In [39]:
```
import pandas as pd
```

In [40]:
```
df=pd.read_csv(r"C:\Users\SAMEEP GUPTA\OneDrive\Desktop\DAV practice files\UCI_Credit_Card.csv\UCI_Credit_Card.csv"
```

In [ ]:
```

```

In [41]:
```
df.head()
```

Out[41]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	P
0	1	20000.0	2	2	1	24	2	2	-1	-1	...	0.0	0.0	0.0	0.0	
1	2	120000.0	2	2	2	26	-1	2	0	0	...	3272.0	3455.0	3261.0	0.0	
2	3	90000.0	2	2	2	34	0	0	0	0	...	14331.0	14948.0	15549.0	1518.0	
3	4	50000.0	2	2	1	37	0	0	0	0	...	28314.0	28959.0	29547.0	2000.0	
4	5	50000.0	1	2	1	57	-1	0	-1	0	...	20940.0	19146.0	19131.0	2000.0	

5 rows × 25 columns

In [42]:
```
cols=list(df.columns)
```

In [43]:
```
data=df[cols[0:2]+cols[4:-1]]
```

In [44]:
```
data.head()
```

Out[44]:

	ID	LIMIT_BAL	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	...	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_
0	1	20000.0	1	24	2	2	-1	-1	-2	-2	...	689.0	0.0	0.0	0.0	
1	2	120000.0	2	26	-1	2	0	0	0	2	...	2682.0	3272.0	3455.0	3261.0	
2	3	90000.0	2	34	0	0	0	0	0	0	...	13559.0	14331.0	14948.0	15549.0	
3	4	50000.0	1	37	0	0	0	0	0	0	...	49291.0	28314.0	28959.0	29547.0	
4	5	50000.0	1	57	-1	0	-1	0	0	0	...	35835.0	20940.0	19146.0	19131.0	

5 rows × 22 columns

In [12]:
```
data.LIMIT_BAL.head()
```

Out[12]:
```
0     20000.0
1    120000.0
2     90000.0
3     50000.0
4     50000.0
Name: LIMIT_BAL, dtype: float64
```

In [13]:
```
marriage_labels=pd.cut(data.MARRIAGE,bins=[0,1,2,3],labels=['Married','Single','Others'])
```

In [14]:
```
marriage_labels
```

Out[14]:
```
0        Married
1         Single
2         Single
3        Married
4        Married
          ...
29995    Married
29996     Single
29997     Single
29998    Married
29999    Married
Name: MARRIAGE, Length: 30000, dtype: category
Categories (3, object): ['Married' < 'Single' < 'Others']
```

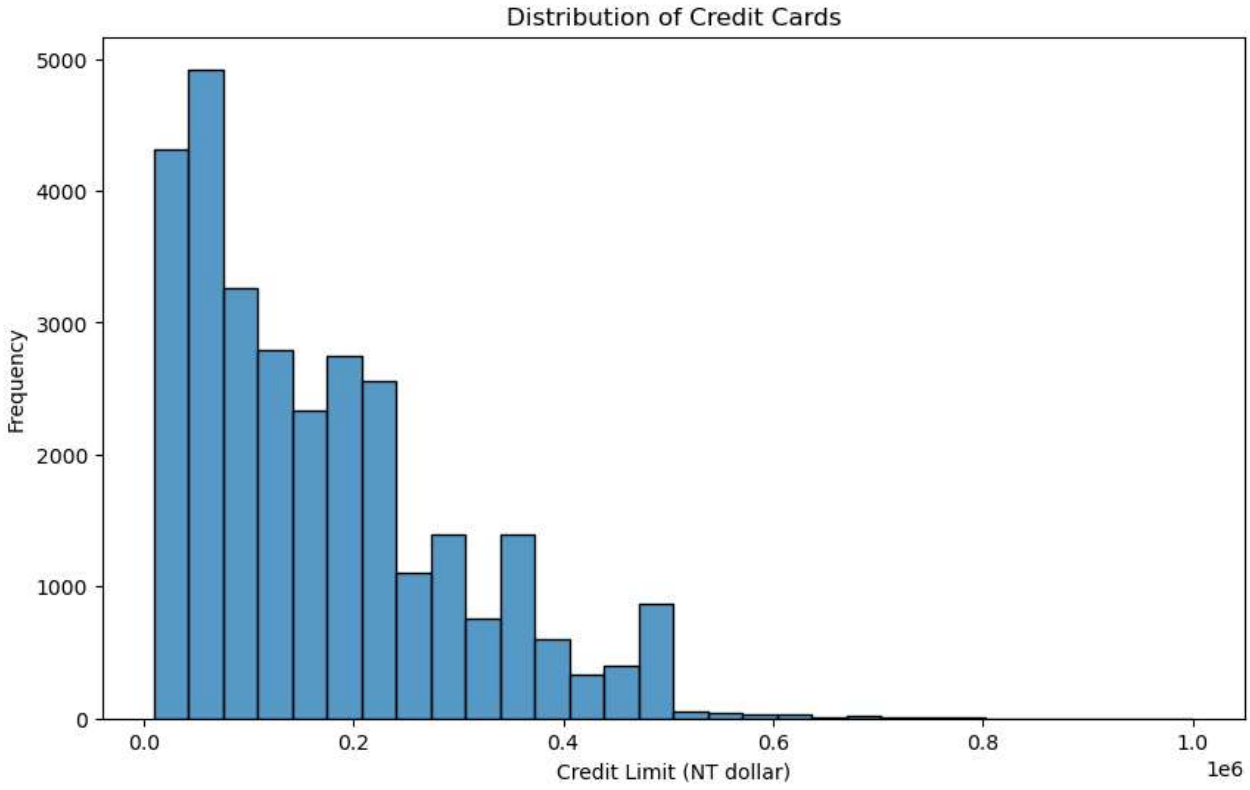
In [15]:
```
mar_labels_counts=marriage_labels.count()
```

In [16]:
```
mar_labels_counts
```

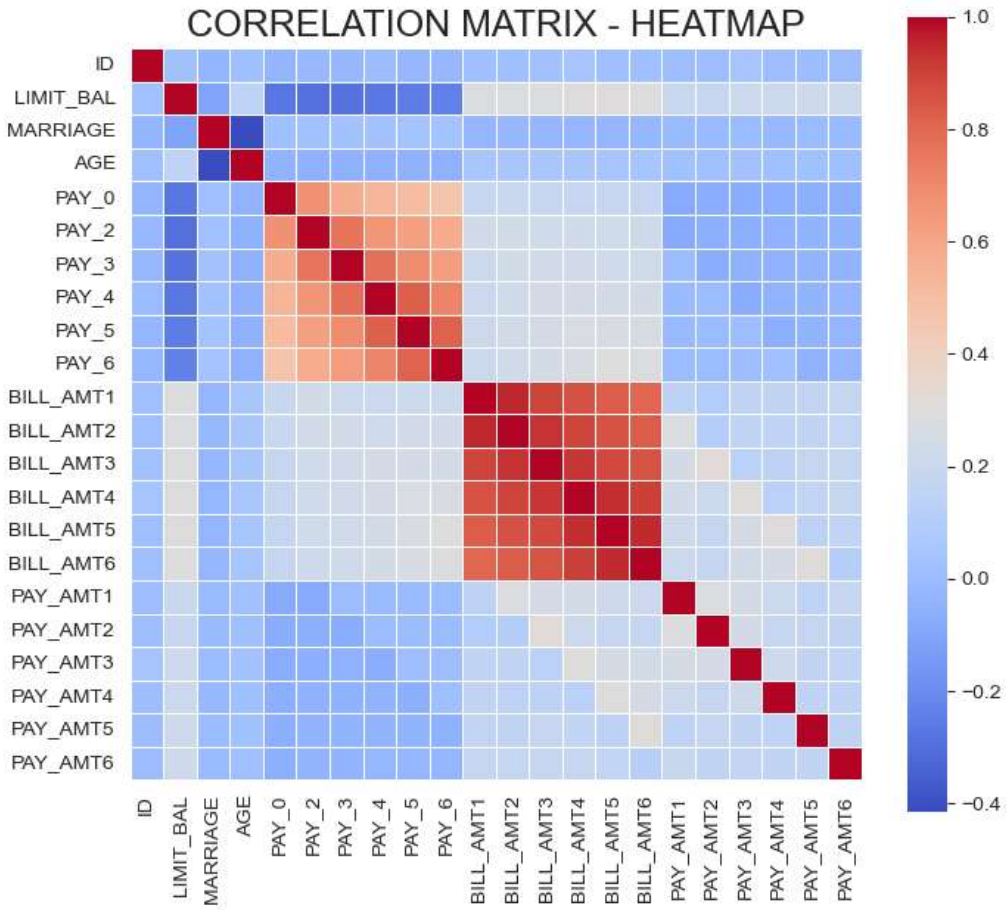
Out[16]: 29946

In [17]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:
```
plt.figure(figsize=(10,6))
sns.histplot(df['LIMIT_BAL'],bins=30)
plt.title('Distribution of Credit Cards')
plt.xlabel('Credit Limit (NT dollar)')
plt.ylabel('Frequency')
plt.show()
```

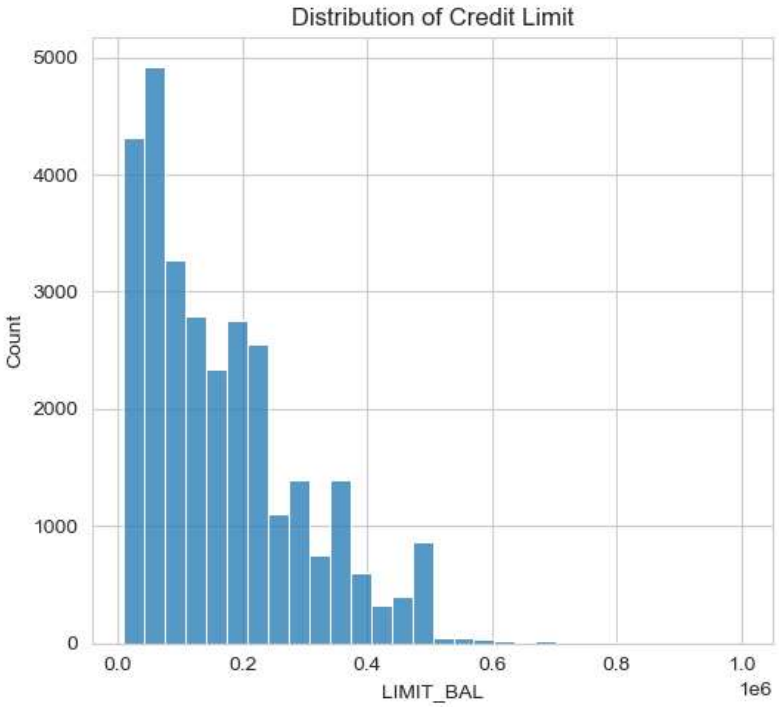


In [27]:
```
cor = data.corr() # .corr is used to find corelation
f,ax = plt.subplots(figsize=(8, 7))
sns.heatmap(cor, cbar = True,  square = True, annot = False, fmt= '.1f',
            xticklabels= True, yticklabels= True
            ,cmap="coolwarm", linewidths=.5, ax=ax)
plt.title('CORRELATION MATRIX - HEATMAP', size=18);
```

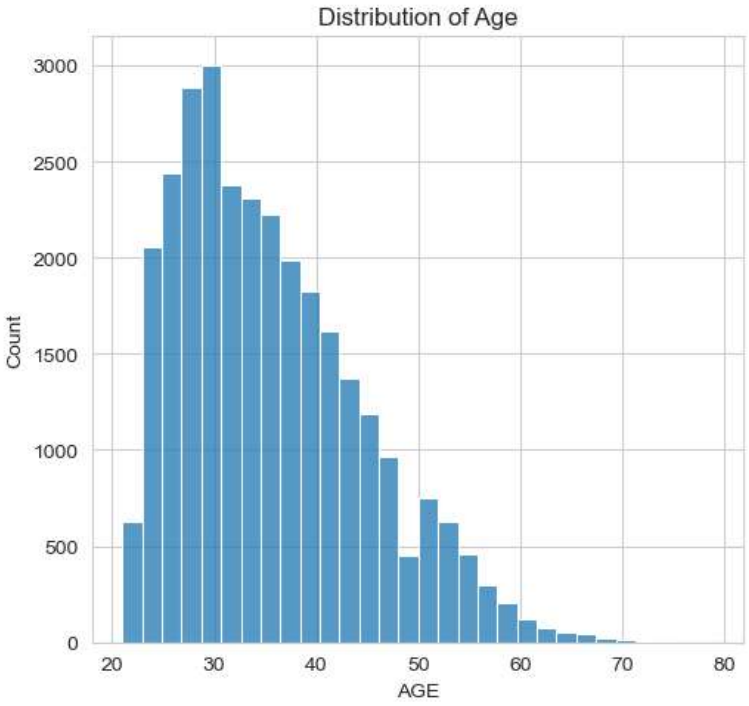


In [24]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

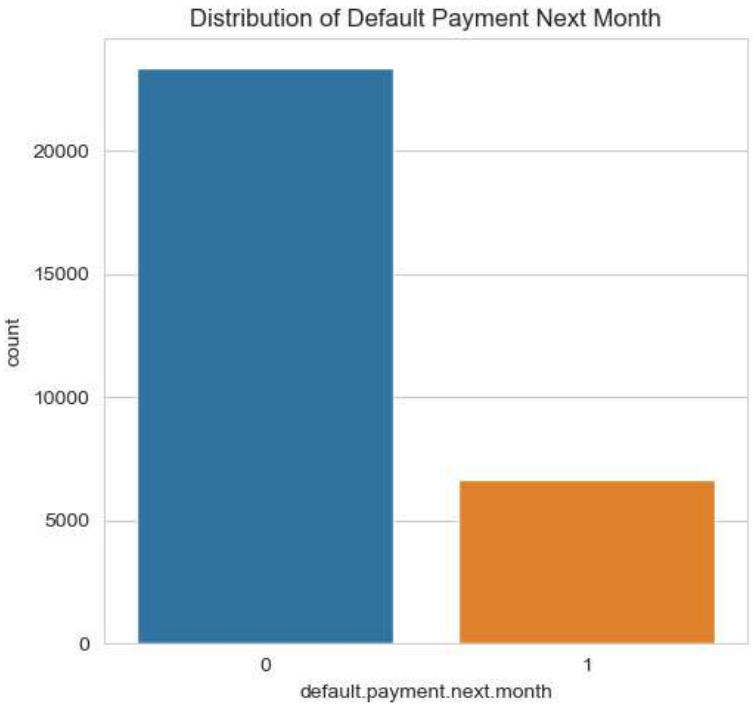
In [29]:
```
# Set the style of seaborn
sns.set_style('whitegrid')

# Plot distributions of 'LIMIT_BAL', 'AGE', and 'default.payment.next.month'
plt.figure(figsize=(15, 5))

# Distribution of 'LIMIT_BAL'
plt.subplot(1, 3, 1)
sns.histplot(data['LIMIT_BAL'], bins=30, kde=False)
plt.title('Distribution of Credit Limit')

plt.tight_layout()
plt.show()
```



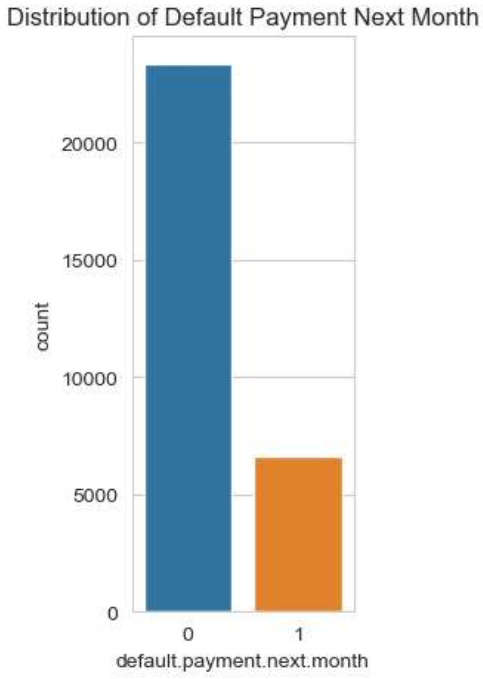
In [30]:
```
# Set the style of seaborn
sns.set_style('whitegrid')

# Plot distributions of 'LIMIT_BAL', 'AGE', and 'default.payment.next.month'
plt.figure(figsize=(15, 5))

# Distribution of 'AGE'
plt.subplot(1, 3, 2)
sns.histplot(data['AGE'], bins=30, kde=False)
plt.title('Distribution of Age')

plt.tight_layout()
plt.show()
```



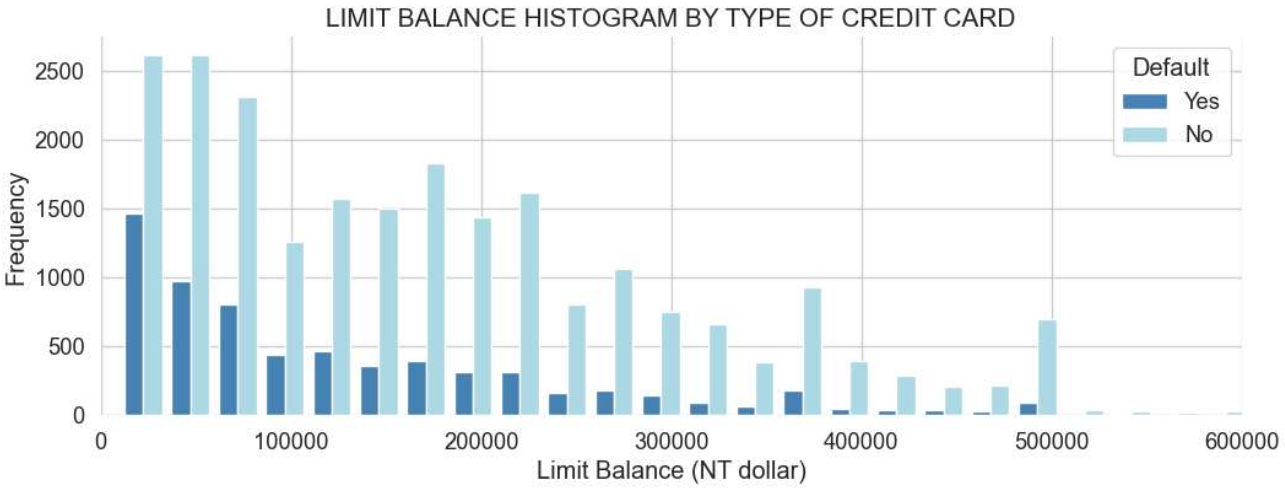
In [46]:
```
# Set the style of seaborn
sns.set_style('whitegrid')

# Plot distributions of 'LIMIT_BAL', 'AGE', and 'default.payment.next.month'
plt.figure(figsize=(15, 5))

# Distribution of 'default.payment.next.month'
plt.subplot(1, 3, 3)
sns.countplot(x=df['default.payment.next.month'])
plt.title('Distribution of Default Payment Next Month')

plt.tight_layout()
plt.show()
```



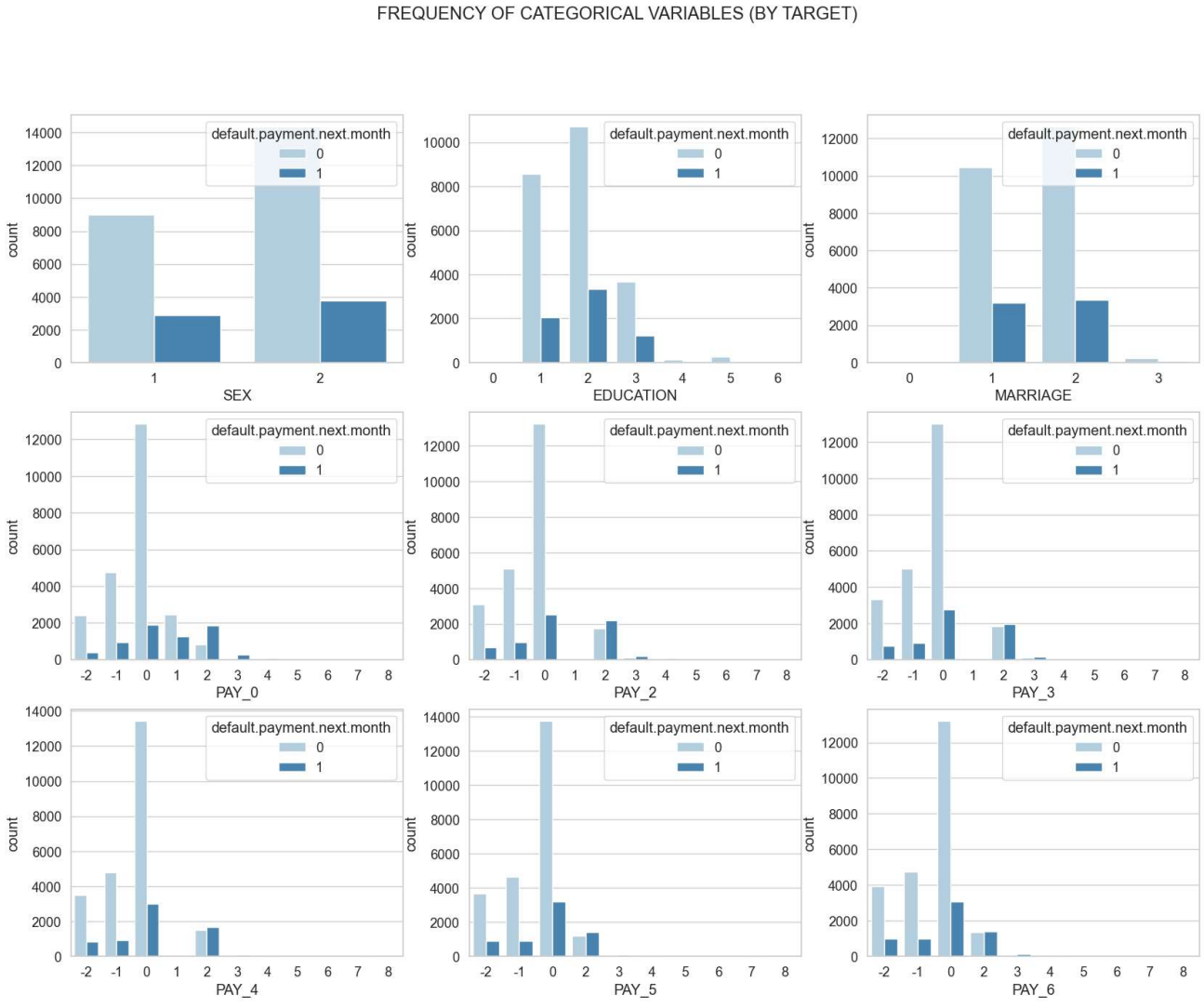
In [45]:
```
# Distribution of 'default.payment.next.month'
plt.subplot(1, 3, 3)
sns.countplot(x=df['default.payment.next.month'])
plt.title('Distribution of Default Payment Next Month')

plt.tight_layout()
plt.show()
```



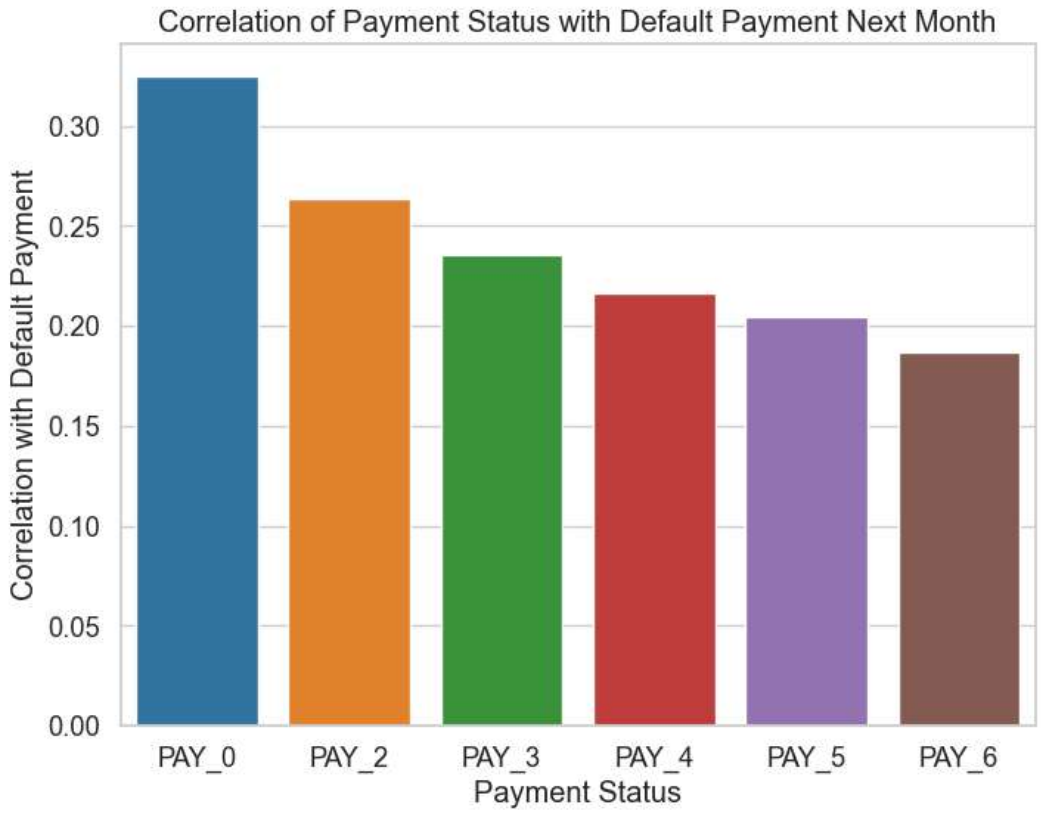
In [48]:
```
x1 = df[df['default.payment.next.month'] == 1]['LIMIT_BAL']
x2 = df[df['default.payment.next.month'] == 0]['LIMIT_BAL']

# Plotting
plt.figure(figsize=(12, 4))
sns.set_context('notebook', font_scale=1.2)
sns.set_color_codes("pastel")
plt.hist([x1, x2], bins=40, density=False, color=['steelblue', 'lightblue'])
plt.xlim([0, 600000])
plt.legend(['Yes', 'No'], title='Default', loc='upper right', facecolor='white')
plt.xlabel('Limit Balance (NT dollar)')
plt.ylabel('Frequency')
plt.title('LIMIT BALANCE HISTOGRAM BY TYPE OF CREDIT CARD', size=15)
plt.box(False)
plt.savefig('ImageName.png', format='png', dpi=200, transparent=True)
plt.show()
```



In [54]:
```
subset = df[['SEX', 'EDUCATION', 'MARRIAGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4',
             'PAY_5', 'PAY_6', 'default.payment.next.month']]

f, axes = plt.subplots(3, 3, figsize=(20, 15), facecolor='white')
f.suptitle('FREQUENCY OF CATEGORICAL VARIABLES (BY TARGET)')
ax1 = sns.countplot(x="SEX", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[0,0])
ax2 = sns.countplot(x="EDUCATION", hue='default.payment.next.month', data=subset, palette="Blues",ax=axes[0,1])
ax3 = sns.countplot(x="MARRIAGE", hue='default.payment.next.month', data=subset, palette="Blues",ax=axes[0,2])
ax4 = sns.countplot(x="PAY_0", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[1,0])
ax5 = sns.countplot(x="PAY_2", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[1,1])
ax6 = sns.countplot(x="PAY_3", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[1,2])
ax7 = sns.countplot(x="PAY_4", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[2,0])
ax8 = sns.countplot(x="PAY_5", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[2,1])
ax9 = sns.countplot(x="PAY_6", hue='default.payment.next.month', data=subset, palette="Blues", ax=axes[2,2])
```



In [56]:
```
# Calculate the correlation between PAY_0 to PAY_6 and default payment next month
pay_status_columns = ['PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6']
default_payment_next_month = 'default.payment.next.month'

# Compute the correlation matrix
pay_status_correlation = df[pay_status_columns + [default_payment_next_month]].corr()

# Extract the correlations with default payment next month
pay_status_correlation_with_default = pay_status_correlation[default_payment_next_month].drop(default_payment_next_m

# Plot the correlations
plt.figure(figsize=(8, 6))
sns.barplot(x=pay_status_correlation_with_default.index, y=pay_status_correlation_with_default.values)
plt.title('Correlation of Payment Status with Default Payment Next Month')
plt.xlabel('Payment Status')
plt.ylabel('Correlation with Default Payment')
plt.show()

# Print the correlation values
print(pay_status_correlation_with_default)
```




```
PAY_0    0.324794
PAY_2    0.263551
PAY_3    0.235253
PAY_4    0.216614
PAY_5    0.204149
PAY_6    0.186866
Name: default.payment.next.month, dtype: float64
```