**Report**

Understanding the Dataset:

data=pd.read\_csv('application\_data.csv')

rows , columns =data.shape

print('Rows:',rows)

print('Columns:',columns)

#### Output:

**Rows: 307511**

**Columns: 122**

Let’s retrieve the data info:

data.info()

output:

<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

* Numerical data -> **quantitative data**
* Categorical data is referred to as “**qualitative data**” in data science since it describes the quality of the entity it represents.

**Null Values:**

import pandas as pd

data=pd.read\_csv('application\_data.csv')

rows , columns =data.shape

pd.set\_option('display.max\_rows', 200)

print(data.isnull().sum())

**SK\_ID\_CURR 0**

**REG\_REGION\_NOT\_WORK\_REGION 0**

**LIVE\_REGION\_NOT\_WORK\_REGION 0**

**REG\_CITY\_NOT\_LIVE\_CITY 0**

**REG\_CITY\_NOT\_WORK\_CITY 0**

**LIVE\_CITY\_NOT\_WORK\_CITY 0**

**ORGANIZATION\_TYPE 0**

**EXT\_SOURCE\_1 173378**

**EXT\_SOURCE\_2 660**

**EXT\_SOURCE\_3 60965**

**APARTMENTS\_AVG 156061**

**BASEMENTAREA\_AVG 179943**

**YEARS\_BEGINEXPLUATATION\_AVG 150007**

**YEARS\_BUILD\_AVG 204488**

**COMMONAREA\_AVG 214865**

**ELEVATORS\_AVG 163891**

**ENTRANCES\_AVG 154828**

**FLOORSMAX\_AVG 153020**

**FLOORSMIN\_AVG 208642**

**LANDAREA\_AVG 182590**

**LIVINGAPARTMENTS\_AVG 210199**

**LIVINGAREA\_AVG 154350**

**NONLIVINGAPARTMENTS\_AVG 213514**

**NONLIVINGAREA\_AVG 169682**

**APARTMENTS\_MODE 156061**

**BASEMENTAREA\_MODE 179943**

**YEARS\_BEGINEXPLUATATION\_MODE 150007**

**YEARS\_BUILD\_MODE 204488**

**COMMONAREA\_MODE 214865**

**ELEVATORS\_MODE 163891**

**ENTRANCES\_MODE 154828**

**FLOORSMAX\_MODE 153020**

**FLOORSMIN\_MODE 208642**

**LANDAREA\_MODE 182590**

**LIVINGAPARTMENTS\_MODE 210199**

**LIVINGAREA\_MODE 154350**

**TARGET 0**

**NAME\_CONTRACT\_TYPE 0**

**CODE\_GENDER 0**

**FLAG\_OWN\_CAR 0**

**FLAG\_OWN\_REALTY 0**

**CNT\_CHILDREN 0**

**AMT\_INCOME\_TOTAL 0**

**AMT\_CREDIT 0**

**AMT\_ANNUITY 12**

**AMT\_GOODS\_PRICE 278**

**NAME\_TYPE\_SUITE 1292**

**NAME\_INCOME\_TYPE 0**

**NAME\_EDUCATION\_TYPE 0**

**NAME\_FAMILY\_STATUS 0**

**NAME\_HOUSING\_TYPE 0**

**REGION\_POPULATION\_RELATIVE 0**

**DAYS\_BIRTH 0**

**DAYS\_EMPLOYED 0**

**DAYS\_REGISTRATION 0**

**DAYS\_ID\_PUBLISH 0**

**OWN\_CAR\_AGE 202929**

**FLAG\_MOBIL 0**

**FLAG\_EMP\_PHONE 0**

**FLAG\_WORK\_PHONE 0**

**FLAG\_CONT\_MOBILE 0**

**FLAG\_PHONE 0**

**FLAG\_EMAIL 0**

**OCCUPATION\_TYPE 96391**

**CNT\_FAM\_MEMBERS 2**

**REGION\_RATING\_CLIENT 0**

**REGION\_RATING\_CLIENT\_W\_CITY 0**

**WEEKDAY\_APPR\_PROCESS\_START 0**

**HOUR\_APPR\_PROCESS\_START 0**

**REG\_REGION\_NOT\_LIVE\_REGION 0**

**NONLIVINGAPARTMENTS\_MODE 213514**

**NONLIVINGAREA\_MODE 169682**

**APARTMENTS\_MEDI 156061**

**BASEMENTAREA\_MEDI 179943**

**YEARS\_BEGINEXPLUATATION\_MEDI 150007**

**YEARS\_BUILD\_MEDI 204488**

**COMMONAREA\_MEDI 214865**

**ELEVATORS\_MEDI 163891**

**ENTRANCES\_MEDI 154828**

**FLOORSMAX\_MEDI 153020**

**FLOORSMIN\_MEDI 208642**

**LANDAREA\_MEDI 182590**

**LIVINGAPARTMENTS\_MEDI 210199**

**LIVINGAREA\_MEDI 154350**

**NONLIVINGAPARTMENTS\_MEDI 213514**

**NONLIVINGAREA\_MEDI 169682**

**FONDKAPREMONT\_MODE 210295**

**HOUSETYPE\_MODE 154297**

**TOTALAREA\_MODE 148431**

**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**

**Treating Null Values:**

emptycol=data.isnull().sum()

emptycol=emptycol[emptycol.values>(0.3\*len(emptycol))]

print(len(emptycol))

checking weather the columns having data loss more than 30%

I got 61

* Lets remove these 61 columns
* emptycol=emptycol[emptycol.values>(0.3\*len(emptycol))]
* emptycol = list(emptycol[emptycol.values>=0.3].index)
* data.drop(labels=emptycol,axis=1,inplace=True)
* print(len(emptycol))
* print(data.shape)

* Lets remove rows which has data loss more the 30%

emptyrow=emptyrow[emptyrow.values>(0.3\*len(emptyrow))]

emptyrow=list(emptyrow[emptyrow.values>=0.3].index)

data.drop(labels=emptyrow,axis=0,inplace=True) emptyrow=data.isnull().sum(axis=1)

print(len(emptyrow))

* Lets remove the unwanted column from the sheet
* no\_need = ['FLAG\_MOBIL', 'FLAG\_EMP\_PHONE', 'FLAG\_WORK\_PHONE', 'FLAG\_CONT\_MOBILE', 'FLAG\_PHONE', 'FLAG\_EMAIL', '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']
* data.drop(labels=no\_need, axis=1, inplace=True)
* Lets remove the xna values which mean not available

Checking where we have xna values in ‘CODE\_GENDER’ and 'ORGANIZATION\_TYPE'.

print(data[data['CODE\_GENDER']=='XNA'].shape)

print(data[data['ORGANIZATION\_TYPE']=='XNA'].shape)

(4, 34) (55374, 34)

CODE\_GENDER has only 4 xna values so we will retrieve them by placing mode value.

data.loc[data['CODE\_GENDER'] == 'XNA', 'CODE\_GENDER'] = 'F'

Lets drop the rows which have XNA in ORGANIZATION\_TYPE

data = data.drop(data.loc[data['ORGANIZATION\_TYPE'] == 'XNA'].index)

data[data['ORGANIZATION\_TYPE'] == 'XNA'].index

* Creating bins and slot for AMT\_INCLOME\_RANGE
* It will use for plotting pie charts, histograms

bins = [0, 25000, 50000, 75000, 100000, 125000, 150000, 175000, 200000, 225000, 250000,275000, 300000, 325000, 350000, 375000, 400000, 425000, 450000, 475000, 500000, 10000000000]

slot = ['0-25000', '25000-50000', '50000-75000', '75000,100000', '100000-125000', '125000-150000', '150000-175000', '175000-200000',

'200000-225000', '225000-250000', '250000-275000', '275000-300000', '300000-325000', '325000-350000', '350000-375000','375000-400000', '400000-425000', '425000-450000', '450000-475000', '475000-500000', '500000 and above']

data['AMT\_INCOME\_RANGE'] = pd.cut(data['AMT\_INCOME\_TOTAL'], bins, labels=slot)

* Creating bins and slot for AMT\_CREDIT\_RANGE
* It will use for plotting pie charts, histograms

bins = [0, 150000, 200000, 250000, 300000, 350000, 400000, 450000, 500000,

550000, 600000, 650000, 700000, 750000, 800000, 850000, 900000, 1000000000]

slots = ['0-150000', '150000-200000', '200000-250000', '250000-300000', '300000-350000', '350000-400000', '400000-450000','450000-500000', '500000-550000', '550000-600000', '600000-650000', '650000-700000', '700000-750000', '750000-800000','800000-850000', '850000-900000', '900000 and above']

data['AMT\_CREDIT\_RANGE'] = pd.cut(data['AMT\_CREDIT'], bins=bins, labels=slots)

* Dividing the data set into two parts they are…
  + Target1\_df 🡪 customers who having difficulties in payback
  + Target0\_df 🡪 other than that
* We will find out the calculate imbalance percentage
* print(round(len(target0\_df)/len(target1\_df),2))

10.55

Creating a function to create to plot a graph.

def uniplot(df,col,title,hue =None):

    sns.set\_style('whitegrid')

    sns.set\_context('talk')

    plt.rcParams["axes.labelsize"] = 20

    plt.rcParams['axes.titlesize'] = 22

    plt.rcParams['axes.titlepad'] = 30

    temp = pd.Series(data = hue)

    fig, ax = plt.subplots()

    width = len(df[col].unique()) + 7 + 4\*len(temp.unique())

    fig.set\_size\_inches(width , 6)

    plt.xticks(rotation=45)

    plt.yscale('log')

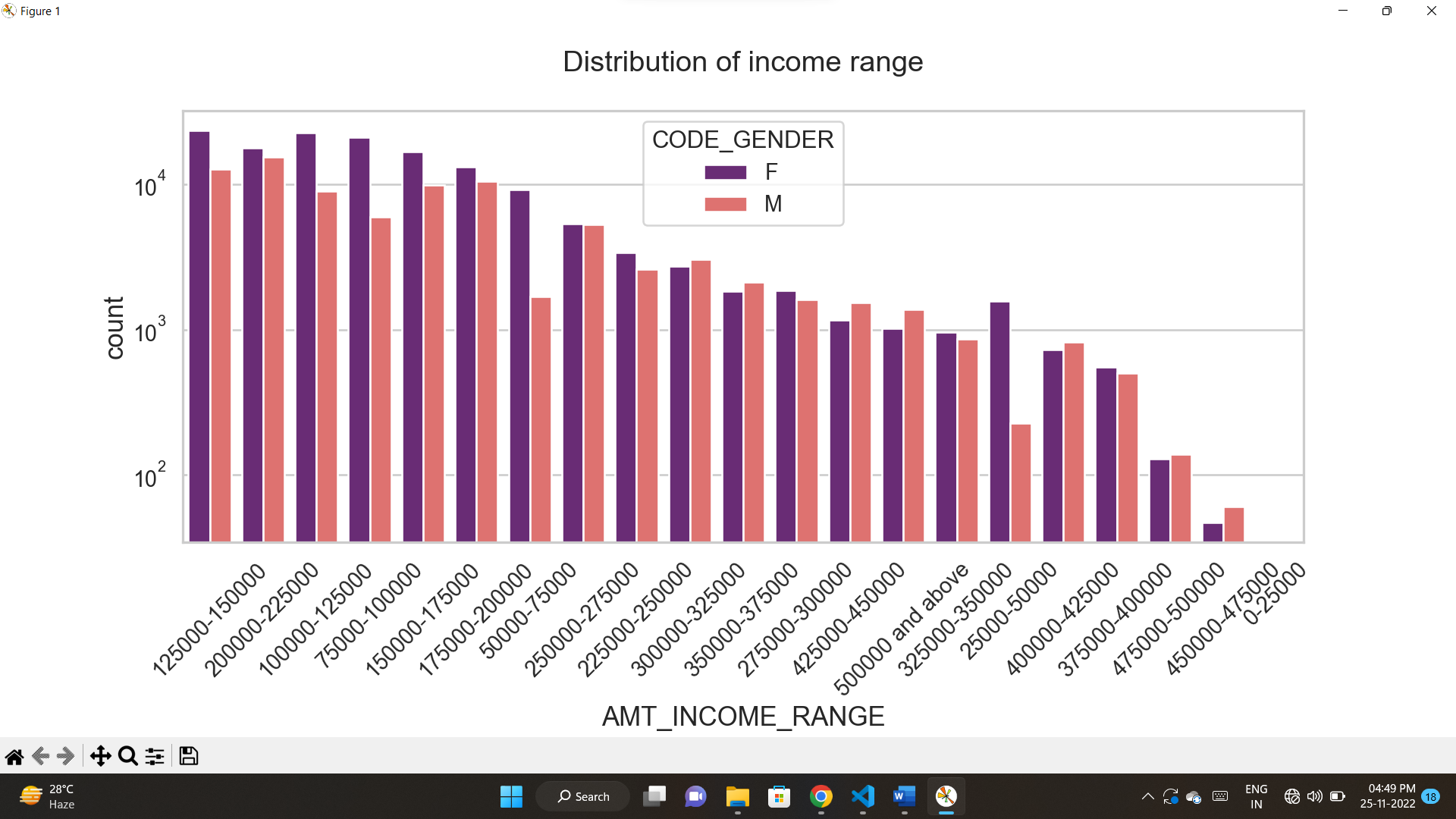
    plt.title(title)

    ax = sns.countplot(data = df, x= col,order=df[col].value\_counts().index,hue = hue,palette='magma')

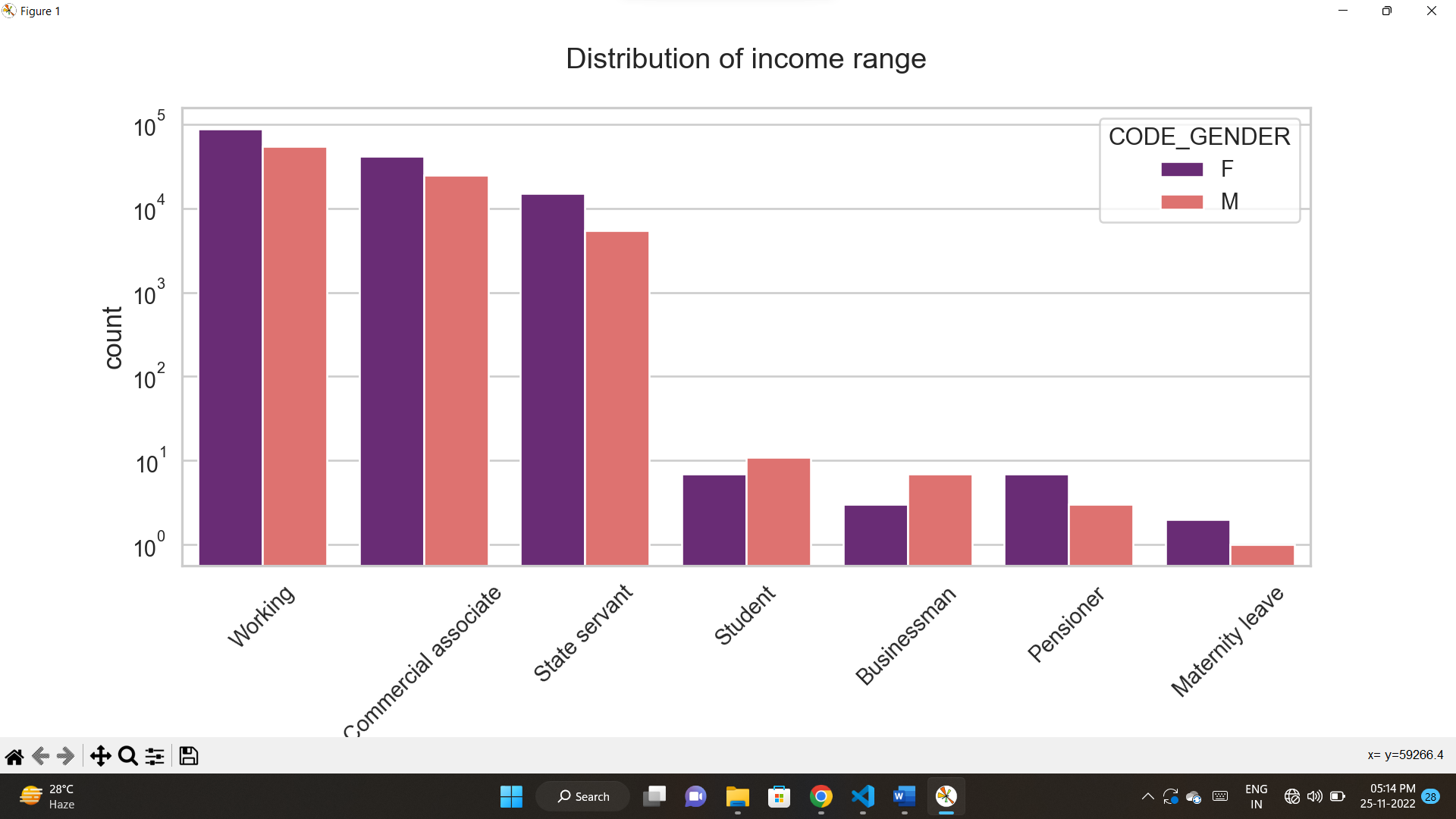
    plt.show()

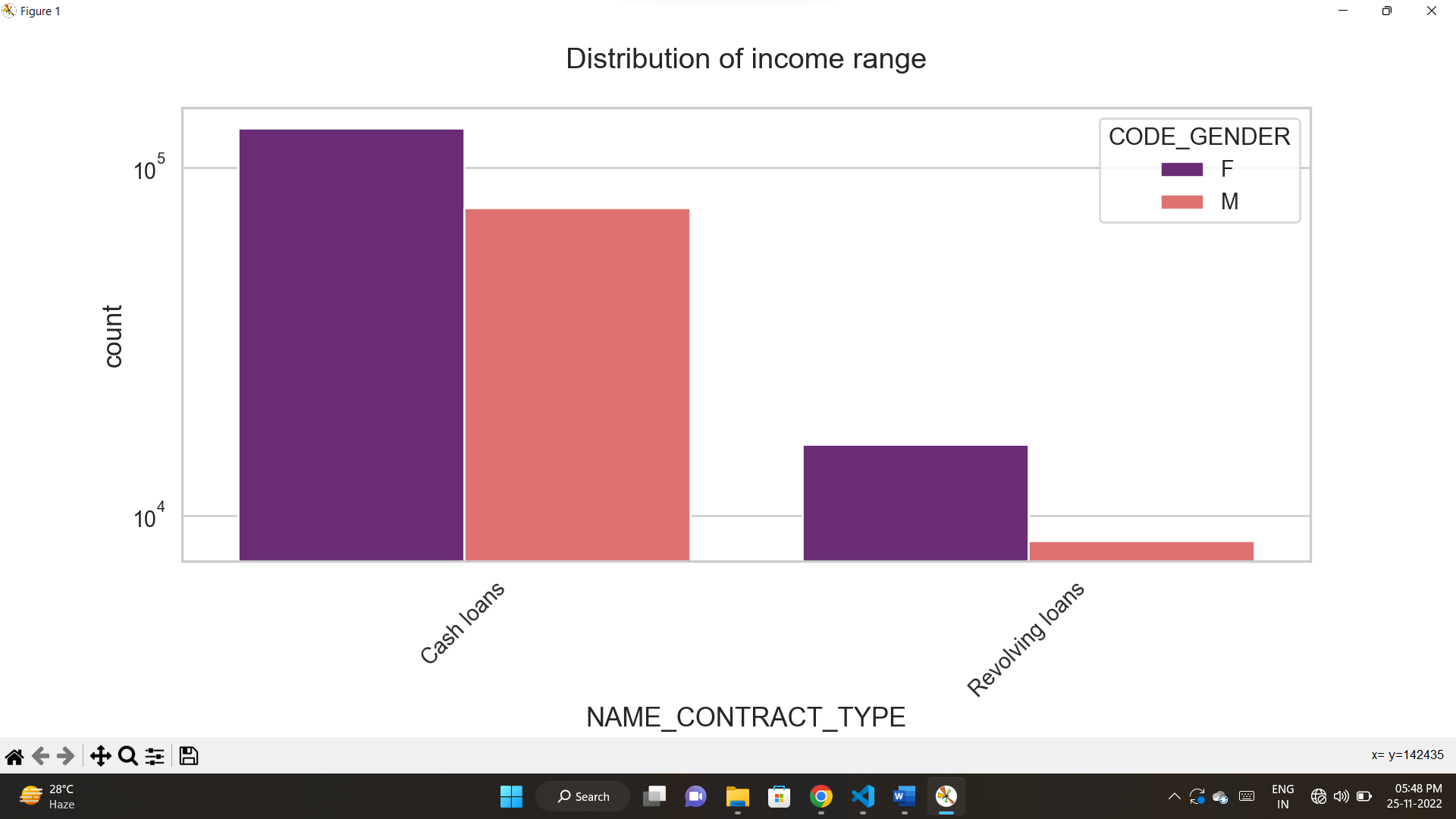
* **analysing the data for customers who having no payment difficulties**

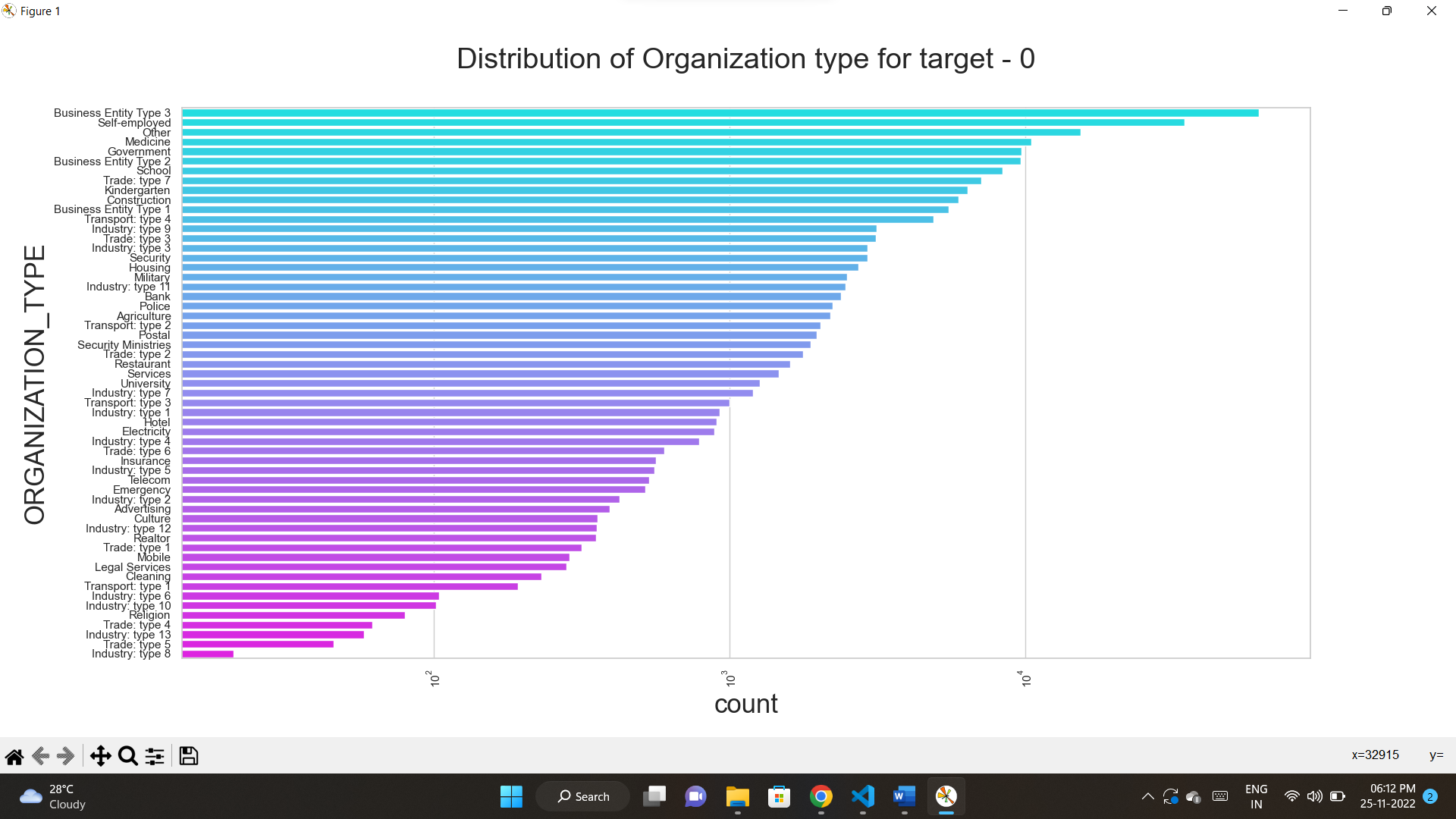
uniplot(target0\_df,col='AMT\_INCOME\_RANGE',title='Distribution of income range',hue='CODE\_GENDER')



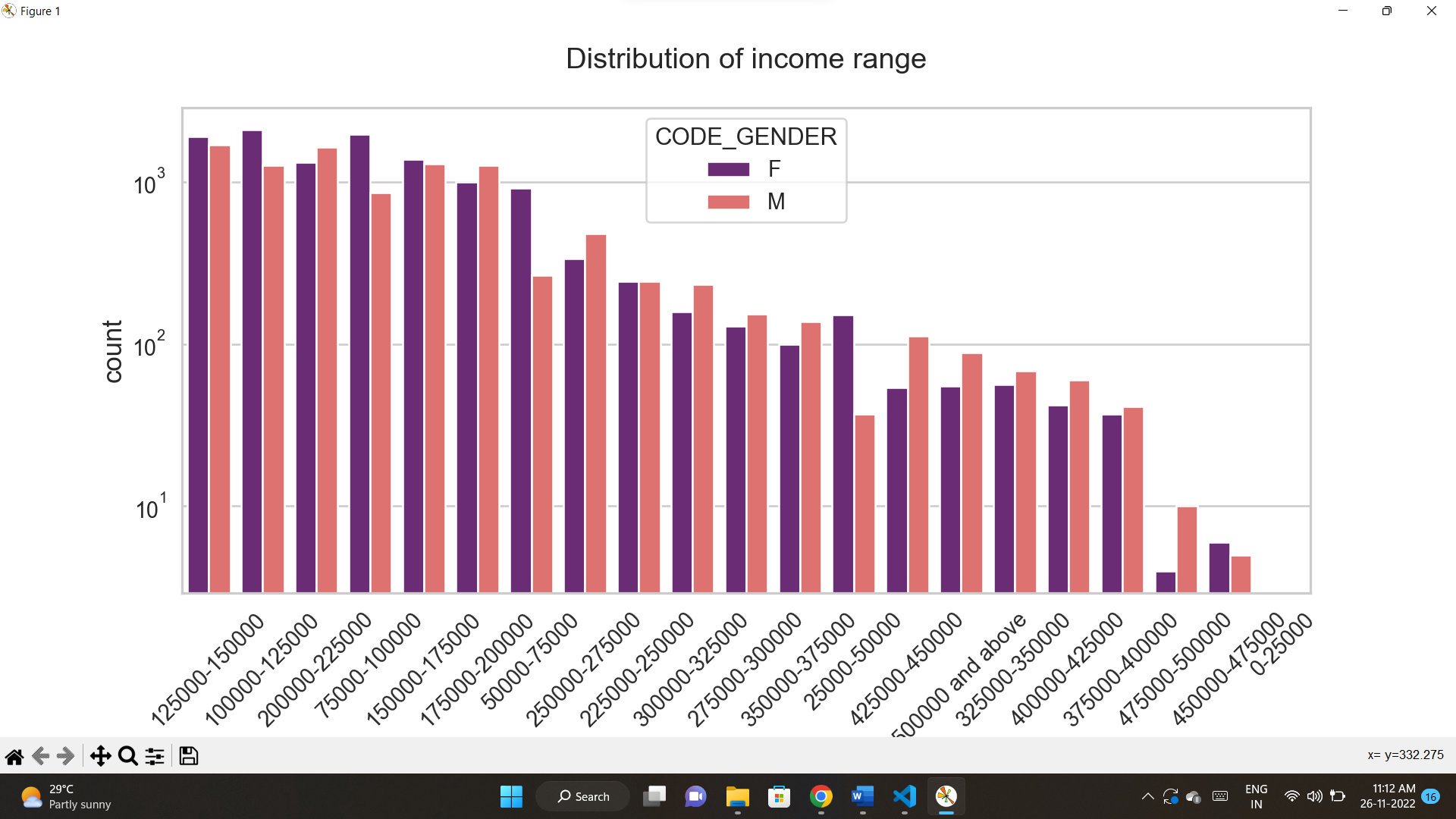
* from this graph we can observe that
  + there are a greater number of credits in 100000-200000
  + in that range there are number of females
  + very less count of credits presents at 450000-475000
  + No one is there at 0-25000
* Plotting a graph for income distribution

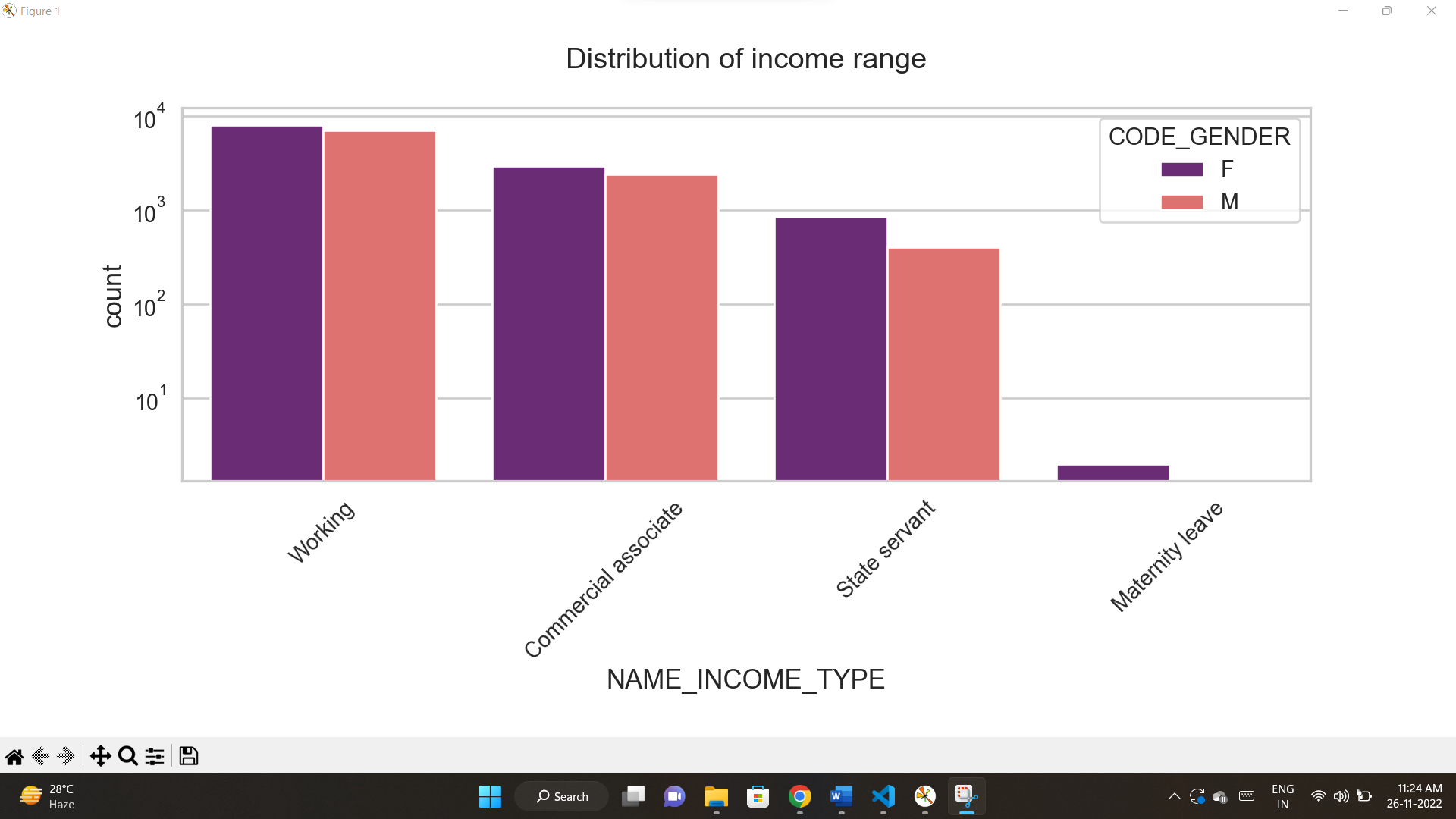


* From this graph we can observe that
  + Working, Commercial associate and state servant are having high range of income.
  + For this range, females are majority.
  + Student, Businessman, Pensioner and Maternity leave has less range of income
* Plotting a graph for contract distribution
* From this graph we can analyse that
  + Cash loans are in higher count
  + Females are more in this range
* Plotting a count graph on organization type



* From this graph we can analyse that
  + Business entity type 3, self-employed, other, medicine and Government has more number of clients.
  + Religion, trade: type 4,industry:type 4, industry:type 13, trade:type 5, industry: type 8 has less number clients
* **analysing the data for customers who having payment difficulties**
* Plotting a graph for income distribution
* uniplot(target1\_df,col='AMT\_INCOME\_RANGE',title='Distribution of income range',hue='CODE\_GENDER')

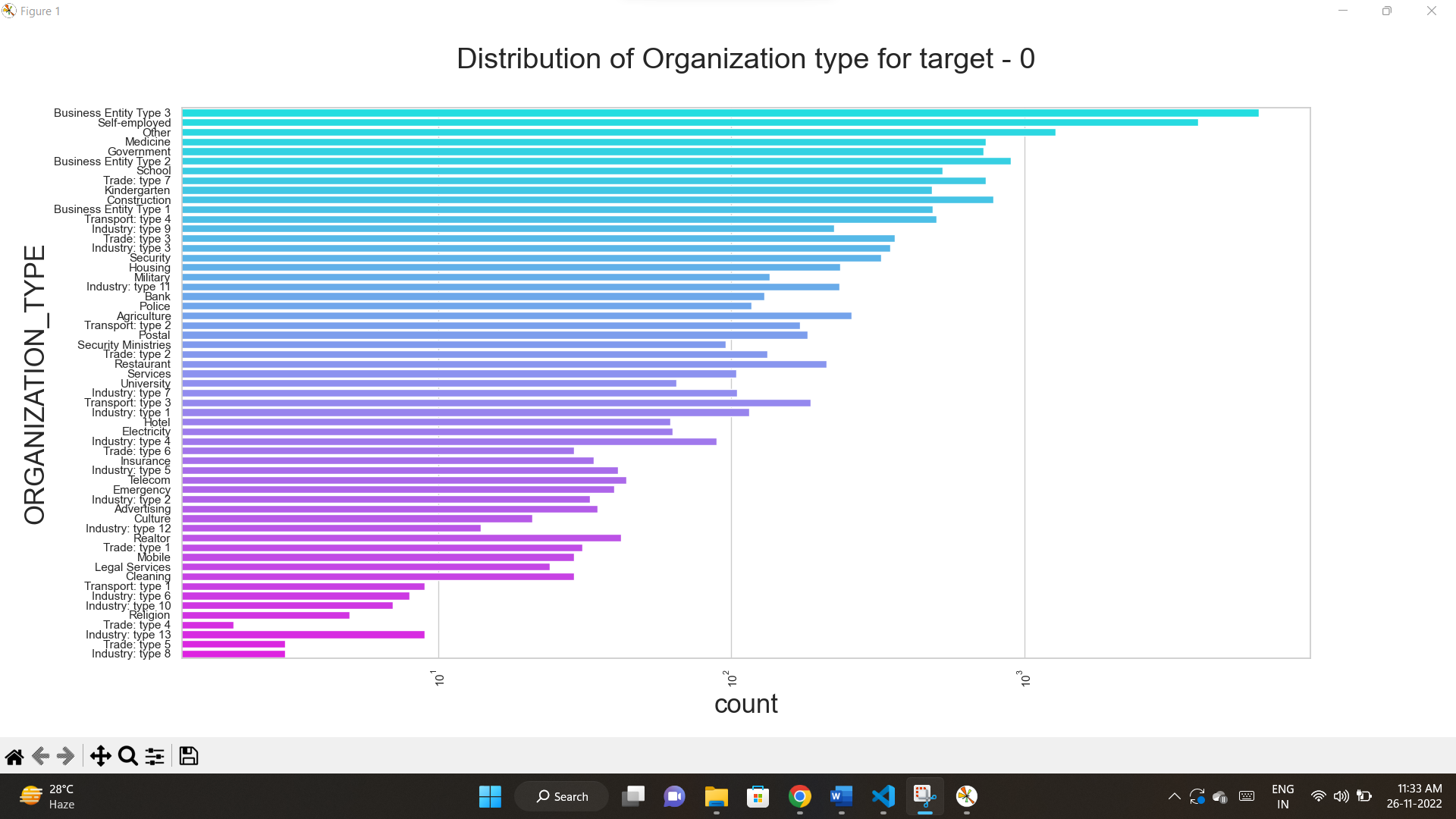


* from this graph we can analyse that
  + there are a greater number of credits from 100000 – 200000 range
  + there are more males than female
  + there are less number of credits greater than 400000
  + males
* plotting a graph on income distribution
* From this graph we can analyse that
  + Working, commercial associate has more range of income
  + In that range male are greater than female
  + State servant and maternity leave has less range of income
  + In that range also males are greater than female
* Plotting a graph on distribution of income range on contract type.

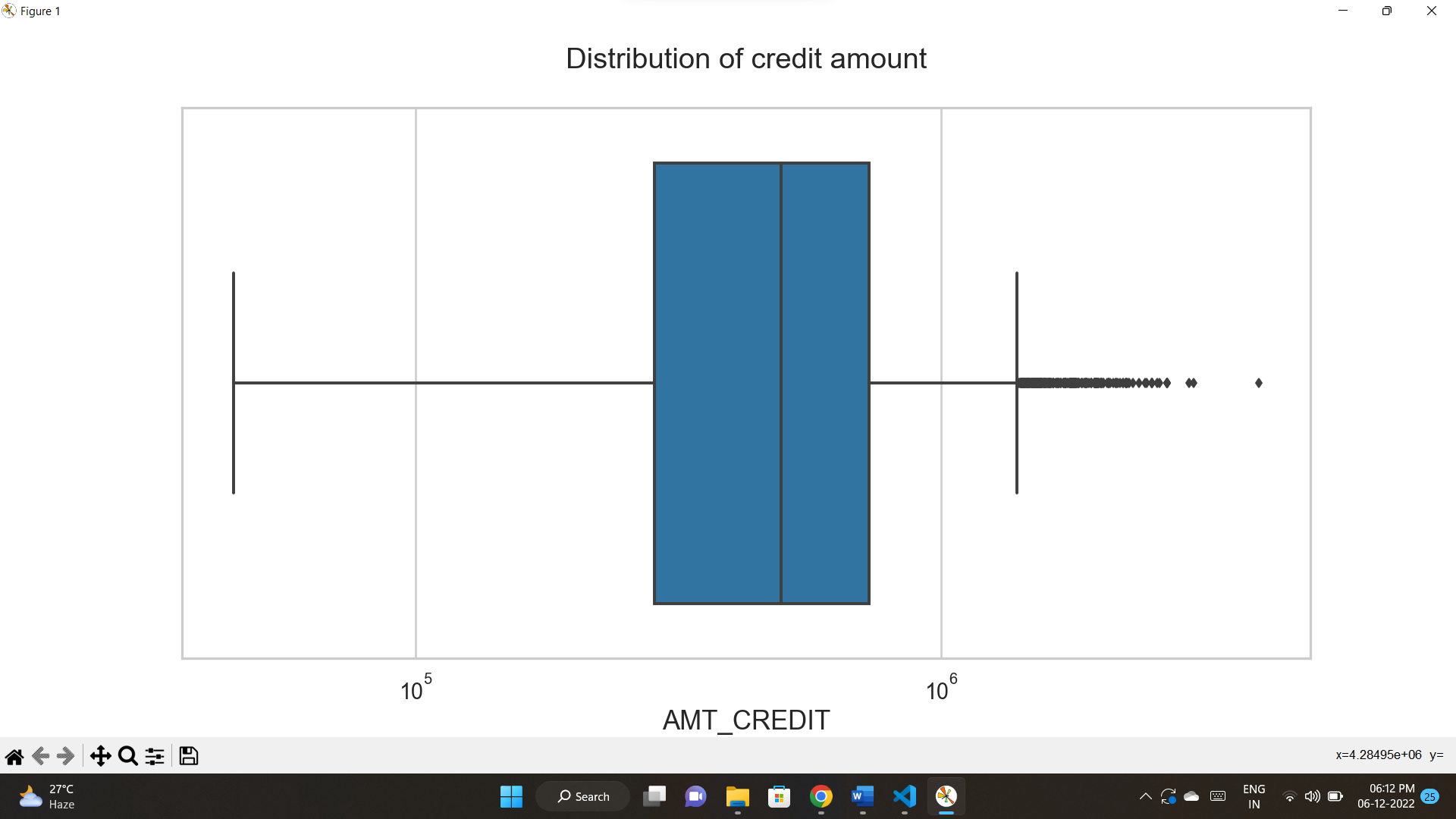
Chart, bar chart

Description automatically generated

* From this graph we can analyse that
  + Cash loans are more than revolving loans
  + In this too males are greater
* Plotting a count graph on organization.



* From this graph we can analyse that
  + Business entity type 3, self-employed, other, medicine and Government has more number of clients.
  + Religion, trade: type 4,industry:type 4, industry:type 13, trade:type 5, industry: type 8 has less number clients
* Box plots for same kind of data
  + Same kind of data are ‘AMT\_INCOME\_TOTAL’,AMT\_CREDIT’, ‘AMT\_ANNUITY’
* Let’s plot the graphs for target0\_df



* From this graph we can conclude that
  + Q1 is bigger than Q3 many of customers lies in Q1
  + There are outliers in credit distribution

Chart, bar chart

Description automatically generated

* There are some outliers in income distribution
* q1 in bigger than the q3 so the more customers are in q1

Chart, bar chart

Description automatically generated

* Let’s plot the graphs for target1\_df

Chart, histogram

Description automatically generated

From this graph

* Q1 is larger than the q3
* Some outliers also present in the graph.

Chart, histogram

Description automatically generated

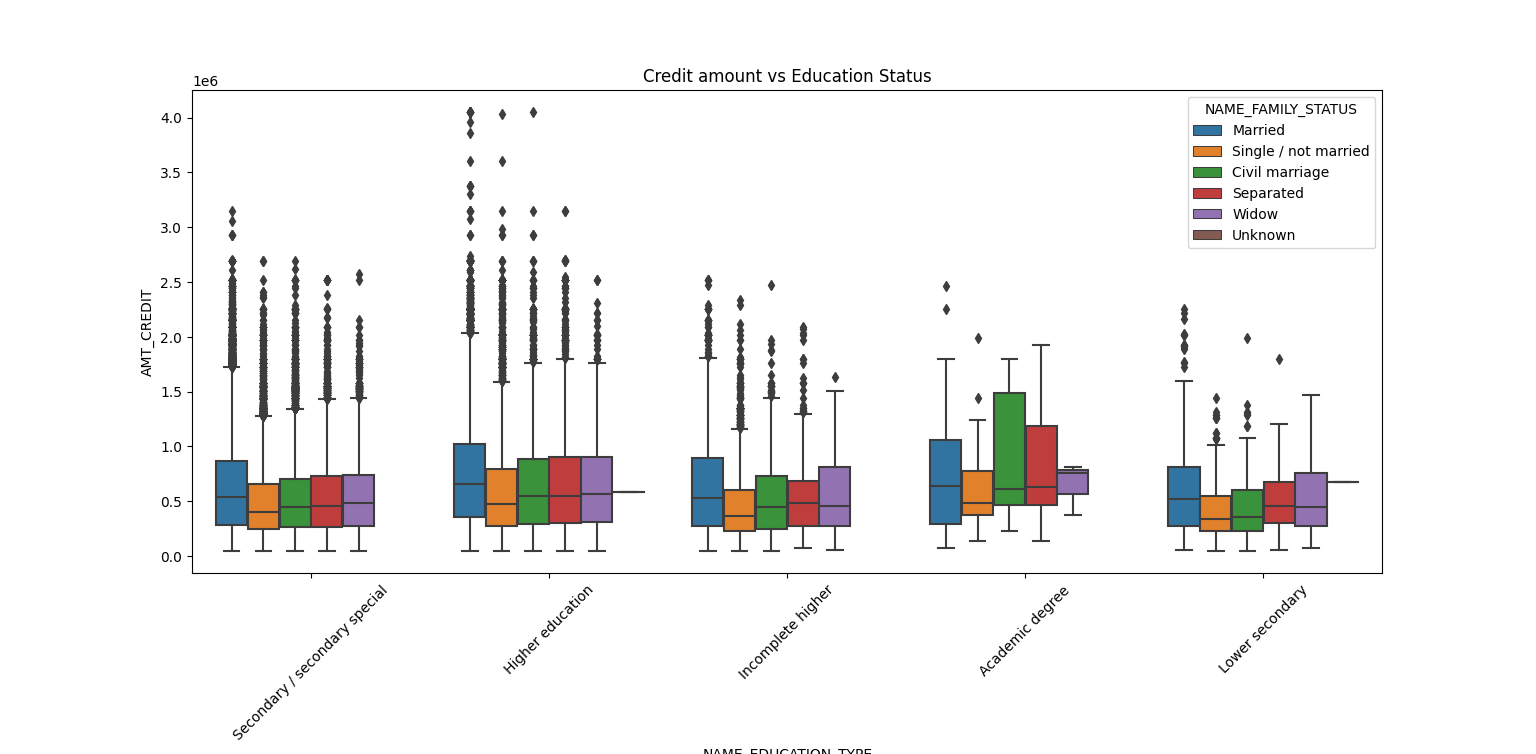
* There are some outliers in income distribution
* q1 in bigger than the q3 so the more customers are in q1

Chart, bar chart

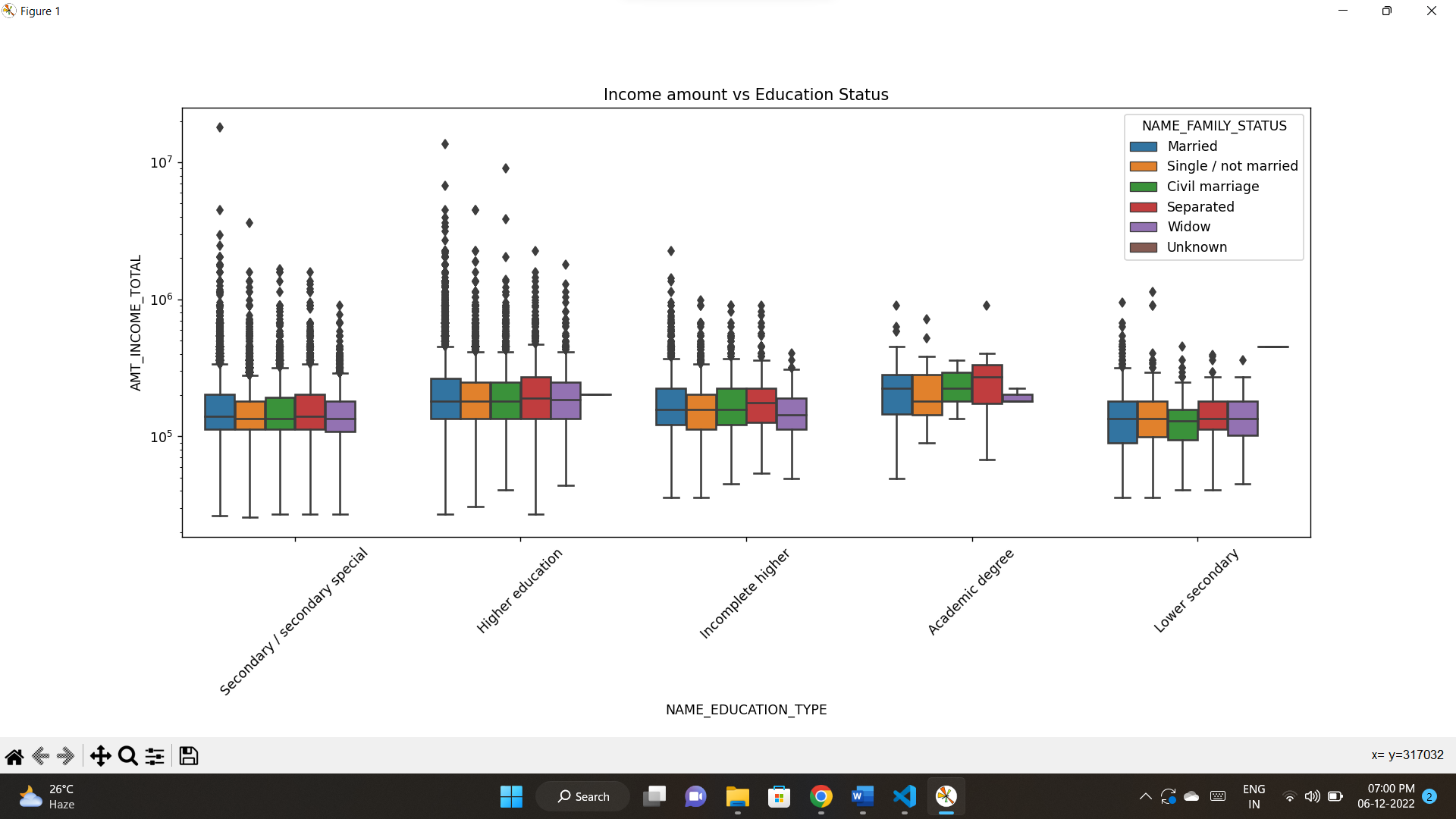
Description automatically generated

* there are some outliers in the AMT\_ANNUITY
* First quartile is bigger than the third one. so many of outliers present in the first outlier.

**BI-VARIATE ANALYSIS**

Plotting graphs for bi-variate analysis for target0\_df

* Academic degree of married, Civil marriage separated has a greater number of credits than any other else.
* Higher education of family status married, single, civil marriage has a greater number of outliers.
* 3rd quartile of civil marriage in academic degree are having greater number of credits than any others



* In Higher education, almost family types have same income amount.
* And it also contains outliers.
* Academic degree has a smaller number of outliers.
* Lower secondary has less income amount than others.
* Plotting graphs for bi-variate analysis for target1\_df

Chart, box and whisker chart

Description automatically generated

* In Academic degree married has a greater number of amount credit and this doesn’t contain any outlier.
* Lower secondary group has less number of credits compare to others.

Chart, box and whisker chart

Description automatically generated

From this graph we can say that,

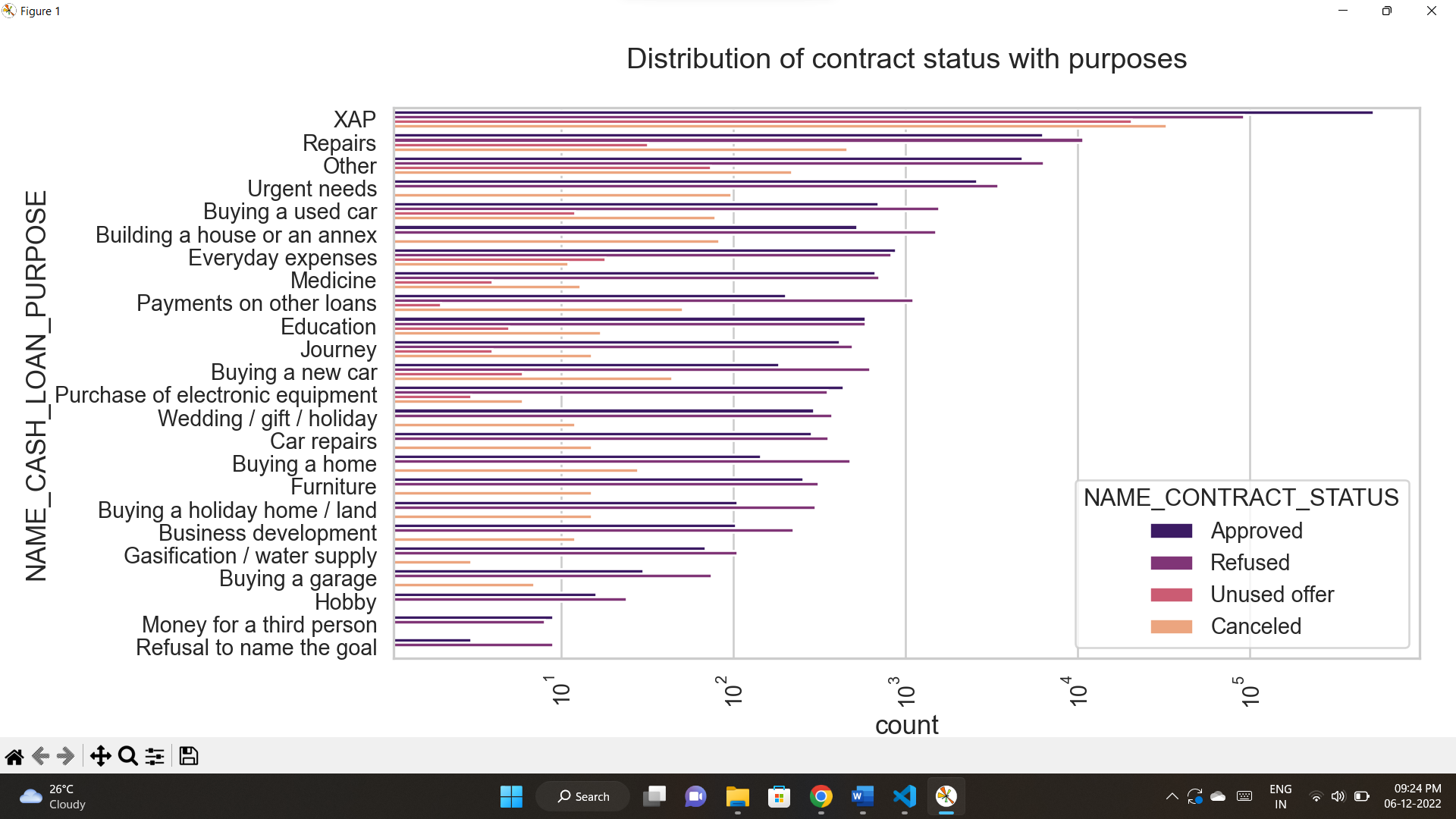
* Academic degree has only one status that is married. This has more income and no outliers.
* Almost all status of Higher education has almost same income.
* Lower secondary has less income than others.

Let’s analysis the previous application details.

* Importing dataset

data1 = pd.read\_csv("previous\_application.csv")

* Knowing the null values sum and dropping the columns where it has 30% data loss
* emptycol1 = data1.isnull().sum()
* emptycol1 = emptycol1[emptycol1.values>(0.3\*len(emptycol1))]
* emptycol1 = list(emptycol1[emptycol1.values>=0.3].index)
* data1.drop(labels=emptycol1,axis=1,inplace=True)
* data1 = data1.drop(data1[data1['NAME\_CASH\_LOAN\_PURPOSE']== 'XNA'].index)
* data1 = data1.drop(data1[data1['NAME\_CASH\_LOAN\_PURPOSE']== 'XNA'].index)
* Merge the previous application data with application data
* new\_data=pd.merge(left=data,right=data1,how='inner',on='SK\_ID\_CURR',suffixes='\_x')
* Renaming the data
* new\_data1 = new\_data.rename({'NAME\_CONTRACT\_TYPE\_' : 'NAME\_CONTRACT\_TYPE','AMT\_CREDIT\_':'AMT\_CREDIT','AMT\_ANNUITY\_':'AMT\_ANNUITY','WEEKDAY\_APPR\_PROCESS\_START\_' : 'WEEKDAY\_APPR\_PROCESS\_START','HOUR\_APPR\_PROCESS\_START\_':'HOUR\_APPR\_PROCESS\_START','NAME\_CONTRACT\_TYPEx':'NAME\_CONTRACT\_TYPE\_PREV','AMT\_CREDITx':'AMT\_CREDIT\_PREV','AMT\_ANNUITYx':'AMT\_ANNUITY\_PREV','WEEKDAY\_APPR\_PROCESS\_STARTx':'WEEKDAY\_APPR\_PROCESS\_START\_PREV','HOUR\_APPR\_PROCESS\_STARTx':'HOUR\_APPR\_PROCESS\_START\_PREV'}, axis=1)
* Dropping the unnecessary data
* new\_data1.drop(['SK\_ID\_CURR','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','WEEKDAY\_APPR\_PROCESS\_START\_PREV','HOUR\_APPR\_PROCESS\_START\_PREV','FLAG\_LAST\_APPL\_PER\_CONTRACT','NFLAG\_LAST\_APPL\_IN\_DAY'],axis=1,inplace=True)
* **Performing univariate analysis**

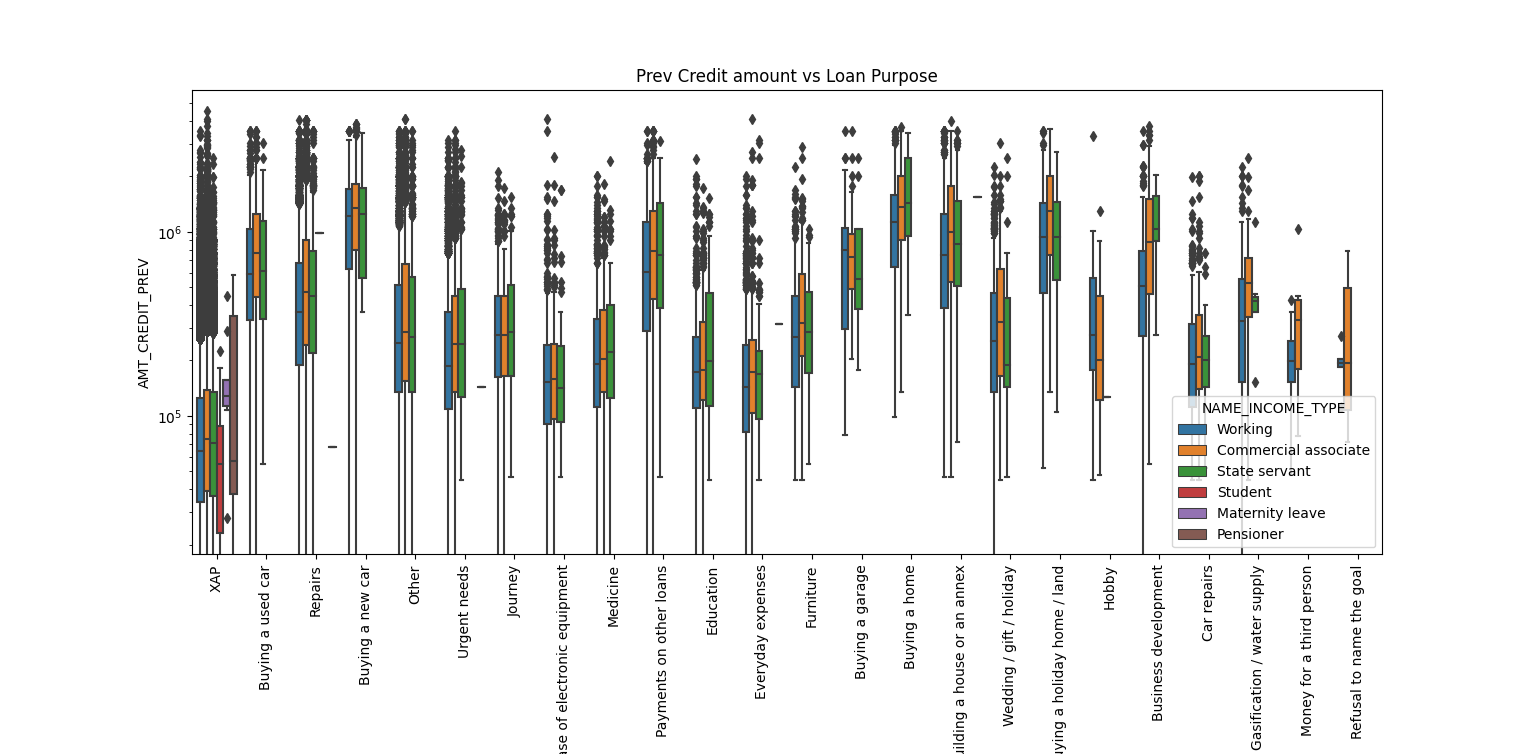


* Most of loans are cancelled in XAP
* Most of loans are approved in XAP
* Less number of approved loans are in Refusal to name the goal.

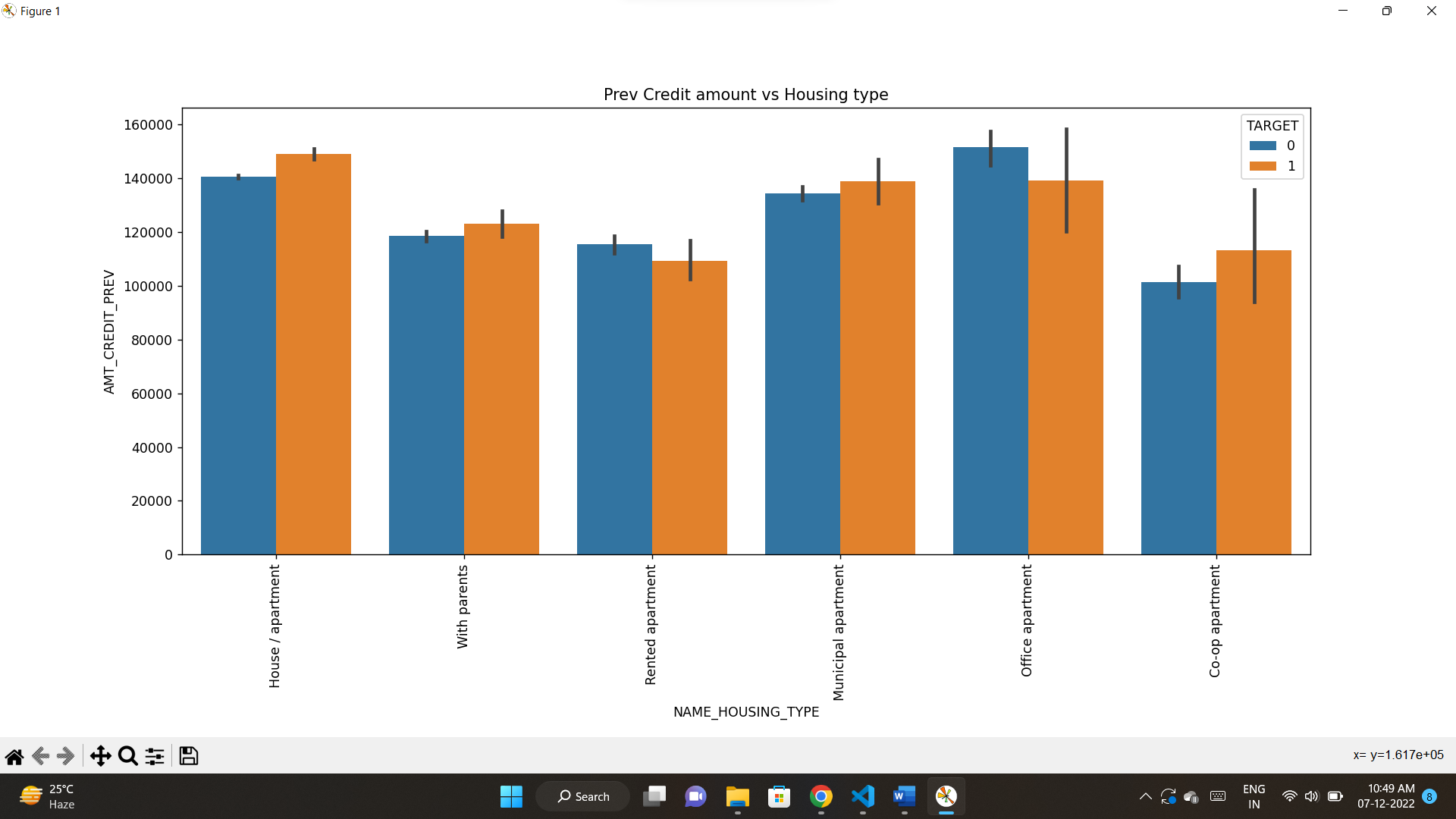
Chart

Description automatically generated

* XAP has difficulties in payment of their loans.
* Least number of loans are in Refusal to name the goal with target 1.



* Loan purpose of house buying has more amount of credit.
* Least amount credited to XAP



* Office apartment has more loan repayment difficulties. So bank need to avoid giving loans to them
* House/apartment has more have no payment difficulties. So bank should focus on these groups.