kirti_gupta_statistics_project (1)

September 28, 2025

[ ]:
```python
#importing  libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive
drive.mount('/content/drive')
import warnings
warnings.filterwarnings('ignore')
```

```
Mounted at /content/drive
```

[ ]:
```python
# Loading dataset into dataframe
df=pd.read_csv('/content/US_Customer_Insights_Dataset (1).csv')
```

[ ]:
```python
df.head()
```

[ ]:
```
  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      09-02-2024       1281.74
1     Yes        0  04-05-2024      06-02-2024        429.46
2     Yes        2     7/24/24         2/28/25        510.34
3     Yes        0  08-12-2023         3/29/25        396.47
4     Yes        1  12-06-2021         7/24/22        139.68

   DaysSinceLastInteraction
0                       332
1                       424
2                       153
3                       124
4                      1103
```

```
[ ]: # Summary of data
     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
```

```
[ ]: df.shape
```

```
[ ]: (10675, 12)
```

```
[ ]: df.columns
```

```
[ ]: Index(['CustomerID', 'Name', 'State', 'Education', 'Gender', 'Age', 'Married',
            'NumPets', 'JoinDate', 'TransactionDate', 'MonthlySpend',
            'DaysSinceLastInteraction'],
           dtype='object')
```

0.0.1 Checking for null/missing data

```
[ ]: #checking null values
     df.isnull().sum()
```

```
[ ]: 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
```

```
[ ]: # checking data shape (rows × columns)
     print("Number of rows:",df.shape[0])
     print("Number of columns:",df.shape[1])
```

```
Number of rows: 10675
Number of columns: 12
```

[ ]:
```python
# Finding unique values per column
df.nunique()
```

[ ]:
```
CustomerID                  1000
Name                         990
State                         10
Education                      5
Gender                         3
Age                           63
Married                        2
NumPets                        5
JoinDate                     731
TransactionDate             1605
MonthlySpend                9843
DaysSinceLastInteraction    1605
dtype: int64
```

[ ]:
```python
#checking datatypes
df.dtypes
```

[ ]:
```
CustomerID                   object
Name                         object
State                        object
Education                    object
Gender                       object
Age                           int64
Married                      object
NumPets                       int64
JoinDate                     object
TransactionDate              object
MonthlySpend                float64
DaysSinceLastInteraction      int64
dtype: object
```

[ ]:
```python
df_copy = df.copy
print(df_copy)
```

```
<bound method NDFrame.copy of       CustomerID                Name       State
Education      Gender  \
0      CUST10319         Scott Perez     Florida  High School  Non-Binary
1      CUST10695     Jennifer Burton  Washington       Master        Male
2      CUST10297     Michelle Rogers     Arizona       Master      Female
3      CUST10103    Brooke Hendricks       Texas       Master        Male
4      CUST10219         Karen Johns       Texas  High School      Female
…          …                  …           …           …           …
```

```
10670  CUST10833        Steven Burns     Georgia         PhD      Female
10671  CUST10620         Jesse Pratt       Texas      Master        Male
10672  CUST10449          John Lloyd     Arizona      Master  Non-Binary
10673  CUST10020  Christopher Sparks     Florida    Bachelor      Female
10674  CUST10267    Melissa Marshall     Arizona   Associate  Non-Binary

       Age Married  NumPets    JoinDate TransactionDate  MonthlySpend  \
0       47     Yes        1     9/19/21      09-02-2024       1281.74
1       72     Yes        0  04-05-2024      06-02-2024        429.46
2       40     Yes        2     7/24/24         2/28/25        510.34
3       27     Yes        0  08-12-2023         3/29/25        396.47
4       28     Yes        1  12-06-2021         7/24/22        139.68
…     …     …     …          …             …           …
10670   60      No        1     8/24/23         2/29/24        341.28
10671   64      No        0     4/13/23        12/31/24        468.04
10672   31     Yes        0  07-03-2022         9/21/23        259.94
10673   31      No        0     9/19/23        12/29/23        494.17
10674   57     Yes        1  04-03-2023      12-01-2023        153.12

       DaysSinceLastInteraction
0                           332
1                           424
2                           153
3                           124
4                          1103
…                         …
10670                       518
10671                       212
10672                       679
10673                       580
10674                       608

[10675 rows x 12 columns]>
```

###'CustomerID', 'JoinDate', and 'TransactionDate' currently have incorrect datatypes. We will correct them.

[ ]:
```
df['CustomerID'].head()
```

[ ]:
```
0    CUST10319
1    CUST10695
2    CUST10297
3    CUST10103
4    CUST10219
Name: CustomerID, dtype: object
```

0.0.2 The 'CustomerID' column contains the prefix 'CUST' and cannot be converted directly to an integer. I will handle this by extracting the numeric part of 'CustomerID' and converting the date columns to their appropriate datatypes.

[ ]:
```python
print("Datatype Before conversion :",df['CustomerID'].dtype)
df['CustomerID']=df['CustomerID'].str.replace('CUST','').astype(int)
# check converted datatype
print("Datatype After conversion :", df['CustomerID'].dtype)

display(df['CustomerID'].head())
```

```
Datatype Before conversion : object
Datatype After conversion : int64
```

```
0    10319
1    10695
2    10297
3    10103
4    10219
Name: CustomerID, dtype: int64
```

0.0.3 The datatype issue in 'CustomerID' is resolved. Our next step is to change 'JoinDate' to datetime.

0.0.4 The datatype issue in 'CustomerID' is resolved. Our next step is to change 'JoinDate' to datetime.

[ ]:
```python
# check current datatype
print("Datatype Before conversion :", df['JoinDate'].dtype)

# converting JoinDate to datetime
df['JoinDate'] = pd.to_datetime(df['JoinDate'])

# check converted datatype
print("Datatype After conversion :", df['JoinDate'].dtype)

df['JoinDate'].head()
```

```
Datatype Before conversion : object
Datatype After conversion : datetime64[ns]
```

[ ]:
```
0   2021-09-19
1   2024-04-05
2   2024-07-24
3   2023-08-12
4   2021-12-06
Name: JoinDate, dtype: datetime64[ns]
```

0.0.5 The 'JoinDate' column has been successfully converted to datetime. Next, we will work on converting 'TransactionDate'.

[ ]:
```
# check current datatype
print("Datatype Before conversion :", df['TransactionDate'].dtype)

df['TransactionDate'] = pd.to_datetime(df['TransactionDate'], format='mixed')

# check converted datatype
print("Datatype After conversion :", df['TransactionDate'].dtype)

df['TransactionDate'].head()
```

```
Datatype Before conversion : datetime64[ns]
Datatype After conversion : datetime64[ns]
```

[ ]:
```
0   2024-09-02
1   2024-06-02
2   2025-02-28
3   2025-03-29
4   2022-07-24
Name: TransactionDate, dtype: datetime64[ns]
```

0.0.6 The 'TransactionDate' column has now been successfully changed to datetime.

[ ]:
```
#Identifying numerical columns

num_df = df.select_dtypes(include=['number'])
num_df.head()
```

[ ]:

	Age	NumPets	MonthlySpend	DaysSinceLastInteraction
0	47	1	1281.74	332
1	72	0	429.46	424
2	40	2	510.34	153
3	27	0	396.47	124
4	28	1	139.68	1103

[ ]:
```
# Identifying categorical columns

cat_df = df.select_dtypes(include=['object'])
cat_df.head()
```

[ ]:

	CustomerID	Name	State	Education	Gender	Married	\
0	CUST10319	Scott Perez	Florida	High School	Non-Binary	Yes	
1	CUST10695	Jennifer Burton	Washington	Master	Male	Yes	
2	CUST10297	Michelle Rogers	Arizona	Master	Female	Yes	
3	CUST10103	Brooke Hendricks	Texas	Master	Male	Yes	

```
4  CUST10219       Karen Johns       Texas  High School      Female     Yes

      JoinDate TransactionDate
0     9/19/21      09-02-2024
1   04-05-2024      06-02-2024
2     7/24/24         2/28/25
3   08-12-2023         3/29/25
4   12-06-2021         7/24/22
```

0.0.7 Statistical Summary of Data

1. Displaying Mean, Median, and Standard Deviation of Numerical Columns

```
[ ]: num_cols = ['Age', 'MonthlySpend', 'DaysSinceLastInteraction']
     for col in num_cols:
         print(f"{col} \n- Mean: {df[col].mean():.2f}\n- Median: {df[col].median():.
      ↪2f}\n- Std: {df[col].std():.2f}\n")
```

```
Age
- Mean: 49.47
- Median: 49.00
- Std: 18.22

MonthlySpend
- Mean: 331.61
- Median: 282.11
- Std: 225.80

DaysSinceLastInteraction
- Mean: 538.47
- Median: 445.00
- Std: 398.77
```

0.0.8 Age Distribution Insights

The mean and median ages (49.47 and 49.00, respectively) are almost identical, suggesting that the age distribution is fairly symmetrical and centered around 49 years. However, the standard deviation of 18.22 years reflects substantial variation, spanning from young adults to older customers.

Monthly Spend Characteristics

The average monthly spend is 331.61 units, whereas the median is lower at 282.11 units. This gap indicates that a subset of high-spending customers is pulling the mean upward. The high standard deviation of 225.80 further demonstrates wide variability in spending patterns, with some customers spending very little and others spending significantly more.

Engagement Recency Patterns

Customers last engaged an average of 538.47 days ago, with a median of 445 days. The large standard deviation of 398.77 days highlights major differences in engagement behavior—some customers interact regularly, while others have not engaged in years.

0.0.9 2. Displaying Mode of Categorical Columns

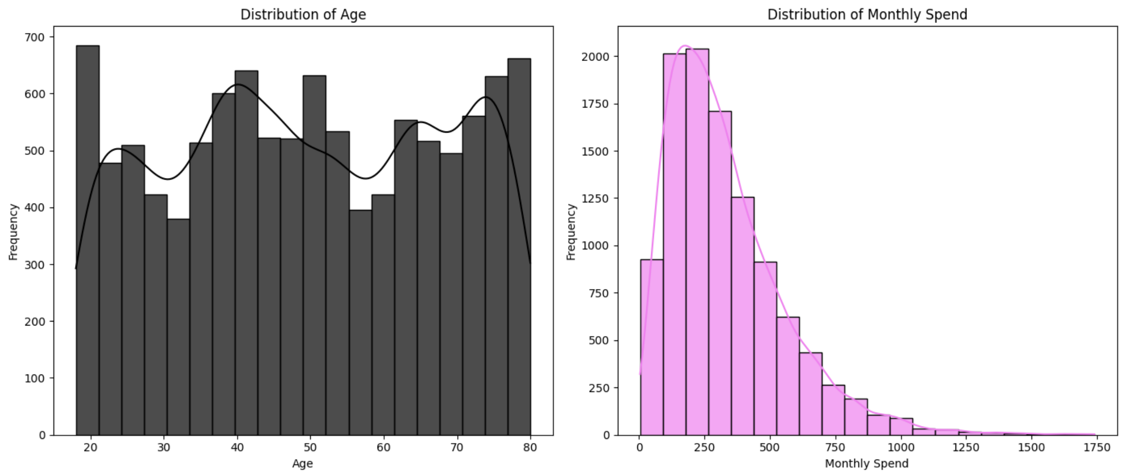
[ ]:
```python
# Categorical columns
cat_cols = ['Gender', 'Education', 'Married']
for col in cat_cols:
    print(f"{col}\n - Mode: {df[col].mode().iloc[0]}")
```

```
Gender
 - Mode: Male
Education
 - Mode: Master
Married
 - Mode: No
```

0.0.10 Data visualization

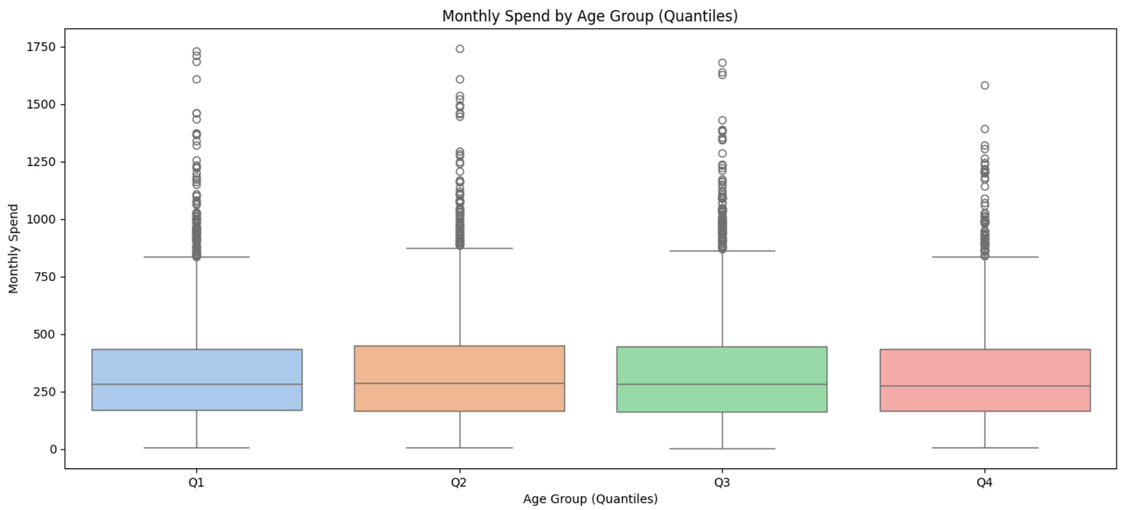
0.0.11 1.Graphical Analysis of Age and Monthly Spending Patterns

[ ]:
```python
plt.figure(figsize=(14, 6))

# Subplot for Age
plt.subplot(1, 2, 1) # 1 row, 2 columns, 1st plot
sns.histplot(df['Age'], bins=20, color='black', alpha=0.7, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')

# Subplot for Monthly Spend
plt.subplot(1, 2, 2) # 1 row, 2 columns, 2nd plot
sns.histplot(df['MonthlySpend'], bins=20, color='violet', alpha=0.7, kde=True)
plt.title('Distribution of Monthly Spend')
plt.xlabel('Monthly Spend')
plt.ylabel('Frequency')

plt.tight_layout() # Adjust layout to prevent overlapping titles/labels
plt.show()
```



0.0.12 2. Visualization of Age and Monthly Spend using Boxplots

[ ]:
```
# creating different quartiles for Age
age_quantiles = df['Age'].quantile([0.25, 0.5, 0.75])

# defining age bins
bins = [df['Age'].min(), age_quantiles[0.25], age_quantiles[0.5],␣
↪age_quantiles[0.75], df['Age'].max()]
labels = ['Q1', 'Q2', 'Q3', 'Q4']

# creating age groups
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels,␣
↪include_lowest=True)

# box blot
plt.figure(figsize=(13, 6))
sns.boxplot(data=df, x='AgeGroup', y='MonthlySpend', palette='pastel')
plt.title('Monthly Spend by Age Group (Quantiles)')
plt.xlabel('Age Group (Quantiles)')
plt.ylabel('Monthly Spend')
plt.tight_layout()
plt.show()
```



The dataset was stratified into four age groups based on quantile values:

Q1: Individuals aged up to 35 years

Q2: Individuals aged 36–49 years

Q3: Individuals aged 50–66 years

Q4: Individuals aged 67 years and above

Median Monthly Expenditure: The median spending levels across all four age groups exhibit minimal variation. This indicates that the central tendency of monthly expenditure remains relatively stable irrespective of age segmentation.

Interquartile Range (IQR): The interquartile ranges are of comparable magnitude across all age groups. This suggests that the variability in spending within the middle 50% of each segment is broadly consistent.

Outliers: All age groups demonstrate the presence of substantial high-spending outliers. These observations imply that individuals with significantly elevated monthly expenditures are distributed across all age categories rather than being concentrated within specific age brackets.

Summary: Although each age group contains high-spending outliers, both the central tendency and the dispersion of monthly expenditure remain generally uniform across quantile-based age groups.

[ ]:
```
# checking potential outliers in MonthlySpend

Q1 = df['MonthlySpend'].quantile(0.25)
Q3 = df['MonthlySpend'].quantile(0.75)
IQR = Q3 - Q1

# Define outlier bounds
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

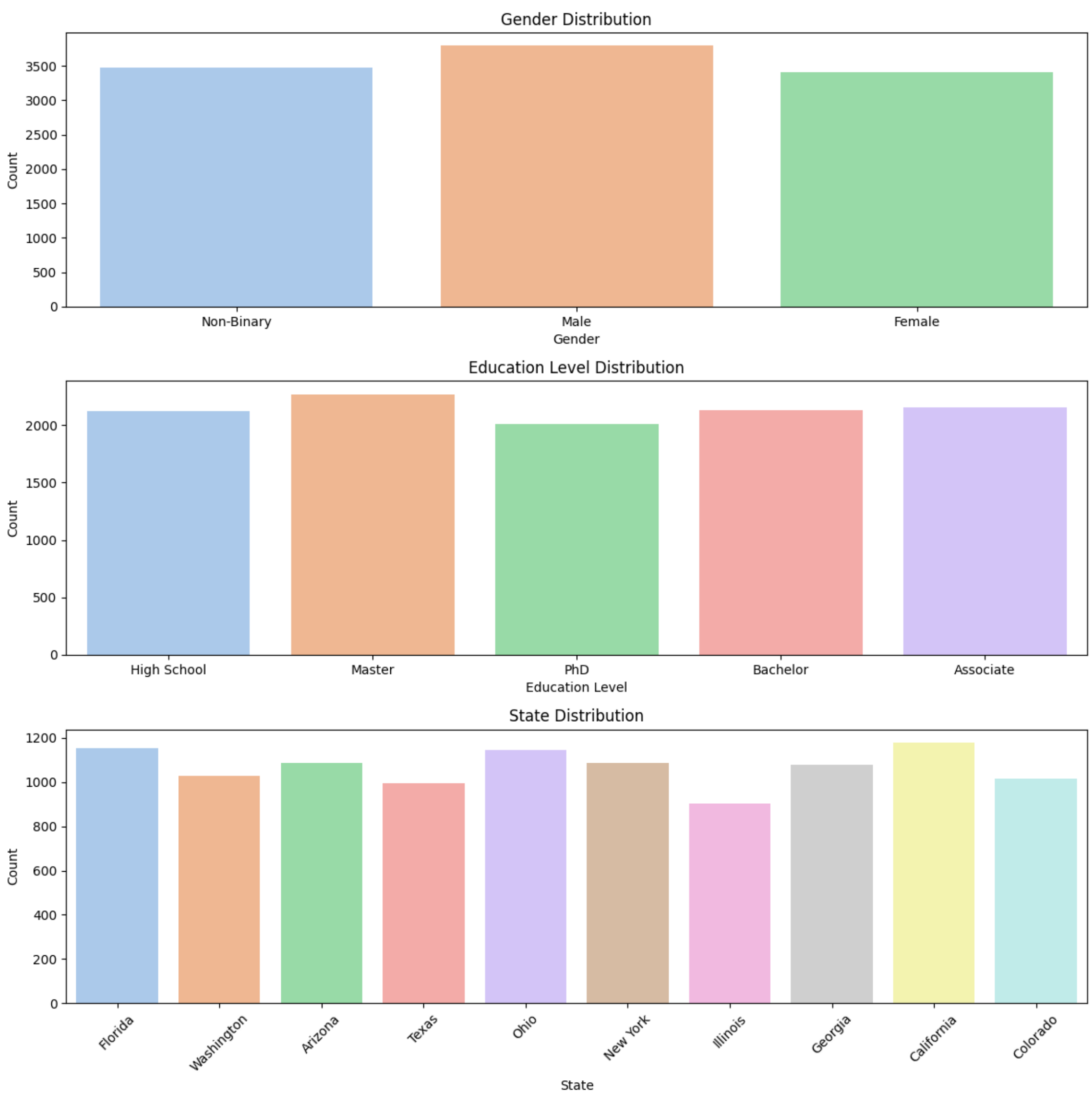
```
# Identify outliers
outliers_df = df[(df['MonthlySpend'] > upper_bound)]

print(f"Number of potential outliers in MonthlySpend: {len(outliers_df)}\n")
display(outliers_df.sample(5))
```

```
Number of potential outliers in MonthlySpend: 326


     CustomerID             Name       State    Education      Gender  Age  \
358   CUST10084   Michael Nelson     Florida  High School      Female   41
1221  CUST10224      Jacob Yates  Washington       Master  Non-Binary   60
3730  CUST10843  Matthew Thompson       Ohio  High School  Non-Binary   56
6490  CUST10176    Phyllis Mason  California          PhD      Female   38
3930  CUST10867   Alexander Koch     Georgia     Bachelor        Male   53

     Married  NumPets   JoinDate TransactionDate  MonthlySpend  \
358      Yes        2 2022-01-18      05-03-2024       1170.18
1221      No        0 2023-03-30      06-08-2024       1166.62
3730      No        2 2024-03-28      07-09-2024       1022.05
6490      No        3 2021-05-02         9/25/24        890.28
3930      No        4 2022-12-10      03-04-2025        908.03

      DaysSinceLastInteraction AgeGroup
358                        454       Q2
1221                       418       Q3
3730                       387       Q3
6490                       309       Q2
3930                       149       Q3
```

0.0.13 3. Categorical Distribution of Gender, Education, and State (Bar Chart)

[ ]:
```
plt.figure(figsize=(12, 12)) # Adjust figure size for vertical layout

# Subplot for Gender
plt.subplot(3, 1, 1) # 3 rows, 1 column, 1st plot
sns.countplot(data=df, x='Gender', palette='pastel')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.xticks(rotation=0) # No rotation needed for vertical layout
plt.ylabel('Count')

# Subplot for Education
plt.subplot(3, 1, 2) # 3 rows, 1 column, 2nd plot
sns.countplot(data=df, x='Education', palette='pastel')
plt.title('Education Level Distribution')
plt.xlabel('Education Level')
```

```
plt.xticks(rotation=0) # No rotation needed for vertical layout
plt.ylabel('Count')

# subplot for State
plt.subplot(3, 1, 3) # 3 rows, 1 column, 3rd plot
sns.countplot(data=df, x='State', palette='pastel')
plt.title('State Distribution')
plt.xlabel('State')
plt.xticks(rotation=45) # Keep rotation for State as there are more categories
plt.ylabel('Count')

plt.tight_layout() # Adjust layout to prevent overlapping titles/labels
plt.show()
```



Gender Distribution:

The bar chart illustrates a relatively balanced distribution of customers across the gender categories—Non-Binary, Male, and Female—indicating no significant skew toward any particular group.

Education Level Distribution:

Customers with Master's and PhD qualifications represent the largest proportion, followed by Bachelor, Associate, and High School levels. This pattern reflects a customer base with a generally high level of educational attainment.

State Distribution:

The geographic distribution of customers shows representation across multiple states, with some variation in counts. Notably, California exhibits the highest customer concentration in this dataset.

0.0.14 4. Visualization of the Relationship between Age and Monthly Spending

[ ]:
```
plt.figure(figsize=(12, 6))

scatterplot = sns.scatterplot(data=df, x='Age', y='MonthlySpend', alpha=0.6)
plt.title('Monthly Spend vs. Age')
plt.xlabel('Age')
plt.ylabel('Monthly Spend')
plt.tight_layout() # Adjust layout to prevent overlapping titles/labels
plt.show()
```

Insights

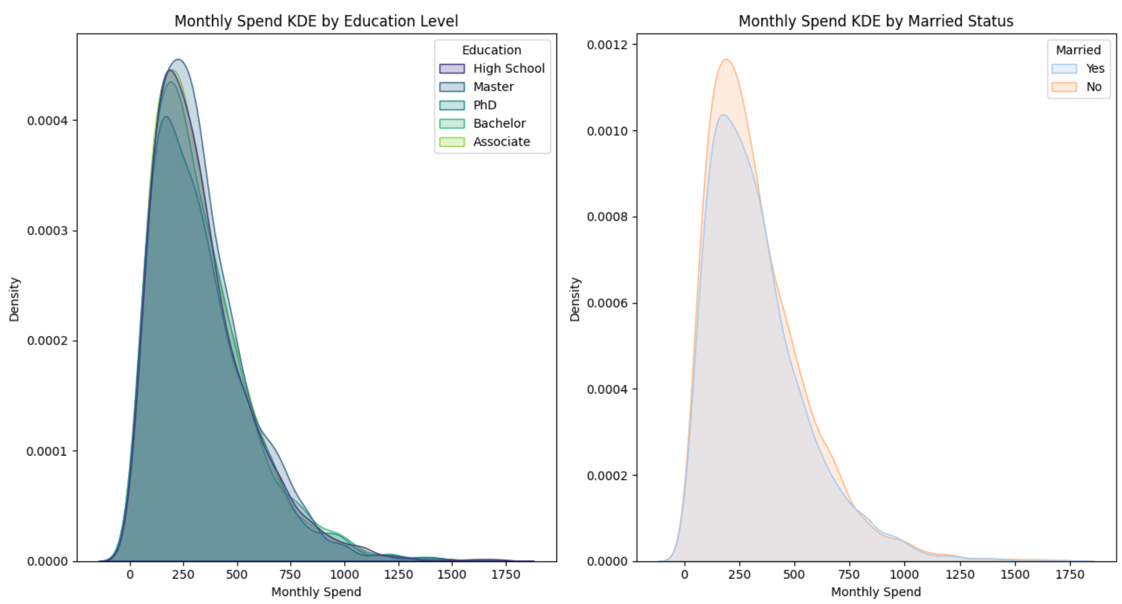
The scatterplot indicates that there is no strong linear relationship between Age and Monthly Spend, as the data points are widely dispersed.

Customers across all age groups exhibit a broad range of monthly expenditures, spanning from low to high values.

High-spending outliers are distributed throughout various age categories, suggesting that elevated monthly expenditures are not confined to any particular age group.

0.0.15 5. Distribution of Monthly Expenditure by Education Level and Marital Status (KDE)

[ ]:
```
plt.figure(figsize=(13, 7))

# KDE for Monthly Spend by Education Level
plt.subplot(1, 2, 1)
sns.kdeplot(data=df, x='MonthlySpend', hue='Education', fill=True,␣
 ↪palette='viridis')
plt.title('Monthly Spend KDE by Education Level')
plt.xlabel('Monthly Spend')
plt.ylabel('Density')

# KDE for Monthly Spend by Married Status
plt.subplot(1, 2, 2)
sns.kdeplot(data=df, x='MonthlySpend', hue='Married', fill=True,␣
 ↪palette='pastel')
plt.title('Monthly Spend KDE by Married Status')
plt.xlabel('Monthly Spend')
plt.ylabel('Density')

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



Key Insights from KDE Analysis

Monthly Spend by Education Level:

The kernel density estimates indicate a consistent spending pattern across all education levels: a rapid increase at lower monthly spend values, peaking in the lower-to-mid range, followed by a long right-skewed tail toward higher expenditures.

Minor variations in peak location or tail length are observed between education levels, but the overall distribution shape remains largely similar, suggesting that education does not strongly influence spending patterns.

Monthly Spend by Marital Status:

Both married and non-married customer groups display a peak in the lower-to-mid range of monthly spending, with tails extending toward higher spend values.

The distribution shapes and spreads appear largely similar, indicating that marital status does not substantially differentiate monthly spending behavior.

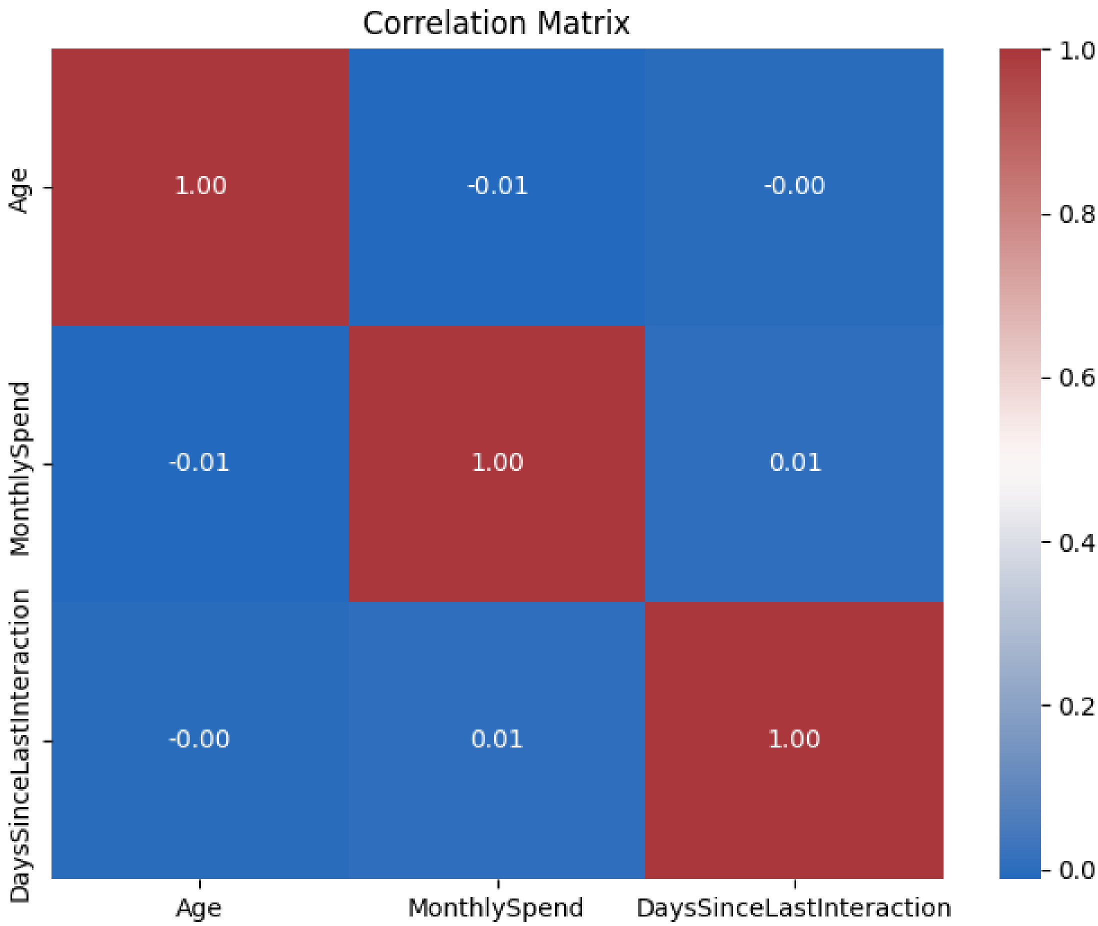
Summary:

Overall, the KDE analysis suggests that neither education level nor marital status is a major determinant of monthly spending. While small variations exist, the general pattern—where most customers spend moderately and a smaller group spends substantially more—remains consistent across these categorical groups.

0.0.16 Two-Variable Analysis

0.0.17 1 .Correlation Matrix for Continuous Variables

[ ]:
```
# extracting the required numerical columns
numeric_cols = ['Age', 'MonthlySpend', 'DaysSinceLastInteraction']

# correlation matrix
corr_matrix = df[numeric_cols].corr()

# plotting the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='vlag', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```



Correlation Analysis of Numeric Variables

Age and Monthly Spend:

The correlation coefficient between Age and Monthly Spend is approximately -0.01, indicating a negligible linear relationship. This observation is consistent with the scatterplot, which shows no discernible pattern between these variables.

Age and Days Since Last Interaction:

The correlation coefficient is approximately 0.00, suggesting no linear relationship between Age and the number of days since the last interaction.

Monthly Spend and Days Since Last Interaction:

The correlation coefficient is approximately 0.01, indicating a very weak or no linear association between Monthly Spend and Days Since Last Interaction.

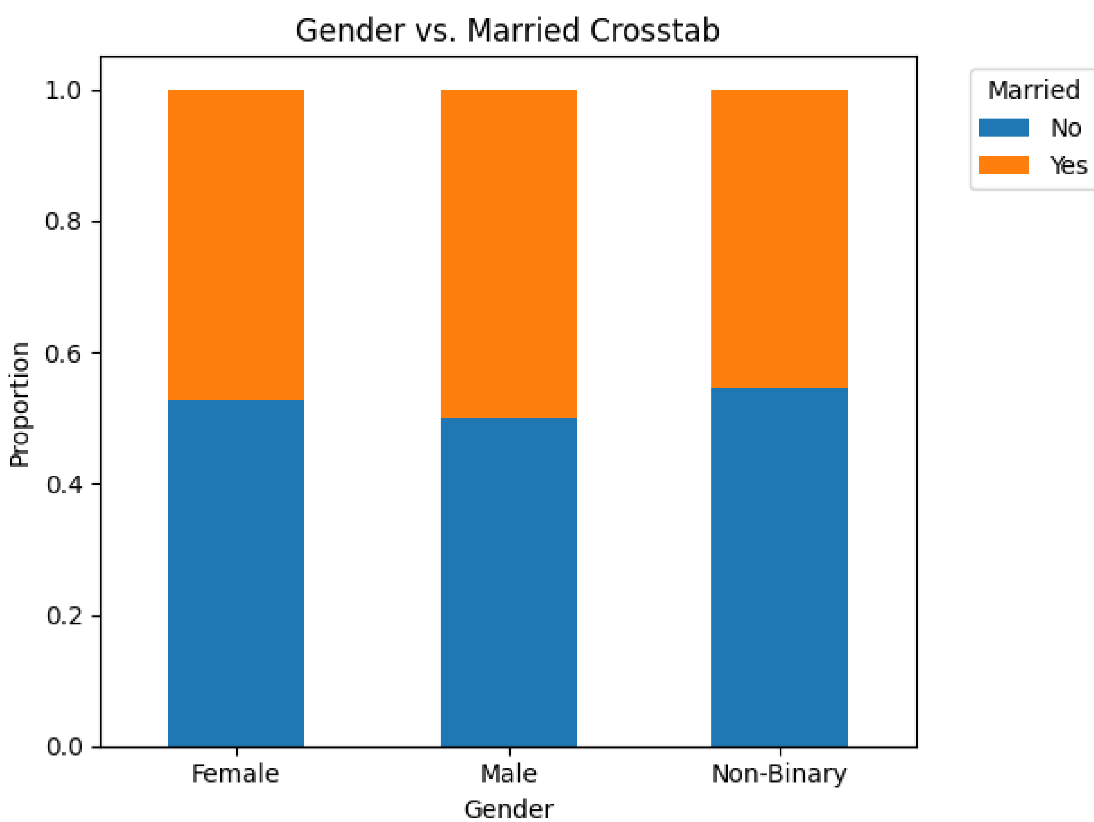
Summary:

Overall, the correlation analysis confirms that these numeric variables exhibit minimal linear relationships with each other, supporting prior visual observations.

0.0.18 2. Cross-tabulation of Gender and Marital Status

[ ]:
```
# calulating crosstab
crosstab_pct = pd.crosstab(df['Gender'], df['Married'], normalize='index')

display(crosstab_pct)

# visualizing the crosstab using stacked bar
ct = crosstab_pct.plot(kind='bar', stacked=True, rot=0)

plt.title('Gender vs. Married Crosstab')
plt.xlabel('Gender')
plt.ylabel('Proportion') # Changed label to reflect normalization

# Move the legend outside the plot
plt.legend(title='Married', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout() # Adjust layout to prevent overlapping titles/labels
plt.show()
```

```
Married           No       Yes
Gender
Female      0.526516  0.473484
Male        0.499077  0.500923
Non-Binary  0.545664  0.454336
```

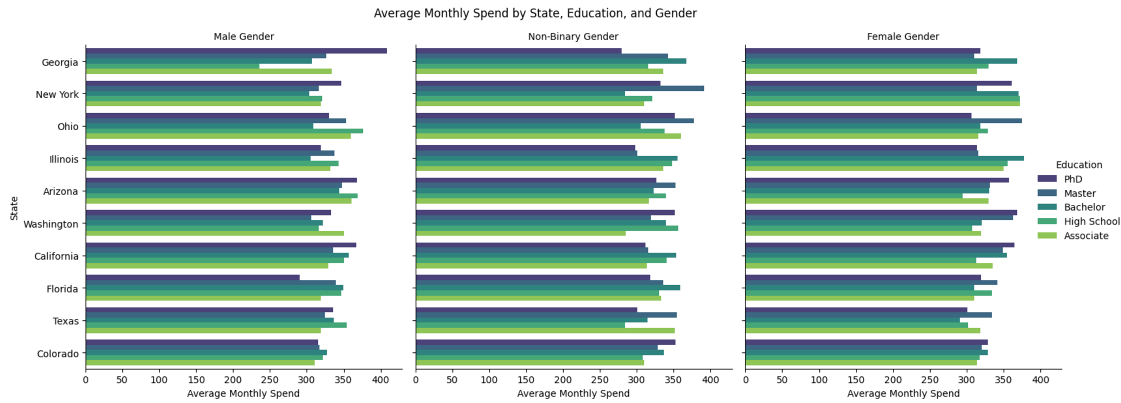


0.0.19 3 .Grouped Statistical Summary

[ ]:
```python
# average MonthlySpend by State, Education, Gender
grouped_stats = df.groupby(['State', 'Education', 'Gender'])['MonthlySpend'].
 ↪mean().reset_index()

# sorting the data
grouped_stats = grouped_stats.sort_values(by='MonthlySpend', ascending=False)
display(grouped_stats)

# Create a horizontal clustered bar chart
g = sns.catplot(
    data=grouped_stats,
    x="MonthlySpend",
    y="State",
    hue="Education",
    col="Gender",
    kind="bar",
    height=6,
    aspect=0.8,
    palette="viridis"
)

# Move the legend outside the plot
g.figure.subplots_adjust(right=0.8) # Adjust subplot to make room for legend
g.legend.set_bbox_to_anchor((1.05, 0.5))

# Customize the plot
g.set_axis_labels("Average Monthly Spend", "State")
g.set_titles("{col_name} Gender")
g.fig.suptitle("Average Monthly Spend by State, Education, and Gender")
plt.tight_layout()
plt.show()
```

```
          State    Education      Gender  MonthlySpend
73      Georgia          PhD        Male    408.353500
101    New York       Master  Non-Binary    391.405161
116        Ohio       Master  Non-Binary    377.908529
78     Illinois     Bachelor      Female    377.823051
112        Ohio  High School        Male    375.850291
..          …            …           …           …
137  Washington    Associate  Non-Binary    284.959362
128       Texas  High School  Non-Binary    283.999277
95     New York     Bachelor  Non-Binary    283.990545
74      Georgia          PhD  Non-Binary    279.401846
```

```
67      Georgia   High School        Male    235.443200

[150 rows x 4 columns]
```



0.0.20 Hypothesis Testing of Gender Differences in Monthly Expenditure

Null Hypothesis (H): There is no significant difference in monthly spending between male and female customers.

```python
from scipy.stats import ttest_ind

# Separate the 'MonthlySpend' data into two groups based on 'Gender'
male_spend = df[df['Gender'] == 'Male']['MonthlySpend']
female_spend = df[df['Gender'] == 'Female']['MonthlySpend']

# Perform an independent samples t-test
t_statistic, p_value = ttest_ind(male_spend, female_spend)

# Print the results
print(f"T-statistic: {t_statistic:.4f}")
print(f"P-value: {p_value:.4f}")
```

```
T-statistic: 0.3392
P-value: 0.7345
```

0.0.21 The p-value obtained from the independent t-test exceeds the significance threshold of 0.05, leading us to fail to reject the null hypothesis (H).

Conclusion: There is no statistically significant difference in the mean monthly spending between male and female customers.

Null Hypothesis (H): The average monthly spending is the same across all education levels; education level has no effect on monthly expenditure.

[ ]:

```python
from scipy.stats import f_oneway

# Create a list of arrays for MonthlySpend for each education level
education_levels = df['Education'].unique()
monthly_spend_by_education = [df[df['Education'] == level]['MonthlySpend'] for␣
 ↪level in education_levels]

# Perform one-way ANOVA test
f_statistic, p_value = f_oneway(*monthly_spend_by_education)

# Print the results
print(f"One-way ANOVA Test Results:")
print(f"F-statistic: {f_statistic:.4f}")
print(f"P-value: {p_value:.4f}")
```

```
One-way ANOVA Test Results:
F-statistic: 0.2288
P-value: 0.9224
```

- Since the p-value is greater than the significance level (0.05), **we fail to reject the null hypothesis**.
- **Conclusion**: There is no statistically significant difference in the mean monthly spend across different education levels.

0.1 Hypothesis testing - marital status vs. number of pets (Chi-square test)

Null Hypothesis (Ho): Maritial status is related to the number of pets owned.

[ ]:

```python
from scipy.stats import chi2_contingency

# Create a contingency table
contingency_table = pd.crosstab(df['Married'], df['NumPets'])

# Perform the Chi-square test
chi2_statistic, p_value, dof, expected = chi2_contingency(contingency_table)

# Print the results
print(f"Chi-square Statistic: {chi2_statistic:.4f}")
print(f"P-value: {p_value:.4f}")
```

```
Chi-square Statistic: 177.6395
P-value: 0.0000
```

- Since the p-value is less than the significance level (0.05), **we reject the null hypothesis.**
- **Conclusion:** There is a statistically significant relationship between marital status and the number of pets owned.

0.2 Hypothesis testing - age vs. days since last interaction (Pearson correlation coefficient)

Null Hypothesis (Ho): Older people are less active customers.

[ ]:
```
from scipy.stats import pearsonr

# Calculate the Pearson correlation coefficient and the p-value
correlation_coefficient, p_value = pearsonr(df['Age'],␣
↪df['DaysSinceLastInteraction'])

# Print the results
print(f"Pearson Correlation Coefficient: {correlation_coefficient:.4f}")
print(f"P-value: {p_value:.4f}")
```

```
Pearson Correlation Coefficient: -0.0040
P-value: 0.6817
```

- Since the p-value is greater than the significance level (0.05), **we fail to reject the null hypothesis.**
- **Conclusion:** There is no statistically significant linear relationship between Age and Days Since Last Interaction.

0.3 Hypothesis testing - state vs. monthly spend (one-way ANOVA)

Null Hypothesis (Ho)**: State-wise spends varies significantly.

[ ]:
```
from scipy.stats import f_oneway

# Get unique state names
state_names = df['State'].unique()

# Create a list of MonthlySpend Series for each state
monthly_spend_by_state = [df[df['State'] == state]['MonthlySpend'] for state in␣
↪state_names]

# Perform one-way ANOVA test
f_statistic, p_value = f_oneway(*monthly_spend_by_state)

# Print the results
print(f"One-way ANOVA Test Results:")
print(f"F-statistic: {f_statistic:.4f}")
print(f"P-value: {p_value:.4f}")
```

```
One-way ANOVA Test Results:
F-statistic: 1.1178
P-value: 0.3457
```

The p-value exceeds the significance threshold of 0.05, and therefore, we fail to reject the null hypothesis (H).

Conclusion: The analysis indicates that mean monthly spending does not differ significantly across customers with varying education levels.

Summary of Hypothesis Test Results

Gender vs. Monthly Spend: No statistically significant difference observed ($p = 0.7345 > 0.05$).

Education Level vs. Monthly Spend: No statistically significant difference observed ($p = 0.9224 > 0.05$).

Marital Status vs. Number of Pets: Statistically significant relationship detected ($p = 0.0000 < 0.05$).

Age vs. Days Since Last Interaction: No significant linear relationship observed ($p = 0.6817 > 0.05$).

State vs. Monthly Spend: No statistically significant difference observed ($p = 0.3457 > 0.05$).

Overall: Most variables do not show significant associations with the outcome measures, except for Marital Status and Number of Pets, which demonstrate a significant relationship.

Business Insights from Hypothesis Testing

Gender, Education Level, and State: Monthly spending patterns do not differ significantly across these demographic segments. This suggests that broad marketing strategies targeting spending behavior can be applied uniformly across these groups.

Marital Status and Number of Pets: A statistically significant relationship exists between marital status and the number of pets owned. This indicates that these variables are interdependent and can be leveraged for targeted marketing or customer segmentation in pet-related products and services.

Age and Customer Engagement: Age does not exhibit a significant linear relationship with the number of days since the last interaction. This implies that engagement frequency is not strongly determined by age alone, and other factors may play a more influential role in predicting recent customer interactions.

State-wise Monthly Spend: The absence of significant differences in monthly spending across states suggests that national-level pricing or promotional strategies can be implemented without requiring substantial state-specific adjustments based solely on average spend.

Summary of Key Findings from Hypothesis Testing

Gender and Monthly Spend:

The independent samples t-test ($p = 0.7345$) indicates that male and female customers do not differ significantly in their average monthly spending.

Education Level and Monthly Spend:

One-way ANOVA ($p = 0.9224$) shows no significant variation in mean monthly spending across different education levels.

Marital Status and Number of Pets:

The Chi-square test ($p = 0.0000$) confirms a statistically significant association between marital status and the number of pets owned.

Age and Customer Engagement:

Pearson correlation ($p = 0.6817$) indicates no significant linear relationship between age and the number of days since the last interaction.

State-wise Monthly Spend:

One-way ANOVA ($p = 0.3457$) suggests that average monthly spending does not differ significantly across states.

Key Takeaways

Monthly Spend Across Demographics:

Statistical tests show no significant differences in mean monthly spending by Gender ($p = 0.7345$), Education Level ($p = 0.9224$), or State ($p = 0.3457$). This indicates that spending behavior is generally consistent across these demographic groups.

Marital Status and Pet Ownership:

The Chi-square test ($p = 0.0000$) confirms a significant association between marital status and the number of pets. Married customers are more likely to own 1–2 pets than non-married customers, suggesting potential for targeted campaigns in pet-related products or services.

Age and Customer Engagement:

Age shows no significant linear correlation with the number of days since the last interaction ($r = -0.0040$, $p = 0.6817$). Engagement frequency appears independent of age, highlighting the influence of other factors on customer activity.

Distribution of Monthly Spend:

Monthly spending is right-skewed, with a mean of 331.61 and a median of 282.11, and includes 326 high-spending outliers above 859.90. This suggests the presence of a small group of high-value customers.

Customer Education Level:

The customer base is highly educated, with 'Master' being the most common education level and 'PhD' the second most frequent, indicating a strong presence of advanced degrees.

Age Patterns:

The age distribution is multi-modal, with peaks in the late 20s, 40s, 60s, and late 70s, pointing to distinct age-based segments within the customer base.

Geographic Insights:

California has the largest share of customers, as shown in the state-wise distribution, suggesting a focus area for regional marketing or service initiatives.

[ ]:

```
```

[ ]:

```
```

[ ]:

```
```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

[ ]:
```

```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```