In [1]:
```python
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
# Load the dataset
df = pd.read_csv('bikes_sales_dataset.csv')
```

In [3]:
```python
df.head()
```

Out[3]:

	ID	Martial Status	Gender	Income	Children	Education	Occupation	Home Owner	Cars	Commute Distance	Region	Age	Age Brackets	Purchas B
0	12496	Married	Female	$40,000	1	Bachelors	Skilled Manual	Yes	0	0-1 Miles	Europe	42	Middle Age	
1	24107	Married	Male	$30,000	3	Partial College	Clerical	Yes	1	0-1 Miles	Europe	43	Middle Age	
2	14177	Married	Male	$80,000	5	Partial College	Professional	No	2	2-5 Miles	Europe	60	Old	
3	24381	Single	Male	$70,000	0	Bachelors	Professional	Yes	1	5-10 Miles	Pacific	41	Middle Age	
4	25597	Single	Male	$30,000	0	Bachelors	Clerical	No	0	0-1 Miles	Europe	36	Middle Age	

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   ID                1000 non-null   int64 
 1   Martial Status    1000 non-null   object
 2   Gender            1000 non-null   object
 3   Income            1000 non-null   object
 4   Children          1000 non-null   int64 
 5   Education         1000 non-null   object
 6   Occupation        1000 non-null   object
 7   Home Owner        1000 non-null   object
 8   Cars              1000 non-null   int64 
 9   Commute Distance  1000 non-null   object
 10  Region            1000 non-null   object
 11  Age               1000 non-null   int64 
 12  Age Brackets      1000 non-null   object
 13  Purchased Bike    1000 non-null   object
dtypes: int64(4), object(10)
memory usage: 109.5+ KB
```

In [5]:
```
# Convert Income to numeric
df['Income'] = df['Income'].replace('[\$,]', '', regex=True).astype(float)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   ID                1000 non-null   int64  
 1   Martial Status    1000 non-null   object 
 2   Gender            1000 non-null   object 
 3   Income            1000 non-null   float64
 4   Children          1000 non-null   int64  
 5   Education         1000 non-null   object 
 6   Occupation        1000 non-null   object 
 7   Home Owner        1000 non-null   object 
 8   Cars              1000 non-null   int64  
 9   Commute Distance  1000 non-null   object 
 10  Region            1000 non-null   object 
 11  Age               1000 non-null   int64  
 12  Age Brackets      1000 non-null   object 
 13  Purchased Bike    1000 non-null   object 
dtypes: float64(1), int64(4), object(9)
memory usage: 109.5+ KB
```

In [7]:
```
df.describe()
```

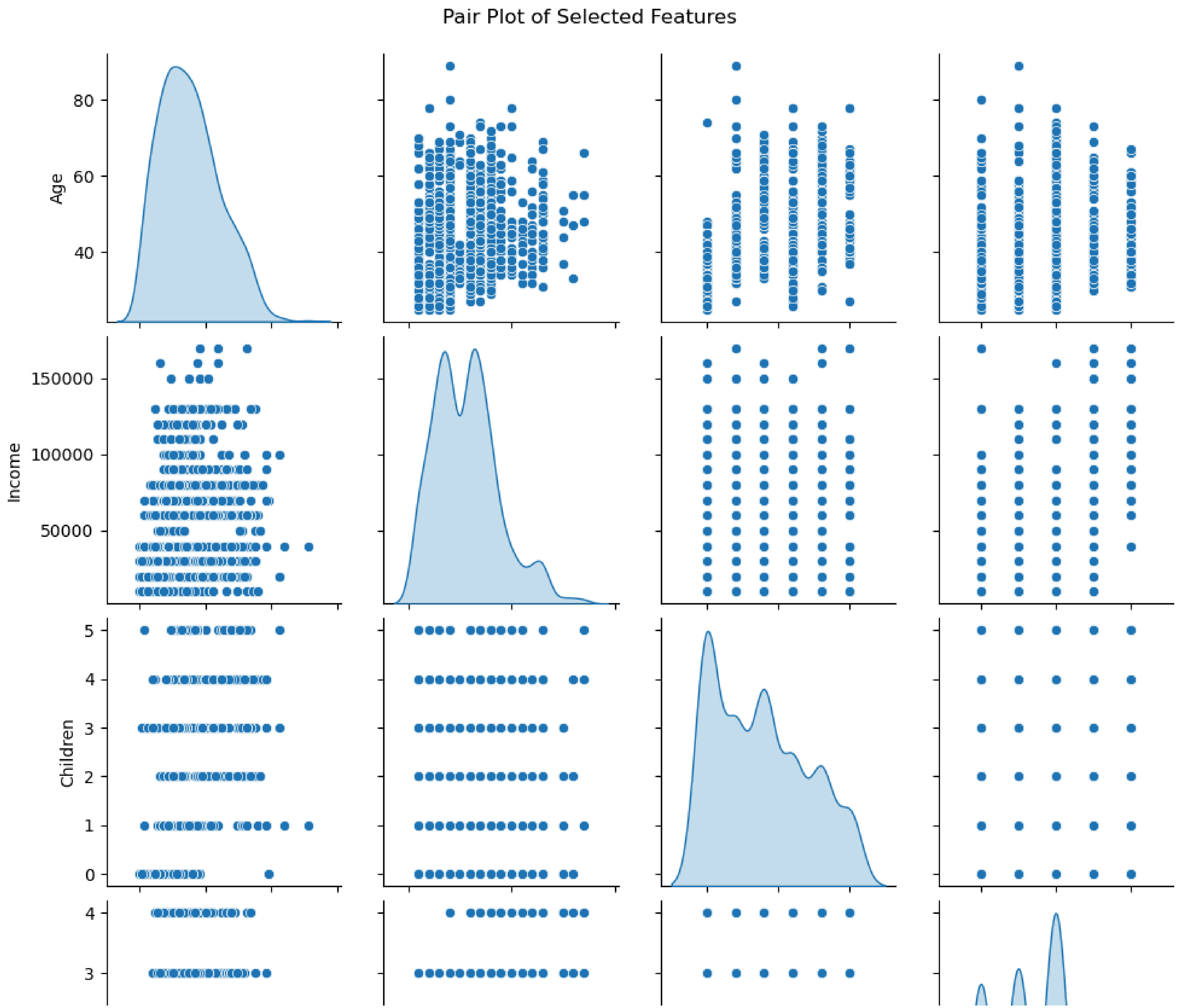
Out[7]:

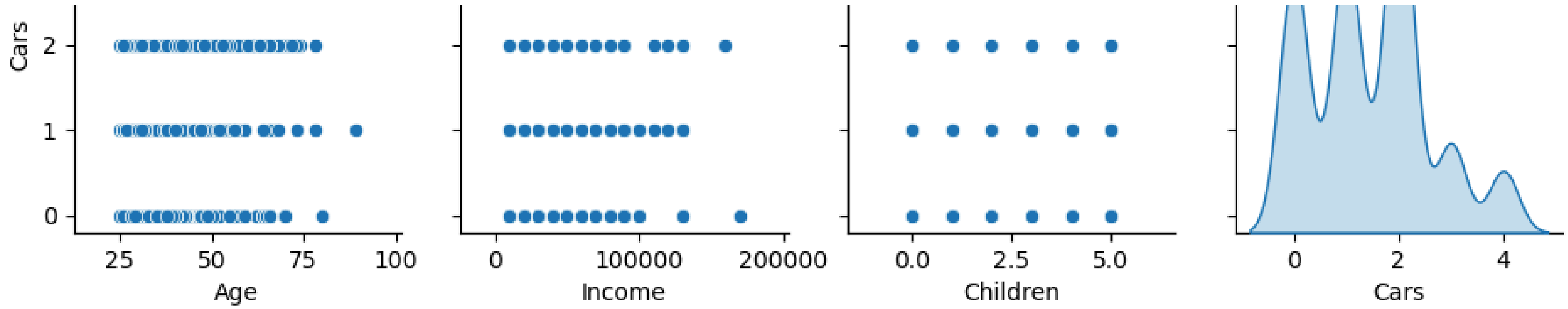
	ID	Income	Children	Cars	Age
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	19965.992000	56360.000000	1.898000	1.442000	44.163000
std	5347.333948	31085.635215	1.628572	1.125123	11.364488
min	11000.000000	10000.000000	0.000000	0.000000	25.000000
25%	15290.750000	30000.000000	0.000000	1.000000	35.000000
50%	19744.000000	60000.000000	2.000000	1.000000	43.000000
75%	24470.750000	70000.000000	3.000000	2.000000	52.000000
max	29447.000000	170000.000000	5.000000	4.000000	89.000000

In [8]:
```
sns.pairplot(df[['Age', 'Income', 'Children', 'Cars', 'Purchased Bike']], diag_kind='kde')

# Display the plot
plt.suptitle('Pair Plot of Selected Features', y=1.02)  # Add a title to the pair plot
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

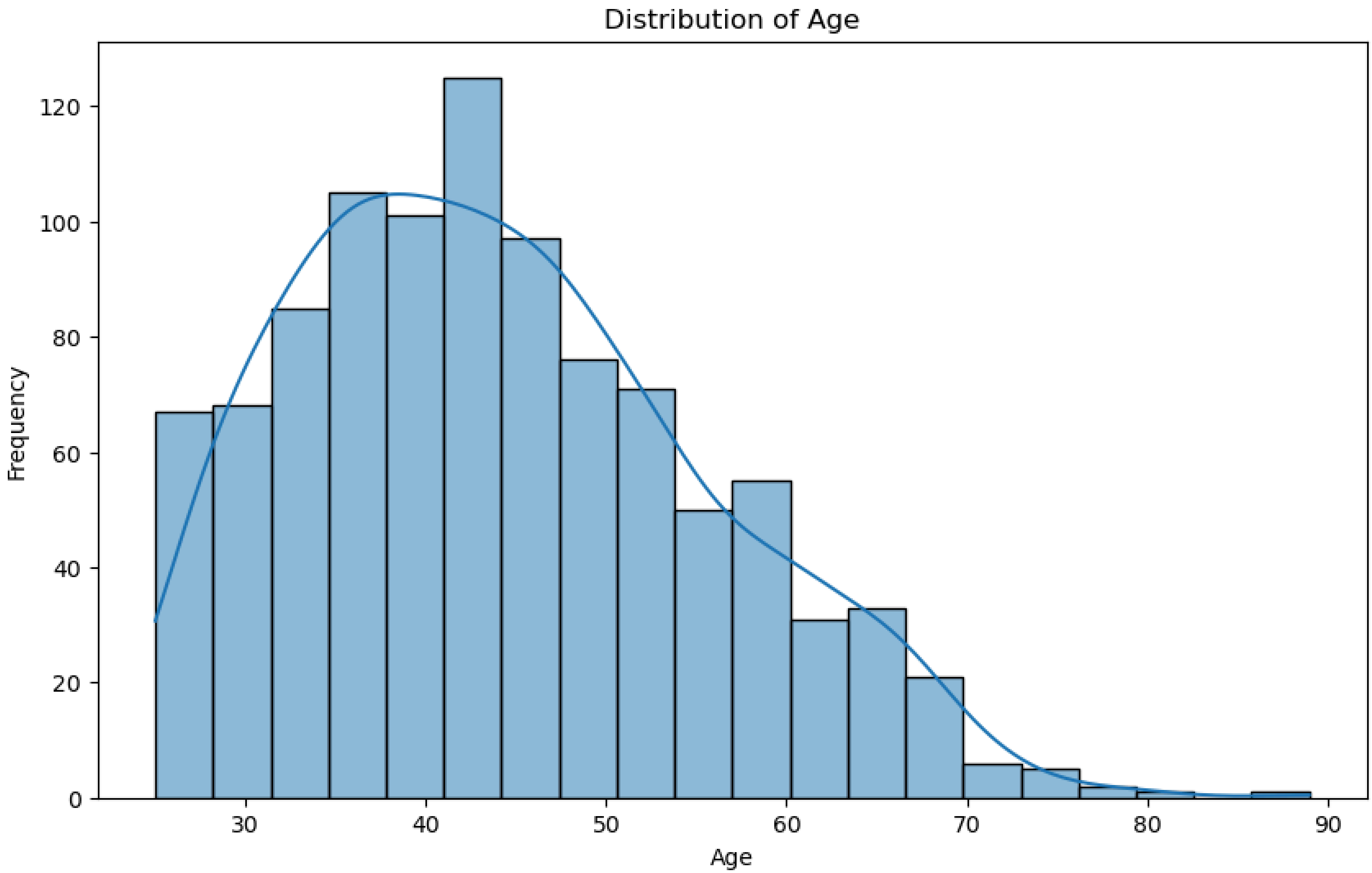

Pair Plot of Selected Features
Age
80
60
40
Income
150000
100000
50000
Children
5
4
3
2
1
0
4
3



Distribution of Age

In [9]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['Age'], bins=20, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

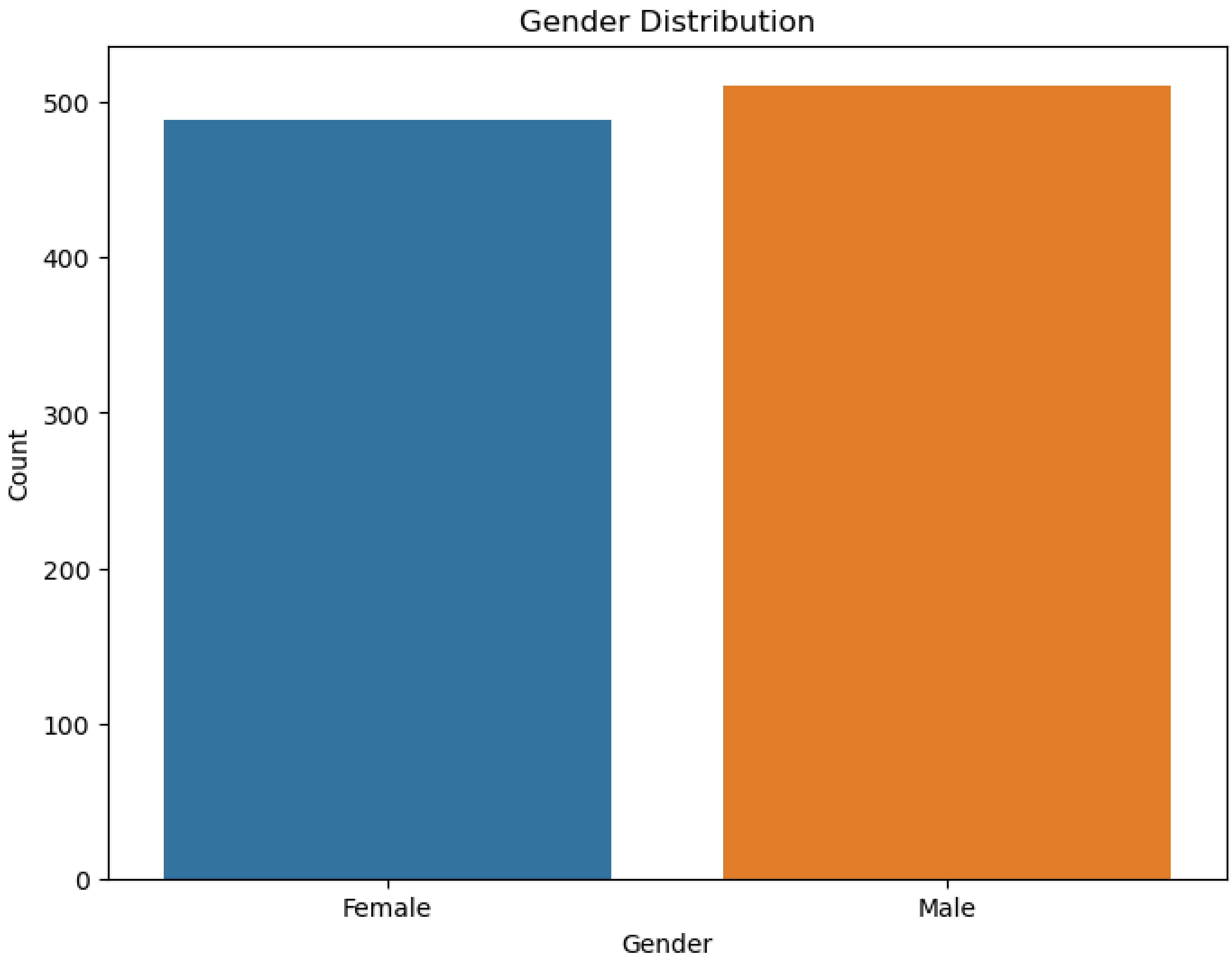


Observation:

- The age distribution appears to be roughly normal, with a majority of customers falling within a certain age range.
- The dataset includes a diverse range of ages, which may help in understanding the buying patterns across different age groups.

Gender Distribution

In [10]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Gender')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



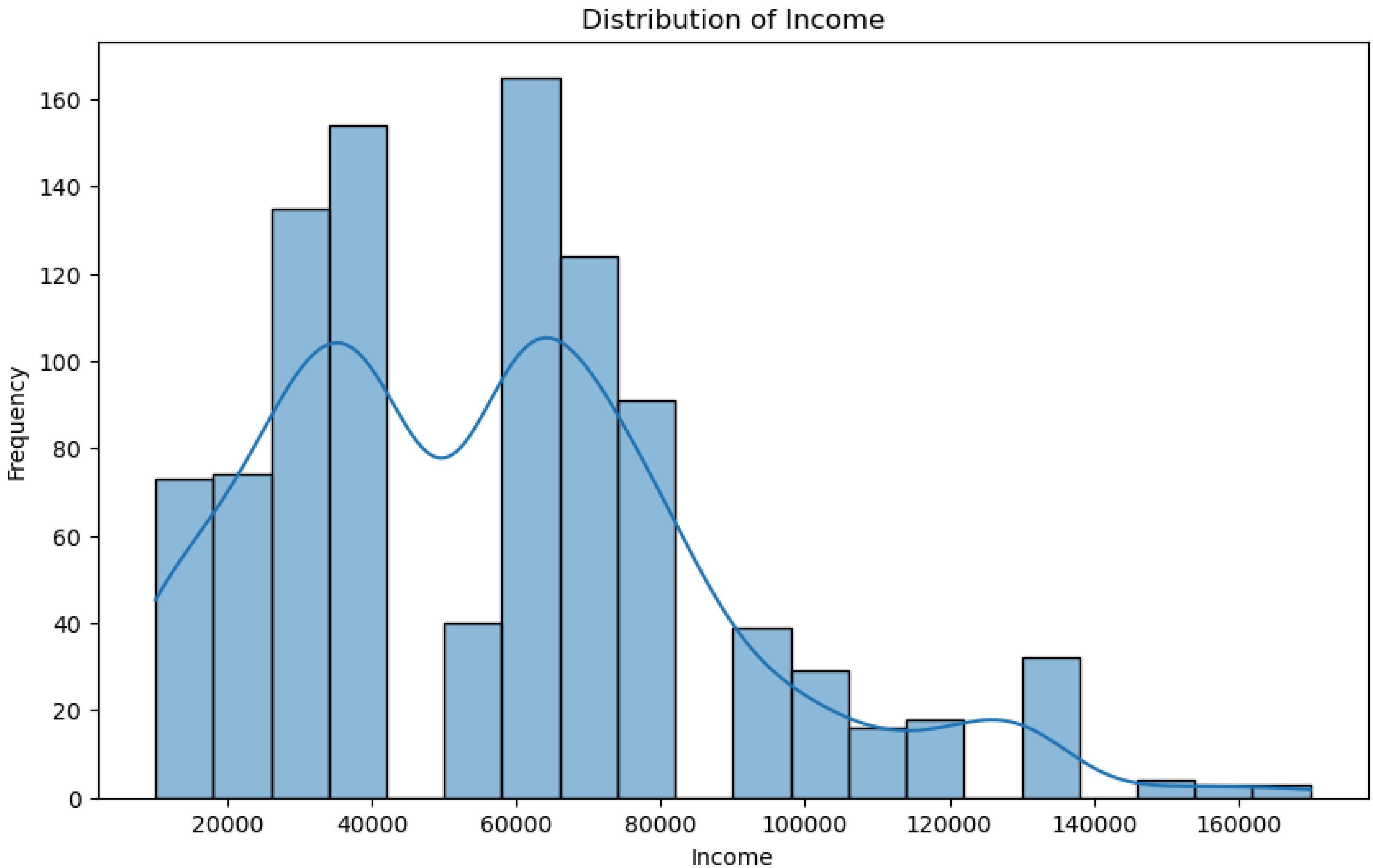
Observation:

- The dataset includes a roughly balanced number of male and female customers.
- This balance can help in analyzing if there are significant differences in purchasing behaviors between genders.

Income Distribution

In [11]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['Income'], bins=20, kde=True)
plt.title('Distribution of Income')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecate
d and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

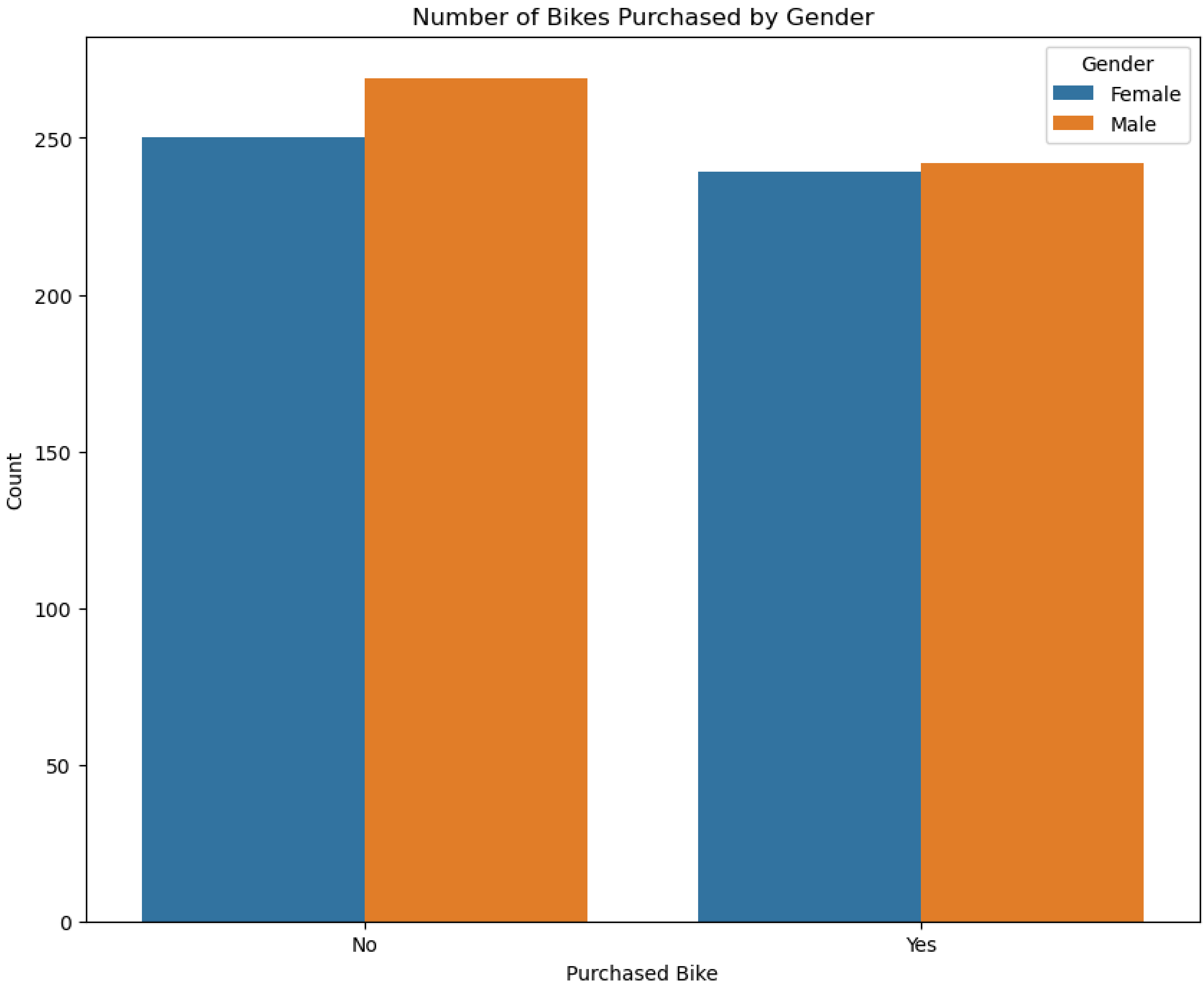


Observation:

- The income distribution shows a wide range, indicating that the dataset includes customers from various economic backgrounds.
- The presence of high-income customers might influence the purchasing behavior towards bikes.

Number of Bikes Purchased by Gender

In [12]:
```
plt.figure(figsize=(10, 8))
sns.countplot(data=df, x='Purchased Bike', hue='Gender')
plt.title('Number of Bikes Purchased by Gender')
plt.xlabel('Purchased Bike')
plt.ylabel('Count')
plt.show()
```

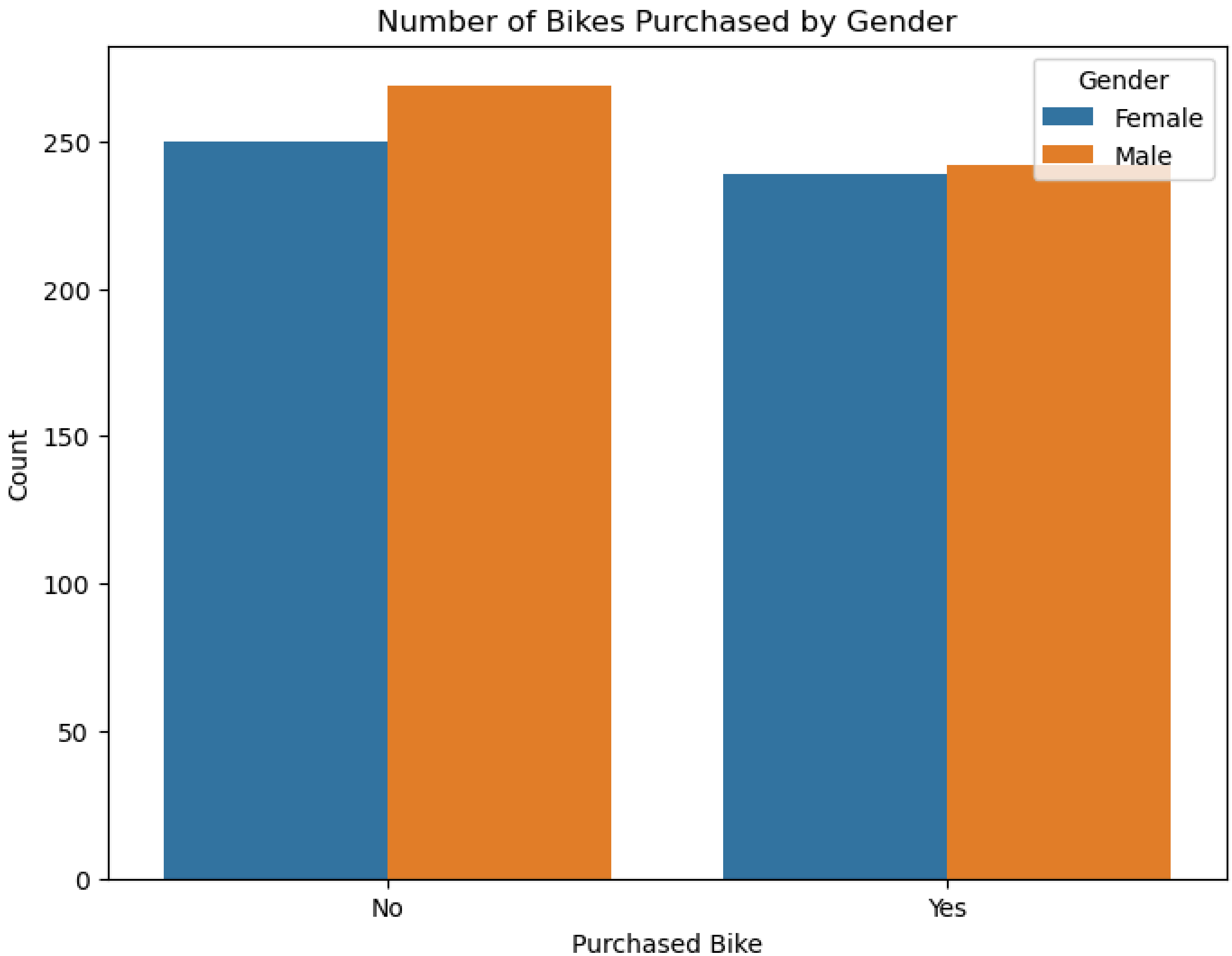

Number of Bikes Purchased by Gender
Gender
Female
Male
Count
0
50
100
150
200
250
No
Yes
Purchased Bike

Observation:

- The visualization indicates whether there are significant differences in bike purchases between male and female customers.
- This can be useful in targeting marketing efforts towards the more likely gender group to purchase bikes.

Average Income by Occupation

In [13]:
```
# Number of Bikes Purchased by Gender
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Purchased Bike', hue='Gender')
plt.title('Number of Bikes Purchased by Gender')
plt.xlabel('Purchased Bike')
plt.ylabel('Count')
plt.show()
```

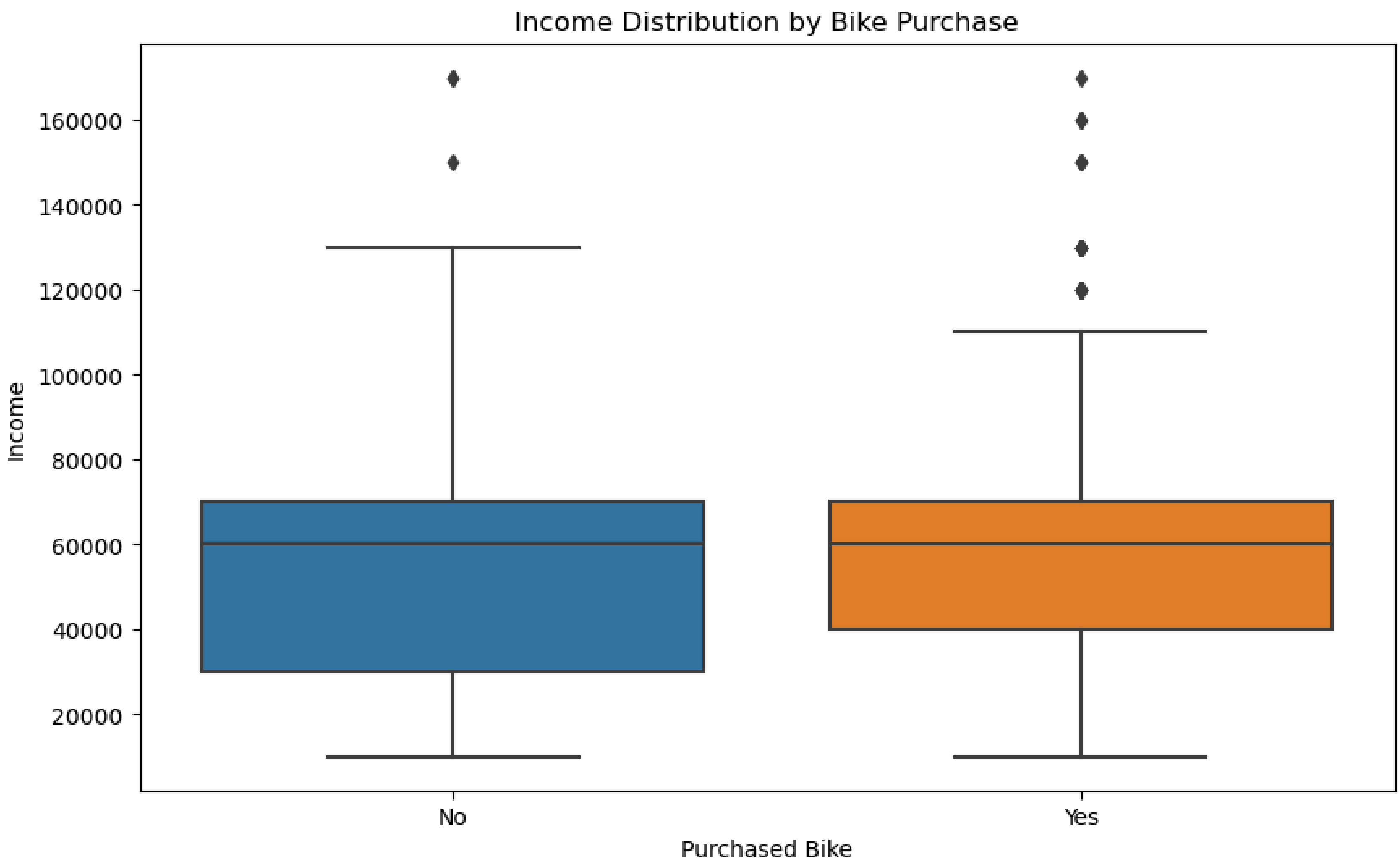


Observation:

- Different occupations have varying average incomes.
- Higher-income occupations might be more likely to purchase bikes, especially higher-end models.

Income Distribution by Bike Purchase

In [14]:
```python
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Purchased Bike', y='Income')
plt.title('Income Distribution by Bike Purchase')
plt.xlabel('Purchased Bike')
plt.ylabel('Income')
plt.show()
```

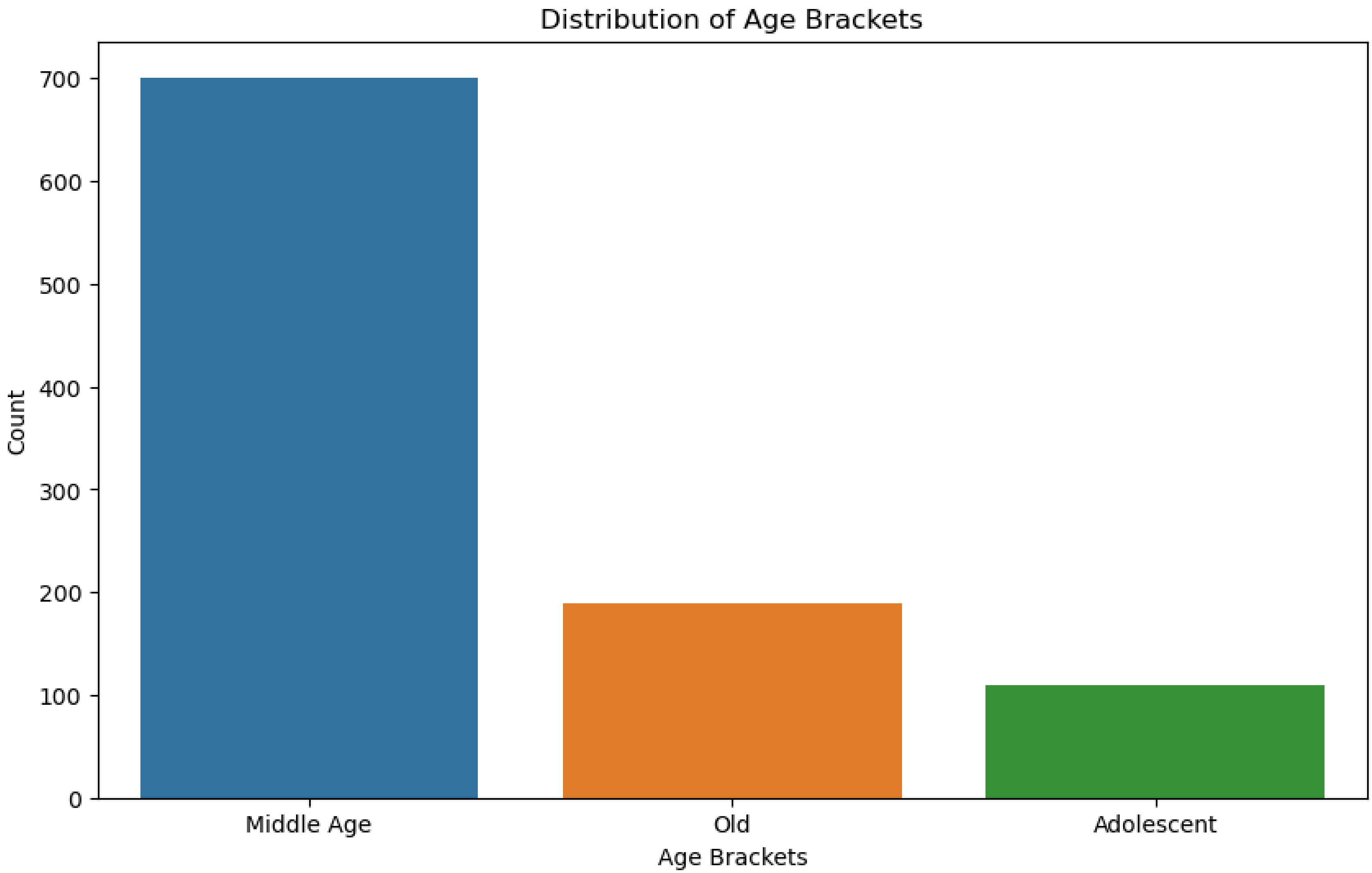


Observation:

- Customers who purchased bikes might show different income patterns compared to those who did not.
- This can help in understanding if income is a significant factor in the decision to purchase a bike.

Age Brackets Distribution

In [15]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Age Brackets')
plt.title('Distribution of Age Brackets')
plt.xlabel('Age Brackets')
plt.ylabel('Count')
plt.show()
```

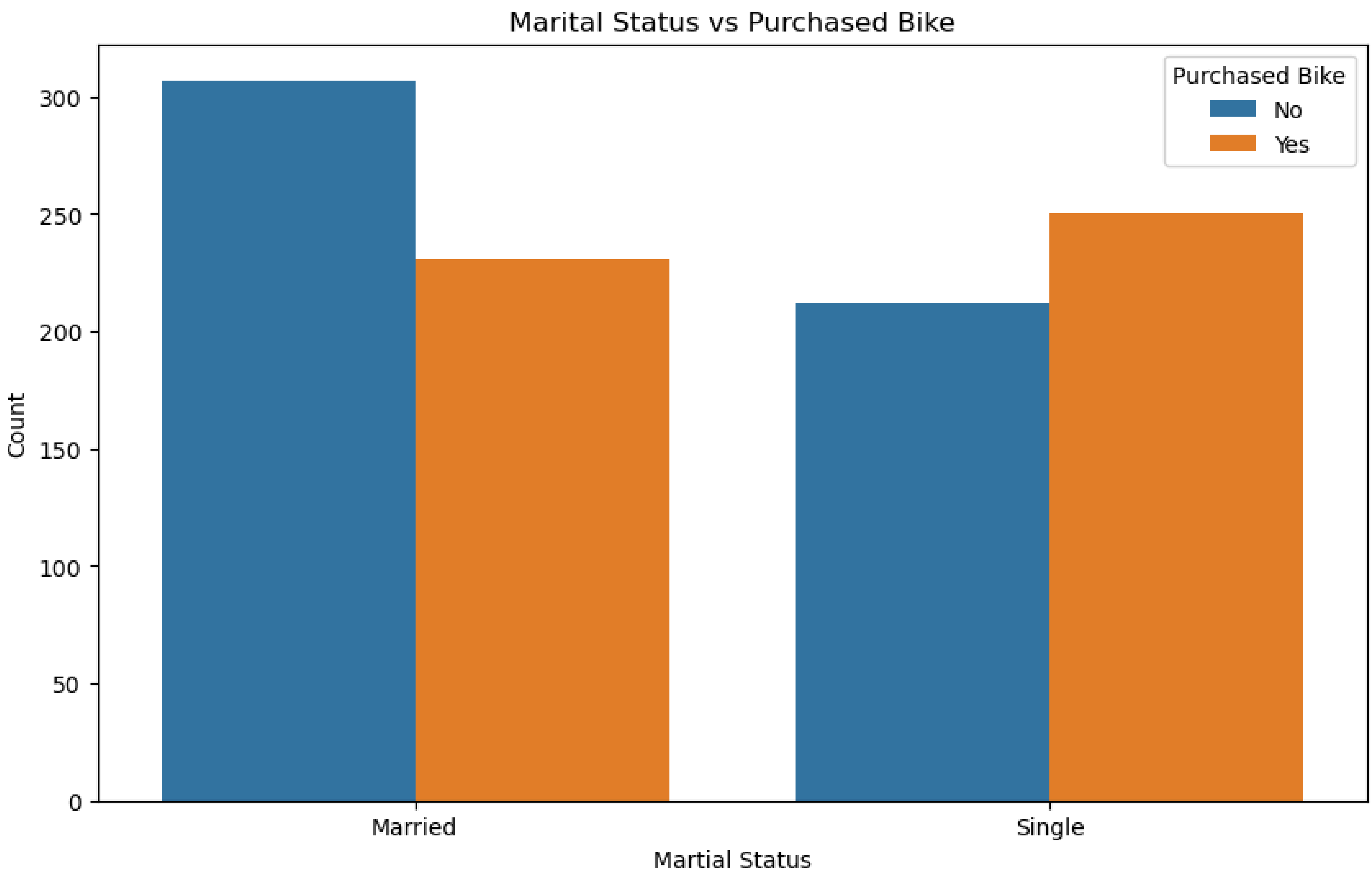


Observation:

- The dataset includes various age brackets, showing the age diversity of the customer base.
- Analyzing purchasing behaviors across different age brackets can help in segmenting the market effectively.

Marital Status vs Purchased Bike

In [16]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Martial Status', hue='Purchased Bike')
plt.title('Marital Status vs Purchased Bike')
plt.xlabel('Martial Status')
plt.ylabel('Count')
plt.show()
```

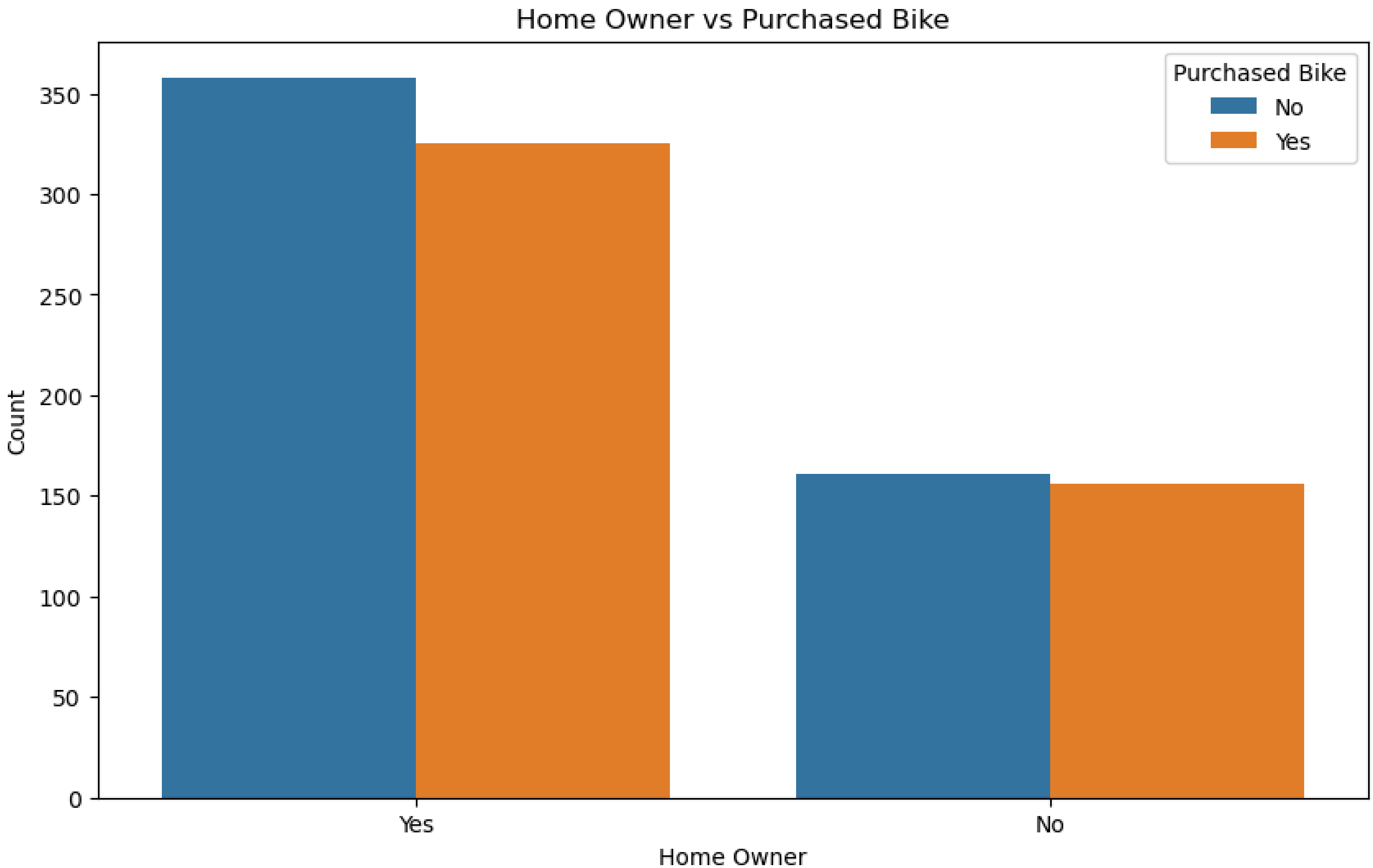


Observation:

- There might be observable differences in bike purchasing behavior based on marital status.
- Understanding this can help in tailoring marketing campaigns to target specific marital status groups more effectively.

Home Owner vs Purchased Bike

In [17]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Home Owner', hue='Purchased Bike')
plt.title('Home Owner vs Purchased Bike')
plt.xlabel('Home Owner')
plt.ylabel('Count')
plt.show()
```

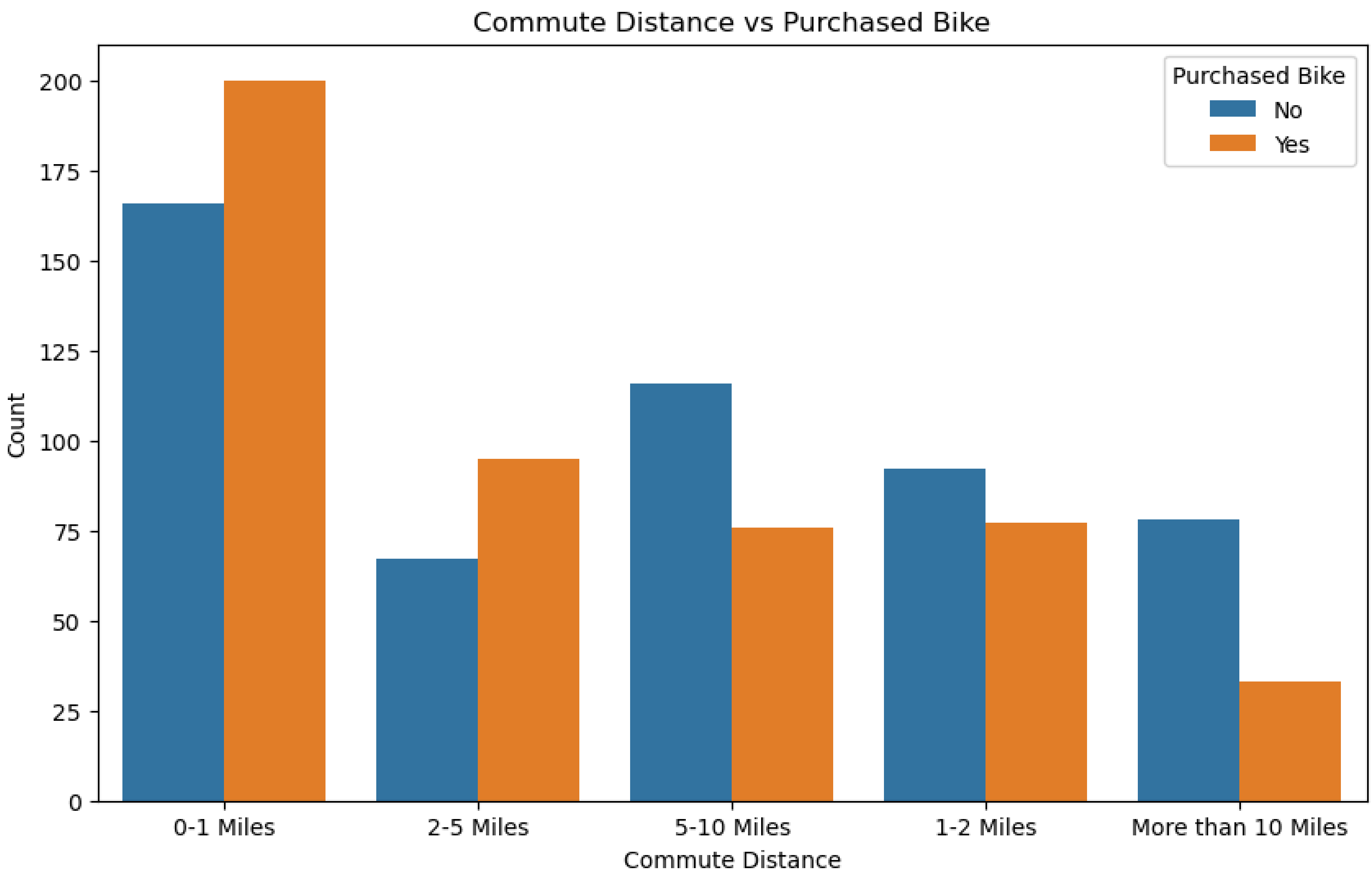


Observation:

- Homeownership might be an indicator of financial stability, which could influence bike purchasing decisions.
- Customers who own homes might have a higher likelihood of purchasing bikes.

Commute Distance vs Purchased Bike

In [18]:
```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Commute Distance', hue='Purchased Bike')
plt.title('Commute Distance vs Purchased Bike')
plt.xlabel('Commute Distance')
plt.ylabel('Count')
plt.show()
```



Observation:

- The distance customers commute could influence their decision to purchase a bike.
- Customers with shorter commute distances might be more inclined to buy bikes for convenience.

Contact Information

For any queries or further information, please feel free to reach out to me through the following platforms:

- **LinkedIn**: Vinay Kumar Panika
- **GitHub**: Vinaypanika