Business Case: Walmart - Confidence Interval and CLT

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as st
```

Load dataset

In [5]:
```
df=pd.read_csv('walmart_data.csv')
```

Verify correct data import by checking the first and last 5 rows

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

Out[6]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_St
0	1000001	P00069042	F	0-17	10	A	2	
1	1000001	P00248942	F	0-17	10	A	2	
2	1000001	P00087842	F	0-17	10	A	2	
3	1000001	P00085442	F	0-17	10	A	2	
4	1000002	P00285442	M	55+	16	C	4+	

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

Out[7]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Mar
550063	1006033	P00372445	M	51-55	13	B	1	
550064	1006035	P00375436	F	26-35	1	C	3	
550065	1006036	P00375436	F	26-35	15	B	4+	
550066	1006038	P00375436	F	55+	1	C	2	
550067	1006039	P00371644	F	46-50	0	B	4+	

List of columns present in the datasert

In [8]:
```
df.columns
```

Out[8]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

Shape of the data in the dataset

In [9]:
```
df.shape
```

Out[9]:
```
(550068, 10)
```

Datatype of each column in the dataset

In [10]:
```
df.dtypes
```

Out[10]:
```
User_ID                        int64
Product_ID                    object
Gender                        object
Age                           object
Occupation                     int64
City_Category                 object
Stay_In_Current_City_Years    object
Marital_Status                 int64
Product_Category               int64
Purchase                       int64
dtype: object
```

Overview of the DataFrame, including row counts, column data types, and memory usage

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype 
---  ------                      --------------   ----- 
 0   User_ID                     550068 non-null  int64 
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64 
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64 
 8   Product_Category            550068 non-null  int64 
 9   Purchase                    550068 non-null  int64 
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

Insights

1. The dataset consists of 10 columns with a mix of alphanumeric values.
2. All the columns apart from Purchase column contains categorical data.
3. The data types of these columns needs to be converted to category type for optimization.

Optimizing the datatypes of columns

In [12]:
```
df[df.columns[:-1]]=df[df.columns[:-1]].apply(lambda x: x.astype('category'))
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype   
---  ------                      --------------   -----   
 0   User_ID                     550068 non-null  category
 1   Product_ID                  550068 non-null  category
 2   Gender                      550068 non-null  category
 3   Age                         550068 non-null  category
 4   Occupation                  550068 non-null  category
 5   City_Category               550068 non-null  category
 6   Stay_In_Current_City_Years  550068 non-null  category
 7   Marital_Status              550068 non-null  category
 8   Product_Category            550068 non-null  category
 9   Purchase                    550068 non-null  int64   
dtypes: category(9), int64(1)
memory usage: 10.3 MB
```

Getting summary statistics for each numerical columns in the dataset.

In [13]:
```
df.describe(include='category')
```

Out[13]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years
count	550068	550068	550068	550068	550068	550068	550068
unique	5891	3631	2	7	21	3	5
top	1001680	P00265242	M	26-35	4	B	1
freq	1026	1880	414259	219587	72308	231173	193821

Insights

1. There are 5891 unique users in the total number of users 550068 which implies that there are repeat customers.
2. There are 3631 products with P00265242 being the highest purchased product with frequency of 1880.
3. Males make up 414259 of the 550068 transactions, suggesting that they are regular buyers during the Black Friday Sale.

4. Customers in the age ranging from 26 to 35 made the highest purchases.
5. There are 21 occupation types and customers with 4 accounted for most of the sales.8
6. All the cities are categorized into 3 types and most of the customers are from category B.
7. Customers who stayed in the city are the frequent customers.
8. Unmarried customers made most of the purchases.
9. Out of the 20 product categories category 5 is the highest bought product.

In [14]:
```
df.describe()
```

Out[14]:

	Purchase
count	550068.000000
mean	9263.968713
std	5023.065394
min	12.000000
25%	5823.000000
50%	8047.000000
75%	12054.000000
max	23961.000000

Based on the above information, The medain purchase is 8047 dollars which is lesser than the mean 9264 dollars which implies that the data is Right Skewed.

Checking for the duplicated values

In [15]:
```
df.duplicated().value_counts()
```

Out[15]:
```
False    550068
dtype: int64
```

Changing the values of marital_status columns 0,1 to Married, Single

In [16]:
```
df['Marital_Status']=df['Marital_Status'].replace({0:'Single', 1:'Married'})
df['Marital_Status'].unique()
```

Out[16]:
```
['Single', 'Married']
Categories (2, object): ['Single', 'Married']
```

Check if there are any null values.

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category              0
Purchase                      0
dtype: int64
```

There are no null values in the dataset.

Category based Spend Analysis

In [18]:
```
female_customers = df[df['Gender'] == 'F']
male_customers = df[df['Gender'] == 'M']
```

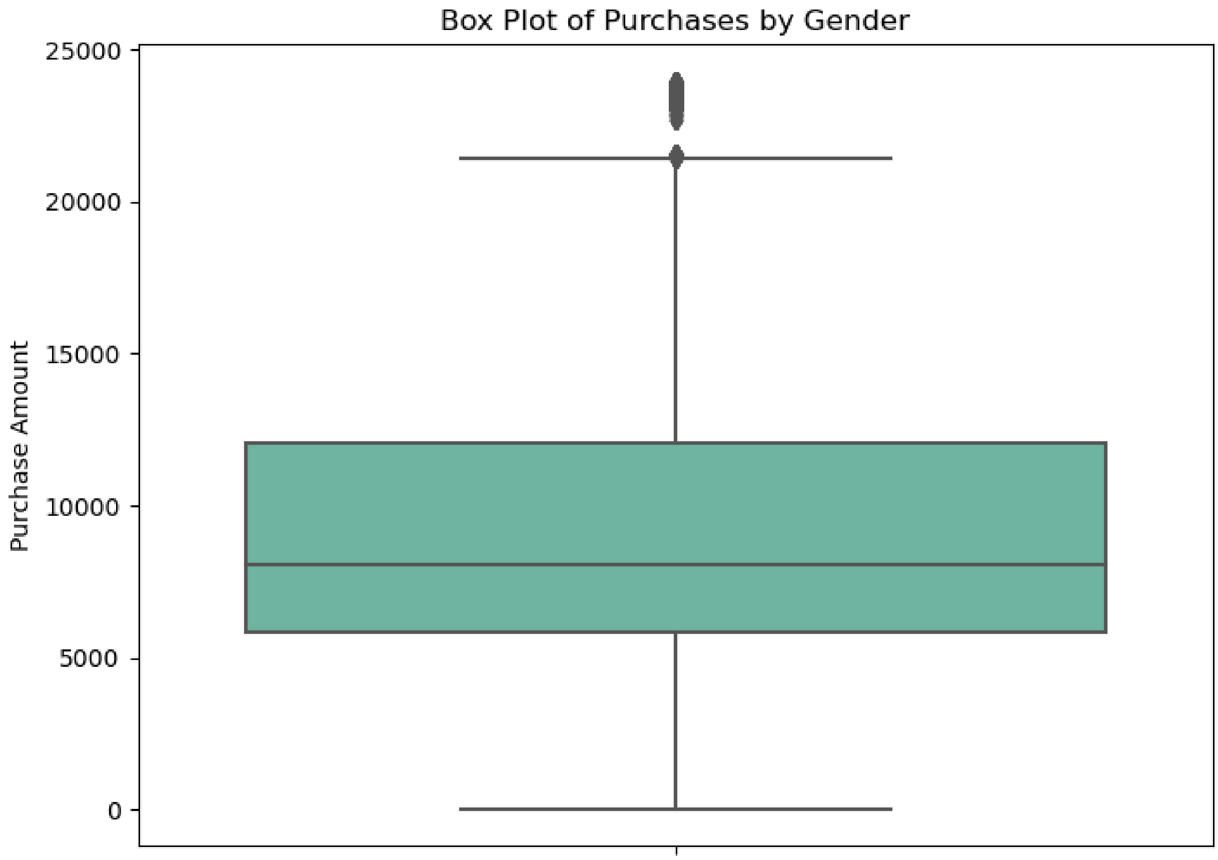
In [19]:
```
Avg_male_spending=male_customers['Purchase'].mean()
Avg_female_spending=female_customers['Purchase'].mean()
Avg_male_spending
```

Out[19]:
```
9437.526040472265
```

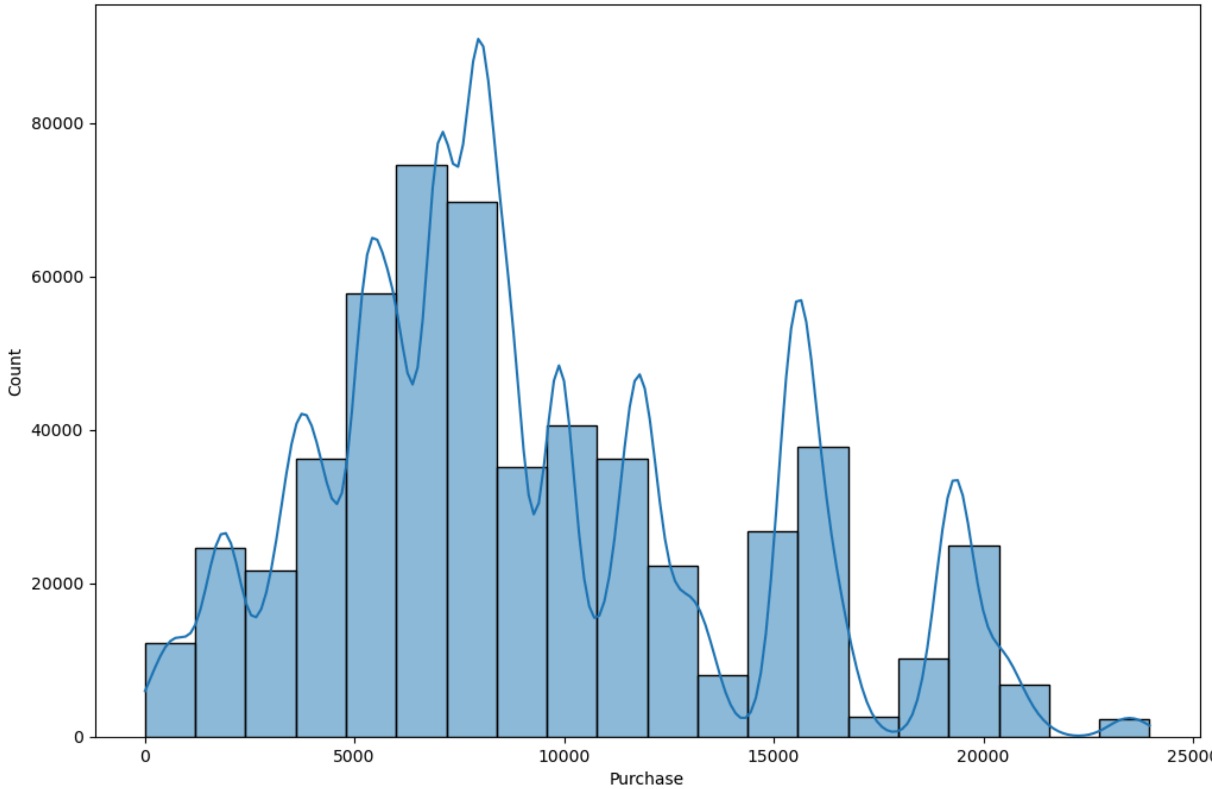
In [20]:
```
Avg_female_spending
```

Out[20]:
```
8734.565765155476
```

In [21]:
```
plt.figure(figsize=(8, 6))

# Plot boxplot to show spending distribution by gender
sns.boxplot(data=df,y='Purchase', palette='Set2')

# Add title and labels
plt.title("Box Plot of Purchases by Gender")

plt.ylabel("Purchase Amount")

# Show the plot
plt.show()
```



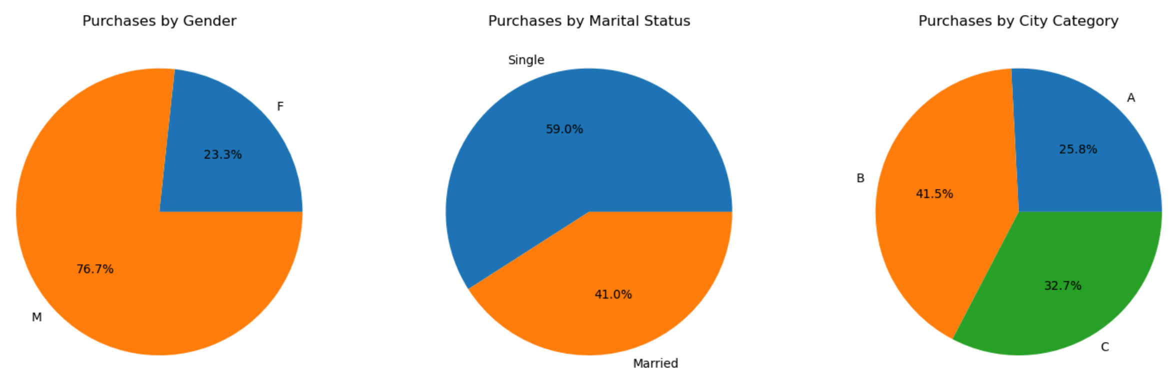
In [22]:
```
plt.figure(figsize=(12,8))
sns.histplot(data=df,x='Purchase',bins=20,kde=True)
plt.show()
```



Based on the above plots spends above 21000 can be considered as outliers.

In [23]:
```
Gender_purchase = df.groupby('Gender')['Purchase'].sum()
Maritalstatus_purchase = df.groupby('Marital_Status')['Purchase'].sum()
City_Category_purchase = df.groupby('City_Category')['Purchase'].sum()
plt.figure(figsize=(18, 6))
plt.subplot(1,3,1)
plt.pie(Gender_purchase, labels=Gender_purchase.index,autopct='%1.1f%%')
plt.title("Purchases by Gender")
plt.subplot(1,3,2)
plt.pie(Maritalstatus_purchase, labels=Maritalstatus_purchase.index,autopct='%1.1f%%
plt.title("Purchases by Marital Status")
plt.subplot(1,3,3)
plt.pie(City_Category_purchase, labels=City_Category_purchase.index,autopct='%1.1f%%
plt.title("Purchases by City Category")
```

Out[23]:
```
Text(0.5, 1.0, 'Purchases by City Category')
```

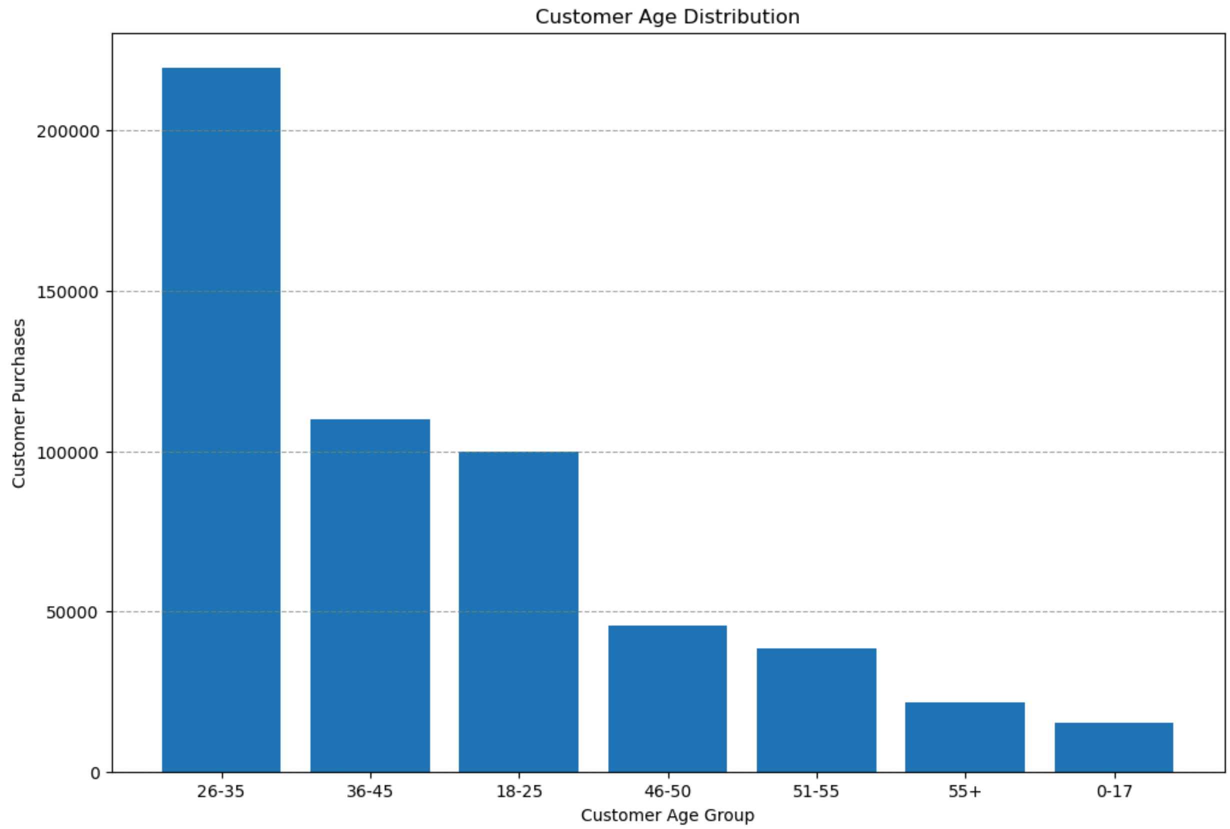


Based on the above data

1. Male customers are purchasing mpre than the female cutomers.
2. Single customers are making more purchases than married customers.
3. Customers from type B cities are making significantly more purchases than the customers from Type A, C cities.

Age distribution of the customers

In [24]:
```
Customer_Age=df['Age'].value_counts()
plt.figure(figsize=(12,8))
plt.bar(Customer_Age.index,Customer_Age.values)
plt.xlabel('Customer Age Group')
plt.ylabel('Customer Purchases')
plt.title("Customer Age Distribution")
plt.grid(axis='y', linestyle='--', color='grey', alpha=0.7)
```

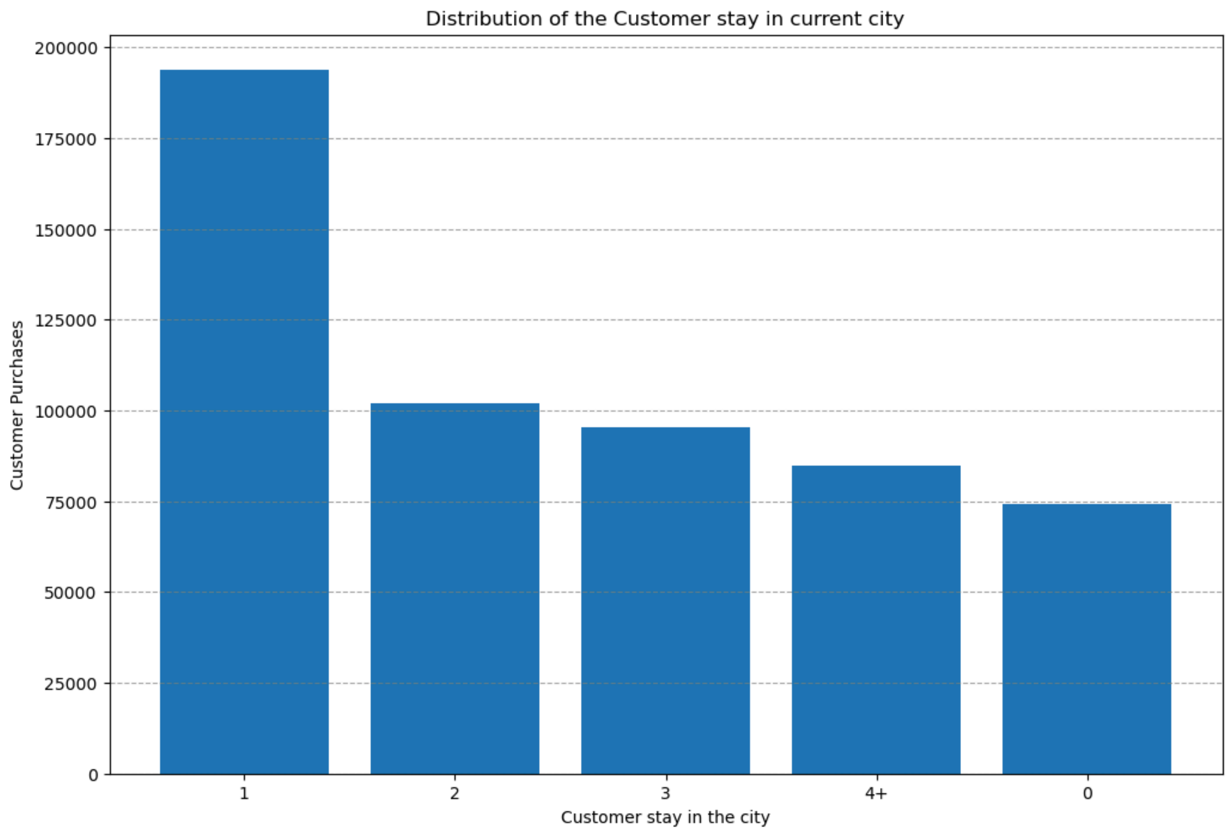


The insights from the above data are:

1. Customers aged 26 to 35 made the most purchases during the Black Friday sale, indicating that young adults were the primary buyers.
2. The next highest purchasing age groups were 36 to 45, followed by 18 to 25, showing that middle-aged and young customers drove most of the sales.
3. Purchases generally decreased as age increased, suggesting that customers under 45 were the most active in the sale.

Distribution of the Customer stay in current city

In [25]:
```
Customer_stay=df['Stay_In_Current_City_Years'].value_counts()
plt.figure(figsize=(12,8))
plt.bar(Customer_stay.index,Customer_stay.values)
plt.xlabel('Customer stay in the city')
plt.ylabel('Customer Purchases')
plt.title("Distribution of the Customer stay in current city")
plt.grid(axis='y', linestyle='--', color='grey', alpha=0.7)
```

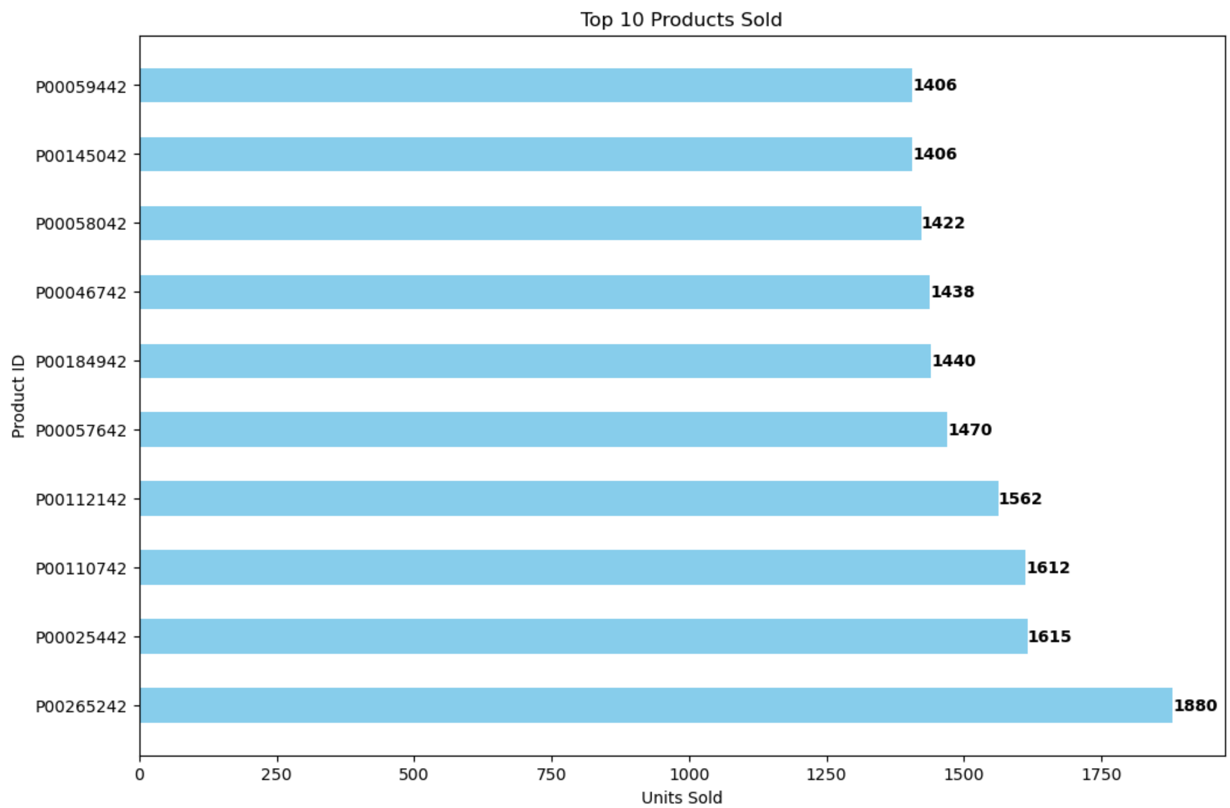


The insights from the above data are:

1. The data indicates that customers who have stayed in their current city for 1 year made the most purchases.
2. They are followed by customers with city tenures of 2, 3, and 4+ years.
3. Purchases tend to decrease with longer stays, which could be addressed by offering loyalty-based discounts, special deals, and a focus on quality for long-term customers.

Top Products and Products Category sales distribution

In [26]:
```
Top_products = df['Product_ID'].value_counts()[0:10]

plt.figure(figsize=(12, 8))

plt.barh(Top_products.index, Top_products.values, height=0.5, color="skyblue")

for index, value in enumerate(Top_products.values):
    plt.text(value, index, str(value), va='center', ha='left', fontweight='bold')
plt.ylabel('Product ID')
plt.xlabel('Units Sold')
plt.title("Top 10 Products Sold")

plt.show()
```



The products sold on Walmart during the Black Friday sale showed minimal variation.

Data Exploration and Statistical Analysis of Customer Spending Patterns

```
Analyzing Gender-Based Spending Trends and Confidence Intervals
```

In [27]:
```
male_customers = df[df['Gender'] == 'M']
female_customers = df[df['Gender'] == 'F']
```

In [34]:
```
avg_m=male_customers['Purchase'].mean()
std_m=male_customers['Purchase'].std()
n=df.shape[0]
confidence_level=0.90
std_m
z_value = st.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_value * (std_m / (n ** 0.5))
clt_male = (avg_m - margin_of_error, avg_m + margin_of_error)
print("Confidence interval for average spending by female customers:", clt_male)
```

```
Confidence interval for average spending by female customers: (9426.232676040503, 9
448.819404904027)
```

In [35]:
```
avg_f=female_customers['Purchase'].mean()
std_f=female_customers['Purchase'].std()
n=df.shape[0]
confidence_level=0.90
std_f
z_value = st.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_value * (std_m / (n ** 0.5))
clt_female = (avg_f - margin_of_error, avg_f + margin_of_error)
print("Confidence interval for average spending by female customers:", clt_female)
```

```
Confidence interval for average spending by female customers: (8723.272400723714, 8
745.859129587237)
```

Insights from the above data:

1. The 90% confidence interval for average sales among male customers ranges from $9,426 to$ 9,448.
2. The 90% confidence interval for average sales among female customers ranges from $8,723 to$ 8,745.

In [28]:
```
avg_m=male_customers['Purchase'].mean()
std_m=male_customers['Purchase'].std()
n=df.shape[0]
confidence_level=0.95
std_m
z_value = st.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_value * (std_m / (n ** 0.5))
clt_male = (avg_m - margin_of_error, avg_m + margin_of_error)
print("Confidence interval for average spending by female customers:", clt_male)
```

```
Confidence interval for average spending by female customers: (9424.069166749367, 9
450.982914195163)
```

In [31]:
```
avg_f=female_customers['Purchase'].mean()
std_f=female_customers['Purchase'].std()
n=df.shape[0]
confidence_level=0.95
std_f
z_value = st.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_value * (std_m / (n ** 0.5))
clt_female = (avg_f - margin_of_error, avg_f + margin_of_error)
print("Confidence interval for average spending by female customers:", clt_female)
```

```
Confidence interval for average spending by female customers: (8721.967628767923, 8
747.163901543028)
```

```
Insights from the above data:
1. The 95% confidence interval for average sales among male customers ranges from
$9,424 to $9,450.
2. The 95% confidence interval for average sales among female customers ranges from
$8,722 to $8,747.
```

In [32]:
```
avg_m=male_customers['Purchase'].mean()
std_m=male_customers['Purchase'].std()
n=df.shape[0]
confidence_level=0.99
std_m
z_value = st.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_value * (std_m / (n ** 0.5))
clt_male = (avg_m - margin_of_error, avg_m + margin_of_error)
print("Confidence interval for average spending by female customers:", clt_male)
```

```
Confidence interval for average spending by female customers: (9419.840710566708, 9
455.211370377821)
```

In [33]:
```
avg_f=female_customers['Purchase'].mean()
std_f=female_customers['Purchase'].std()
n=df.shape[0]
confidence_level=0.95
std_f
z_value = st.norm.ppf((1 + confidence_level) / 2)
margin_of_error = z_value * (std_m / (n ** 0.5))
clt_female = (avg_f - margin_of_error, avg_f + margin_of_error)
print("Confidence interval for average spending by female customers:", clt_female)
```

```
Confidence interval for average spending by female customers: (8721.108891432577, 8
748.022638878374)
```

Insights from the above data:

1. The 99% confidence interval for average sales among male customers ranges from $9,419 to$ 9,455.
2. The 99% confidence interval for average sales among female customers ranges from $8,721 to$ 8,748.

1. The average sales show minimal variation across confidence levels of 90%, 95%, and 99%. Additionally, the average purchases for male and female customers do not overlap, with male spending exceeding female spending by approximately $750.
2. Despite the smaller spending difference, the lower percentage of female customers suggests an opportunity for Walmart to increase engagement by running gender-specific ad campaigns and targeted offers to attract more female shoppers.