCESS_START   307511 non-null  object 
 32  HOUR_APPR_PROCESS_START      307511 non-null  int64  
 33  REG_REGION_NOT_LIVE_REGION   307511 non-null  int64  
 34  REG_REGION_NOT_WORK_REGION   307511 non-null  int64  
 35  LIVE_REGION_NOT_WORK_REGION  307511 non-null  int64  
 36  REG_CITY_NOT_LIVE_CITY       307511 non-null  int64  
 37  REG_CITY_NOT_WORK_CITY       307511 non-null  int64  
 38  LIVE_CITY_NOT_WORK_CITY      307511 non-null  int64  
 39  ORGANIZATION_TYPE            252137 non-null  object 
 40  EXT_SOURCE_2                 306851 non-null  float64
 41  EXT_SOURCE_3                 246546 non-null  float64
 42  OBS_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 43  DEF_30_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 44  OBS_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 45  DEF_60_CNT_SOCIAL_CIRCLE     306490 non-null  float64
 46  DAYS_LAST_PHONE_CHANGE       307510 non-null  float64
 47  FLAG_DOCUMENT_2              307511 non-null  int64  
 48  FLAG_DOCUMENT_3              307511 non-null  int64  
 49  FLAG_DOCUMENT_4              307511 non-null  int64  
 50  FLAG_DOCUMENT_5              307511 non-null  int64  
 51  FLAG_DOCUMENT_6              307511 non-null  int64  
 52  FLAG_DOCUMENT_7              307511 non-null  int64  
 53  FLAG_DOCUMENT_8              307511 non-null  int64  
 54  FLAG_DOCUMENT_9              307511 non-null  int64  
 55  FLAG_DOCUMENT_10             307511 non-null  int64  
 56  FLAG_DOCUMENT_11             307511 non-null  int64  
 57  FLAG_DOCUMENT_12             307511 non-null  int64  
 58  FLAG_DOCUMENT_13             307511 non-null  int64  
 59  FLAG_DOCUMENT_14             307511 non-null  int64  
 60  FLAG_DOCUMENT_15             307511 non-null  int64  
 61  FLAG_DOCUMENT_16             307511 non-null  int64  
 62  FLAG_DOCUMENT_17             307511 non-null  int64  
 63  FLAG_DOCUMENT_18             307511 non-null  int64  
 64  FLAG_DOCUMENT_19             307511 non-null  int64  
 65  FLAG_DOCUMENT_20             307511 non-null  int64  
 66  FLAG_DOCUMENT_21             307511 non-null  int64  
 67  AMT_REQ_CREDIT_BUREAU_HOUR   265992 non-null  float64
 68  AMT_REQ_CREDIT_BUREAU_DAY    265992 non-null  float64
 69  AMT_REQ_CREDIT_BUREAU_WEEK   265992 non-null  float64
 70  AMT_REQ_CREDIT_BUREAU_MON    265992 non-null  float64
 71  AMT_REQ_CREDIT_BUREAU_QRT    265992 non-null  float64
 72  AMT_REQ_CREDIT_BUREAU_YEAR   265992 non-null  float64
 73  AGE                          307511 non-null  float64
 74  AGE_GROUP                    307511 non-null  category
 75  REGISTRATION_YEARS           307511 non-null  float64
 76  YEARS_ID_PUBLISH             307511 non-null  float64
 77  AMT_INCOME_GROUP             304809 non-null  category
 78  BLN_OWN_CAR                  307511 non-null  int64  
 79  BLN_OWN_REALTY               307511 non-null  int64  
dtypes: category(2), float64(24), int64(42), object(12)
memory usage: 183.6+ MB
```

Checking Outliers.

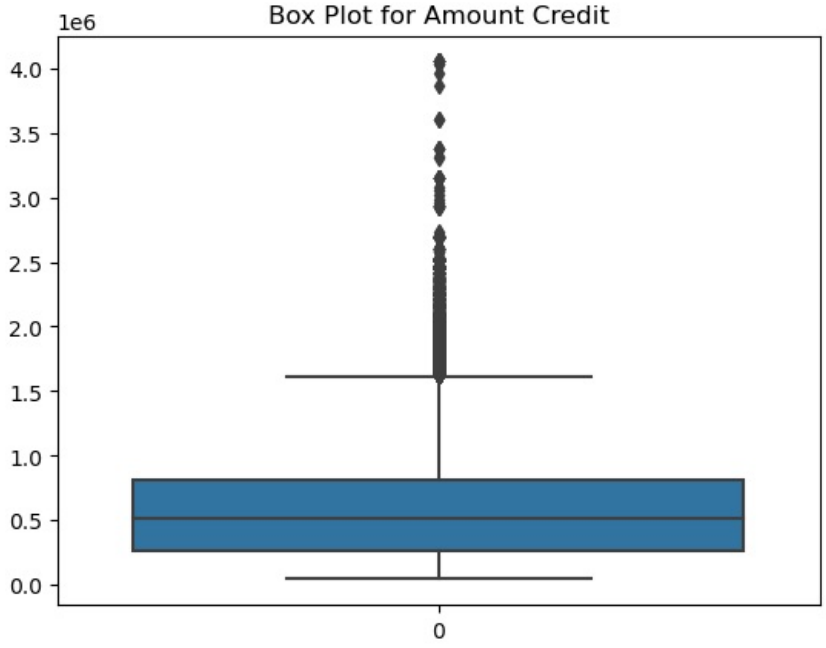
Outliers are data which are different and do not fall into the normal distribution of data. One common visualization used to detect outliers is box plot.

In [38]:
```
sns.boxplot(df["CNT_CHILDREN"])
plt.title("Box Plot for children count")
plt.show()
```



In [39]:
```
# checking for outliers in amount credit
print(df["AMT_CREDIT"].quantile([0.0,0.25,0.5,0.75,0.90,0.95,0.99,1.0]))
sns.boxplot(df["AMT_CREDIT"])
plt.title("Box Plot for Amount Credit")
plt.show()
```

```
0.00      45000.0
0.25     270000.0
0.50     513531.0
0.75     808650.0
0.90    1133748.0
0.95    1350000.0
0.99    1854000.0
1.00    4050000.0
Name: AMT_CREDIT, dtype: float64
```



In [41]:
```
df.columns
```

Out[41]:
```
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', 'FLAG_MOBIL',
       'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE',
       'FLAG_EMAIL', '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', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
       '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_2', 'FLAG_DOCUMENT_3',
       'FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6',
       'FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9',
       'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11', 'FLAG_DOCUMENT_12',
       'FLAG_DOCUMENT_13', 'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15',
       'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17', 'FLAG_DOCUMENT_18',
       'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21',
       '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', 'AGE',
       'AGE_GROUP', 'REGISTRATION_YEARS', 'YEARS_ID_PUBLISH',
       'AMT_INCOME_GROUP', 'BLN_OWN_CAR', 'BLN_OWN_REALTY'],
      dtype='object')
```

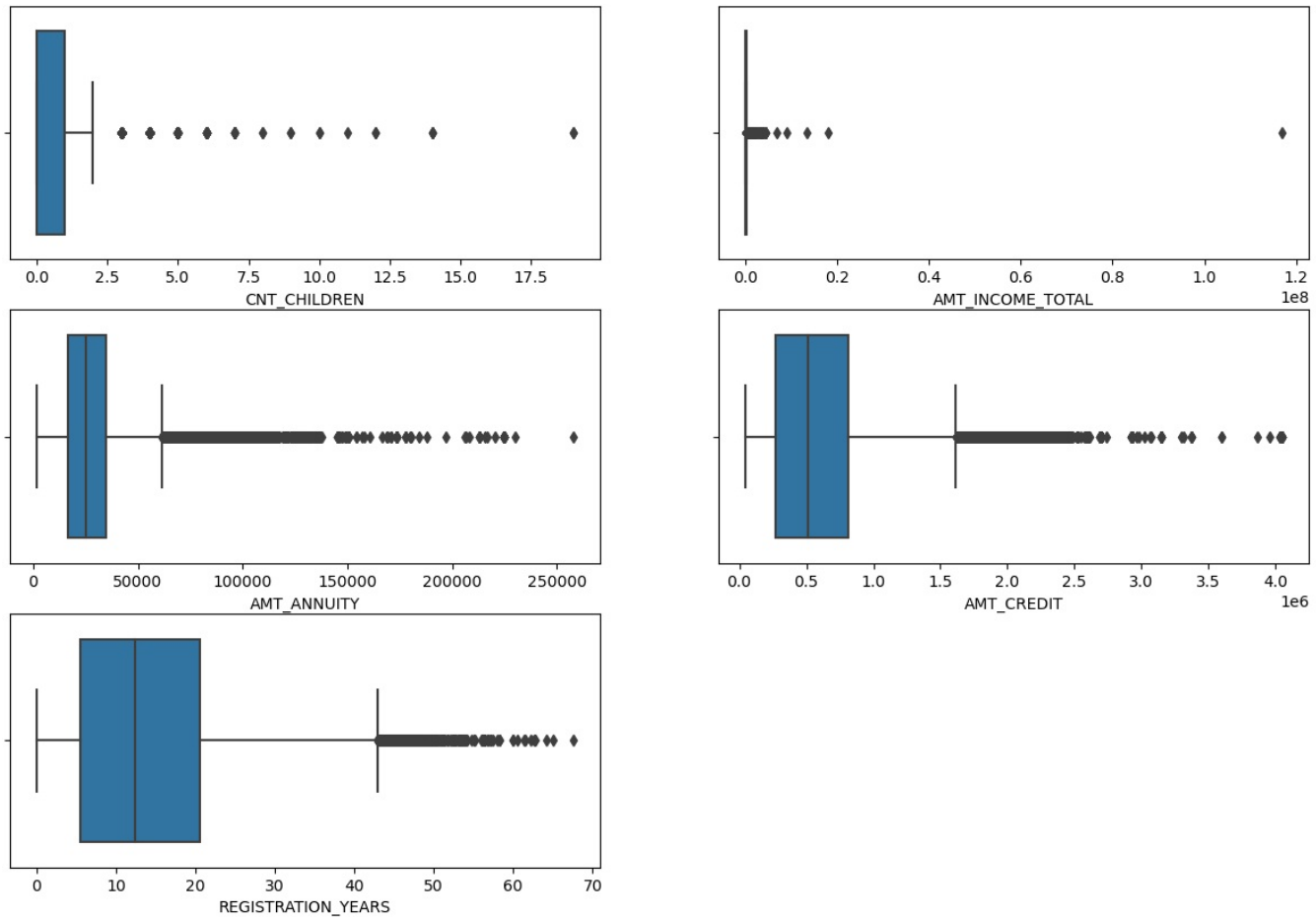
In [42]:
```
df[['SK_ID_CURR', 'TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
       'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL',
       'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE', 'NAME_TYPE_SUITE',
       'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE', 'NAME_FAMILY_STATUS',
       'NAME_HOUSING_TYPE', 'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH',
       'DAYS_EMPLOYED', 'DAYS_REGISTRATION', 'DAYS_ID_PUBLISH', 'FLAG_MOBIL',
       'FLAG_EMP_PHONE', 'FLAG_WORK_PHONE', 'FLAG_CONT_MOBILE', 'FLAG_PHONE',
       'FLAG_EMAIL', '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', 'EXT_SOURCE_2', 'EXT_SOURCE_3',
       '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_2', 'FLAG_DOCUMENT_3',
       'FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6',
       'FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9',
       'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11', 'FLAG_DOCUMENT_12',
       'FLAG_DOCUMENT_13', 'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15',
       'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17', 'FLAG_DOCUMENT_18',
       'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21',
       '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', 'AGE',
       'AGE_GROUP', 'REGISTRATION_YEARS', 'YEARS_ID_PUBLISH',
       'AMT_INCOME_GROUP', 'BLN_OWN_CAR', 'BLN_OWN_REALTY']].describe(percentiles=[.05,.25,.5,.75,.95])
```

Out[42]:

	SK_ID_CURR	TARGET	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	REGION_PO
count	307511.000000	307511.000000	307511.000000	3.075110e+05	3.075110e+05	307499.000000	3.072330e+05	
mean	278180.518577	0.080729	0.417052	1.687979e+05	5.990260e+05	27108.573909	5.383962e+05	
std	102790.175348	0.272419	0.722121	2.371231e+05	4.024908e+05	14493.737315	3.694465e+05	
min	100002.000000	0.000000	0.000000	2.565000e+04	4.500000e+04	1615.500000	4.050000e+04	
5%	117945.500000	0.000000	0.000000	6.750000e+04	1.350000e+05	9000.000000	1.350000e+05	
25%	189145.500000	0.000000	0.000000	1.125000e+05	2.700000e+05	16524.000000	2.385000e+05	
50%	278202.000000	0.000000	0.000000	1.471500e+05	5.135310e+05	24903.000000	4.500000e+05	
75%	367142.500000	0.000000	1.000000	2.025000e+05	8.086500e+05	34596.000000	6.795000e+05	
95%	438427.500000	1.000000	2.000000	3.375000e+05	1.350000e+06	53325.000000	1.305000e+06	
max	456255.000000	1.000000	19.000000	1.170000e+08	4.050000e+06	258025.500000	4.050000e+06	

In [44]:
```
plt.figure(figsize = (15, 10))
plt.subplot(3, 2, 1)
sns.boxplot(x ='CNT_CHILDREN', data = df)
plt.subplot(3, 2, 2)
sns.boxplot(x ='AMT_INCOME_TOTAL', data = df)
plt.subplot(3, 2, 3)
sns.boxplot(x ='AMT_ANNUITY', data = df)
plt.subplot(3, 2, 4)
sns.boxplot(x ='AMT_CREDIT', data = df)
```

```
plt.subplot(3, 2, 5)
sns.boxplot(x ='REGISTRATION_YEARS', data = df)
plt.show()
```



In [45]:
```
df["AMT_INCOME_TOTAL"].describe()
```

Out[45]:
```
count    3.075110e+05
mean     1.687979e+05
std      2.371231e+05
min      2.565000e+04
25%      1.125000e+05
50%      1.471500e+05
75%      2.025000e+05
max      1.170000e+08
Name: AMT_INCOME_TOTAL, dtype: float64
```

In [46]:
```
#Creating bins to convert AMT_INCOME_TOTAL into the categorical values
df["INCOME_BRACKET"]=pd.cut(df["AMT_INCOME_TOTAL"],[0,100000,125000,175000,225000,1000000000],labels=["Very Low
```

In [47]:
```
df["INCOME_BRACKET"]
```

Out[47]:
```
0              High
1         Very High
2          Very Low
3            Medium
4               Low
             ...
307506       Medium
307507     Very Low
307508       Medium
307509       Medium
307510       Medium
Name: INCOME_BRACKET, Length: 307511, dtype: category
Categories (5, object): ['Very Low' < 'Low' < 'Medium' < 'High' < 'Very High']
```

In [48]:
```
df['AMT_CREDIT'].describe()
```

Out[48]:
```
count    3.075110e+05
mean     5.990260e+05
std      4.024908e+05
min      4.500000e+04
25%      2.700000e+05
50%      5.135310e+05
75%      8.086500e+05
max      4.050000e+06
Name: AMT_CREDIT, dtype: float64
```

In [49]:
```
##Creating bins to convert 'AMT_CREDIT' into categorical value

df['CREDIT_BRACKETS']=pd.cut(df['AMT_CREDIT'],[0,200000,500000,800000,1000000,1000000000],labels=['Very Low','L
```

In [50]:
```
df["CREDIT_BRACKETS"]
```

Out[50]:
```
0              Low
1         Very High
2          Very Low
3               Low
4            Medium
             ...
307506          Low
307507          Low
307508       Medium
307509          Low
307510       Medium
Name: CREDIT_BRACKETS, Length: 307511, dtype: category
Categories (5, object): ['Very Low' < 'Low' < 'Medium' < 'High' < 'Very High']
```

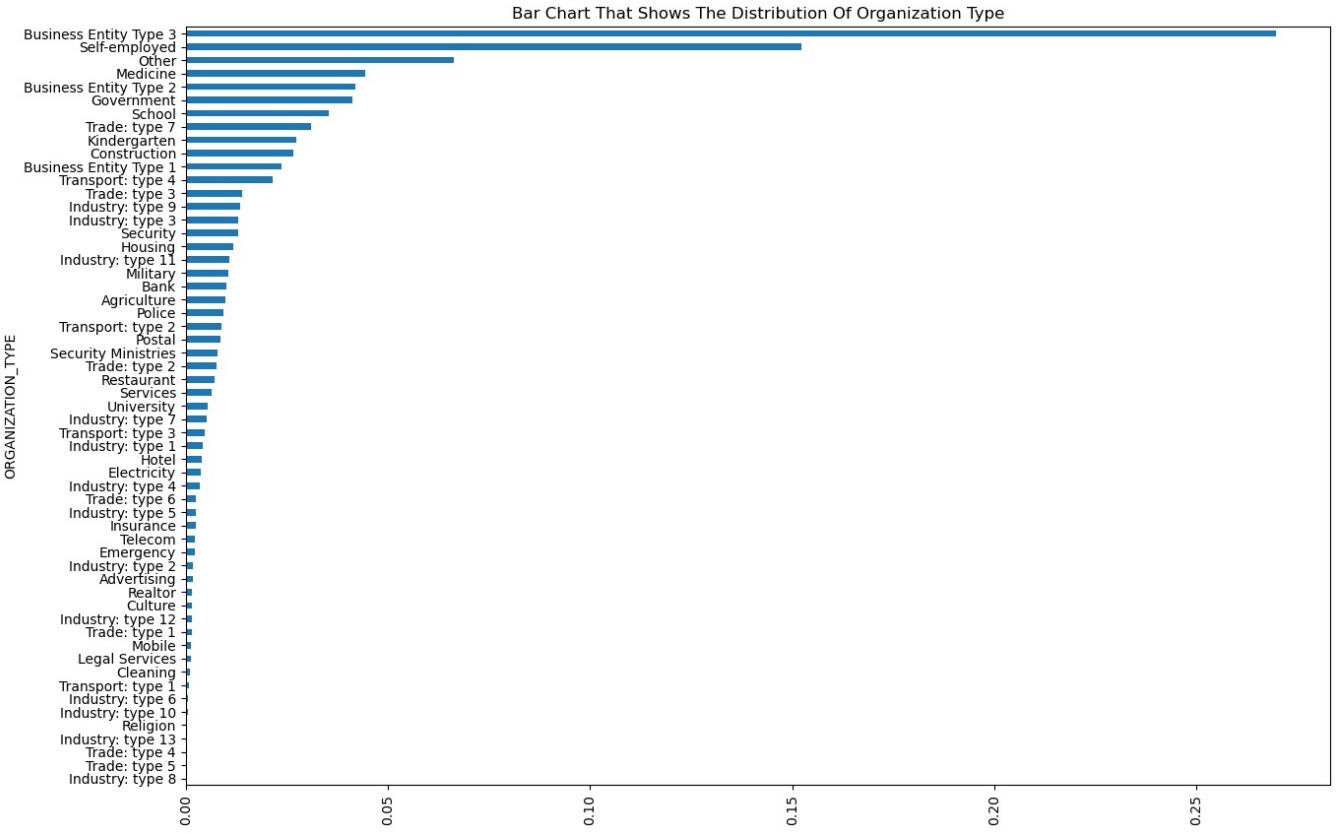
EXPLORATORY DATA ANALYSIS (EDA)

In [51]:
```
print('Percentage of people with payment difficulties : ',100*round(len(df[df.TARGET==1])/len(df),4),'%')
print('Percentage of people with no payment diffiiculties : ',100*round(len(df[df.TARGET==0])/len(df),4),'%')
```

```
Percentage of people with payment difficulties :  8.07 %
Percentage of people with no payment diffiiculties :  91.93 %
```

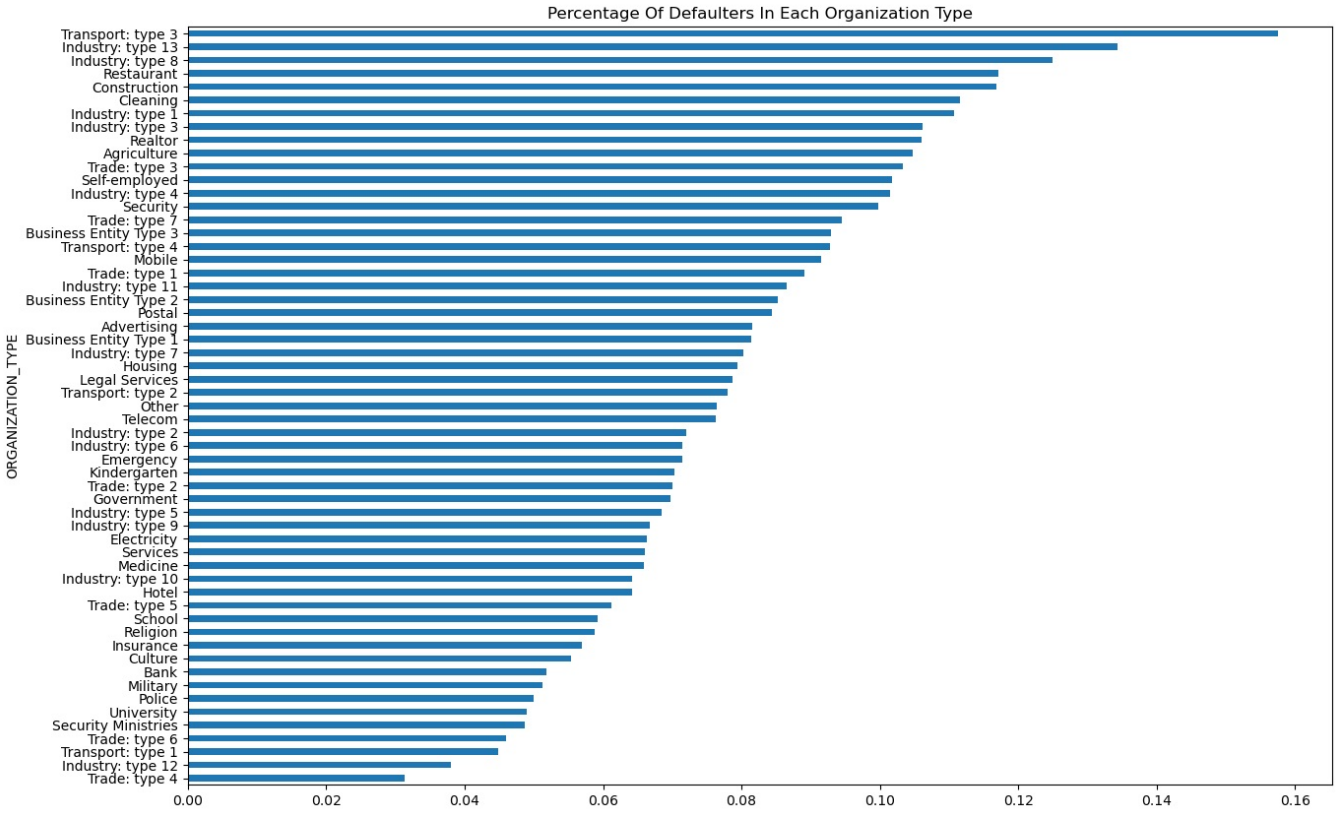
Organization Type

In [56]:
```
plt.figure(figsize=(15,10))
plt.title("Bar Chart That Shows The Distribution Of Organization Type")
df["ORGANIZATION_TYPE"].value_counts(normalize=True).sort_values(ascending=True).plot.barh()
plt.xticks(rotation=90)
plt.show()
```



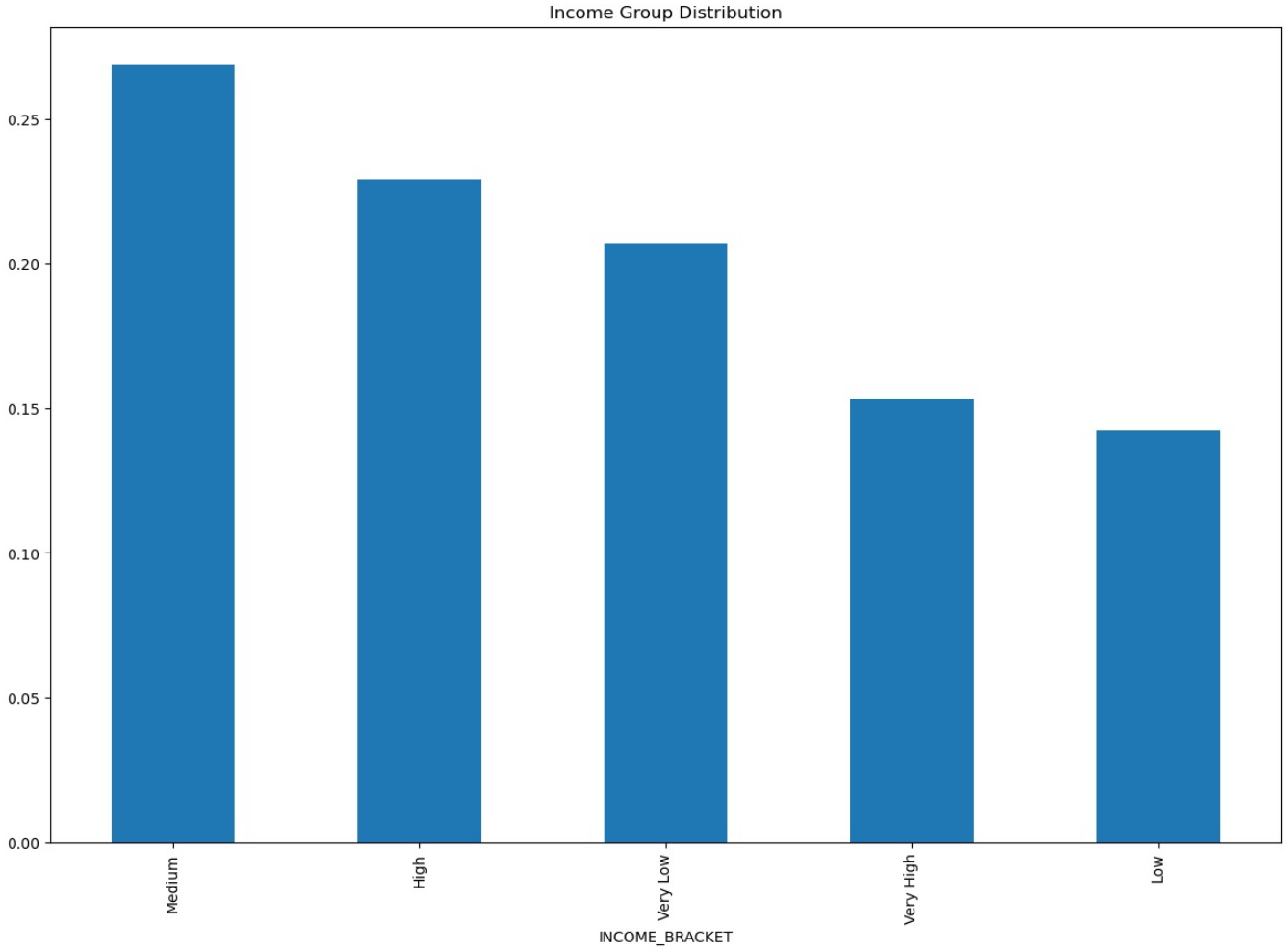
Percentage of defualters in each Organization Type

In [58]:
```
plt.figure(figsize=(15,10))
plt.title("Percentage Of Defaulters In Each Organization Type")
df.groupby("ORGANIZATION_TYPE")["TARGET"].mean().sort_values(ascending=True).plot.barh()
plt.show()
```



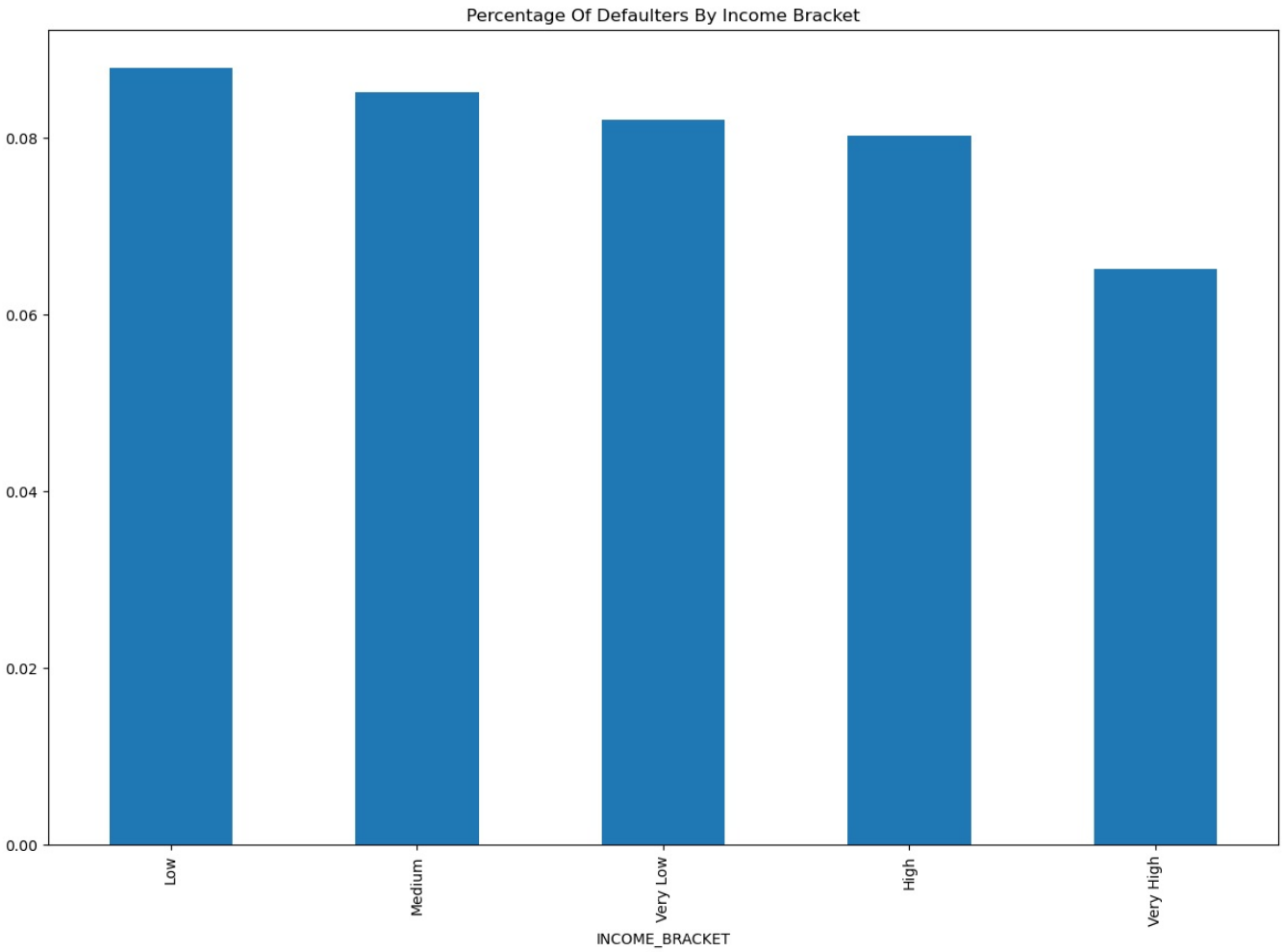
Income Group Distribustion

In [61]:
```
plt.figure(figsize=(15,10))
plt.title("Income Group Distribution")
df["INCOME_BRACKET"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



Percentage of Defaulters by Income Bracket

In [64]:
```
plt.figure(figsize=(15,10))
plt.title("Percentage Of Defaulters By Income Bracket")
df.groupby("INCOME_BRACKET")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



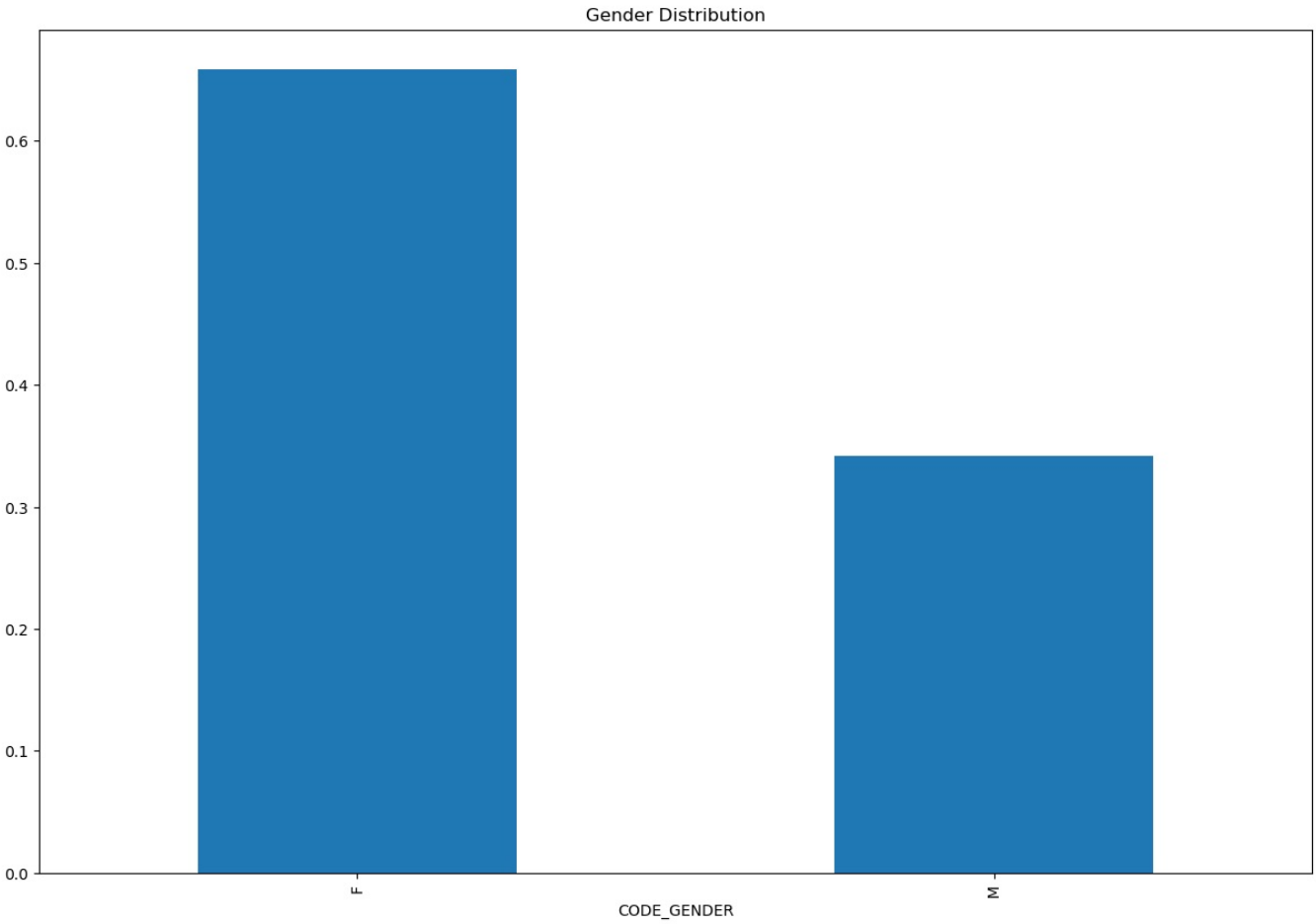
In [71]:
```
df["CODE_GENDER"].value_counts()
```

Out[71]:
```
CODE_GENDER
F      202448
M      105059
XNA         4
Name: count, dtype: int64
```

In [72]:
```
df["CODE_GENDER"]= df["CODE_GENDER"].replace("XNA",np.NaN)
```

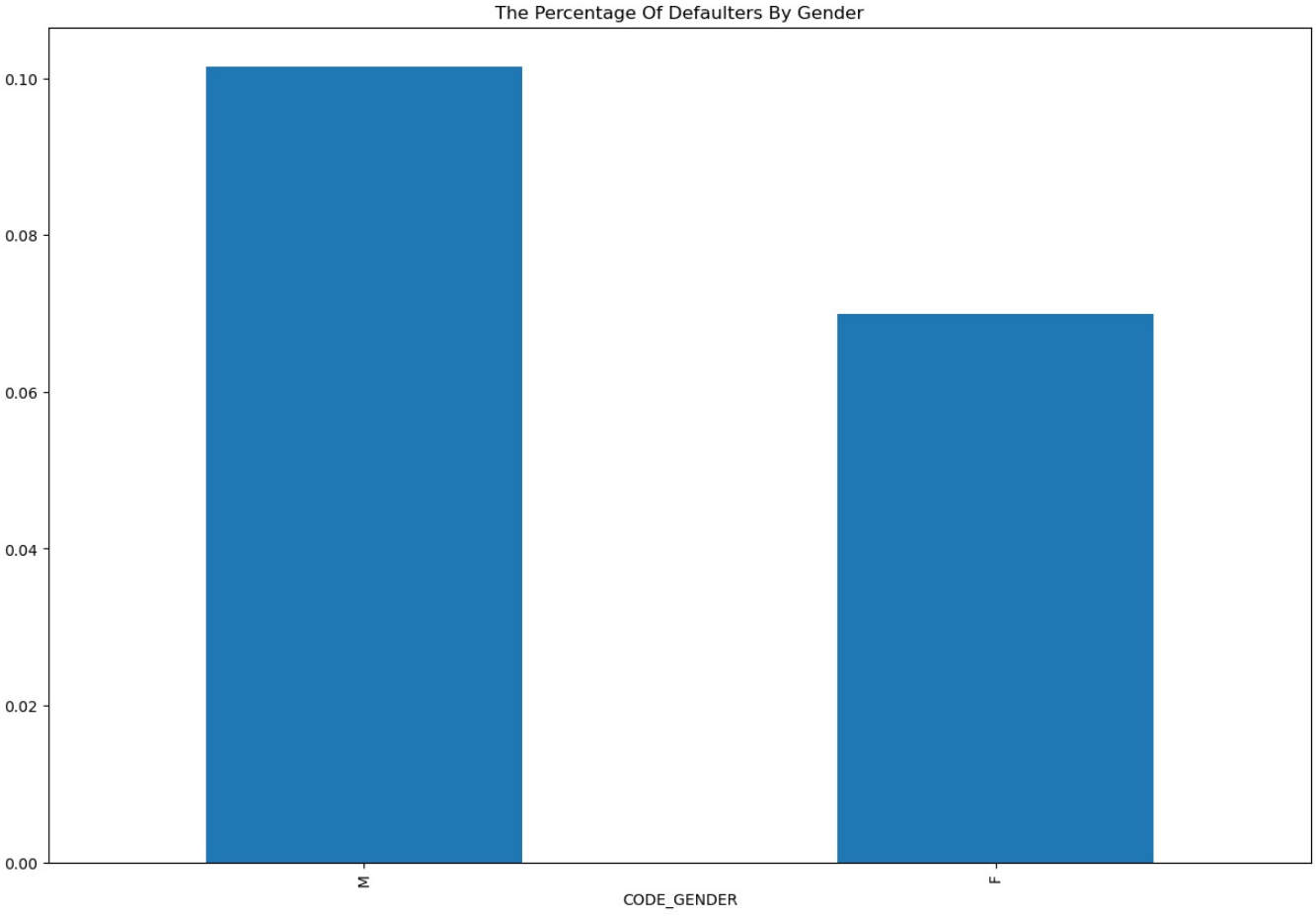
Gender Distribution

In [73]:
```
plt.figure(figsize=(15,10))
plt.title("Gender Distribution")
df["CODE_GENDER"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



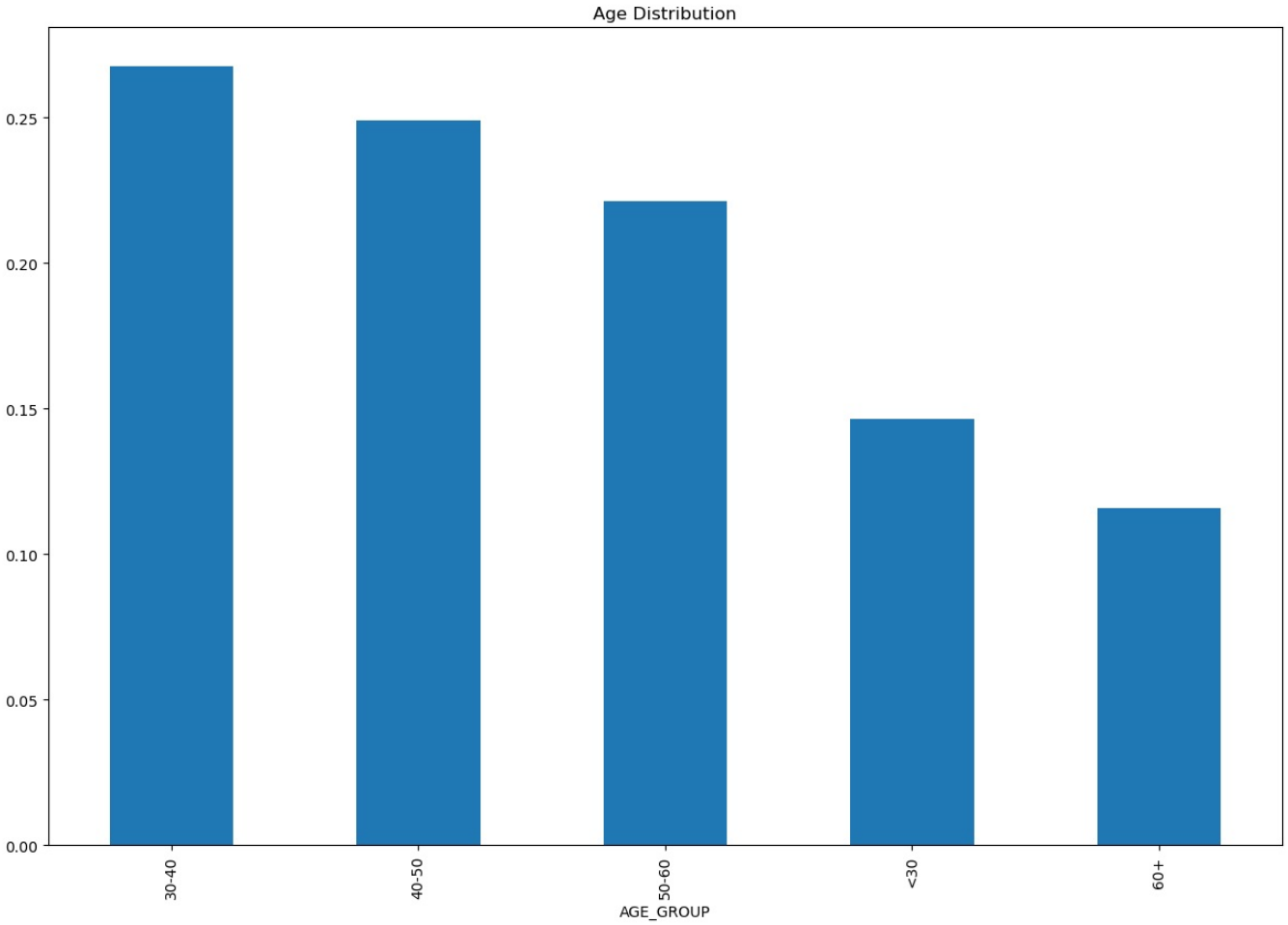
Percentage Of Defaulters By Gender

In [75]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Gender")
df.groupby("CODE_GENDER")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



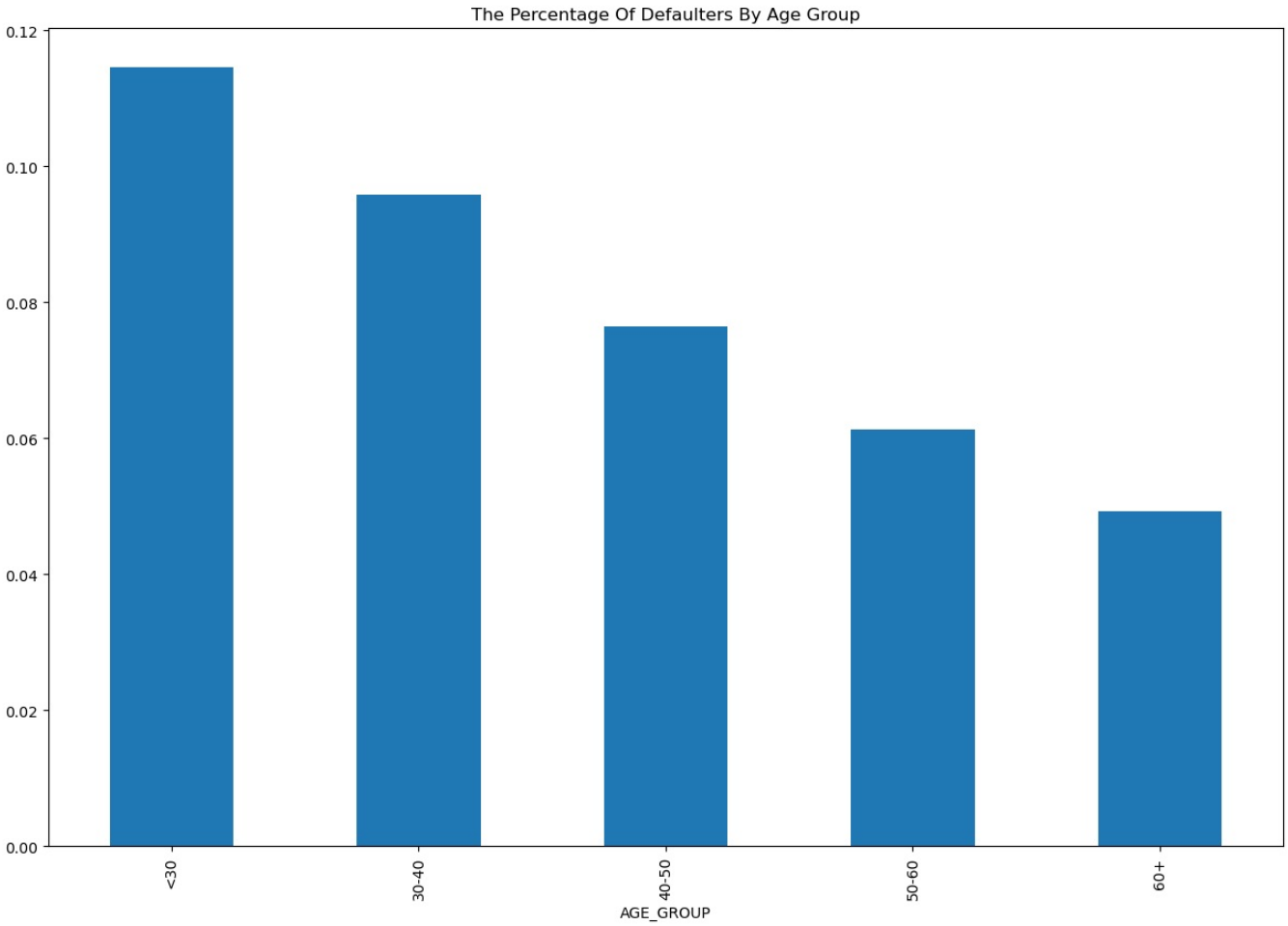
Age Distribution

In [76]:
```
plt.figure(figsize=(15,10))
plt.title("Age Distribution")
df["AGE_GROUP"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



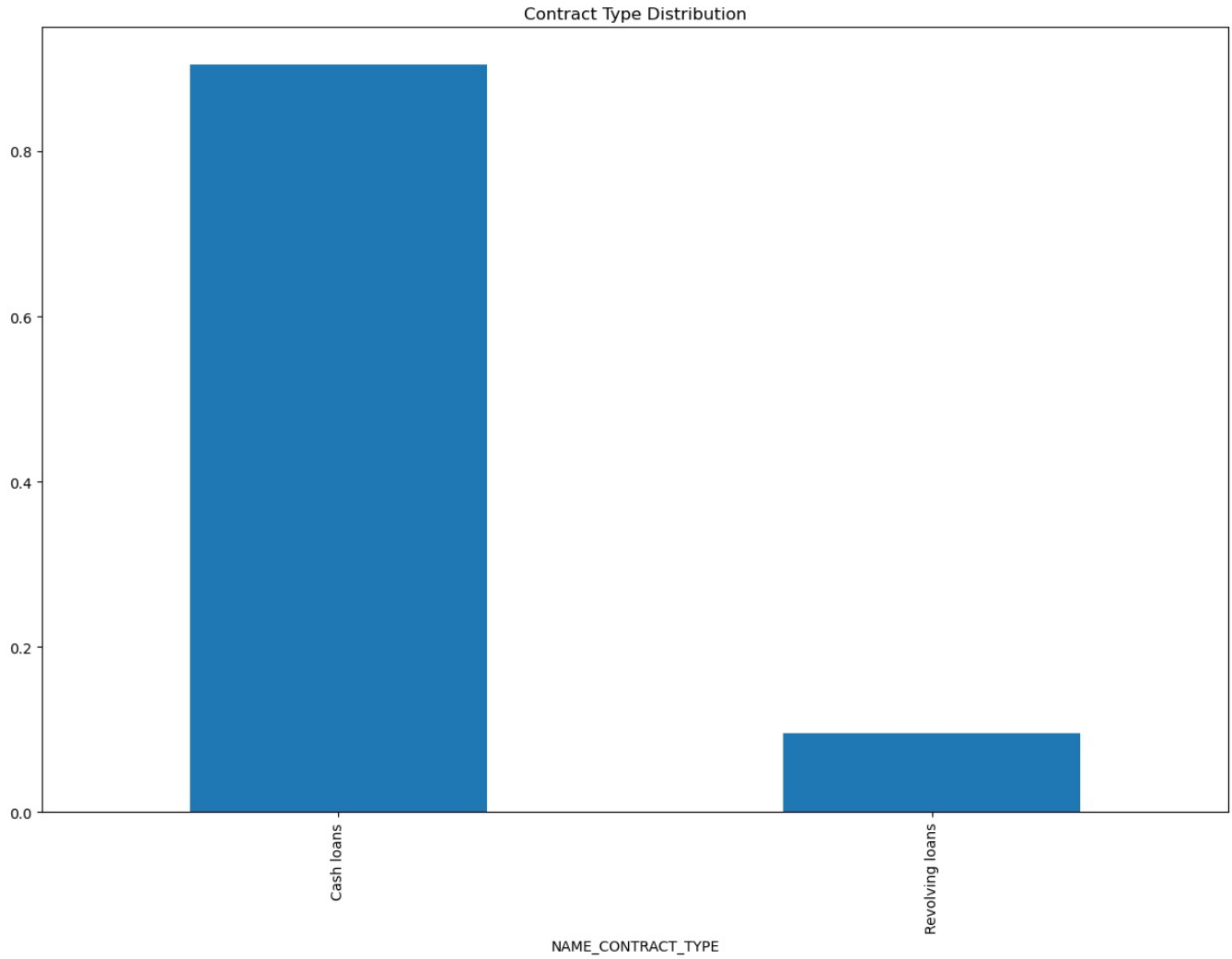
Percentage Of Defaulters By Age Group

In [77]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Age Group")
df.groupby("AGE_GROUP")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



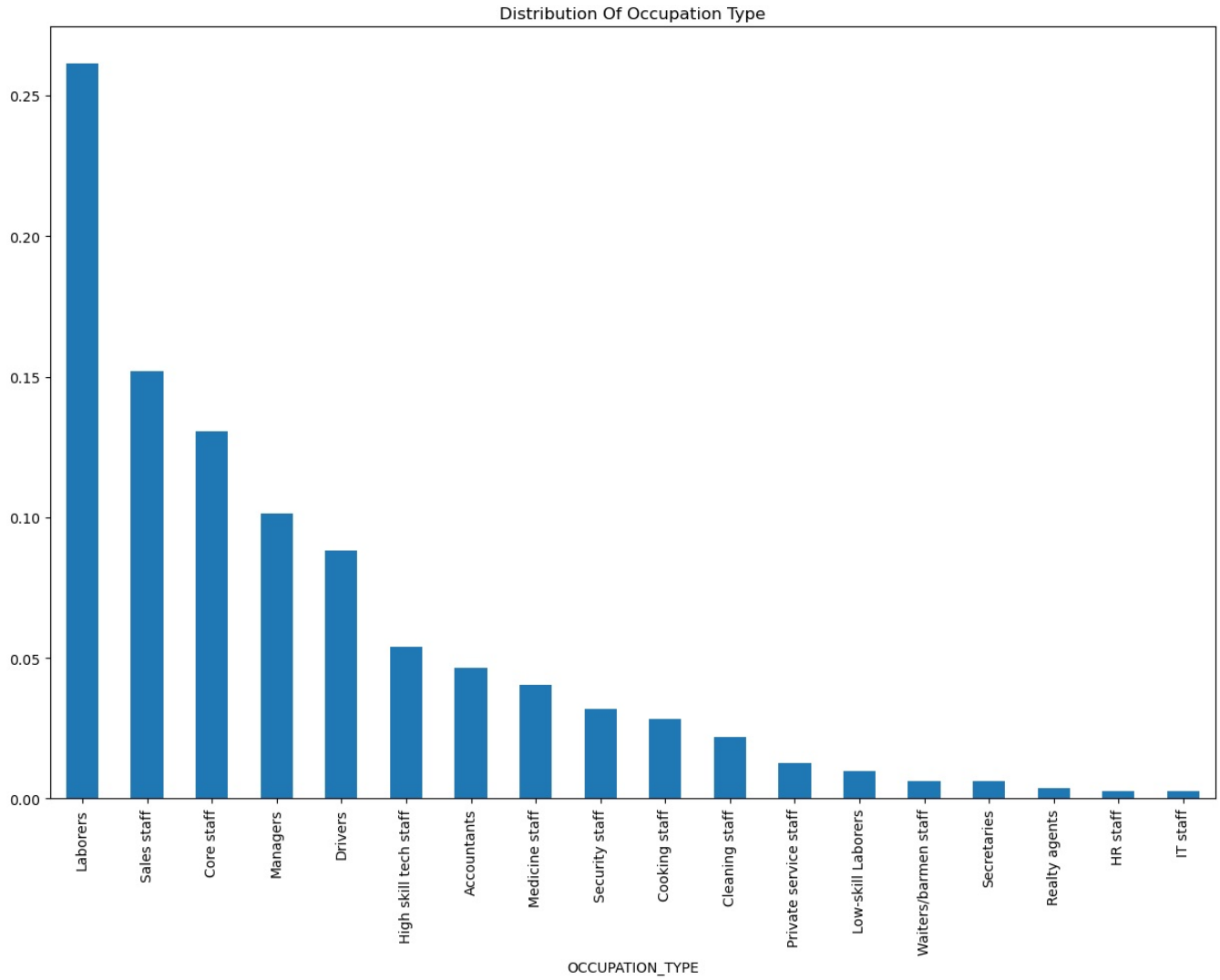
Type Of Loan Distribution

In [80]:
```
plt.figure(figsize=(15,10))
plt.title("Contract Type Distribution")
df["NAME_CONTRACT_TYPE"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



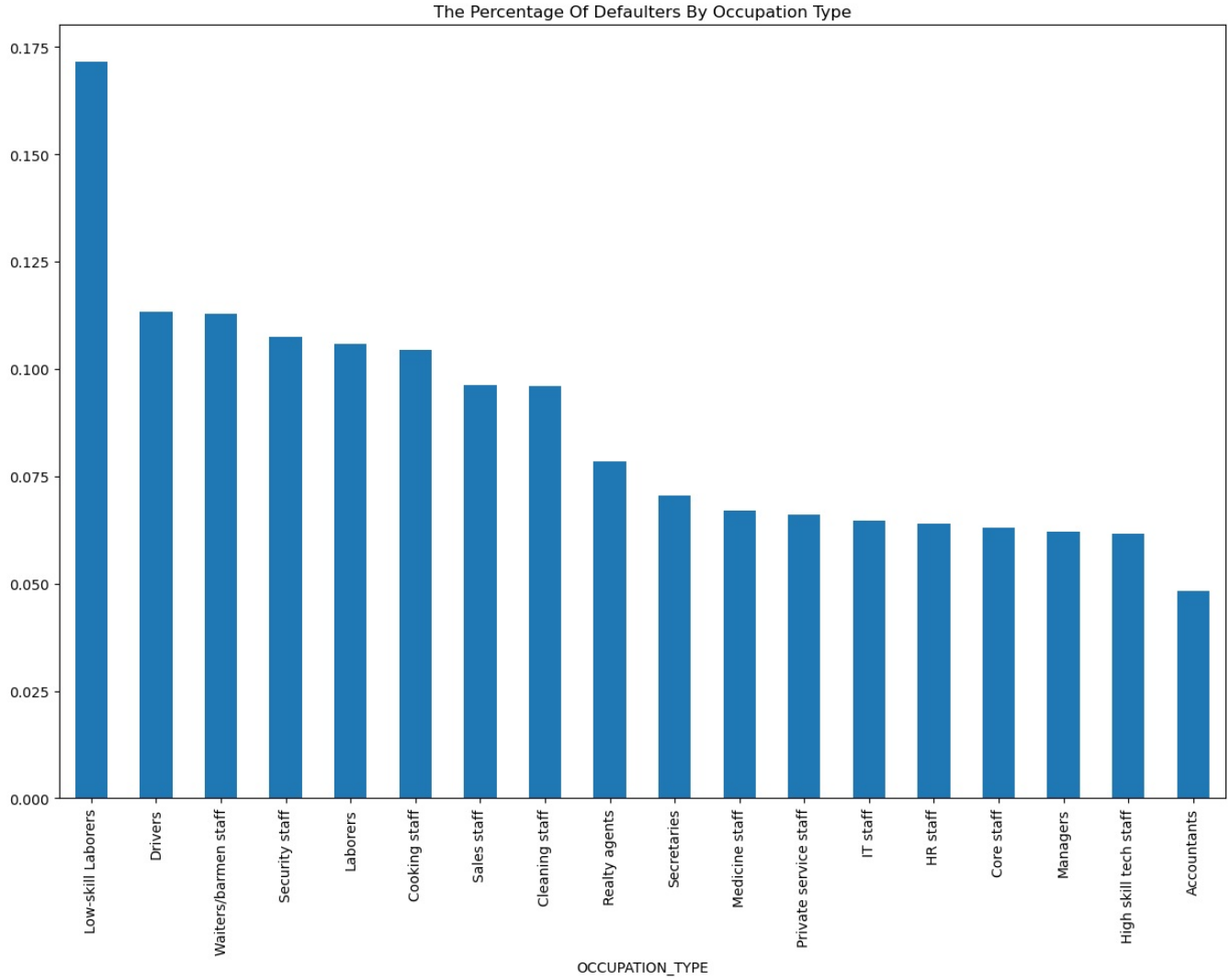
Distribution Of Occupation Type

In [81]:
```
plt.figure(figsize=(15,10))
plt.title("Distribution Of Occupation Type")
df["OCCUPATION_TYPE"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



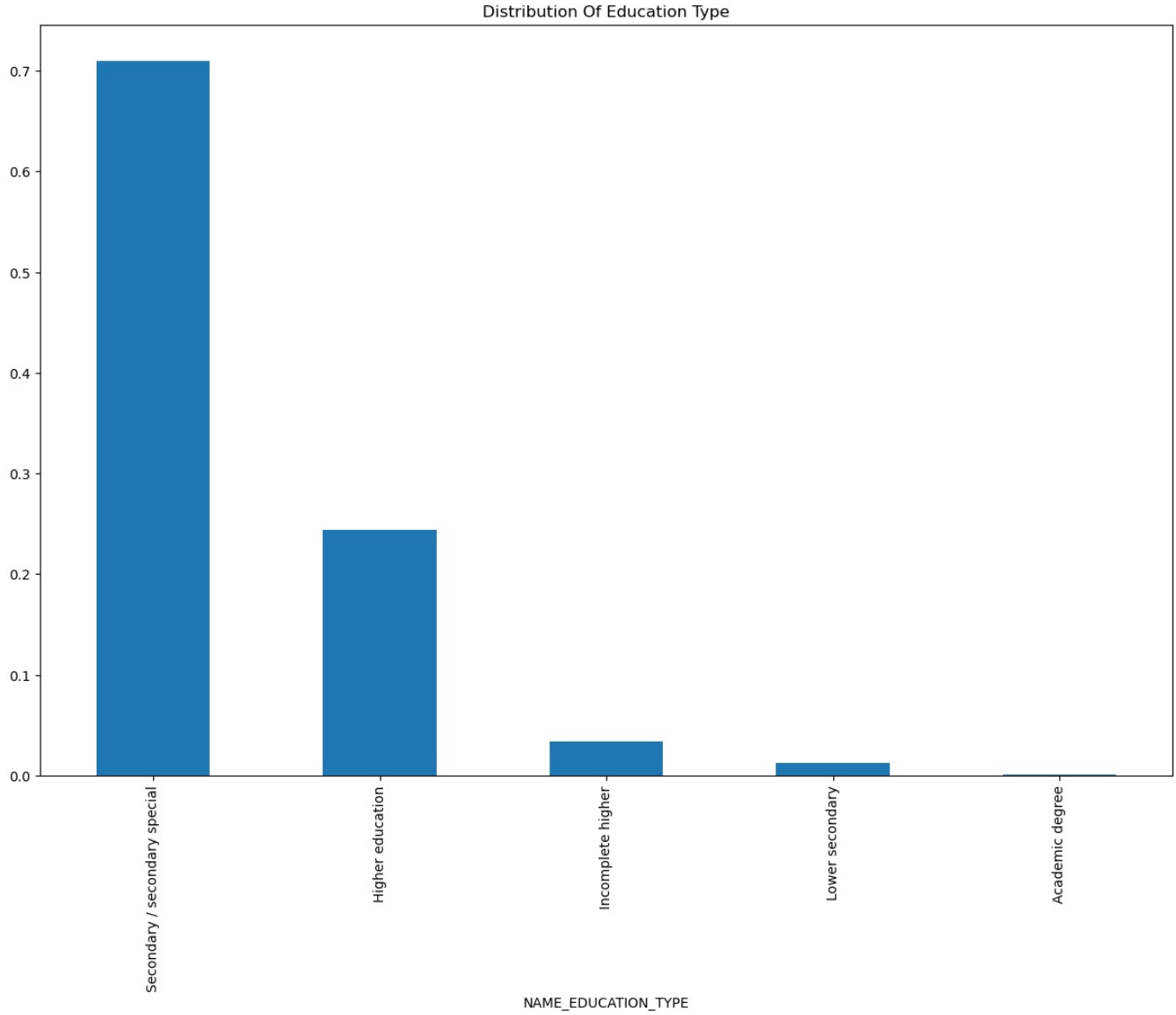
The Percentage Of Defaulters By Occupation Type

In [82]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Occupation Type")
df.groupby("OCCUPATION_TYPE")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```

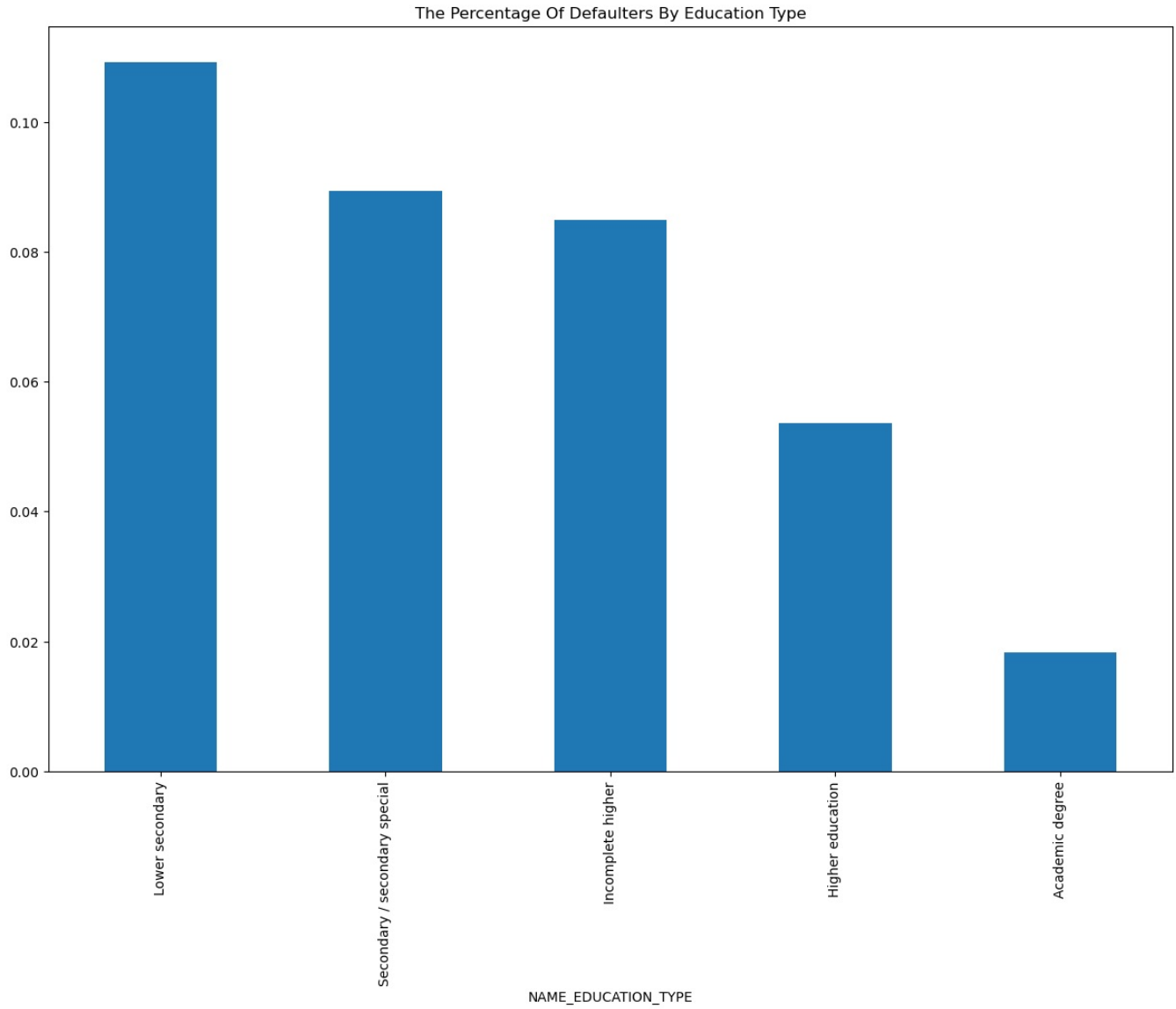


Distribution Of Education Type

In [83]:
```
plt.figure(figsize=(15,10))
plt.title("Distribution Of Education Type")
df["NAME_EDUCATION_TYPE"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



In [84]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Education Type")
df.groupby("NAME_EDUCATION_TYPE")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



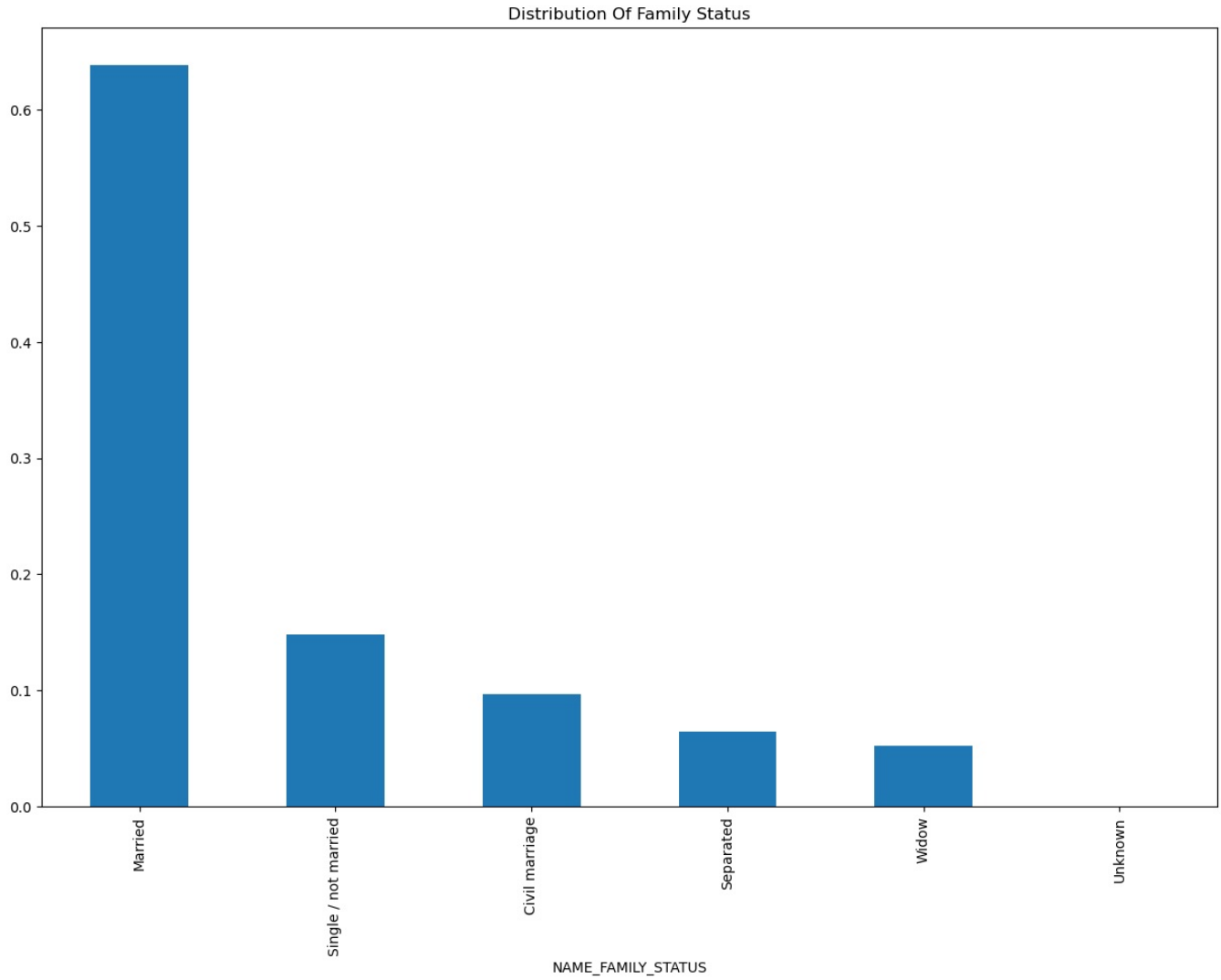
In [88]:
```
df["NAME_FAMILY_STATUS"].value_counts()
```

Out[88]:
```
NAME_FAMILY_STATUS
Married                 196432
Single / not married     45444
Civil marriage           29775
Separated                19770
Widow                    16088
Unknown                      2
Name: count, dtype: int64
```

In [89]:
```
df["NAME_FAMILY_STATUS"]= df["NAME_FAMILY_STATUS"].replace("unknown",np.NaN)
```

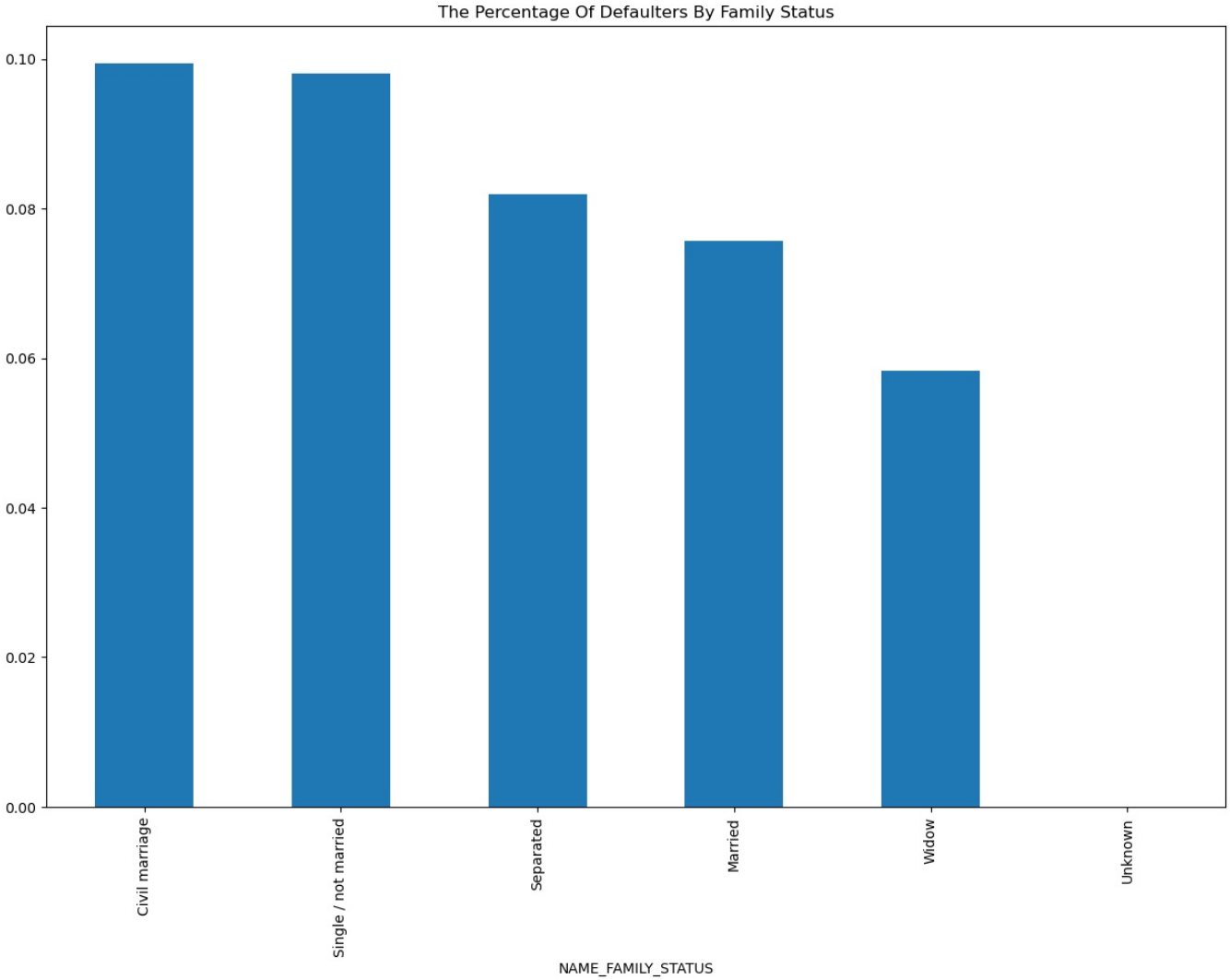
Distribution Of The Family Status

In [19]:
```
plt.figure(figsize=(15,10))
plt.title("Distribution Of Family Status")
df["NAME_FAMILY_STATUS"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```

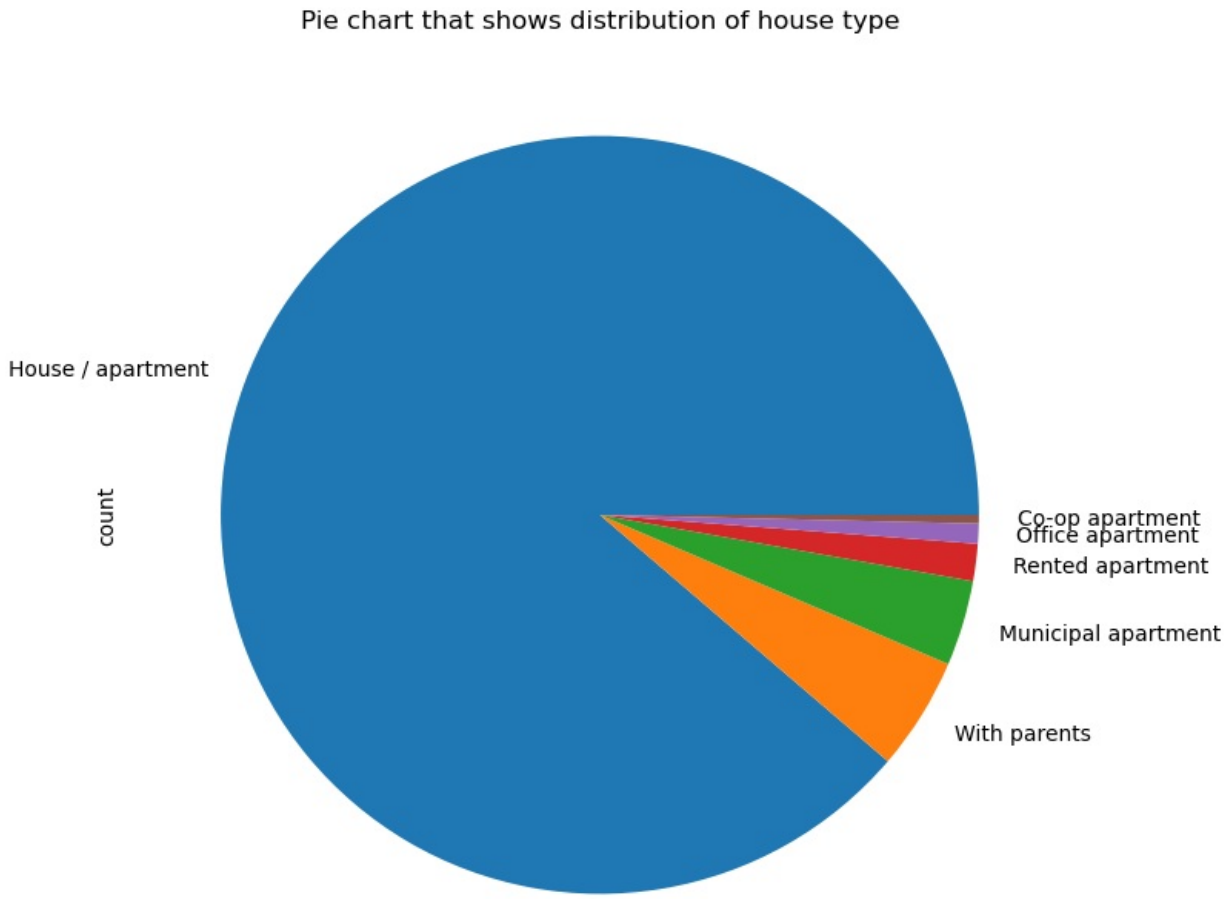


Percentage Of Defualters By Family Status

In [86]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Family Status")
df.groupby("NAME_FAMILY_STATUS")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



In [92]:
```
#Analysing house type
#Plot a pie chart
plt.figure(figsize=(13,8))
plt.title("Pie chart that shows distribution of house type")
df["NAME_HOUSING_TYPE"].value_counts().plot.pie()
plt.show()
```

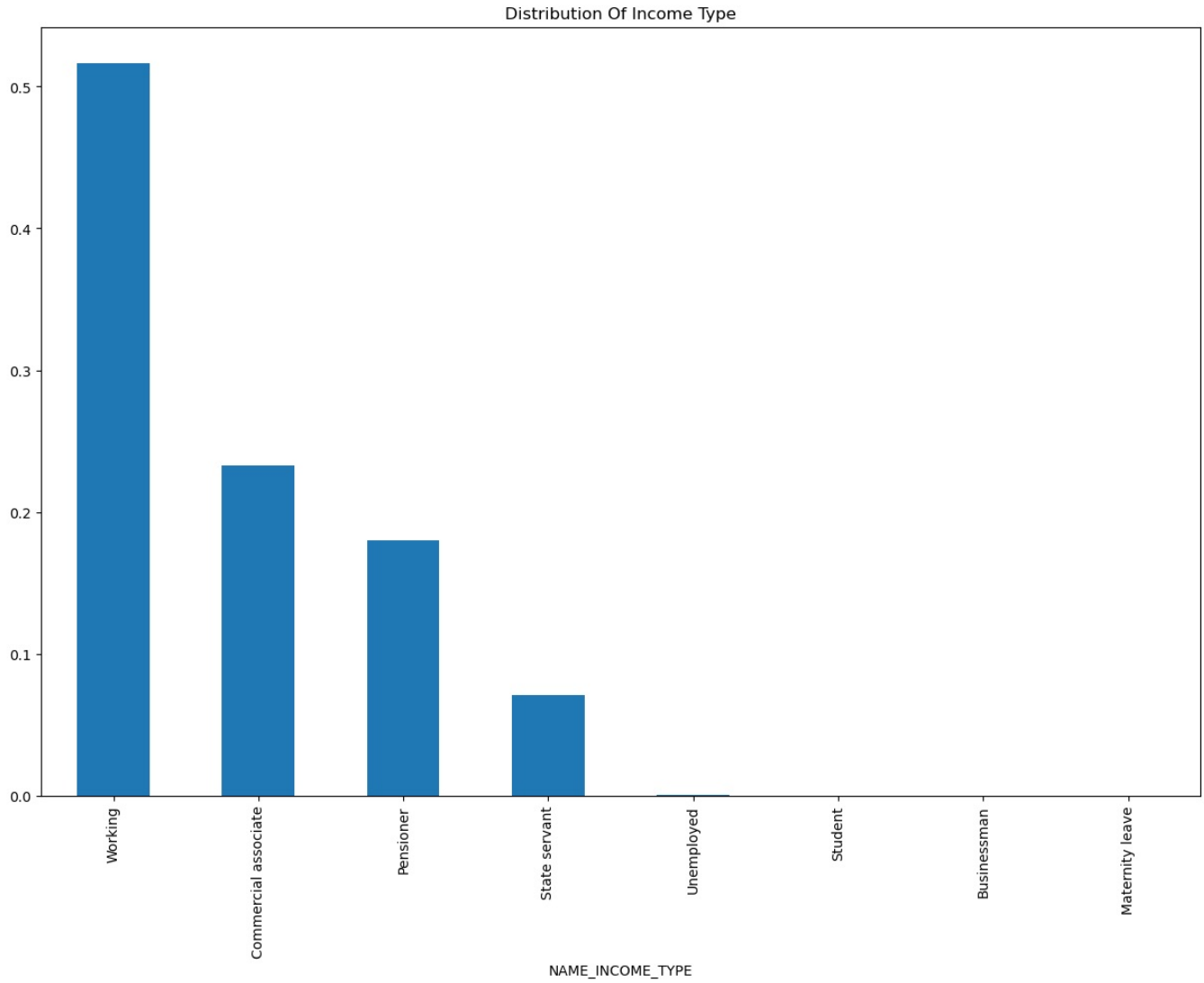


In [93]:
```
df["NAME_INCOME_TYPE"].value_counts()
```

Out[93]:
```
NAME_INCOME_TYPE
Working                 158774
Commercial associate     71617
Pensioner                55362
State servant            21703
Unemployed                  22
Student                     18
Businessman                 10
Maternity leave              5
Name: count, dtype: int64
```

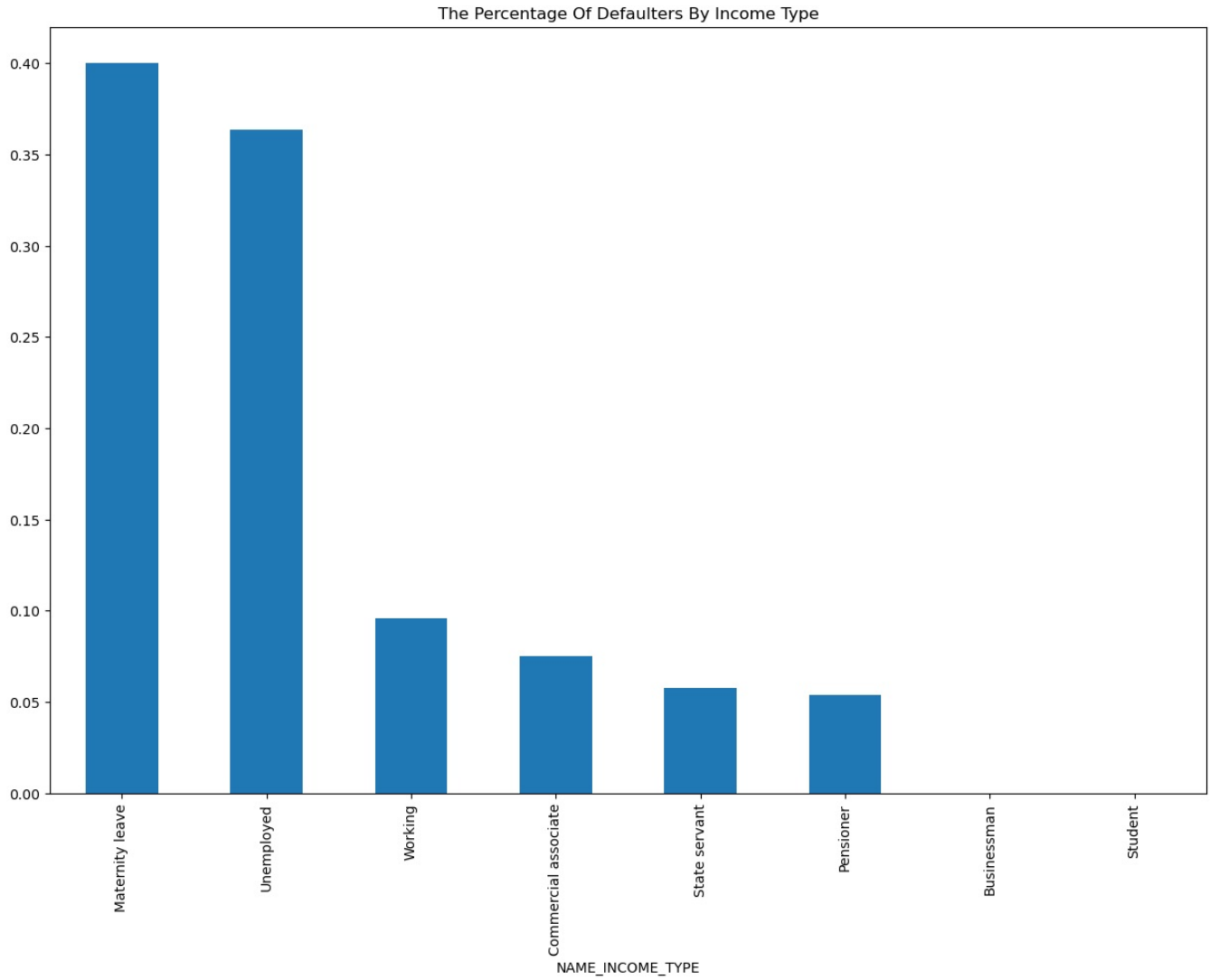
Distribution Of Income Type

In [94]:
```
plt.figure(figsize=(15,10))
plt.title("Distribution Of Income Type")
df["NAME_INCOME_TYPE"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



Percentage Of Defaulters By Income Type

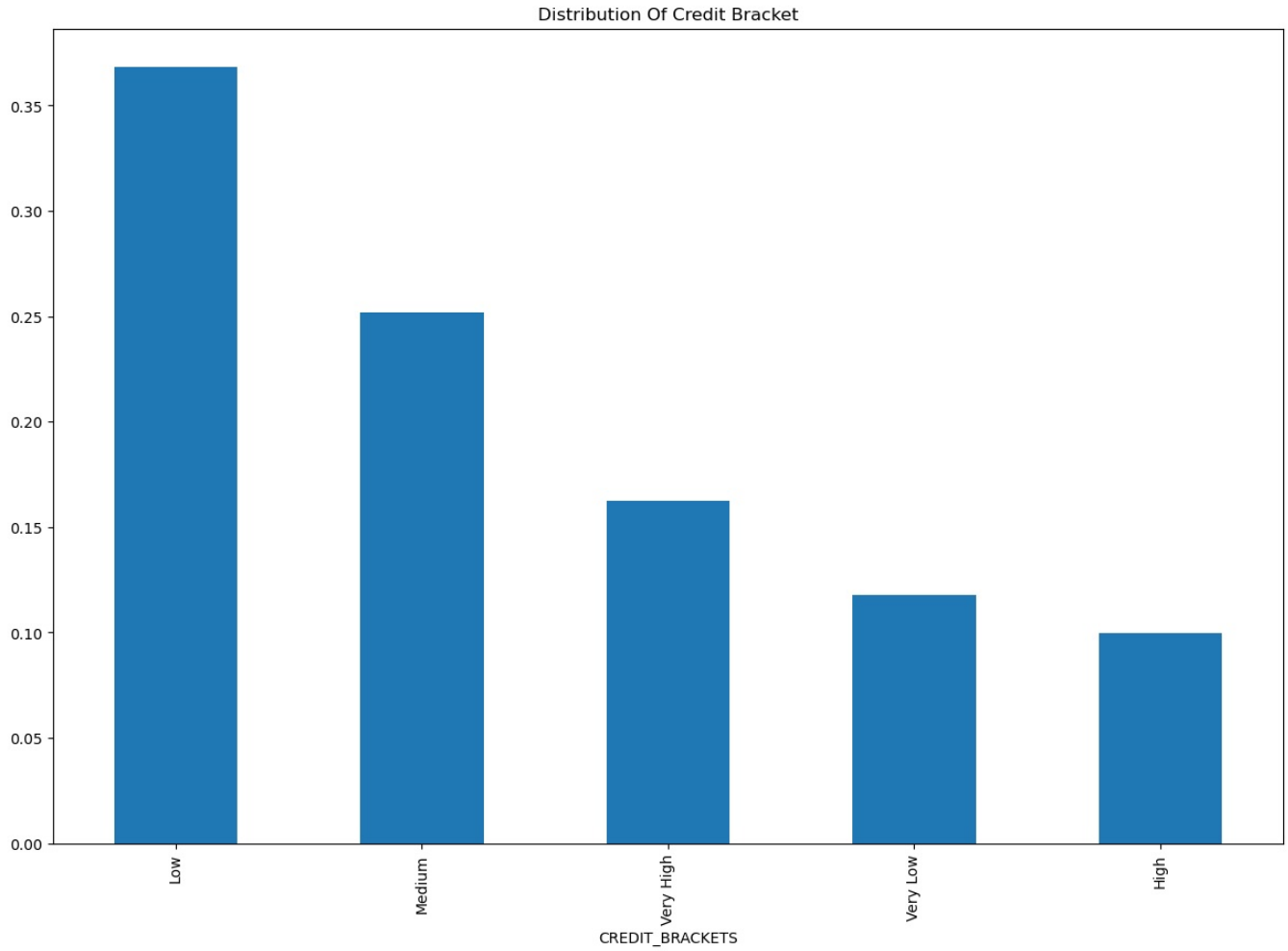
In [96]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Income Type")
df.groupby("NAME_INCOME_TYPE")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



Distribution Of Credit Bracket

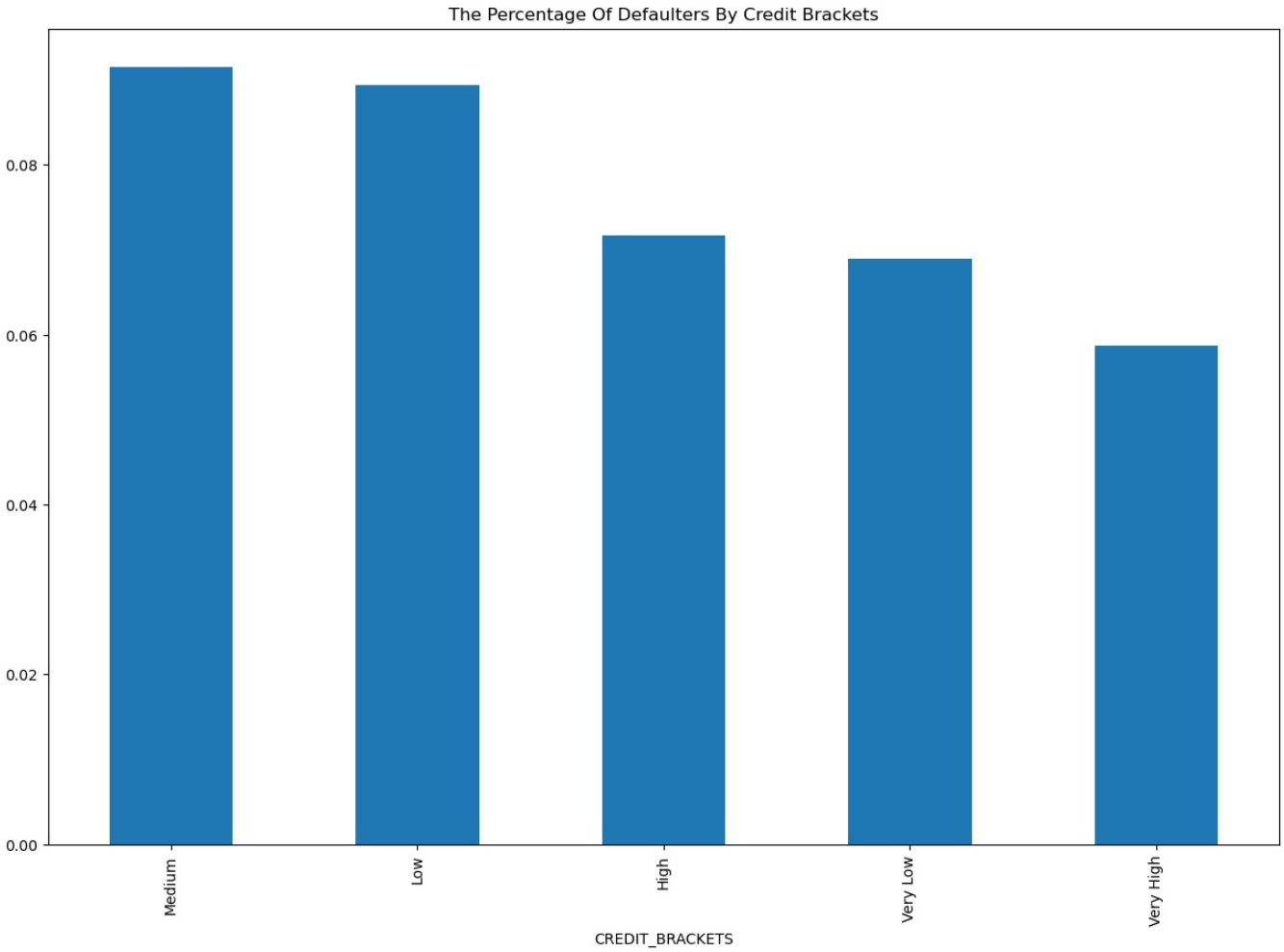
In [98]:

```
plt.figure(figsize=(15,10))
plt.title("Distribution Of Credit Bracket")
df["CREDIT_BRACKETS"].value_counts(normalize=True).plot.bar()
plt.xticks(rotation=90)
plt.show()
```



Percentage Of Defaulters By Credit Brackets

In [99]:
```
plt.figure(figsize=(15,10))
plt.title("The Percentage Of Defaulters By Credit Brackets")
df.groupby("CREDIT_BRACKETS")["TARGET"].mean().sort_values(ascending=False).plot.bar()
plt.show()
```



Load Previous Application Data

In [6]:
```
df_prevapp = pd.read_csv('previous_application.csv')
df_prevapp.head()
```

Out[6]:

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

In [101…
```
df_prevapp.shape
```

Out[101]: (1670214, 37)

In [102…
```
df_prevapp.dtypes
```

Out[102]:
```
SK_ID_PREV                    int64
SK_ID_CURR                    int64
NAME_CONTRACT_TYPE           object
AMT_ANNUITY                 float64
AMT_APPLICATION             float64
AMT_CREDIT                  float64
AMT_DOWN_PAYMENT            float64
AMT_GOODS_PRICE             float64
WEEKDAY_APPR_PROCESS_START   object
HOUR_APPR_PROCESS_START       int64
FLAG_LAST_APPL_PER_CONTRACT  object
NFLAG_LAST_APPL_IN_DAY        int64
RATE_DOWN_PAYMENT           float64
RATE_INTEREST_PRIMARY       float64
RATE_INTEREST_PRIVILEGED    float64
NAME_CASH_LOAN_PURPOSE       object
NAME_CONTRACT_STATUS         object
DAYS_DECISION                 int64
NAME_PAYMENT_TYPE            object
CODE_REJECT_REASON           object
NAME_TYPE_SUITE              object
NAME_CLIENT_TYPE             object
NAME_GOODS_CATEGORY          object
NAME_PORTFOLIO               object
NAME_PRODUCT_TYPE            object
CHANNEL_TYPE                 object
SELLERPLACE_AREA              int64
NAME_SELLER_INDUSTRY         object
CNT_PAYMENT                 float64
NAME_YIELD_GROUP             object
PRODUCT_COMBINATION          object
DAYS_FIRST_DRAWING          float64
DAYS_FIRST_DUE              float64
DAYS_LAST_DUE_1ST_VERSION   float64
DAYS_LAST_DUE               float64
DAYS_TERMINATION            float64
NFLAG_INSURED_ON_APPROVAL   float64
dtype: object
```

In [103…
```
df_prevapp.isna().sum()
```

Out[103]:
```
SK_ID_PREV                       0
SK_ID_CURR                       0
NAME_CONTRACT_TYPE               0
AMT_ANNUITY                 372235
AMT_APPLICATION                  0
AMT_CREDIT                       1
AMT_DOWN_PAYMENT            895844
AMT_GOODS_PRICE             385515
WEEKDAY_APPR_PROCESS_START       0
HOUR_APPR_PROCESS_START          0
FLAG_LAST_APPL_PER_CONTRACT      0
NFLAG_LAST_APPL_IN_DAY           0
RATE_DOWN_PAYMENT           895844
RATE_INTEREST_PRIMARY      1664263
RATE_INTEREST_PRIVILEGED   1664263
NAME_CASH_LOAN_PURPOSE           0
NAME_CONTRACT_STATUS             0
DAYS_DECISION                    0
NAME_PAYMENT_TYPE                0
CODE_REJECT_REASON               0
NAME_TYPE_SUITE             820405
NAME_CLIENT_TYPE                 0
NAME_GOODS_CATEGORY              0
NAME_PORTFOLIO                   0
NAME_PRODUCT_TYPE                0
CHANNEL_TYPE                     0
SELLERPLACE_AREA                 0
NAME_SELLER_INDUSTRY             0
CNT_PAYMENT                 372230
NAME_YIELD_GROUP                 0
PRODUCT_COMBINATION            346
DAYS_FIRST_DRAWING          673065
DAYS_FIRST_DUE              673065
DAYS_LAST_DUE_1ST_VERSION   673065
DAYS_LAST_DUE               673065
DAYS_TERMINATION            673065
NFLAG_INSURED_ON_APPROVAL   673065
dtype: int64
```

In [7]:
```
# Merge previous application data with current data.
df_merge = pd.merge(left=df,right=df_prevapp, how='left', left_on='SK_ID_CURR', right_on='SK_ID_CURR')
df.head(20)
```

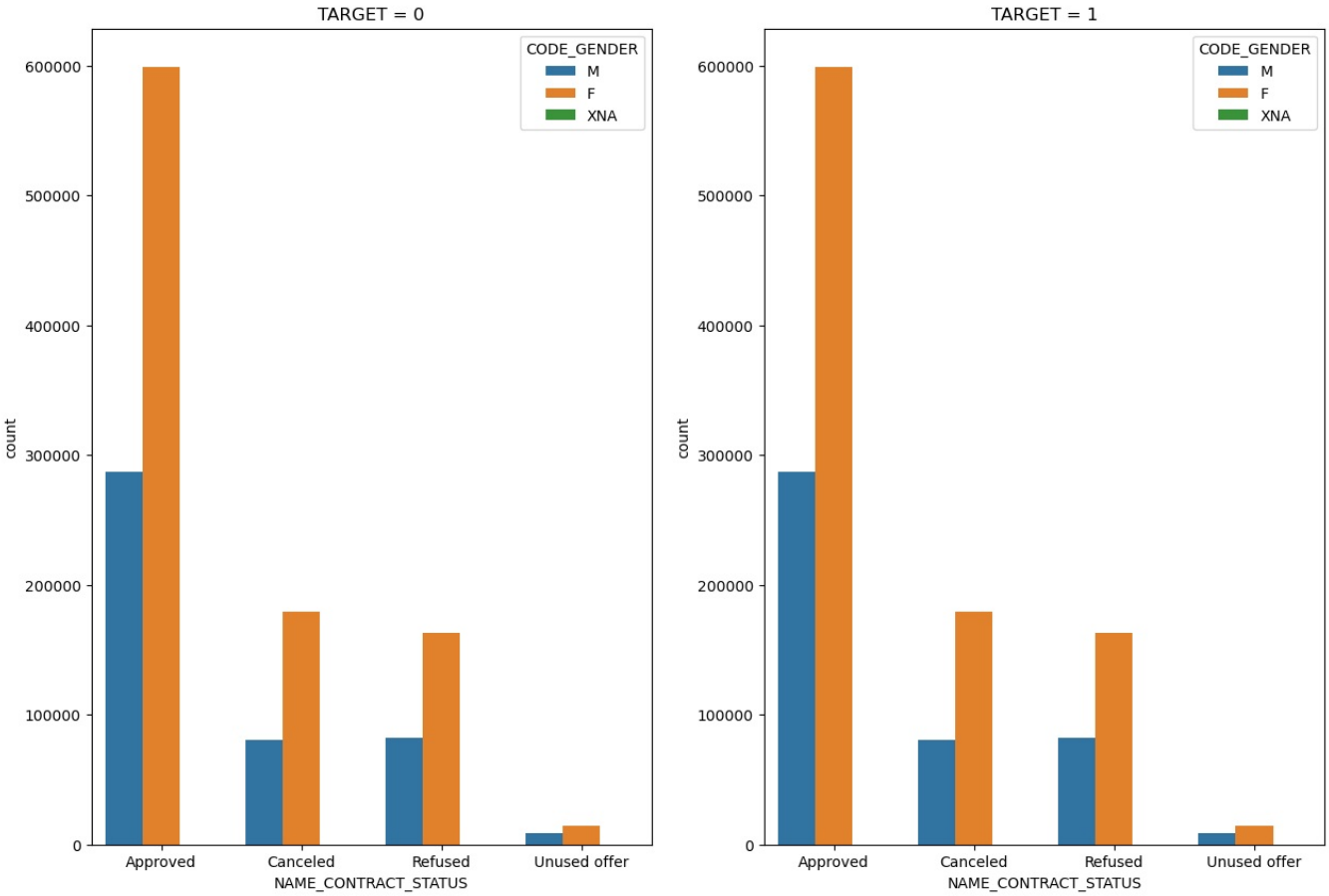
Out[7]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME
0	100002	1	Cash loans	M	N	Y	0	20
1	100003	0	Cash loans	F	N	N	0	27
2	100004	0	Revolving loans	M	Y	Y	0	6
3	100006	0	Cash loans	F	N	Y	0	13
4	100007	0	Cash loans	M	N	Y	0	12
5	100008	0	Cash loans	M	N	Y	0	9
6	100009	0	Cash loans	F	Y	Y	1	17
7	100010	0	Cash loans	M	Y	Y	0	36
8	100011	0	Cash loans	F	N	Y	0	11
9	100012	0	Revolving loans	M	N	Y	0	13
10	100014	0	Cash loans	F	N	Y	1	11
11	100015	0	Cash loans	F	N	Y	0	3
12	100016	0	Cash loans	F	N	Y	0	6
13	100017	0	Cash loans	M	Y	N	1	22
14	100018	0	Cash loans	F	N	Y	0	18
15	100019	0	Cash loans	M	Y	Y	0	15
16	100020	0	Cash loans	M	N	N	0	10
17	100021	0	Revolving loans	F	N	Y	1	8
18	100022	0	Revolving loans	F	N	Y	0	11
19	100023	0	Cash loans	F	N	Y	1	9

In [14]:

```
#Gender-wise breakdown of the previous loan application status across target values

plt.figure(figsize=(15,10))
plt.subplot(1, 2, 1)
plt.title('TARGET = 0')
sns.countplot(x='NAME_CONTRACT_STATUS',hue='CODE_GENDER',data=df_merge)
plt.subplot(1, 2, 2)
plt.title('TARGET = 1')
sns.countplot(x='NAME_CONTRACT_STATUS',hue='CODE_GENDER',data=df_merge)
plt.show()
```



CONCLUSION

The following Groups are more likely to default.

1. Low income group

2. Age group <30

3. Low Skilled Laborers occupation type.

4. Transport Type 3 organization Type.

5. Lower Secondary Education type.

In [ ]: