Unlocking Customer Insights: A Statistical Investigation

1.Understand Data

In [1]:
```
import pandas as pd
import numpy as np
path = "/content/stats.csv"
df = pd.read_csv(path)
df.head()
print("Shape:", df.shape)
print("\nDtypes:\n", df.dtypes)

nulls = df.isna().sum().sort_values(ascending=False)
print("\nMissing values:\n", nulls)
```

```
Shape: (10675, 12)

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

Missing values:
 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
```

In [2]:
```
uniques = df.nunique(dropna=True).sort_values(ascending=False)
print("\nUnique counts:\n", uniques)
```

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

In [3]:
```python
# Separate columns by type
numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
categorical_cols = df.select_dtypes(exclude=[np.number]).columns.tolist()

print("Numeric columns:", numeric_cols)
print("Categorical columns:", categorical_cols)
```

```
Numeric columns: ['Age', 'NumPets', 'MonthlySpend', 'DaysSinceLastInteraction']
Categorical columns: ['CustomerID', 'Name', 'State', 'Education', 'Gender', 'Marr
ied', 'JoinDate', 'TransactionDate']
```

2: Descriptive Statistics

1) Numeric variables (Age, MonthlySpend, DaysSinceLastInteraction)

In [4]:
```python
# Descriptive stats for numeric variables
num_summary = df[['Age','MonthlySpend','DaysSinceLastInteraction']].agg(
    ['mean','median','std']
).T

num_summary
```

Out[4]:

	mean	median	std
Age	49.474567	49.00	18.221365
MonthlySpend	331.610315	282.11	225.799253
DaysSinceLastInteraction	538.469883	445.00	398.766747

2) Categorical variables (Gender, Education, Married) → mode

In [5]:
```python
# Mode for categorical variables
categorical_cols = ['Gender', 'Education', 'Married']

print("=== Mode (Most Frequent Value) - Categorical ===\n")
for col in categorical_cols:
    mode_val = df[col].mode()
    if not mode_val.empty:
        print(f"{col}: {mode_val[0]}")
    else:
        print(f"{col}: No mode (or all missing)")
```

```
=== Mode (Most Frequent Value) - Categorical ===

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

1. **Age:**
   - Customers are on average **about 49 years old**.
   - The majority of the customer base falls in the **middle-age group (40–50 years)**.
2. **Spending:**
   - On average, customers spend **~$330 per month**.
   - Spending varies a lot, since the standard deviation is quite high.
3. **Activity:**
   - On average, customers had their **last interaction 1.5 years (538 days) ago**.
   - This means they are **not very active**, with large gaps in engagement.
4. **Most Common Profile:**
   - **Male**
   - **Master's Degree**
   - **Not Married**

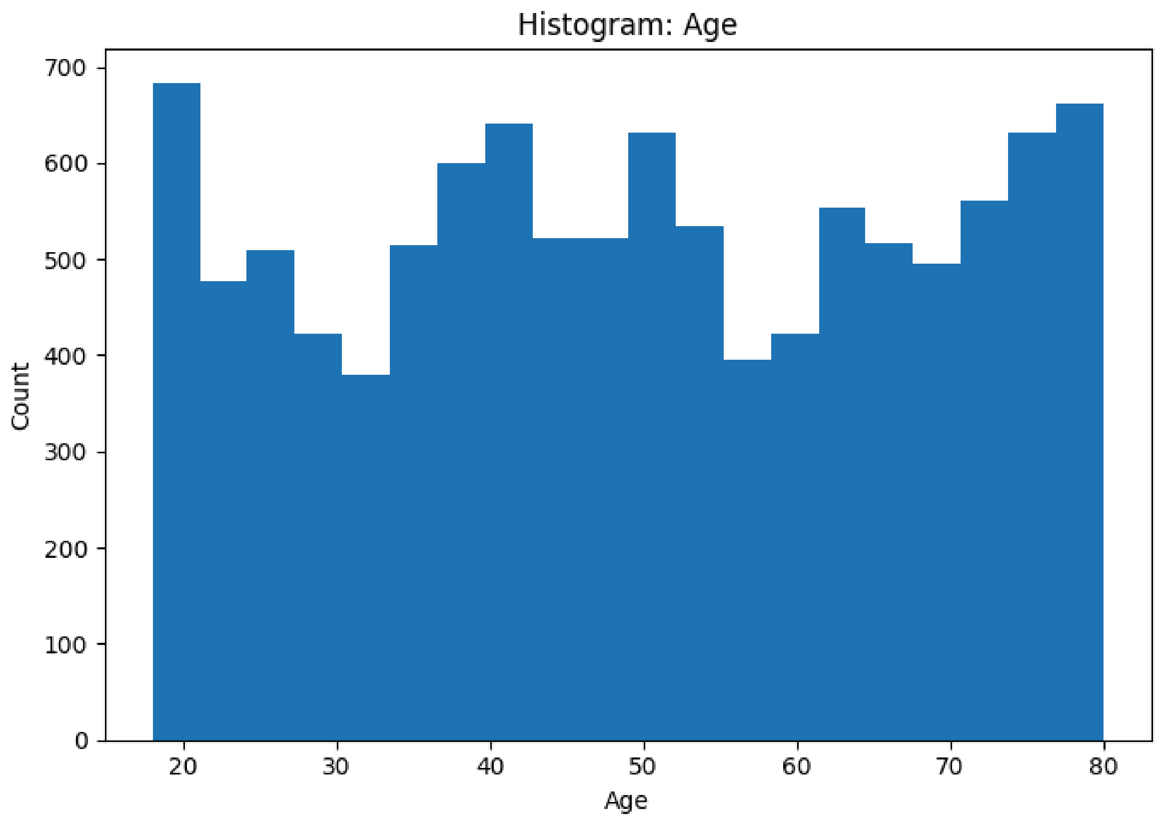
Conclusion: Your customers are mostly **middle-aged males with a Master's degree, unmarried, and mid-level spenders**. However, they are **not very active**, which suggests that stronger **engagement strategies** are needed to keep them involved.

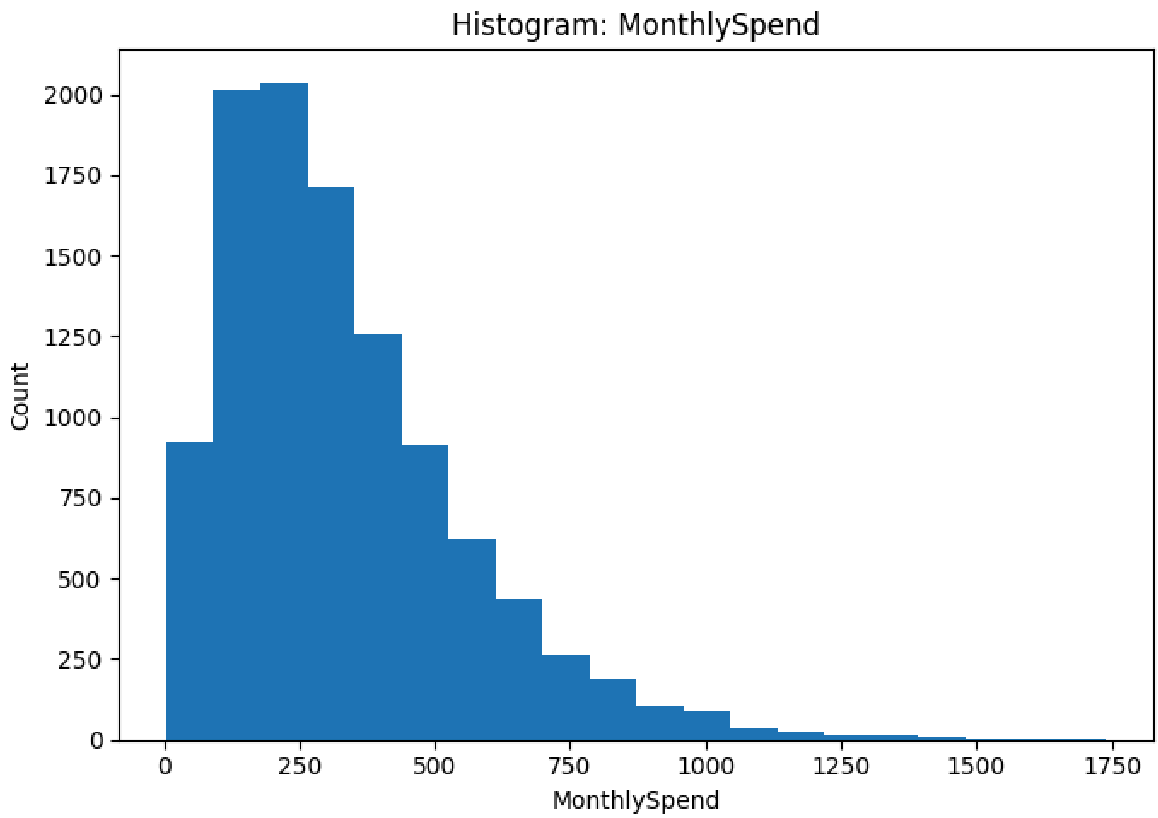
3: Data Visualization

1) Histograms: Age, MonthlySpend

In [32]:

```
import matplotlib.pyplot as plt
for col in ["Age", "MonthlySpend"]:
    if col in df.columns:
        plt.figure(figsize=(7,5))
        df[col].dropna().plot(kind="hist", bins=20)
        plt.title(f"Histogram: {col}")
        plt.xlabel(col); plt.ylabel("Count")
        plt.tight_layout()
        plt.show()
    else:
        print(f"Column not found for histogram: {col}")
```

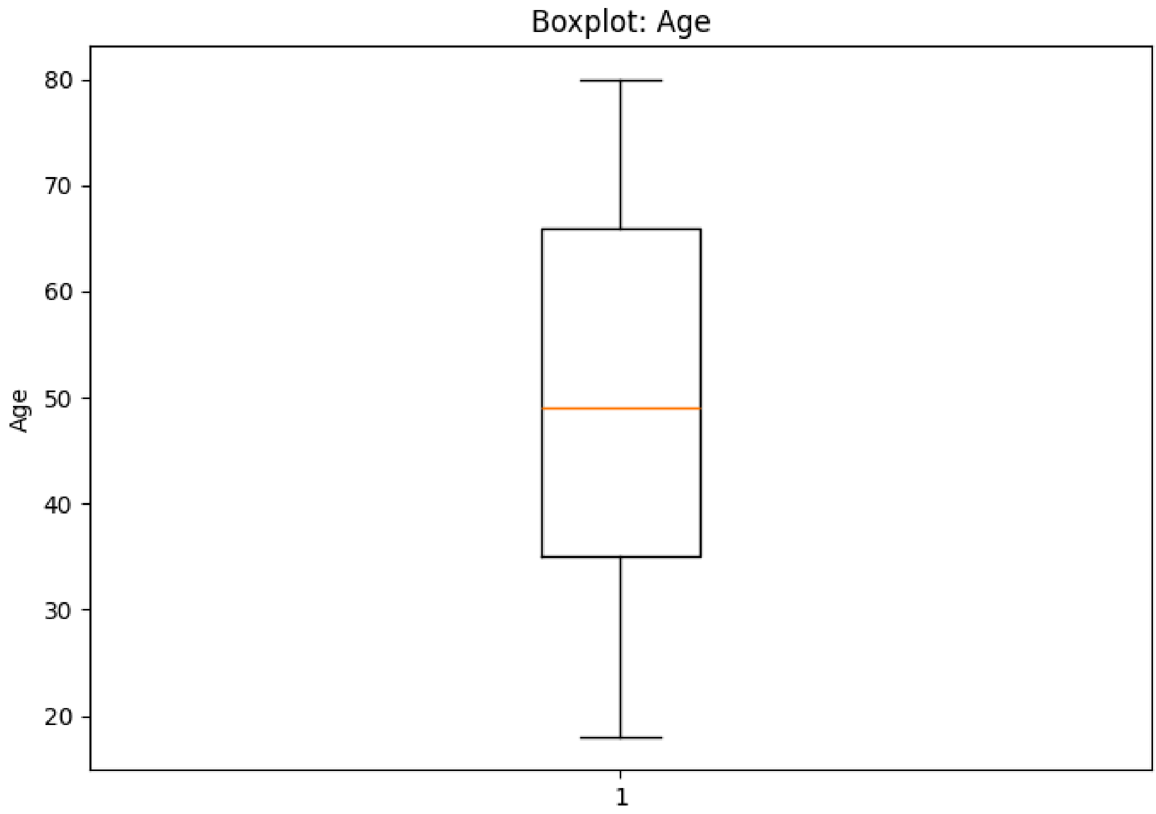


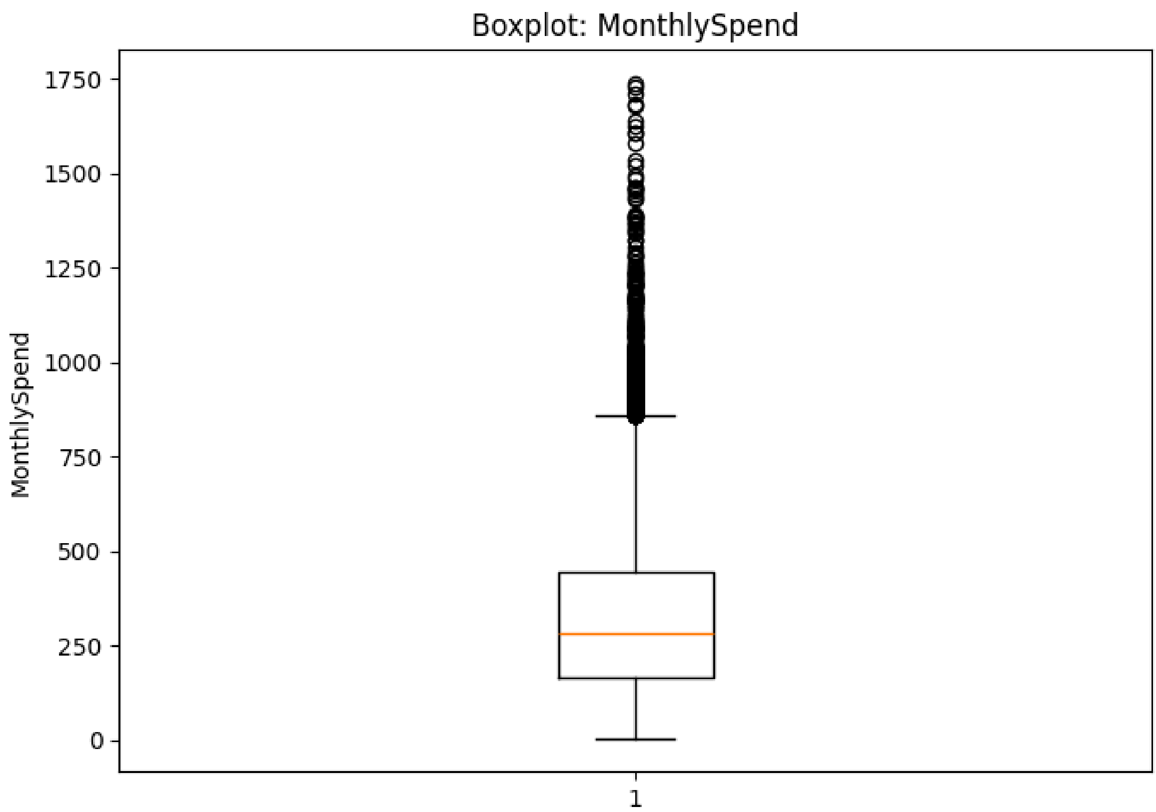


In [31]:

```
for col in ["Age", "MonthlySpend"]:
    if col in df.columns:
        plt.figure(figsize=(7,5))
        plt.boxplot(df[col].dropna(), vert=True)
        plt.title(f"Boxplot: {col}")
        plt.ylabel(col)
        plt.tight_layout()
        plt.show()
```

```
    else:
        print(f"Column not found for boxplot: {col}")
```

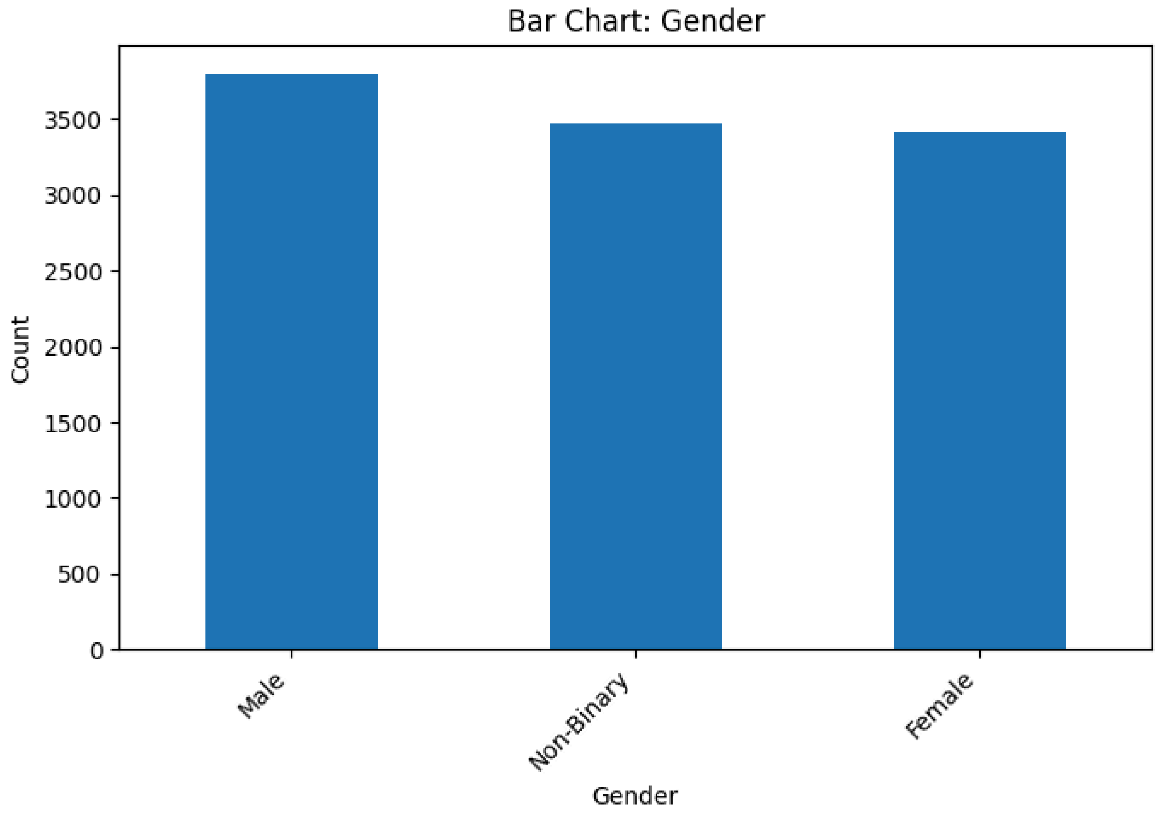


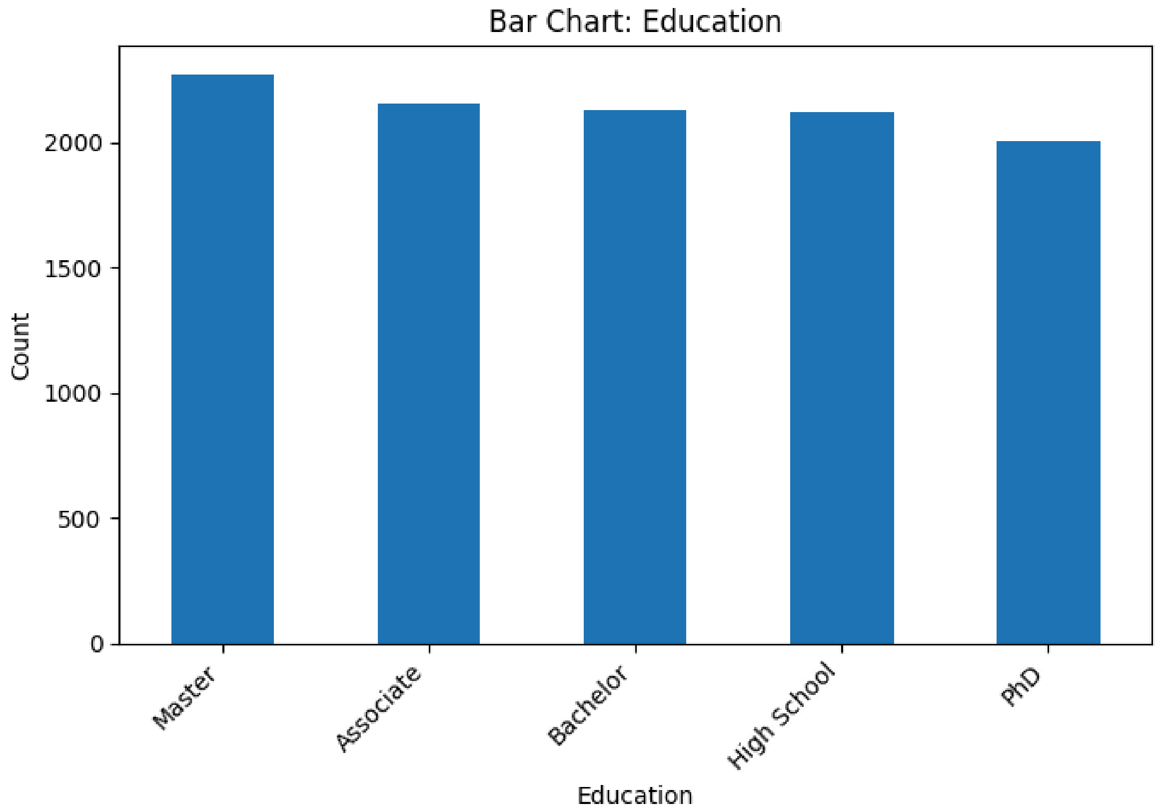


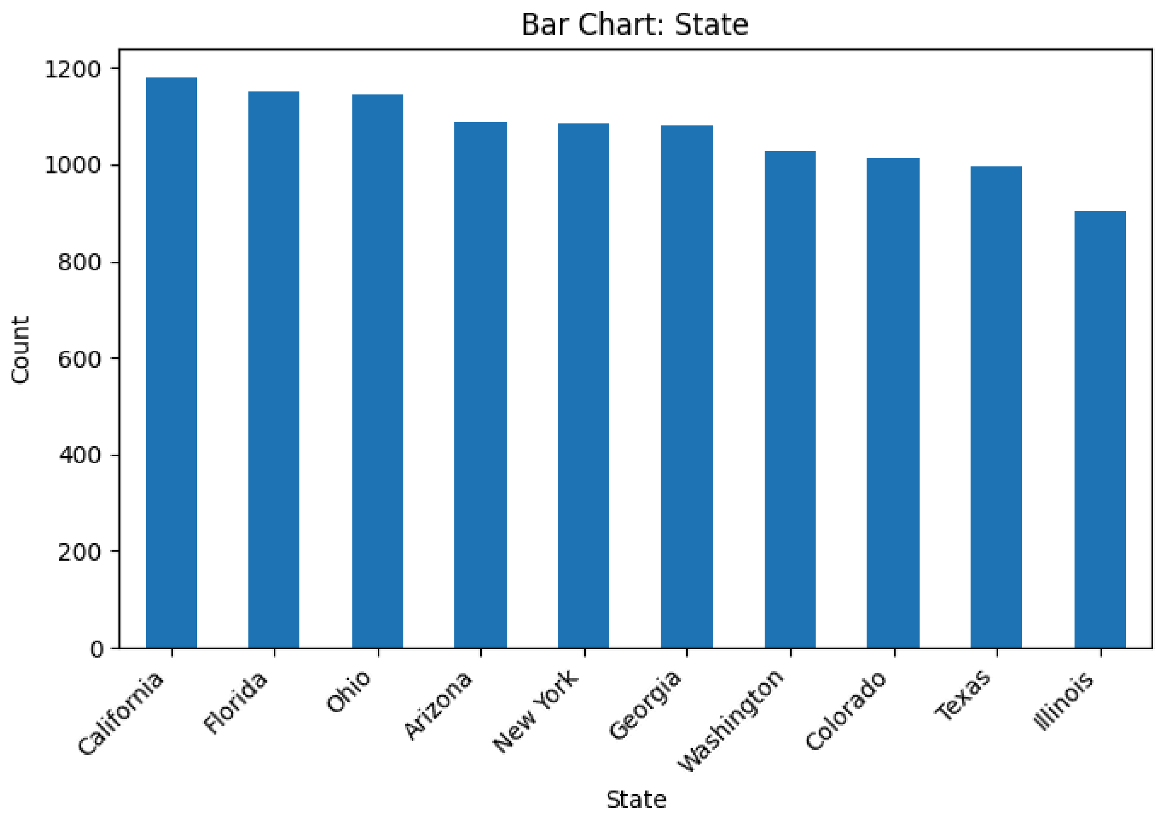
- **Create a bar chart for Gender, Education, State**

In [30]:
```
for col in ["Gender", "Education", "State"]:
    if col in df.columns:
        counts = df[col].value_counts(dropna=False)
        plt.figure(figsize=(7,5))
```

```
        counts.plot(kind="bar")
        plt.title(f"Bar Chart: {col}")
        plt.xlabel(col); plt.ylabel("Count")
        plt.xticks(rotation=45, ha="right")
        plt.tight_layout()
        plt.show()
    else:
        print(f"Column not found for bar chart: {col}")
```

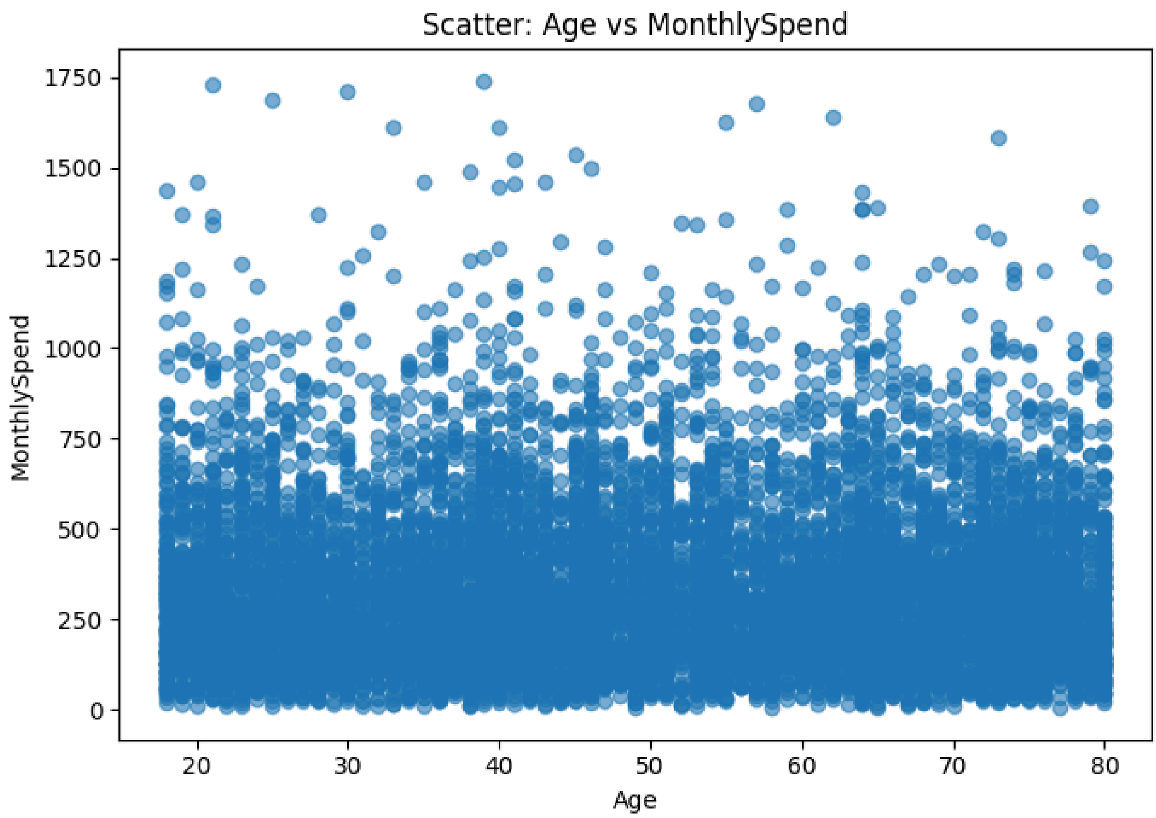






Scatterplot: Age vs MonthlySpend

In [36]:
```
if all(c in df.columns for c in ["Age", "MonthlySpend"]):
    plt.figure(figsize=(7,5))
    plt.scatter(df["Age"], df["MonthlySpend"], alpha=0.6)
    plt.title("Scatter: Age vs MonthlySpend")
    plt.xlabel("Age"); plt.ylabel("MonthlySpend")
    plt.tight_layout()
    plt.show()
else:
    print("Scatter skipped: 'Age' or 'MonthlySpend' missing.")
```

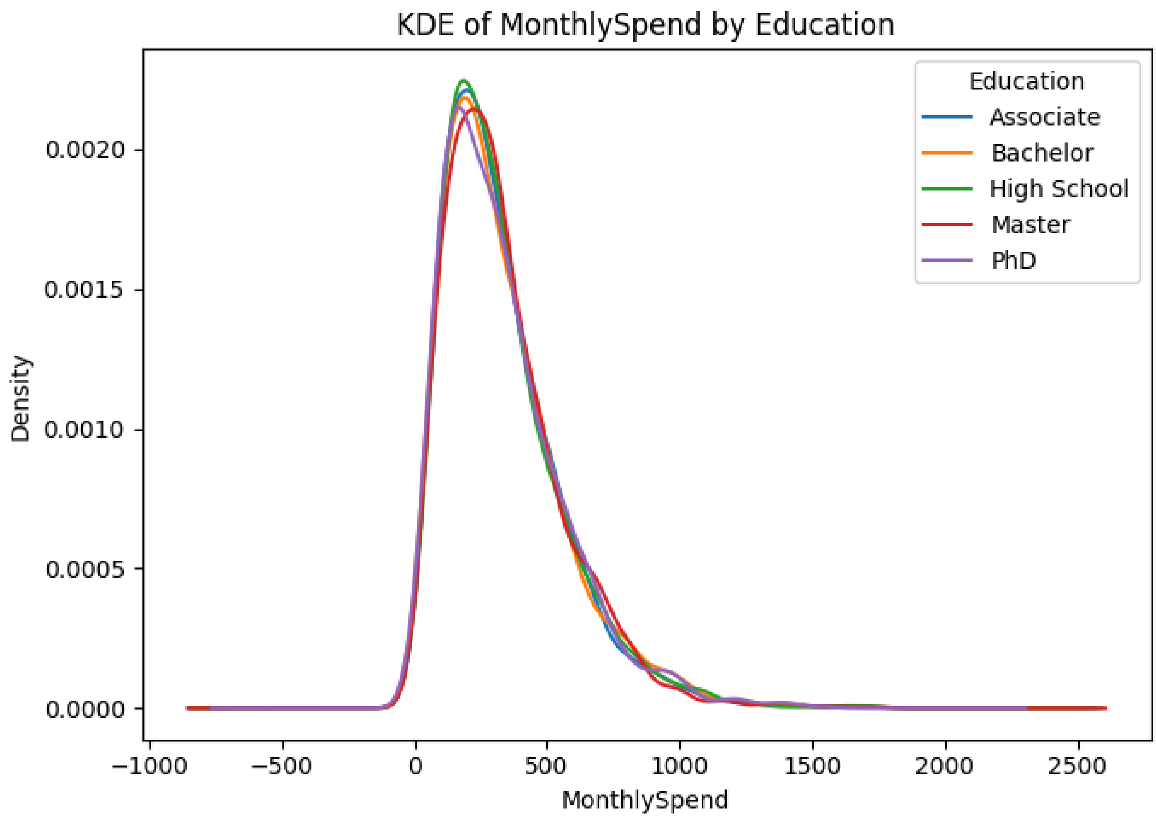


KDE: Spending behavior by Education OR Marital Status

In [10]:

```
# KDE: Spending behavior by Education OR Marital Status
# We'll prefer Education; if not present, try Married.
cat_col = None
if "Education" in df.columns:
    cat_col = "Education"
elif "Married" in df.columns:
    cat_col = "Married"

if cat_col and "MonthlySpend" in df.columns:
    # Use pandas' kde (matplotlib backend). It needs SciPy (Colab has it).
    plt.figure(figsize=(7,5))
    groups = df[[cat_col, "MonthlySpend"]].dropna()
    # Only plot categories with at least a few points
    for level, sub in groups.groupby(cat_col):
        if len(sub) >= 5 and sub["MonthlySpend"].nunique() > 1:
            sub["MonthlySpend"].plot(kind="kde", label=str(level))
    plt.title(f"KDE of MonthlySpend by {cat_col}")
    plt.xlabel("MonthlySpend"); plt.ylabel("Density")
    plt.legend(title=cat_col)
    plt.tight_layout()
    plt.show()
else:
    print("KDE skipped: Need 'MonthlySpend' and either 'Education' or 'Married'.
```



4. Bivariate Analysis

Business Purpose: Check how customer attributes relate to one another

Correlation matrix (numeric variables)

In [11]:
```
# Select only numeric columns automatically
num_df = df.select_dtypes(include=[np.number])

# Correlation matrix (Pearson)
corr = num_df.corr(numeric_only=True)

print("=== Correlation Matrix (numeric variables) ===")
corr.round(3)
```

```
=== Correlation Matrix (numeric variables) ===
```

Out[11]:

	Age	NumPets	MonthlySpend	DaysSinceLastInteraction
Age	1.000	-0.023	-0.012	-0.004
NumPets	-0.023	1.000	0.021	-0.055
MonthlySpend	-0.012	0.021	1.000	0.006
DaysSinceLastInteraction	-0.004	-0.055	0.006	1.000

Crosstab of Gender vs Married

In [12]:
```
# Count Crosstab
ct_counts = pd.crosstab(df.get("Gender"), df.get("Married"))
```

```python
print("=== Crosstab: Gender x Married (Counts) ===")
display(ct_counts)

ct_row_pct = pd.crosstab(df.get("Gender"), df.get("Married"), normalize="index")
print("\n=== Crosstab: Gender x Married (Row %) ===")
ct_row_pct.round(2)
```

```
=== Crosstab: Gender x Married (Counts) ===
```

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

```
=== Crosstab: Gender x Married (Row %) ===
```

Out[12]:

Married	No	Yes
Gender		
Female	52.65	47.35
Male	49.91	50.09
Non-Binary	54.57	45.43

Grouped stats: average MonthlySpend by State, Education, Gender

In [13]:

```python
if "MonthlySpend" in df.columns:
    grouped = (
        df.dropna(subset=["MonthlySpend"])
          .groupby(["State","Education","Gender"], dropna=False)["MonthlySpend"]
          .mean()
          .reset_index()
          .rename(columns={"MonthlySpend":"AvgMonthlySpend"})
    )
    print("=== Avg MonthlySpend by State, Education, Gender ===")
    print(grouped.round(2))
else:
    print("Column 'MonthlySpend' not found.")
```

```
=== Avg MonthlySpend by State, Education, Gender ===
          State  Education      Gender  AvgMonthlySpend
0       Arizona  Associate      Female           329.19
1       Arizona  Associate        Male           360.35
2       Arizona  Associate  Non-Binary           316.10
3       Arizona   Bachelor      Female           330.91
4       Arizona   Bachelor        Male           344.25
..          ...        ...         ...              ...
145  Washington     Master        Male           305.58
146  Washington     Master  Non-Binary           318.77
147  Washington        PhD      Female           368.06
148  Washington        PhD        Male           333.00
149  Washington        PhD  Non-Binary           351.27

[150 rows x 4 columns]
```

Correlation Matrix (numeric variables):

The correlations among Age, NumPets, MonthlySpend, and DaysSinceLastInteraction are all very weak (close to 0), meaning no strong linear relationships exist between these numeric attributes.

Crosstab (Gender × Married):

About 52.7% of Females are not married while 47.3% are married.

For Males, it is almost evenly split (49.9% No, 50.1% Yes).

Among Non-Binary customers, 54.6% are not married and 45.4% are married.

Grouped Stats (Average MonthlySpend by State, Education, Gender):

Average MonthlySpend varies by customer attributes.

Example: In Arizona, Associate-level Males spend the most (360.35) compared to Females (329.19) and Non-Binary (316.10).

In Washington, PhD-level Females spend more (368.06) compared to Males (333.00).

Overall Insight: Customer spending behavior is influenced more by State, Education, and Gender grouping than by numeric variables like Age or Pets, since correlations are weak. Marital status distribution differs slightly across genders, while spending shows clearer differences across demographic groups.

5: Formulate Hypotheses

1. Do males and females spend differently? → Independent t-test

In [14]:
```python
from scipy import stats
male_spend = df.loc[df["Gender"]=="Male", "MonthlySpend"].dropna()
female_spend = df.loc[df["Gender"]=="Female", "MonthlySpend"].dropna()

t_stat, p_val = stats.ttest_ind(male_spend, female_spend, equal_var=False)
print("=== Independent t-test: Male vs Female MonthlySpend ===")
print(f"t = {t_stat:.3f}, p = {p_val:.3f}")
if p_val < 0.05:
    print("Result: Significant difference between Male and Female spending.")
else:
    print("Result: No significant difference between Male and Female spending.")
```

```
=== Independent t-test: Male vs Female MonthlySpend ===
t = 0.339, p = 0.735
Result: No significant difference between Male and Female spending.
```

2. Does education level impact average monthly spend? → One-way ANOVA

In [15]:
```python
groups = [g["MonthlySpend"].dropna() for _, g in df.groupby("Education")]
f_stat, p_val = stats.f_oneway(*groups)
print("\n=== One-way ANOVA: Education vs MonthlySpend ===")
```

```python
print(f"F = {f_stat:.3f}, p = {p_val:.3f}")
if p_val < 0.05:
    print("Result: Education level significantly impacts MonthlySpend.")
else:
    print("Result: No significant impact of Education on MonthlySpend.")
```

```
=== One-way ANOVA: Education vs MonthlySpend ===
F = 0.229, p = 0.922
Result: No significant impact of Education on MonthlySpend.
```

3. Is marital status related to the number of pets? → Chi-square test

In [16]:
```python
if "Married" in df.columns and "NumPets" in df.columns:
    ctab = pd.crosstab(df["Married"], df["NumPets"])
    chi2, p_val, dof, exp = stats.chi2_contingency(ctab)
    print("\n=== Chi-square Test: Marital Status vs NumPets ===")
    print(f"Chi2 = {chi2:.3f}, p = {p_val:.3f}")
    if p_val < 0.05:
        print("Result: Marital status is related to number of pets.")
    else:
        print("Result: Marital status is NOT related to number of pets.")
```

```
=== Chi-square Test: Marital Status vs NumPets ===
Chi2 = 177.640, p = 0.000
Result: Marital status is related to number of pets.
```

4. Are older people less active? → Correlation (Age vs DaysSinceLastInteraction)

In [17]:
```python
if "Age" in df.columns and "DaysSinceLastInteraction" in df.columns:
    corr, p_val = stats.pearsonr(df["Age"].dropna(), df["DaysSinceLastInteractic
    print("\n=== Correlation: Age vs DaysSinceLastInteraction ===")
    print(f"r = {corr:.3f}, p = {p_val:.3f}")
    if p_val < 0.05:
        print("Result: Age and DaysSinceLastInteraction are significantly correl
    else:
        print("Result: No significant correlation.")
```

```
=== Correlation: Age vs DaysSinceLastInteraction ===
r = -0.004, p = 0.682
Result: No significant correlation.
```

5. Does state-wise spend vary significantly? → ANOVA

In [18]:
```python
if "State" in df.columns:
    groups_state = [g["MonthlySpend"].dropna() for _, g in df.groupby("State")]
    f_stat, p_val = stats.f_oneway(*groups_state)
    print("\n=== One-way ANOVA: State vs MonthlySpend ===")
    print(f"F = {f_stat:.3f}, p = {p_val:.3f}")
    if p_val < 0.05:
        print("Result: MonthlySpend differs significantly by State.")
    else:
        print("Result: No significant difference in MonthlySpend across States."
```

```
=== One-way ANOVA: State vs MonthlySpend ===
F = 1.118, p = 0.346
Result: No significant difference in MonthlySpend across States.
```

1. **Male vs Female Spending (t-test):** There is **no significant difference** in average MonthlySpend between males and females ($p = 0.735$).

2. **Education vs MonthlySpend (ANOVA):** Education level has **no significant effect** on MonthlySpend (p = 0.922).

3. **Marital Status vs Number of Pets (Chi-square):** Marital status is **significantly related** to the number of pets owned (p < 0.001).

4. **Age vs DaysSinceLastInteraction (Correlation):** There is **no significant correlation** between Age and DaysSinceLastInteraction (r ≈ -0.004, p = 0.682).

5. **State vs MonthlySpend (ANOVA):** Average MonthlySpend does **not differ significantly** across different states (p = 0.346).

Overall: **Marital status and number of pets are the only attributes that show a statistically significant relationship.**

6. Hypothesis Testing with Assumptions

Male vs Female Spending (Independent t-test)

Null Hypothesis (H0): Male and Female customers have the same average MonthlySpend.
Alternate Hypothesis (H1): Male and Female customers spend differently.

In [19]:

```
from statsmodels.stats.weightstats import ttest_ind
from statsmodels.formula.api import ols
import statsmodels.api as sm

male = df.loc[df["Gender"]=="Male","MonthlySpend"].dropna()
female = df.loc[df["Gender"]=="Female","MonthlySpend"].dropna()

print("=== Assumption Checks ===")
# Normality (Shapiro-Wilk)
print("Shapiro Male:", stats.shapiro(male.sample(500) if len(male)>500 else male
print("Shapiro Female:", stats.shapiro(female.sample(500) if len(female)>500 els

# Homogeneity of variance (Levene’s test)
print("Levene Test:", stats.levene(male, female))

print("\n=== Independent t-test ===")
t_stat, p_val, dfree = ttest_ind(male, female, usevar="unequal")
print(f"t = {t_stat:.3f}, p = {p_val:.3f}, df = {dfree:.1f}")

# 95% confidence interval of difference
diff_mean = male.mean() - female.mean()
se = np.sqrt(male.var(ddof=1)/len(male) + female.var(ddof=1)/len(female))
ci_low, ci_high = diff_mean - 1.96*se, diff_mean + 1.96*se
print(f"Mean Difference = {diff_mean:.2f}, 95% CI = [{ci_low:.2f}, {ci_high:.2f}
```

```
=== Assumption Checks ===
Shapiro Male: ShapiroResult(statistic=np.float64(0.9199031586552185), pvalue=np.f
loat64(1.209956097909071e-15))
Shapiro Female: ShapiroResult(statistic=np.float64(0.9203873500095084), pvalue=n
p.float64(1.3602518873665636e-15))
Levene Test: LeveneResult(statistic=np.float64(0.1267563861615179), pvalue=np.flo
at64(0.7218295518516542))

=== Independent t-test ===
t = 0.339, p = 0.735, df = 7119.7
Mean Difference = 1.81, 95% CI = [-8.66, 12.29]
```

Education vs MonthlySpend (ANOVA)

H0: Average MonthlySpend is the same across education groups. H1: At least one education group differs.

In [38]:
```
model = ols("MonthlySpend ~ C(Education)", data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

print("\n=== One-way ANOVA: Education vs MonthlySpend ===")
print(anova_table)

# Post-hoc (if significant) → Tukey test
from statsmodels.stats.multicomp import pairwise_tukeyhsd

if anova_table["PR(>F)"].iloc[0] < 0.05:
    tukey = pairwise_tukeyhsd(df["MonthlySpend"], df["Education"])
    print(tukey)
```

```
=== One-way ANOVA: Education vs MonthlySpend ===
                    sum_sq       df         F    PR(>F)
C(Education)  4.667660e+04      4.0  0.228807  0.922359
Residual      5.441704e+08  10670.0       NaN       NaN
```

Marital Status vs Number of Pets (Chi-square)

H0: Marital status and number of pets are independent. H1: Marital status and number of pets are related.

In [21]:
```
ctab = pd.crosstab(df["Married"], df["NumPets"])
chi2, p, dof, exp = stats.chi2_contingency(ctab)

print("\n=== Chi-square Test: Married vs NumPets ===")
print("Chi2 =", chi2, " p =", p, " dof =", dof)
```

```
=== Chi-square Test: Married vs NumPets ===
Chi2 = 177.63953668537033  p = 2.3957232932397494e-37  dof = 4
```

Age vs DaysSinceLastInteraction (Correlation)

H0: Age and activity (days since last interaction) are uncorrelated. H1: Age and activity are correlated.

In [22]:
```
corr, p_val = stats.pearsonr(df["Age"].dropna(), df["DaysSinceLastInteraction"].
print("\n=== Correlation: Age vs DaysSinceLastInteraction ===")
print(f"r = {corr:.3f}, p = {p_val:.3f}")
```

```
=== Correlation: Age vs DaysSinceLastInteraction ===
r = -0.004, p = 0.682
```

State vs MonthlySpend (ANOVA)

H0: Average MonthlySpend is the same across states. H1: At least one state differs.

In [23]:
```python
model2 = ols("MonthlySpend ~ C(State)", data=df).fit()
anova_state = sm.stats.anova_lm(model2, typ=2)

print("\n=== One-way ANOVA: State vs MonthlySpend ===")
print(anova_state)
```

```
=== One-way ANOVA: State vs MonthlySpend ===
                sum_sq       df         F    PR(>F)
C(State)  5.128908e+05      9.0  1.117842  0.345719
Residual  5.437042e+08  10665.0       NaN       NaN
```

---

1. Independent t-test (Male vs Female MonthlySpend)

- **Assumption check:** Shapiro tests show that spending is **not normally distributed** ($p < 0.05$), but Levene's test indicates equal variances ($p = 0.72$).
- **Test result:** $t = 0.339$, $p = 0.735$ → **No significant difference** in spending between males and females.
- **Mean difference:** Males spend about **1.81 more** on average, but the 95% CI [-8.66, 12.29] crosses zero, confirming no reliable difference.

---

2. One-way ANOVA (Education vs MonthlySpend)

- $F = 0.229$, $p = 0.922$ → **Education level does not significantly affect spending.**

---

3. Chi-square Test (Marital Status vs NumPets)

- Chi2 = 177.64, $p < 0.001$ → **Marital status is significantly related to the number of pets owned.**

---

4. Correlation (Age vs DaysSinceLastInteraction)

- $r = -0.004$, $p = 0.682$ → **No significant correlation**; Age does not explain customer activity (recency of interaction).

---

5. One-way ANOVA (State vs MonthlySpend)

- $F = 1.118$, $p = 0.346$ → **No significant difference** in spending across states.

Overall Conclusion: Among all hypotheses, only **Marital Status vs Number of Pets** shows a statistically significant relationship. All other tests (Gender, Education, Age, State) show **no significant effects on spending or activity.**

7:Present Business Insights

1. Customer Profile Most customers are **middle-aged (~49 years)**, predominantly **Male**, with **Master's degrees**, and **Unmarried**. This represents the **dominant segment**.

2. Spending Behavior Average monthly spend is about **$330**, with a high standard deviation (~$226). This shows **diverse spending patterns**, suggesting both high-value and low-value customer groups exist.

3. Engagement Customers are generally **inactive**, with the last interaction occurring on average **538 days (~1.5 years)** ago. This indicates a **critical need for re-engagement initiatives**.

4. Demographics vs Spending Statistical tests show **Gender, Education, Age, and State do not significantly impact MonthlySpend** (all p-values > 0.3). Hence, **traditional demographics are weak predictors of spending**.

5. Lifestyle Signals Marital status is strongly associated with pet ownership (Chi-square p < 0.001). This provides the **most actionable segmentation factor**. For example, campaigns targeting **unmarried pet owners** can yield higher engagement.

Strategic Conclusion Your customer base is mostly **unmarried, middle-aged, Master's degree holders who spend moderately but remain inactive**. Since demographics have limited impact on spending, **lifestyle-based segmentation (Marital status + Pets)** should drive strategy. **Re-engagement campaigns tailored for these lifestyle clusters** hold the highest potential business impact.
