In [1]:
```
import pandas as pd
import numpy as np
```

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

# Initial data preview
print("Initial Dataset:")
print(df.head())
```

```
Initial Dataset:
   Order ID Customer Name      Gender         Category  Quantity   Price  \
0  ORD-1000    Customer 0        Male      Electronics         6  363.75
1  ORD-1001    Customer 1        Male  Health & Beauty         8     NaN
2  ORD-1002    Customer 2      Female            Books         8  452.38
3  ORD-1003    Customer 3  Non-Binary      Electronics         8  437.44
4  ORD-1004    Customer 4      Female  Health & Beauty         1  353.61

         State Payment Method Purchase Date    Total
0   California           Cash    2023-01-01  2182.50
1     New York         PayPal    2023-01-02      NaN
2     New York         PayPal    2023-01-03  3619.04
3     New York         PayPal    2023-01-04  3499.52
4     New York    Credit Card    2023-01-05   353.61
```

In [3]:
```
# Check for missing values
print("\nMissing Values:")
print(df.isnull().sum())
```

```
Missing Values:
Order ID           0
Customer Name     12
Gender             0
Category           0
Quantity           0
Price             11
State              0
Payment Method     0
Purchase Date      0
Total             11
dtype: int64
```

In [4]:
```
# Fill missing Customer Names with "Anonymous"
df['Customer Name'] = df['Customer Name'].fillna('Anonymous')
```

In [5]:
```
df.head()
```

Out[5]:

	Order ID	Customer Name	Gender	Category	Quantity	Price	State	Payment Method	Purchas Dat
0	ORD-1000	Customer 0	Male	Electronics	6	363.75	California	Cash	2023-01 0
1	ORD-1001	Customer 1	Male	Health & Beauty	8	NaN	New York	PayPal	2023-01 0
2	ORD-1002	Customer 2	Female	Books	8	452.38	New York	PayPal	2023-01 0
3	ORD-1003	Customer 3	Non-Binary	Electronics	8	437.44	New York	PayPal	2023-01 0
4	ORD-1004	Customer 4	Female	Health & Beauty	1	353.61	New York	Credit Card	2023-01 0

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

Out[7]:

	Order ID	Customer Name	Gender	Category	Quantity	Price	State	Payment Method	Purcha Da
95	ORD-1095	Customer 95	Non-Binary	Health & Beauty	5	348.53	Florida	PayPal	2023-0
96	ORD-1096	Customer 96	Non-Binary	Home Decor	6	247.83	Florida	Debit Card	2023-0
97	ORD-1097	Customer 97	Non-Binary	Health & Beauty	6	204.74	California	Cash	2023-0
98	ORD-1098	Customer 98	Female	Electronics	4	275.52	Texas	Debit Card	2023-0
99	ORD-1099	Customer 99	Female	Clothing	6	20.89	New York	Cash	2023-0

In [8]:
```
df
```

Out[8]:

	Order ID	Customer Name	Gender	Category	Quantity	Price	State	Payment Method	Purcha Da
0	ORD-1000	Customer 0	Male	Electronics	6	363.75	California	Cash	2023-0
1	ORD-1001	Customer 1	Male	Health & Beauty	8	NaN	New York	PayPal	2023-0
2	ORD-1002	Customer 2	Female	Books	8	452.38	New York	PayPal	2023-0
3	ORD-1003	Customer 3	Non-Binary	Electronics	8	437.44	New York	PayPal	2023-0
4	ORD-1004	Customer 4	Female	Health & Beauty	1	353.61	New York	Credit Card	2023-0
...	...	...	...	...	...	...	...	...	
95	ORD-1095	Customer 95	Non-Binary	Health & Beauty	5	348.53	Florida	PayPal	2023-0
96	ORD-1096	Customer 96	Non-Binary	Home Decor	6	247.83	Florida	Debit Card	2023-0
97	ORD-1097	Customer 97	Non-Binary	Health & Beauty	6	204.74	California	Cash	2023-0
98	ORD-1098	Customer 98	Female	Electronics	4	275.52	Texas	Debit Card	2023-0
99	ORD-1099	Customer 99	Female	Clothing	6	20.89	New York	Cash	2023-0

100 rows × 10 columns

In [9]:
```
# Fill missing Prices with the mean price
mean_price = df['Price'].mean()
df['Price'] = df['Price'].fillna(mean_price)
```

In [10]:
```
df
```

Out[10]:

	Order ID	Customer Name	Gender	Category	Quantity	Price	State	Payment Method	Pu
0	ORD-1000	Customer 0	Male	Electronics	6	363.750000	California	Cash	2(
1	ORD-1001	Customer 1	Male	Health & Beauty	8	279.048427	New York	PayPal	2(
2	ORD-1002	Customer 2	Female	Books	8	452.380000	New York	PayPal	2(
3	ORD-1003	Customer 3	Non-Binary	Electronics	8	437.440000	New York	PayPal	2(
4	ORD-1004	Customer 4	Female	Health & Beauty	1	353.610000	New York	Credit Card	2(
...	...	...	...	...	...	...	...	...	
95	ORD-1095	Customer 95	Non-Binary	Health & Beauty	5	348.530000	Florida	PayPal	2(
96	ORD-1096	Customer 96	Non-Binary	Home Decor	6	247.830000	Florida	Debit Card	2(
97	ORD-1097	Customer 97	Non-Binary	Health & Beauty	6	204.740000	California	Cash	2(
98	ORD-1098	Customer 98	Female	Electronics	4	275.520000	Texas	Debit Card	2(
99	ORD-1099	Customer 99	Female	Clothing	6	20.890000	New York	Cash	2(

100 rows × 10 columns

In [11]:
```
print(df.isnull().sum())
```

```
Order ID           0
Customer Name      0
Gender             0
Category           0
Quantity           0
Price              0
State              0
Payment Method     0
Purchase Date      0
Total             11
dtype: int64
```

In [12]:
```
# Recalculate Total
df['Total'] = df['Quantity'] * df['Price']
```

In [13]:
```
df.head()
```

Out[13]:

	Order ID	Customer Name	Gender	Category	Quantity	Price	State	Payment Method	Pur
0	ORD-1000	Customer 0	Male	Electronics	6	363.750000	California	Cash	202
1	ORD-1001	Customer 1	Male	Health & Beauty	8	279.048427	New York	PayPal	202
2	ORD-1002	Customer 2	Female	Books	8	452.380000	New York	PayPal	202
3	ORD-1003	Customer 3	Non-Binary	Electronics	8	437.440000	New York	PayPal	202
4	ORD-1004	Customer 4	Female	Health & Beauty	1	353.610000	New York	Credit Card	202

In [14]:
```
# Replace 'Unknown' in Category with 'Miscellaneous'
df['Category'] = df['Category'].replace('Unknown', 'Miscellaneous')
```

In [15]:
```
# Convert Purchase Date to datetime
df['Purchase Date'] = pd.to_datetime(df['Purchase Date'])
```

In [16]:
```
df.head()
```

Out[16]:

	Order ID	Customer Name	Gender	Category	Quantity	Price	State	Payment Method	Pur
0	ORD-1000	Customer 0	Male	Electronics	6	363.750000	California	Cash	202
1	ORD-1001	Customer 1	Male	Health & Beauty	8	279.048427	New York	PayPal	202
2	ORD-1002	Customer 2	Female	Books	8	452.380000	New York	PayPal	202
3	ORD-1003	Customer 3	Non-Binary	Electronics	8	437.440000	New York	PayPal	202
4	ORD-1004	Customer 4	Female	Health & Beauty	1	353.610000	New York	Credit Card	202

In [17]:
```
# Check cleaned data
print("\nCleaned Dataset:")
print(df.info())
```

```
Cleaned Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order ID        100 non-null    object
 1   Customer Name   100 non-null    object
 2   Gender          100 non-null    object
 3   Category        100 non-null    object
 4   Quantity        100 non-null    int64
 5   Price           100 non-null    float64
 6   State           100 non-null    object
 7   Payment Method  100 non-null    object
 8   Purchase Date   100 non-null    datetime64[ns]
 9   Total           100 non-null    float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(6)
memory usage: 7.9+ KB
None
```

In [18]:
```
# Save the cleaned dataset
cleaned_csv_path = 'cleaned_ecommerce_data.csv'
df.to_csv(cleaned_csv_path, index=False)
print(f"Cleaned dataset saved: {cleaned_csv_path}")
```

```
Cleaned dataset saved: cleaned_ecommerce_data.csv
```

In [23]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [24]:
```
# Set visualization style
sns.set_style('whitegrid')

# Top Categories by Revenue
category_revenue = df.groupby('Category')['Total'].sum().sort_values(ascending=F

plt.figure(figsize=(10, 6))
```

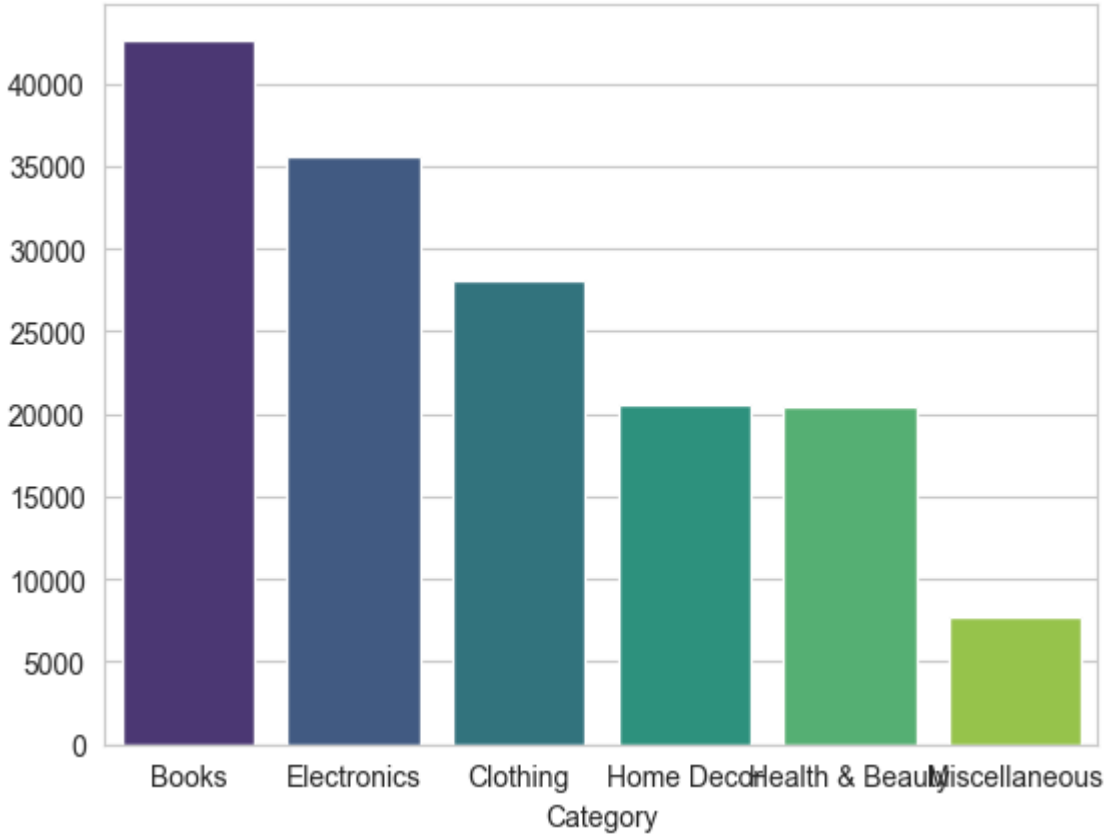
Out[24]:
```
<Figure size 1000x600 with 0 Axes>
```

```
<Figure size 1000x600 with 0 Axes>
```

In [25]:
```
sns.barplot(x=category_revenue.index, y=category_revenue.values, palette='viridi
```

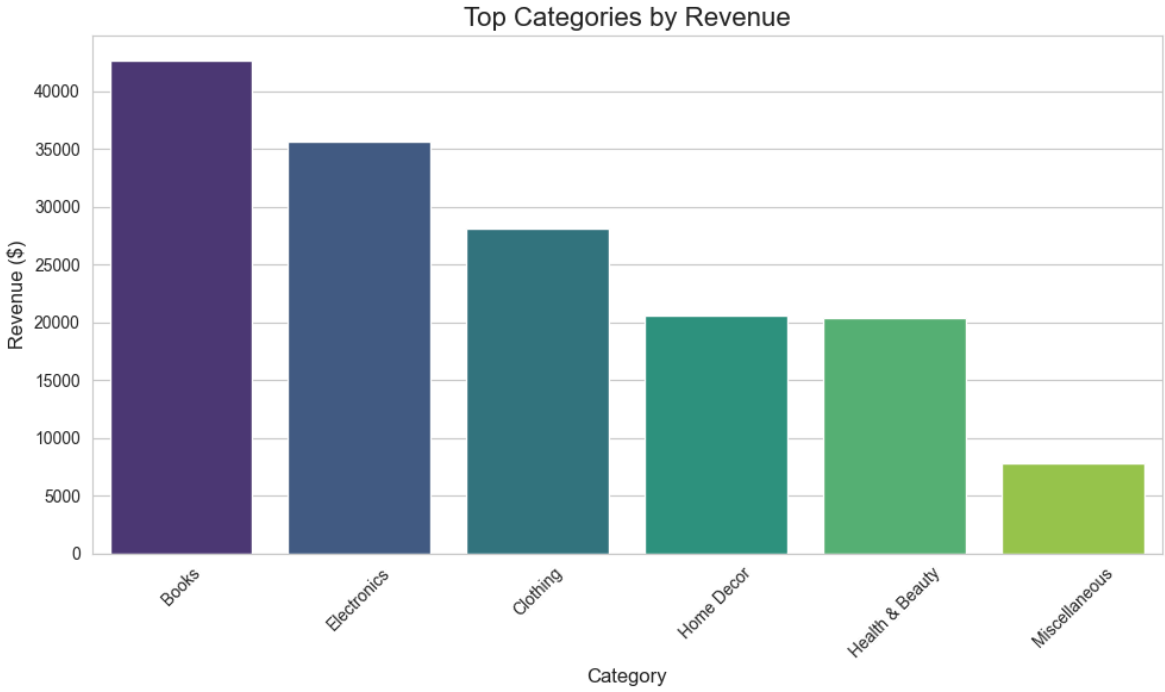
```
C:\Users\switc\AppData\Local\Temp\ipykernel_17820\3810512292.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x=category_revenue.index, y=category_revenue.values, palette='virid
is')
```

Out[25]:
```
<Axes: xlabel='Category'>
```

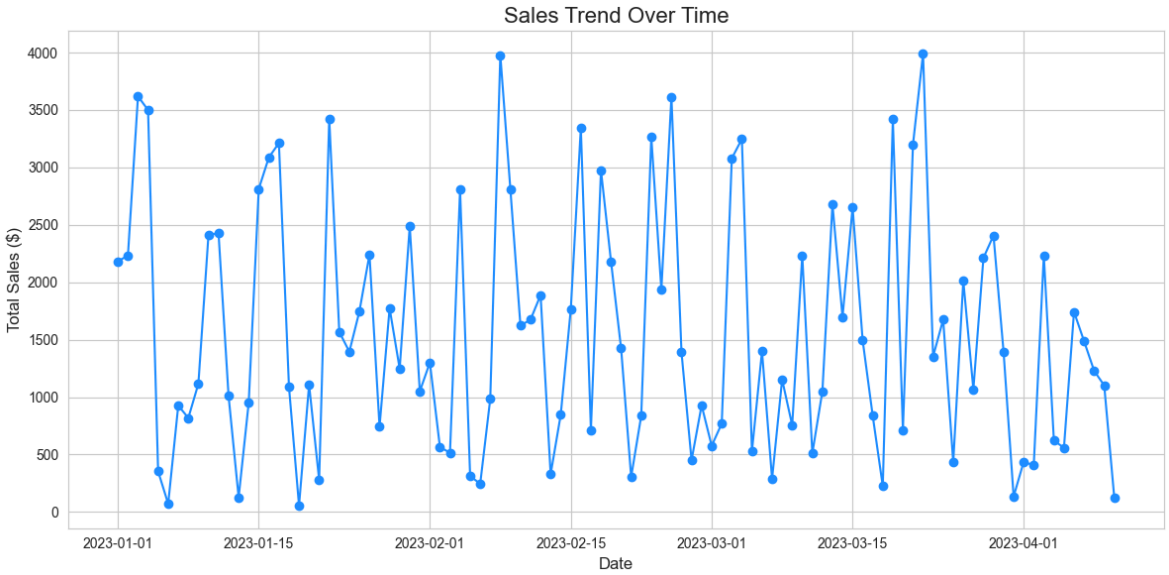


In [32]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x=category_revenue.index, y=category_revenue.values, palette='viridi
plt.title('Top Categories by Revenue', fontsize=16)
plt.xlabel('Category', fontsize=12)
plt.ylabel('Revenue ($)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\switc\AppData\Local\Temp\ipykernel_17820\3870076080.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.barplot(x=category_revenue.index, y=category_revenue.values, palette='virid
is')
```



In [28]:
```
# Sales Trend Over Time
sales_trend = df.groupby('Purchase Date')['Total'].sum()

plt.figure(figsize=(12, 6))
plt.plot(sales_trend.index, sales_trend.values, marker='o', color='dodgerblue')
plt.title('Sales Trend Over Time', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Total Sales ($)', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.savefig('sales_trend.png')
plt.show()
```



In [41]:
```
import plotly.express as px
```

In [40]:
```
# Payment Method Distribution
payment_counts = df['Payment Method'].value_counts()

fig = px.pie(
    values=payment_counts.values,
```

```
    names=payment_counts.index,
    title='Payment Method Distribution',
    color_discrete_sequence=px.colors.sequential.RdBu
)

fig.show()
```

In [ ]: