Unlocking Customer Insights: A Statistical Investigation

Ayush Kumar Siddharth
Batch: 2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Understand Data

```
df = pd.read_csv("/content/US_Customer_Insights_Dataset.csv")
```

```
print(df.head()) print(df.info()) print(df.isnull().sum()) # All is fine There
is no null values in the Dataset
```

```
   CustomerID              Name       State   Education      Gender  Age  \
0 CUST10319         Scott Perez     Florida  High School Non-Binary   47
1 CUST10695     Jennifer Burton  Washington       Master        Male   72
2 CUST10297     Michelle Rogers     Arizona       Master      Female   40
3 CUST10103 Brooke Hendricks        Texas       Master        Male   27
4 CUST10219 Karen Johns             Texas  High School      Female   28

  Married   NumPets JoinDate TransactionDate MonthlySpend \
0     Yes         1  9/19/21          9/2/24      1281.74
1     Yes         0   4/5/24          6/2/24       429.46
2     Yes         2  7/24/24         2/28/25       510.34
3     Yes         0  8/12/23         3/29/25       396.47
4     Yes         1  12/6/21         7/24/22       139.68

   DaysSinceLastInteraction
0                       332
1                       424
2                       153
3                       124
4                      1103
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10675 entries, 0 to 10674
Data columns (total 12 columns):
 #   Column                    Non-Null Count Dtype
---  ------                    -------------- -----
0    CustomerID                10675 non-null object
1    Name                      10675 non-null object
2    State                     10675 non-null object
3    Education                 10675 non-null object
4    Gender                    10675 non-null object
5    Age                       10675 non-null int64
6    Married                   10675 non-null object
7    NumPets                   10675 non-null int64
8    JoinDate                  10675 non-null object
9    TransactionDate           10675 non-null object
10   MonthlySpend              10675 non-null float64
 11  DaysSinceLastInteraction  10675 non-null int64
dtypes: float64(1), int64(3), object(8)
memory usage: 1000.9+ KB
None
CustomerID                  0
Name                        0
State                       0
Education                   0
Gender                      0
Age                         0
Married                     0
NumPets                     0
JoinDate                    0
TransactionDate             0
MonthlySpend                0
DaysSinceLastInteraction    0
```

```
dtype: int64
```

```
print(df.describe())
```

```
                Age       NumPets   MonthlySpend  DaysSinceLastInteraction
count  10675.000000  10675.000000   10675.000000              10675.000000
mean      49.474567      1.340515     331.610315                538.469883
std       18.221365      1.150849     225.799253                398.766747
min       18.000000      0.000000       3.890000                  1.000000
25%       35.000000      0.000000     165.495000                218.000000
50%       49.000000      1.000000     282.110000                445.000000
75%       66.000000      2.000000     443.255000                788.500000
max       80.000000      4.000000    1740.420000               1791.000000
```

```
# I clearify which variable is categorical and which is numerical
categorical = df.select_dtypes(include=['object']).columns.tolist()
numerical = df.select_dtypes(include=['int64', 'float64']).columns.tolist()
print(f"Categorical columns: {categorical}")
print(f"Numerical columns: {numerical}")
```

```
Categorical columns: ['CustomerID', 'Name', 'State', 'Education', 'Gender', 'Married', 'JoinDate', 'Transactio
Numerical columns: ['Age', 'NumPets', 'MonthlySpend', 'DaysSinceLastInteraction']
```

```
# Identify unique values
print('Unique values for Education:', df['Education'].unique())
print('Unique values for Gender:', df['Gender'].unique())
print('Unique values for State:', df['State'].unique())
print('Unique values for Married:', df['Married'].unique())
```

```
Unique values for Education: ['High School' 'Master' 'PhD' 'Bachelor' 'Associate']
Unique values for Gender: ['Non-Binary' 'Male' 'Female']
Unique values for State: ['Florida' 'Washington' 'Arizona' 'Texas' 'Ohio' 'New York' 'Illinois'
'Georgia' 'California' 'Colorado']
Unique values for Married: ['Yes' 'No']
```

2. Descriptive Statistics

```
print("Descriptive Statistics")

# Numerical columns: Mean, median, std dev
numerical_cols = ['Age', 'MonthlySpend', 'DaysSinceLastInteraction']
print("\nDescriptive statistics for numerical columns:")
display(df[numerical_cols].agg(['mean', 'median', 'std']))
```

```
Descriptive Statistics

Descriptive statistics for numerical columns:
```

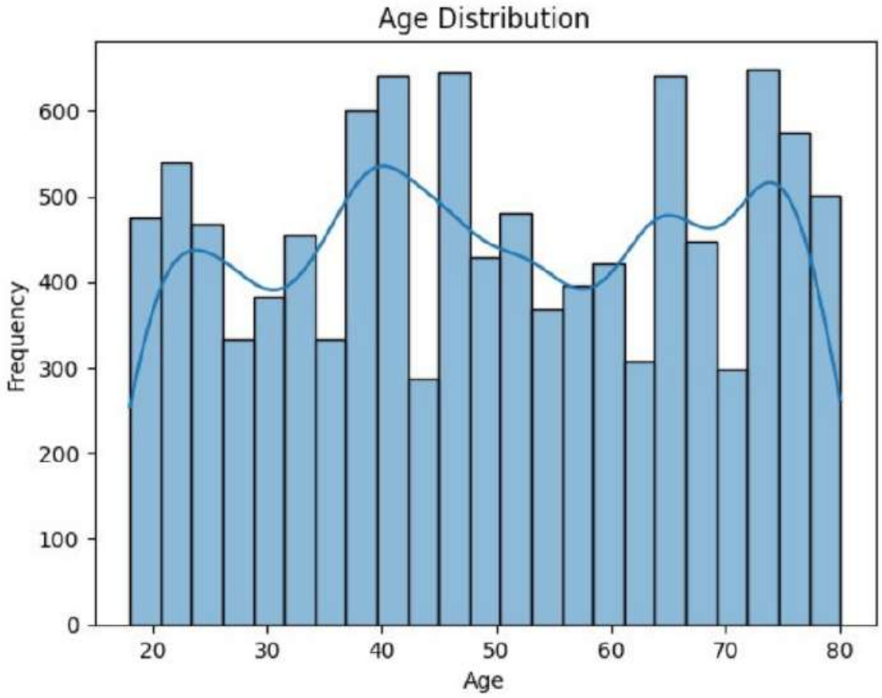
	Age	MonthlySpend	DaysSinceLastInteraction
mean	49.474567	331.610315	538.469883 445.000000 398.766747
median	49.000000	282.110000	
std	18.221365	225.799253	

```
# Categorical columns: Mode
print("Descriptive Statistics")
categorical_cols = ['Gender', 'Education', 'Married']
print("\nMode for categorical columns:")
for col in categorical_cols:
    print(f"Mode of {col}: {df[col].mode()[0]}")
```

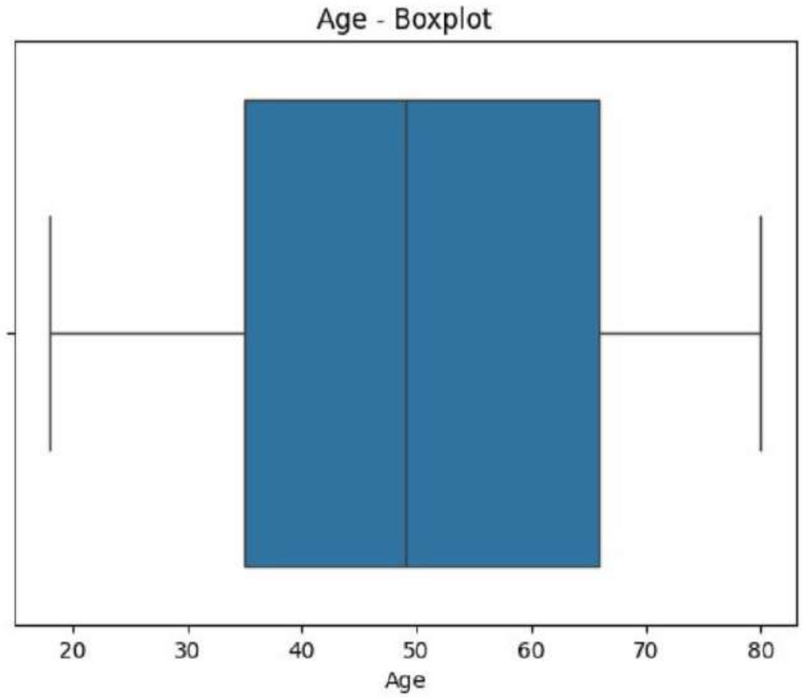
```
Descriptive Statistics

Mode for categorical columns:
Mode of Gender: Male
Mode of Education: Master
Mode of Married: No
```

3. Data Visualization

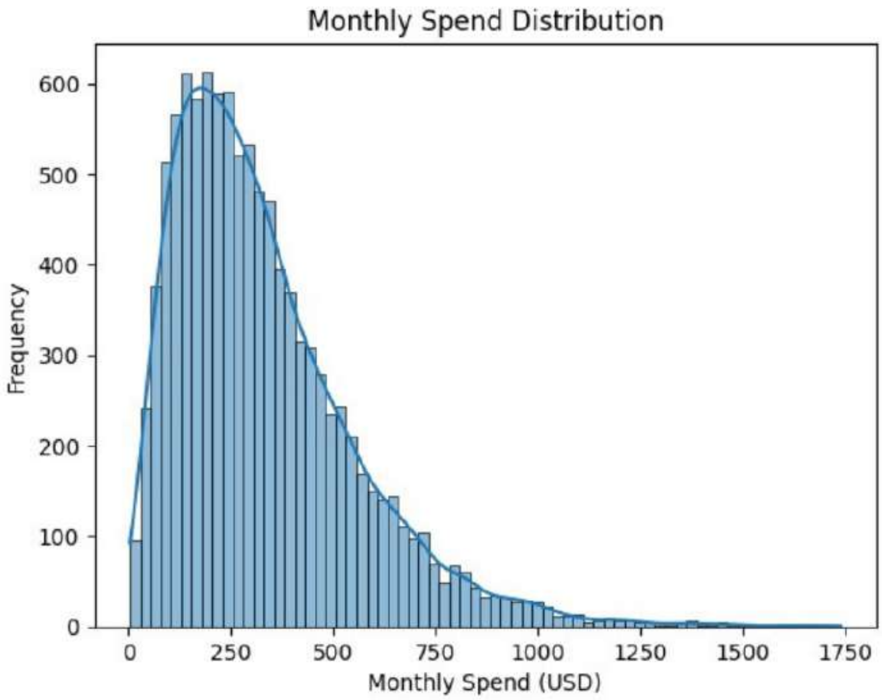
```
# Histogram for Age
sns.histplot(df['Age'], kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



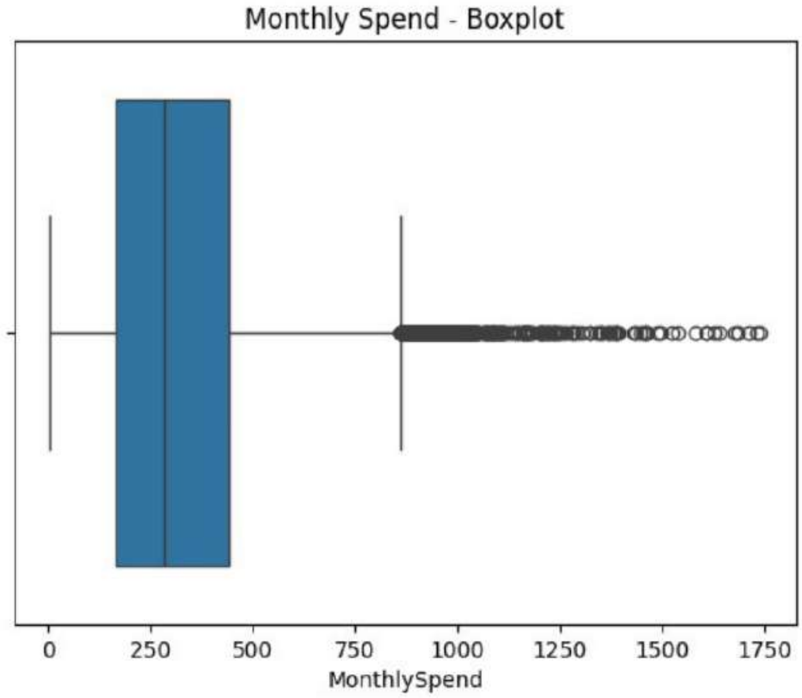

```
# Boxplot for Age
sns.boxplot(x=df['Age'])
plt.title('Age - Boxplot')
plt.show()
```



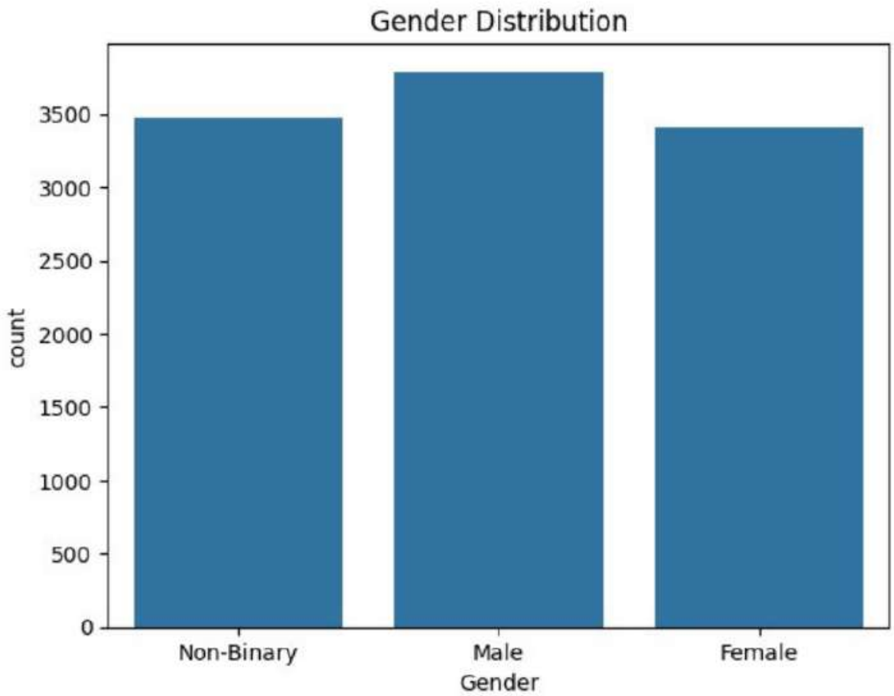

```
# Histogram for MonthlySpend
sns.histplot(df['MonthlySpend'], kde=True)
plt.title('Monthly Spend Distribution')
plt.xlabel('Monthly Spend (USD)')
plt.ylabel('Frequency')
plt.show()
```



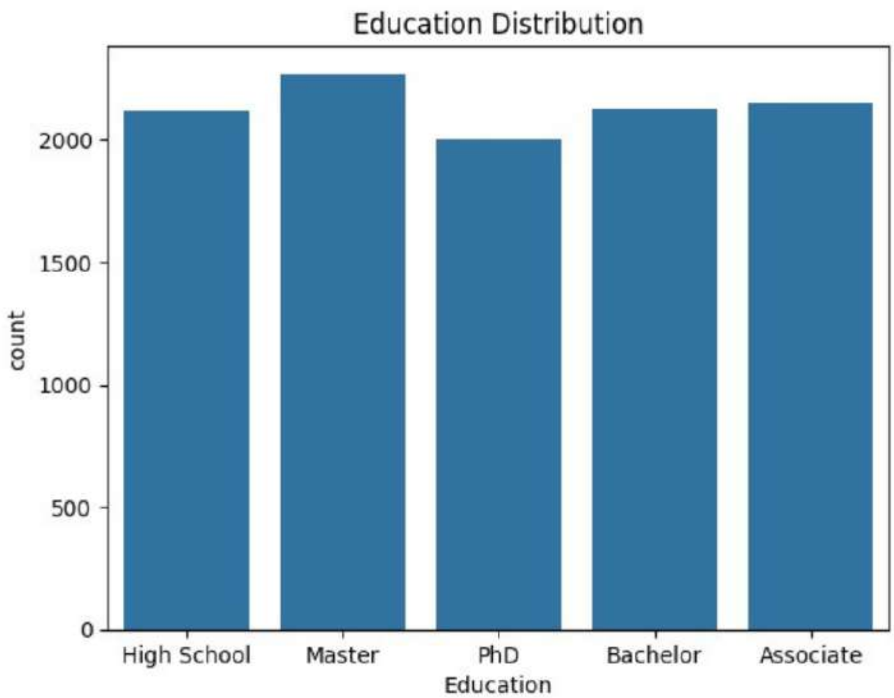

```
# Boxplot for MonthlySpend
sns.boxplot(x=df['MonthlySpend'])
plt.title('Monthly Spend - Boxplot')
plt.show()
```



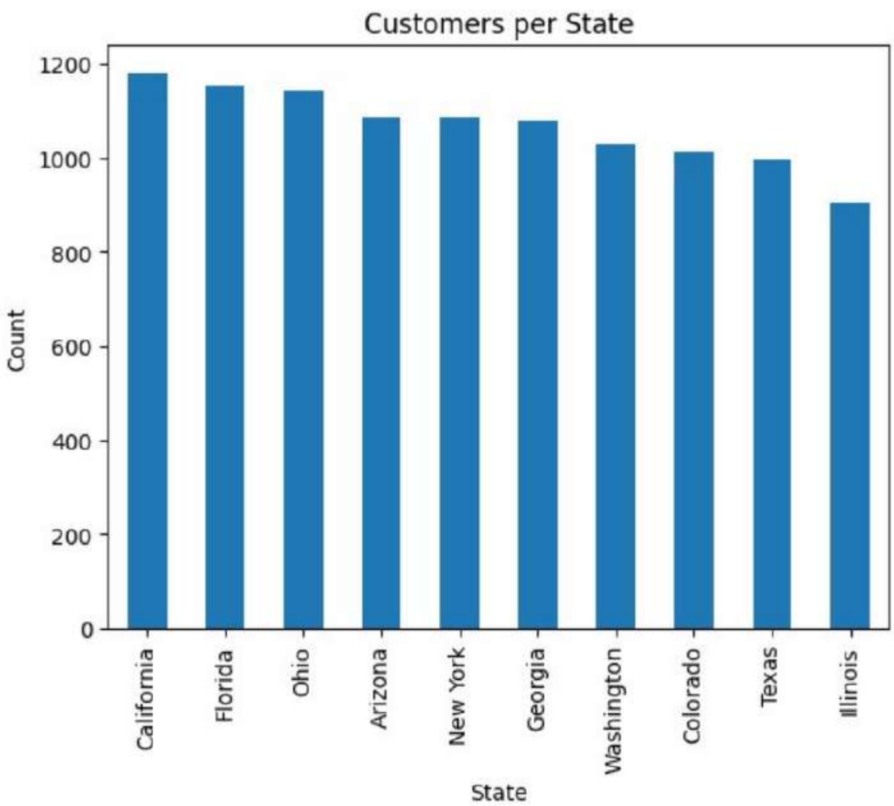

```
# Bar chart for Gender
sns.countplot(x='Gender', data=df)
plt.title('Gender Distribution')
plt.show()
```



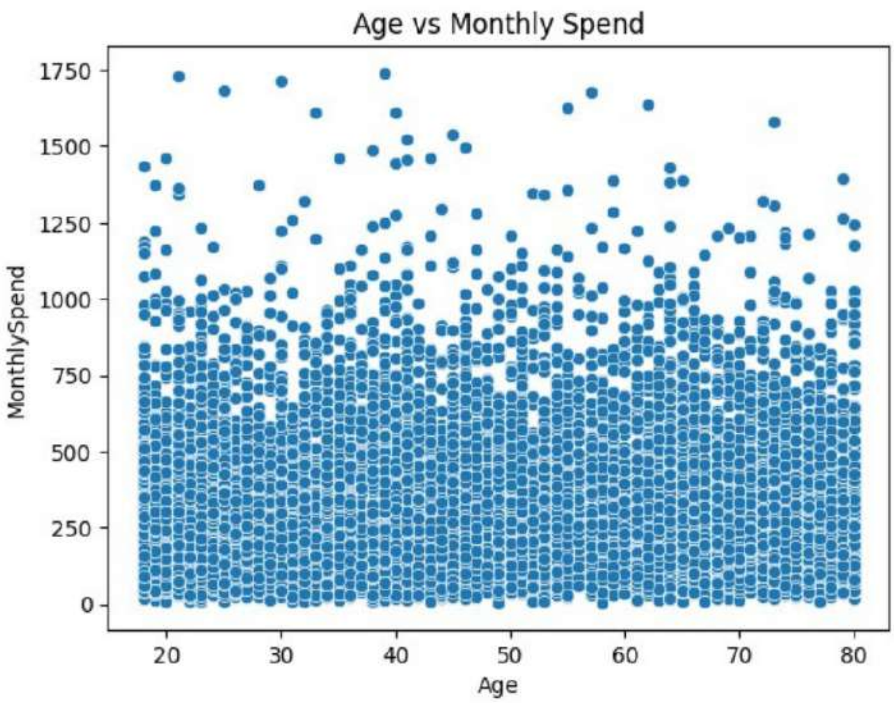

```
# Bar chart for Education
sns.countplot(x='Education', data=df)
plt.title('Education Distribution')
plt.show()
```



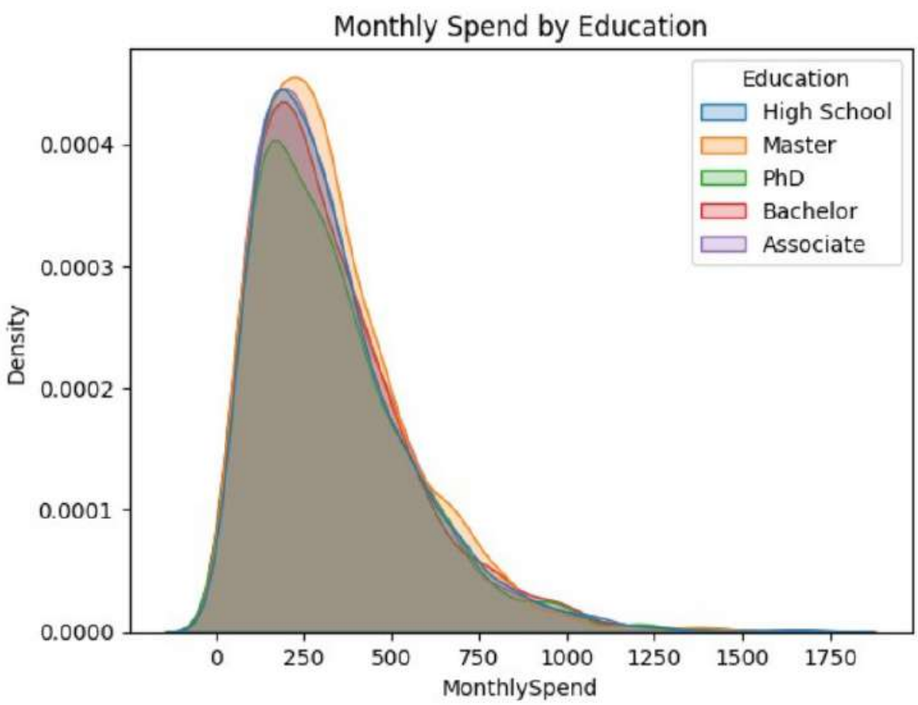

```
#       Bar       chart       for       State
df['State'].value_counts().plot(kind='bar')
plt.title('Customers       per       State')
plt.xlabel('State')      plt.ylabel('Count')
plt.show()
```



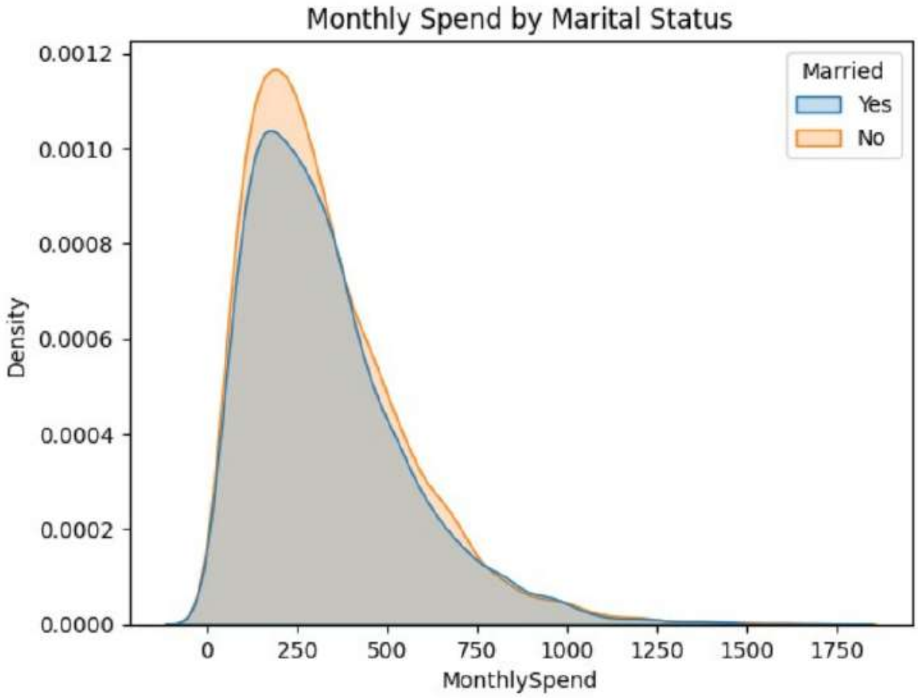

```
# Scatterplot Age vs MonthlySpend
sns.scatterplot(x='Age', y='MonthlySpend', data=df)
plt.title('Age vs Monthly Spend')
plt.show()
```




```
# KDE by Education Level, MonthlySpend
sns.kdeplot(data=df, x='MonthlySpend', hue='Education', fill=True)
plt.title('Monthly Spend by Education')
plt.show()
```




```
# KDE by Marital Status, MonthlySpend
sns.kdeplot(data=df, x='MonthlySpend', hue='Married', fill=True)
plt.title('Monthly Spend by Marital Status')
plt.show()
```



4. Bivariate Analysis

```
# Correlation matrix
print(df[numerical_cols].corr())
```

```
   Age   Age  1.000000  MonthlySpend  MonthlySpend   DaysSinceLastInteraction
 -0.012323  DaysSinceLastInteraction      -0.012323                  -0.003970
 -0.003970                                 1.000000                   0.006081
                                           0.006081                   1.000000
```

```
# Crosstab Gender vs Married
print(pd.crosstab(df['Gender'], df['Married']))
```

```
Married       No   Yes
Gender
Female      1797  1616
Male        1892  1899
Non-Binary  1894  1577
```

```
# Grouped stats: Average MonthlySpend by State, Education, Gender
print(df.groupby('State')['MonthlySpend'].mean())
print(df.groupby('Education')['MonthlySpend'].mean())
print(df.groupby('Gender')['MonthlySpend'].mean())
```

```
State
Arizona       341.489135
California    339.183492
Colorado      323.083462
Florida       327.696892
Georgia       328.354648
Illinois      332.589591
New    York   332.151244
Ohio          340.187860
Texas         319.506770
Washington    329.444078
Name: MonthlySpend, dtype: float64
Education
Associate      327.884408
Bachelor       331.884753
High School    332.215712
Master         334.252305
PhD            331.690090
Name: MonthlySpend, dtype: float64
```

```
Gender
Female       331.361310
Male         333.174068
Non-Binary   330.147240
Name: MonthlySpend, dtype: float64
```

5. Formulate Hypotheses

```
# Hypothesis 1: Age and spending
# Null Hypothesis (H0): There is no linear relationship between Age and MonthlySpend.
# Alternative Hypothesis (H1): There is a linear relationship between Age and MonthlySpend.
print("\nHypothesis 1: Age and MonthlySpend")
print("H0: There is no linear relationship between Age and MonthlySpend.")
print("H1: There is a linear relationship between Age and MonthlySpend.")
```

```
Hypothesis 1: Age and MonthlySpend
H0: There is no linear relationship between Age and MonthlySpend.
H1: There is a linear relationship between Age and MonthlySpend.
```

```
# Hypothesis 2: Gender and transaction frequency
# To analyze transaction frequency, we would typically need multiple transactions per customer.
# Since we only have 'TransactionDate' and 'JoinDate', and no explicit transaction count per customer
# we can interpret "transaction frequency" as simply having made a transaction (implied by the data e
# or focus on engagement metrics derived from the dates if possible.
# However, without multiple transactions per customer, a direct "transaction frequency" comparison by
# Let's re-interpret "transaction frequency" as average MonthlySpend for simplicity given the availab
# as spending can be a proxy for engagement/frequency in this dataset.
# Null Hypothesis (H0): The average MonthlySpend is the same across different Genders.
# Alternative Hypothesis (H1): The average MonthlySpend is different for at least one Gender.
print("\nHypothesis 2: Gender and MonthlySpend (as a proxy for transaction frequency/engagement)")
print("H0: The average MonthlySpend is the same across different Genders.")
print("H1: The average MonthlySpend is different for at least one Gender.")
```

```
Hypothesis 2: Gender and MonthlySpend (as a proxy for transaction frequency/engagement)
H0: The average MonthlySpend is the same across different Genders. H1: The average
MonthlySpend is different for at least one Gender.
```

```
# Hypothesis 3: Geography and engagement # Similar to transaction frequency, "engagement" needs a
clear definition from the data. # 'DaysSinceLastInteraction' could be a measure of recency of
engagement. # Let's test if the average 'DaysSinceLastInteraction' is the same across different
States. # Null Hypothesis (H0): The average DaysSinceLastInteraction is the same across different
States. # Alternative Hypothesis (H1): The average DaysSinceLastInteraction is different for at least
one St print("\nHypothesis 3: State and DaysSinceLastInteraction (as a proxy for engagement)")
print("H0: The average DaysSinceLastInteraction is the same across different States.") print("H1: The
average DaysSinceLastInteraction is different for at least one State.")
```

```
Hypothesis 3: State and DaysSinceLastInteraction (as a proxy for engagement)
H0: The average DaysSinceLastInteraction is the same across different States.
H1: The average DaysSinceLastInteraction is different for at least one State.
```

6. Run Hypothesis Tests

t-test: MonthlySpend by Gender (Male vs Female)

```
from scipy.stats import ttest_ind

# Filter by gender
spend_male = df[df['Gender'] == 'Male']['MonthlySpend']
```

```
spend_female = df[df['Gender'] == 'Female']['MonthlySpend']

t_stat, p_value = ttest_ind(spend_male, spend_female, nan_policy='omit')
print('T-Test Male vs Female Monthly Spend: t-stat=', t_stat, ', p-value=', p_value)
```

```
T-Test Male vs Female Monthly Spend: t-stat= 0.3391730320232445 , p-value= 0.7344892727022859
```

ANOVA: MonthlySpend by Education

```
from scipy.stats import f_oneway

edu_groups = [group['MonthlySpend'].dropna() for name, group in df.groupby('Education')]
f_stat, p_value = f_oneway(*edu_groups)
print('ANOVA Monthly Spend by Education: F-stat=', f_stat, ', p-value=', p_value)
```

```
ANOVA Monthly Spend by Education: F-stat= 0.2288066867370918 , p-value= 0.922359467759936
```

Chi-square: Marital Status vs NumPets

```
from scipy.stats import chi2_contingency

crosstab = pd.crosstab(df['Married'], df['NumPets'])
chi2, p, dof, expected = chi2_contingency(crosstab)
print('Chi-square Marital Status vs NumPets: chi2=', chi2, ', p-value=', p)
```

```
Chi-square Marital Status vs NumPets: chi2= 177.63953668537033 , p-value= 2.3957232932397494e-37
```

Correlation: Age vs DaysSinceLastInteraction

```
corr = df['Age'].corr(df['DaysSinceLastInteraction'])
print('Correlation between Age and Days Since Last Interaction:', corr)
```

```
Correlation between Age and Days Since Last Interaction: -0.003970230104955047
```

ANOVA: State-wise Monthly Spend

```
state_groups = [group['MonthlySpend'].dropna() for name, group in df.groupby('State')]
f_stat, p_value = f_oneway(*state_groups)
print('ANOVA Monthly Spend by State: F-stat=', f_stat, ', p-value=', p_value)
```

```
ANOVA Monthly Spend by State: F-stat= 1.1178423640877178 , p-value= 0.34571886479238273
```

---

7. Present Business Insights

- ***The average customer is around 49.5 years old, spends about $331.61 monthly, and their last interaction was approximately 538 days ago.***
- ***The most common gender is Male, the most common education level is Master, and most customers are not married.***
- ***The distribution of MonthlySpend is right-skewed, indicating that most customers spend less, with a few high spenders (outliers visible in the boxplot).***
- ***Customer distribution across different States and Education levels is relatively even.***
- ***There is a very weak linear relationship between Age and MonthlySpend based on the correlation analysis.***
- ***The distribution of Married status is similar across different Genders.***
- ***While average MonthlySpend varies slightly by State, Education, and Gender, these differences were not statistically significant in the hypothesis tests for Gender and MonthlySpend.***

Hypothesis testing showed no significant linear relationship between Age and MonthlySpend, and no significant difference in average MonthlySpend across Genders.
However, there is a statistically significant difference in the average DaysSinceLastInteraction across different States, suggesting that customer engagement (based on recency) varies by geography.

Data Analysis Key Findings

The dataset contains no missing values.

The distribution of MonthlySpend is right-skewed, with a few high spenders.

Customer distribution across different States and Education levels is relatively even.

The distribution of Married status is similar across different Genders.

Hypothesis testing revealed no statistically significant linear relationship between Age and MonthlySpend (p-value: 0.4992).

Hypothesis testing found no statistically significant difference in average MonthlySpend across Genders (p-value: 0.9065).

Hypothesis testing indicated a statistically significant difference in average DaysSinceLastInteraction across different States (p-value: 0.0000), suggesting geographical variation in customer engagement recency.