Course 2 - Task 2: Prepare & Explore Data for Credit One Customer Loan Defaults

Author: Jason Rodriguez

Course Objectives

1. Continued learning focus on Exploratory Data Analysis & PreProcessing Data

Business Problems

1. Credit One is experiencing an ***increase*** in customer defaults which ultimtely can lead to lost in clients and business
2. Credit One does not have an reliable method for identifying which customer may default and how to predict credit balance.

Business Goals

1. Credit One seeking solution to ***predict the right amount of credit*** to extend to customers so it does not over extend themselves and reduce risks with customers defaulting.

Purpose of Remaining Sections

The following sections are broken into sections that step through the process of understanding and making inferences of the available Credit One data. The sections are as follows

1. Setting Up the Environment
2. PreProcessing & Initial Data Analysis
3. EDA: Univariate Analysis
4. EDA: Bivariate Analysis
5. EDA: Multi-variate Analysis
6. EDA: Correlation / Covariance Analysis

1.0 - Setting Up The Environment

In [1]:
```
#IMPORTING LIBRARIES
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import pandas_profiling
import matplotlib as mpl
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
import pylab as pylab
from math import sqrt

#Set Default MatPlot Figure Size
pylab.rcParams['figure.figsize']=(10.0,8.0)

from sklearn import preprocessing
```

In [2]:
```
#SKLearn Components
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, LabelEncoder, MinMaxS
caler
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_squared_error, r2_score, accuracy_sco
re
from sklearn.ensemble import RandomForestRegressor, RandomForestClassi
fier, GradientBoostingRegressor
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR, SVC
```

Data Load

In [3]:
```
#Read Data Source File
credit = pd.read_csv("/Users/JasonRodriguez/Documents/UT-Data-Analytic
s-Program/2020-Cohort/C2-T2/Source-Data/default of credit card clients
.csv")
```

In [4]:
```
#Validate first 5 row in the dataframe and successful data load
credit.head()
```

Out[4]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	B
0	1	20000	female	university	1	24	2	2	-1	-1	...	
1	2	120000	female	university	2	26	-1	2	0	0	...	
2	3	90000	female	university	2	34	0	0	0	0	...	
3	4	50000	female	university	1	37	0	0	0	0	...	
4	5	50000	male	university	1	57	-1	0	-1	0	...	

5 rows × 25 columns

OBSERVATION: 25 columns (features) within dataframe

2.0 - Pre-Processing & Initial Data Analysis

In [5]:
```
#High-level and basic statistical details of the dataframe
credit.describe()
```

Out[5]:

	ID	LIMIT_BAL	MARRIAGE	AGE	PAY_0	PAY_2
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	15000.500000	167484.322667	1.551867	35.485500	-0.016700	-0.133767
std	8660.398374	129747.661567	0.521970	9.217904	1.123802	1.197186
min	1.000000	10000.000000	0.000000	21.000000	-2.000000	-2.000000
25%	7500.750000	50000.000000	1.000000	28.000000	-1.000000	-1.000000
50%	15000.500000	140000.000000	2.000000	34.000000	0.000000	0.000000
75%	22500.250000	240000.000000	2.000000	41.000000	0.000000	0.000000
max	30000.000000	1000000.000000	3.000000	79.000000	8.000000	8.000000

8 rows × 22 columns

Data Inferences:

1. 30k records
2. STD is low among all features except age which could be expected
3. 3 object columns (Sex, Education, and Default status) did not display which are nominal values all others are numeric

In [6]:
```
#Dataframe information - list of featurs, type, non-null count
credit.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   ID                          30000 non-null  int64 
 1   LIMIT_BAL                   30000 non-null  int64 
 2   SEX                         30000 non-null  object
 3   EDUCATION                   30000 non-null  object
 4   MARRIAGE                    30000 non-null  int64 
 5   AGE                         30000 non-null  int64 
 6   PAY_0                       30000 non-null  int64 
 7   PAY_2                       30000 non-null  int64 
 8   PAY_3                       30000 non-null  int64 
 9   PAY_4                       30000 non-null  int64 
 10  PAY_5                       30000 non-null  int64 
 11  PAY_6                       30000 non-null  int64 
 12  BILL_AMT1                   30000 non-null  int64 
 13  BILL_AMT2                   30000 non-null  int64 
 14  BILL_AMT3                   30000 non-null  int64 
 15  BILL_AMT4                   30000 non-null  int64 
 16  BILL_AMT5                   30000 non-null  int64 
 17  BILL_AMT6                   30000 non-null  int64 
 18  PAY_AMT1                    30000 non-null  int64 
 19  PAY_AMT2                    30000 non-null  int64 
 20  PAY_AMT3                    30000 non-null  int64 
 21  PAY_AMT4                    30000 non-null  int64 
 22  PAY_AMT5                    30000 non-null  int64 
 23  PAY_AMT6                    30000 non-null  int64 
 24  default payment next month  30000 non-null  object
dtypes: int64(22), object(3)
memory usage: 5.7+ MB
```

OBSERVATION: All columns are non-null, 3 object types, 22 integer types

In [7]:
```
#Pandas Profiling
#pandas_profiling.ProfileReport(credit)
```

In [8]:
```
#Check for Missing Values
print(credit.isnull().sum())
```

```
ID                            0
LIMIT_BAL                     0
SEX                           0
EDUCATION                     0
MARRIAGE                      0
AGE                           0
PAY_0                         0
PAY_2                         0
PAY_3                         0
PAY_4                         0
PAY_5                         0
PAY_6                         0
BILL_AMT1                     0
BILL_AMT2                     0
BILL_AMT3                     0
BILL_AMT4                     0
BILL_AMT5                     0
BILL_AMT6                     0
PAY_AMT1                      0
PAY_AMT2                      0
PAY_AMT3                      0
PAY_AMT4                      0
PAY_AMT5                      0
PAY_AMT6                      0
default payment next month    0
dtype: int64
```

OBSERVATION: no features contain null values

In [9]:
```
#Print Column Names within Dataframe
credit.columns
```

Out[9]:
```
Index(['ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PA
Y_0',
       'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BI
LL_AMT2',
       'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1
',
       'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',
       'default payment next month'],
      dtype='object')
```

In [10]:
```
#Rename Columns to drive consistency within dateframe
credit = credit.rename(columns={'default payment next month': 'Default
_Status', 'PAY_0':'PAY_1'})
```

In [11]:
```
#validation check to see if PAY_0 and Default Payment Next Month were
changed
credit.columns
```

Out[11]:
```
Index(['ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PA
Y_1',
       'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BI
LL_AMT2',
       'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1
',
       'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',
       'Default_Status'],
      dtype='object')
```

In [12]:
```
#Drop Un-needed columns
credit = credit.drop(['ID'], axis=1)
```

***Note* dropping ID feature is a good practice given it does not add value in Machine learning it would give the ID column more strength b/c it so linear**

In [13]:
```
#Validating ID column was droppped
credit.columns
```

Out[13]:
```
Index(['LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE', 'PAY_1',
'PAY_2',
       'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'BILL_AMT1', 'BILL_AMT2',
       'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1
',
       'PAY_AMT2', 'PAY_AMT3', 'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6',
       'Default_Status'],
      dtype='object')
```

In [14]:
```
#Drop Duplicates
credit = credit.drop_duplicates()
```

In [15]:
```
#Replacing characters with numeric Values per source definition
credit['SEX'].replace(['male','female'],[1,2], inplace=True) #1=male; 2=female
credit['EDUCATION'].replace(['graduate school', 'university', 'high school', 'other'], [1, 2, 3, 0],inplace=True)
credit['Default_Status'].replace(['default', 'not default'],[1,0],inplace=True) #1-default; 2-not defaulted
```

In [16]:
```
#Binning Age for EDA and visualization purposes only
credit['AGE_BIN2']=pd.cut(credit['AGE'],bins=[20,30,40,50,60,70,80], labels=['20-30', '30-40', '40-50', '50-60', '60-70', '70+'])
credit.AGE_BIN2.unique()
```

Out[16]:
```
[20-30, 30-40, 50-60, 40-50, 60-70, 70+]
Categories (6, object): [20-30 < 30-40 < 40-50 < 50-60 < 60-70 < 70+
]
```

Data Transformation Replace -2, -1 value in all Payment Status columns (PAY_1 to PAY_6) to zero (0). From a business perspective, -2 (no consumption); -1 (paid in full); 0 (use of revolving credit) are all considered good/same and can be changed to zero (0). Otherwise, Machine Learning may consider -2, -1 as bad values. Values 1 to 8 are payment delays and need not be changed.

In [17]:
```
Columns = ['PAY_1', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6']

for i in range(len(Columns)):
    credit[Columns[i]] = credit[Columns[i]].replace(-2, 0)
    credit[Columns[i]] = credit[Columns[i]].replace(-1, 0)
    i = i+1
```

3.0 - EDA: UNIVARIATE ANALYSIS

Objective to explore data distribution of each feature individually within the dataframe

Categorical Features:

1. Sex
2. Education
3. Marriage
4. Default Status
5. Pay 1-6

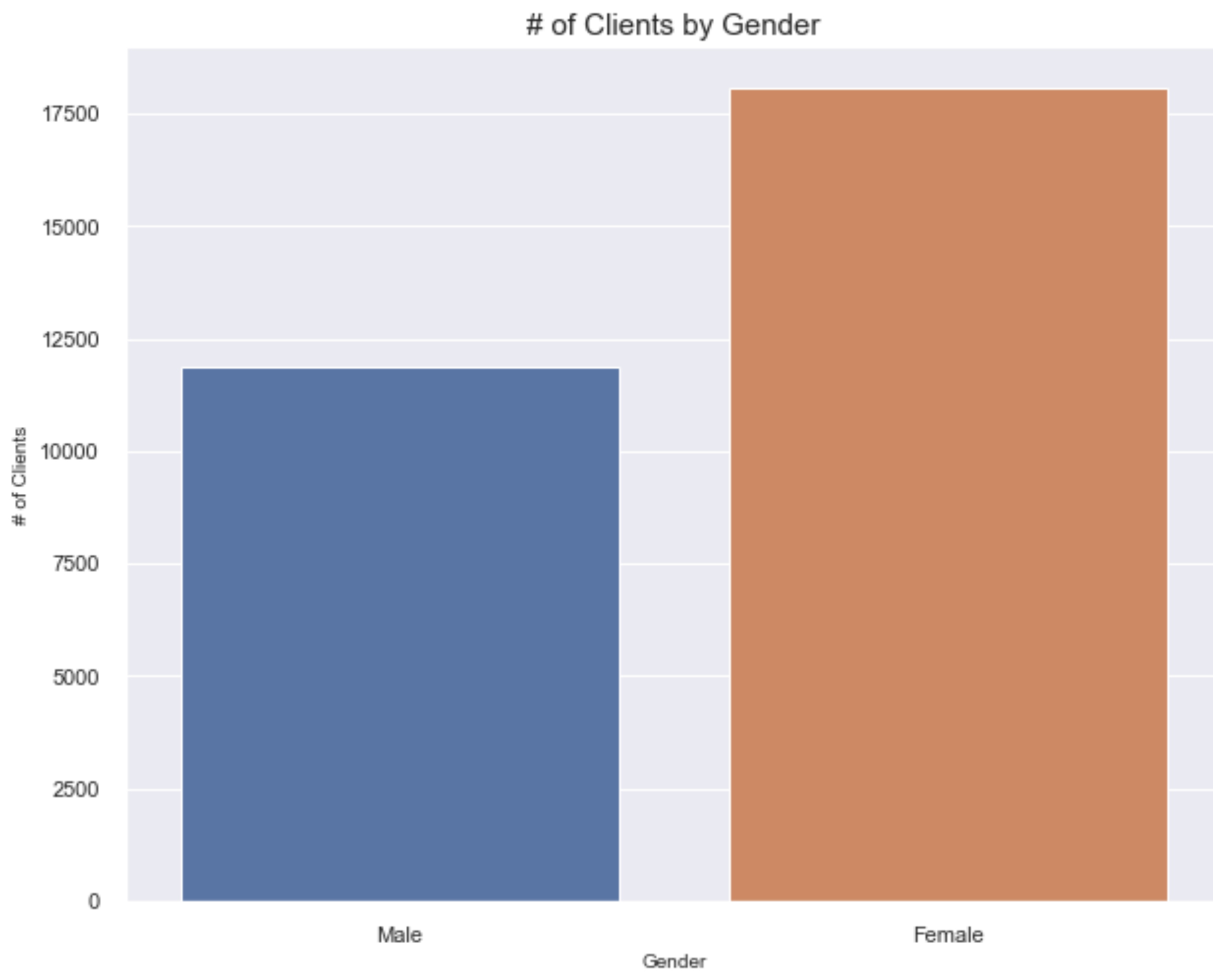
Continuous Featurse:

1. Credit Limit
2. Age
3. Bill Amt 1-6
4. Payment Amt 1-6

--Feature: Gender--

In [18]:
```
# Using Seaborn Countplot to Visualize the Sex distribution

sns.set(style="darkgrid")
gender = sns.countplot(x="SEX",data =credit, palette = 'deep')

# Configure X and Y axis
gender.set_xticklabels(['Male', 'Female'])
gender.set_xlabel("Gender", fontsize=10)
gender.set_ylabel("# of Clients", fontsize=10)

#Set title
plt.title('# of Clients by Gender', fontsize=15)
```

```
Out[18]: Text(0.5, 1.0, '# of Clients by Gender')
```



OBSERVATION:

1. Females are largest population at 60% (18091)
2. Males make up 40% (11874)

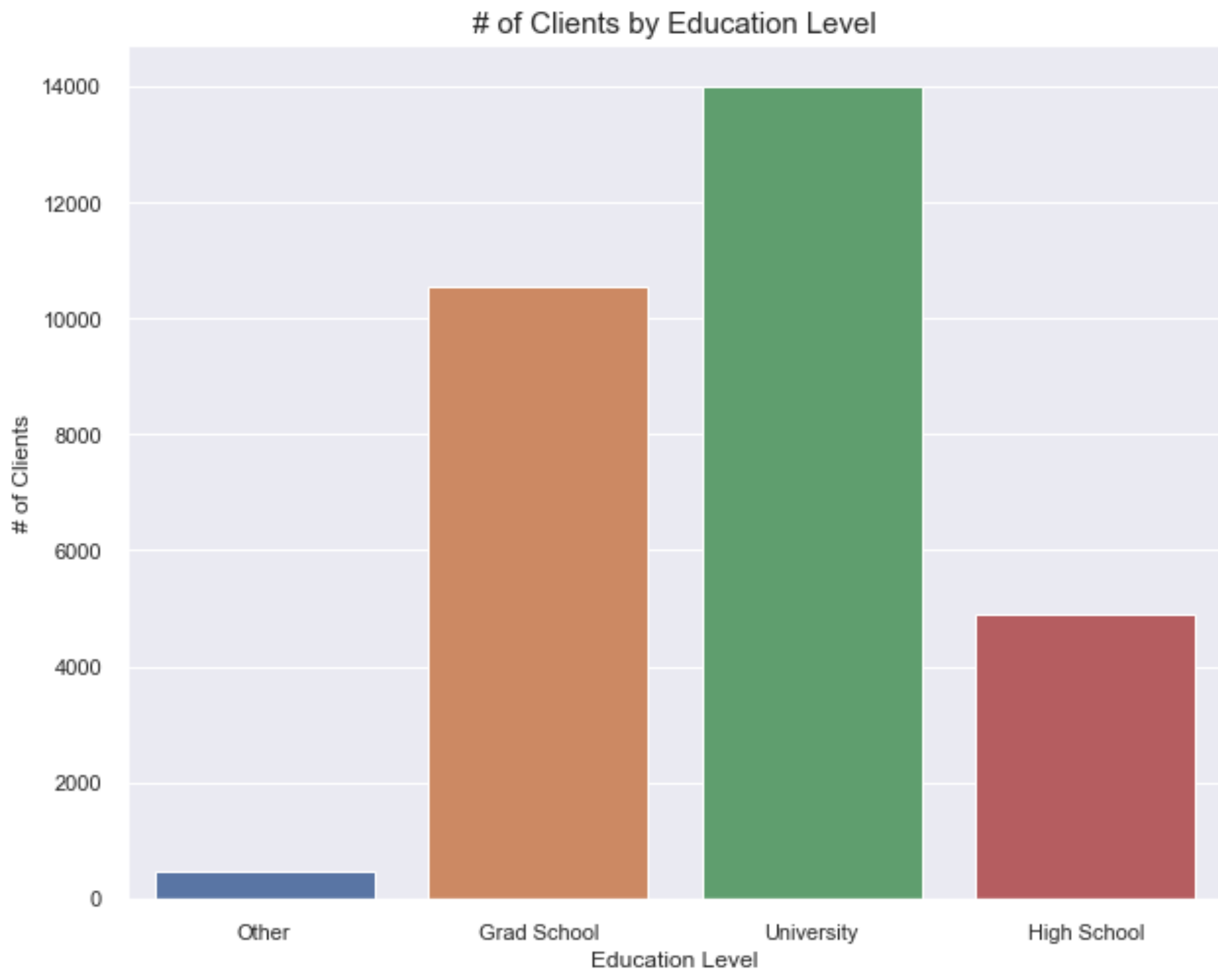
--Feature: Education--

In [19]:
```
#Understand quick count of education across categories
credit['EDUCATION'].value_counts()
```

Out[19]:
```
2    14019
1    10563
3     4915
0      468
Name: EDUCATION, dtype: int64
```

In [20]:
```
# Countplot to Visualize the Education distribution

sns.set(style="darkgrid")
education = sns.countplot(x="EDUCATION",data =credit, palette = 'deep'
)

# Configure X and Y axis
education.set_xticklabels(['Other','Grad School', 'University', 'High
School'])
education.set_xlabel("Education Level", fontsize=12)
education.set_ylabel("# of Clients", fontsize=12)

#Set title
plt.title('# of Clients by Education Level', fontsize=15)
```

Out[20]:
```
Text(0.5, 1.0, '# of Clients by Education Level')
```



OBSERVATION:

1. 82% of clients have post secondary education - 47% (University) & 35% (Grad School)
2. 16% have high school
3. 2% have other

--Feature: Marriage--

In [21]:
```
#Understand quick count of marriage across categories
credit['MARRIAGE'].value_counts()
```

Out[21]:
```
2    15945
1    13643
3      323
0       54
Name: MARRIAGE, dtype: int64
```

OBSERVATION:

0 - Other; 1 - Married; 2 - Single; 3 - Divorce

1. Other - 0.2% of population
2. Married - 45.5% of population
3. Single - 53.2% of population
4. Divorce - 1.1% of population

In [22]:
```
# Countplot to Visualize the Marriage distribution

sns.set(style="darkgrid")
marriage = sns.countplot(x="MARRIAGE",data =credit, palette = 'deep')

# Configure X and Y axis
marriage.set_xticklabels(['Other','Married', 'Single', 'Divorce'])
marriage.set_xlabel("Marriage Status", fontsize=12)
marriage.set_ylabel("# of Clients", fontsize=12)

#Set title
plt.title('# of Clients by Marriage Status', fontsize=15)
```

Out[22]:
```
Text(0.5, 1.0, '# of Clients by Marriage Status')
```

OBSERVATION:

1. Over half of the population base are ***SINGLE*** at 53.2%
2. Second higher concentration are married cliented at 45.5%

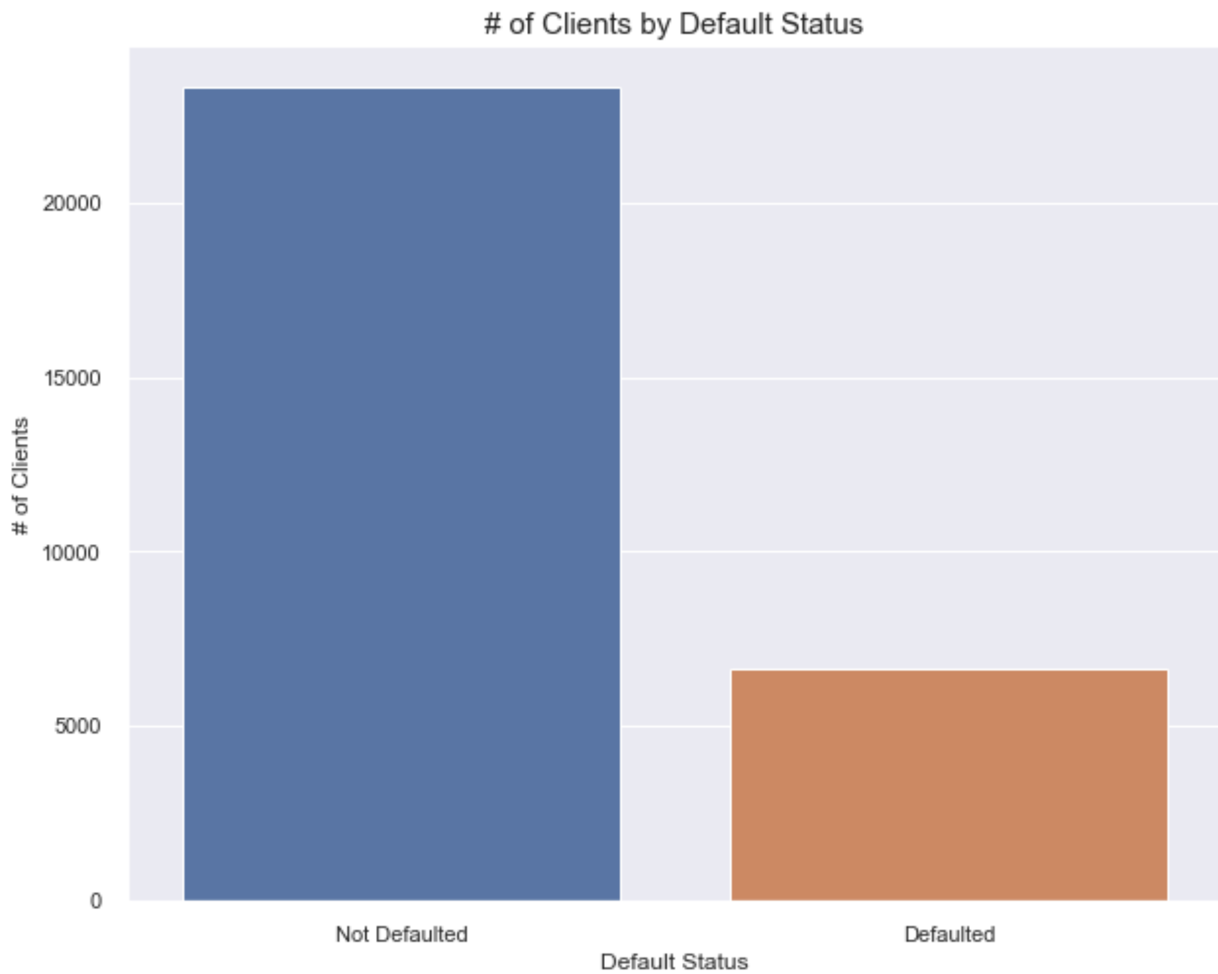
--Feature: Default Status--

In [23]:
```
#Understand quick count of marriage across categories
credit['Default_Status'].value_counts()
```

Out[23]:
```
0    23335
1     6630
Name: Default_Status, dtype: int64
```

In [24]:
```
# Countplot to Visualize the Default Status distribution

sns.set(style="darkgrid")
default = sns.countplot(x="Default_Status",data =credit, palette = 'de
ep')

# Configure X and Y axis
default.set_xticklabels(['Not Defaulted','Defaulted'])
default.set_xlabel("Default Status", fontsize=12)
default.set_ylabel("# of Clients", fontsize=12)

#Set title
plt.title('# of Clients by Default Status', fontsize=15)
```

Out[24]:
```
Text(0.5, 1.0, '# of Clients by Default Status')
```



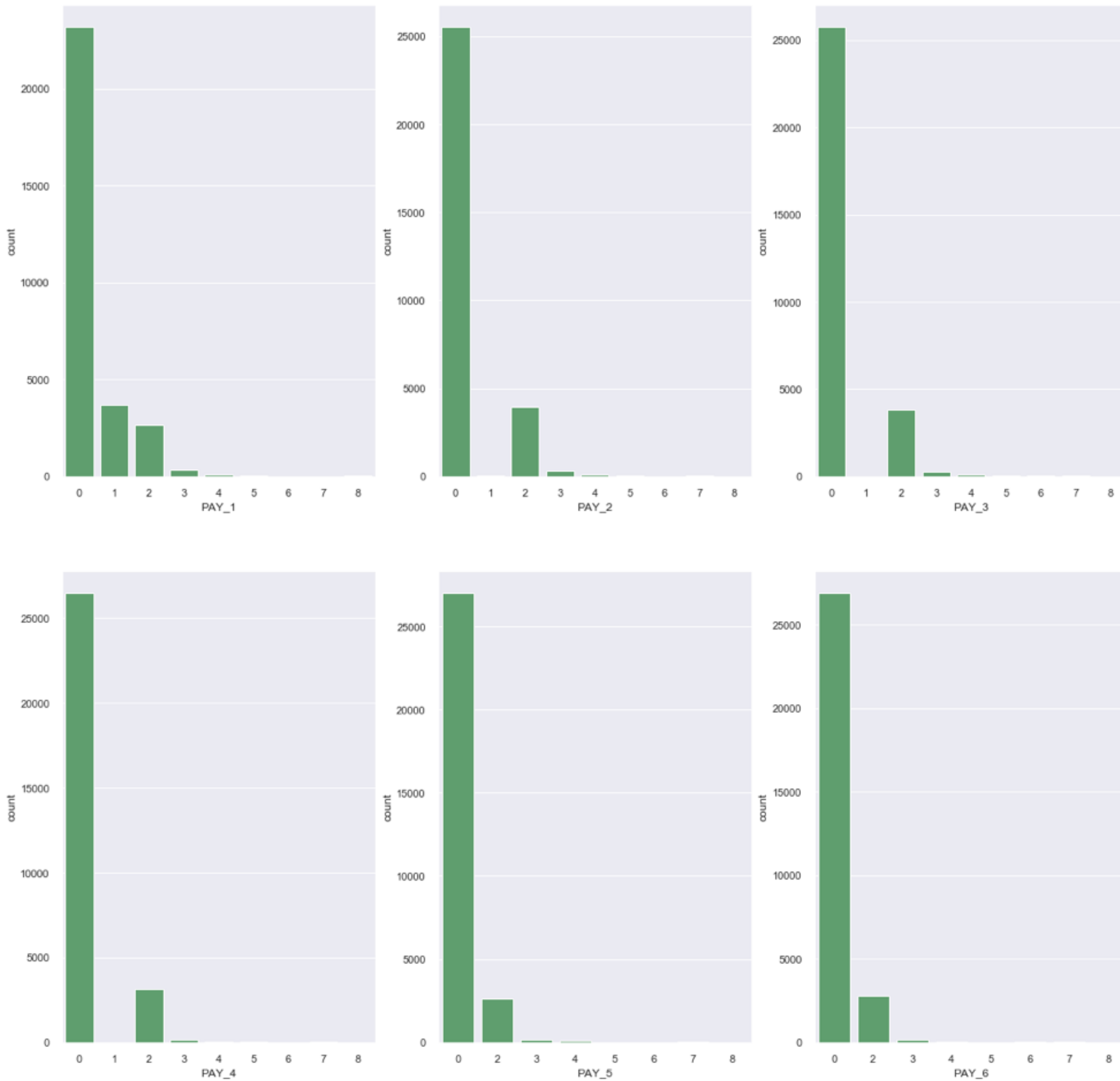
OBSERVATION:

1. 22% of customers ***DEFAULTED*** (6630 clients defaulted)
2. 78% of customers did ***NOT DEFAULT*** (23335 clients did not default)

--Feature: Pay Status--

In [25]:
```
#Set up chart for 6 subplots for 6 columns

fig,ax = plt.subplots(2,3,figsize=(20,20))
sns.set(font_scale=1, style="darkgrid")

#Creating subplots
sns.countplot(x='PAY_1', data=credit, ax=ax[0,0], color='g')
sns.countplot(x='PAY_2', data=credit, ax=ax[0,1], color='g')
sns.countplot(x='PAY_3', data=credit, ax=ax[0,2], color='g')
sns.countplot(x='PAY_4', data=credit, ax=ax[1,0], color='g')
sns.countplot(x='PAY_5', data=credit, ax=ax[1,1], color='g')
sns.countplot(x='PAY_6', data=credit, ax=ax[1,2], color='g')
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11a26b400>
```

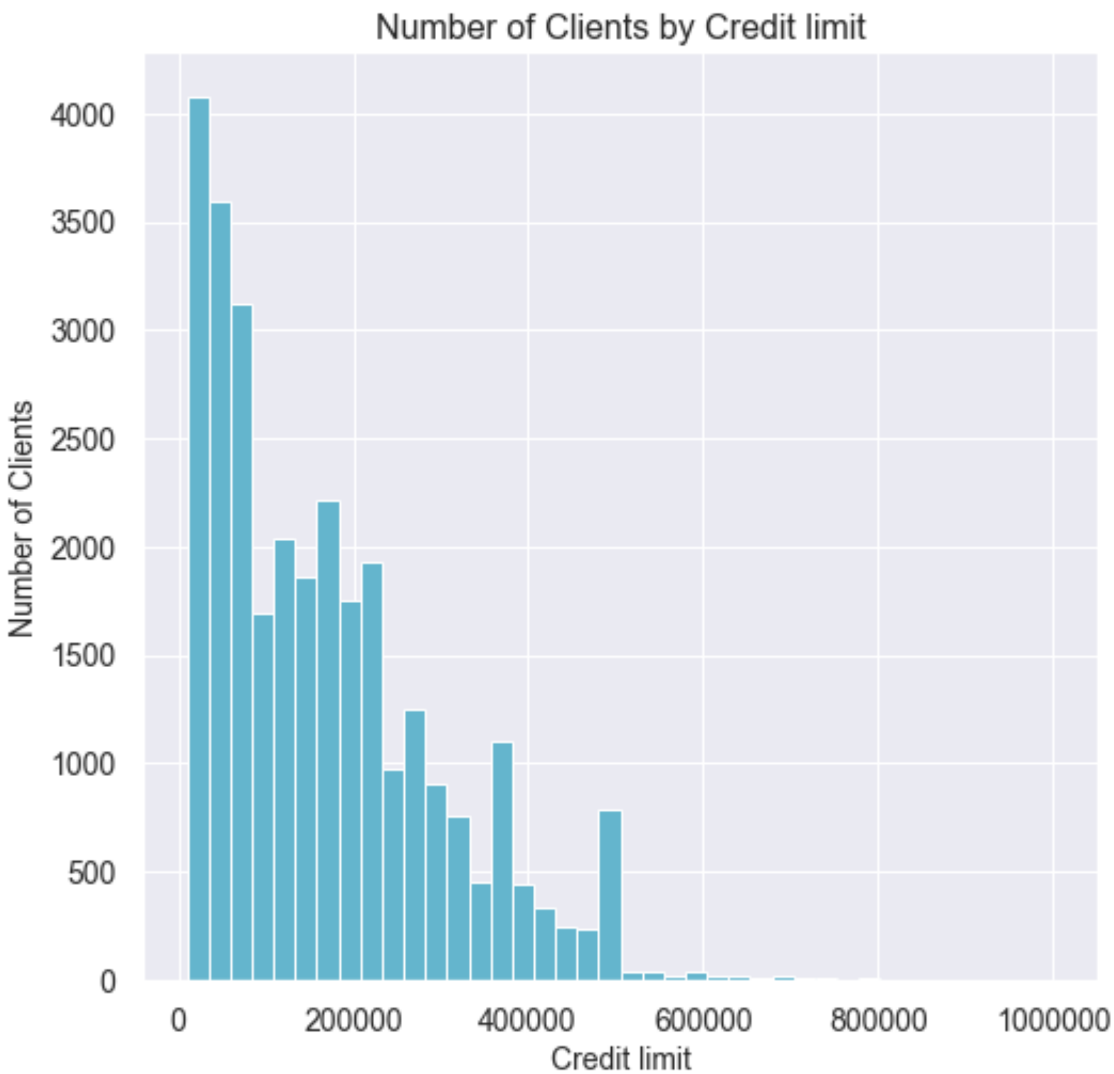


OBSERVATION: For payment the highest value is set at 0 (zero) across Pay_0 to Pay_6

--Feature: Credit Limit--

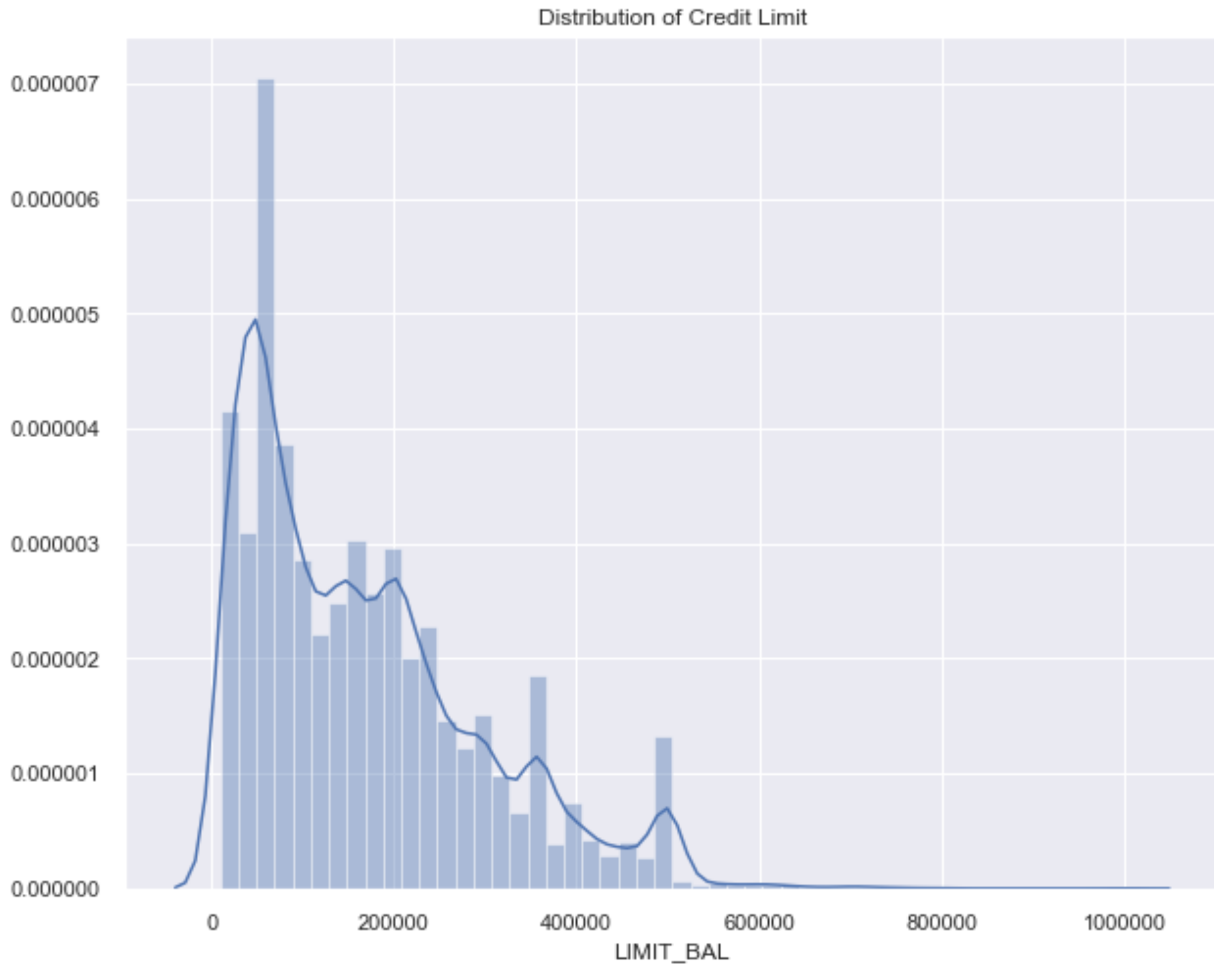
In [26]:
```
plt.figure(figsize=(8,8))

# Plot the graph
plt.hist(credit['LIMIT_BAL'],color="c",bins=40)

# Configure X and Y axis
plt.xlabel('Credit limit', fontsize=14)
plt.ylabel('Number of Clients', fontsize=14)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

# Set the title
plt.title("Number of Clients by Credit limit", fontsize=16)
```

Out[26]:
```
Text(0.5, 1.0, 'Number of Clients by Credit limit')
```



In [27]:
```
#Credit_Limit = credit
cred_lim=sns.distplot(credit['LIMIT_BAL'])
cred_lim.set_title("Distribution of Credit Limit")
```

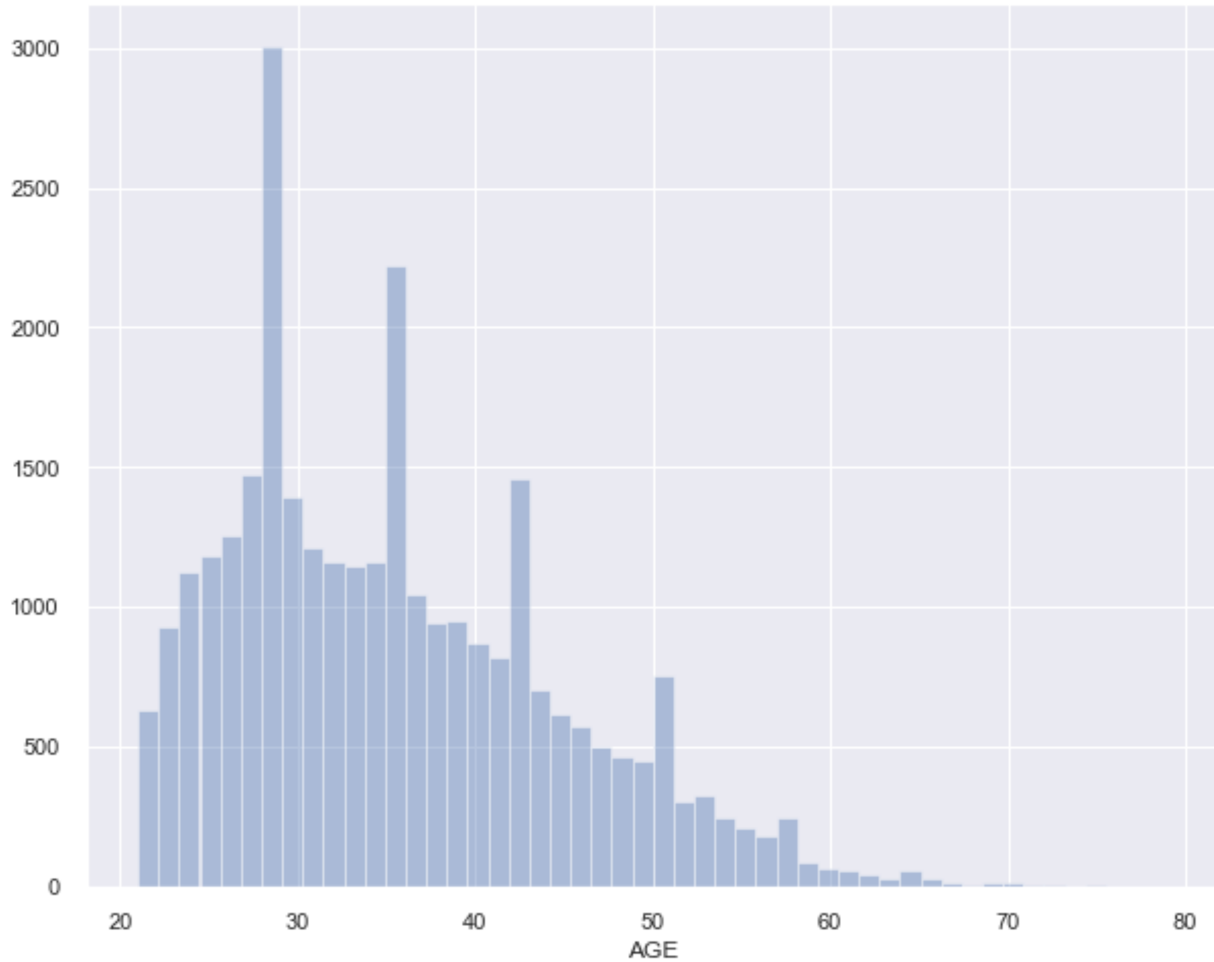
Out[27]:
```
Text(0.5, 1.0, 'Distribution of Credit Limit')
```



In [ ]:

--Feature: Age--

In [28]:
```
#Age distribution
sns.distplot(credit['AGE'],norm_hist=False,kde=False);
```



OBSERVATION:

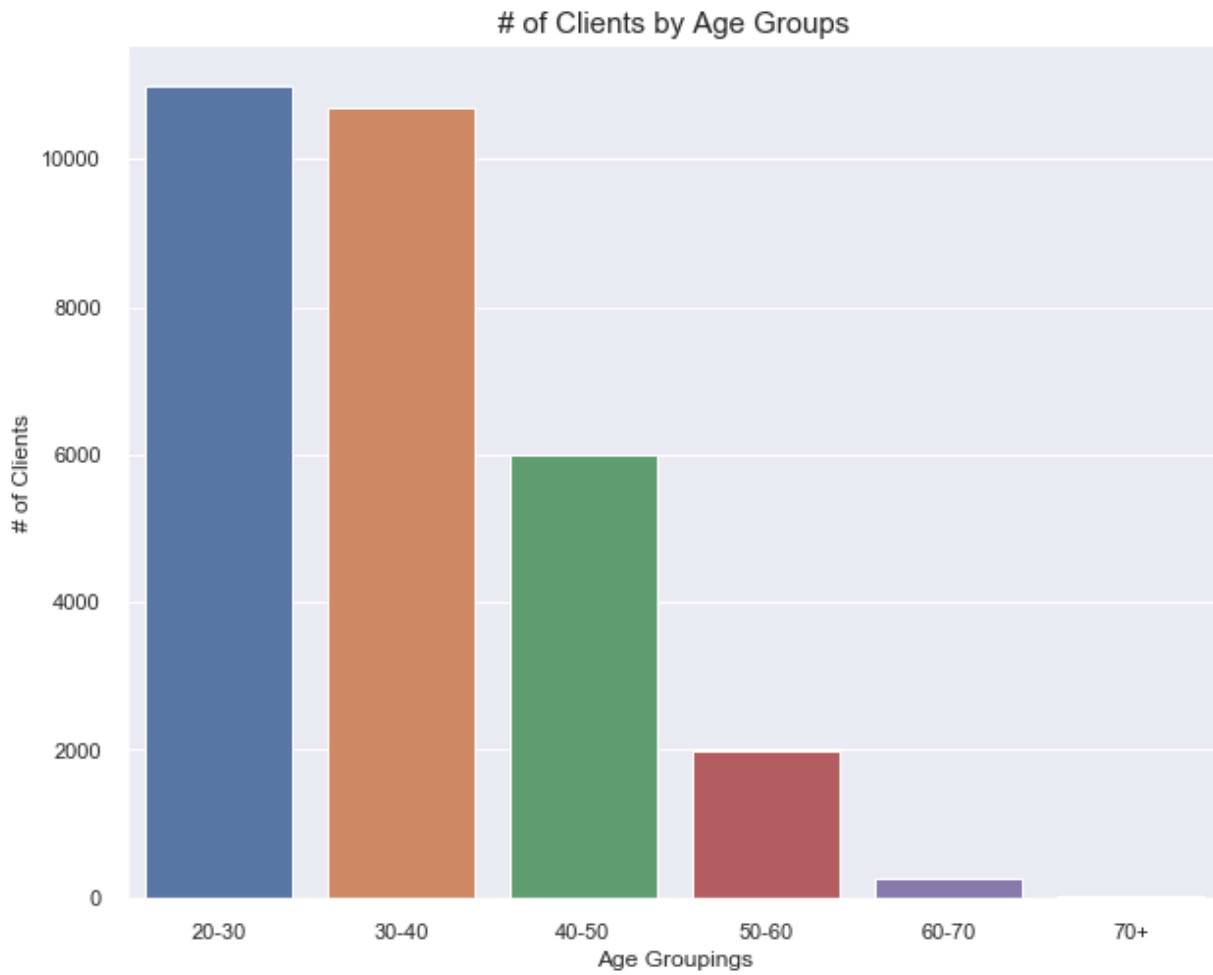
1. Youngest client is about 21 years old
2. Oldest client is about 77 years old
3. Largest client age is about 28 years old
4. Most client is age range from 24 to 38 years old
5. Appears some age groups have spikes in the number of clients

In [29]:
```
#Counting Age Bin
credit['AGE_BIN2'].value_counts()
```

Out[29]:
```
20-30    10997
30-40    10702
40-50     5997
50-60     1997
60-70      257
70+         15
Name: AGE_BIN2, dtype: int64
```

In [30]:
```
# Countplot to Visualize the Age Bin distribution

sns.set(style="darkgrid")
AgeBin = sns.countplot(x="AGE_BIN2",data =credit, palette = 'deep')

# Configure X and Y axis
AgeBin.set_xlabel("Age Groupings", fontsize=12)
AgeBin.set_ylabel("# of Clients", fontsize=12)

#Set title
plt.title('# of Clients by Age Groups', fontsize=15)
```

Out[30]:
```
Text(0.5, 1.0, '# of Clients by Age Groups')
```

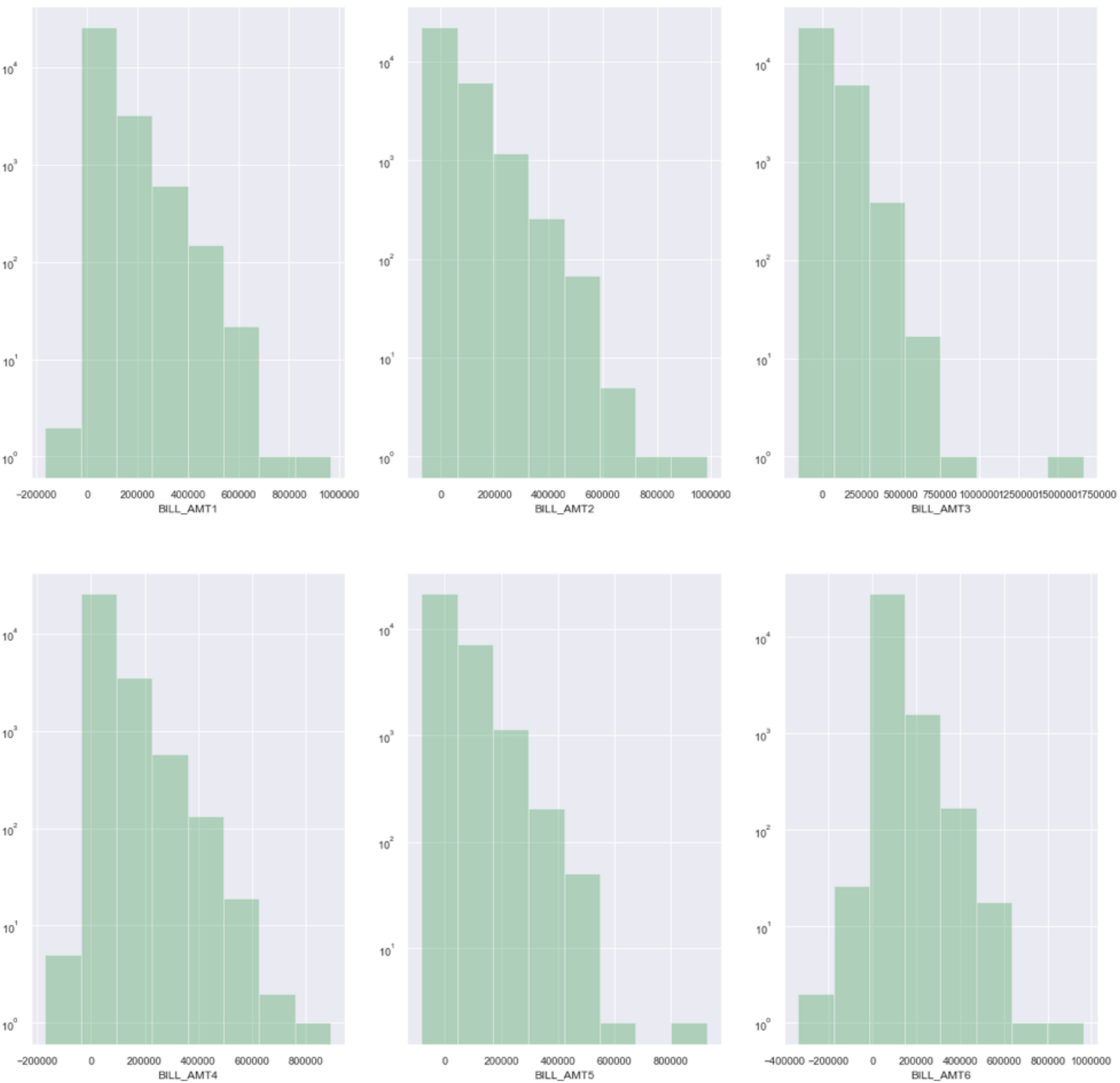


OBSERVATIONS: 20-30 make up 37% of the client population; while 30-40 make up 36% of the client population

--Feature: Bill Amt--

In [31]:
```
#Set up chart for 6 subplots for 6 columns

fig,ax = plt.subplots(2,3,figsize=(20,20))
sns.set(font_scale=1, style="darkgrid")

#Creating subplots
sns.distplot(credit['BILL_AMT1'], bins=8, kde=False, rug=False, ax=ax[
0,0], color='g', hist_kws={'log':True})
sns.distplot(credit['BILL_AMT2'], bins=8, kde=False, rug=False, ax=ax[
0,1], color='g', hist_kws={'log':True})
sns.distplot(credit['BILL_AMT3'], bins=8, kde=False, rug=False, ax=ax[
0,2], color='g', hist_kws={'log':True})
sns.distplot(credit['BILL_AMT4'], bins=8, kde=False, rug=False, ax=ax[
1,0], color='g', hist_kws={'log':True})
sns.distplot(credit['BILL_AMT5'], bins=8, kde=False, rug=False, ax=ax[
1,1], color='g', hist_kws={'log':True})
sns.distplot(credit['BILL_AMT6'], bins=8, kde=False, rug=False, ax=ax[
1,2], color='g', hist_kws={'log':True})
```

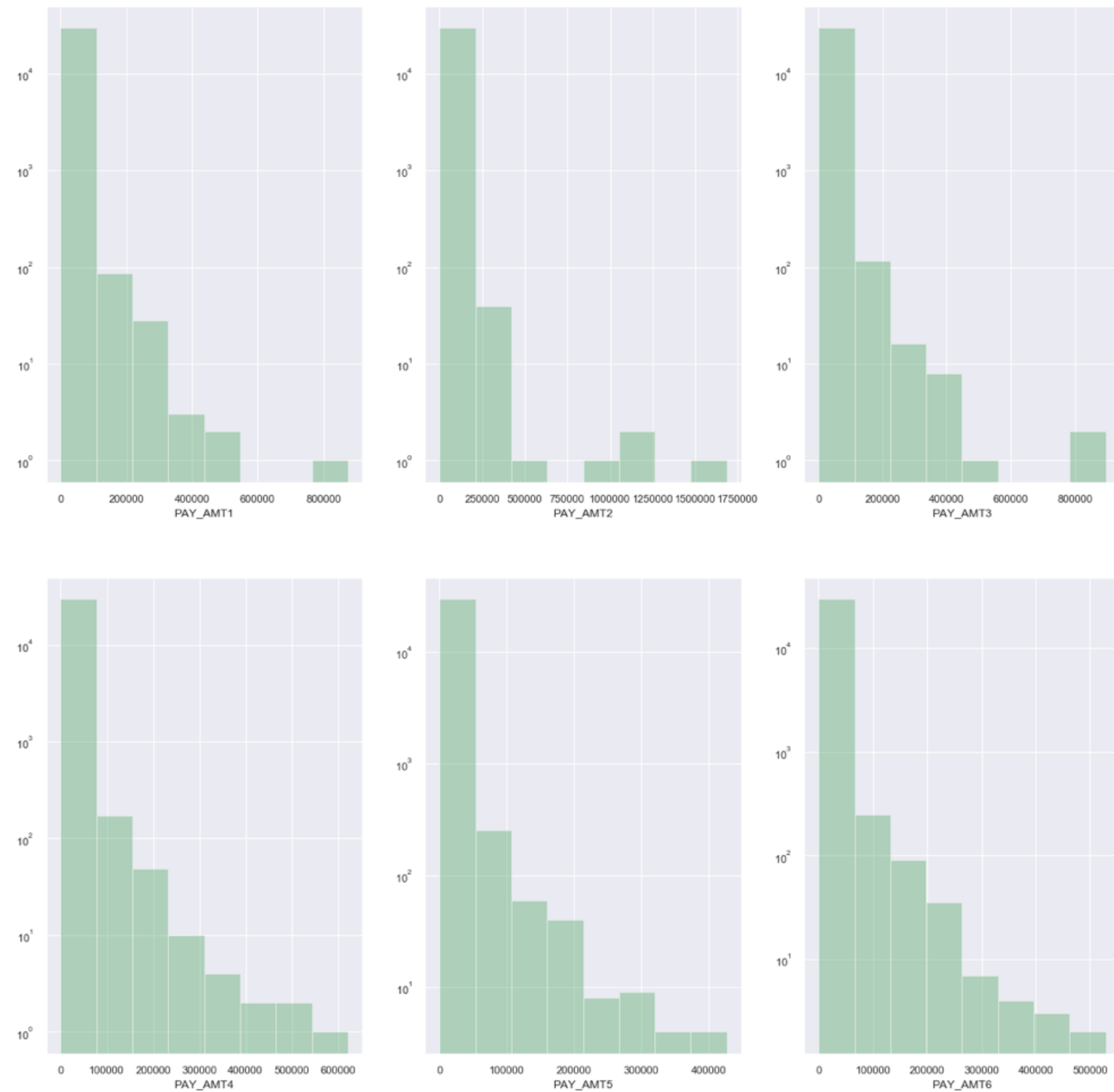
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x11b4ebc70>



OBSERVATION:

In [32]:
```
#Set up chart for 6 subplots for 6 columns

fig,ax = plt.subplots(2,3,figsize=(20,20))
sns.set(font_scale=1, style="darkgrid")

#Creating subplots
sns.distplot(credit['PAY_AMT1'], bins=8, kde=False, rug=False, ax=ax[0
,0], color='g', hist_kws={'log':True})
sns.distplot(credit['PAY_AMT2'], bins=8, kde=False, rug=False, ax=ax[0
,1], color='g', hist_kws={'log':True})
sns.distplot(credit['PAY_AMT3'], bins=8, kde=False, rug=False, ax=ax[0
,2], color='g', hist_kws={'log':True})
sns.distplot(credit['PAY_AMT4'], bins=8, kde=False, rug=False, ax=ax[1
,0], color='g', hist_kws={'log':True})
sns.distplot(credit['PAY_AMT5'], bins=8, kde=False, rug=False, ax=ax[1
,1], color='g', hist_kws={'log':True})
sns.distplot(credit['PAY_AMT6'], bins=8, kde=False, rug=False, ax=ax[1
,2], color='g', hist_kws={'log':True})
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x11c236b50>



OBSERVATION:

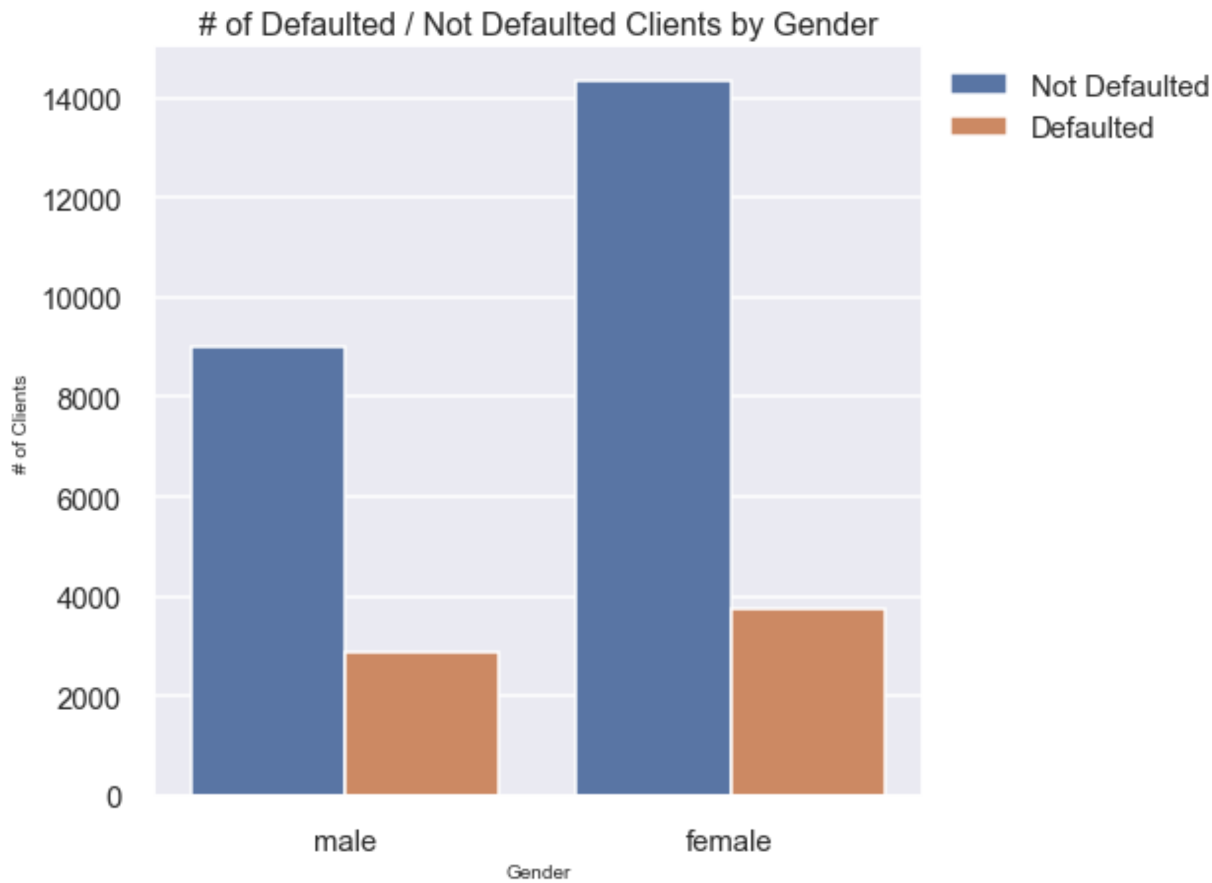
4.0 - EDA: BIVARIATE ANALYSIS

Objective is to explore the relationship of various features to each other. Primary focus will be on Default Status

1. Default Status & Gender
2. Default Status & Education
3. Default Status & Marriage
4. Default Status & Age
5. Default Status & Credit Limit
6. Default Status & Payment
7. Default Status & Bill Amount
8. Default Status & Pay Amount

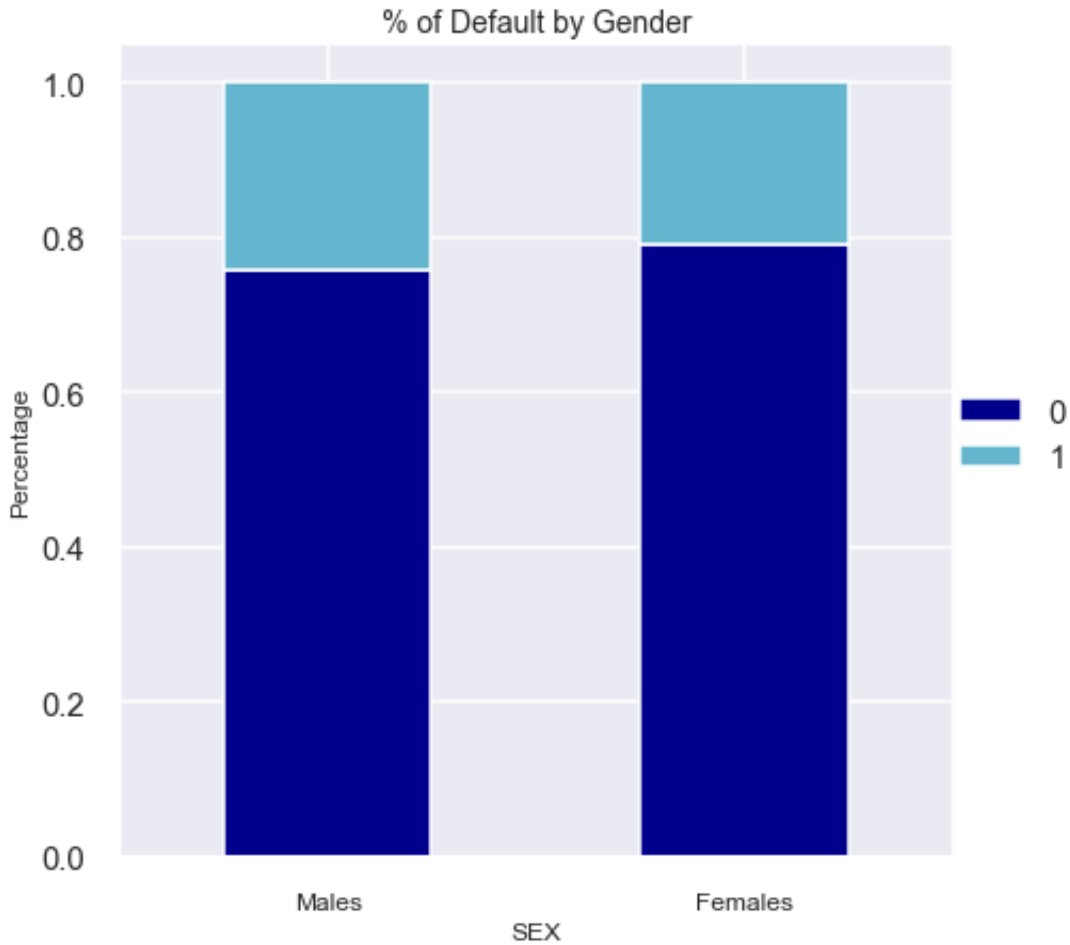
4.1 - Default Status vs Gender

In [33]:
```
#Default vs Sex
sns.set(rc={'figure.figsize':(7,7)})
sns.set_context("talk", font_scale=0.9)

#Set x and y axis labels
genderdefault = sns.countplot(x='SEX', hue='Default_Status', data=cred
it)
genderdefault.set_xticklabels(['male', 'female'])
genderdefault.set_ylabel('# of Clients', fontsize=10)
genderdefault.set_xlabel('Gender', fontsize=10)

#Set Title
plt.title("# of Defaulted / Not Defaulted Clients by Gender", fontsize
=16)

#Set Legend
legend_labels, _=genderdefault.get_legend_handles_labels()
genderdefault.legend(legend_labels, ['Not Defaulted', 'Defaulted'], bb
ox_to_anchor=(1,1))

plt.show()
```



In [34]:
```
#Visualize % Default by Gender
s=pd.crosstab(credit['SEX'], credit['Default_Status']).apply(lambda r:
r/r.sum(), axis=1)

colors = ['darkblue','c']

s.loc[:,:].plot.bar(stacked=True, color=colors, figsize=(7,7))

index = np.arange(2)
labels = ['Males', 'Females']


plt.xlabel('SEX', fontsize=12)
plt.ylabel('Percentage', fontsize=12)
plt.title('% of Default by Gender', fontsize=14)
plt.xticks(index, labels, fontsize=12, rotation=0)

plt.legend(bbox_to_anchor=(1.17, .6))
```

Out[34]: <matplotlib.legend.Legend at 0x11c60c280>



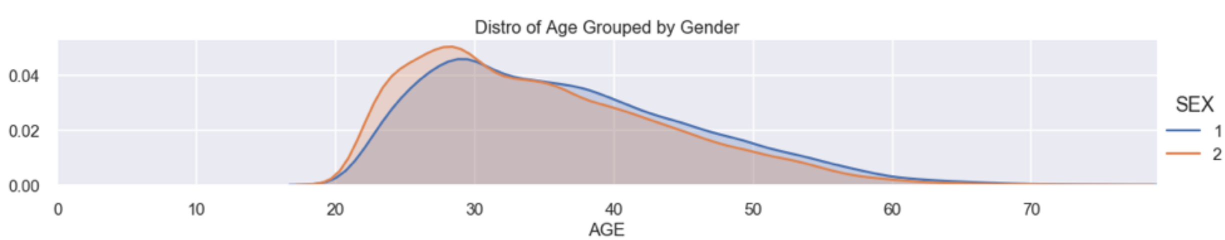
In [35]:
```
#Default count by gender

default0 = credit.groupby(credit['SEX'][credit['Default_Status']==0]).
size().reset_index(name='Not Default')

default1 = credit.groupby(credit['SEX'][credit['Default_Status']==1]).
size().reset_index(name='Default')

total = credit.groupby('SEX').size().reset_index(name='Total')

percent_default = round((default1['Default']/total['Total'])*100,2).re
set_index(name='% Defaulted')
percent_not_default = round((default0['Not Default']/total['Total'])*1
00,2).reset_index(name='% Not Defaulted')

sexTable = default0.join(default1['Default']).join(total['Total']).joi
n(percent_default['% Defaulted']).join(percent_not_default['% Not Defa
ulted'])
sexTable['SEX'] = ['Male', 'Female']
sexTable
```

Out[35]:

	SEX	Not Default	Default	Total	% Defaulted	% Not Defaulted
0	Male	9005	2869	11874	24.16	75.84
1	Female	14330	3761	18091	20.79	79.21

In [36]:
```
#Grouped by Education
fig = sns.FacetGrid(credit, hue='SEX', aspect=5)
fig.map(sns.kdeplot, 'AGE', shade=True)
older=credit['AGE'].max()
fig.set(xlim=(0,older))
fig.set(title='Distro of Age Grouped by Gender')
fig.add_legend()
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x11cb8e970>

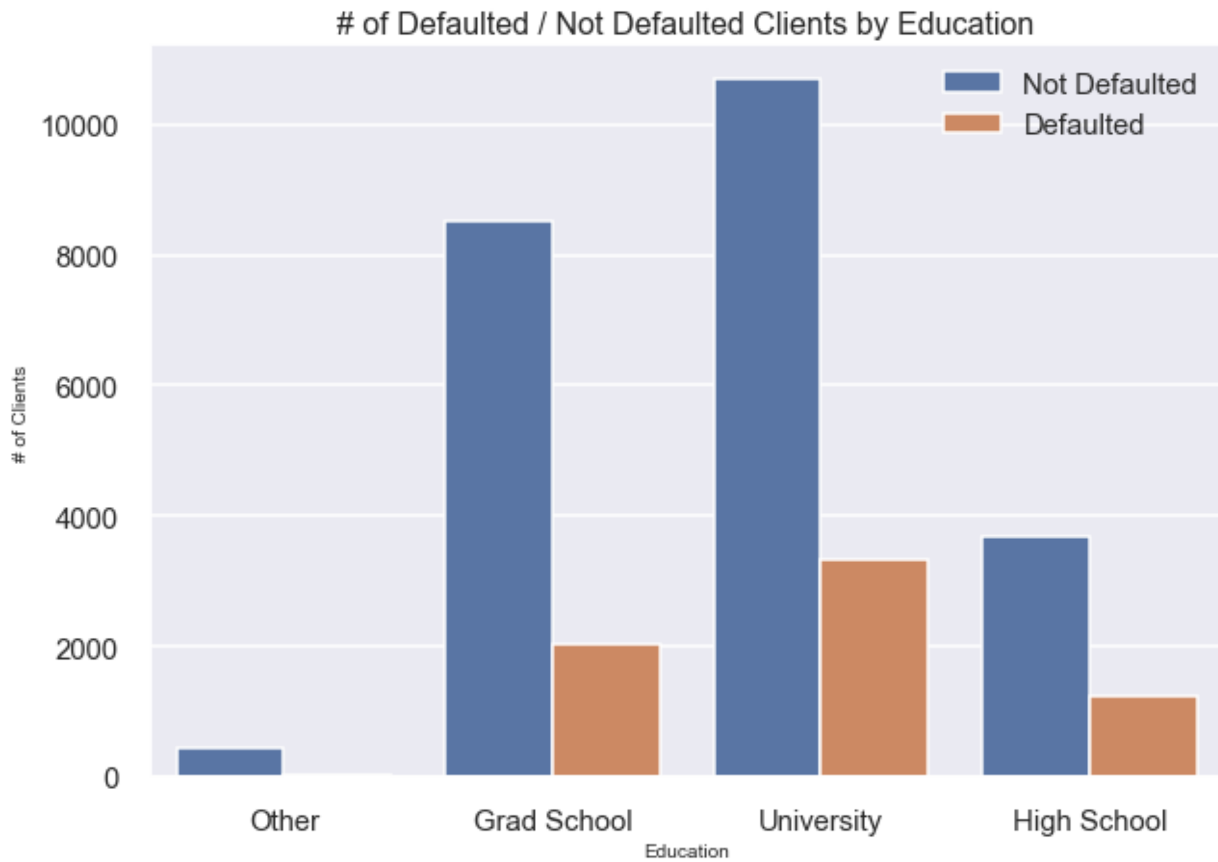


OBSERVATION:

1. 20.79% of Females Default -vs- 24.16% of Males Defaulting
2. Males typically default more often than females by nearly 4% more.
3. Females are largest population at 60% (18091), especially under the age of 30
4. Males make up 40% (11874) and have higher population starting about 35 years of age
5. Nearly ***80%*** of the females population does ***not default***

4.2 - Default Status & Education

In [37]:
```
#Default vs Education
sns.set(rc={'figure.figsize':(10,7)})
sns.set_context("talk", font_scale=0.9)

#Set x and y axis labels
EduDefault = sns.countplot(x='EDUCATION', hue='Default_Status', data=c
redit)
EduDefault.set_xticklabels(['Other', 'Grad School', 'University', 'Hig
h School'])
EduDefault.set_ylabel('# of Clients', fontsize=10)
EduDefault.set_xlabel('Education', fontsize=10)

#Set Title
plt.title("# of Defaulted / Not Defaulted Clients by Education", fonts
ize=16)

#Set Legend
legend_labels, _=EduDefault.get_legend_handles_labels()
EduDefault.legend(legend_labels, ['Not Defaulted', 'Defaulted'], bbox_
to_anchor=(1,1))

plt.show()
```



OBSERVATION:

1. "Other" category appears to be 100% not defaulted but very limit clients in this group
2. Client with 'University' degrees have greatest amount of defaulted & not defaulted but also have the largest population base

In [38]:
```
#Default count by Education level
default0 = credit.groupby(credit['EDUCATION'][credit['Default_Status']==0]).size().reset_index(name='Not Default')

default1 = credit.groupby(credit['EDUCATION'][credit['Default_Status']==1]).size().reset_index(name='Default')

total = credit.groupby('EDUCATION').size().reset_index(name='Total')

percent_default = round((default1['Default']/total['Total'])*100,2).reset_index(name='Percent_Defaulted')
percent_not_default = round((default0['Not Default']/total['Total'])*100,2).reset_index(name='Percent_Not_Defaulted')

eduTable = default0.join(default1['Default']).join(total['Total']).join(percent_default['Percent_Defaulted']).join(percent_not_default['Percent_Not_Defaulted'])
eduTable['EDUCATION'] = ['Other', 'Grad School', 'University', 'High School']
eduTable
```

Out[38]:

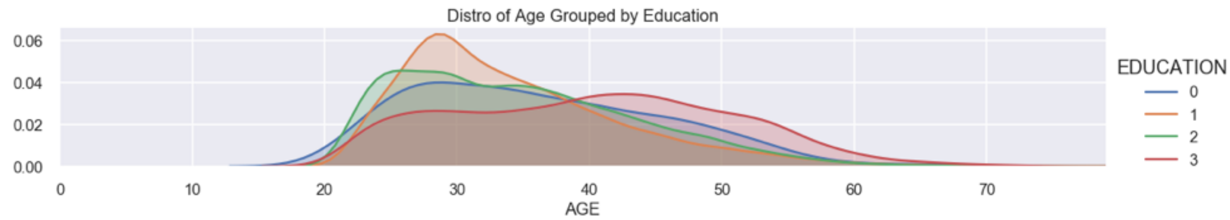
	EDUCATION	Not Default	Default	Total	Percent_Defaulted	Percent_Not_Defaulted
0	Other	435	33	468	7.05	92.95
1	Grad School	8531	2032	10563	19.24	80.76
2	University	10691	3328	14019	23.74	76.26
3	High School	3678	1237	4915	25.17	74.83

OBSERVATION:

1. High School clients have a lower count compare to Grad School and University but ***has higher percentage*** of defaults at 25.17%
2. Other has the lowest count but highest percentage of not defaulting - question: is this due to lower count or something else?
3. 19.24% of Grad clients & 23.73% of University clients default

In [39]:
```
#Grouped by Education
fig = sns.FacetGrid(credit, hue='EDUCATION', aspect=5)
fig.map(sns.kdeplot, 'AGE', shade=True)
older=credit['AGE'].max()
fig.set(xlim=(0,older))
fig.set(title='Distro of Age Grouped by Education')
fig.add_legend()
```

Out[39]: <seaborn.axisgrid.FacetGrid at 0x11944cc10>



OBSERVATION:

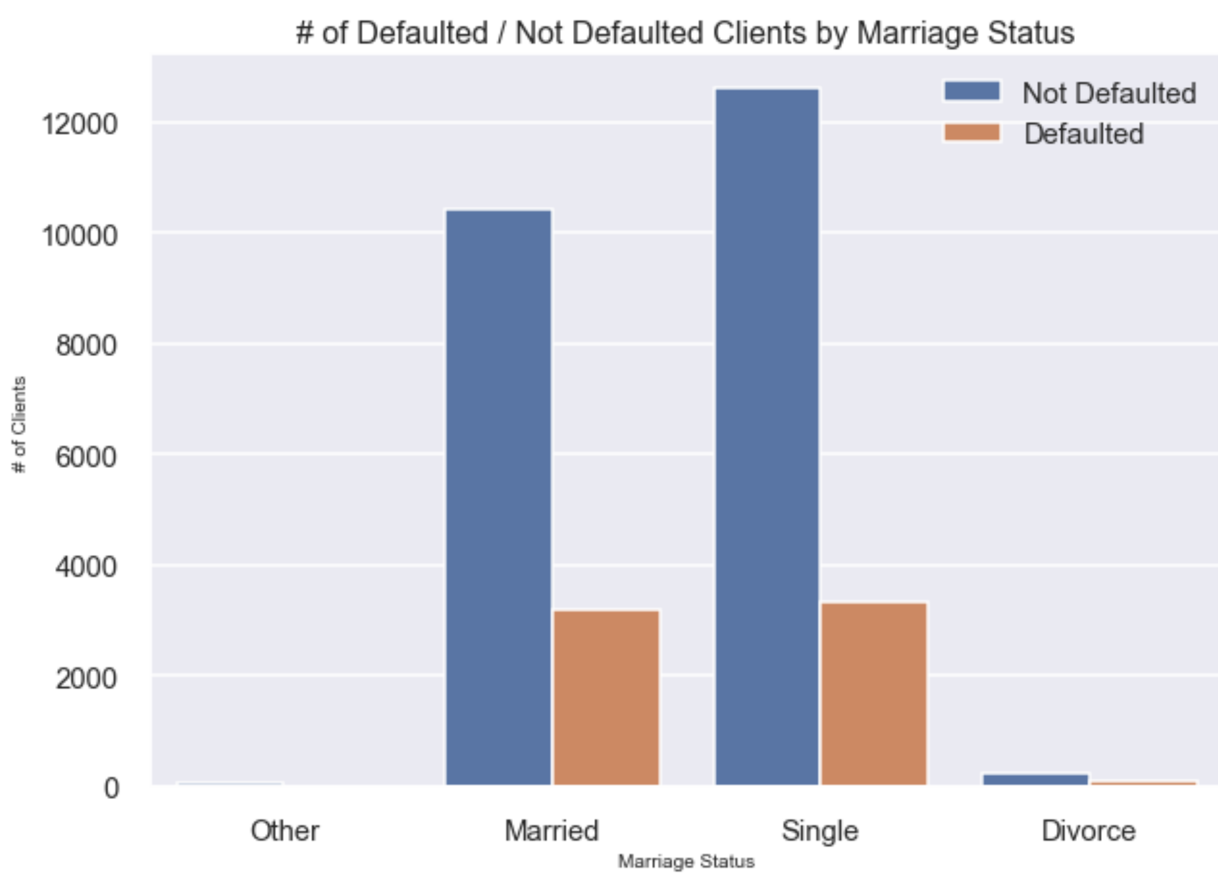
1. Grad School has largest volume between 25 and 35 year old when compare to other education categories
2. High School has largest volume after age 39 years old when compare to other education categories

In [ ]:

4.3 - Default Status & Marriage

In [40]:

```
#Default vs Marriage
sns.set(rc={'figure.figsize':(10,7)})
sns.set_context("talk", font_scale=0.9)

#Set x and y axis labels
MarriageDefault = sns.countplot(x='MARRIAGE', hue='Default_Status', data=credit)
MarriageDefault.set_xticklabels(['Other', 'Married', 'Single', 'Divorce'])
MarriageDefault.set_ylabel('# of Clients', fontsize=10)
MarriageDefault.set_xlabel('Marriage Status', fontsize=10)

#Set Title
plt.title("# of Defaulted / Not Defaulted Clients by Marriage Status", fontsize=16)

#Set Legend
legend_labels, _=MarriageDefault.get_legend_handles_labels()
MarriageDefault.legend(legend_labels, ['Not Defaulted', 'Defaulted'], bbox_to_anchor=(1,1))

plt.show()
```



In [41]:
```
#Default count by marriage

default0 = credit.groupby(credit['MARRIAGE'][credit['Default_Status']==0]).size().reset_index(name='Not Default')

default1 = credit.groupby(credit['MARRIAGE'][credit['Default_Status']==1]).size().reset_index(name='Default')

total = credit.groupby('MARRIAGE').size().reset_index(name='Total')

percent_default = round((default1['Default']/total['Total'])*100,2).reset_index(name='% Defaulted')
percent_not_default = round((default0['Not Default']/total['Total'])*100,2).reset_index(name='% Not Defaulted')

marryTable = default0.join(default1['Default']).join(total['Total']).join(percent_default['% Defaulted']).join(percent_not_default['% Not Defaulted'])
marryTable['MARRIAGE'] = ['Other', 'Married', 'Single', 'Divorce']
marryTable
```

Out[41]:

	MARRIAGE	Not Default	Default	Total	% Defaulted	% Not Defaulted
0	Other	49	5	54	9.26	90.74
1	Married	10442	3201	13643	23.46	76.54
2	Single	12605	3340	15945	20.95	79.05
3	Divorce	239	84	323	26.01	73.99

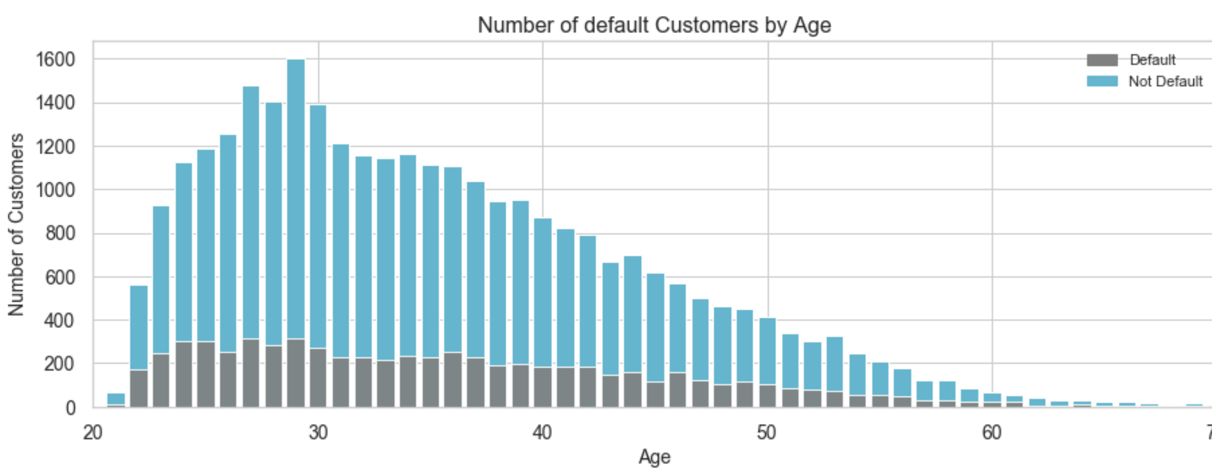
OBSERVATION:

1. Divorce clients have ***higher default percentage*** compare to any other marriage status
2. "Other" clients have ***highest percentage*** of not defaulting on loans
3. Married and Single client are fairly distributed and % of defaults are tight (23%-Married & 20%-Singles)

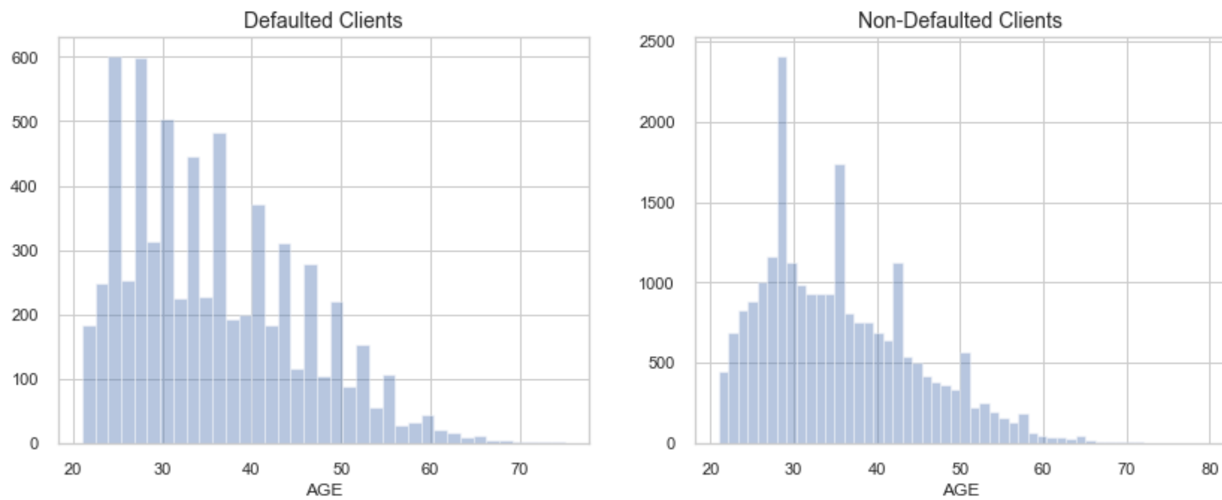
4.4 - Default Status & Age

In [42]:
```
## Plot distribution of Age data with default, not default count
sns.set(style="whitegrid")
plt.figure(figsize=(15,5))

# Plot the graph
total = credit['AGE'].value_counts()
default = credit['AGE'][(credit['Default_Status']==1)].value_counts()

plt.bar(total.index, total, align='center', color='c')
plt.bar(default.index, default, align='center', color='gray', alpha=0.
9)

# Set X and y axis labels
plt.xlabel("Age", fontsize=14)
plt.ylabel('Number of Customers', fontsize=14)
plt.xlim([20,70])
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

# Set the legend
gray_patch = mpatches.Patch(color='gray', label='Default')
c_patch=mpatches.Patch(color='c', label='Not Default')
plt.legend(handles=[gray_patch,c_patch],loc=1)

# Set the title
plt.title ("Number of default Customers by Age", fontsize=16)
```

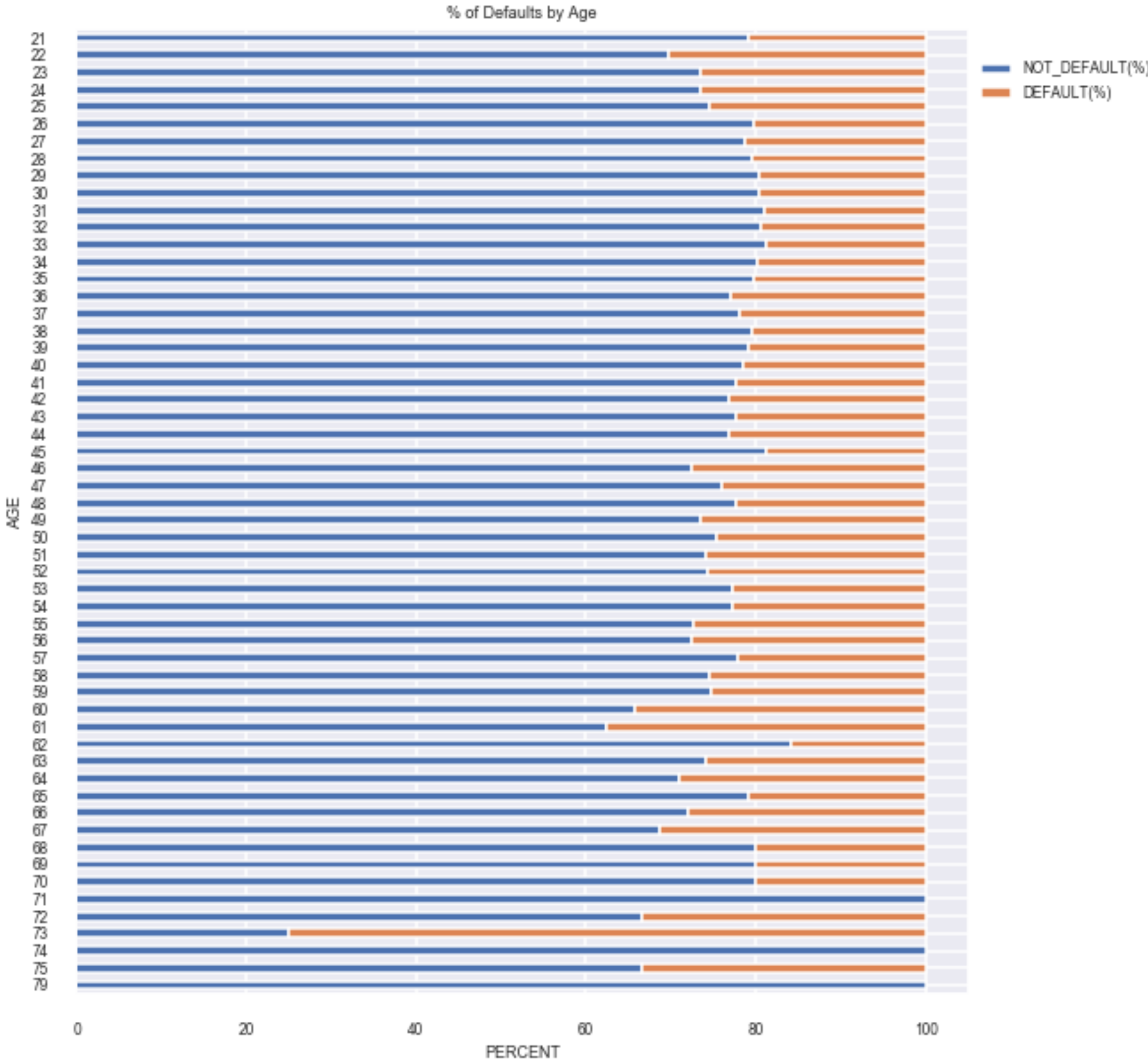
Out[42]:
```
Text(0.5, 1.0, 'Number of default Customers by Age')
```



In [43]:
```
#AGe by Default
fig, (ax2, ax3) = plt.subplots(1, 2, figsize=(14,5))

#ax1.set_title('All Clients', fontsize=14)
ax2.set_title('Defaulted Clients', fontsize=14)
ax3.set_title('Non-Defaulted Clients', fontsize=14)

#sns.distplot(credit['AGE'], norm_hist=False, kde=False, ax=ax1);
sns.distplot(credit['AGE'][credit['Default_Status']==0], norm_hist=Fal
se, kde=False, ax=ax3);
sns.distplot(credit['AGE'][credit['Default_Status']==1], norm_hist=Fal
se, kde=False, ax=ax2);
```



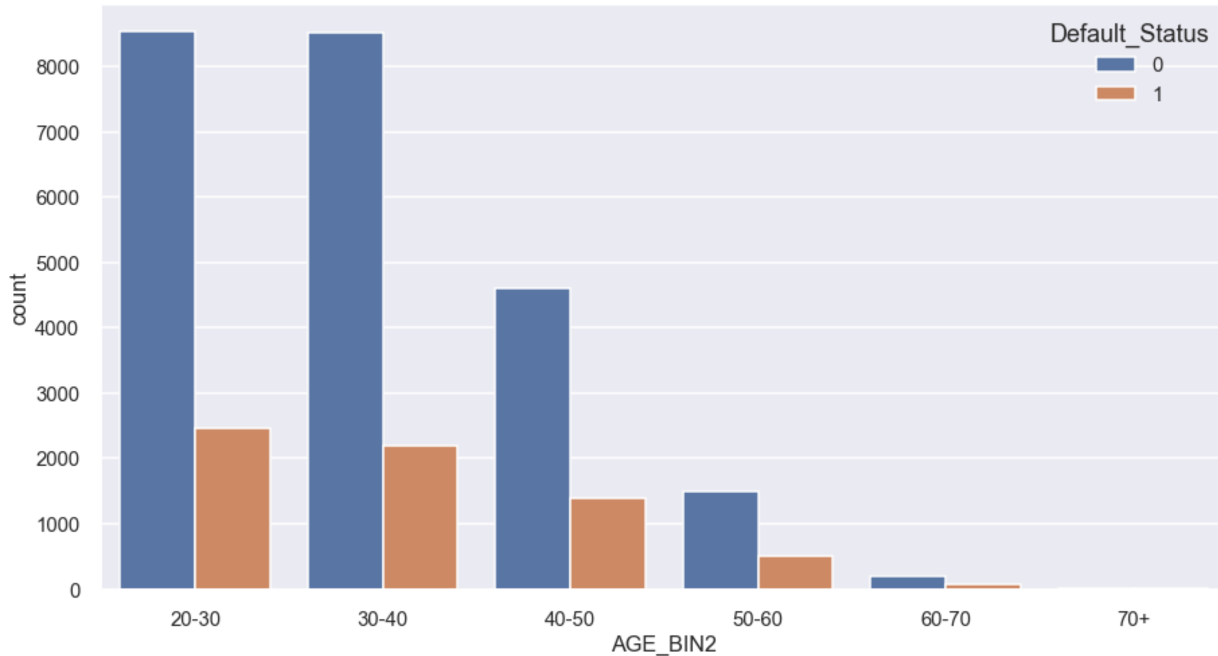
In [44]:
```
#Default Percentage by Age
default0 = credit.groupby(credit['AGE'][credit['Default_Status'] == 0]
).size().reset_index(name='NOT_DEFAULT')
default0 = default0.fillna(0)
default1 = credit.groupby(credit['AGE'][credit['Default_Status'] == 1]
).size().reset_index(name='DEFAULT')
default1 = default1.fillna(0)
total = credit.groupby('AGE').size().reset_index(name='TOTAL')

ageTable = total.join(default0.set_index('AGE'),on='AGE').join(default
1.set_index('AGE'),on='AGE')
ageTable = ageTable[['AGE', 'NOT_DEFAULT', 'DEFAULT', 'TOTAL']]
ageTable = ageTable.fillna(0)
ageTable

ageTable['NOT_DEFAULT'] = round((ageTable['NOT_DEFAULT']/ageTable['TOT
AL'])*100,2)
ageTable['DEFAULT'] = round((ageTable['DEFAULT']/ageTable['TOTAL'])*10
0,2)

agePct = ageTable.iloc[:,0:3]
agePct = agePct.rename(columns={'NOT_DEFAULT': 'NOT_DEFAULT(%)', 'DEFA
ULT': 'DEFAULT(%)'})

agePct


sns.set(rc={'figure.figsize':(9,10)})
sns.set_context("talk", font_scale=0.5)

ax = agePct.sort_index(ascending=False).plot(x='AGE', kind='barh', sta
cked=True, title='% of Defaults by Age')
ax.set_xlabel('PERCENT')
ax.get_legend().set_bbox_to_anchor((1, 0.98))
plt.show()
```



OBSERVATION:

1. 73 year old clients have the have % of defaults
2. Rest of the distribution on % defaulted is fairly consistent with all the other ages

In [45]:
```
#Default vs Age Bin
sns.set(rc={'figure.figsize':(15,8)})
sns.set_context("talk", font_scale=0.9)

ageBin = sns.countplot(x='AGE_BIN2', hue='Default_Status', data=credit
)
plt.show()
```



In [46]:
```
table_age = pd.crosstab(index=[credit.Default_Status], columns=[credit
.AGE_BIN2])
table_age
```

Out[46]:

AGE_BIN2	20-30	30-40	40-50	50-60	60-70	70+
Default_Status						
0	8527	8514	4602	1493	189	10
1	2470	2188	1395	504	68	5

Observation:

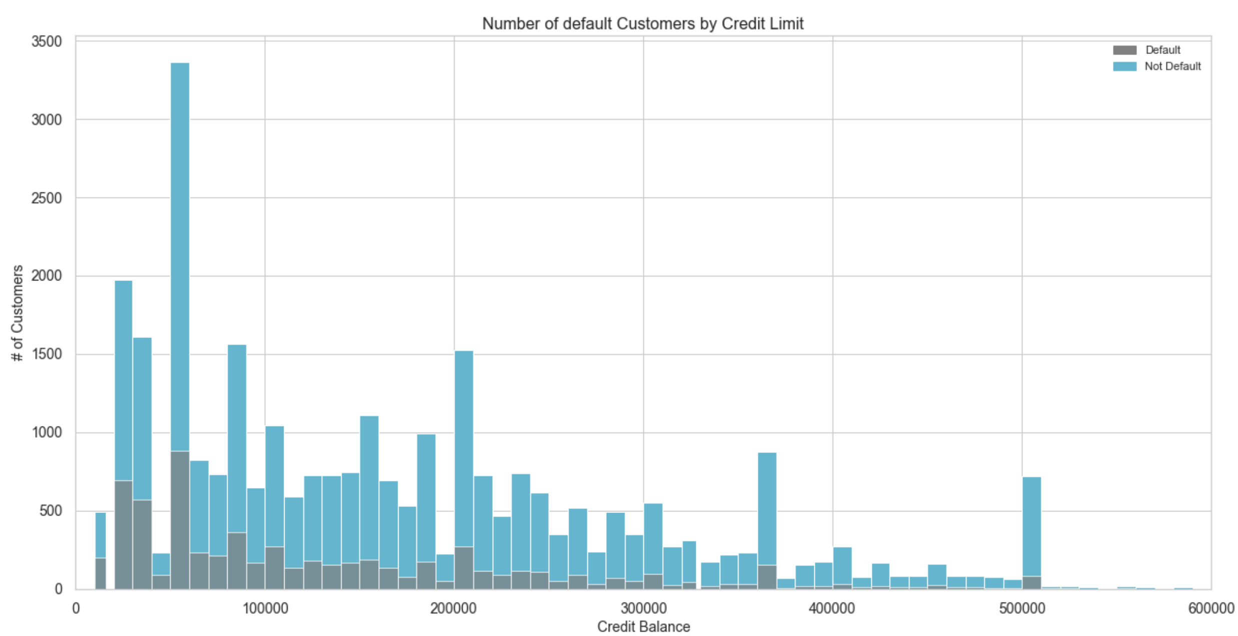
1. 20-30 year olds make up 36.7% of population
2. 30-40 year olds make up 35.7% of population
3. 70+ year olds default ***33%*** of the time compare to other groups which there is a slighted incremental decrase with every age group going working down from 70+

- 20-30: 22% default
- 30-40: 20% default
- 40-50: 23% default
- 50-60: 25% default
- 60-70: 26% default

4.5 - Default Status & Credit Limit

In [47]:
```
## Plot distribution of Credit Limit data with default, not default co
unt
sns.set(style="whitegrid")
plt.figure(figsize=(20,10))

# Plot the graph
plt.hist(credit['LIMIT_BAL'], sorted(credit['LIMIT_BAL'].unique()), co
lor='c')
plt.hist(credit['LIMIT_BAL'][(credit['Default_Status']==1)], sorted(cr
edit['LIMIT_BAL'].unique()), color='grey', alpha=0.7)


# Set X and y axis labels
plt.xlabel('Credit Balance', fontsize=14)
plt.ylabel('# of Customers', fontsize=14)
plt.xlim([0,600000])
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

# Set the legend
gray_patch = mpatches.Patch(color='gray', label='Default')
c_patch=mpatches.Patch(color='c', label='Not Default')
plt.legend(handles=[gray_patch,c_patch],loc=1)

# Set the title
plt.title ("Number of default Customers by Credit Limit", fontsize=16)
```

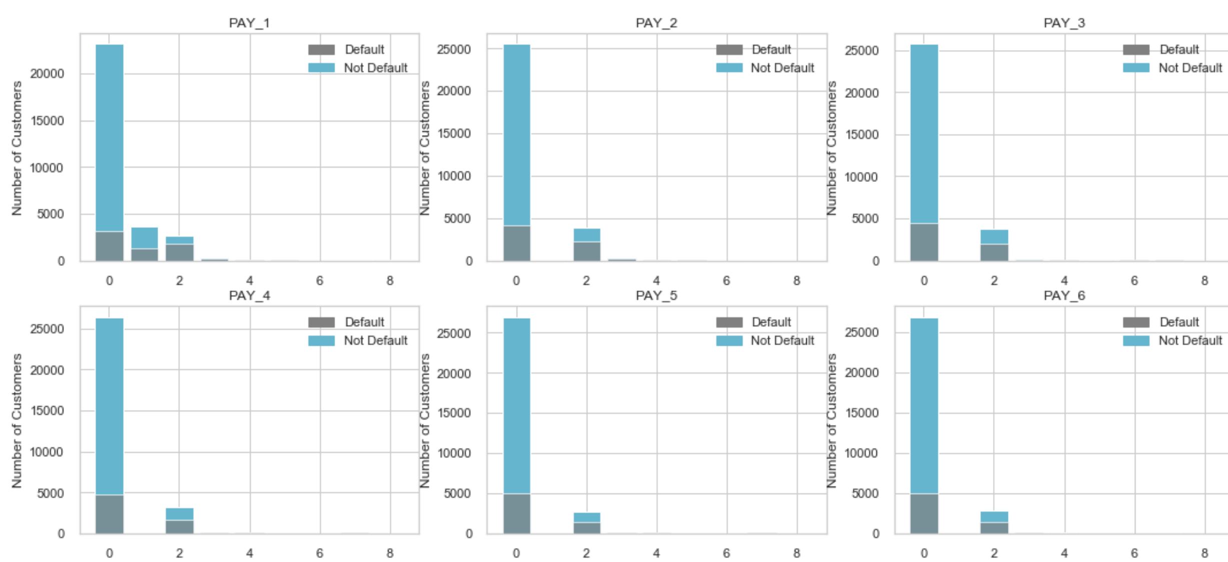
Out[47]:
```
Text(0.5, 1.0, 'Number of default Customers by Credit Limit')
```



4.6 - Default Status & Payment

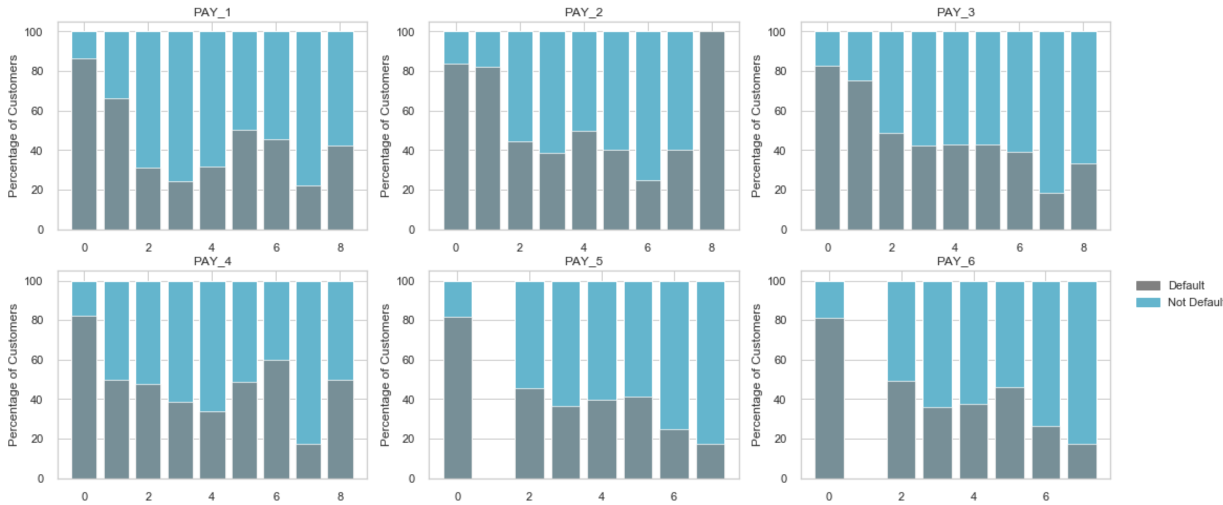
In [48]:

```
# List of all 6 columns
pay_status_columns =['PAY_1','PAY_2','PAY_3','PAY_4','PAY_5','PAY_6']

# Set up subplots
figure, ax = plt.subplots(2,3)
figure.set_size_inches(18,8)

# Get each column and plot in subplots (0,0),(0,1),(0,2) first and then (1,0) (1,1) (1,2) using (i/3) and (i%3)
for i in range(len(pay_status_columns)):
    row,col = int(i/3), i%3

    d  = credit[pay_status_columns[i]].value_counts()
    x = credit[pay_status_columns[i]][(credit['Default_Status']==1)].value_counts()
    ax[row,col].bar(d.index, d, align='center', color='c')
    ax[row,col].bar(x.index, x, align='center', color='gray', alpha=0.7)

    ax[row,col].set_ylabel("Number of Customers")
    ax[row,col].set_title(pay_status_columns[i])

    # Set the legend
    gray_patch = mpatches.Patch(color='gray', label='Default')
    c_patch=mpatches.Patch(color='c', label='Not Default')
    ax[row,col].legend(handles=[gray_patch,c_patch],loc=1)
```



In [49]:
```
# List of all 6 columns
pay_status_columns =['PAY_1','PAY_2','PAY_3','PAY_4','PAY_5','PAY_6']

# Set up subplots
figure, ax = plt.subplots(2,3)
figure.set_size_inches(18,8)

# Get each column and plot in subplots (0,0),(0,1),(0,2) first and the
n (1,0) (1,1) (1,2) using (i/3) and (i%3)
for i in range(len(pay_status_columns)):
    row,col = int(i/3), i%3


    filter = credit[pay_status_columns[i]][(credit['Default_Status']==
0)].unique()
    x= credit[pay_status_columns[i]][(credit['Default_Status']==0)].va
lue_counts()
    d  = credit[pay_status_columns[i]][credit[pay_status_columns[i]].i
sin (filter)].value_counts()
    percent=x/d*100

    ax[row,col].bar(d.index, 100, align='center', color='c')
    ax[row,col].bar(percent.index, percent, align='center', color='gra
y',alpha=0.7)

    # Set X and Y axis labels, title
    ax[row,col].set_ylabel("Percentage of Customers")
    ax[row,col].set_title(pay_status_columns[i])

    # Set the legend
gray_patch = mpatches.Patch(color='gray', label='Default')
c_patch=mpatches.Patch(color='c', label='Not Default')
plt.legend(handles=[gray_patch,c_patch],bbox_to_anchor=(1.05, 1))
```

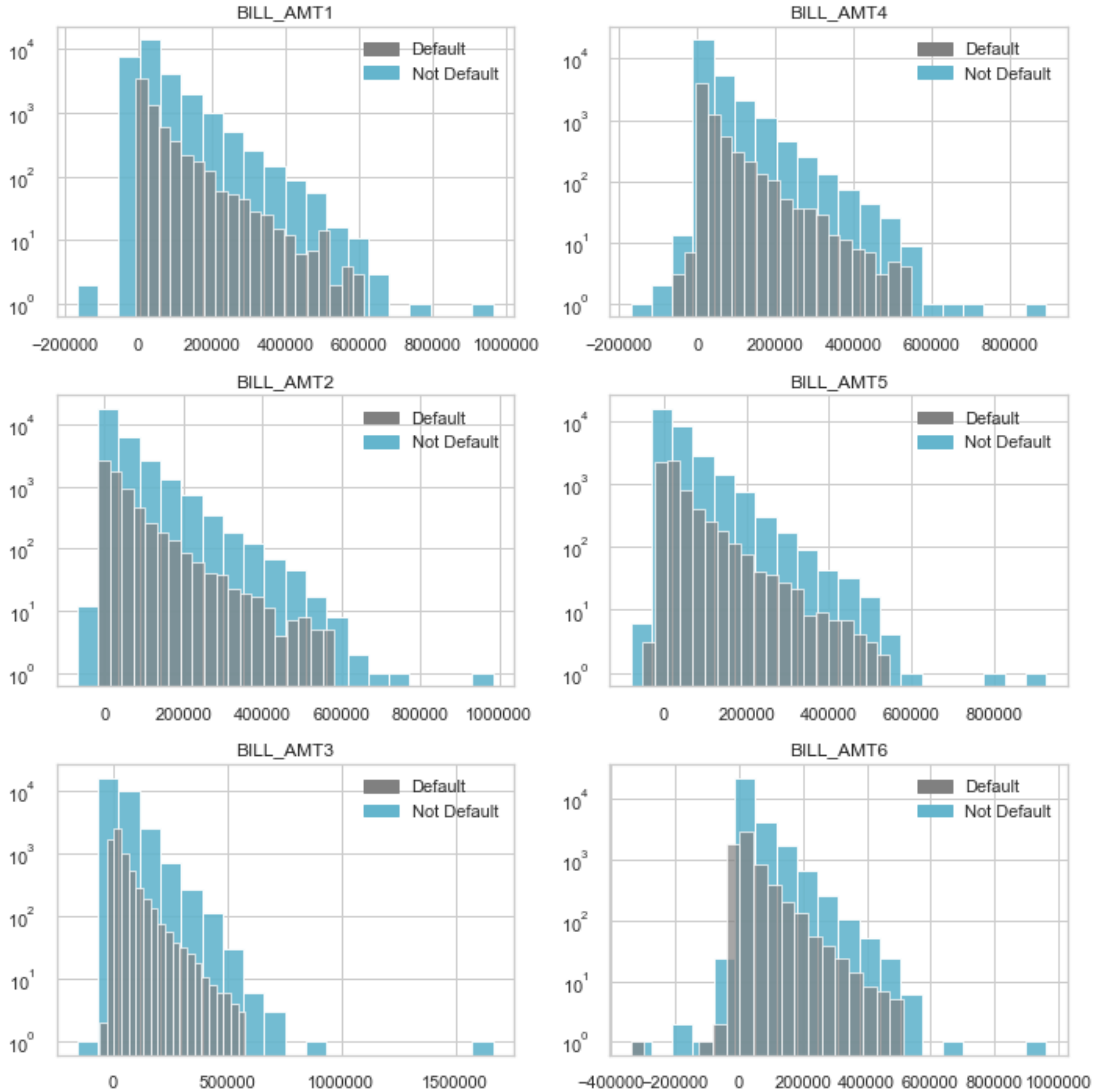
Out[49]: <matplotlib.legend.Legend at 0x11d34f730>



4.7 - Default Status & Bill Amount

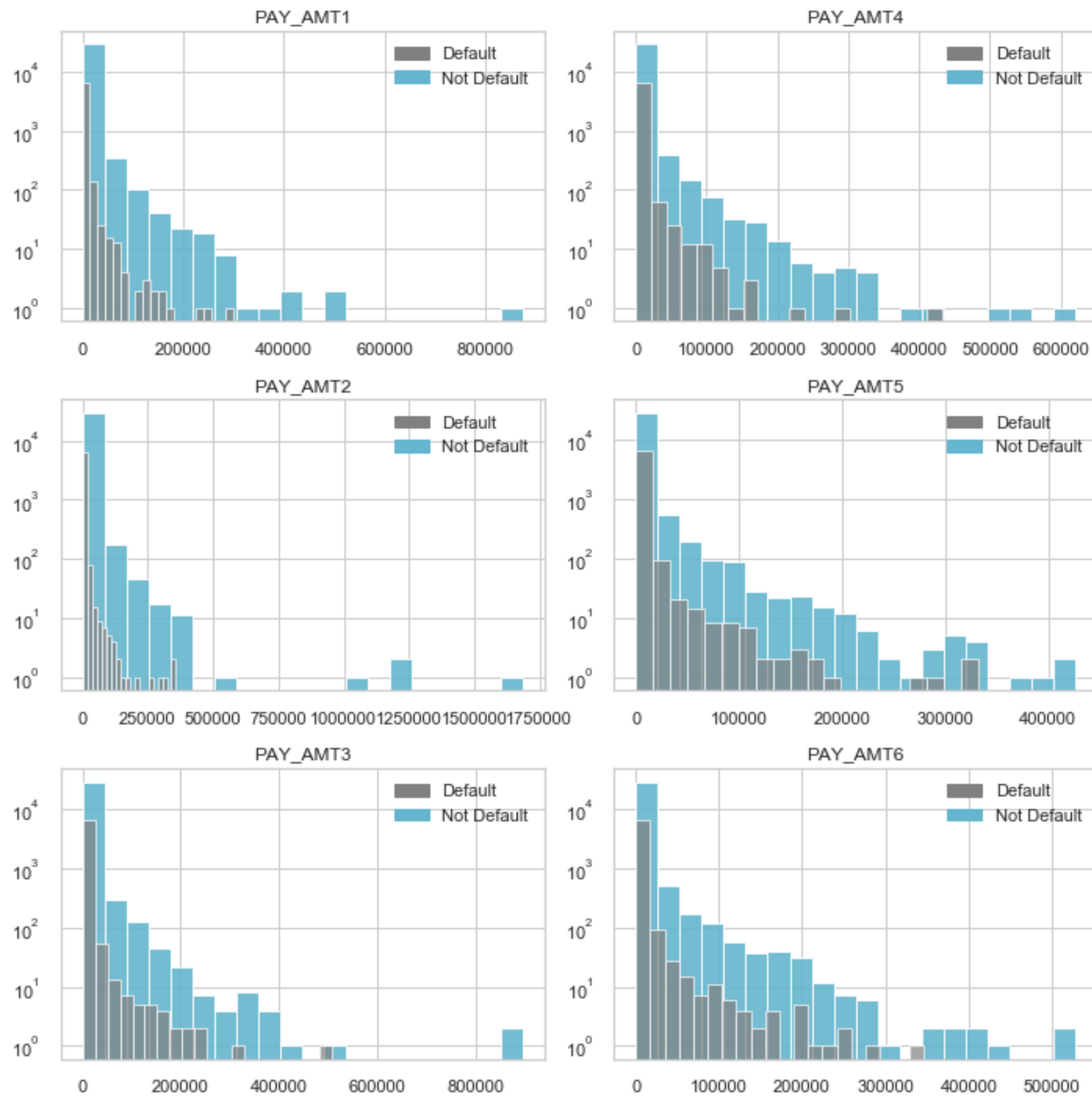
In [50]:

```
# List of all 6 columns
bill_amt_columns =['BILL_AMT1','BILL_AMT2','BILL_AMT3','BILL_AMT4','BI
LL_AMT5','BILL_AMT6']

# Set up subplots
figure, ax = plt.subplots(3,2)
figure.set_size_inches(10,10)

# Get each column and plot in subplots (0,0),(0,1),(0,2) first and the
n (1,0) (1,1) (1,2) using (i/3) and (i%3)

for i in range(len(bill_amt_columns)):
    row,col =  i%3, int(i/3)

    ax[row,col].hist(credit[bill_amt_columns[i]], bins=20, color ='c',
alpha=.9)
    ax[row,col].hist(credit[bill_amt_columns[i]][(credit['Default_Stat
us']==1)],bins=20,color='gray',alpha = 0.7)
    ax[row,col].set_title(bill_amt_columns[i])
    #adding scaling to make the graph more helpful
    ax[row,col].set_yscale('log', nonposy='clip')

    # Set the legend
    gray_patch = mpatches.Patch(color='gray', label='Default')
    c_patch=mpatches.Patch(color='c', label='Not Default')
    ax[row,col].legend(handles=[gray_patch,c_patch],loc=1)

plt.tight_layout()
```



4.8 - Default Status & Pay Amount

In [51]:
```
# List of all 6 columns
pay_amt_columns =['PAY_AMT1','PAY_AMT2','PAY_AMT3','PAY_AMT4','PAY_AMT
5','PAY_AMT6']

# Set up subplots
figure, ax = plt.subplots(3,2)
figure.set_size_inches(10,10)

# Get each column and plot in subplots (0,0),(0,1),(0,2) first and the
n (1,0) (1,1) (1,2) using (i/3) and (i%3)

for i in range(len(pay_amt_columns)):
    row,col =  i%3, int(i/3)

    ax[row,col].hist(credit[pay_amt_columns[i]], bins=20, color ='c',
alpha=.9)
    ax[row,col].hist(credit[pay_amt_columns[i]][(credit['Default_Statu
s']==1)],bins=20,color='gray',alpha = 0.7)
    ax[row,col].set_title(pay_amt_columns[i])
    #adding scaling to make the graph more helpful
    ax[row,col].set_yscale('log', nonposy='clip')

    # Set the legend
    gray_patch = mpatches.Patch(color='gray', label='Default')
    c_patch=mpatches.Patch(color='c', label='Not Default')
    ax[row,col].legend(handles=[gray_patch,c_patch],loc=1)

plt.tight_layout()
```



5.0 - Multivariate Analysis of Defaulting or Not

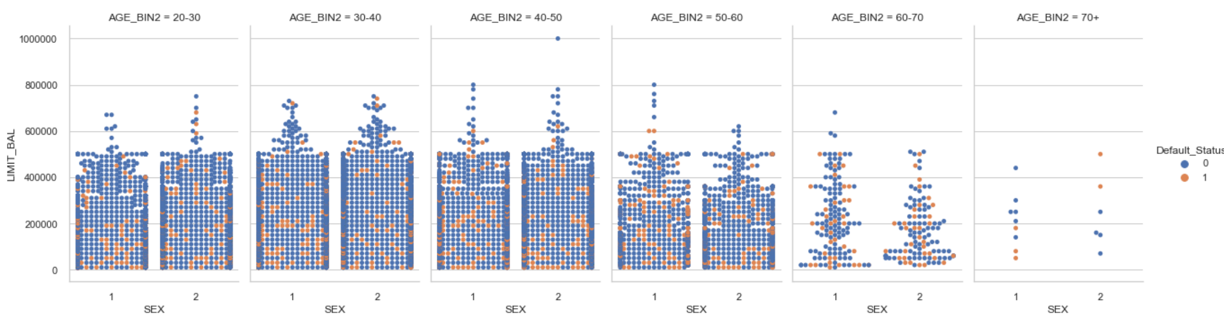
Explore and analyze data information cross multiple features using Default Status as constant

1. Default Status -vs- Gender by Age
2. Default Status -vs- Gender by Education
3. Default Status -vs- Gender by Marriage Status

5.1 - Default Status by Gender by Age

In [52]:
```
#Cat Plot: credit balance & default status distributed by Gender and A
ge
plt.figure(figsize=(15,15))
sns.catplot(x="SEX", y="LIMIT_BAL", hue="Default_Status", kind="swarm"
,
               col="AGE_BIN2", aspect=.6, data=credit);
```

```
<Figure size 1080x1080 with 0 Axes>
```



In [53]:
```
table_gender_age = pd.crosstab(index=[credit.Default_Status,credit.SEX
], columns=[credit.AGE_BIN2])
table_gender_age.unstack()
```

Out[53]:

AGE_BIN2	20-30		30-40		40-50		50-60		60-70		70+	
SEX	1	2	1	2	1	2	1	2	1	2	1	2
Default_Status												
0	2922	5605	3347	5167	1939	2663	686	807	105	84	6	4
1	912	1558	1013	1175	645	750	261	243	35	33	3	2

OBSERVATION:

1. Female has highest credit limit balance associated with it in age group 40-50
2. High % of ***NOT DEFAULTING*** are females within the 30-40 age range at 81%
3. both Males and Females over 70 have a 33% chance of defaulting

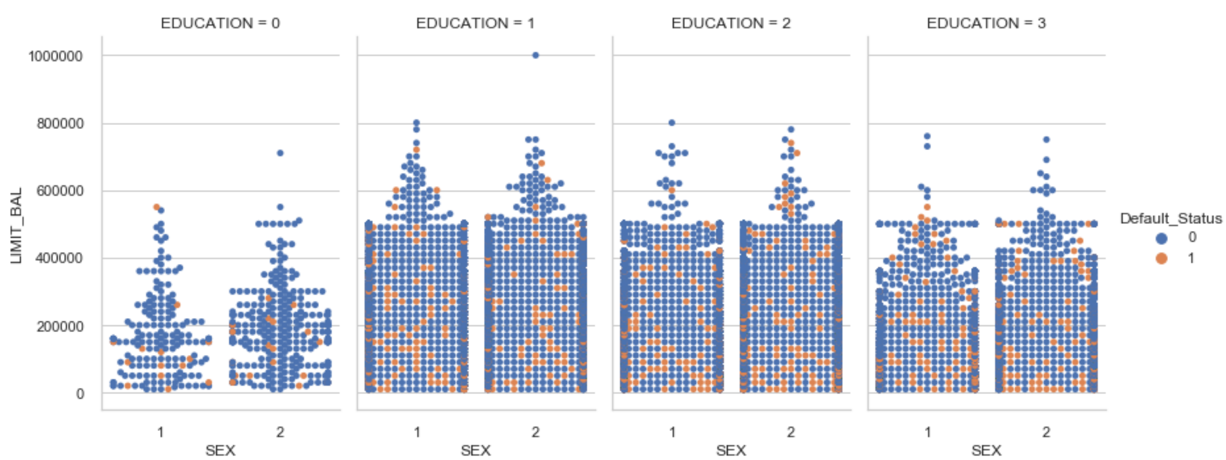
ADDITIONAL:

1. 20-30 - 65% Female population with a 22% Defaulted -vs- 35% Male population w/ 24% Defaulted
2. 30-40 - 59% Females pop. w/ 19% Defaulted -vs- 41% Male pol w/ 23% Defaulted
3. 40-50 - 57% Female pop. w/ 22% Defaulted -vs- 43% Male w/ 25% Defaulted
4. 50-60 - 53% Female pop. w/ 23% Defaulted -vs- 44% Male w/ 28% Defaulted
5. 60-70 - 46% Female pop. w/ 28% Defaulted -vs- 54% Male w/ 25% Defaulted
6. 70+ - 40% Female pop. w/ 33% Defaulted -vs- 60% Male w/ 33% Defaulted

Take Away: Females have better record for not defaulting across the age group

Default Status by Gender and Education

In [55]:
```
#Cat Plot: Education by Gender Categories by Default Status
sns.catplot(x="SEX", y="LIMIT_BAL", hue="Default_Status",
                col="EDUCATION", aspect=.6, kind="swarm", data=credit);
```



In [56]:
```
table_gender_edu = pd.crosstab(index=[credit.Default_Status,credit.SEX
], columns=[credit.EDUCATION])
table_gender_edu.unstack()
```

Out[56]:

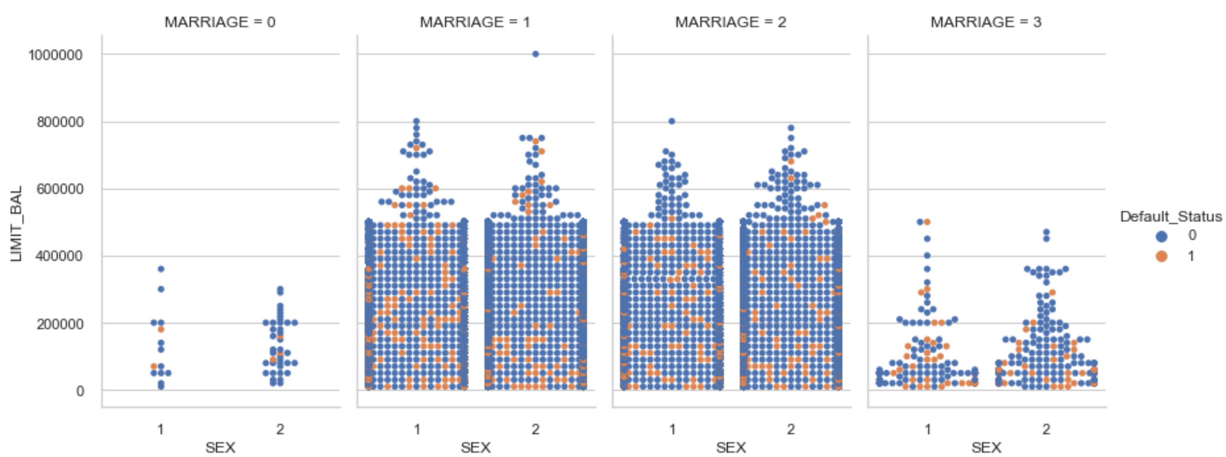
EDUCATION	0		1		2		3	
SEX	1	2	1	2	1	2	1	2
Default_Status								
0	156	279	3442	5089	3962	6729	1445	2233
1	14	19	904	1128	1406	1922	545	692

OBSERVATION

1. female continue to show better percentate at not defaulting when compare to males across education level

Default Status by Gender and Marriage

In [58]:
```
#Cat Plot: credit balance distributed by Marriage and Sex
sns.catplot(x="SEX", y="LIMIT_BAL", hue="Default_Status", kind="swarm"
,
            col="MARRIAGE", aspect=.6, data=credit);
```



OBSERVATION:

1. Distribution of credit limit is fairly symmetric

In [59]:
```
table_age = pd.crosstab(index=[credit.Default_Status,credit.SEX], colu
mns=[credit.MARRIAGE])
table_age.unstack()
```

Out[59]:

MARRIAGE	**0**		**1**		**2**		**3**	
SEX	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
Default_Status								
0	12	37	3841	6601	5061	7544	91	148
1	2	3	1343	1858	1484	1856	40	44

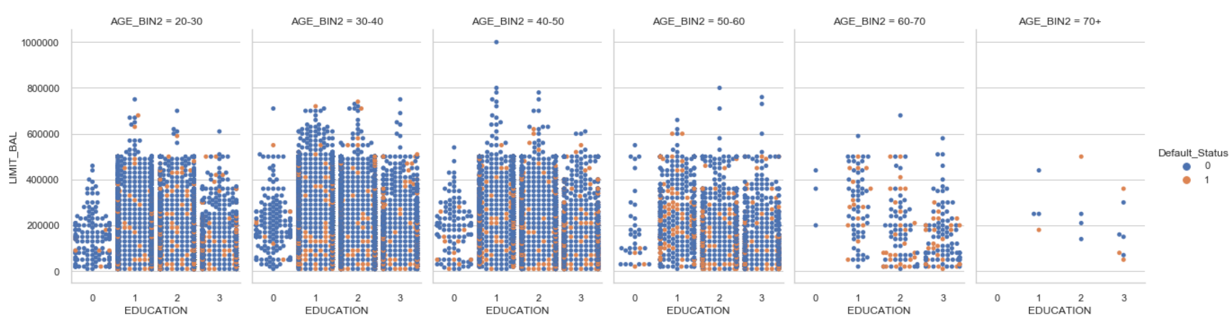
OBSERVATION: Single Males & Females did not default as much compare to Married counterparts.

- Not Defaulting %: Single Female (80%) -vs- Married Female (78%)
- Not Defaulting %: Single Male (77%) -vs- Married Male (74%)

Divorce females and males had highest default percentages at 23% (female) and 31% (males)

Default Status by Age and Education

In [61]:
```
#Cat Plot: Education by Age Categories by Default Status
sns.catplot(x="EDUCATION", y="LIMIT_BAL", hue="Default_Status",
            col="AGE_BIN2", aspect=.6,
            kind="swarm", data=credit);
```



OBSERVATION:

1. distribution of credit limit among education categories by age seem consistent
2. distribution of default status seems equally distributed

In [62]:
```
table_age = pd.crosstab(index=[credit.Default_Status,credit.EDUCATION]
, columns=[credit.AGE_BIN2])
table_age.unstack()
```

Out[62]:

AGE_BIN2	20-30				30-40				40-50		...	50-60		60-70	
EDUCATION	0	1	2	3	0	1	2	3	0	1	...	2	3	0	1
Default_Status															
0	151	3496	4093	787	156	3344	3945	1069	95	1275	...	518	584	3	52
1	9	799	1369	293	8	782	1087	311	12	323	...	192	196	0	15

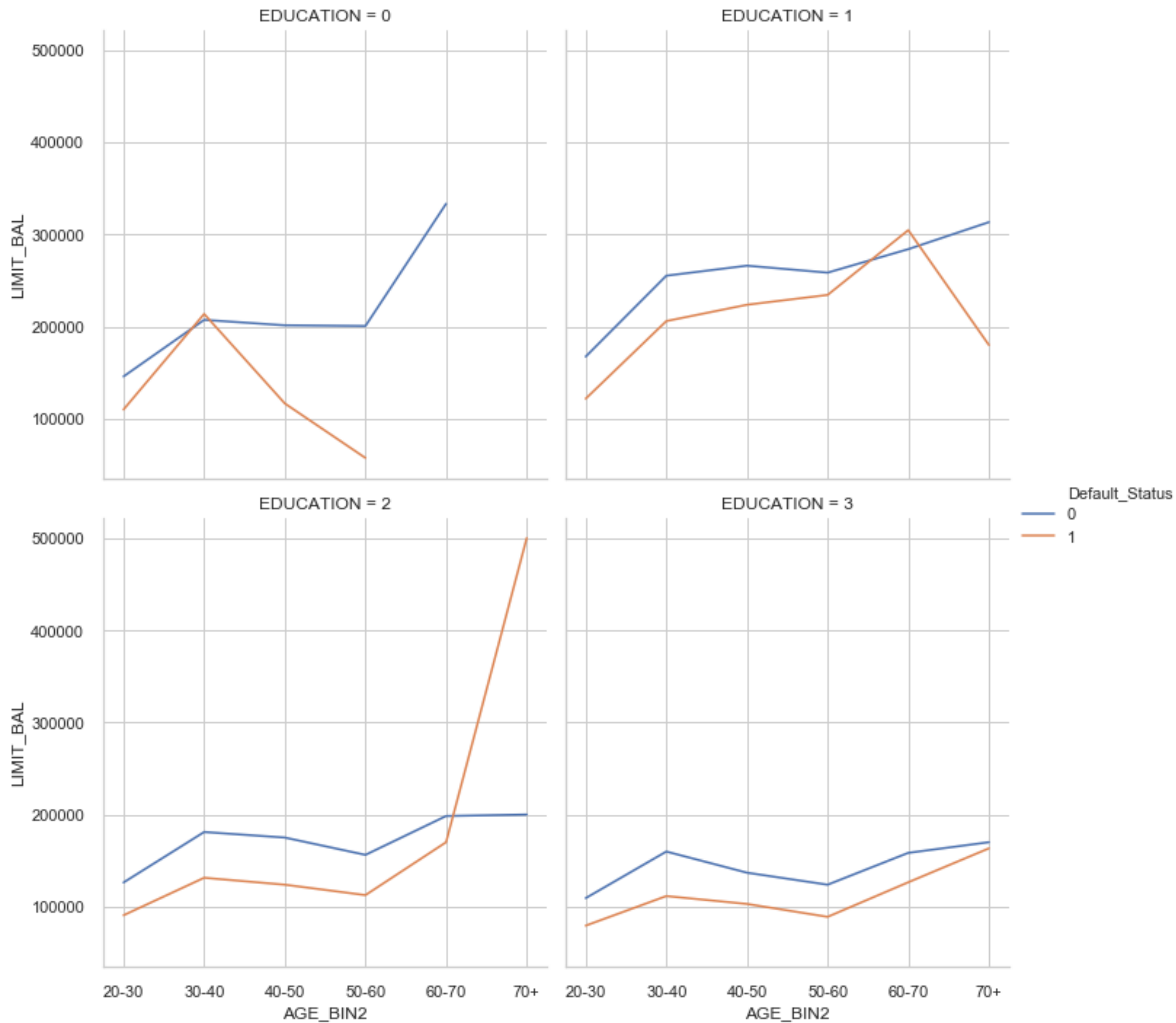
2 rows × 24 columns

Observation:

1. Millennials (27% of pop) - 24% Defaulted; 76% Not Defaulted
2. Adults (26% of pop) - 19% Defaulted; 81% Not Defaulted
3. Early Boomers (23% of pop) - 21% Defaulted; 79% Not Defaulted
4. Late Boomers (25% of pop) - 24% Defaulted; 76% Not Defaults

In [63]:
```
sns.relplot(x="AGE_BIN2", y="LIMIT_BAL", hue="Default_Status",
                col="EDUCATION",ci=None, col_wrap=2, data=credit,kind=
'line')
```

Out[63]: <seaborn.axisgrid.FacetGrid at 0x1189fbee0>



OBSERVATION

1. High School clients typically have lower defaults
2. Grad school clients have higher defaults percentage with higher credit balances

Features Transformation

In [64]:
```
#Label Encode SEX & Default Status, 2 class variable

le = LabelEncoder()
le.fit(credit['SEX'])
credit['SEX']=le.transform(credit['SEX'])
```

In [65]:
```
le = LabelEncoder()
le.fit(credit['Default_Status'])
credit['Default_Status']=le.transform(credit['Default_Status'])
```

In [66]:
```
#ONE HOT ENCODE EDUCATION
#Create 4 new Education columns
credit=pd.get_dummies(credit, prefix=['EDUCATION'], columns=['EDUCATIO
N'])
```

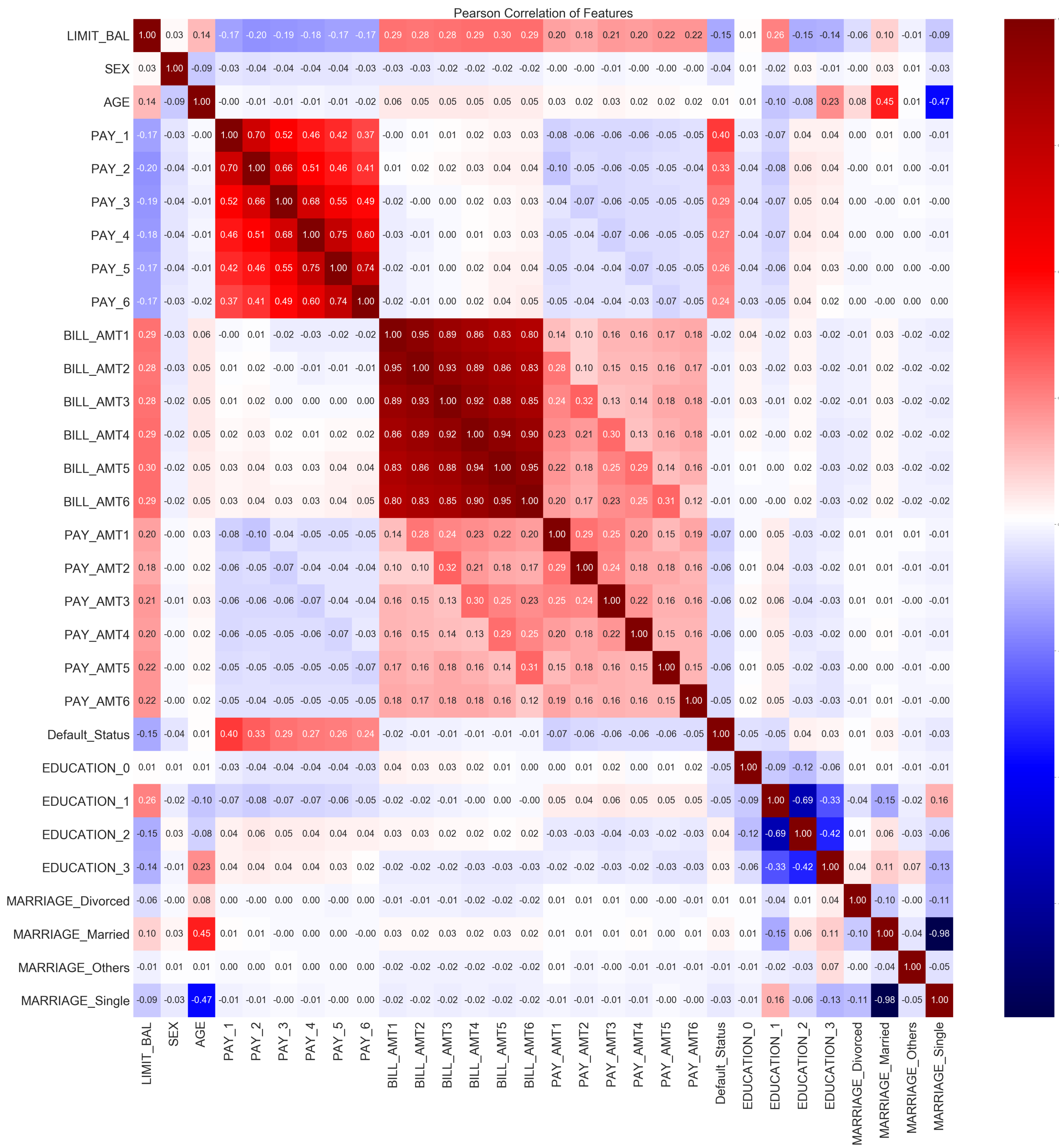
In [67]:
```
#ONE HOT ENCODE MARRIAGE feature
credit['MARRIAGE'].replace({1: "Married",
                            2: "Single",
                            3: "Divorced",
                            0: "Others"}, inplace=True)
credit=pd.get_dummies(credit, prefix=['MARRIAGE'], columns=['MARRIAGE'
])
```

OBSERVATION/EXPECTATION:

1. Marriage split into 4 units Marriage_Married, Marriage_Single,...
2. Education split into 4 units - Education_grad school, Education_university, ...

CORRELATION

In [68]:
```
## Visualize the Correlation matrix

# Configure x and y axis
corrMat=credit.corr()
sns.set()
plt.figure(figsize=(70,70))
plt.xticks(fontsize=48)
plt.yticks(fontsize=48)
plt.title('Pearson Correlation of Features', size =48)

# Plotting the data using heatmap
g = sns.heatmap(corrMat,annot=True,cmap="seismic",annot_kws={"size": 3
5},fmt=".2f")
```



OBSERVATION:

1. Payment Status (PAY0-Pay6) is highly correlated to each other
2. Bill Amount (Bill_Amt1-Bill_Amt6) is highly correlated to each other
3. Default Status have strong correlation to Payment status
4. Pay1-6, Default Status, Education_1-3, and Marriage_Single have low or negative correlation to LIMIT_Bal (target var)

In [69]:
```
#Covariance
#Covariance is often used to gauge the linear degree of change between two variables. This is important when studying
#the impact features might have on default rates
covMat=credit.cov()
print(covMat)
```

```
                LIMIT_BAL          SEX           AGE        
PAY_1  \
LIMIT_BAL     1.683769e+10  1587.661442  173039.339301 -16877.8
73897
SEX           1.587661e+03     0.239246      -0.410621     -0.0
12879
AGE           1.730393e+05    -0.410621      84.998429     -0.0
09829
PAY_1        -1.687787e+04    -0.012879      -0.009829      0.5
78744
PAY_2        -2.050707e+04    -0.017530      -0.064945      0.4
26370
PAY_3        -1.961575e+04    -0.015882      -0.103301      0.3
11113
PAY_4        -1.782781e+04    -0.014731      -0.050299      0.2
66606
PAY_5        -1.582440e+04    -0.013464      -0.086958      0.2
31846
PAY_6        -1.556707e+04    -0.011153      -0.110926      0.2
03506
BILL_AMT1     2.732380e+09 -1213.254869   38091.519524    -27.9
82760
BILL_AMT2     2.575221e+09 -1086.999787   35534.308472    512.4
19937
BILL_AMT3     2.553507e+09  -834.307489   34258.762123    719.1
89774
BILL_AMT4     2.458630e+09  -689.404217   30382.357043   1097.7
85070
BILL_AMT5     2.335917e+09  -506.382574   27587.344198   1436.8
43369
BILL_AMT6     2.248110e+09  -488.090781   26072.712836   1383.1
80778
PAY_AMT1      4.202977e+08    -1.946093    3982.705240   -996.6
06751
PAY_AMT2      5.342208e+08   -15.674467    4617.629754  -1001.2
71004
PAY_AMT3      4.808967e+08   -74.098910    4739.452137   -837.7
00166
PAY_AMT4      4.138037e+08   -17.075936    3079.001455   -736.5
45768
PAY_AMT5      4.312685e+08   -12.449121    3209.948117   -624.0
06623
```

```
PAY_AMT6          5.073244e+08   -24.057555    3182.740556   -657.8
49513
Default_Status   -8.288039e+03    -0.008069       0.052121      0.1
25116
EDUCATION_0       2.166981e+02     0.000516       0.010233     -0.0
02396
EDUCATION_1       1.602623e+04    -0.005349      -0.442245     -0.0
24924
EDUCATION_2      -9.535892e+03     0.006248      -0.357724      0.0
15465
EDUCATION_3      -6.707041e+03    -0.001414       0.789735      0.0
11855
MARRIAGE_Divorced -7.476895e+02   -0.000100       0.078807      0.0
00330
MARRIAGE_Married  6.697394e+03     0.007416       2.062329      0.0
04953
MARRIAGE_Others  -6.213684e+01     0.000247       0.004193      0.0
00025
MARRIAGE_Single  -5.887568e+03    -0.007562      -2.145330     -0.0
05308

                         PAY_2         PAY_3         PAY_4
PAY_5  \
LIMIT_BAL        -20507.068684 -19615.745616 -17827.810601 -15824.4
01123
SEX                  -0.017530     -0.015882     -0.014731     -0.0
13464
AGE                  -0.064945     -0.103301     -0.050299     -0.0
86958
PAY_1                 0.426370      0.311113      0.266606      0.2
31846
PAY_2                 0.643307      0.420743      0.312958      0.2
66274
PAY_3                 0.420743      0.625200      0.408458      0.3
12716
PAY_4                 0.312958      0.408458      0.579426      0.4
07136
PAY_5                 0.266274      0.312716      0.407136      0.5
15191
PAY_6                 0.233478      0.278533      0.328129      0.3
80073
BILL_AMT1           674.772832  -1211.367179  -1421.933883   -989.7
66117
BILL_AMT2           895.005705    -65.609193   -690.181086   -446.9
01100
BILL_AMT3          1262.955502    123.702252    132.972942    162.1
92177
BILL_AMT4          1663.136812    834.448475    710.895055   1110.2
55314
BILL_AMT5          1960.146349   1251.483515   1254.912344   1576.8
```

```
82369
BILL_AMT6           2006.289685   1357.222101   1497.048682   1818.6
19258
PAY_AMT1           -1297.613781   -517.188011   -688.592497   -639.7
11873
PAY_AMT2           -1015.972218  -1344.808758   -648.431501   -656.1
07906
PAY_AMT3            -847.609136   -805.275491   -994.536800   -472.1
59310
PAY_AMT4            -680.239986   -668.772926   -660.998893   -742.9
02567
PAY_AMT5            -598.840967   -615.414039   -614.691102   -569.4
26788
PAY_AMT6            -616.561188   -682.766950   -662.304969   -589.6
43850
Default_Status         0.108902      0.094165      0.084977      0.0
77681
EDUCATION_0           -0.003802     -0.004052     -0.003444     -0.0
03128
EDUCATION_1           -0.031930     -0.026972     -0.024701     -0.0
19558
EDUCATION_2            0.022732      0.018795      0.017053      0.0
14919
EDUCATION_3            0.013000      0.012229      0.011091      0.0
07766
MARRIAGE_Divorced     -0.000016      0.000358      0.000346     -0.0
00353
MARRIAGE_Married       0.003991     -0.000381      0.001747      0.0
00680
MARRIAGE_Others        0.000090      0.000253      0.000134      0.0
00034
MARRIAGE_Single       -0.004066     -0.000230     -0.002227     -0.0
00361

                          PAY_6     BILL_AMT1  ...      PAY_AMT6  \
LIMIT_BAL         -15567.065145  2.732380e+09  ...  5.073244e+08
SEX                   -0.011153 -1.213255e+03  ... -2.405755e+01
AGE                   -0.110926  3.809152e+04  ...  3.182741e+03
PAY_1                  0.203506 -2.798276e+01  ... -6.578495e+02
PAY_2                  0.233478  6.747728e+02  ... -6.165612e+02
PAY_3                  0.278533 -1.211367e+03  ... -6.827670e+02
PAY_4                  0.328129 -1.421934e+03  ... -6.623050e+02
PAY_5                  0.380073 -9.897661e+02  ... -5.896438e+02
PAY_6                  0.511916 -1.039695e+03  ... -5.782348e+02
BILL_AMT1          -1039.694681  5.425520e+09  ...  2.347356e+08
BILL_AMT2           -505.744101  4.989564e+09  ...  2.204475e+08
BILL_AMT3             77.628559  4.559032e+09  ...  2.247628e+08
BILL_AMT4            943.301698  4.077469e+09  ...  2.031373e+08
BILL_AMT5           1814.164614  3.716733e+09  ...  1.774197e+08
BILL_AMT6           1957.451127  3.521673e+09  ...  1.221863e+08
```

```
PAY_AMT1            -567.859620  1.709557e+08  ...  5.472063e+07
PAY_AMT2            -658.715050  1.684171e+08  ...  6.460844e+07
PAY_AMT3            -507.137340  2.033350e+08  ...  5.096734e+07
PAY_AMT4            -342.409537  1.825462e+08  ...  4.398026e+07
PAY_AMT5            -732.446757  1.878467e+08  ...  4.209185e+07
PAY_AMT6            -578.234782  2.347356e+08  ...  3.163765e+08
Default_Status         0.072554 -6.040984e+02  ... -3.931670e+02
EDUCATION_0           -0.003074  3.312850e+02  ...  3.521399e+01
EDUCATION_1           -0.016006 -8.305245e+02  ...  4.280937e+02
EDUCATION_2            0.013781  1.106182e+03  ... -2.346051e+02
EDUCATION_3            0.005299 -6.069426e+02  ... -2.287026e+02
MARRIAGE_Divorced      0.000126 -8.739021e+01  ... -1.953814e+01
MARRIAGE_Married      -0.000762  9.276526e+02  ...  5.292396e+01
MARRIAGE_Others        0.000092 -5.565510e+01  ... -5.376048e+00
MARRIAGE_Single        0.000544 -7.846073e+02  ... -2.800977e+01

                   Default_Status  EDUCATION_0   EDUCATION_1  EDUCAT
ION_2  \
LIMIT_BAL            -8288.039420   216.698093  16026.234853 -9535.8
92010
SEX                     -0.008069     0.000516     -0.005349      0.0
06248
AGE                      0.052121     0.010233     -0.442245     -0.3
57724
PAY_1                    0.125116    -0.002396     -0.024924      0.0
15465
PAY_2                    0.108902    -0.003802     -0.031930      0.0
22732
PAY_3                    0.094165    -0.004052     -0.026972      0.0
18795
PAY_4                    0.084977    -0.003444     -0.024701      0.0
17053
PAY_5                    0.077681    -0.003128     -0.019558      0.0
14919
PAY_6                    0.072554    -0.003074     -0.016006      0.0
13781
BILL_AMT1             -604.098379   331.285023   -830.524506  1106.1
82049
BILL_AMT2             -422.662880   270.884647   -676.485488  1012.1
22545
BILL_AMT3             -408.384846   248.250076   -419.833047   763.6
33782
BILL_AMT4             -274.056305   166.461641   -103.258529   687.9
02795
BILL_AMT5             -173.164564    88.354442     40.404079   590.7
29081
BILL_AMT6             -135.235463    24.542393    -59.387122   723.5
04466
PAY_AMT1              -502.268504     9.032523    396.551292  -274.0
64254
```

```
PAY_AMT2            -561.188409   34.208346   491.344602  -382.4
28087
PAY_AMT3            -411.839273   48.559726   473.261608  -314.1
42998
PAY_AMT4            -370.207072    2.042319   347.226400  -211.9
07625
PAY_AMT5            -350.228855   13.930756   346.753554  -163.2
16149
PAY_AMT6            -393.167028   35.213988   428.093695  -234.6
05082
Default_Status         0.172309   -0.002354    -0.010184     0.0
07548
EDUCATION_0           -0.002354    0.015375    -0.005506    -0.0
07307
EDUCATION_1           -0.010184   -0.005506     0.228255    -0.1
64926
EDUCATION_2            0.007548   -0.007307    -0.164926     0.2
48974
EDUCATION_3            0.004990   -0.002562    -0.057822    -0.0
76741
MARRIAGE_Divorced      0.000418    0.000099    -0.002131     0.0
00363
MARRIAGE_Married       0.006086    0.000698    -0.036588     0.0
15124
MARRIAGE_Others       -0.000232   -0.000028    -0.000502    -0.0
00643
MARRIAGE_Single       -0.006273   -0.000769     0.039221    -0.0
14845

                   EDUCATION_3  MARRIAGE_Divorced  MARRIAGE_Married
\
LIMIT_BAL         -6707.040936        -747.689481       6697.394397
SEX                  -0.001414          -0.000100          0.007416
AGE                   0.789735           0.078807          2.062329
PAY_1                 0.011855           0.000330          0.004953
PAY_2                 0.013000          -0.000016          0.003991
PAY_3                 0.012229           0.000358         -0.000381
PAY_4                 0.011091           0.000346          0.001747
PAY_5                 0.007766          -0.000353          0.000680
PAY_6                 0.005299           0.000126         -0.000762
BILL_AMT1          -606.942566         -87.390207        927.652636
BILL_AMT2          -606.521704         -99.038557        796.006363
BILL_AMT3          -592.050810        -106.236881        897.227909
BILL_AMT4          -751.105907        -130.250252        742.594393
BILL_AMT5          -719.487603        -130.315512        758.474633
BILL_AMT6          -688.659737        -111.399812        628.648894
PAY_AMT1           -131.519561          13.370609         57.482847
PAY_AMT2           -143.124861          20.881739        129.684938
PAY_AMT3           -207.678335           9.899963         46.837612
PAY_AMT4           -137.361093           3.121364        113.862271
```

```
PAY_AMT5            -197.468161          -2.597642          16.692618
PAY_AMT6            -228.702601         -19.538141          52.923962
Default_Status         0.004990           0.000418           0.006086
EDUCATION_0           -0.002562           0.000099           0.000698
EDUCATION_1           -0.057822          -0.002131          -0.036588
EDUCATION_2           -0.076741           0.000363           0.015124
EDUCATION_3            0.137125           0.001669           0.020765
MARRIAGE_Divorced      0.001669           0.010663          -0.004908
MARRIAGE_Married       0.020765          -0.004908           0.248010
MARRIAGE_Others        0.001173          -0.000019          -0.000821
MARRIAGE_Single       -0.023607          -0.005736          -0.242282

                   MARRIAGE_Others  MARRIAGE_Single
LIMIT_BAL               -62.136840     -5887.568076
SEX                       0.000247        -0.007562
AGE                       0.004193        -2.145330
PAY_1                     0.000025        -0.005308
PAY_2                     0.000090        -0.004066
PAY_3                     0.000253        -0.000230
PAY_4                     0.000134        -0.002227
PAY_5                     0.000034        -0.000361
PAY_6                     0.000092         0.000544
BILL_AMT1               -55.655104      -784.607326
BILL_AMT2               -45.543512      -651.424294
BILL_AMT3               -50.126334      -740.864695
BILL_AMT4               -43.803558      -568.540583
BILL_AMT5               -40.799395      -587.359727
BILL_AMT6               -39.830652      -477.418429
PAY_AMT1                  3.829617       -74.683073
PAY_AMT2                 -5.669673      -144.897003
PAY_AMT3                 -2.169343       -54.568232
PAY_AMT4                 -3.545566      -113.438069
PAY_AMT5                 -4.830478        -9.264498
PAY_AMT6                 -5.376048       -28.009773
Default_Status           -0.000232        -0.006273
EDUCATION_0              -0.000028        -0.000769
EDUCATION_1              -0.000502         0.039221
EDUCATION_2              -0.000643        -0.014845
EDUCATION_3               0.001173        -0.023607
MARRIAGE_Divorced        -0.000019        -0.005736
MARRIAGE_Married         -0.000821        -0.242282
MARRIAGE_Others           0.001799        -0.000959
MARRIAGE_Single          -0.000959         0.248977

[30 rows x 30 columns]
```

In [ ]: