USC Viterbi
School of Engineering

ISE-535 MODULE 2 HOMEWORK

SPRING 2025

Bruce Wilcox,
PhD

GENERAL INSTRUCTIONS

- For this problem, you will be using the dataset in the file named customer_dataset.csv
- For each problem, you are to create the necessary Python code to answer the questions and then copy and paste screenshots of your code and the relevant response into this PowerPoint.
- When you are ready to submit your solutions, you should save this file as a PDF file and upload it to Gradescope.
- When uploading it to Gradescope, please follow the Gradescope prompt to indicate which pages in your solution correspond to the assignment questions

CUSTOMER DATA ANALYSIS

VARIABLE CLASSIFICATION/UNIVARIATE ANALYSIS

A company wants to better understand its customers by analyzing demographics, past purchase behavior, and engagement patterns. This dataset provides information that can help in customer segmentation, spending behavior analysis, and product recommendations.

Dataset variables:

Customer_ID – A unique identifier for each customer.

Age – The customer's age, useful for demographic analysis.

Number_of_Previous_Jobs – The number of jobs the customer has held, potentially indicating career stability.

Purchase_Count – Total number of purchases made by the customer.

Income_Level – Customer's income group: Low, Medium, or High.

Subscription_Years – How many years the customer has been subscribed to the company's services.

Has_Mobile_App – Whether the customer uses the mobile app (0 = No, 1 = Yes).

Annual_Spend – Total amount spent annually by the customer.

Education_Years – The number of years of education completed.

Review_Rating – The customer's average product review rating (1 to 5 stars).

1) Variable classification

- Load the data into a Pandas DataFrame
- Display the first few rows
- Classify each variable as:
  - Measure (Continuous or Discrete)
  - Category (Nominal or Ordinal)
- Modify your Pandas DataFrame as appropriate to ensure that Python treats each variable according to your classification

Screenshot of code to load dataframe and display first few rows:

Import data

```
7]: data = pd.read_csv('customer Dataset.csv')
```

Head of First Few Entries

```
9]: data.head(4)
```

9]:

Previous_Jobs	Income_Level	Subscription_Years	Has_Mobile_App	Purchase_Count	Review_Rating	Annual_Spend	Education_Years
15.000000	High	0.84	0	8.176449	5	44208.19	14.0
15.000000	Low	2.90	0	3.168492	3	14373.85	NaN
14.272860	Medium	3.57	1	6.304580	4	31689.01	20.0
12.412951	Low	1.50	1	2.328660	5	26693.14	NaN

Head of First Few Entries

```
]: data.head(4)
```

]:

	Customer_ID	Age	Number_of_Previous_Jobs	Income_Level	Subscription_Years	Has_Mobile_App	Purchase_Count	Review
0	1	86.674826	15.000000	High	0.84	0	8.176449	
1	2	86.045678	15.000000	Low	2.90	0	3.168492	
2	3	91.364302	14.272860	Medium	3.57	1	6.304580	
3	4	92.064757	12.412951	Low	1.50	1	2.328660	

Variable classification table:

Variable	Measure/Category	Continuous/Discrete Nominal/Ordinal
Customer_ID	Category	Nominal
Age	Measure	Discrete
Number_of_Previous_Jobs	Measure	Discrete
Purchase_Count	Measure	Discrete
Income_Level	Category	Ordinal
Subscription_Years	Measure	Discrete
Has_Mobile_App	Category	Nominal
Annual_Spend	Measure	Continuous
Education_Years	Measure	Discrete
Review_Rating	Category	Ordinal

Make necessary modifications to your dataframe to correspond to your variable classifications. For example, if you are going to treat a numeric variable as a category, use the Pandas “astype(‘category’) method. Display your code and the resulting datatypes (using .dtypes) on the following page

Dataset Features Conversion

```
## Imputing the Null Entries
df=data
df['Age'].fillna(df['Age'].mean(), inplace=True) # type : ignore
df['Annual_Spend'].fillna(df['Annual_Spend'].median(), inplace=True) # type : ignore
df['Education_Years'].fillna(df['Education_Years'].median(), inplace=True) # type : ignore


## Converting the Data Types
df['Number_of_Previous_Jobs'] = df['Number_of_Previous_Jobs'].astype('int64')  # Discrete # type : ignore
df['Has_Mobile_App'] = df['Has_Mobile_App'].astype('category')  # Categorical/Nominal
df['Purchase_Count'] = df['Purchase_Count'].astype('int64')  # Discrete
df['Education_Years'] = df['Education_Years'].astype('int64')  # Discrete
data['Income_Level'] = data['Income_Level'].astype('category').cat.codes



## Converting Age and Subscription years

df['Age'] = df['Age'].round().astype(int)
df['Subscription_Months'] = (df['Subscription_Years'] * 12).round().astype(int)
df = df.drop(['Subscription_Years'], axis=1)

df['Income_Level'] = df['Income_Level'].astype('category')
df['Review_Rating'] = df['Review_Rating'].astype('category')
```

CUSTOMER DATA ANALYSIS

VARIABLE CLASSIFICATION/UNIVARIATE ANALYSIS

-> Dropped Identifier Column Customer ID

-> Created a new feature called subscription months and dropped subscription years for ease of interpretation

Updated Data Type Description

```
Age                          int64
Number_of_Previous_Jobs      int64
Income_Level              category
Has_Mobile_App            category
Purchase_Count               int64
Review_Rating             category
Annual_Spend               float64
Education_Years              int64
Subscription_Months          int64
dtype: object
```

```
[361]: data.head(6)
```

[361]:

	Age	Number_of_Previous_Jobs	Income_Level	Has_Mobile_App	Purchase_Count	Review_Rating	Annual_Spend	Education_Years	Subscription_Months
0	87	15	0	0	8	5	44208.19	14	10
1	86	15	1	0	3	3	14373.85	13	35
2	91	14	2	1	6	4	31689.01	20	43
3	92	12	1	1	2	5	26693.14	13	18
4	85	12	1	0	4	3	12150.15	13	23
5	94	15	1	0	1	3	18146.82	13	5

2) Univariate Analysis – Descriptive Statistics

Create a table summarizing the descriptive statistics for your measures. On the following page, paste a screenshot of your code to generate the table and the table itself. Be sure that your statistics include Skew and Kurtosis

CUSTOMER DATA ANALYSIS

UNIVARIATE ANALYSIS

[367…

```
import pandas as pd
import scipy.stats as stats


numeric_columns = data.select_dtypes(include=['number']).columns
desc_stats = data[numeric_columns].describe().T

# Calculate skewness and kurtosis for numeric columns
desc_stats['skew'] = data[numeric_columns].skew()
desc_stats['kurtosis'] = data[numeric_columns].kurtosis()

# For categorical columns, unique values and mode
for col in data.select_dtypes(include=['category']).columns:
    desc_stats.loc[col, 'unique'] = data[col].nunique()
    desc_stats.loc[col, 'mode'] = data[col].mode()[0]


desc_stats
```

	count	mean	std	min	25%	50%	75%	max	skew	kurtosis	unique	mode
Age	1000.0	35.66500	7.214348	19.00	32.0000	35.000	38.000	94.00	4.187335	29.986393	NaN	NaN
Number_of_Previous_Jobs	1000.0	2.68400	2.020452	0.00	1.0000	2.000	4.000	15.00	1.931317	8.550344	NaN	NaN
Purchase_Count	1000.0	3.45000	2.033134	0.00	2.0000	3.000	5.000	12.00	0.499586	0.298372	NaN	NaN
Annual_Spend	1000.0	11433.16554	8542.161652	-4798.32	5506.7375	9464.785	15562.355	44208.19	1.073247	0.786004	NaN	NaN
Education_Years	1000.0	13.04100	2.886807	6.00	13.0000	13.000	13.000	20.00	0.037561	1.043922	NaN	NaN
Subscription_Months	1000.0	59.91000	60.908339	0.00	17.0000	41.000	81.250	452.00	1.991001	5.528214	NaN	NaN
Income_Level	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.0	1.0
Has_Mobile_App	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2.0	1.0
Review_Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	5.0

2) Univariate Analysis – Measure Visualizations

Create a series of histograms and boxplots to visualize your measures. On the following pages, paste screenshots of your code and of the visualizations.

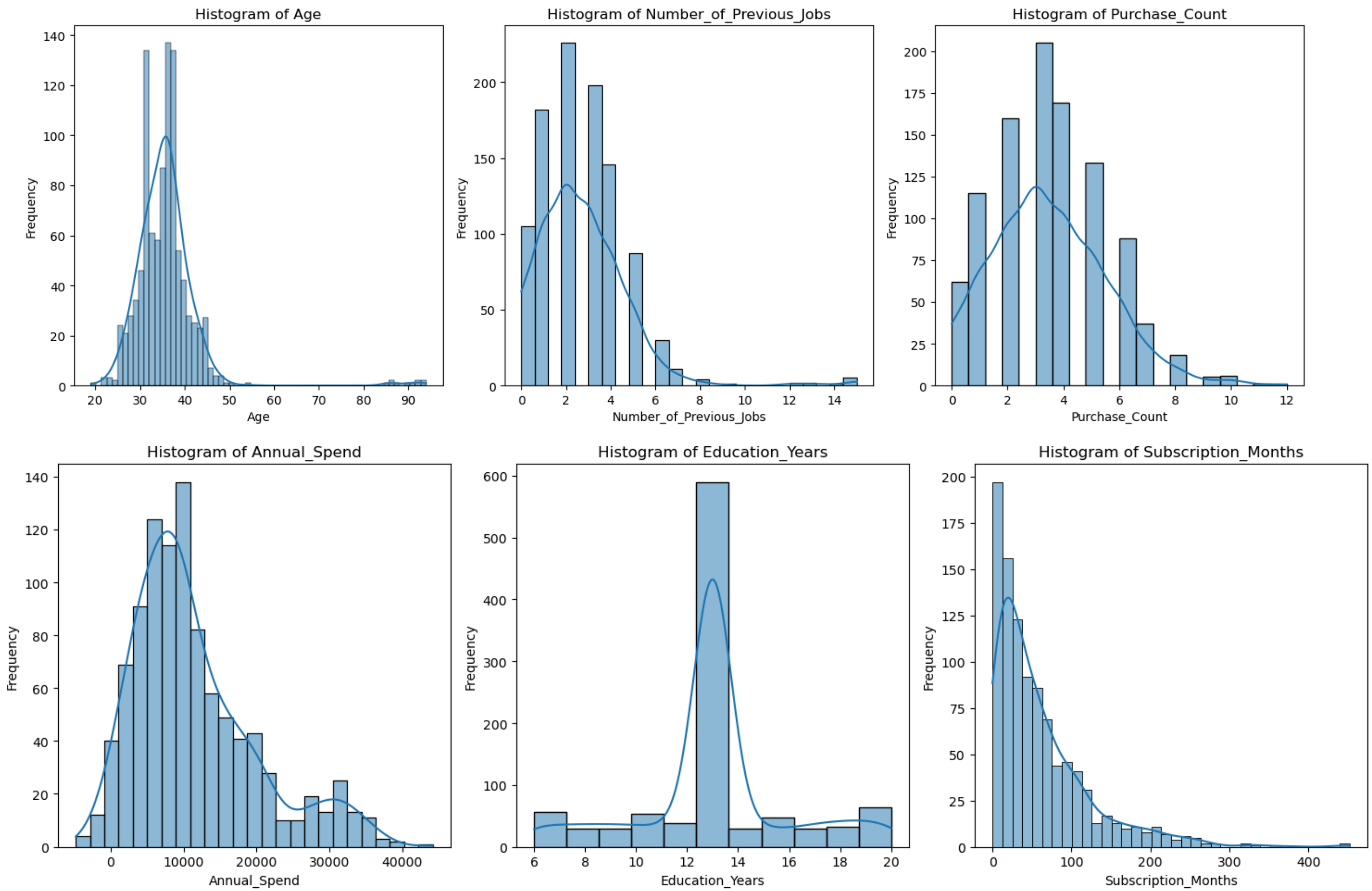
Univariate measure Visualizations

371]:
```
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(15, 10))

# List of numeric columns for visualization
numeric_columns = data.select_dtypes(include=['number']).columns

# Generate histograms
for i, col in enumerate(numeric_columns, 1):
    plt.subplot(2, len(numeric_columns)//2, i)
    sns.histplot(data[col], kde=True)
    plt.title(f'Histogram of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

# Generate boxplots
plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_columns, 1):
    plt.subplot(2, len(numeric_columns)//2, i)
    sns.boxplot(x=data[col])
    plt.title(f'Boxplot of {col}')
    plt.xlabel(col)

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

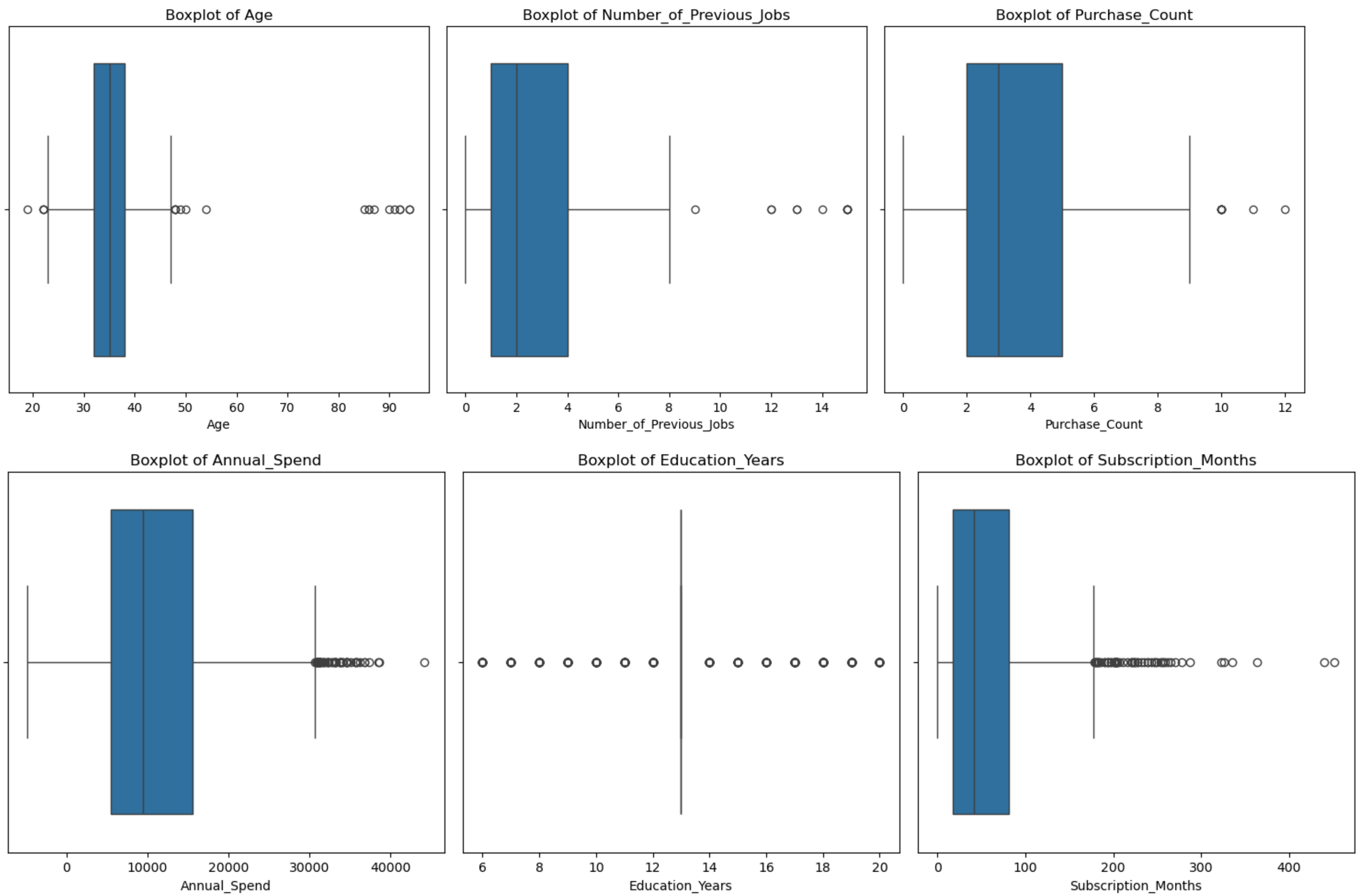
CUSTOMER DATA ANALYSIS

UNIVARIATE ANALYSIS

2) Univariate Analysis – Measure Visualizations

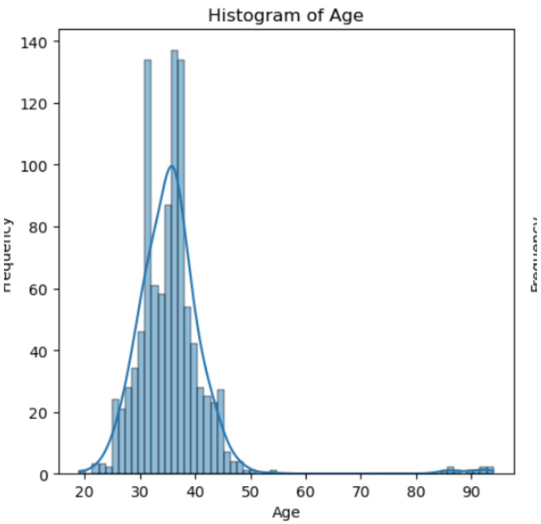


2) Univariate Analysis – Measure Visualizations

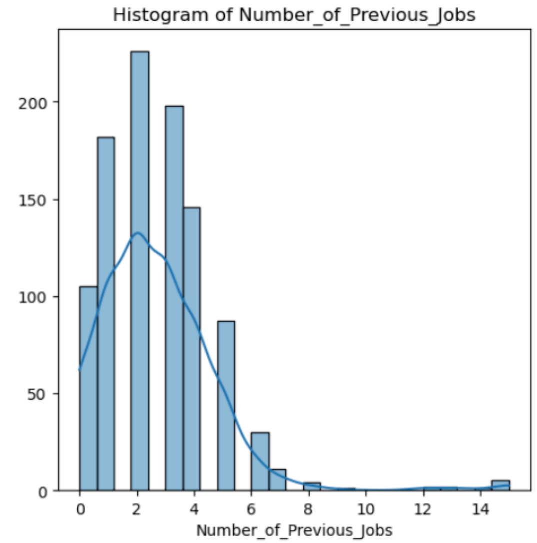


2) Univariate Analysis – Measures

Based on a review of the histograms comment below on the distribution shape of each variable and whether it seems reasonable given the nature of the variable

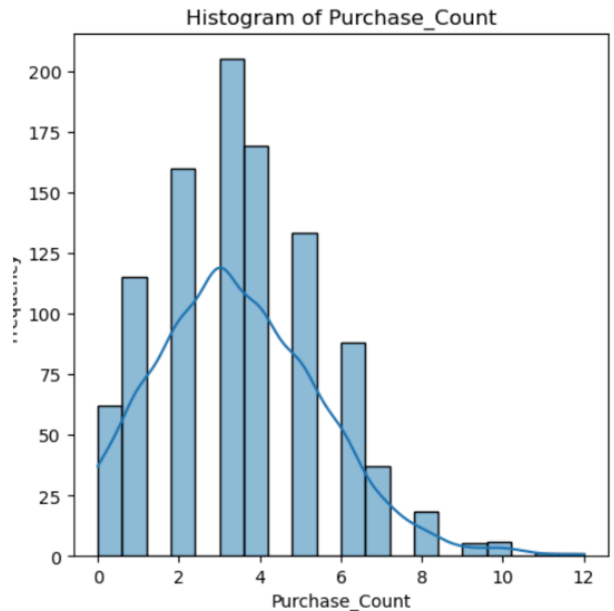


Age : Shows a distribution with highly right skewed values including outliers to the right with age group of 80-90+. The distribution is peaked up and not widely spread out, having a high kurtosis. Seems reasonable.

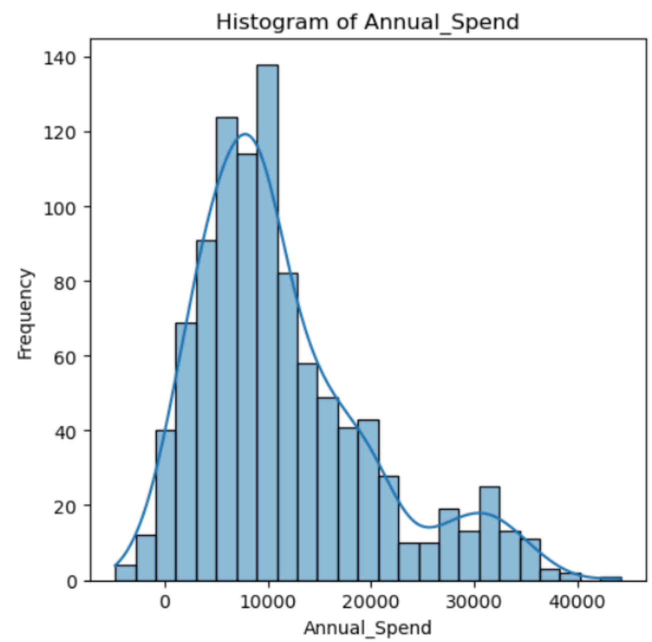


Number of Previous Jobs: It is right skewed with some outliers having 10-14 jobs; with moderate kurtosis with mean values of previous jobs as 2.68 (close to 3). Seems reasonable but has outliers with 10-14 previous jobs

2) Univariate Analysis – Measure Visualizations



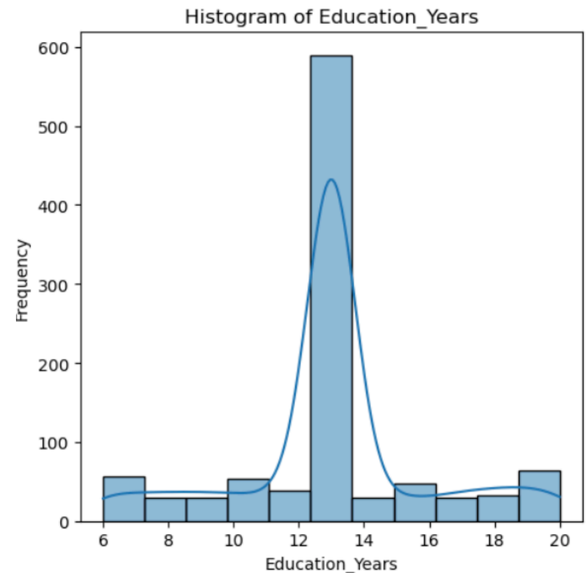
Purchase Count: The distribution is slightly right skewed with an average purchase made by customers as 3.4 (3-4). Seems Reasonable for the variable

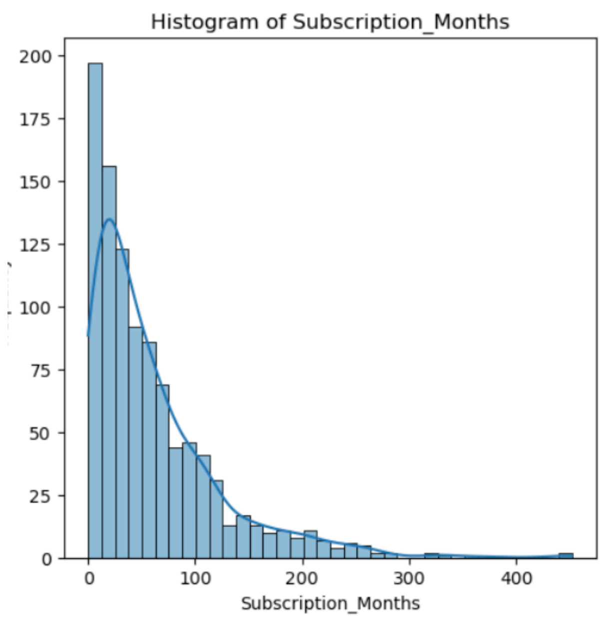


Annual Spend : The distribution is right skewed and seems reasonable with average spending amount around $11,433, having outliers with people making high value purchases. Seems Reasonable, and depends.

2) Univariate Analysis – Measure Visualizations

Education years : Seems Reasonable with most people having around 12 years of education with very light skewness - resembling a normal distribution. Almost equal amount of people have 6-12 and 12-20 years of education.





Subscription Months: Perfectly right skewed having outliers of 200+ months of subscription and majority of customers being not subscribed. Seems Reasonable.

2) Univariate Analysis – Measures

Based on a review of the histograms and box plot, which variable appears to have the most extreme outliers?

- Age has Most Extreme outliers with high skewness of 4.187335

2) Univariate Analysis – Categories

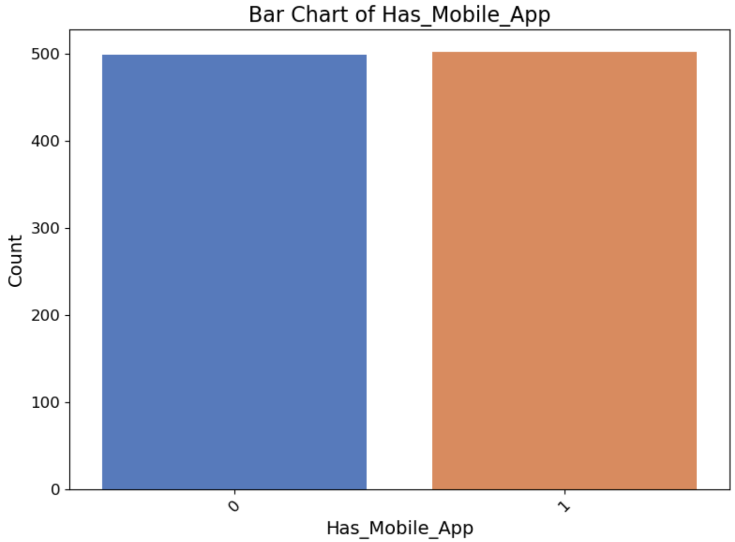
Create bar charts of the categories (don't include Customer_ID). Paste on the following pages your code and the bar chart visualizations

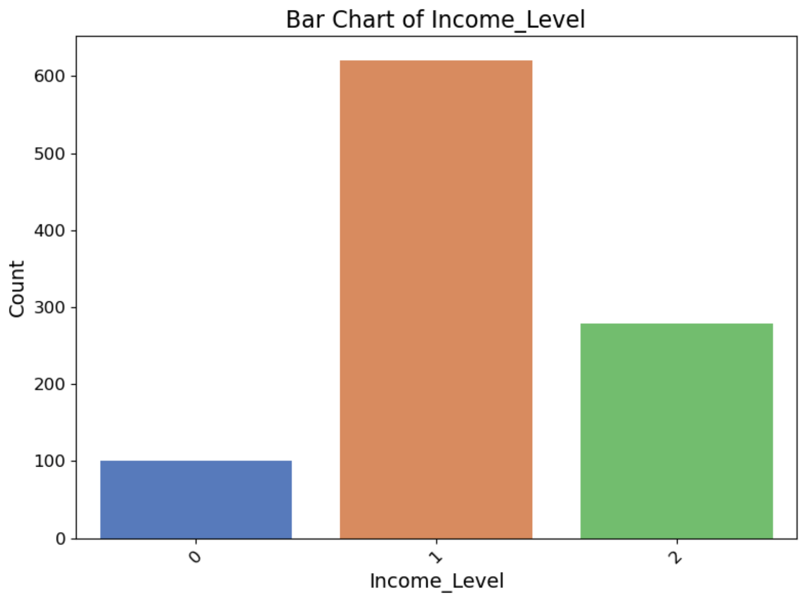
```
76]: import matplotlib.pyplot as plt
import seaborn as sns
import warnings

# Ignore warnings
warnings.filterwarnings('ignore')

# List of categorical columns to plot
categorical_columns = ['Income_Level', 'Has_Mobile_App', 'Review_Rating']

# Create a bar plot for each categorical column
for col in categorical_columns:
    plt.figure(figsize=(8, 6))  # Set figure size
    sns.countplot(data=data, x=col, palette='muted')
    plt.title(f'Bar Chart of {col}', fontsize=16)
    plt.xlabel(col, fontsize=14)  # X-axis label
    plt.ylabel('Count', fontsize=14)  # Y-axis label
    plt.xticks(rotation=45, fontsize=12)
    plt.yticks(fontsize=12)
    plt.tight_layout()
    plt.show()  # Show plot
```

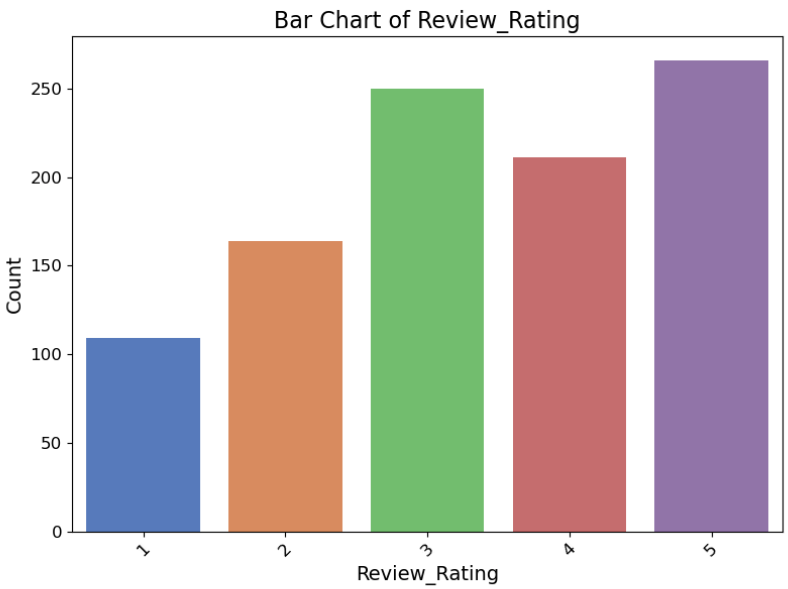
CUSTOMER DATA ANALYSIS

UNIVARIATE ANALYSIS

2) Univariate Analysis – Categories







3) Univariate Analysis – Diagnosis. Paste below a table listing the number of missing values in each variable

Missing Data Handling

```
[322]: data.isna().sum()

[322]: Customer_ID                  0
       Age                         60
       Number_of_Previous_Jobs      0
       Income_Level                 0
       Subscription_Years           0
       Has_Mobile_App               0
       Purchase_Count               0
       Review_Rating                0
       Annual_Spend                40
       Education_Years            558
       dtype: int64
```

3) For each variable, we wish to determine if they are Missing Completely at Random (MCAR) or Missing at Random (MAR). A simple test is to create a new binary column in the dataframe for each variable that has missing values that indicates whether that row is missing that variable. Then, using this new column, calculate frequency counts for each of the other category variables for those rows with and without missing values. If the frequency counts are significantly different, that is an indication that variable may not be MCAR.

On the following pages, show your code that does this analysis and summarize your conclusions.

Missing Value Analysis

[392]:

```
data_m = pd.read_csv('customer Dataset.csv')

# Creating binary columns indicating whether each variable is missing
for col in ['Age', 'Annual_Spend', 'Education_Years']:
    data_m[f'{col}_missing'] = data_m[col].isna().astype(int)


categorical_columns = ['Income_Level', 'Has_Mobile_App', 'Review_Rating']

# For each variable with missing data, calculate frequency counts by 'missing' status
for col in ['Age', 'Annual_Spend', 'Education_Years']:
    print(f"\nFrequency counts for missing vs non-missing {col} values:\n")
    for cat_col in categorical_columns:
        print(f"\n--- {cat_col} ---")
        print("Missing:")
        print(data_m[data_m[f'{col}_missing'] == 1][cat_col].value_counts())
        print("Not Missing:")
        print(data_m[data_m[f'{col}_missing'] == 0][cat_col].value_counts())
        print("\n")
```

CUSTOMER DATA ANALYSIS

UNIVARIATE ANALYSIS - DIAGNOSIS

Variables Age, Annual Spending, Education years has missing values

Analyzing the Missing Criteria in Age:

```
--- Income_Level ---
Missing:
Income_Level
Low       43
Medium    11
High       6
Name: count, dtype: int64
Not Missing:
Income_Level
Low       578
Medium    268
High       94
Name: count, dtype: int64
```

```
--- Has_Mobile_App ---
Missing:
Has_Mobile_App
1    30
0    30
Name: count, dtype: int64
Not Missing:
Has_Mobile_App
1    472
0    468
Name: count, dtype: int64
```

```
--- Review_Rating ---
Missing:
Review_Rating
5    16
4    14
3    12
1    10
2     8
Name: count, dtype: int64
Not Missing:
Review_Rating
5    250
3    238
4    197
2    156
1     99
Name: count, dtype: int64
```

Analyzing the Missing Criteria in Annual Spending:

```
--- Income_Level ---
Missing:
Income_Level
Low        24
Medium     13
High        3
Name: count, dtype: int64
Not Missing:
Income_Level
Low       597
Medium    266
High       97
Name: count, dtype: int64
```

```
--- Has_Mobile_App ---
Missing:
Has_Mobile_App
0    22
1    18
Name: count, dtype: int64
Not Missing:
Has_Mobile_App
1    484
0    476
Name: count, dtype: int64
```

```
--- Review_Rating ---
Missing:
Review_Rating
5    17
4    10
3     7
2     4
1     2
Name: count, dtype: int64
Not Missing:
Review_Rating
5    249
3    243
4    201
2    160
1    107
Name: count, dtype: int64
```

CUSTOMER DATA ANALYSIS

UNIVARIATE ANALYSIS - DIAGNOSIS

Analyzing the Missing Criteria in Education Years:

```
--- Income_Level ---
Missing:
Income_Level
Low    558
Name: count, dtype: int64
Not Missing:
Income_Level
Medium    279
High      100
Low        63
Name: count, dtype: int64
```

```
--- Has_Mobile_App ---
Missing:
Has_Mobile_App
0    279
1    279
Name: count, dtype: int64
Not Missing:
Has_Mobile_App
1    223
0    219
Name: count, dtype: int64
```

```
--- Review_Rating ---
Missing:
Review_Rating
5    157
3    146
4    119
2     84
1     52
Name: count, dtype: int64
Not Missing:
Review_Rating
5    109
3    104
4     92
2     80
1     57
Name: count, dtype: int64
```

Conclusions:

› * **Missing in Age : (MCAR)**

1. By looking at the Income level, the age variable seems to be missing for low income group, however, the not missing values shows otherwise, considering the proportion size of each category.

2. Having a mobile app does not seem to have any relation with missingness in Age.

3. Missing Ages are high in high reviews but also considering the Not missing values ranges of review ratings, it shows that the age is randomly missing without any underlying strong evidence. Hence, this could be Missing Completely at Random. (MCAR)

* **Missing in Annual Spending : (MAR)**

1. The low income level group has a high missing reports of annual spend, but the not missing stats show this might be at random.

2. Having a mobile app does seem to have a slight relation with missingness in Annual spending, with those without apps having no annual spending reported, whereas those with apps having less missing values.

3. Missing Annual Spendings are high in high reviews but also considering the Not missing values ranges of review ratings, it shows that the age is randomly missing without any underlying strong evidence. This could be Missing at Random. (MAR)

CUSTOMER DATA ANALYSIS

UNIVARIATE ANALYSIS - DIAGNOSIS

* **Missing in Education years : (MNCAR) - Not missing completely at Random**

1. By looking at the Income level, the Education years variable seems to be missing one and only for low income group. This is a strong indication that customers fail to provide education years whose income levels are low.

2. Having a mobile app does not seem to have any relation with missingness in Education Years.

3. Missing education levels are high in high reviews but also considering the Not missing values ranges of review ratings, it shows that the education is randomly missing without any underlying strong evidence with respect to this.

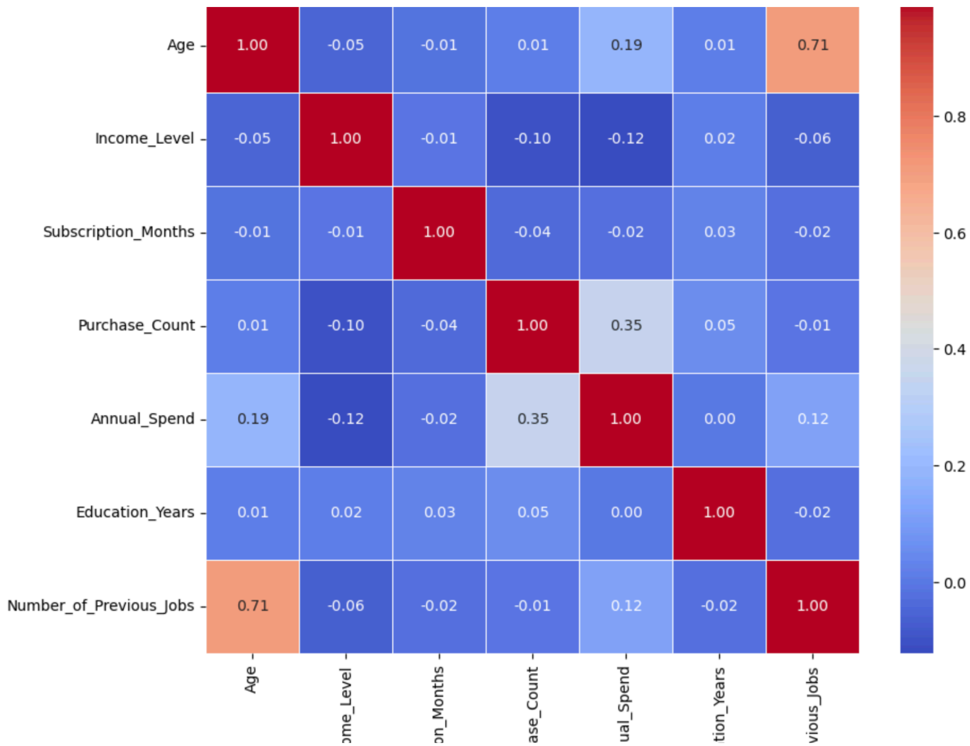
The variable Education can be considered not missing at random considering the Income Level.

CUSTOMER DATA ANALYSIS

BIVARIATE ANALYSIS

4) Create and display below a correlation matrix of the measures in your dataset

[66]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'data' is the dataframe containing your dataset

# Step 1: Select numeric variables (you can adjust this list based on your dataset)
numeric_columns = ['Age', 'Income_Level','Subscription_Months', 'Purchase_Count', 'Annual_Spend', 'Education_Years', 'Number_of_Previous_Jobs'

# Step 2: Calculate the correlation matrix0
correlation_matrix = data[numeric_columns].corr()

# Step 3: Visualize the correlation matrix as a heatmap
plt.figure(figsize=(10, 8))  # Set the size of the figure
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', cbar=True, square=True, linewidths=0.5)
plt.title("Correlation Matrix of Numeric Variables")
plt.show()
```



4) Comment below on the correlation matrix. What correlations exist? Do you see anything unexpected?

Weak Positive Correlation: Age and Annual Spend (0.15)

This makes sense as Young people (teens to young adults) might spend slightly less than middle aged or adults who might be more financially stable.

Moderate Positive Correlation : Annual Spend and Purchase Count (0.35)

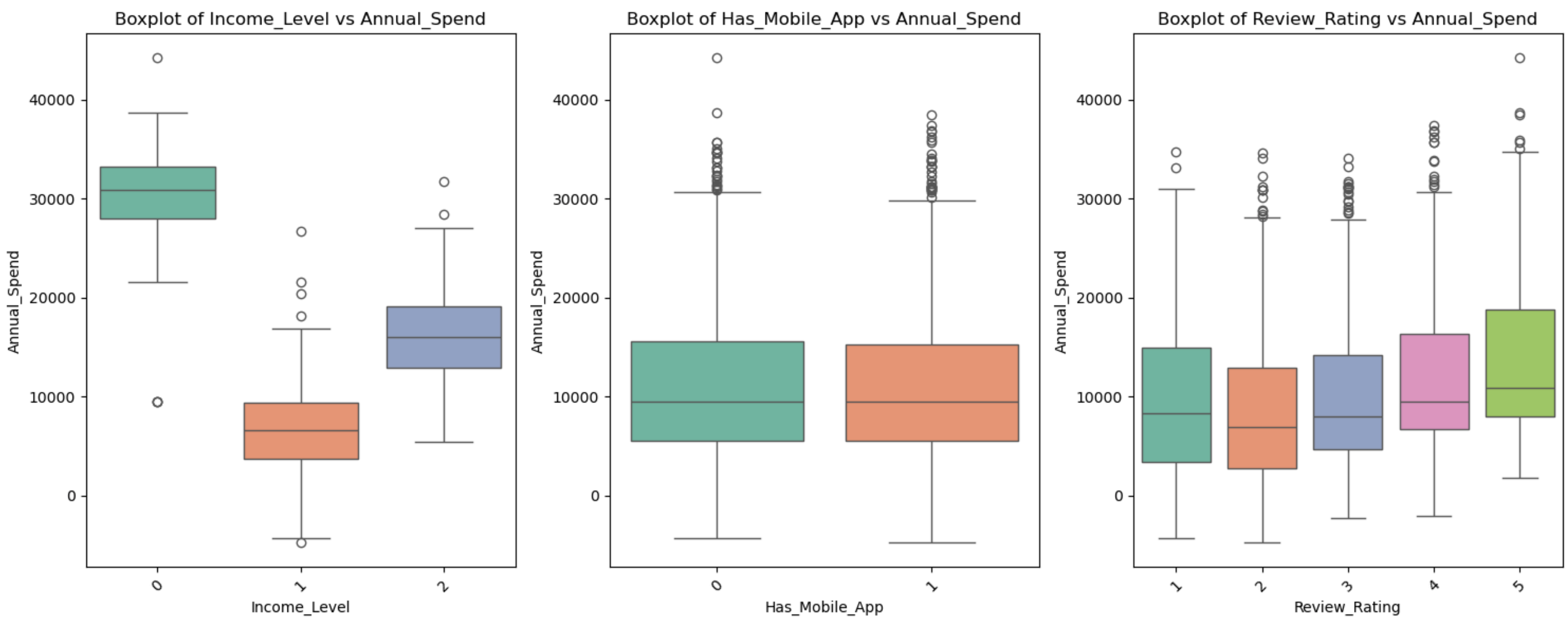
This also makes sense as more the annual expense more the number of purchases made.

High Positive Correlation: 0.71 Correlation between Age and Number of previous jobs

CUSTOMER DATA ANALYSIS

BIVARIATE ANALYSIS

4) Create and display below side-by-side boxplots of each category against Annual_Spend.



4) Comment below on the side-by-side boxplots. What correlations exist? Do you see anything unexpected?

The Income Level and Annual spending, has an unexpected behavior having more annual spending for low income category customers, whereas less annual spendings for high income level compared to low income customers. We might assume there might be a mindless spending for low income customers, compared to medium-high income cohort showing a diligent spending.

The presence of mobile app and Annual spending does no have any significant correlation.

The annual spending of high review customers is higher than low reviewed customers, and shows a gradual progression of spendings with review rates, which is quite expected.

4) Create scatterplots of each measure vs. annual spend. Paste your code and resulting plots on the following page(s).

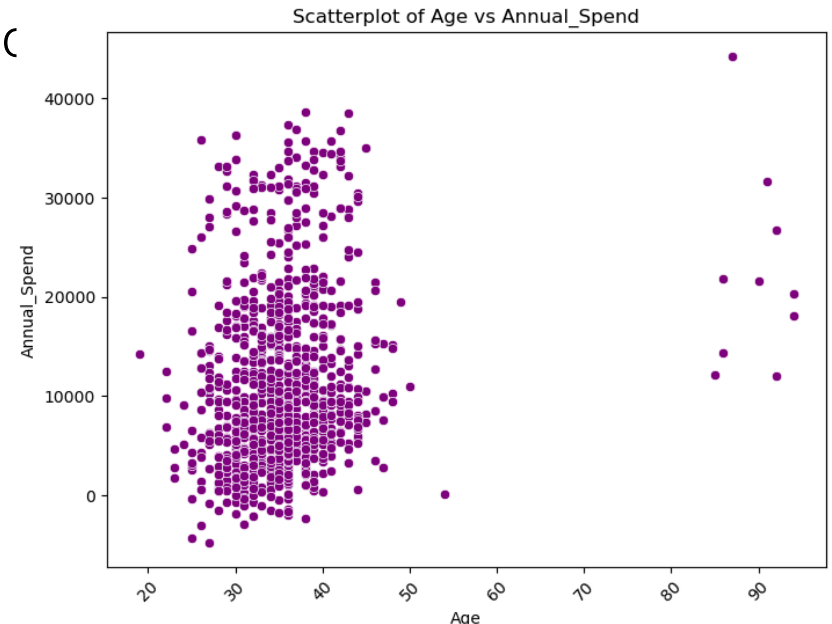
Measures Vs Annual Spend

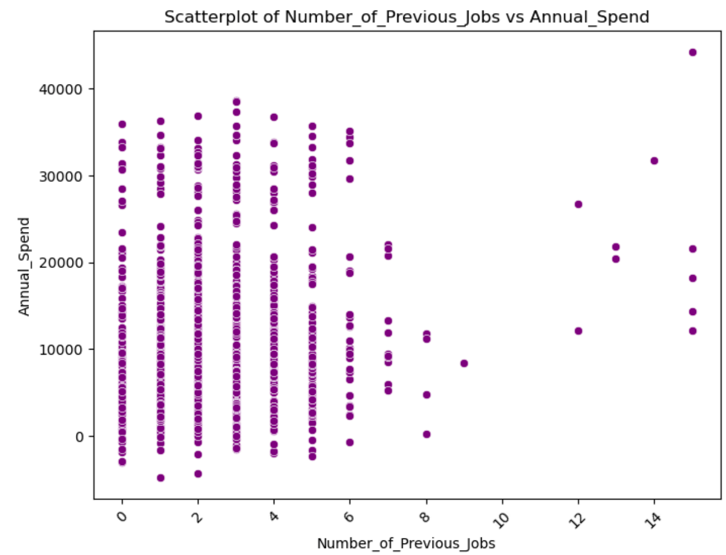
.8]:
```
import seaborn as sns
import matplotlib.pyplot as plt

numerical_vars = ['Age', 'Number_of_Previous_Jobs', 'Purchase_Count', 'Education_Years', 'Subscription_Months']

# Step 2: Create scatterplot for each numerical variable against Annual_Spend
for var in numerical_vars:
    plt.figure(figsize=(8, 6))  # Set the size of the figure
    sns.scatterplot(x=var, y='Annual_Spend', data=data, color='purple')
    plt.title(f'Scatterplot of {var} vs Annual_Spend')
    plt.xticks(rotation=45)  # Rotate x-axis labels if necessary
    plt.show()
```

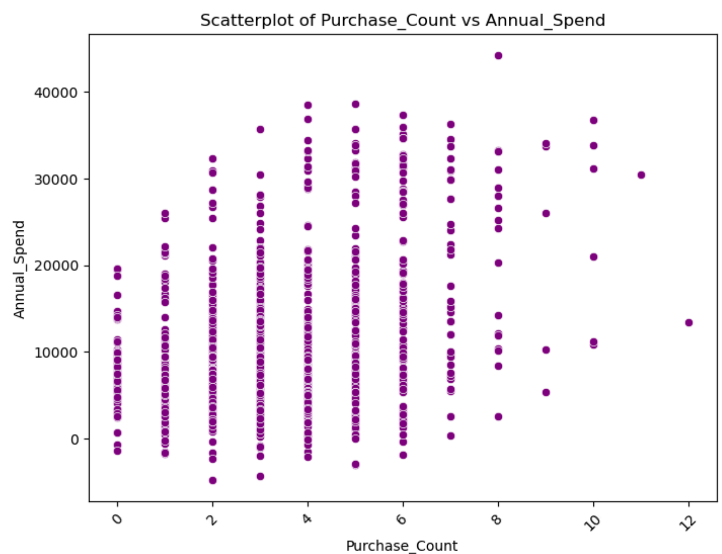
CUSTOMER DATA ANALYSIS

BIVARIATE ANALYSIS

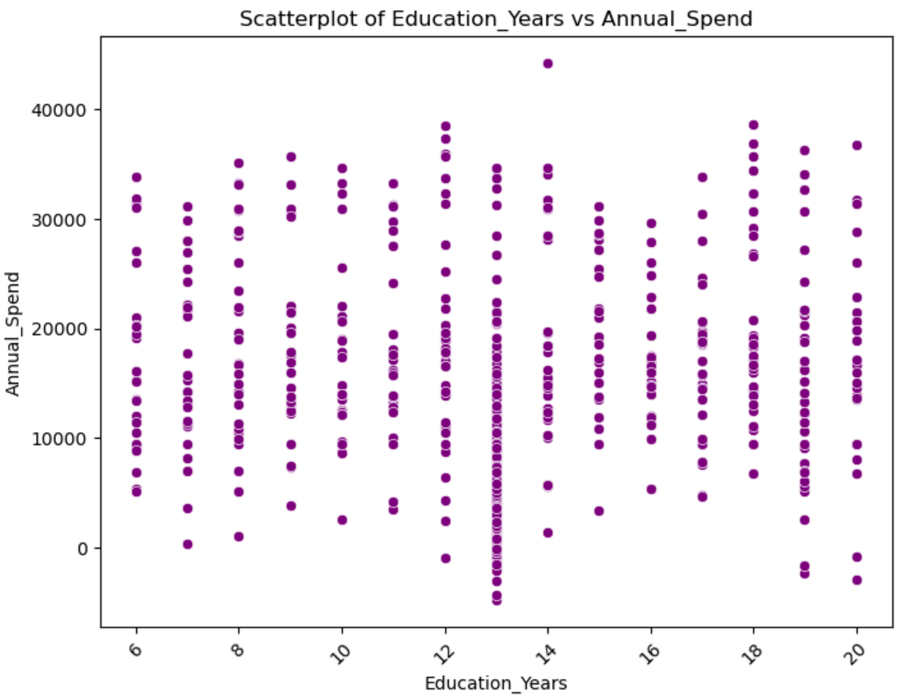
Scatterplo

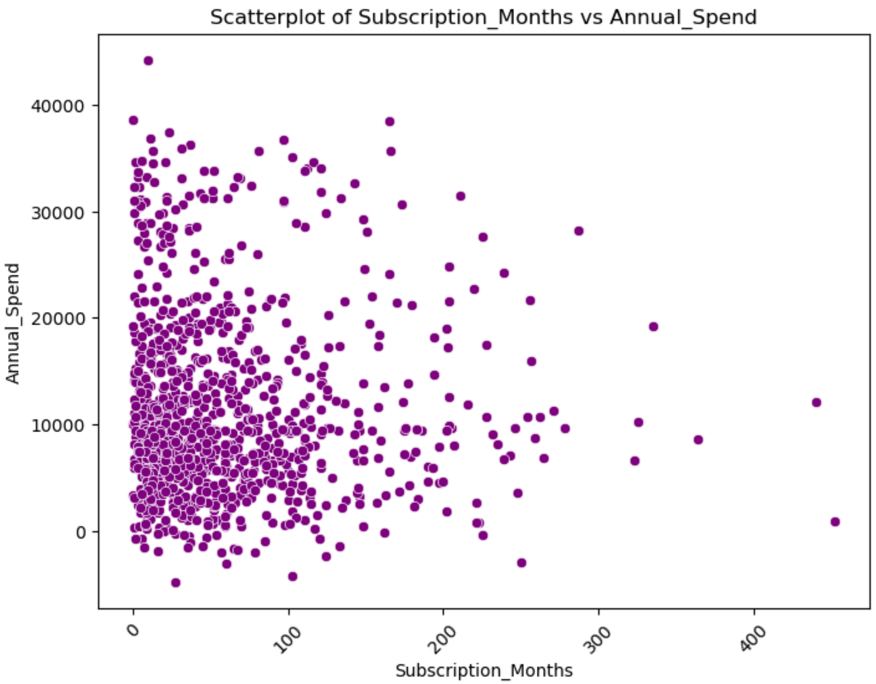






Scatterplots





CUSTOMER DATA ANALYSIS

BIVARIATE ANALYSIS - DIAGNOSIS

5) To identify potential multicollinearity, calculate the Variance Inflation Factor (VIF) statistics for each of the measures. Hint: use the variance_inflation_factor function 9n the statsmodels.stats.outlier_influence library. On the following code display your code and the resulting statistics.

```
[70]: import pandas as pd
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant


numerical_vars = ['Age', 'Number_of_Previous_Jobs', 'Purchase_Count',
                  'Annual_Spend', 'Education_Years', 'Subscription_Months']
X = data[numerical_vars]


vif_data = pd.DataFrame()
vif_data["variable"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

print(vif_data)
```

CUSTOMER DATA ANALYSIS

BIVARIATE ANALYSIS - DIAGNOSIS

VIF calculations:

	variable	VIF
0	Age	25.081613
1	Number_of_Previous_Jobs	4.765390
2	Purchase_Count	4.309726
3	Annual_Spend	3.314980
4	Education_Years	14.715171
5	Subscription_Months	1.943472

VIF calculations: Comment below on your interpretation of the VIF statistics

Age: Age has a high VIF of 25.08 showing high significance to the other variables, especially the number of years of education.

Number of education years: A VIF 14.715 shows a high significance among the other variables.

Since the age and number of years of education has a high variance compared to others, they are considered to be highly significant and potentially signify multicollinearity.