CREDIT EDA ASSIGNMENT

BY-
APOORVA NEDUNOORI
(APFE22m00993)

PROBLEM STATEMENT

- The loan providing companies find it hard to give loans to the people due to their insufficient or non-existent credit history.
- Because of that, some consumers use it to their advantage by becoming a defaulter. Suppose you work for a consumer finance company which specializes in lending various types of loans to urban customers.
- You have to use EDA to analyze the patterns present in the data. This will ensure that the applicants capable of repaying the loan are not rejected.

BUSINESS UNDERSTANDING

- When the company receives a loan application, the company has to decide for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:
- If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company
- If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company.

BUSINESS UNDERSTANDING

- The data given , contains the information about the loan application at the time of applying for the loan. It contains two types of scenarios:
- **The client with payment difficulties:** he/she had late payment more than X days on at least one of the first Y instalments of the loan in our sample,
- **All other cases:** All other cases when the payment is paid on time.

When a client applies for a loan, there are four types of decisions that could be taken by the client/company):

- **Approved:** The Company has approved loan Application
- **Cancelled:** The client cancelled the application sometime during approval. Either the client changed her/his mind about the loan or in some cases due to a higher risk of the client, he received worse pricing which he did not want.
- **Refused:** The company had rejected the loan (because the client does not meet their requirements etc.).
- **Unused offer:** Loan has been cancelled by the client but at different stages of the process.

In this case study, we use EDA to understand how consumer attributes and loan attributes influence the tendency to default.

Business Objectives

- This case study aims to identify patterns which indicate if a client has difficulty paying their instalments which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc.
- This will ensure that the consumers capable of repaying the loan are not rejected. Identification of such applicants using EDA is the aim of this case study.

- In other words, the company wants to understand the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default.

- The company can utilize this knowledge for its portfolio and risk assessment.

Dataset Given:

- The given dataset has 3 files as explained below:

- *1. 'application_data.csv'* contains all the information of the client at the time of application.
  The data is about whether a **client has payment difficulties.**

- *2. 'previous_application.csv'* contains information about the client's previous loan data. It contains the data on whether the previous application had been **Approved, Cancelled, Refused or Unused offer.**

- *3. 'columns_description.csv'* is data dictionary which describes the meaning of the variables.

Application_data Analysis

- ”TARGET” column gives us the information if the client is a defaulter or not.
- “TARGET”, 1 implies a defaulter, 0 is a non-defaulter
- The shape of the dataset is 307511,122
- There are many different types of datatypes like object, int, float present in the dataset.

Application_data Analysis

- The columns greater than 45% of missing values are dropped.
- The median is added to impute the missing values in the existing columns.
- Outliers are identified and treated.
- Converted negative values present in “date” columns into positive numbers.

Application_data Analysis

- Filtered Categorical, Quantitative values on ”obj”,”Int”,”Float” datatypes.
- Merged the “SK_ID_CURR” column using pivot table as it is present in both the datasets.
- The total count of missing values in the rows are also shown in the notebook.

Application_data Analysis

- Binning is done on “AMT_INCOME_TOTAL”, “AMT_CREDIT” using Histogram.
- Univariate analysis is done on the application_data dataset.
- Bivariate analysis is also done on the application_data dataset.
- Correlation on the TARGET variables are shown using HEATMAPS.

Application_data Analysis

jupyter Credit EDA Analysis Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [1]:
```
#Importing all the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#Ignoring Warnings
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
#importing the csv file
inp0=pd.read_csv("application_data.csv")
pd.set_option('display.max_columns', None)
pd. set_option('display.max_rows', 500)

inp0.head()
```

Out[3]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDI
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.

In [4]:
```
inp0.shape
```

Out[4]: (307511, 122)

Importing the necessary libraries and csv files

Merging the data on SK_ID_CURR column

We can see the coloumn SK_ID_CURR is same in both the data frames, indexing it using pivot table function

In [37]:
```
df1 = pd.pivot_table(inp1,index=["SK_ID_CURR","NAME_CONTRACT_TYPE"])
df1.head(3)
```

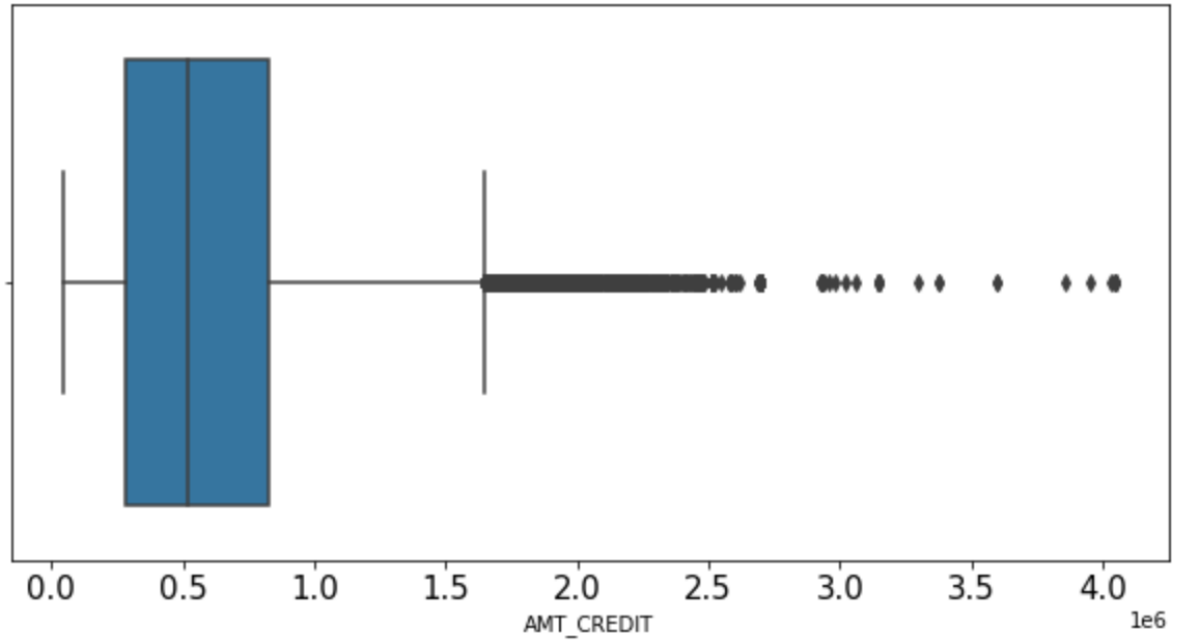
Out[37]:

		AMT_ANNUITY	AMT_CREDIT	AMT_GOODS_PRICE	AMT_INCOME_TOTAL	AMT_REQ_CREDIT_BUREAU_DAY	AMT_REQ_C
SK_ID_CURR	NAME_CONTRACT_TYPE						
100002	Cash loans	24700.5	406597.5	351000.0	202500.0	0.0	
100003	Cash loans	35698.5	1293502.5	1129500.0	270000.0	0.0	
100004	Revolving loans	6750.0	135000.0	135000.0	67500.0	0.0	

In [ ]:
```
```

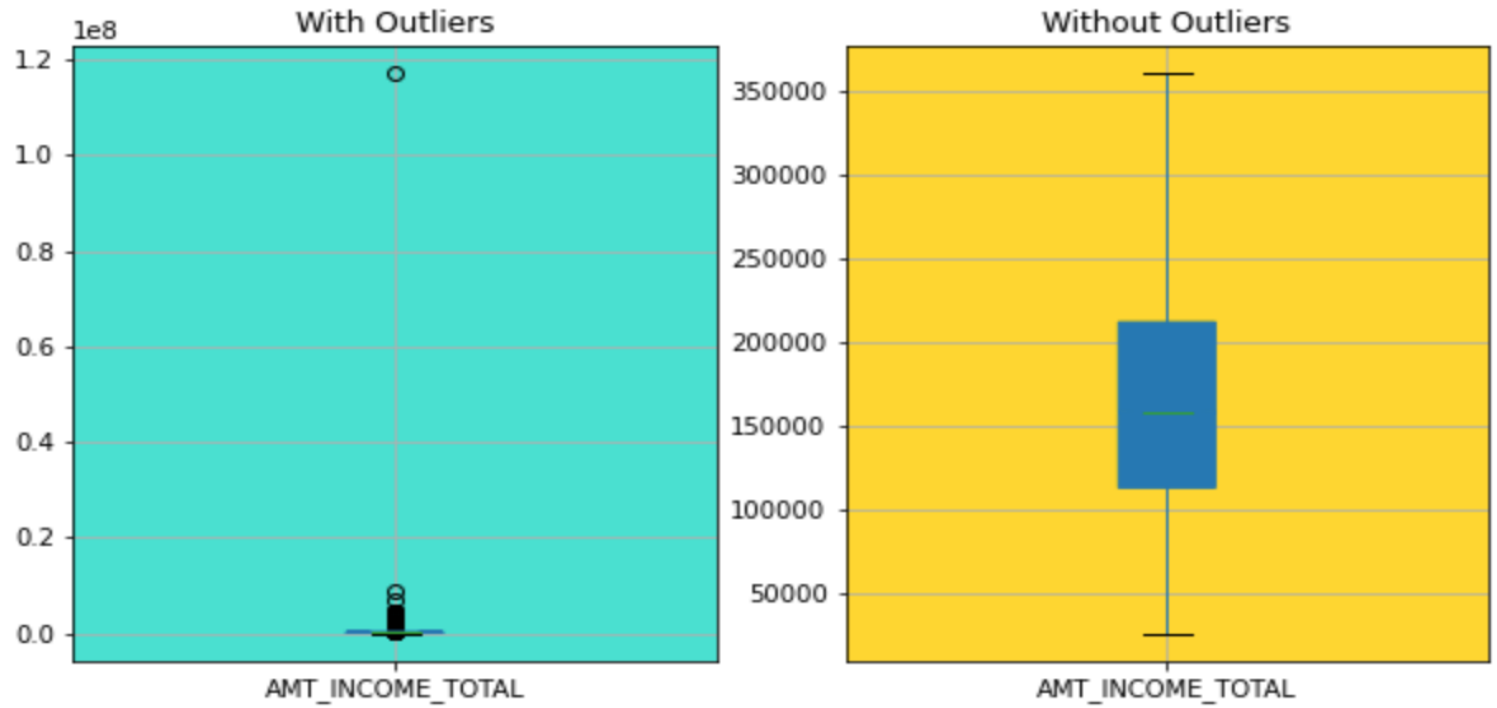
Checking for OUTLIERS in AMT_CREDIT column

In [39]:
```
plt.figure(figsize=(10,5))
sns.boxplot(inp1["AMT_CREDIT"])
plt.xticks(fontsize=15)
plt.show()
```



Treatment of Outliers in AMT_INCOME_TOTAL column

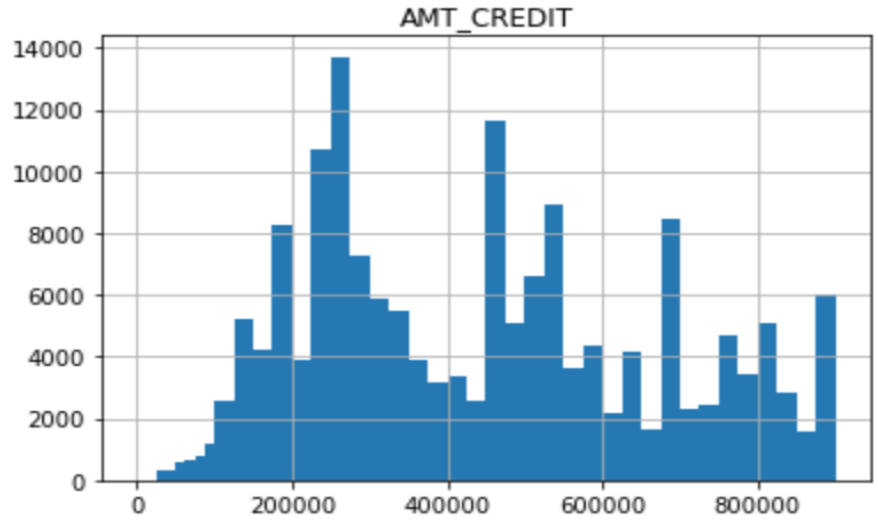
```
inp1['AMT_INCOME_TOTAL'].describe()
```




```
Before removing: [25650.0, 117000000.0]
After removing: [25650.0, 360000.0]
```

UNIVARIATE ANALYSIS

In [59]:
```
# SHOW THE DISTRIBUTION OF AMT_CREDIT
inp1_Amt_Credit = inp1[['AMT_CREDIT']]

# DEFINE THE BINS
bins = [0, 25000, 50000, 62500, 75000, 87500, 100000, 125000, 150000, 175000, 200

# PLOT A HISTOGRAM TO SEE THE DISTRIBUTION OF INCOME
inp1_Amt_Credit.hist(bins= bins, range=[2.565000e+04,1.170000e+08])

plt.show()

inp1_Amt_Credit.describe()
```

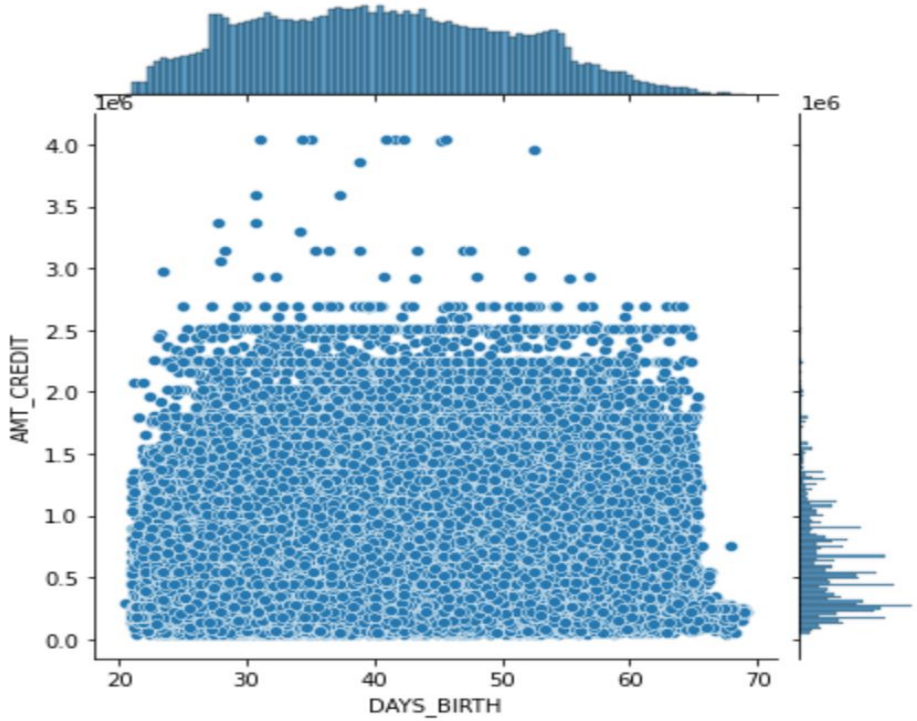


BIVARIATE ANALYSIS

JOINT PLOT between DAYS_BIRTH AND AMT_CREDIT

In [73]:
```
plt.figure(figsize=(25,25))
sns.jointplot('DAYS_BIRTH','AMT_CREDIT', inp_Zero);
plt.xticks(rotation='vertical')
plt.show()
```

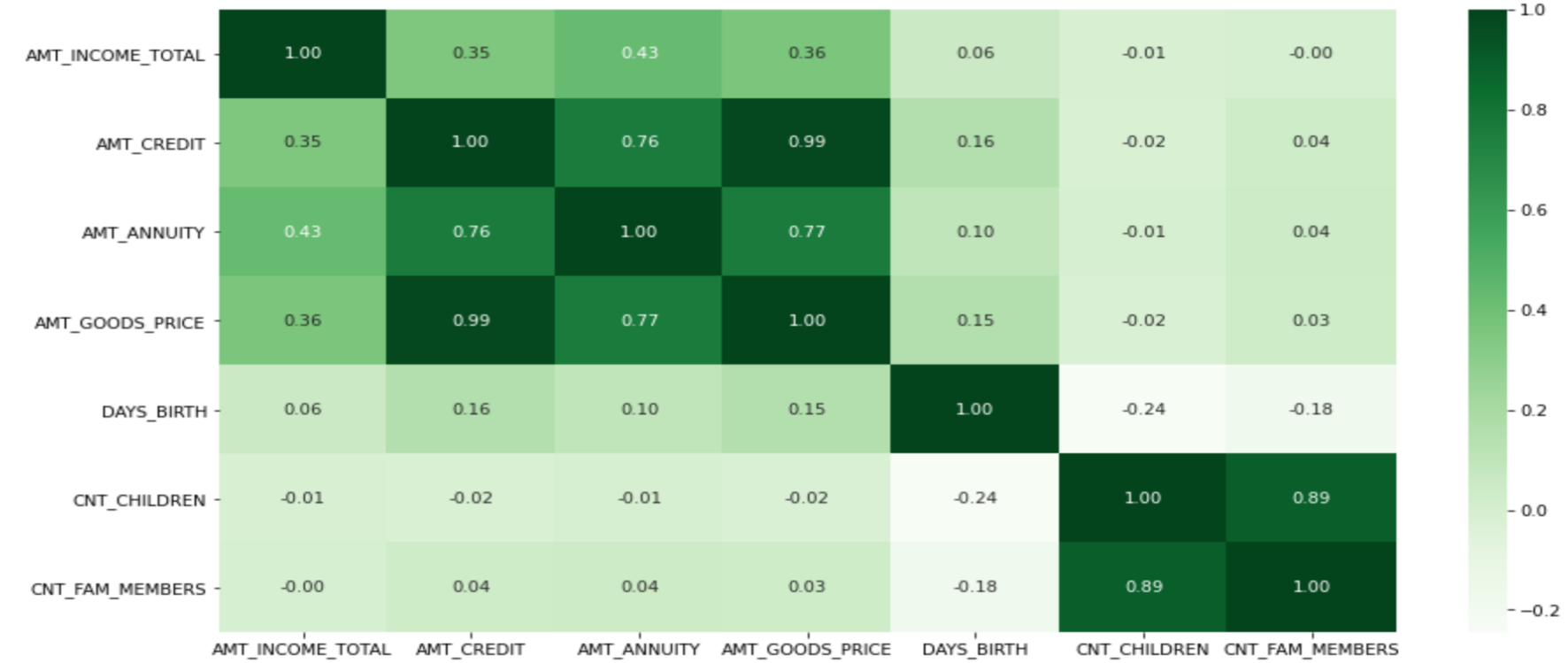
```
<Figure size 1800x1800 with 0 Axes>
```



CORRELATION – HEATMAP ON WHO PAY ON TIME

HEAT MAP - TO TEST CORRELATION BETWEEN CRITICAL QUANTIATIVE VALUES IN THE CURRENT APPLICATION SET FOR CLIENTS WHO PAY ON TIME

In [76]:
```
hm=inp_Zero[['AMT_INCOME_TOTAL','AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE','DAYS_BIRTH','CNT_CHILDREN','CNT_FAM_MEM
plt.figure(figsize=(15,8))
sns.heatmap(hm,annot = True, fmt = ".2f", cmap = "Greens")
plt.show()
```



DATA IMBALANCE

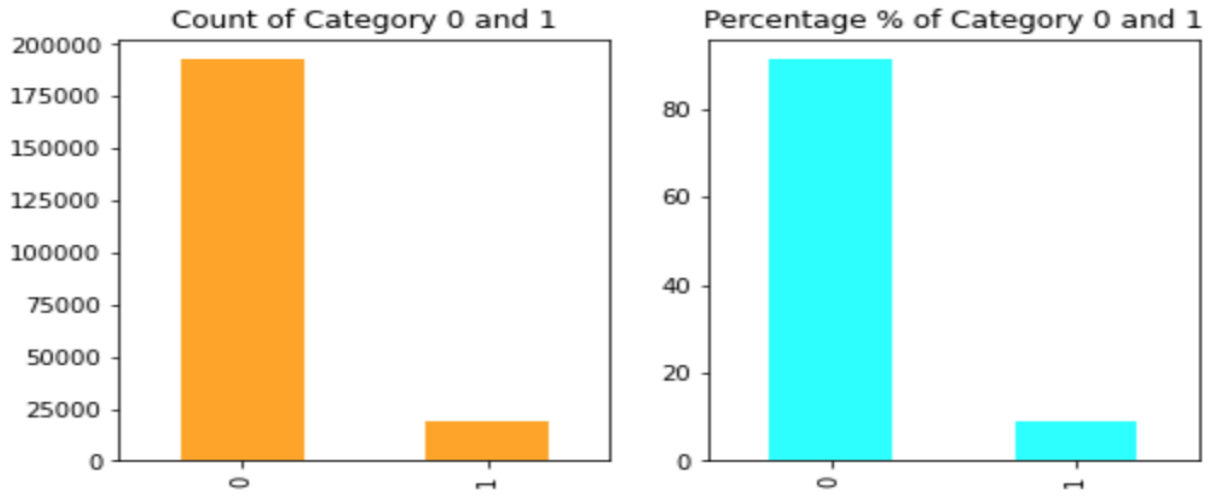
- In Application_data dataset, 91.2% are non-defaulters and 8.79% are defaulters of the loan amount as per my analysis.

- Segmentation is done on the basis of the TARGET variable, as non-defaulters and defaulters of loan.

Data Imbalance BAR GRAPH

PLOT THE IMBALANCE COUNT AND PERCENTAGE FOR TARGET VALUES 0 AND 1.

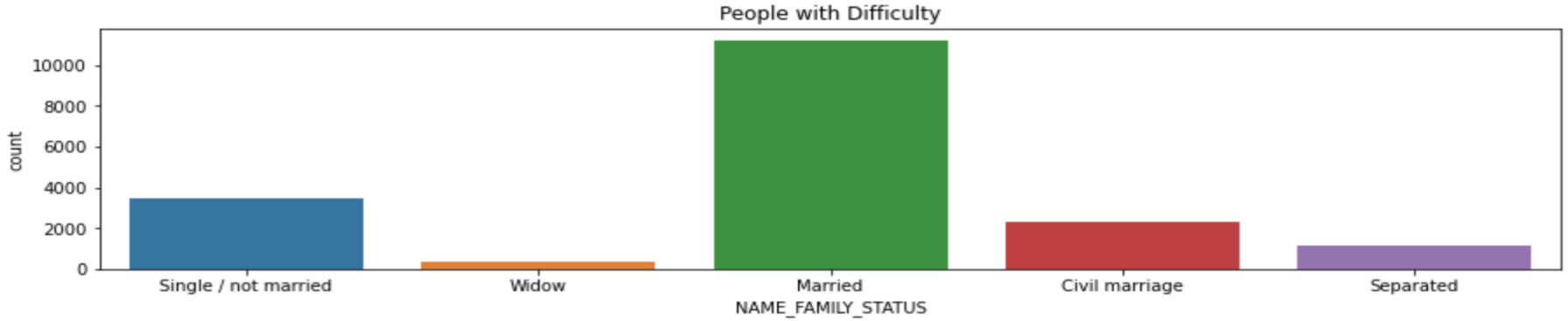
In [121]:
```
inp1_TARGET = inp1_Imbalance.unstack()

plt.figure(figsize=(8,4))

# PLOT THE VALUES FOR 0 AND 1 BASED ON COUNT
plt.subplot(121); inp1_TARGET['counts'].plot(kind = 'bar',color="orange")
plt.title("Count of Category 0 and 1")

# PLOT THE VALUES FOR 0 AND 1 BASED ON PERCENTAGE
plt.subplot(122)
inp1_TARGET['percentage'].plot(kind = 'bar',color="cyan")
plt.title("Percentage % of Category 0 and 1")

plt.show()
```

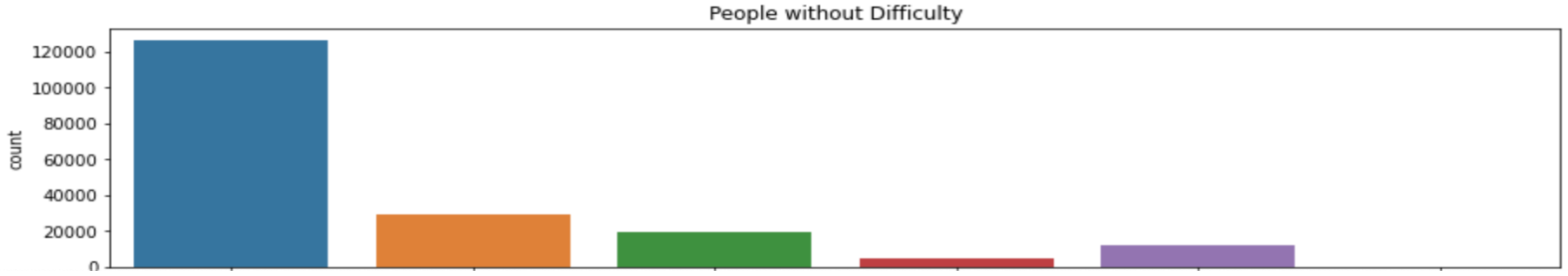


Segmented Univariate Analysis on FAMILY_STATUS

SEGMENTED UNIVARIATE-UNIVARIATE ANALYSIS DISTRIBUTION BASED ON FAMILY TYPE

In [54]:
```
# PEOPLE WITH DIFFICULTY
plt.figure(figsize=(15,3))
sns.countplot(inp_One['NAME_FAMILY_STATUS']).set_title("People with Difficulty")
plt.show()

# PEOPLE WITHOUT DIFFICULTY
plt.figure(figsize=(15,3))
sns.countplot(inp_Zero['NAME_FAMILY_STATUS']).set_title("People without Difficulty")
plt.show()
```





Previous_data Analysis

- The SK_ID_CURR is the main focused column.
- The shape of the dataset is 1670214,37.
- The columns greater than 30% of missing values are dropped.
- Outliers are identified and dropped.

Previous_data Analysis

- Converted negative values present in the “Date”column into positive numbers.
- Binning is done on the “DAYS_DECISION”.
- Total count of missing values in the rows are also shown in the notebook.
- Univariate analysis is done.
- Bivariate analysis is done.
- Correlation is shown using HEATMAPS.

Previous_data Analysis

ANALYSIS ON PREVIOUS DATASET

In [78]:
```
pre_ds=pd.read_csv("previous_application.csv")
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 500)

pre_ds.head()
```

Out[78]:

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

In [79]:
```
pre_ds.shape
```

Out[79]: (1670214, 37)

Merging on SK_ID_CURR

We can see the coloumn SK_ID_CURR is same in both the data frames, indexing it using pivot table functions.

In [81]:
```
df2 = pd.pivot_table(pre_ds,index=["SK_ID_CURR","NAME_CONTRACT_TYPE"])
df2.head(3)
```

Out[81]:

		AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS_PRICE	CNT_PAYMENT	DAYS_D
SK_ID_CURR	NAME_CONTRACT_TYPE							
100001	Consumer loans	3951.000	24835.5	23787.0	2520.0	24835.5	8.0	
100002	Consumer loans	9251.775	179055.0	179055.0	0.0	179055.0	24.0	
100003	Cash loans	98356.995	900000.0	1035882.0	NaN	900000.0	12.0	

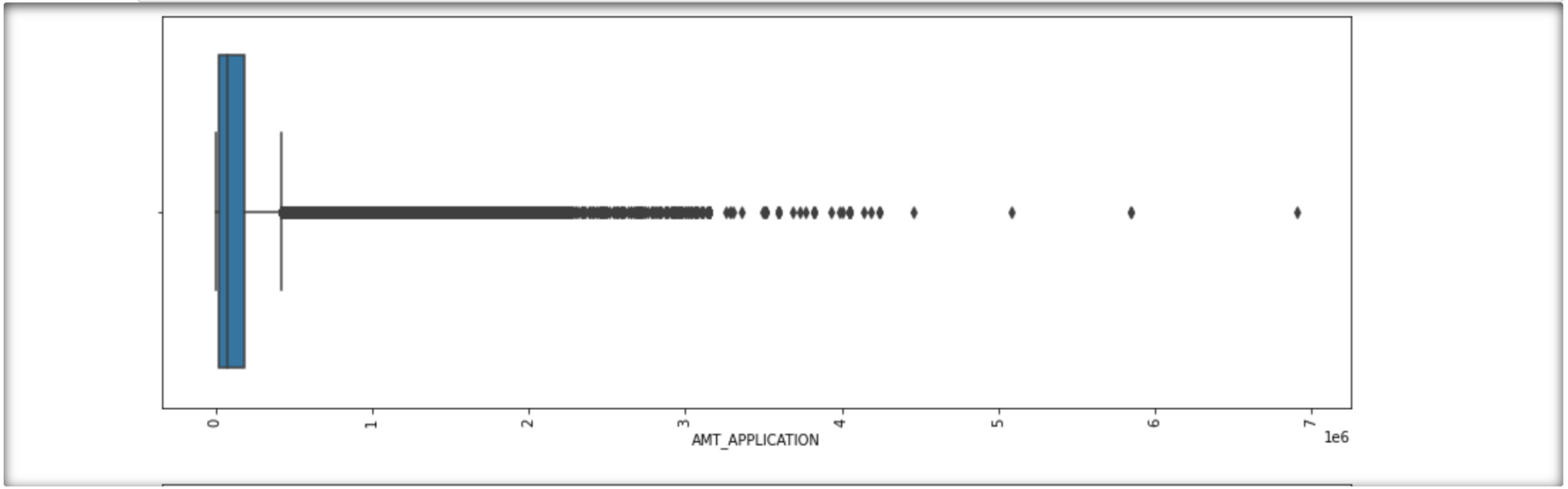
In [ ]:
```
```

Identifying OUTLIERS

IDENTIFYING OUTLIERS

BOX CHART FOR THE "PREVIOUS" APPLICATION'S QUANTITATIVE VARIABLES

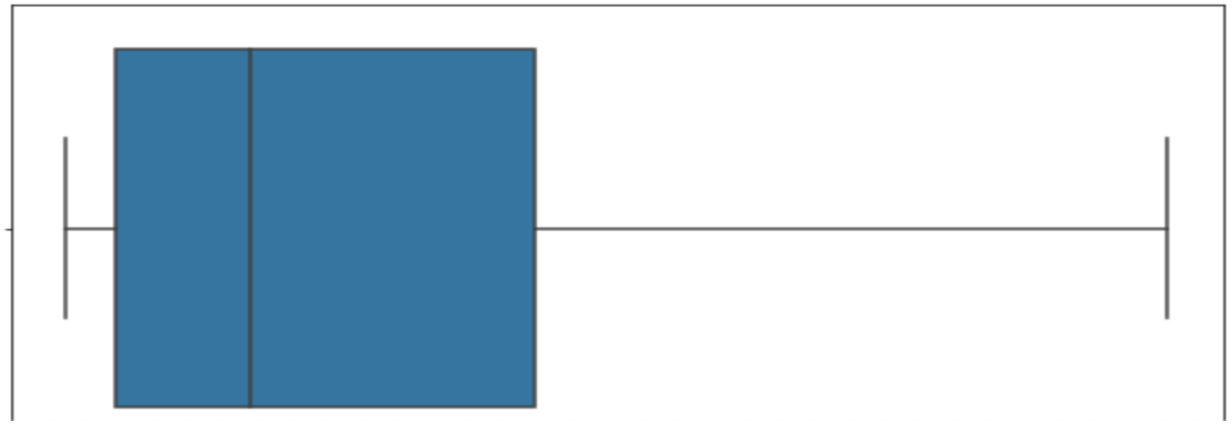
In [94]:
```
prev_box = ['AMT_ANNUITY', 'AMT_APPLICATION', 'AMT_CREDIT', 'AMT_GOODS_PRICE', 'DAYS_DECISION', 'CNT_PAYMENT']
for i in Df2[prev_box]:
    plt.figure(1,figsize=(15,5))
    sns.boxplot(Df2[i])
    plt.xticks(rotation = 90,fontsize =10)
    plt.show()
```



OUTLIERS TREATMENT

In [107]:
```
prev_box_Df2 = ['AMT_APPLICATION','AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE','DAYS_DECISION']
clean_prev_Df2 = Df_prev[prev_box_Df2]
for i in clean_prev_Df2.columns:
    Q1 = clean_prev_Df2[i].quantile(0.25)
    Q3 = clean_prev_Df2[i].quantile(0.75)

    IQR = Q3 - Q1

    lower_fence = Q1 - 1.5*IQR
    upper_fence = Q3 + 1.5*IQR

    clean_prev_Df2[i][clean_prev_Df2[i] <= lower_fence] = lower_fence
    clean_prev_Df2[i][clean_prev_Df2[i] >= upper_fence] = upper_fence

    print(lower_fence,upper_fence)

    plt.figure(1,figsize=(10,5))
    sns.boxplot(clean_prev_Df2[i])
    plt.xticks(rotation =90,fontsize =10)
    plt.show()
```

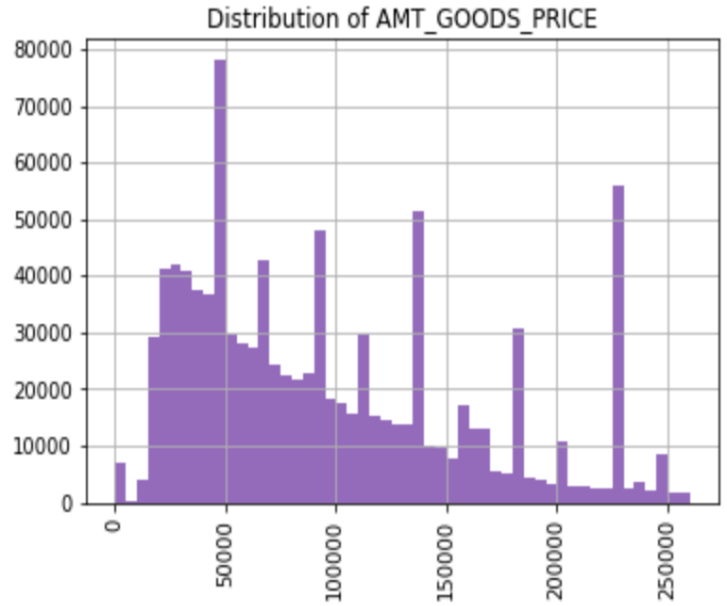
```
-223740.0 422820.0
```



UNIVARIATE ANALYSIS

UNIVARIATE ANALYSIS - AMT_GOODS_PRICE

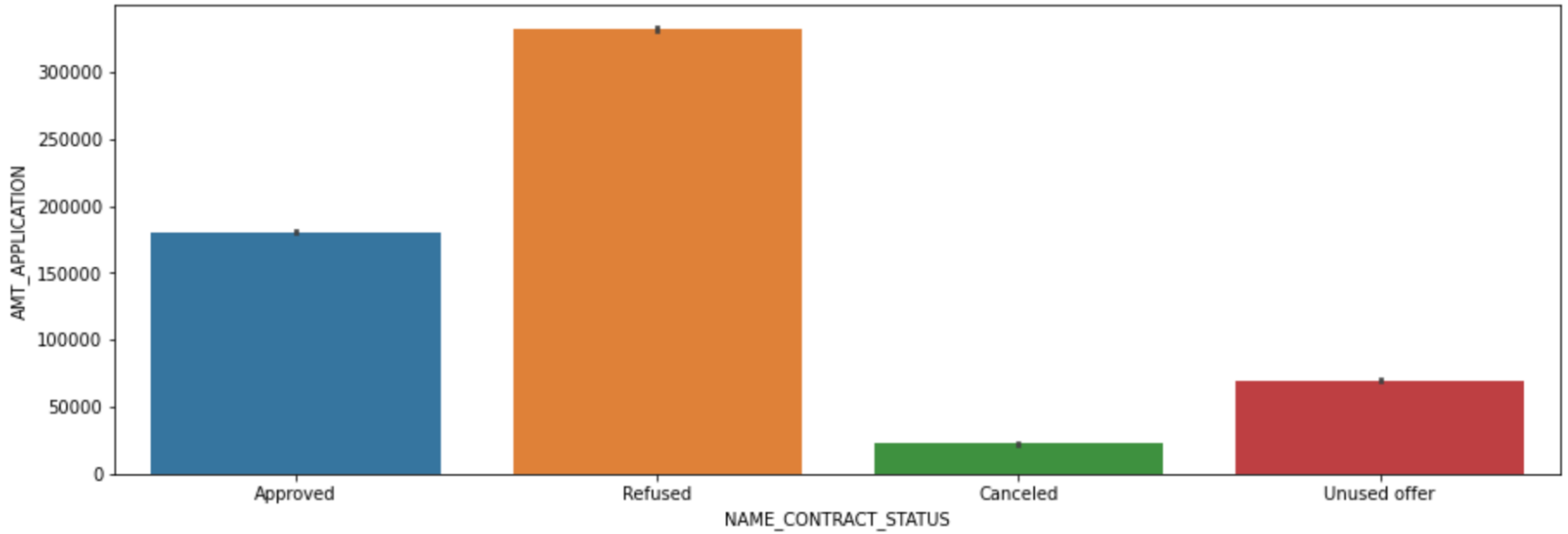
In [113]:
```
clean_prev_Df_AMT_GOODS_PRICE = clean_prev_Df2[['AMT_GOODS_PRICE']]
bins=[0,5000,10000,15000,20000,25000,30000,35000,40000,45000,50000,55000,60000,65000,70000,75000,80000,85000,90000,9500
min = clean_prev_Df_AMT_GOODS_PRICE.describe().min();max = clean_prev_Df_AMT_GOODS_PRICE.describe().max()
range1=[min['AMT_GOODS_PRICE'], max['AMT_GOODS_PRICE']]
clean_prev_Df_AMT_GOODS_PRICE.hist(bins=bins, range=range1, color = ['C4']); plt.title("Distribution of AMT_GOODS_PRICE
plt.xticks(rotation = 90,fontsize =10)
plt.show()
```



BIVARIATE ANALYSIS

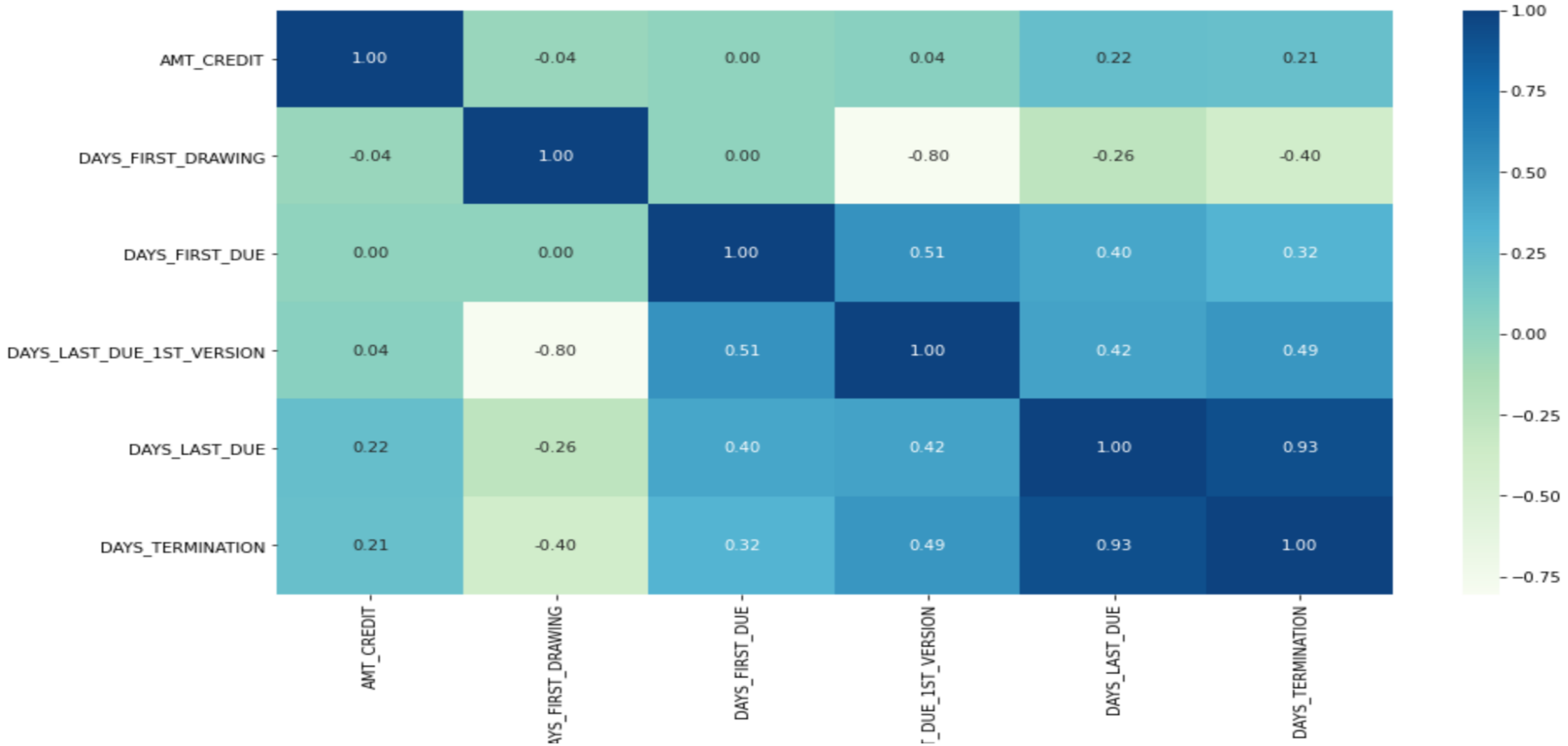
NAME_CONTRACT_STATUS

In [116]:
```
plt.figure(figsize=(15,5))
sns.barplot(x="NAME_CONTRACT_STATUS", y="AMT_APPLICATION",data=Df_prev)
plt.show()
```



CORRELATION SHOWN USING HEATMAP

```
In [119]: test=Df_prev[['AMT_CREDIT','DAYS_FIRST_DRAWING', 'DAYS_FIRST_DUE', 'DAYS_LAST_DUE_1ST_VERSION','DAYS_LAST_DUE', 'DAYS_T
          plt.figure(figsize=(15,8))
          sns.heatmap(test,annot = True, fmt = ".2f", cmap = "GnBu")
          plt.show()
```



Conversion of Negative Numbers into Positive numbers in “DATE”

CHANGE NEGATIVE VALUE TO ABSOLUTE VALUE

In [105]:
```
Df_prev['DAYS_DECISION'] = round(Df_prev['DAYS_DECISION'].abs(),2).head(15)
Df_prev.head(5)
```

Out[105]:

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

Observations on both the DATASETS

- 91.2 % are non-defaulters, 8.79% are defaulters of loan.
- 18.94% of loans have been cancelled by the customer.
- The average age of Clients is around 44 years.
- Most of the loan defaulters come in the income range of 112000 to 202000
- Most of the defaulters loan amount is in the range of 285000 to 733000
- Most of the defaulters loan Annuity is in the range of 24000 to 26000.

Observations on both the DATASETS

- The loan amount sanctioned has a strong correlation with the AMT_GOODS_PRICE and AMT_ANNUITY
- The first drawing, First due, Last Due and Last termination has "NO" bearing to the sanction loan amount.