California State University, Los Angeles

Analytics Insight into Credit Card Dynamics: Unveiling Consumer Behavior and Financial Patterns

Snehil Sarkar

Sapan Shah

CIS 5270-01: Business Intelligence

Dr. Shilpa Balan

15 May. 2024

Table of Contents

Introduction

In the 1960s, a big step forward in credit card technology helped make them more popular. An engineer named Forrest Parry figured out how to put magnetic tape on the back of cards. This let stores swipe the card to get your information when you pay. Before that, magnetic tape was used for music, not cards.

First, there are the companies that give out credit cards – we call them "issuers." These are the banks or financial institutions where we apply for a credit card. They decide whether to approve us and manage our account. When we buy something with the card, we are basically borrowing money from the issuer. Examples of issuers include big names like Chase and Citi, as well as smaller ones like local banks or credit unions.

Then, there are the companies that handle the technical side of credit card transactions – we call them "payment networks." These companies process the transactions when we swipe or tap our card to make a purchase. In the U.S., the main ones are Visa, Mastercard, American Express, and Discover. They make sure the money gets from our account to the store's account smoothly.

In today's financial world, credit card companies want to understand their customers better. They use data analysis to figure out what customers need and how to improve their services. This project uses Python to look at a big set of data from a credit card company. This dataset includes detailed profiles of customers, encompassing demographic information (like age and gender) and financial metrics (such as salary, credit card limit, and transactional behavior). By studying this data, we hope to uncover insights that can help the company make informed decisions and enhance customer satisfaction.

The data we're looking at has lots of different kinds of information about customers. It tells us things like how old they are, if they're married, how much money they make, and more. We will use tools in Python to organize and visualize this data, so we can see if there are any patterns or trends that can help us understand customers better.

We have some questions we want to answer with this data. We want to know things like how old most customers are, if different groups of customers are more likely to stop using their credit cards, and how people's credit limits are affected by their education level and other financial metrics. By finding answers to these questions, we hope to give the credit card company useful insights that can help them improve their services and make their customers happier.

The results of our analysis can make a big difference for the credit card company. If they understand their customers better, they can make changes to their services that people will like more. This could mean offering new types of credit cards, changing how they advertise, or even just making their website easier to use. By using data to make these decisions, the company can stay competitive and keep their customers happy.

This project is all about using data to learn more about credit card customers. By studying the data closely, we hope to find helpful information that the credit card company can use to make their services better. With the right insights, they can make smart decisions that keep their customers happy and their business successful.

Data Description Tables

Dataset : Credit Card Customers

This dataset offers a detailed profile of customers associated with a credit card company. Each row corresponds to an individual customer, providing a comprehensive array of attributes.

These attributes encompass demographic information like age, marital status, and gender, alongside financial metrics such as salary, credit card limit, and transactional behavior.

Dataset URL: https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers/data

Column Name	Data Type	Description	Example
Client Num	Integer	Unique Identification number of customers	818770008
Attrition Flag	Text	Indicates whether the customer is an existing customer or has churned	Existing Customer or Attrited Customer
Customer Age	Integer	Age of the customer	49 years
Gender	Text	Male/Female	M- Male Customer F- Female Customer
Dependent Count	Integer	Indicates the number of dependents (e.g., children, spouses) that the customer has.	4
Education Level	Text	Represents the highest level of education attained by the customer, such as "High School" or "Graduate", “Doctorate” ,”College”, “Uneducated”, “Postgraduate”.	Graduate
Marital Status	Text	Describes the marital status of the customer such as “Single”, “Married”, “Divorced”.	Married
Income Category	Text	Indicates the income range of the customer.	$60k-$80k
Card Category	Text	Specifies the type of credit card held by the customer.	Blue or Silver or Gold

Months on Book	Integer	Represents the number of months that the customer has been a cardholder.	40 months
Total Relationship Count	Integer	Indicates the total number of products held by the customer (e.g., credit cards, loans) with the company.	6
Months Inactive 12 months	Integer	Describes the number of months in the past 12 months during which the customer has been inactive (i.e., not engaged in transactions).	2 months
Contacts Count 12 months	Integer	Represents the number of times the customer has been contacted by the company within the past 12 months.	3 times
Credit Limit	Float	Specifies the credit limit assigned to the customer's credit card.	$8968
Total Revolving Bal	Integer	Indicates the total balance on the customer's revolving credit account (e.g., credit card balance).	$777
Avg_Open_To_Buy	Float	Represents the average amount of credit available to the customer for purchases (credit limit minus current balance).	$7392
Total_Amt_Chng_Q4_Q1	Float	Describes the percentage change in the total transaction amount between the last quarter and the first quarter.	1.335
Total_Trans_Amt	Integer	Represents the total transaction amount (in dollars) for the customer.	$1144

Total_Trans_Ct	Integer	Indicates the total number of transactions made by the customer.	42
Total_Ct_Chng_Q4_Q1	Float	Describes the percentage change in the total transaction count between the last quarter and the first quarter.	1.625
Avg_Utilization_Ratio	Float	Represents the average utilization ratio of the customer's credit limit across all accounts.	0.061

Data Cleaning: Credit Card Customers

1. **Renaming column names for data clarity:**

It is essential to ensure that the data we work with is not only accurate but also easily understandable. So before starting data cleaning or any analysis, I renamed the column name When encountering column names like " Avg_Open_To_Buy," clarity can be compromised. To address this, I routinely rename columns to more intuitive and descriptive names, such as " Available_Credit," making it clearer that this column represents the average amount of credit available to the customer for purchases (credit limit minus current balance). This approach enhances both my own understanding and that of my colleagues, while also improving the data's interpretability and usability for any subsequent analysis or modeling endeavors.

Code for Data Cleaning-1:

```
print(df.columns)

# Define the new column names for easy interpretation
new_column_names = {
    "CLIENTNUM": "Client_ID",
    "Attrition_Flag": "Churn_Status",
    "Customer_Age": "Age",
    "Gender": "Gender",
    "Dependent_count": "Dependents",
    "Education_Level": "Education_Level",
    "Marital_Status": "Marital_Status",
    "Income_Category": "Income_Range",
    "Card_Category": "Card_Type",
    "Months_on_book": "Tenure_Months",
    "Total_Relationship_Count": "Product_Count",
    "Months_Inactive_12_mon": "Months_Inactive_12",
    "Contacts_Count_12_mon": "Contacts_Count_12",
    "Credit_Limit": "Credit_Limit",
    "Total_Revolving_Bal": "Revolving_Balance",
    "Avg_Open_To_Buy": "Available_Credit",
    "Total_Amt_Chng_Q4_Q1": "Amt_Change_Q4Q1",
    "Total_Trans_Amt": "Total_Trans_Amount",
    "Total_Trans_Ct": "Total_Trans_Count",
    "Total_Ct_Chng_Q4_Q1": "Trans_Count_Change_Q4Q1",
    "Avg_Utilization_Ratio": "Avg_Utilization_Ratio",
}
# Rename the columns
df.rename(columns=new_column_names, inplace=True)
print(df.columns)
```

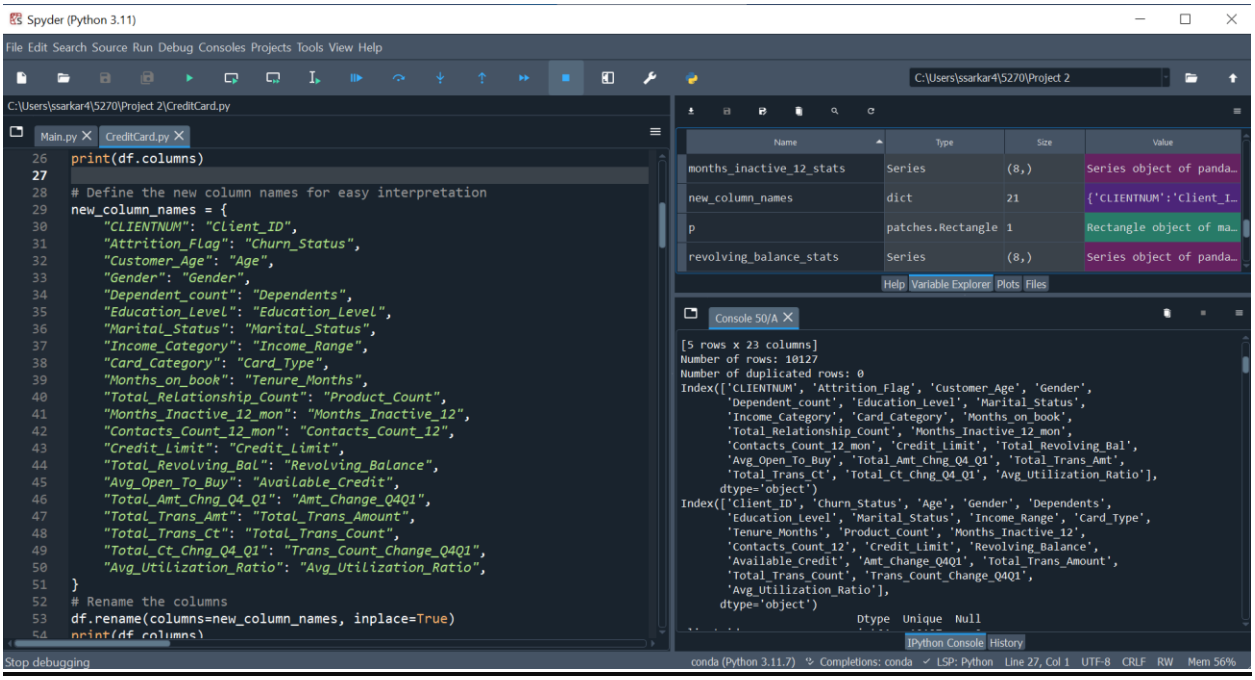
Pre-Cleaning:

```
Index(['CLIENTNUM', 'Attrition_Flag', 'Customer_Age', 'Gender',
       'Dependent_count', 'Education_Level', 'Marital_Status',
       'Income_Category', 'Card_Category', 'Months_on_book',
       'Total_Relationship_Count', 'Months_Inactive_12_mon',
       'Contacts_Count_12_mon', 'Credit_Limit', 'Total_Revolving_Bal',
       'Avg_Open_To_Buy', 'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt',
       'Total_Trans_Ct', 'Total_Ct_Chng_Q4_Q1', 'Avg_Utilization_Ratio'],
      dtype='object')
```

Post Cleaning:

```
Index(['Client_ID', 'Churn_Status', 'Age', 'Gender', 'Dependents',
       'Education_Level', 'Marital_Status', 'Income_Range', 'Card_Type',
       'Tenure_Months', 'Product_Count', 'Months_Inactive_12',
       'Contacts_Count_12', 'Credit_Limit', 'Revolving_Balance',
       'Available_Credit', 'Amt_Change_Q4Q1', 'Total_Trans_Amount',
       'Total_Trans_Count', 'Trans_Count_Change_Q4Q1',
       'Avg_Utilization_Ratio'],
      dtype='object')
```

Full Interface Screenshot:



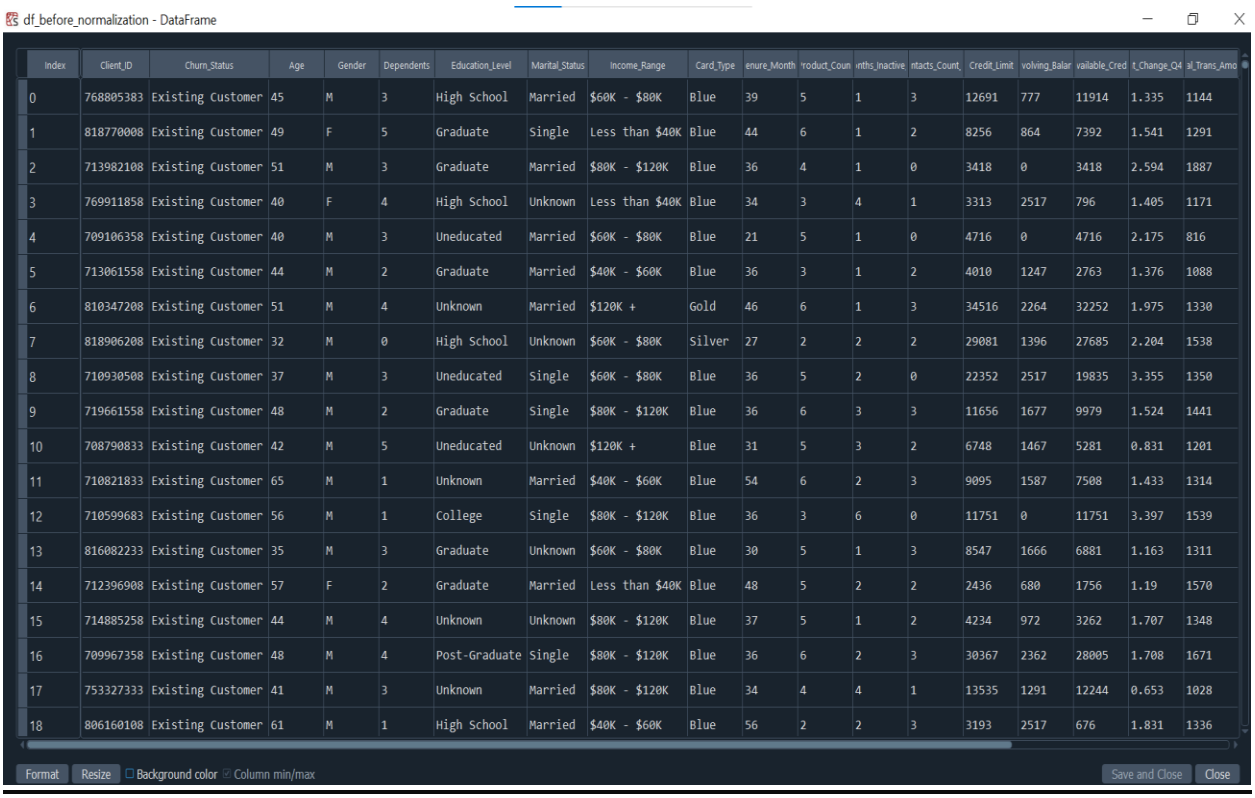
2. Normalization of column name and its values:

By converting all column names to lowercase, we ensure that they're uniform and easier to reference in our analysis. Similarly, by converting all text values to lowercase, we avoid any inconsistencies that might arise from variations in letter case for example ‘male’ and ‘Male’ will be treated as two different things. This standardization process helps maintain the integrity of our data and makes it more straightforward to manipulate and interpret. Ultimately, by ensuring that our dataset follows a consistent format, we lay a solid foundation for conducting effective analysis and drawing meaningful insights.

Code for Data Cleaning-2:

```
#DataCleaning2
# Create a copy of the original dataframe
df_before_normalization = df.copy()

# Function to normalize column names and text values to lowercase
def normalize_text_and_column_names(df):
    # Normalize column names
    df.columns = [col.lower() for col in df.columns]
    # Normalize text values in the DataFrame
    for col in df.select_dtypes(include=['object']):
        df[col] = df[col].str.lower()
    return df

#calling the function
df = normalize_text_and_column_names(df)
```

Pre-Cleaning:



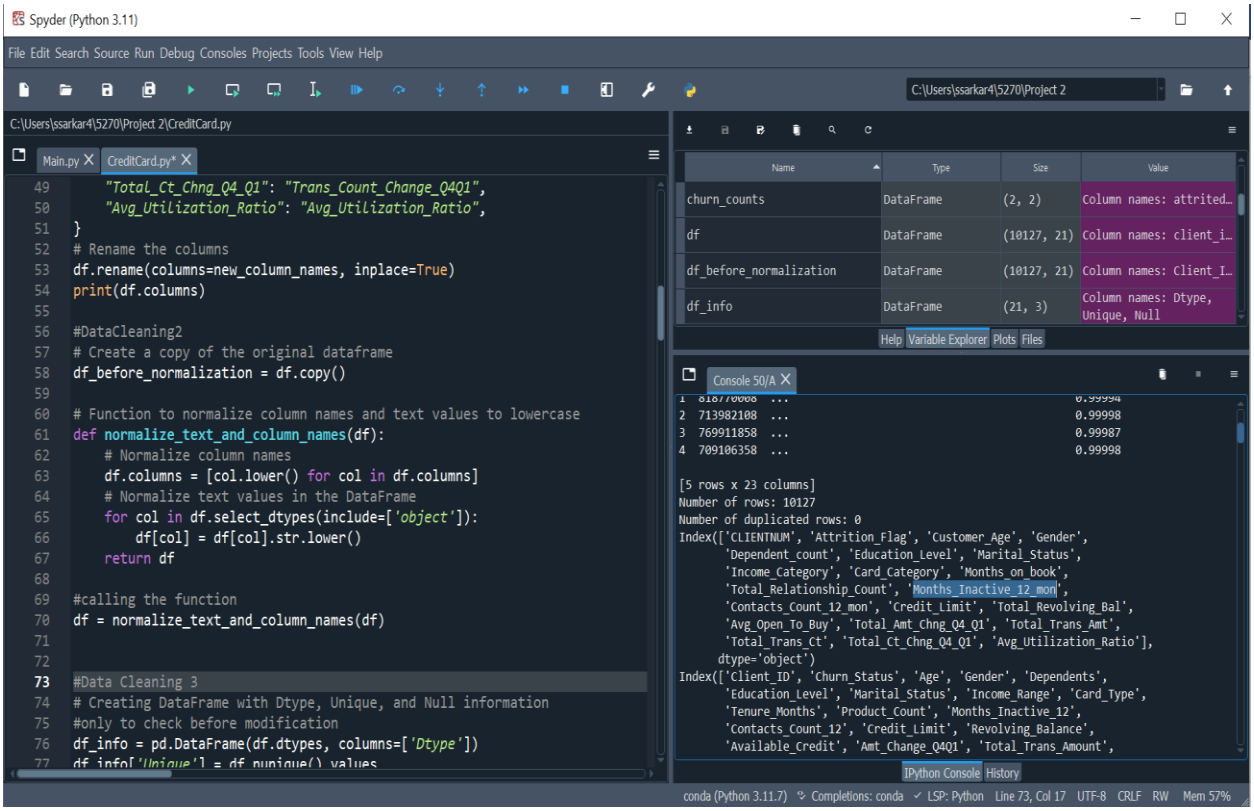
Index	Client_ID	Churn_Status	Age	Gender	Dependents	Education_Level	Marital_Status	Income_Range	Card_Type	enure_Month	roduct_Coun	nths_Inactive	ntacts_Count_	Credit_Limit	volving_Balar	vailable_Cred	t_Change_Q4	al_Trans_Amo
0	768805383	Existing Customer	45	M	3	High School	Married	$60K - $80K	Blue	39	5	1	3	12691	777	11914	1.335	1144
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	44	6	1	2	8256	864	7392	1.541	1291
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K - $120K	Blue	36	4	1	0	3418	0	3418	2.594	1887
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	34	3	4	1	3313	2517	796	1.405	1171
4	709106358	Existing Customer	40	M	3	Uneducated	Married	$60K - $80K	Blue	21	5	1	0	4716	0	4716	2.175	816
5	713061558	Existing Customer	44	M	2	Graduate	Married	$40K - $60K	Blue	36	3	1	2	4010	1247	2763	1.376	1088
6	810347208	Existing Customer	51	M	4	Unknown	Married	$120K +	Gold	46	6	1	3	34516	2264	32252	1.975	1330
7	818906208	Existing Customer	32	M	0	High School	Unknown	$60K - $80K	Silver	27	2	2	2	29081	1396	27685	2.204	1538
8	710930508	Existing Customer	37	M	3	Uneducated	Single	$60K - $80K	Blue	36	5	2	0	22352	2517	19835	3.355	1350
9	719661558	Existing Customer	48	M	2	Graduate	Single	$80K - $120K	Blue	36	6	3	3	11656	1677	9979	1.524	1441
10	708790833	Existing Customer	42	M	5	Uneducated	Unknown	$120K +	Blue	31	5	3	2	6748	1467	5281	0.831	1201
11	710821833	Existing Customer	65	M	1	Unknown	Married	$40K - $60K	Blue	54	6	2	3	9095	1587	7508	1.433	1314
12	710599683	Existing Customer	56	M	1	College	Single	$80K - $120K	Blue	36	3	6	0	11751	0	11751	3.397	1539
13	816082233	Existing Customer	35	M	3	Graduate	Unknown	$60K - $80K	Blue	30	5	1	3	8547	1666	6881	1.163	1311
14	712396908	Existing Customer	57	F	2	Graduate	Married	Less than $40K	Blue	48	5	2	2	2436	680	1756	1.19	1570
15	714885258	Existing Customer	44	M	4	Unknown	Unknown	$80K - $120K	Blue	37	5	1	2	4234	972	3262	1.707	1348
16	709967358	Existing Customer	48	M	4	Post-Graduate	Single	$80K - $120K	Blue	36	6	2	3	30367	2362	28005	1.708	1671
17	753327333	Existing Customer	41	M	3	Unknown	Married	$80K - $120K	Blue	34	4	4	1	13535	1291	12244	0.653	1028
18	806160108	Existing Customer	61	M	1	High School	Married	$40K - $60K	Blue	56	2	2	3	3193	2517	676	1.831	1336

Post Cleaning:

df - DataFrame

index	client_id	churn_status	age	gender	dependents	education_level	marital_status	income_range	card_type	enure_month	roduct_coun	nths_inactive_	ntacts_count_	credit_limit	volving_balan	vailable_cred	t_change_q4	al_trans_amo
0	768805383	existing customer	45	m	3	high school	married	$60k - $80k	blue	39	5	1	3	12691	777	11914	1.335	1144
1	818770008	existing customer	49	f	5	graduate	single	less than $40k	blue	44	6	1	2	8256	864	7392	1.541	1291
2	713982108	existing customer	51	m	3	graduate	married	$80k - $120k	blue	36	4	1	0	3418	0	3418	2.594	1887
3	769911858	existing customer	40	f	4	high school	unknown	less than $40k	blue	34	3	4	1	3313	2517	796	1.405	1171
4	709106358	existing customer	40	m	3	uneducated	married	$60k - $80k	blue	21	5	1	0	4716	0	4716	2.175	816
5	713061558	existing customer	44	m	2	graduate	married	$40k - $60k	blue	36	3	1	2	4010	1247	2763	1.376	1088
6	810347208	existing customer	51	m	4	graduate	married	$120k +	gold	46	6	1	3	34516	2264	32252	1.975	1330
7	818906208	existing customer	32	m	0	high school	unknown	$60k - $80k	silver	27	2	2	2	29081	1396	27685	2.204	1538
8	710930508	existing customer	37	m	3	uneducated	single	$60k - $80k	blue	36	5	2	0	22352	2517	19835	3.355	1350
9	719661558	existing customer	48	m	2	graduate	single	$80k - $120k	blue	36	6	3	3	11656	1677	9979	1.524	1441
10	708790833	existing customer	42	m	5	uneducated	unknown	$120k +	blue	31	5	3	2	6748	1467	5281	0.831	1201
11	710821833	existing customer	65	m	1	graduate	married	$40k - $60k	blue	54	6	2	3	9095	1587	7508	1.433	1314
12	710599683	existing customer	56	m	1	college	single	$80k - $120k	blue	36	3	6	0	11751	0	11751	3.397	1539
13	816082233	existing customer	35	m	3	graduate	unknown	$60k - $80k	blue	30	5	1	3	8547	1666	6881	1.163	1311
14	712396908	existing customer	57	f	2	graduate	married	less than $40k	blue	48	5	2	2	2436	680	1756	1.19	1570
15	714885258	existing customer	44	m	4	graduate	unknown	$80k - $120k	blue	37	5	1	2	4234	972	3262	1.707	1348
16	709967358	existing customer	48	m	4	post-graduate	single	$80k - $120k	blue	36	6	2	3	30367	2362	28005	1.708	1671
17	753327333	existing customer	41	m	3	uneducated	married	$80k - $120k	blue	34	4	4	1	13535	1291	12244	0.653	1028
18	806160108	existing customer	61	m	1	high school	married	$40k - $60k	blue	56	2	2	3	3193	2517	676	1.831	1336

Full Interface Screenshot:



3. **Converting Data Types:**

   By changing columns like 'churn_status', 'gender', 'education_level', 'marital_status', 'income_range', and 'card_type' into categories, we've organized our data better for making cool visuals. This not only makes our graphs look nicer but also helps us show important stuff from our data. Plus, it helps our computer use less memory, which means we can analyze data faster. So, by using categories, we make our data analysis smoother and our visualizations more awesome.By changing columns like 'churn_status', 'gender', 'education_level', 'marital_status', 'income_range', and 'card_type' into categories, we have organized our data better for making cool visuals. This not only makes our graphs look nicer but also helps us show important stuff from our data. Plus, it helps our computer use less memory, which means we can analyze data faster. So, by using categories, we make our data analysis smoother and our visualizations more awesome.

Code for Data Cleaning-3:

```
#Data Cleaning 3
# Creating DataFrame with Dtype, Unique, and Null information
#only to check before modification
df_info = pd.DataFrame(df.dtypes, columns=['Dtype'])
df_info['Unique'] = df.nunique().values
df_info['Null'] = df.isnull().sum().values
# Displaying the DataFrame
print(df_info)

# Convert object columns to category data type
df['churn_status'] = df['churn_status'].astype('category')
df['gender'] = df['gender'].astype('category')
df['education_level'] = df['education_level'].astype('category')
df['marital_status'] = df['marital_status'].astype('category')
df['income_range'] = df['income_range'].astype('category')
df['card_type'] = df['card_type'].astype('category')

# Creating DataFrame with Dtype, Unique, and Null information
#only to check after modification
df_info = pd.DataFrame(df.dtypes, columns=['Dtype'])
df_info['Unique'] = df.nunique().values
df_info['Null'] = df.isnull().sum().values
# Displaying the DataFrame
print(df_info)

```

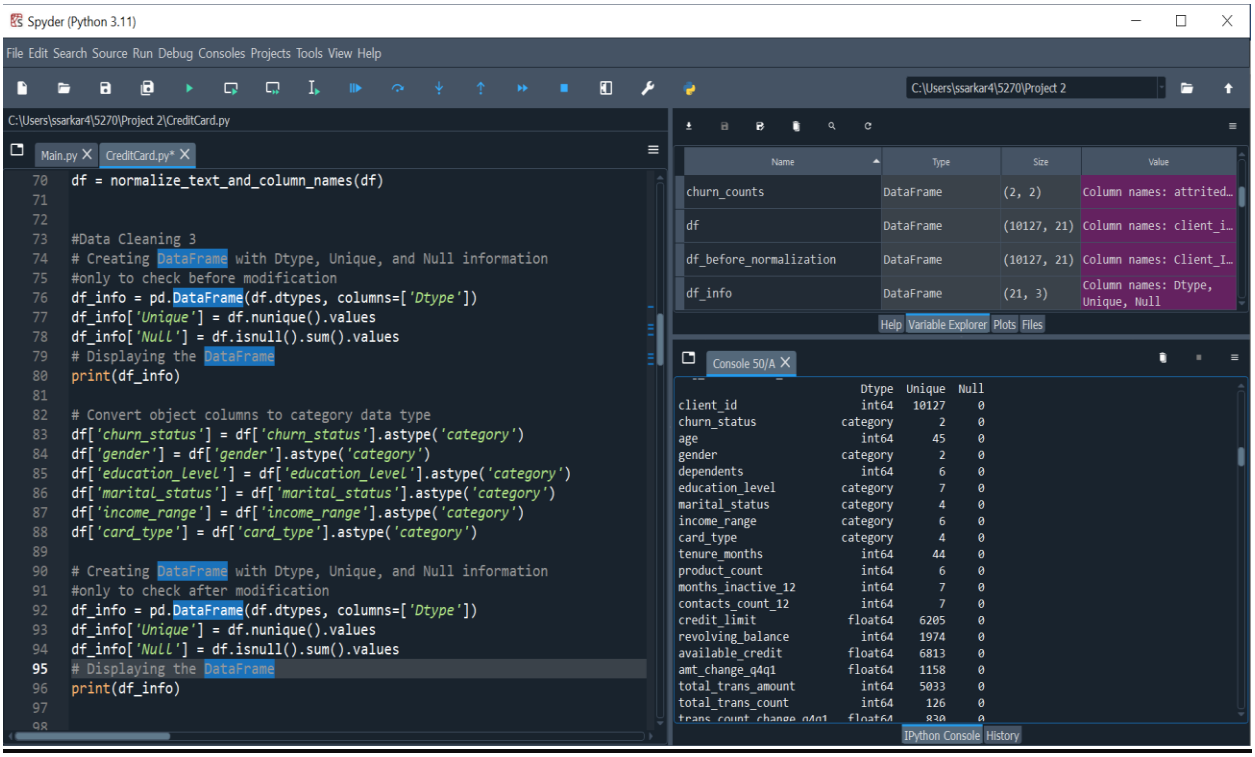
Pre-Cleaning:

	Dtype	Unique	Null
client_id	int64	10127	0
churn_status	object	2	0
age	int64	45	0
gender	object	2	0
dependents	int64	6	0
education_level	object	7	0
marital_status	object	4	0
income_range	object	6	0
card_type	object	4	0
tenure_months	int64	44	0
product_count	int64	6	0
months_inactive_12	int64	7	0
contacts_count_12	int64	7	0
credit_limit	float64	6205	0
revolving_balance	int64	1974	0
available_credit	float64	6813	0
amt_change_q4q1	float64	1158	0
total_trans_amount	int64	5033	0
total_trans_count	int64	126	0
trans_count_change_q4q1	float64	830	0

Post Cleaning:

	Dtype	Unique	Null
client_id	int64	10127	0
churn_status	category	2	0
age	int64	45	0
gender	category	2	0
dependents	int64	6	0
education_level	category	7	0
marital_status	category	4	0
income_range	category	6	0
card_type	category	4	0
tenure_months	int64	44	0
product_count	int64	6	0
months_inactive_12	int64	7	0
contacts_count_12	int64	7	0
credit_limit	float64	6205	0
revolving_balance	int64	1974	0
available_credit	float64	6813	0
amt_change_q4q1	float64	1158	0
total_trans_amount	int64	5033	0
total_trans_count	int64	126	0
trans_count_change_q4q1	float64	830	0

Full Interface Screenshot:



4. Imputing Missing Values:

We recognized that some categories were marked as 'unknown,' potentially distorting our analysis or visuals. To address this, we proportionally increased the counts of other categories to fill in for these 'unknown' values. This ensured that our dataset remained balanced and representative without introducing any bias. This step was critical because when we later create charts or analyze the data, we want to ensure that we are working with accurate information that truly reflects the underlying trends.

Code for Data Cleaning-4:

```
# Data Cleaning 4
# Print the counts of unique values in the 'Income_Range' column before imputation
income_range_counts_before = df['income_range'].value_counts()
print("Income Range Counts before imputation:")
print(income_range_counts_before)

# Print the counts of unique values in the 'Education_Level' column before imputation
education_level_counts_before = df['education_level'].value_counts()
print("Education Level Counts before imputation:")
print(education_level_counts_before)

# Calculate the distribution of income categories (excluding 'Unknown' values)
income_range_distribution = df[df['income_range'] != 'unknown']['income_range'].value_counts(normalize=True)

# Calculate the distribution of education levels (excluding 'Unknown' values)
education_level_distribution = df[df['education_level'] != 'unknown']['education_level'].value_counts(normalize=True)

# Impute missing values based on the distribution
def impute_missing_categories(row, distribution, column):
    if row[column] == 'unknown':
        return np.random.choice(distribution.index, p=distribution.values)
    else:
        return row[column]

# Apply the imputation function to the 'Income_Range' column
df['income_range'] = df.apply(lambda row: impute_missing_categories(row, income_range_distribution, 'income_range'), axis=1)
```

```

# Apply the imputation function to the 'Education_Level' column
df['education_level'] = df.apply(lambda row: impute_missing_categories(row, education_level_distribution, 'education_level'),

# Count the occurrences of each unique value in the 'Income_Range' column after imputation
income_range_counts_after = df['income_range'].value_counts()

# Print the counts of unique values in the 'Income_Range' column after imputation
print("Income Range Counts after imputation:")
print(income_range_counts_after)

# Count the occurrences of each unique value in the 'Education_Level' column after imputation
education_level_counts_after = df['education_level'].value_counts()

# Print the counts of unique values in the 'Education_Level' column after imputation
print("Education Level Counts after imputation:")
print(education_level_counts_after)
```

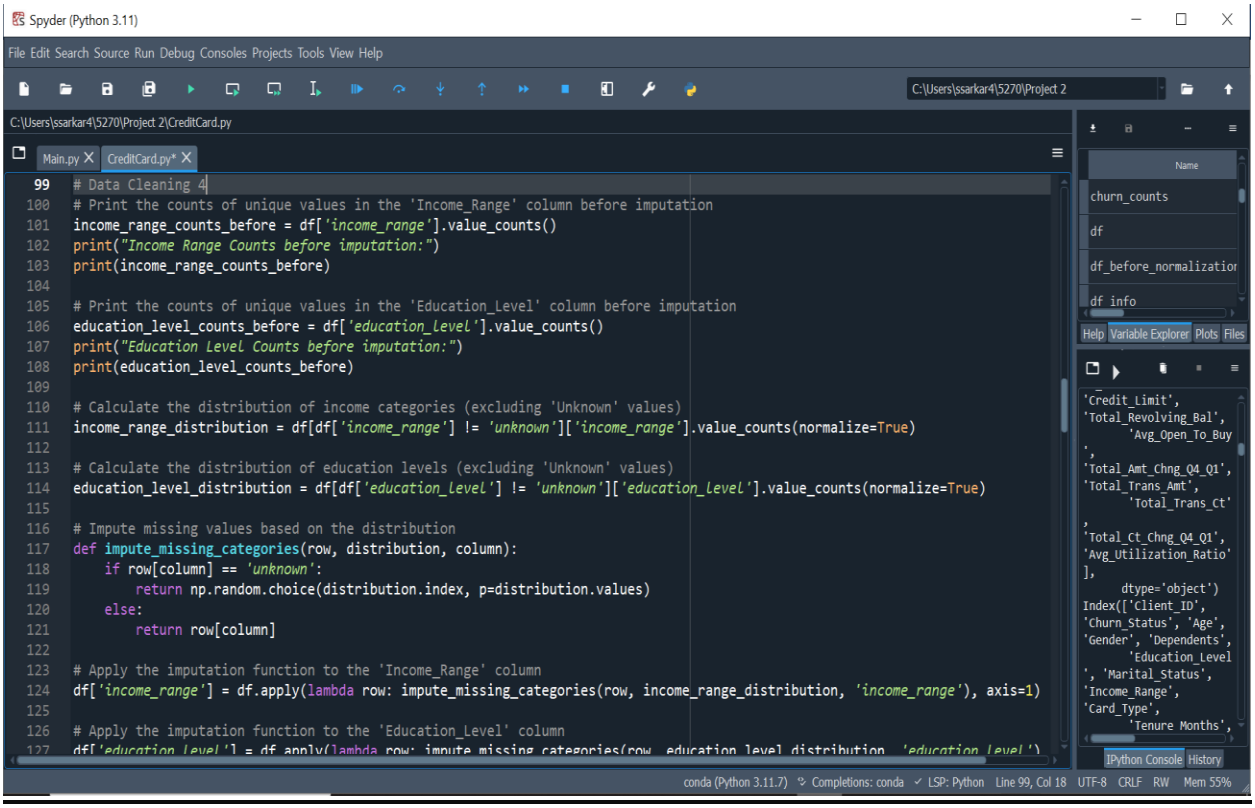
Pre-Cleaning:

```
Income Range Counts before imputation:
income_range
less than $40k     3561
$40k - $60k        1790
$80k - $120k       1535
$60k - $80k        1402
unknown            1112
$120k +             727
Name: count, dtype: int64
Education Level Counts before imputation:
education_level
graduate          3128
high school       2013
unknown           1519
uneducated        1487
college           1013
post-graduate      516
doctorate          451
```

Post Cleaning:

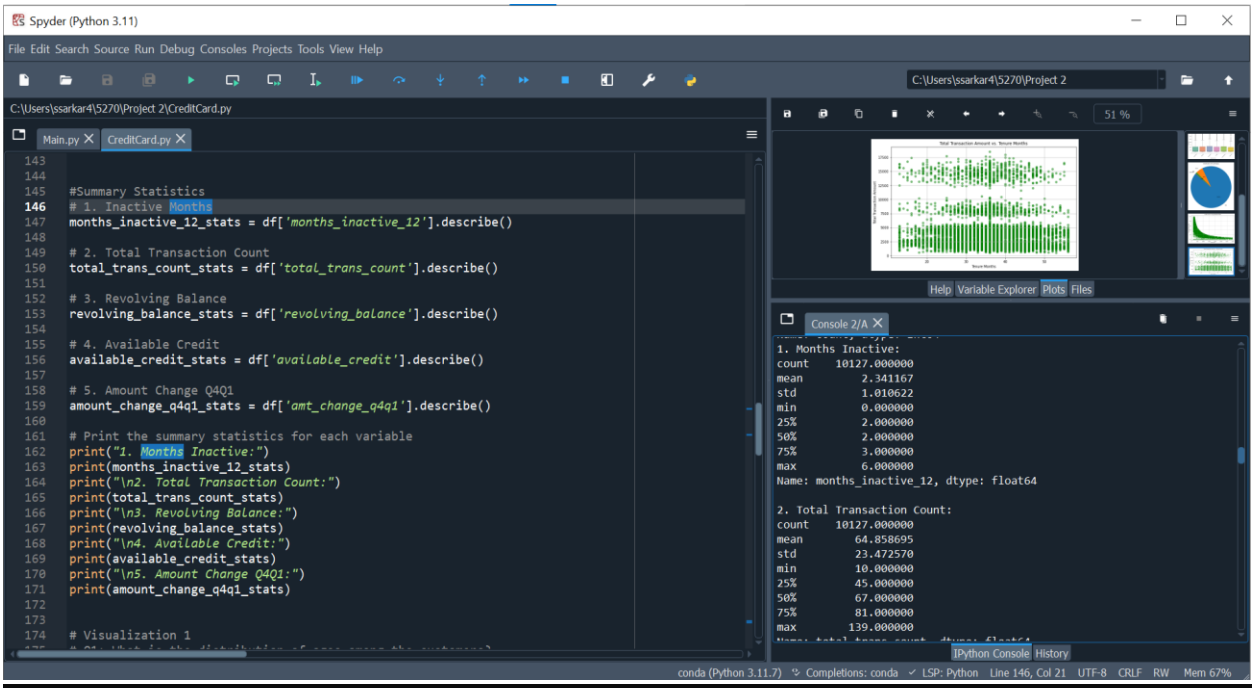
```
Income Range Counts after imputation:
income_range
less than $40k    4016
$40k - $60k       1998
$80k - $120k      1718
$60k - $80k       1574
$120k +            821
Name: count, dtype: int64
Education Level Counts after imputation:
education_level
graduate          3692
high school       2380
uneducated        1729
college           1179
post-graduate      615
doctorate          532
```

Full Interface Screenshot:



Summary Statistic:

Full Interface Screenshot:



1. **Months Inactive 12:** Exploring how long customers stayed inactive in the last year gives us important insights into their behavior. On average, customers did not use their accounts for about 2.3 months. But this average is not the whole story, there is also a lot of variation in how long customers were inactive, as shown by the standard deviation of about 1.01. This means that some customers were inactive for much longer periods than others. Understanding these differences is crucial for figuring out why customers are not using their accounts and how we can keep them engaged. Additionally, the median duration of inactivity, which is 2 months, gives us a clear idea of the typical length of time customers stay inactive. This knowledge helps us develop strategies to retain customers and keep them actively using their accounts.

```
1. Months Inactive:
count    10127.000000
mean         2.341167
std          1.010622
min          0.000000
25%          2.000000
50%          2.000000
75%          3.000000
max          6.000000
Name: months_inactive_12, dtype: float64
```

2. **Total Transaction Count:** The 'Total Transaction Count' parameter offers valuable insights into customer behavior regarding credit card usage. With an average transaction count of approximately 64, we observe considerable variability, as indicated by the standard deviation of around 23. This variability underscores diverse transaction patterns among customers. For instance, the median transaction count of 67 signifies the midpoint of the dataset, suggesting that half of the customers conducted more than 67 transactions, while the other half engaged in fewer transactions. These statistical metrics provide a detailed understanding of customer engagement levels and serve as a foundation for tailored strategies aimed at enhancing customer satisfaction and retention.

```
2. Total Transaction Count:
count    10127.000000
mean        64.858695
std         23.472570
min         10.000000
25%         45.000000
50%         67.000000
75%         81.000000
max        139.000000
Name: total_trans_count, dtype: float64
```

3. **Revolving Balance:** On average, customers maintain a revolving balance of approximately $1162.81, which signifies the amount of unpaid credit carried forward from one billing cycle to the next. However, the standard deviation of $814.99 indicates significant variability in the extent of revolving balances among customers, highlighting diverse credit utilization behaviors. Notably, the median revolving balance stands at $1276.00, offering a clear indication of the central tendency of unpaid credit across our customer base. By looking at these numbers, we can see how customers use their credit cards. This helps us figure out patterns and make plans to help customers manage their balances better.

```
3. Revolving Balance:
count    10127.000000
mean      1162.814061
std        814.987335
min          0.000000
25%        359.000000
50%       1276.000000
75%       1784.000000
max       2517.000000
Name: revolving_balance, dtype: float64
```

4. **Available Credit:** Understanding customers' financial flexibility and credit utilization is pivotal for effective financial management. On average, customers possess an available credit of approximately $7469.14, signifying their remaining credit limit for further expenditures. However, this average is accompanied by a notable standard deviation of $9090.69, indicating significant variability in available credit levels among customers. This variability reflects differences in creditworthiness and spending capacity. Additionally, the median available credit, at $3474.00, provides a clearer picture of the typical available credit amount experienced by our clientele. These insights allow us to

assess customers' financial health and tailor strategies to optimize credit management and financial well-being.

```
4. Available Credit:
count    10127.000000
mean      7469.139637
std       9090.685324
min          3.000000
25%       1324.500000
50%       3474.000000
75%       9859.000000
max      34516.000000
Name: available_credit, dtype: float64
```

5. **Amount Change Q4Q1:** The 'Amount Change Q4Q1' metric provides insights into how customers' spending behavior changes from one quarter to the next. On average, customers experience an amount change of approximately 0.76, indicating a moderate increase in spending between quarters. However, the standard deviation of 0.22 suggests variability in spending changes among customers, with some experiencing larger fluctuations than others. The median amount change, at 0.736, offers additional context, representing the typical spending change experienced by our customers. Understanding these statistics enables us to assess spending trends and tailor strategies to address changing customer needs and preferences.

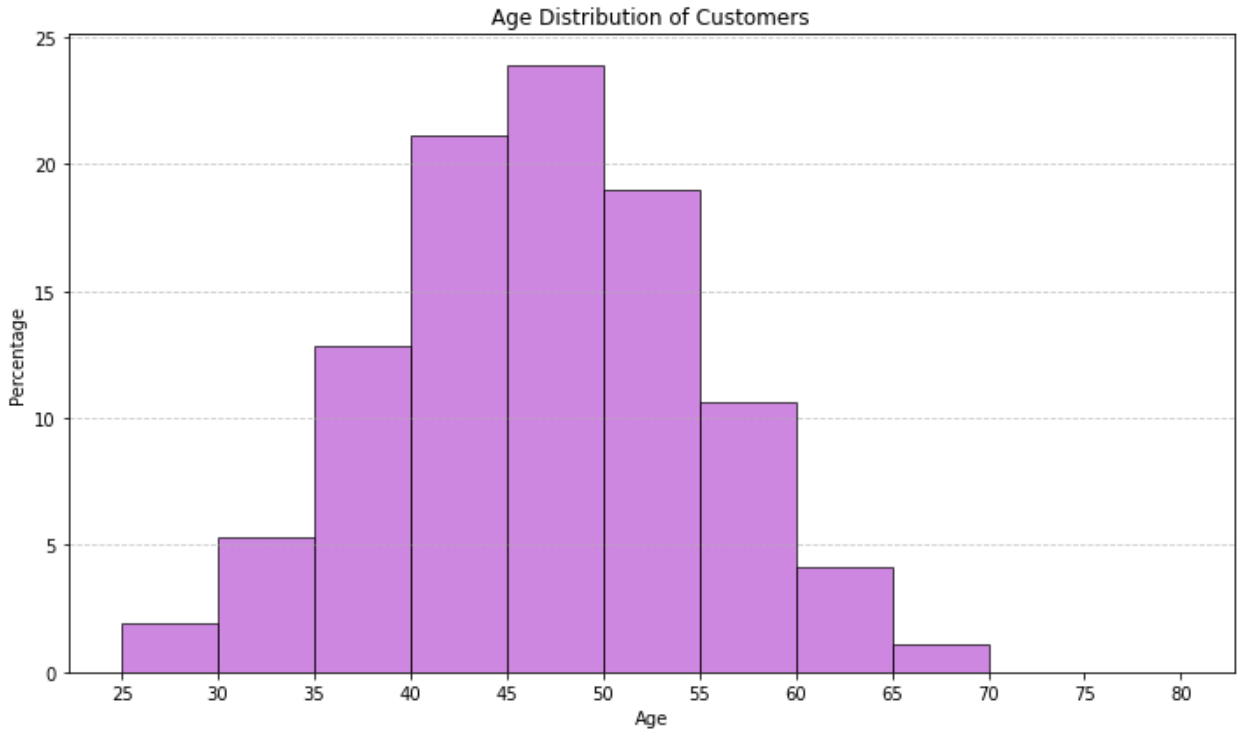
```
5. Amount Change Q4Q1:
count    10127.000000
mean         0.759941
std          0.219207
min          0.000000
25%          0.631000
50%          0.736000
75%          0.859000
max          3.397000
Name: amt_change_q4q1, dtype: float64
```

Through a meticulous examination of these summary statistics, we gain profound insights into customer behavior and financial dynamics within our credit card dataset. These insights serve as a compass, guiding strategic decisions aimed at enhancing customer satisfaction, optimizing product offerings, and driving business growth.

Analysis & Visualization:

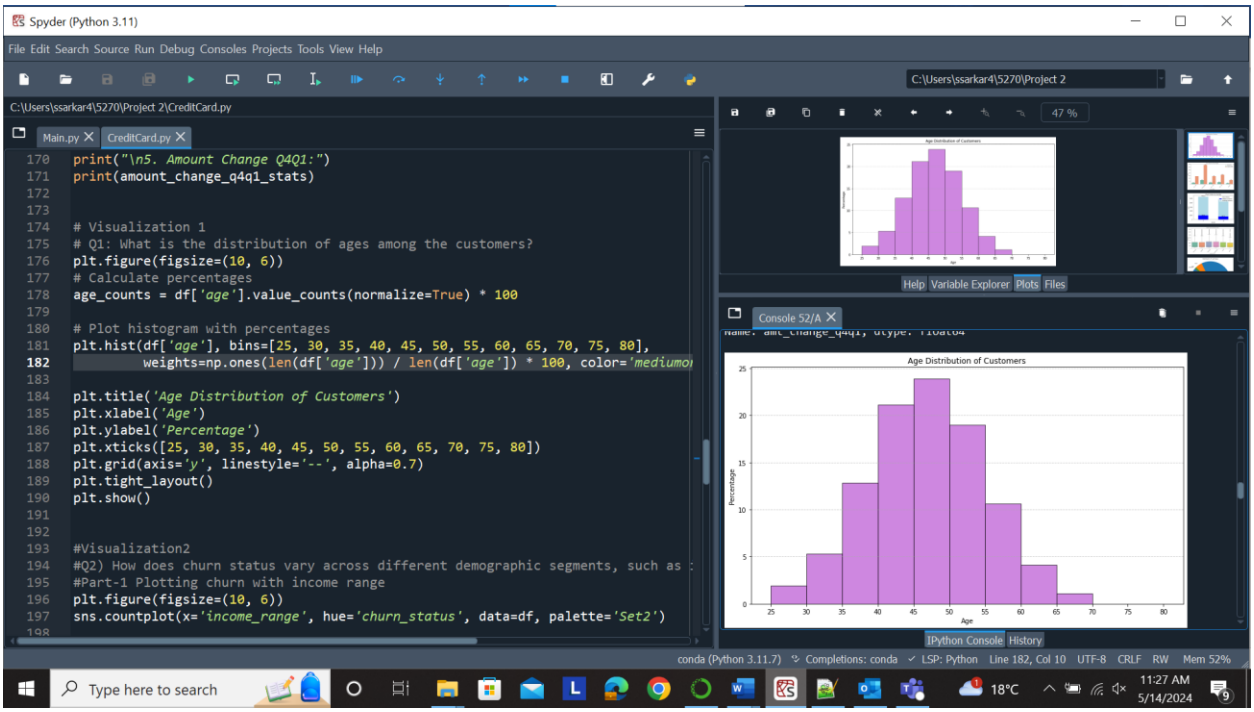
Analysis Question 1: What is the distribution of ages among the customers?

```
# Visualization 1
# Q1: What is the distribution of ages among the customers?
plt.figure(figsize=(10, 6))
# Calculate percentages
age_counts = df['age'].value_counts(normalize=True) * 100

# Plot histogram with percentages
plt.hist(df['age'], bins=[25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80],
         weights=np.ones(len(df['age'])) / len(df['age']) * 100, color='mediumorchid', edgecolor='black', alpha=0.7)

plt.title('Age Distribution of Customers')
plt.xlabel('Age')
plt.ylabel('Percentage')
plt.xticks([25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80])
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



This above histogram indicates that customers belonging to the age group of 45-50 hold the highest number of credit cards among all, followed by the age groups of 40-45 and 50-55, while those who fall into the age group of 65-70 have the lowest number. These findings suggest that middle-aged individuals form a significant customer base for credit card services. On the other hand, the youngest age group of 25-30 represents one of the smallest numbers of credit card holders. This lower percentage indicates that younger individuals may have a relatively lower propensity for credit card usage. By studying the percentage of credit cards held by each age group, companies can identify trends and preferences. This analysis is invaluable for informing marketing strategies, product development, and customer retention efforts. They could also develop new card features or services targeted towards specific age groups, enhancing overall customer satisfaction.

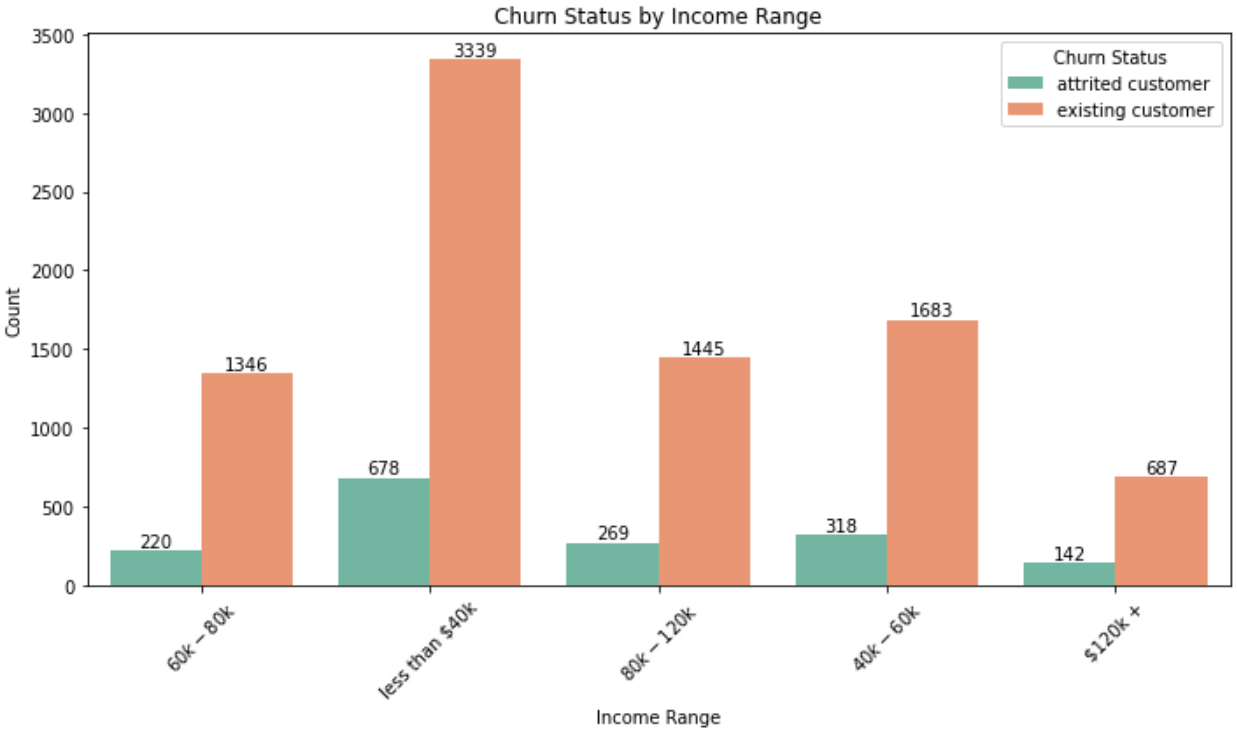
Full Interface Screenshot:

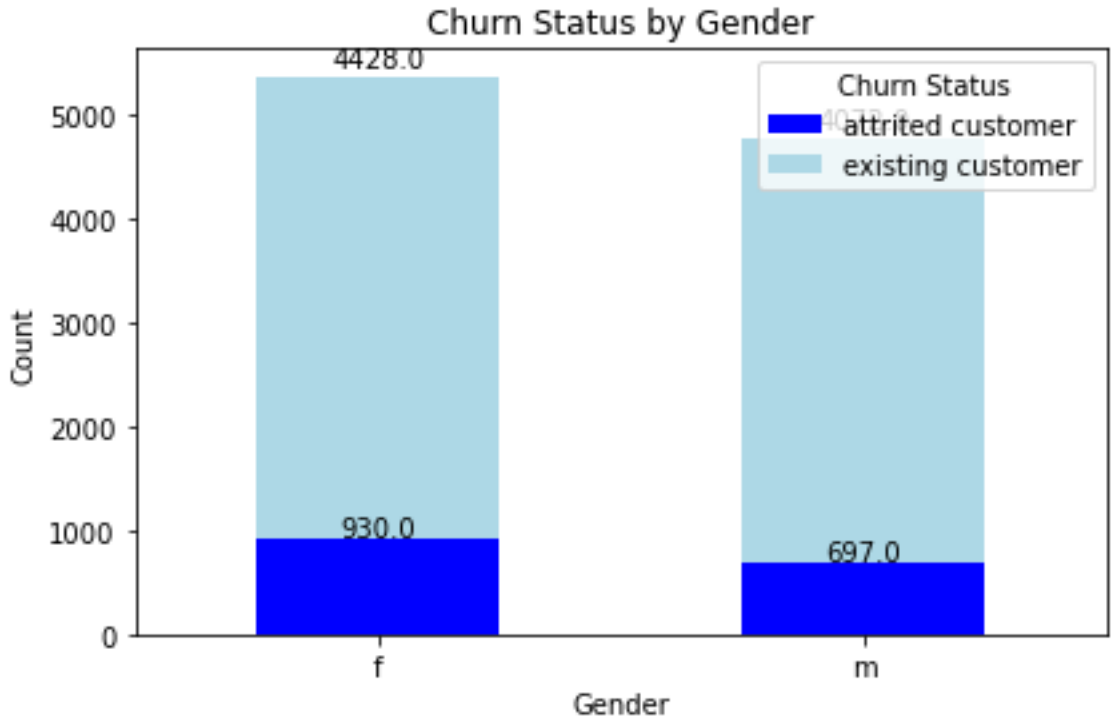


Analysis Question 2: How does churn status vary across different demographic segments, such as income range and gender?

```
#Visualization2
#Q2) How does churn status vary across different demographic segments, such as income range and gender?
#Part-1 Plotting churn with income range
plt.figure(figsize=(10, 6))
sns.countplot(x='income_range', hue='churn_status', data=df, palette='Set2')

# Adding values on top of each bar
for p in plt.gca().patches:
    height = p.get_height()
    plt.gca().text(p.get_x() + p.get_width() / 2, height + 3, int(height), ha='center', va='bottom')

plt.title('Churn Status by Income Range')
plt.xlabel('Income Range')
plt.ylabel('Count')
plt.legend(title='Churn Status')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
#Part2 Plotting churn with gender
# Group data by gender and churn status, and calculate churn counts
churn_counts = df.groupby(['gender', 'churn_status']).size().unstack()

# Plotting the grouped bar chart
plt.figure(figsize=(10, 6))
ax = churn_counts.plot(kind='bar', stacked=True, color=['blue', 'lightblue'])
plt.title('Churn Status by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Churn Status')

# Adding text annotations for counts on top of each bar
for p in ax.patches:
    width = p.get_width()
    height = p.get_height()
    x, y = p.get_xy()
    ax.annotate(f'{height}', (x + width/2, y + height*1.02), ha='center')

plt.tight_layout()
plt.show()

```



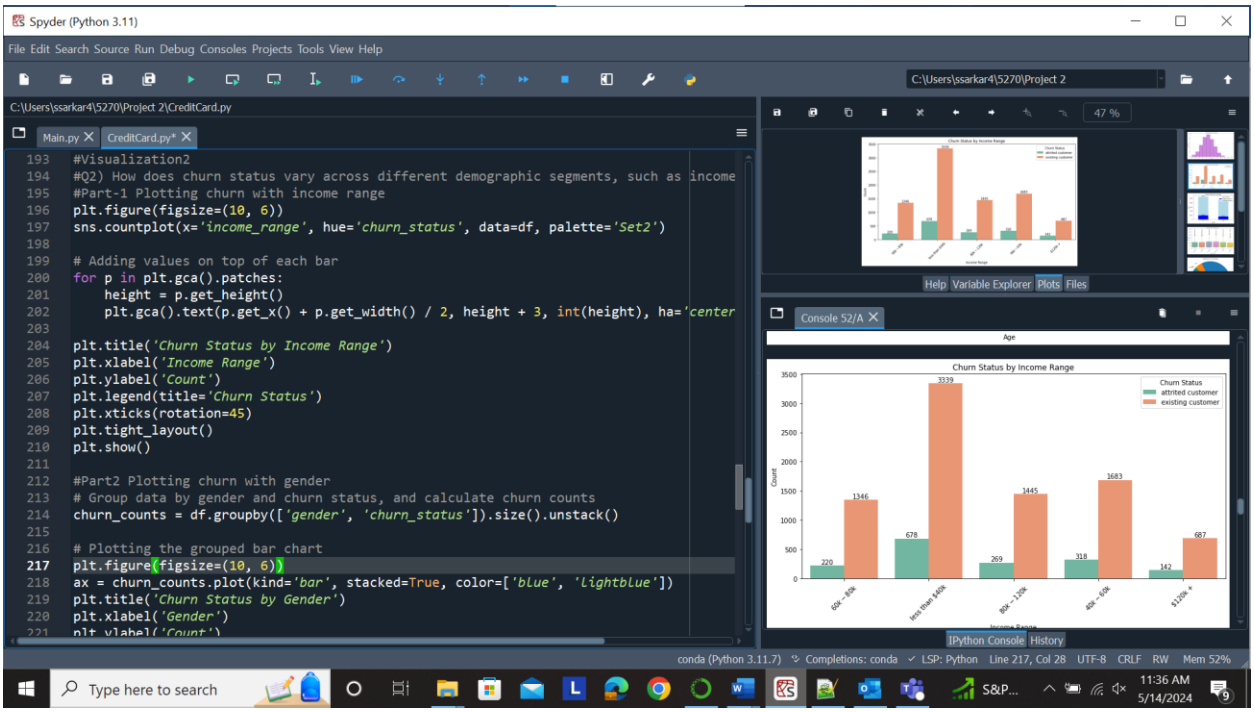


The first visualization suggests that the highest number of customers who churned their credit cards come from the income range of under $40k. Therefore, it can be assumed that

customers in the lowest income brackets are more prone to churning due to financial constraints or dissatisfaction with the services. On the other hand, at the extreme opposite, customers with the highest income range of more than $120k are less likely to churn with this product. In the second visualization, upon comparing by gender, it is evident that a higher number of female customers (17% of total female customers) preferred to discontinue using this product compared to male customers (14% of total male customers).

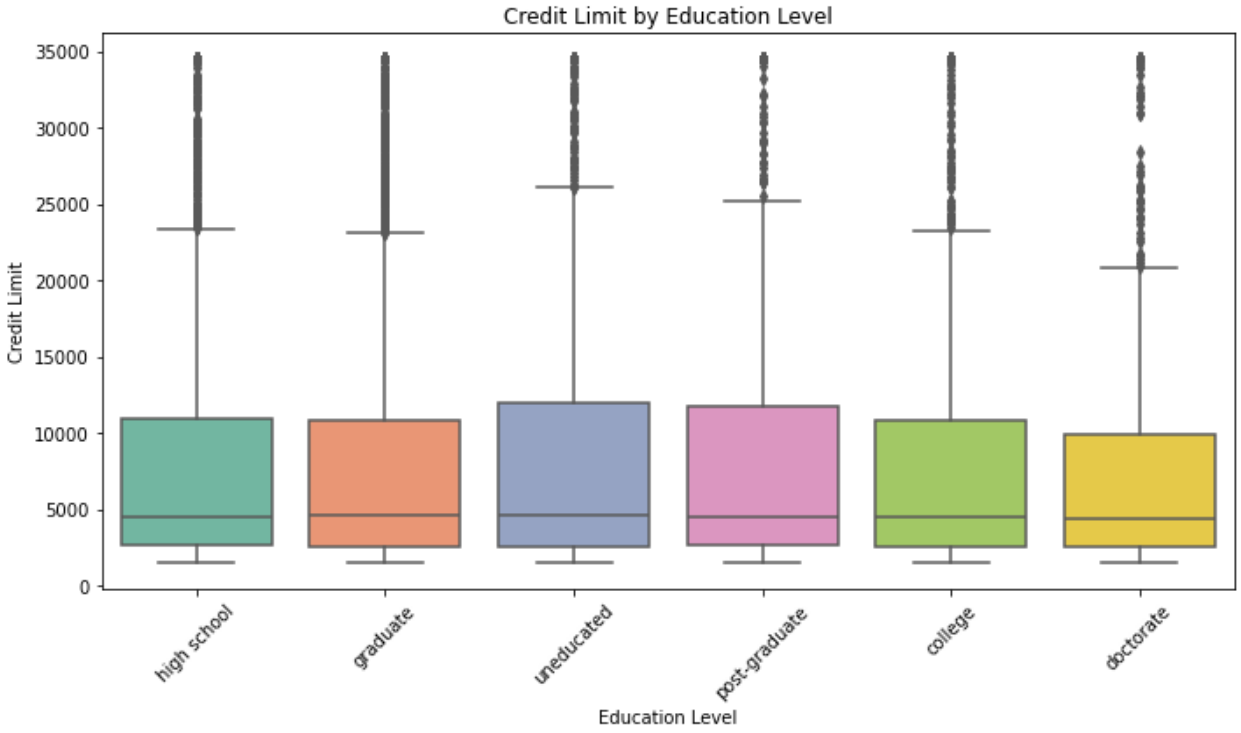
Insights from the analysis, credit card companies can develop proactive retention strategies aimed at reducing churn rates among high-value customers. For example, they can offer personalized rewards, perks, or upgraded services to incentivize loyal customers to stay. They can design their product and service innovations tailored to different income segments and also optimize their marketing efforts.

Full Interface Screenshot:



Analysis Question 3: How does the distribution of credit limits vary across different education levels?

```
#Visulization3
#Q3) How does the distribution of credit limits vary across different education levels?
# Plotting the box plot
plt.figure(figsize=(10, 6))
sns.boxplot(x='education_level', y='credit_limit', data=df, palette='Set2')
plt.title('Credit Limit by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Credit Limit')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

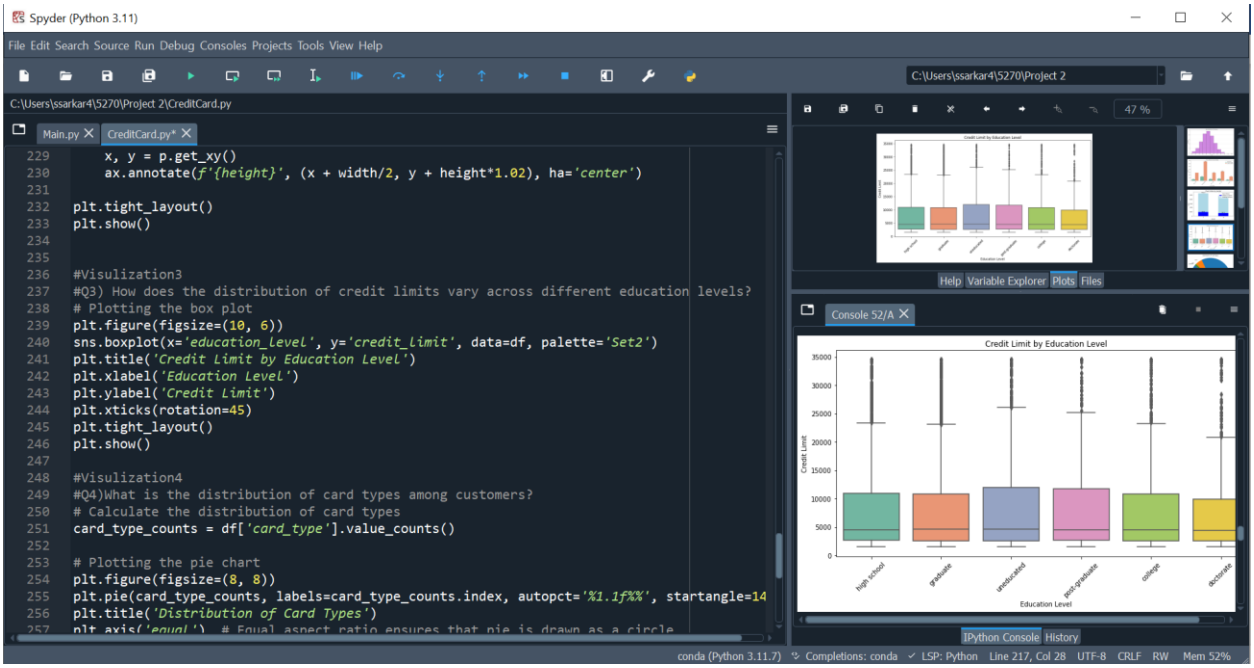


This box whisker plot shows the credit limit by education level. The credit limit ranges from $0 to $35,000. The box in the boxplot represents the middle 50% of the data. The line in the middle of the box represents the median credit limit. The whiskers extend from the top and bottom of the box to the lowest and highest values within 1.5 times the interquartile range (IQR) from the quartiles. Values outside the whiskers are considered outliers.

Based on the plot, we can see that the median credit limit increases as the education level increases. The IQR is also larger for higher education levels, which means that the data is more spread out for those groups. There are also outliers for the graduate, post-graduate, and doctorate education levels. The spread of credit limits is the smallest for the uneducated group and largest for the doctorate group. There is a larger gap between the median credit limit for the high school group and the graduate group than there is between any other two groups.

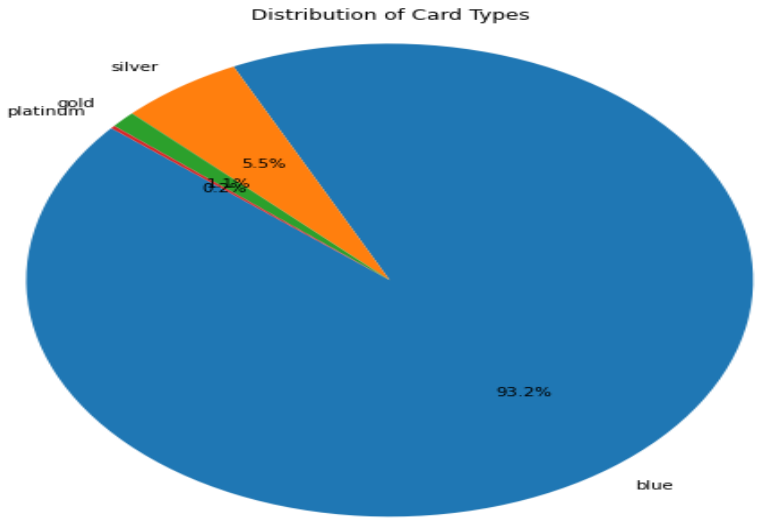
The analysis can help assess the risk associated with issuing credit cards to different customer segments. Customers with lower education levels and credit limits may be deemed higher risk, prompting the credit card company to offer them lower limits or higher interest rates. Develop new credit card products specifically designed with benefits tailored to the needs of different groups. Outliers in the data could indicate fraudulent activity, so control them by adjusting existing customer benefits.

Full Interface Screenshot:



Analysis Question 4: What is the distribution of card types among customers?

```
#Visulization4
#Q4)What is the distribution of card types among customers?
# Calculate the distribution of card types
card_type_counts = df['card_type'].value_counts()

# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(card_type_counts, labels=card_type_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Distribution of Card Types')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle
plt.show()
```



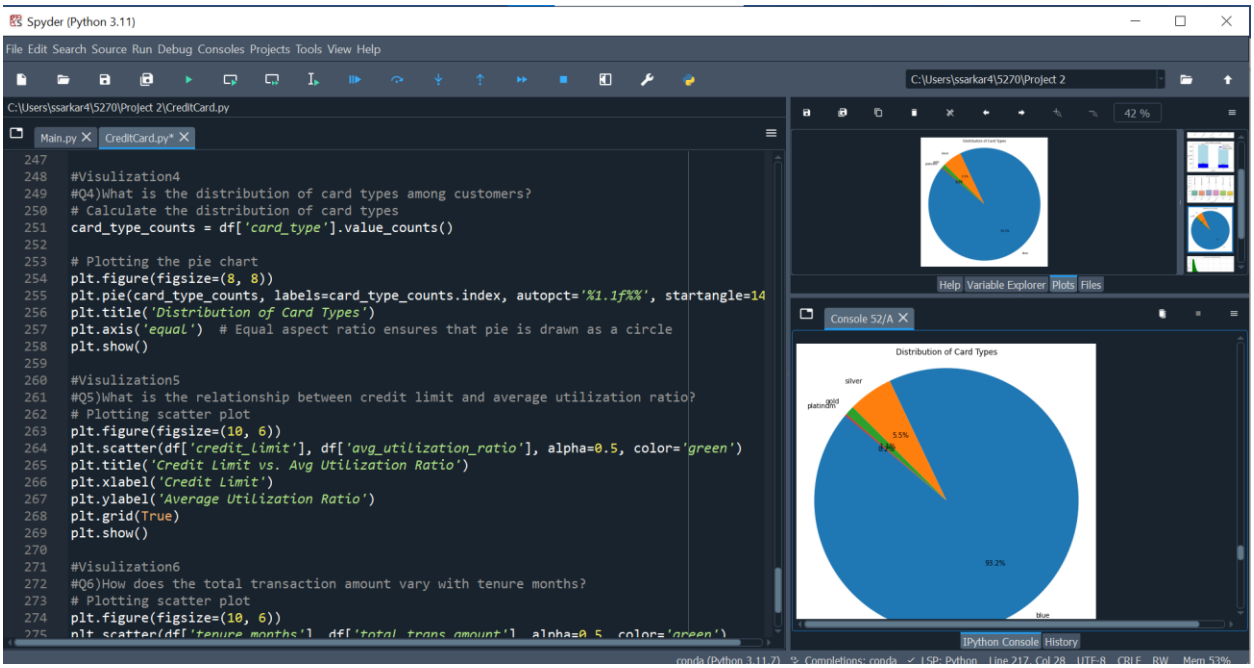
The pie chart shows the types of cards that customers have. Each part of the pie is a different card type, and its size shows how many customers have that type of card. By looking at the sizes, we can see which cards are most popular. This helps banks understand what customers like and make cards that fit their needs.

The Highlight Table gives more details. It shows that most people have the 'Blue' card, with 9436 holders, making up 93.18% of all cardholders. This suggests that the 'Blue' card might be the most popular because it's easy to get or offers good benefits. The 'Silver' card is next, but it is less common, with only 5.48% of cardholders. This might mean it's a fancier or more specialized card.

'Gold' and 'Platinum' cards are even rarer, with just 1.15% and 0.20% of cardholders respectively. These cards are probably for wealthier customers who want special perks.

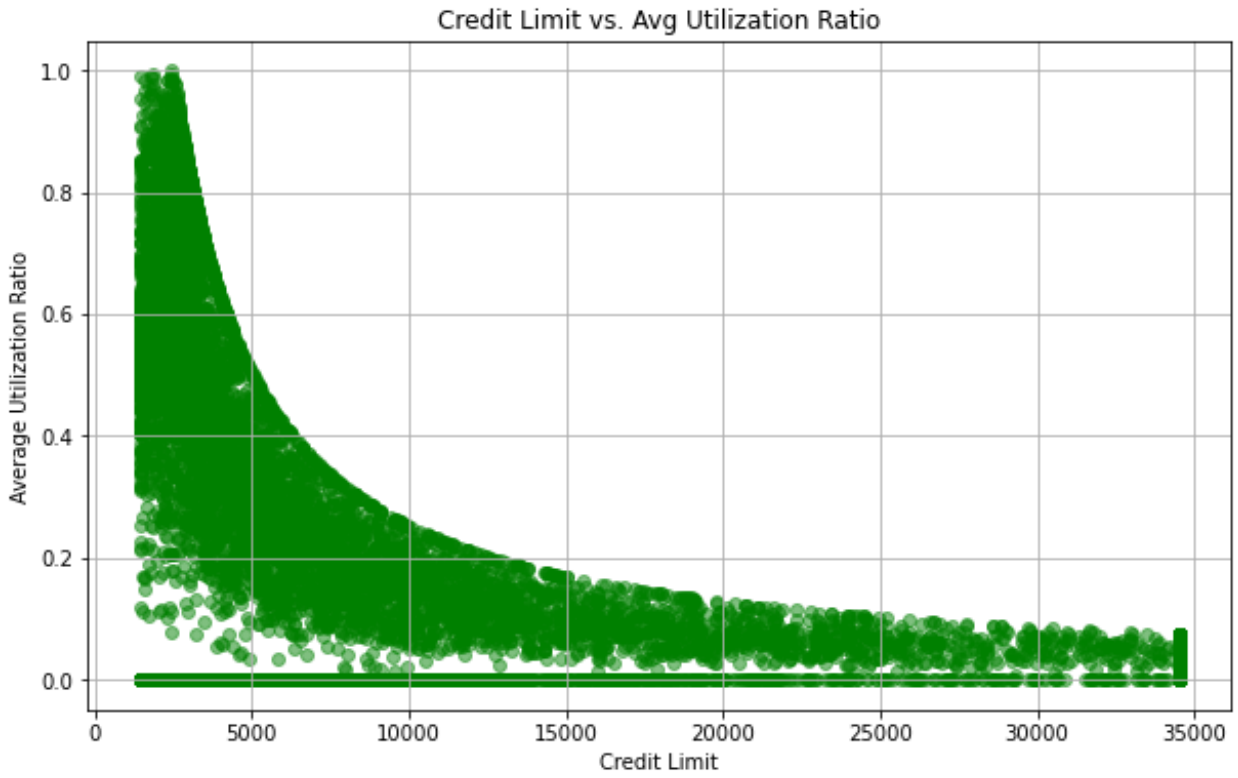
This gives a good idea about market segmentation. Understanding these numbers is important for card companies so they can make cards that people want and target the right customers with their marketing.

Full Interface Screenshot:



Analysis Question 5: What is the relationship between credit limit and average utilization ratio?

```
#Visulization5
#Q5)What is the relationship between credit limit and average utilization ratio?
# Plotting scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(df['credit_limit'], df['avg_utilization_ratio'], alpha=0.5, color='green')
plt.title('Credit Limit vs. Avg Utilization Ratio')
plt.xlabel('Credit Limit')
plt.ylabel('Average Utilization Ratio')
plt.grid(True)
plt.show()
```



First, let's clarify what the average utilization ratio means. For instance, if a customer has a credit limit of $10,000 and typically borrows $2,000, which is 20% of their credit limit, then that percentage is their average utilization ratio.

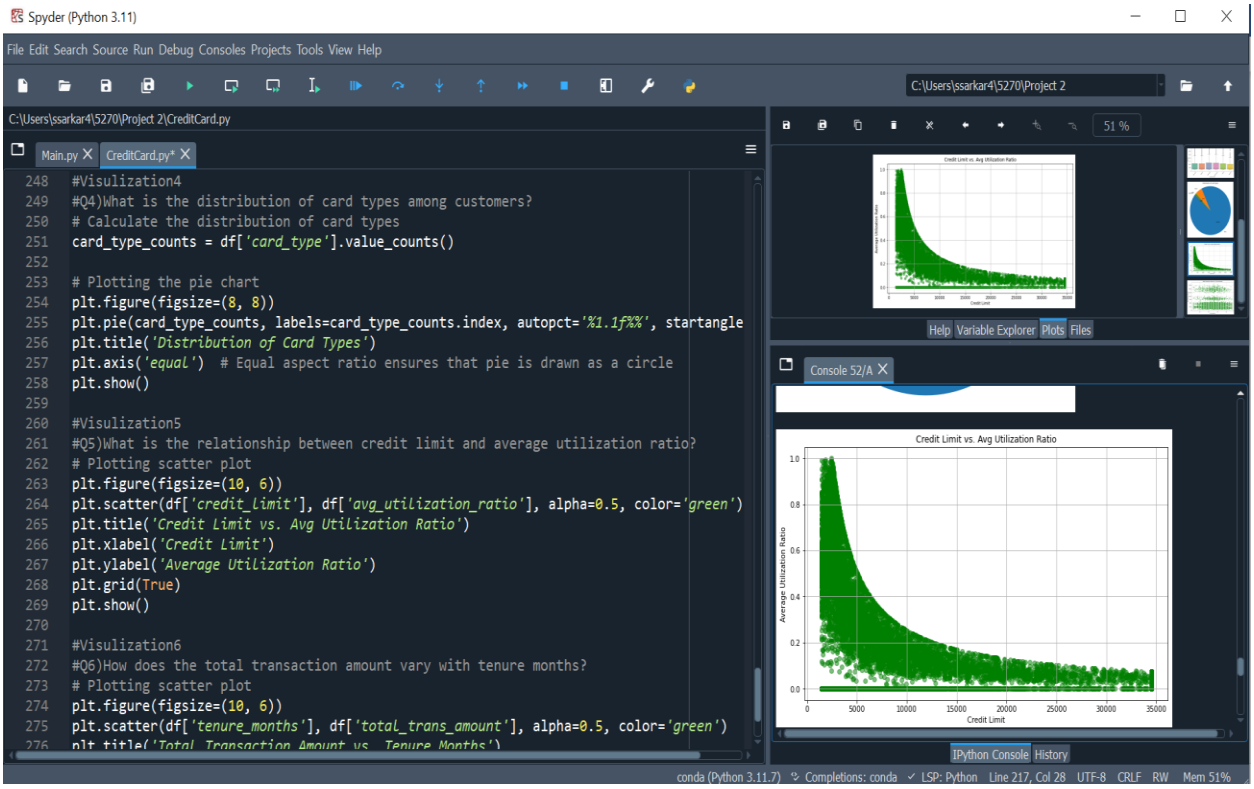
The scatter plot shows how credit limits and average utilization ratios are related for customers. Each dot on the plot stands for one customer, with their credit limit on the horizontal axis and average utilization ratio on the vertical axis. Customers with a credit limit of $14,000 or more experience a proportionate decrease in their average utilization ratio (AUR) from 20% while customers with credit limits in range of $1500-$3000 use almost full 100% credit.

If there's a line sloping from the bottom-left to the top-right of the plot, it means there's a positive correlation. This suggests that customers with higher credit limits usually have lower average utilization ratios. Knowing this connection is crucial for evaluating credit risk and planning good

credit management strategies. The scatter plot reveals a reverse connection between credit limits and average utilization ratios.

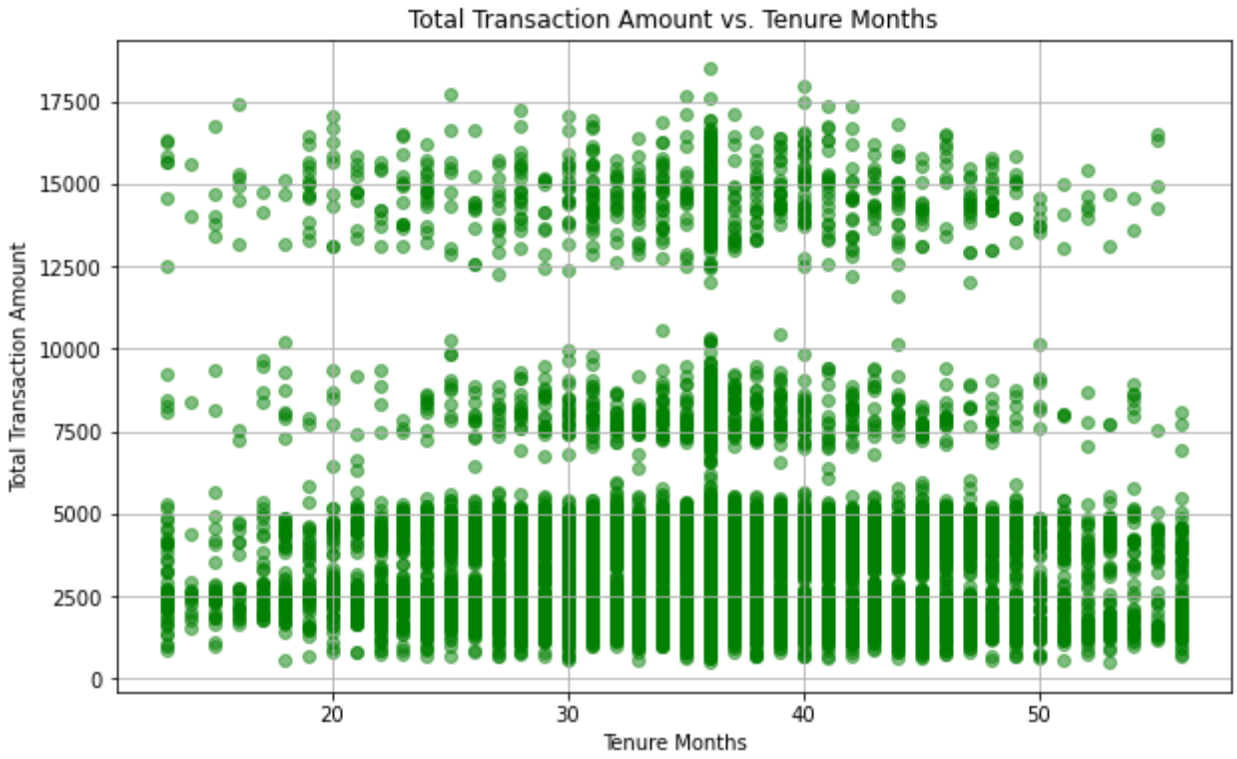
This analysis suggests that customers who use a smaller proportion of their credit limit tend to have higher credit limits. This could be an indicator of good financial management, where maintaining a low utilization ratio leads to increased credit trustworthiness and, consequently, higher credit limits. The credit card company might also be rewarding such behavior with higher credit limits to incentivize low utilization ratios. These insights can be used to develop strategies for encouraging responsible credit usage among cardholders.

Full Interface Screenshot:



Analysis Question 6: How does the total transaction amount vary with tenure months?

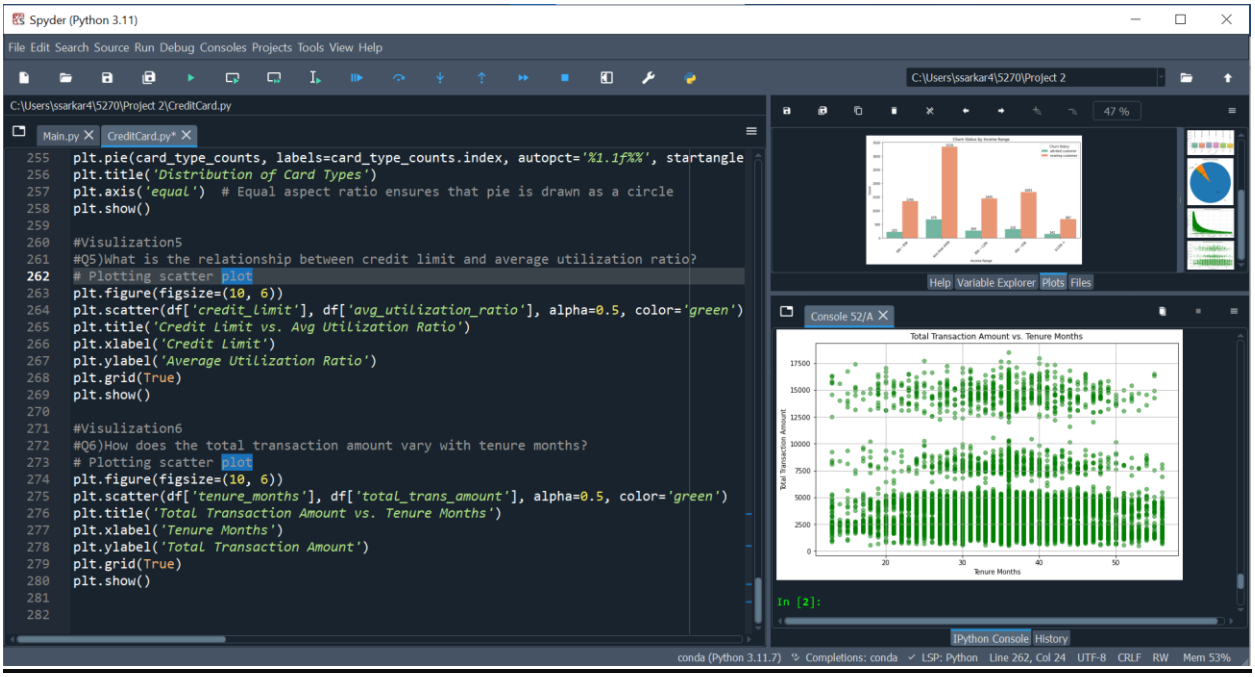
```
#Visulization6
#Q6)How does the total transaction amount vary with tenure months?
# Plotting scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(df['tenure_months'], df['total_trans_amount'], alpha=0.5, color='green')
plt.title('Total Transaction Amount vs. Tenure Months')
plt.xlabel('Tenure Months')
plt.ylabel('Total Transaction Amount')
plt.grid(True)
plt.show()
```



The scatter graph shows a positive correlation between total transaction amount and tenure months, which means that customers tend to spend more money on their credit cards the longer they have been using them. The above graph indicates that customers with a tenure between 30 to 40 months are more likely to spend a higher amount than those with a shorter tenure.

There are various possible reasons for their inclination to spend higher amounts than those with shorter tenures. Firstly, increased trust could play a significant role. Customers tend to feel more comfortable using their credit cards for larger purchases once they have established a track record of responsible usage and timely repayment. Secondly, the possibility of an increased credit limit over time could also contribute to higher spending. Responsible credit card users often receive rewards in the form of expanded credit limits, enabling them to make larger transactions with ease. Lastly, changing spending habits may be a factor. As individuals go through different stages of life, their financial priorities and habits evolve accordingly. Consequently, customers with longer tenures might experience changes in their lifestyles or financial circumstances that lead to higher expenditure. These factors collectively suggest a correlation between tenure length and spending behavior, highlighting the complex interplay of trust, credit management, and evolving consumer preferences in influencing credit card usage patterns.

Full Interface Screenshot:



Conclusion

In conclusion, delving into credit card data offers valuable insights for companies aiming to understand their customers better. This deeper understanding can pave the way for improved services and enhanced customer satisfaction. Our analysis revealed several key findings:

Firstly, we observed that middle-aged individuals constitute the primary user base for credit cards, with younger demographics displaying lower usage rates. This demographic breakdown informs targeted marketing strategies tailored to different age groups.

Secondly, concerning churn rates, customers with lower incomes are more likely to discontinue credit card usage. Additionally, a higher proportion of female customers tend to cease card usage compared to their male counterparts.

Thirdly, there exists a positive correlation between education levels and credit limits, with higher education levels typically associated with higher credit limits. This insight aids in understanding creditworthiness and tailoring credit offerings accordingly.

Moreover, our analysis highlighted the prevalence of basic card types among customers, with more specialized cards being less common. This information guides product development and marketing efforts to align with customer preferences.

Furthermore, we uncovered a relationship between credit limits and utilization ratios, indicating that customers with higher credit limits tend to utilize less of their available credit. This underscores the importance of financial management and responsible credit usage.

Lastly, considering additional insights into factors such as customer loyalty or changes in spending habits over time could provide a deeper understanding of the observed correlations.

This nuanced understanding enables companies to refine their strategies and offerings, ultimately fostering customer loyalty and satisfaction.

Works Cited

Nayeem, Noman. "Mastering Exploratory Data Analysis (EDA): A Comprehensive Python Pandas Guide for Data Insights." Medium, 28 Nov. 2023, https://medium.com/@nomannayeem/mastering-exploratory-data-analysis-eda-a-comprehensive-python-pandas-guide-for-data-insights-c0be7c5b8889. Accessed 10 May. 2024.

Appiah, Kevin. "Analysis: Credit Card Utilization Predictors." Medium, 1 Sep. 2023, https://medium.com/@kevinappiah23/analysis-credit-card-utilization-predictors-c944e7854513. Accessed 10 May. 2024.

PricewaterhouseCoopers (PwC). "Shaping Consumer Behavior." PwC Financial Services, Accessed 14 May 2024, https://www.pwc.com/us/en/industries/financial-services/library/shaping-consumer-behavior.html. Accessed 10 May. 2024.

"Credit Card Spending and Borrowing Since the Start of the COVID-19 Pandemic." Boston Federal Reserve, 19 Oct. 2023, https://www.bostonfed.org/publications/current-policy-perspectives/2023/credit-card-spending-and-borrowing-since-the-start-of-the-covid-19-pandemic.aspx. Accessed 14 May 2024. Accessed 14 May 2024.

"Data Cleaning Techniques." Finance Alliance, 2 June 2023, https://www.financealliance.io/data-cleaning-techniques/. Accessed 14 May 2024.

Soucy, Paul. "Credit Card Data, Statistics and Research." NerdWallet, 2024, https://www.nerdwallet.com/article/credit-cards/credit-card-data. Accessed 14 May 2024.

Frankel, Robin Saks. "The History of Credit Cards." Forbes Advisor, 2021, https://www.forbes.com/advisor/credit-cards/history-of-credit-cards/. Accessed 14 May 2024.