**CREDIT CARD DEFAULT PREDICTION**

**GROUP – 3:**

**RISHIK ANGARA** – CB.EN.U4CSE21104

**ANWESHA BAIDYA** – CB.EN.U4CSE21105

**G GEETH VISHNU** – CB.EN.U4CSE21119

**U JASWANTH** – CB.EN.U4CSE21164

**DATASET:**

<https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset>

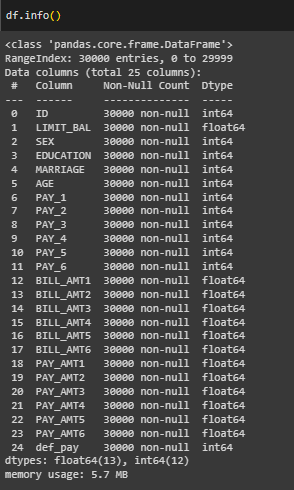
**DATASET INFO:**

* The dataset contains information on credit card usage and demographics of credit card holders in Taiwan.
* The data was collected from April to September 2005.
* The dataset contains 25 variables, including the ID of the credit card holder, the amount of credit limit, the gender, education, marital status, and age of the credit card holder.
* The target variable is the default payment in September, 2005.
* The dataset can be used to build models to predict whether a credit card holder is likely to default on their payments.

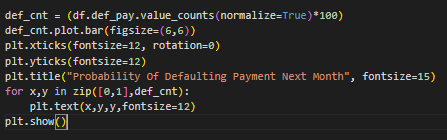
**ATRTRIBUTES:**

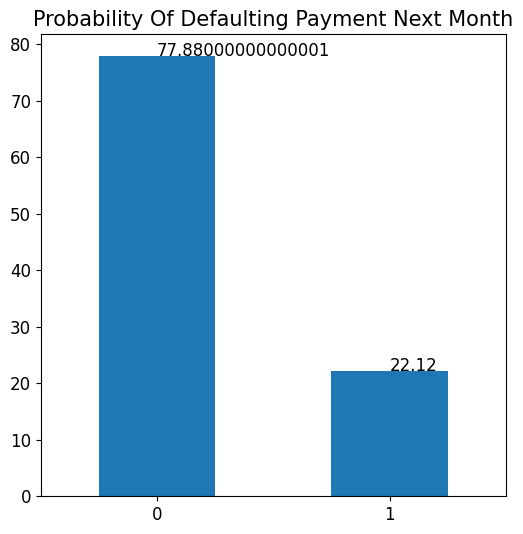
There are 25 variables:

* **ID**: ID of each client
* **LIMIT\_BAL**: Amount of given credit in NT dollars (includes individual and family/supplementary credit
* **SEX**: Gender (1=male, 2=female)
* **EDUCATION**: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
* **MARRIAGE**: Marital status (1=married, 2=single, 3=others)
* **AGE**: Age in years
* **PAY\_1**: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
* **PAY\_2**: Repayment status in August, 2005 (scale same as above)
* **PAY\_3**: Repayment status in July, 2005 (scale same as above)
* **PAY\_4**: Repayment status in June, 2005 (scale same as above)
* **PAY\_5**: Repayment status in May, 2005 (scale same as above)
* **PAY\_6**: Repayment status in April, 2005 (scale same as above)
* **BILL\_AMT1**: Amount of bill statement in September, 2005 (NT dollar)
* **BILL\_AMT2**: Amount of bill statement in August, 2005 (NT dollar)
* **BILL\_AMT3**: Amount of bill statement in July, 2005 (NT dollar)
* **BILL\_AMT4**: Amount of bill statement in June, 2005 (NT dollar)
* **BILL\_AMT5**: Amount of bill statement in May, 2005 (NT dollar)
* **BILL\_AMT6**: Amount of bill statement in April, 2005 (NT dollar)
* **PAY\_AMT1**: Amount of previous payment in September, 2005 (NT dollar)
* **PAY\_AMT2**: Amount of previous payment in August, 2005 (NT dollar)
* **PAY\_AMT3**: Amount of previous payment in July, 2005 (NT dollar)
* **PAY\_AMT4**: Amount of previous payment in June, 2005 (NT dollar)
* **PAY\_AMT5**: Amount of previous payment in May, 2005 (NT dollar)
* **PAY\_AMT6**: Amount of previous payment in April, 2005 (NT dollar)
* **default.payment.next.month**: Default payment (1=yes, 0=no)

****

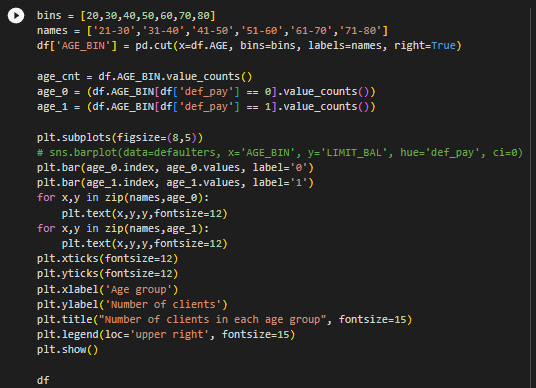
**VISUALIZATION:**

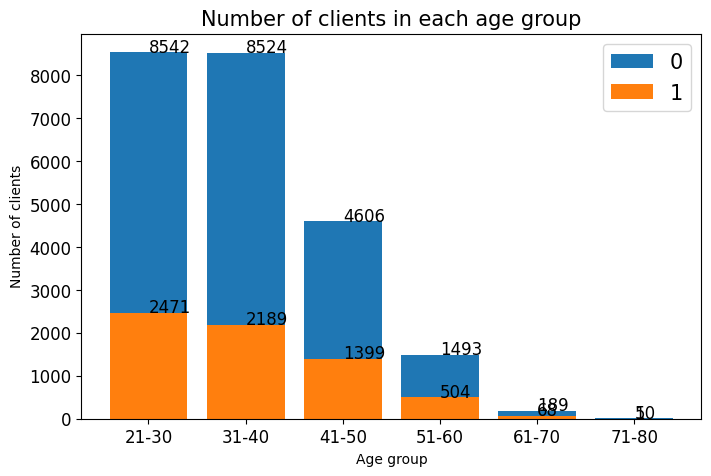
****



**INFERENCE:**

We can see that the dataset consists of 77% clients are not expected to default payment whereas 23% clients are expected to default the payment.





**INFERENCE:**

We have maximum clients from 21-30 age group followed by 31-40. Hence with increasing age group the number of clients that will default the payment next month is decreasing. Hence we can see that Age is important feature to predict the default payment for next month.

def draw\_histograms(df, variables, n\_rows, n\_cols, n\_bins):

    fig, axes = plt.subplots(nrows=n\_rows, ncols=n\_cols, figsize=(15, 10))

    for i, var\_name in enumerate(variables):

        row = i // n\_cols

        col = i % n\_cols

        sns.histplot(data=df, x=var\_name, bins=n\_bins, ax=axes[row, col])

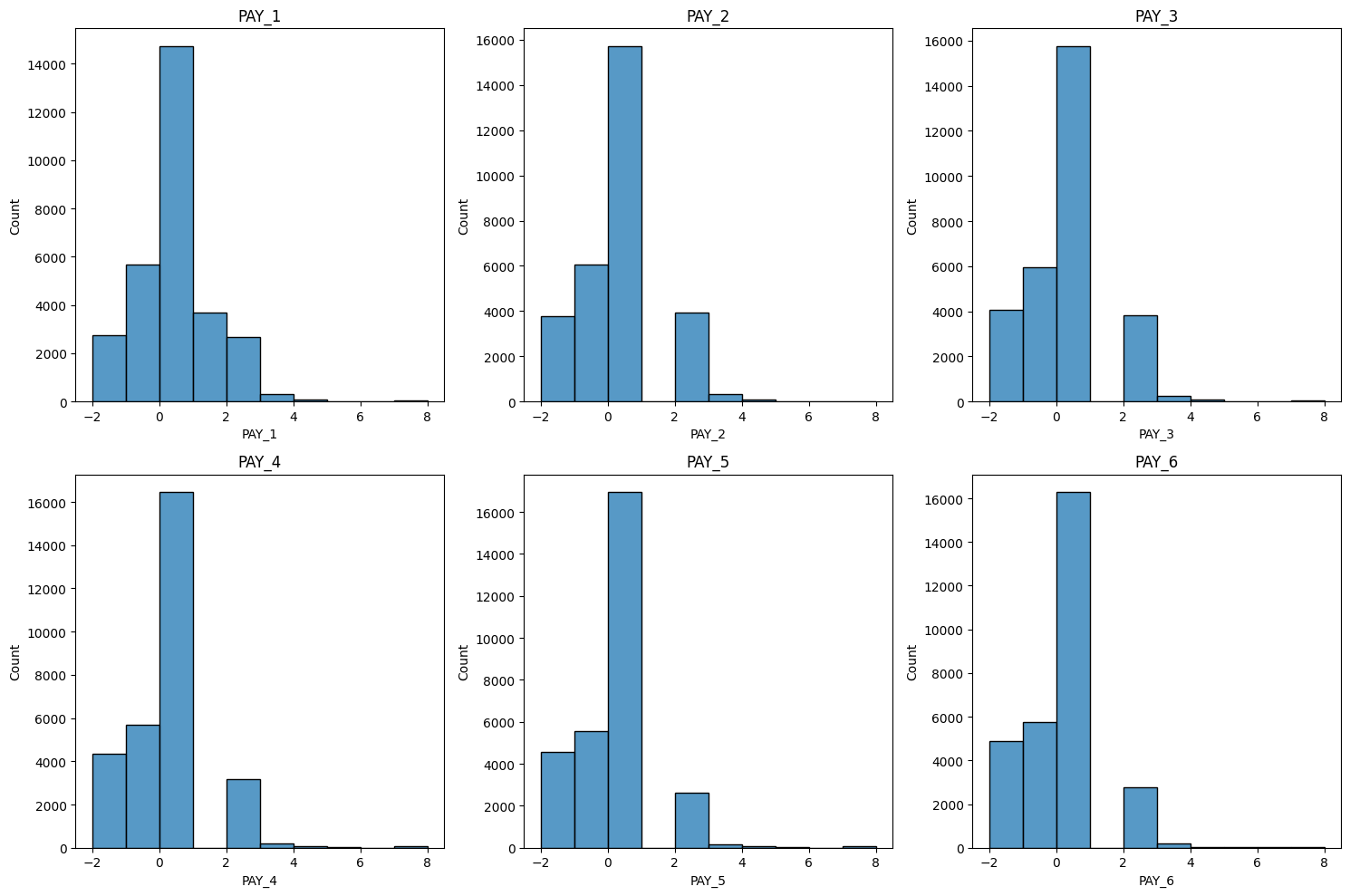
        axes[row, col].set\_title(var\_name)

    fig.tight\_layout()

    plt.show()

late = df[['PAY\_1','PAY\_2', 'PAY\_3', 'PAY\_4', 'PAY\_5', 'PAY\_6']]

draw\_histograms(late, late.columns, 2, 3, 10)



**INFERENCE:**

These are the graphs about number of people delaying payment in September, August, July, June, May, April months. We can observe that many people are paying duly

pay\_x\_fts = ['PAY\_1', 'PAY\_2', 'PAY\_3', 'PAY\_4', 'PAY\_5', 'PAY\_6']

plt.figure(figsize=(15,12))

for i,col in enumerate(pay\_x\_fts):

    plt.subplot(3,2,i + 1)

    ax = sns.barplot(x = col, y = "def\_pay", data = df, palette = 'rocket', errorbar = None)

    plt.ylabel("% of Default", fontsize= 12)

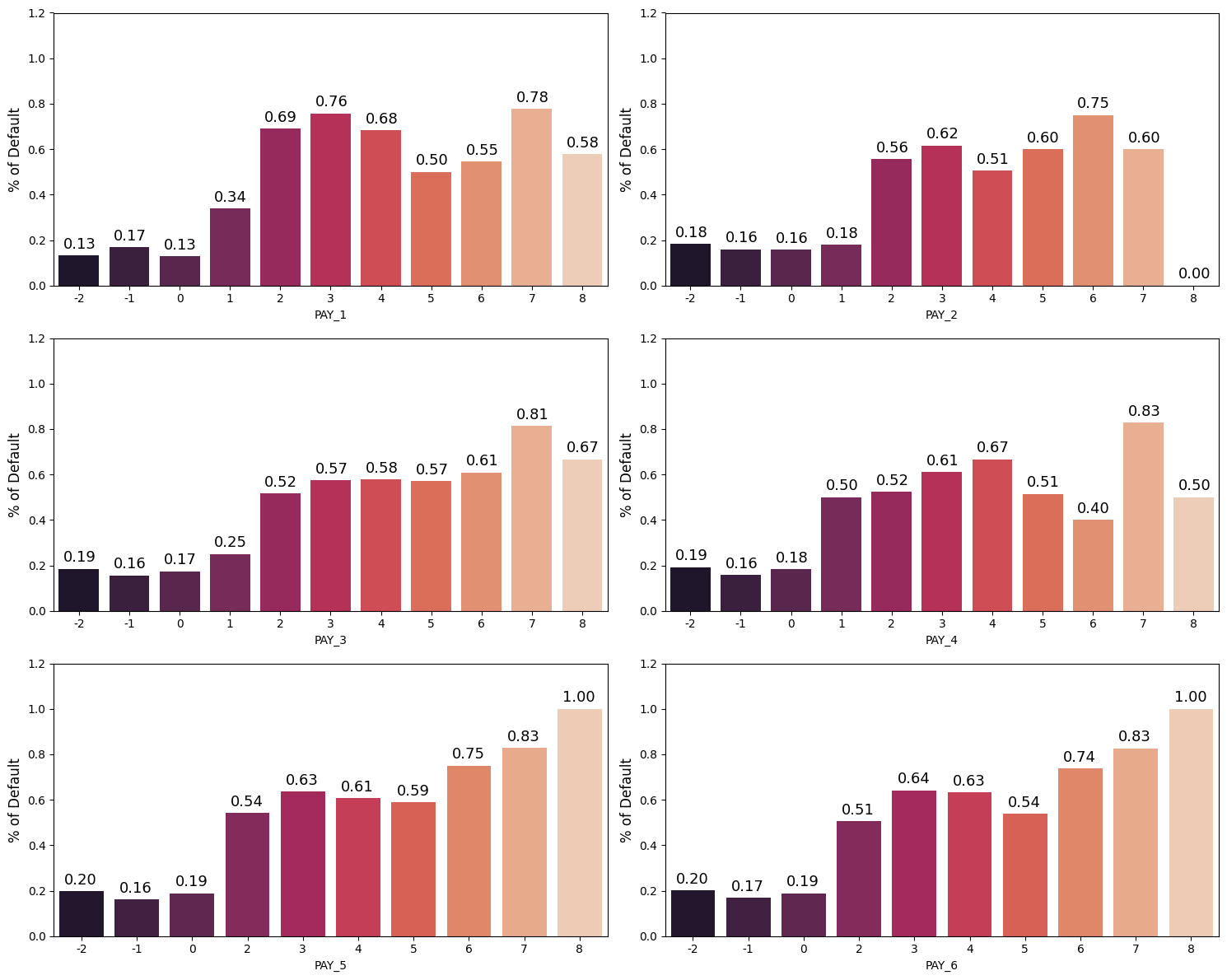
    plt.ylim(0,1.2)

    plt.tight\_layout()

    for p in ax.patches:

        ax.annotate("%.2f" %(p.get\_height()), (p.get\_x()+0.09, p.get\_height()+0.03),fontsize=13)

plt.show()

****

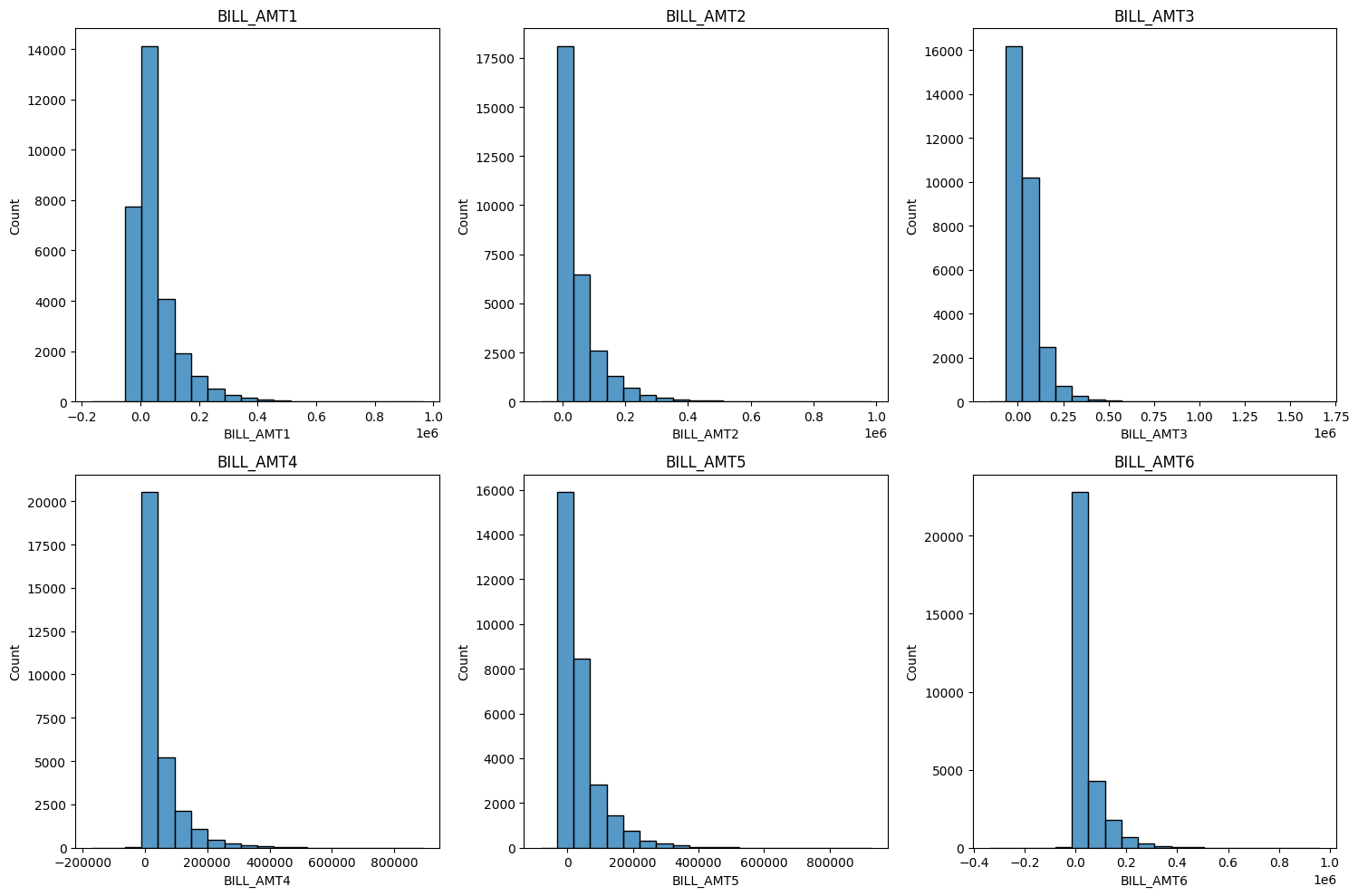
**INFERENCE:**

These are the graphs about percentage of clients going default in September, august, July, June, May, April months. There is more percentage of default in months May and April

bill\_amtx\_fts = ['BILL\_AMT1','BILL\_AMT2', 'BILL\_AMT3', 'BILL\_AMT4', 'BILL\_AMT5', 'BILL\_AMT6']

bills = df[bill\_amtx\_fts]

draw\_histograms(bills, bills.columns, 2, 3, 20)

****

**INFERENCE:**

These graphs show about the Amount of bill statement in months September, August, July, June, May, April

plt.figure(figsize=(15,12))

for i,col in enumerate(bill\_amtx\_fts):

    plt.subplot(3,2,i + 1)

    sns.kdeplot(df.loc[(df['def\_pay'] == 0), col], label = 'No Default',fill = True)

    sns.kdeplot(df.loc[(df['def\_pay'] == 1), col], label = 'Default', fill = True)

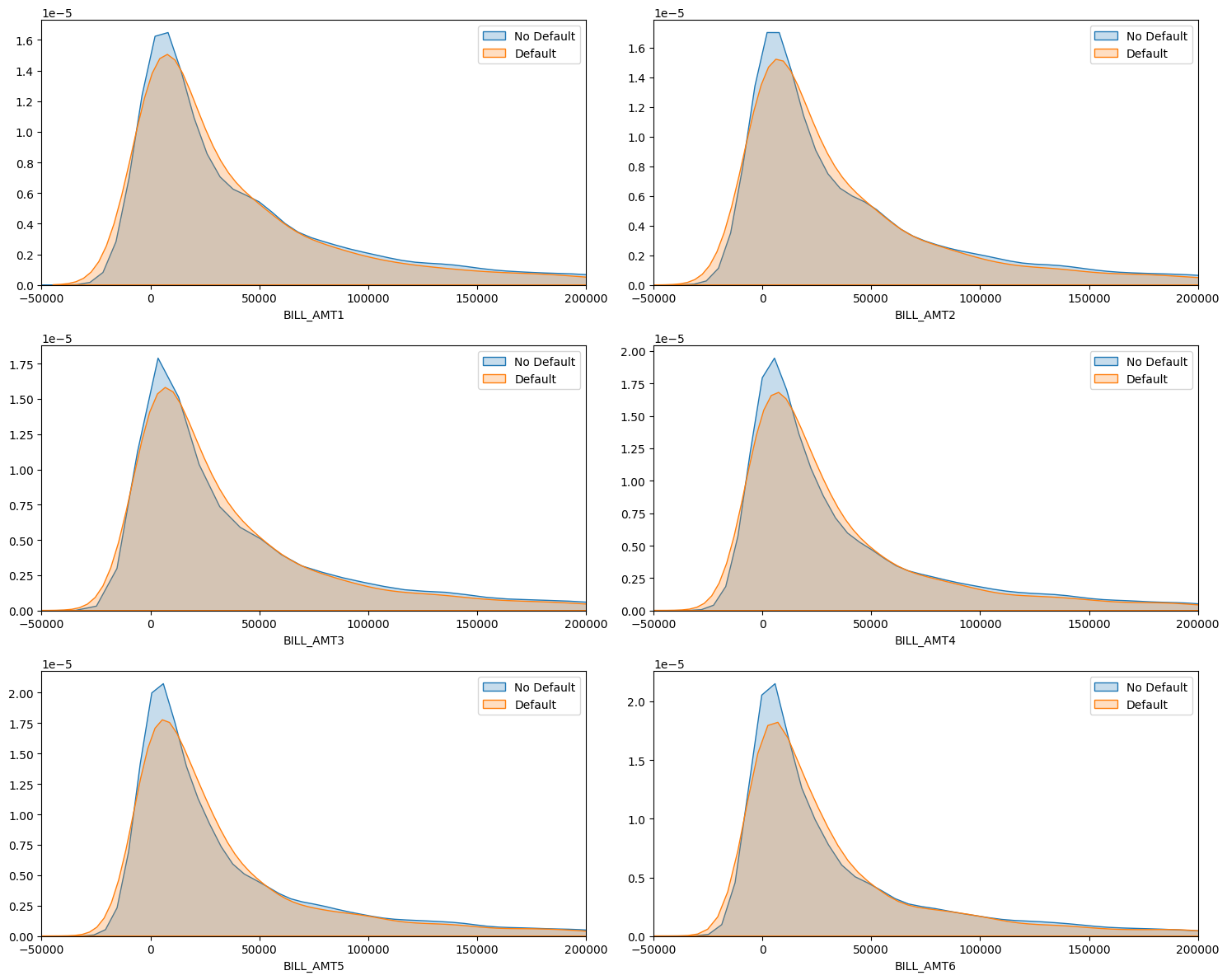
    plt.xlim(-50000,200000)

    plt.ylabel('')

    plt.legend()

    plt.tight\_layout()

plt.show()

****

**INEFERENCE:**

These plots are comparing the distribution of bill amounts for two classes: 'No Default' (where def\_pay is 0) and 'Default' (where def\_pay is 1).

plt.figure(figsize=(15, 12))

for i,col in enumerate(bill\_amtx\_bins):

    plt.subplot(3,2,i + 1)

    ax = sns.countplot(data = df, x = col, hue="def\_pay", palette = 'rocket')

    plt.ylim(0,13000)

    plt.ylabel('')

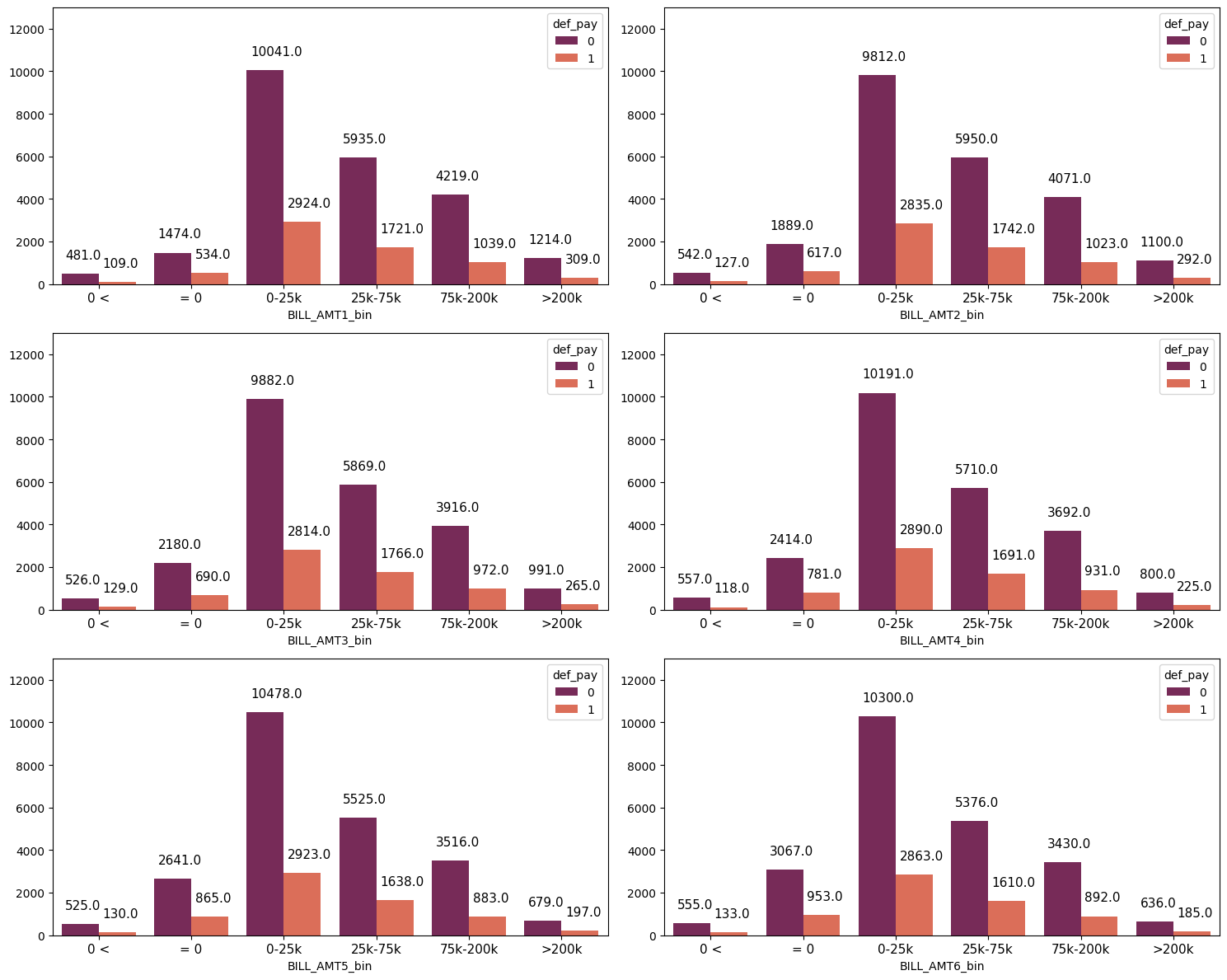
    plt.xticks([0,1,2,3,4,5],['0 <', '= 0', '0-25k', '25k-75k', '75k-200k', '>200k'], fontsize = 11)

    plt.tight\_layout()

    for p in ax.patches:

        ax.annotate((p.get\_height()), (p.get\_x()+0.04, p.get\_height()+700), fontsize = 11)

plt.show()

****

**INFERENCE:**

These graphs show the distribution of customers across different bill amount bins for two classes: 'No Default' and 'Default'. The x-axis typically represents the bill amount bins, while the y-axis represents the count of observations. The plots help you understand how different bill amount ranges are related to the likelihood of default. For example, you can see whether customers with higher bill amounts (e.g., '>200k') are more or less likely to default compared to those with lower bill amounts (e.g., '0-25k').

plt.figure(figsize=(15,12))

for i,col in enumerate(bill\_amtx\_bins):

    plt.subplot(3,2,i + 1)

    ax = sns.barplot(x = col, y = "def\_pay", data = df, palette = 'rocket', errorbar = None)

    plt.ylabel("% of Default", fontsize= 12)

    plt.ylim(0,0.5)

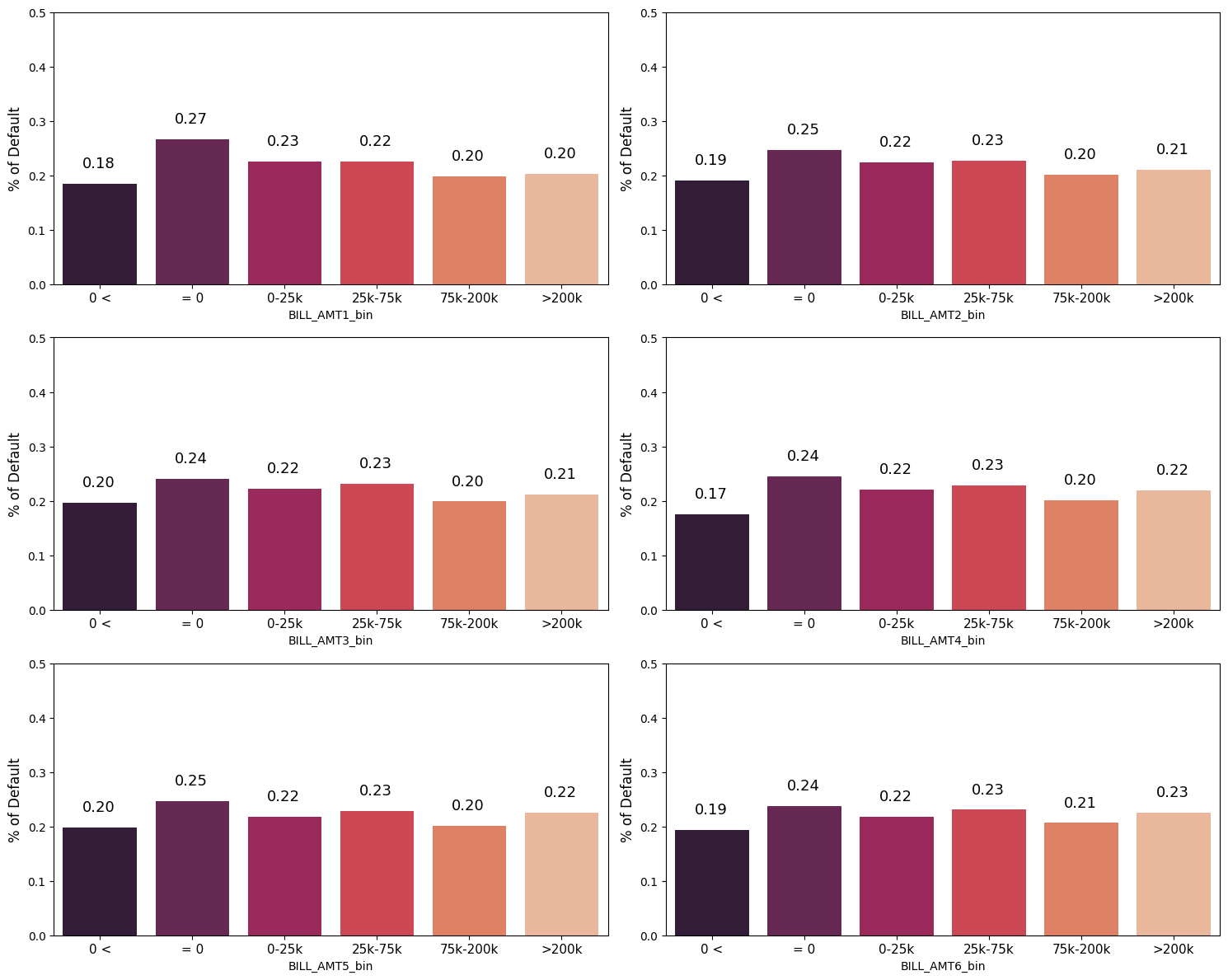
    plt.xticks([0,1,2,3,4,5],['0 <', '= 0', '0-25k', '25k-75k', '75k-200k', '>200k'], fontsize = 11)

    plt.tight\_layout()

    for p in ax.patches:

        ax.annotate("%.2f" %(p.get\_height()), (p.get\_x()+0.21, p.get\_height()+0.03),fontsize=13)

plt.show()

****

**INFERENCE:**

These show the percentage of default people going in bill statements of different categories

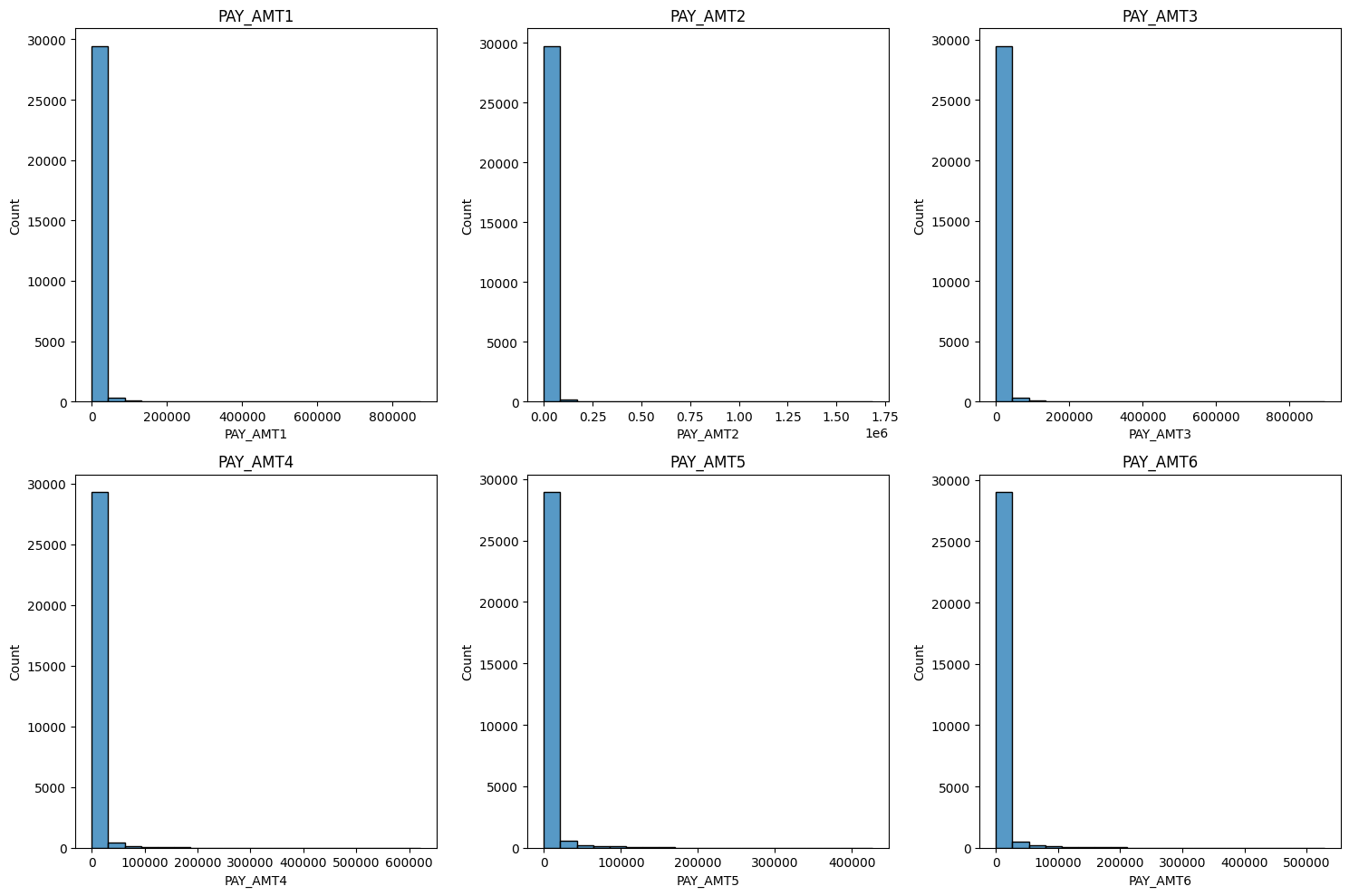
'0 <', '= 0', '0-25k', '25k-75k', '75k-200k', '>200k'

Those who have a negative bill statement have a lower chance of default than the rest. What stands out is that there is a little higher chance of default for those who didn't have a bill in the previous months.

pay\_amtx\_fts = ['PAY\_AMT1','PAY\_AMT2', 'PAY\_AMT3', 'PAY\_AMT4', 'PAY\_AMT5', 'PAY\_AMT6']

pay = df[pay\_amtx\_fts]

draw\_histograms(pay, pay.columns, 2, 3, 20)

****

**INFERENCE:**

These graphs show the amount of previous payment in the respective months September, August, July, June, May, April

plt.figure(figsize=(15,12))

for i,col in enumerate(pay\_amtx\_fts):

    plt.subplot(3,2,i + 1)

    sns.kdeplot(df.loc[(df['def\_pay'] == 0), col], label = 'No Default', fill = True)

    sns.kdeplot(df.loc[(df['def\_pay'] == 1), col], label = 'Default', fill = True)

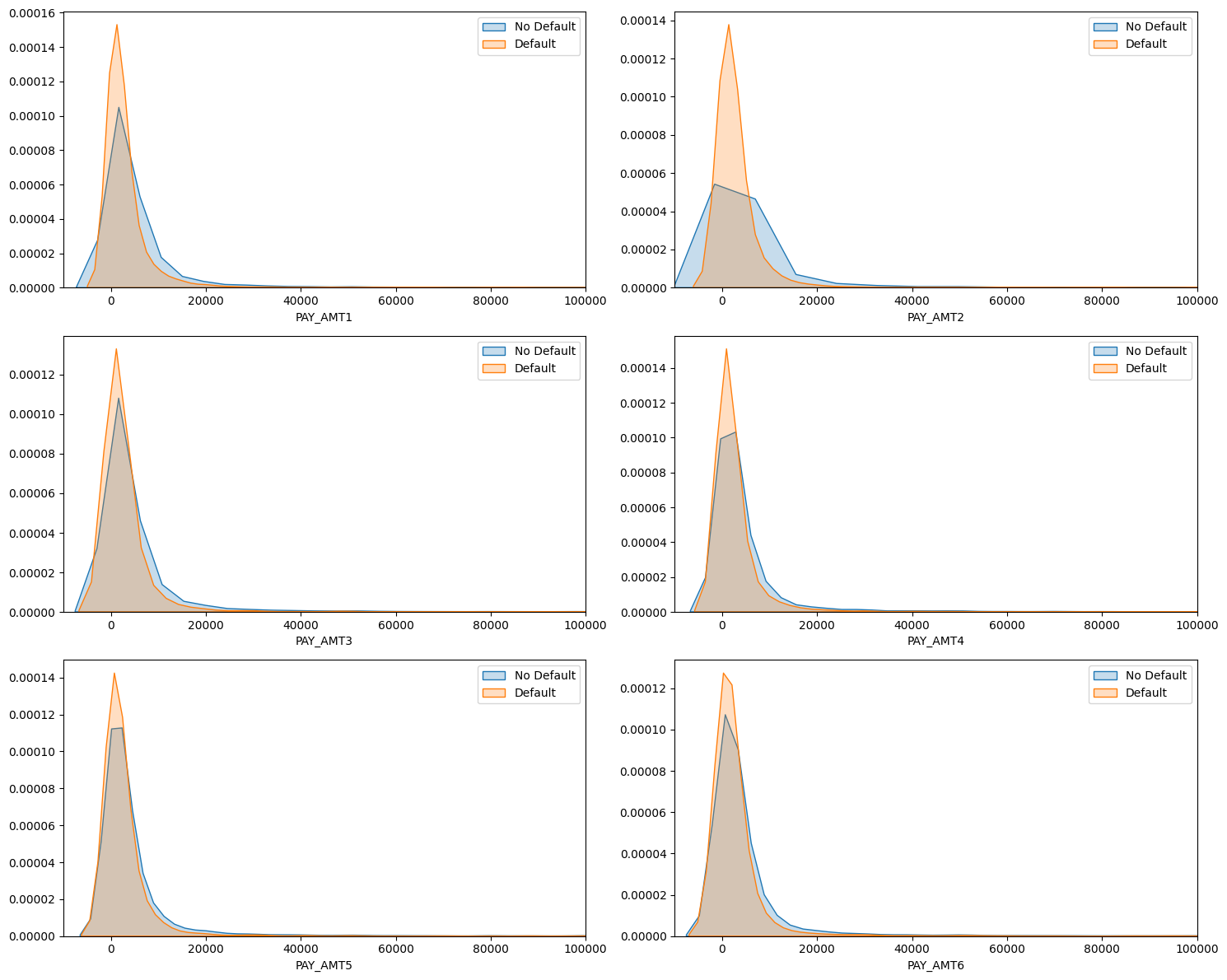
    plt.xlim(-10000,100000)

    plt.ylabel('')

    plt.legend()

    plt.tight\_layout()

plt.show()

****

**INFERENCE:**

These plots compare the distribution of payment amounts for two classes: 'No Default' (where def\_pay is 0) and 'Default' (where def\_pay is 1). These provide a visual representation of how payment amounts are distributed for different classes, aiding in understanding the relationship between payment behavior and default.

plt.figure(figsize=(15,12))

for i,col in enumerate(pay\_amtx\_bins):

    plt.subplot(3,2,i + 1)

    ax = sns.countplot(data = df, x = col, hue="def\_pay", palette = 'rocket')

    plt.ylim(0,23000)

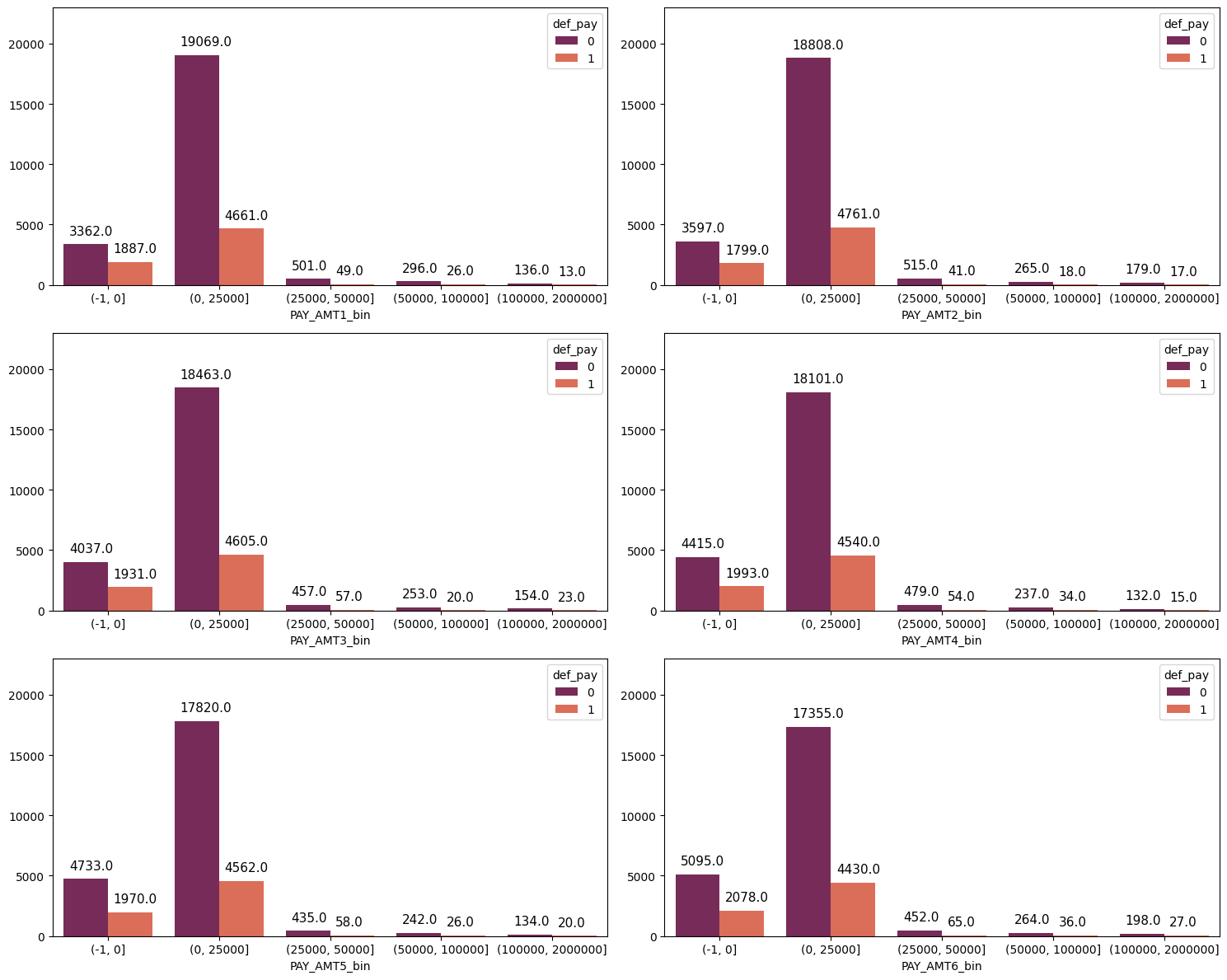
    plt.ylabel('')

    plt.tight\_layout()

    for p in ax.patches:

        ax.annotate((p.get\_height()), (p.get\_x()+0.05, p.get\_height()+800), fontsize=11)

plt.show()

****

**INEFERENCE:**

The plots show how different payment amount bins relate to the likelihood of default. You can observe whether certain payment amount categories are associated with a higher or lower likelihood of default.

plt.figure(figsize=(15,12))

for i,col in enumerate(pay\_amtx\_bins):

    plt.subplot(3,2,i + 1)

    ax = sns.barplot(x = col, y = "def\_pay", data = df, palette = 'rocket', errorbar = None)

    plt.ylabel("% of Default", fontsize= 12)

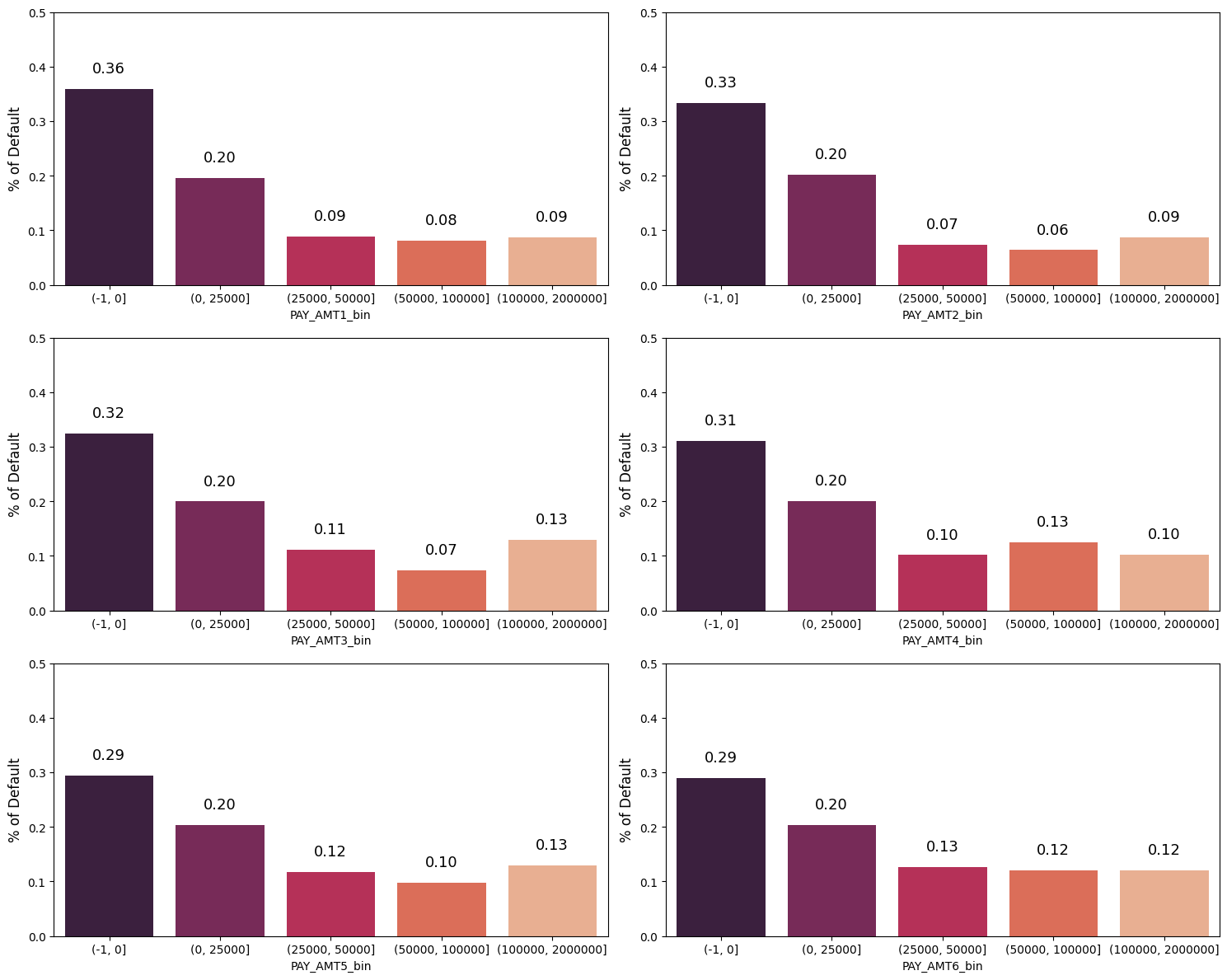
    plt.ylim(0,0.5)

    plt.tight\_layout()

    for p in ax.patches:

        ax.annotate("%.2f" %(p.get\_height()), (p.get\_x()+0.25, p.get\_height()+0.03),fontsize=13)

plt.show()

****

**INFERENCE:**

These graphs show the percentage of default people going in months September, August, July, June, May, April with Amount of previous payment to be paid

There is a higher default rate among those who paid nothing in previous months and lower rates among those paid over 25k of NT dollars.

def show\_value\_counts(col):

    print(col)

    value\_counts = df[col].value\_counts()

    percentage = value\_counts / len(df) \* 100

    result\_df = pd.DataFrame({'Value': value\_counts.index, 'Count': value\_counts, 'Percentage': percentage})

    result\_df = result\_df.sort\_values(by='Value')

    print(result\_df)

    print('--------------------------')

    generate\_pie\_plot(result\_df)

def generate\_pie\_plot(data\_frame):

    plt.figure(figsize=(6, 4))

    plt.pie(data\_frame['Count'], labels=data\_frame['Value'], autopct='%1.1f%%')

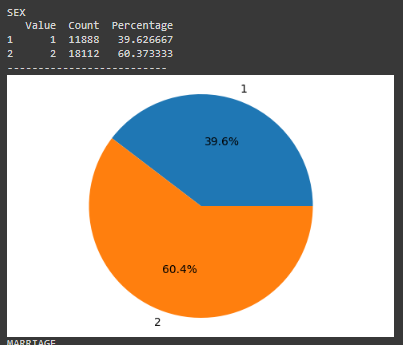
    plt.axis('equal')

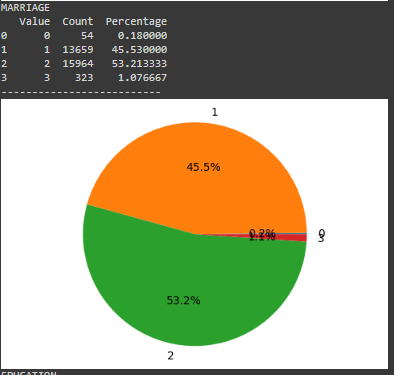
    plt.show()

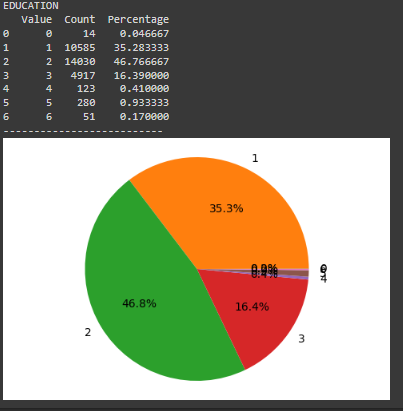
show\_value\_counts('SEX')

show\_value\_counts('MARRIAGE')

show\_value\_counts('EDUCATION')

****

****

****

**INFERENCE:**

There are significantly more women than men

Most of Credit Card holders have university and graduate level education.

fig, axes = plt.subplots(nrows=3, ncols=1, figsize=(15, 13))

# Count plot for MARRIAGE

ax1 = sns.countplot(data=df, x='MARRIAGE', hue='def\_pay', palette='rocket', ax=axes[0])

ax1.set\_xlabel("Marital Status", fontsize=12)

ax1.set\_ylabel("Number of Clients", fontsize=12)

ax1.set\_ylim(0, 15000)

ax1.set\_xticks([0, 1, 2])

ax1.set\_xticklabels(['Married', 'Single', 'Others'], fontsize=11)

for p in ax1.patches:

    ax1.annotate(int(p.get\_height()), (p.get\_x() + 0.12, p.get\_height() + 500))

# Count plot for EDUCATION

ax2 = sns.countplot(data=df, x='EDUCATION', hue='def\_pay', palette='rocket', ax=axes[1])

ax2.set\_xlabel("Educational Background", fontsize=12)

ax2.set\_ylabel("Number of Clients", fontsize=12)

ax2.set\_ylim(0, 15000)

ax2.set\_xticks([0, 1, 2, 3])

ax2.set\_xticklabels(['Grad School', 'University', 'High School', 'Others'], fontsize=11)

for p in ax2.patches:

    ax2.annotate(int(p.get\_height()), (p.get\_x() + 0.12, p.get\_height() + 500))

# Count plot for SEX

ax3 = sns.countplot(data=df, x='SEX', hue='def\_pay', palette='rocket', ax=axes[2])

ax3.set\_xlabel("Gender", fontsize=12)

ax3.set\_ylabel("Number of Clients", fontsize=12)

ax3.set\_ylim(0, 15000)

ax3.set\_xticks([0, 1])

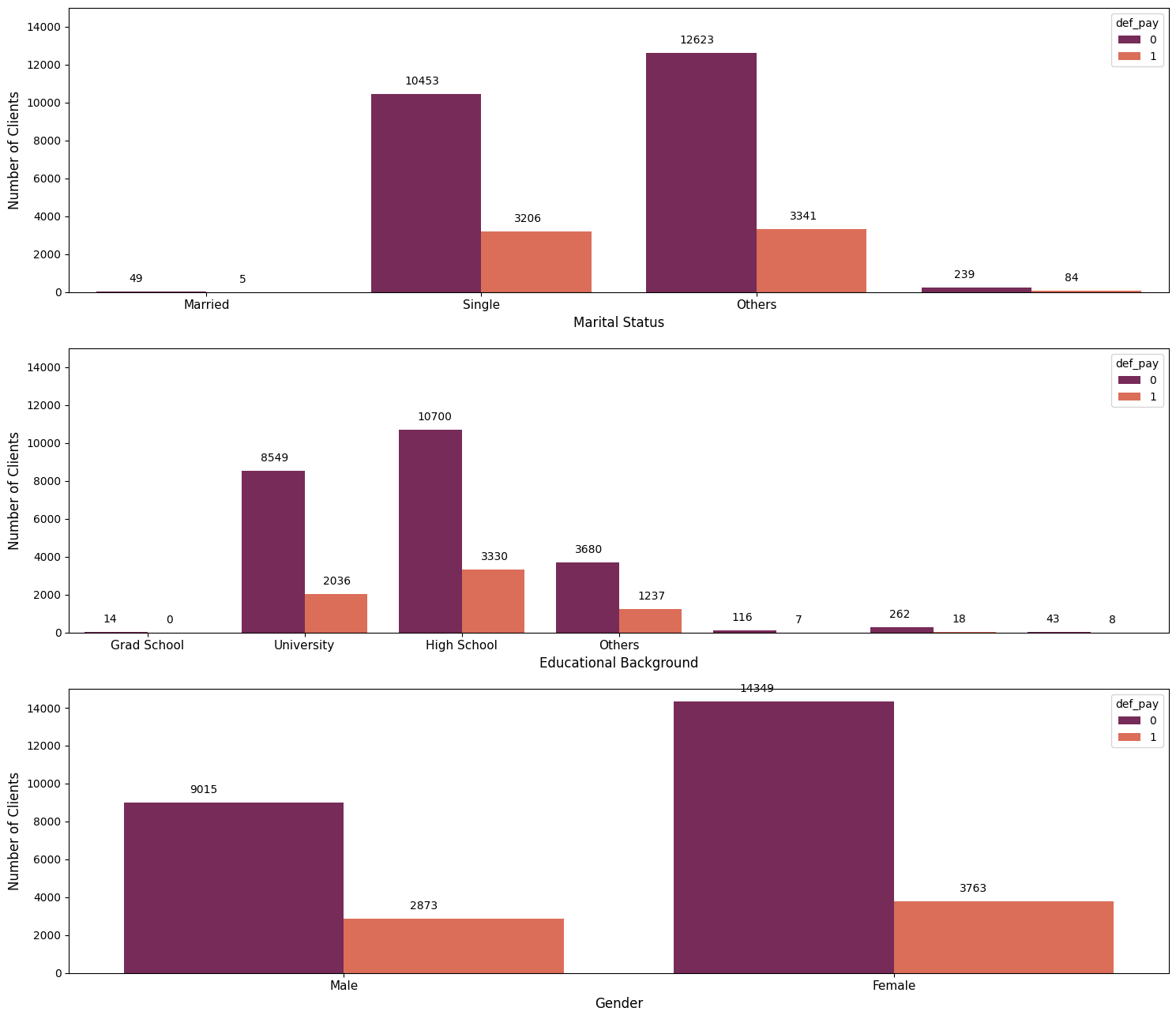
ax3.set\_xticklabels(['Male', 'Female'], fontsize=11)

for p in ax3.patches:

    ax3.annotate(int(p.get\_height()), (p.get\_x() + 0.12, p.get\_height() + 500))

plt.tight\_layout()

plt.show()

****

fig, axes = plt.subplots(nrows=3, ncols=1, figsize=(18, 15))

# Bar plot for EDUCATION

ax1 = sns.barplot(x="EDUCATION", y="def\_pay", data=df, palette='rocket', errorbar=None, ax=axes[0])

ax1.set\_ylabel("% of Default", fontsize=12)

ax1.set\_ylim(0, 0.5)

ax1.set\_xticks([0, 1, 2, 3])

ax1.set\_xticklabels(['Grad School', 'University', 'High School', 'Others'], fontsize=13)

for p in ax1.patches:

    ax1.annotate("%.2f" % (p.get\_height()), (p.get\_x() + 0.30, p.get\_height() + 0.03), fontsize=15)

# Bar plot for MARRIAGE

ax2 = sns.barplot(x="MARRIAGE", y="def\_pay", data=df, palette='rocket', errorbar=None, ax=axes[1])

ax2.set\_ylabel("% of Default", fontsize=12)

ax2.set\_ylim(0, 0.5)

ax2.set\_xticks([0,1,2])

ax2.set\_xticklabels(['Married', 'Single', 'Others'], fontsize=13)

for p in ax2.patches:

    ax2.annotate("%.2f" % (p.get\_height()), (p.get\_x() + 0.30, p.get\_height() + 0.03), fontsize=15)

# Bar plot for SEX

ax3 = sns.barplot(x="SEX", y="def\_pay", data=df, palette='rocket', errorbar=None, ax=axes[2])

ax3.set\_ylabel("% of Default", fontsize=12)

ax3.set\_ylim(0, 0.5)

ax3.set\_xticks([0, 1])

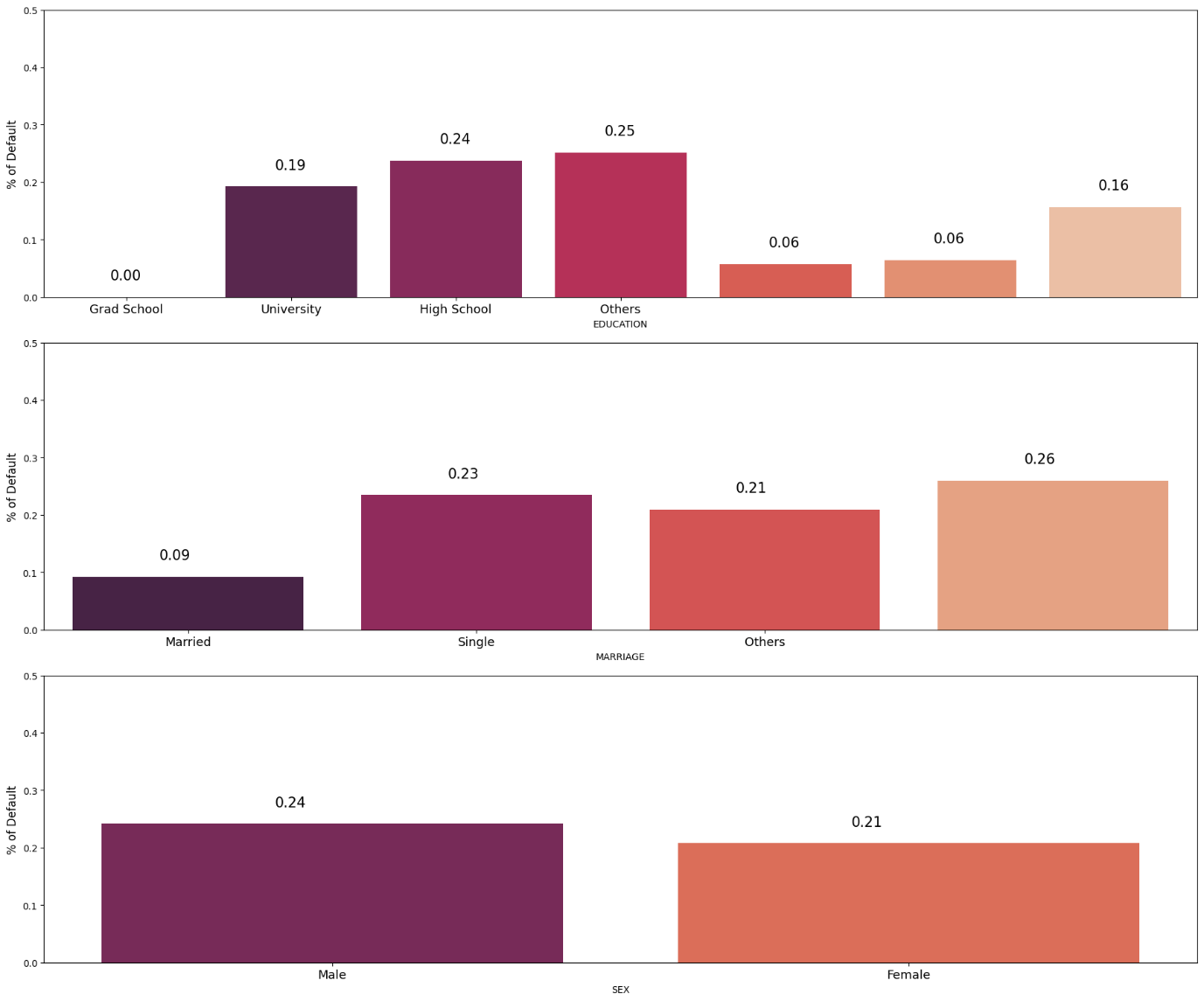
ax3.set\_xticklabels(['Male', 'Female'], fontsize=13)

for p in ax3.patches:

    ax3.annotate("%.2f" % (p.get\_height()), (p.get\_x() + 0.30, p.get\_height() + 0.03), fontsize=15)

plt.tight\_layout()

plt.show()

****

**INFERENCE:**

The likelihood of being a defaulter decreases as your education level increases.

Married and other marital statuses (possibly including divorced) have an approximately 0.24 probability of being defaulters, whereas single individuals have a lower likelihood at 0.21.

Despite a smaller number of males in the dataset compared to females, males exhibit a higher likelihood of being defaulters.

plt.figure(figsize=(12,4))

ax = sns.barplot(x = "SEX", y = "def\_pay", hue = "MARRIAGE", data = df, palette = 'rocket', errorbar = None)

plt.ylabel("% of Default", fontsize= 12)

plt.ylim(0,0.5)

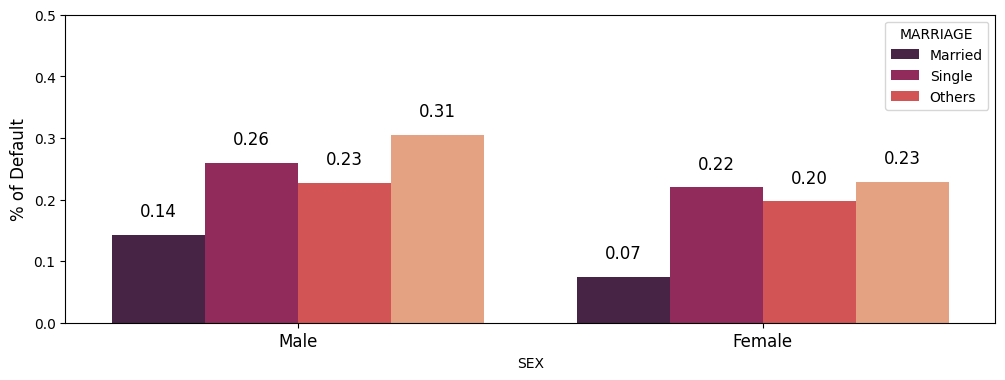
plt.xticks([0,1],['Male', 'Female'], fontsize = 12)

plt.legend(['Married', 'Single','Others'], title = 'MARRIAGE')

for p in ax.patches:

    ax.annotate("%.2f" %(p.get\_height()), (p.get\_x()+0.06, p.get\_height()+0.03),fontsize=12)

plt.show()

****

plt.figure(figsize=(12,4))

ax = sns.barplot(x = "SEX", y = "def\_pay", hue = "EDUCATION", data = df, palette = 'rocket', errorbar = None)

plt.ylabel("% of Default", fontsize= 12)

plt.ylim(0,0.5)

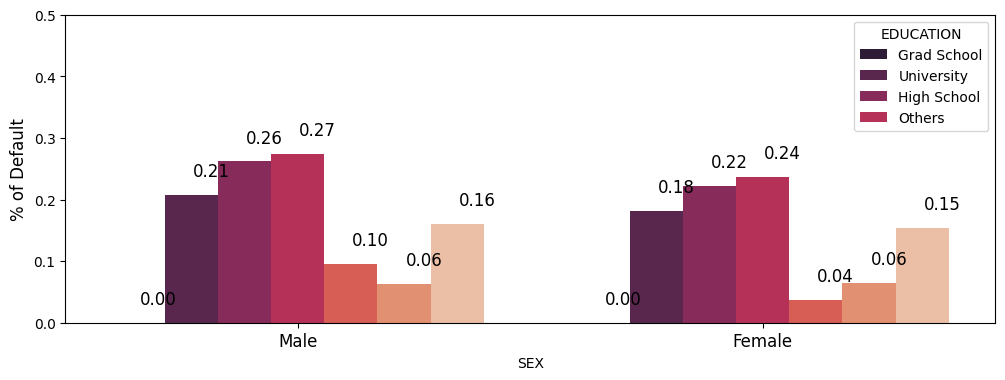
plt.xticks([0,1],['Male', 'Female'], fontsize = 12)

plt.legend(['Grad School', 'University', 'High School', 'Others'], title = 'EDUCATION')

for p in ax.patches:

    ax.annotate("%.2f" %(p.get\_height()), (p.get\_x()+0.06, p.get\_height()+0.03),fontsize=12)

plt.show()

****

plt.figure(figsize=(12,4))

ax = sns.barplot(x = "MARRIAGE", y = "def\_pay", hue = "EDUCATION", data = df, palette = 'rocket', errorbar = None)

plt.ylabel("% of Default", fontsize= 12)

plt.ylim(0,0.5)

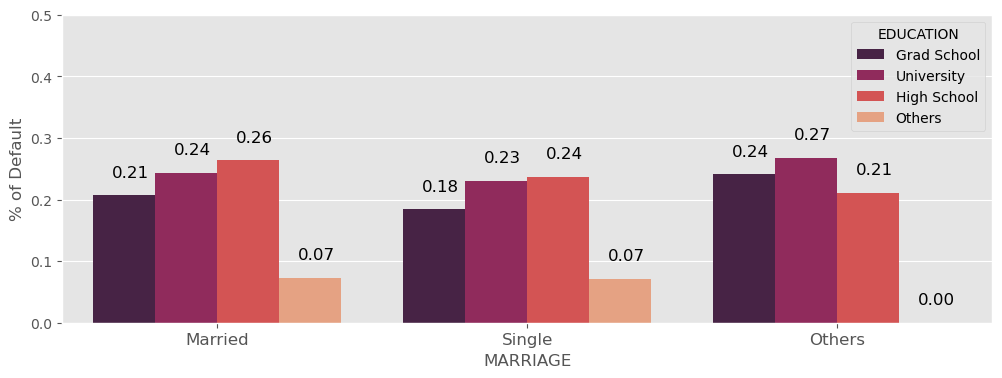
plt.xticks([0,1,2],['Married', 'Single','Others'], fontsize = 12)

plt.legend(['Grad School', 'University', 'High School', 'Others'], title = 'EDUCATION')

for p in ax.patches:

    ax.annotate("%.2f" %(p.get\_height()), (p.get\_x()+0.06, p.get\_height()+0.03),fontsize=12)

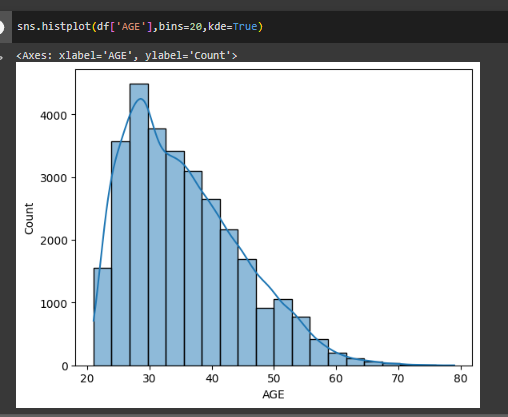
plt.show()



**INFERENCE:**

Being male, married, and having a high school education seems to increase the likelihood of being a defaulter.

People who are marked as "Others" in their marital status (likely indicating divorced individuals) have a notable probability of around 0.29 for facing defaults, which is a relatively higher occurrence.

****

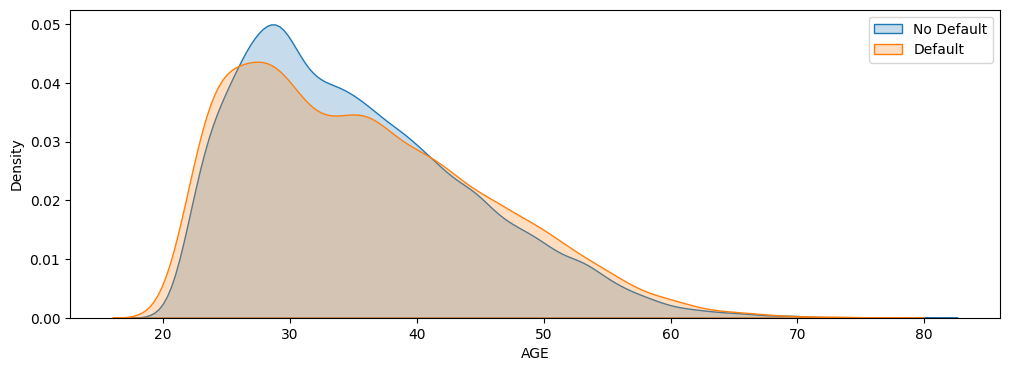
plt.figure(figsize=(12,4))

sns.kdeplot(df.loc[(df['def\_pay'] == 0), 'AGE'], label = 'No Default', fill = True)

sns.kdeplot(df.loc[(df['def\_pay'] == 1), 'AGE'], label = 'Default', fill = True)

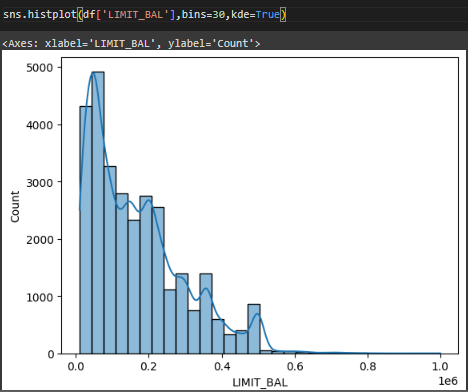
plt.legend()

plt.show()

****

**INFERENCE:**

The age group between 20 to 30 years old appears to have a higher propensity for defaults.



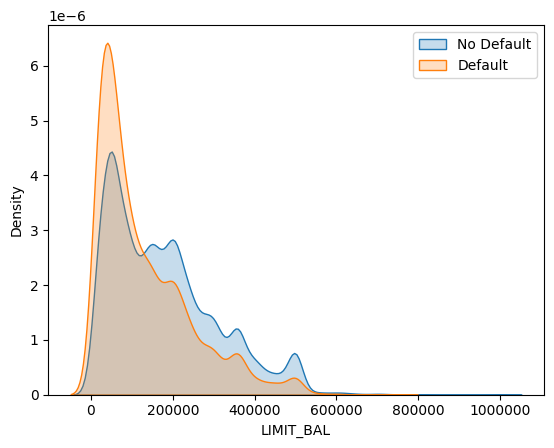
sns.kdeplot(df.loc[(df['def\_pay'] == 0), 'LIMIT\_BAL'], label = 'No Default', fill = True)

sns.kdeplot(df.loc[(df['def\_pay'] == 1), 'LIMIT\_BAL'], label = 'Default', fill = True)

plt.ticklabel\_format(style='plain', axis='x')

plt.legend()

plt.show()



plt.figure(figsize=(15, 20))

plt.subplot(4, 1, 1)

sns.boxplot(x="MARRIAGE", y="LIMIT\_BAL", data=df, palette='rocket', showmeans=True,

            meanprops={"markerfacecolor": "red", "markeredgecolor": "black", "markersize": "10"})

plt.ticklabel\_format(style='plain', axis='y')

plt.xticks([0, 1, 2], ['Married', 'Single', 'Others'], fontsize=11)

plt.subplot(4, 1, 2)

sns.boxplot(x="EDUCATION", y="LIMIT\_BAL", data=df, palette='rocket', showmeans=True,

            meanprops={"markerfacecolor": "red", "markeredgecolor": "black", "markersize": "10"})

plt.ticklabel\_format(style='plain', axis='y')

plt.xticks([0, 1, 2, 3], ['Grad School', 'University', 'High School', 'Others'], fontsize=11)

plt.subplot(4, 1, 3)

sns.boxplot(x="SEX", y="LIMIT\_BAL", data=df, palette='rocket', showmeans=True,

            meanprops={"markerfacecolor": "red", "markeredgecolor": "black", "markersize": "10"})

plt.ticklabel\_format(style='plain', axis='y')

plt.xticks([0, 1], ['Male', 'Female'], fontsize=12)

plt.subplot(4, 1, 4)

sns.boxplot(x="AgeBin", y="LIMIT\_BAL", data=df, palette='rocket', showmeans=True, order=AgeBin\_order,

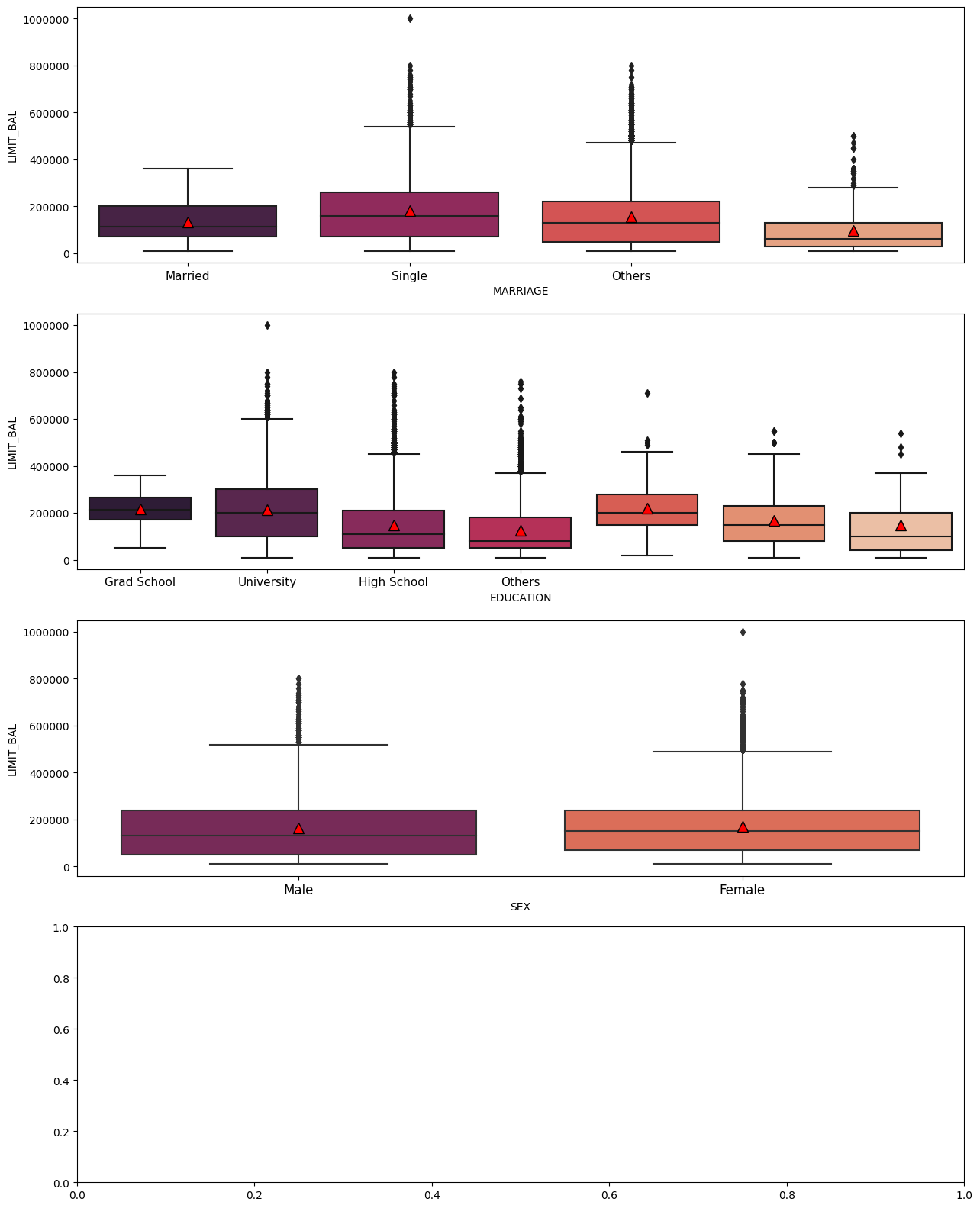
            meanprops={"markerfacecolor": "red", "markeredgecolor": "black", "markersize": "10"})

plt.ticklabel\_format(style='plain', axis='y')

plt.xlabel("Age Group", fontsize=12)

plt.tight\_layout()

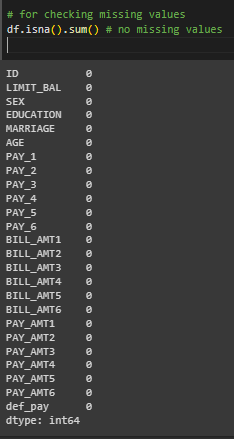
plt.show()



**INFERENCE**:

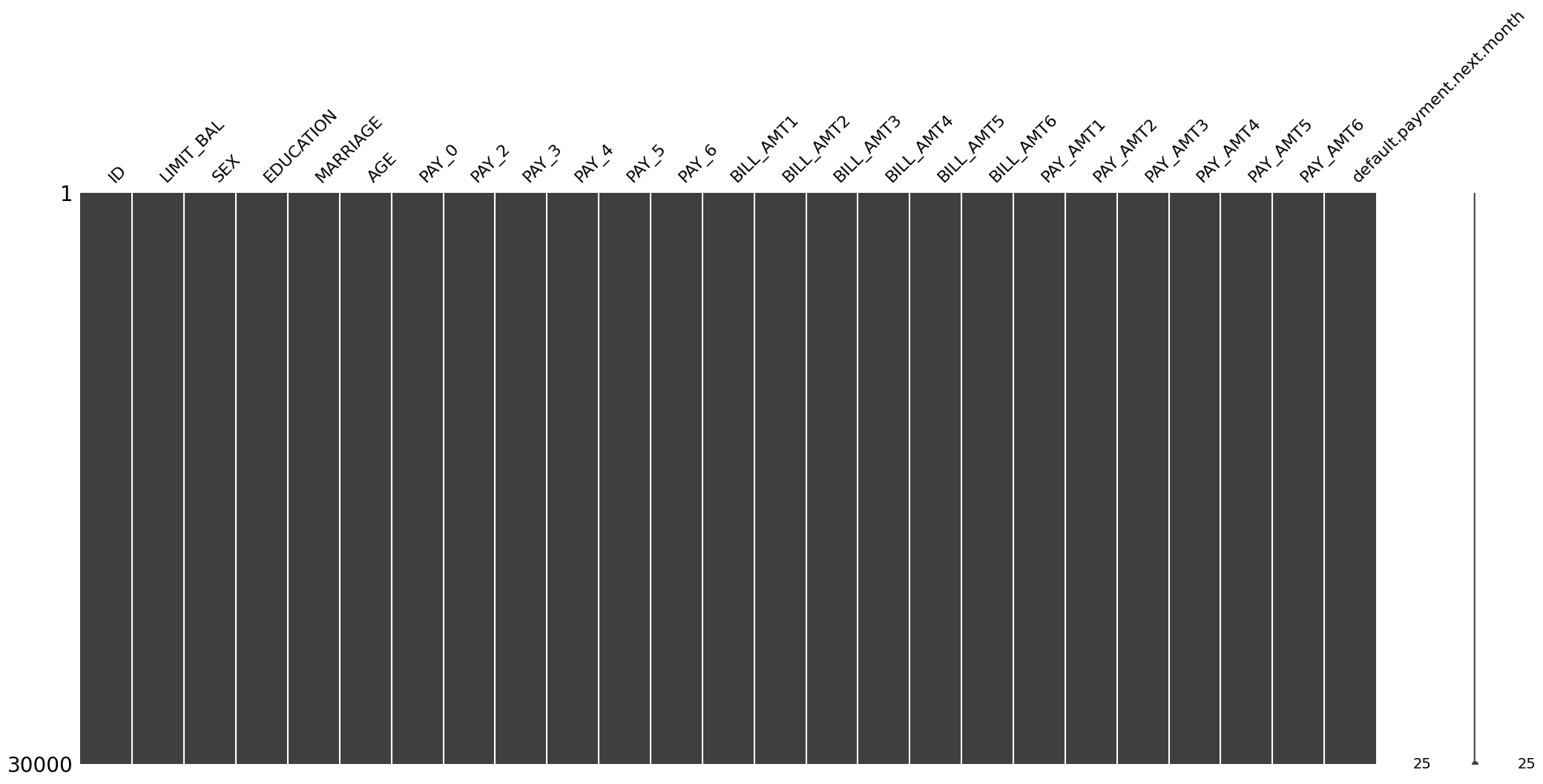
The Person with Highest Credit Limit (i.e. 1 million) is a female, married and belongs to 40 to 50 age group

**HANDLING MISSING DATA:**



No missing value hence no imputation, now we directly move towards visualzation of dataset

**NOICE DATA CHECKING:**



Renaming columns with meaningful names

