```python
import pandas as pd
import numpy as np
import os
import sys
import json
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.metrics import confusion_matrix

data = pd.read_csv("customer_shopping_data.csv")

data["invoice_date"] = pd.to_datetime(data["invoice_date"],
format="%d/%m/%Y")

data
```

```
      invoice_no customer_id  gender  age          category  quantity
price  \
0        I138884     C241288  Female   28          Clothing         5
1500.40
1        I317333     C111565    Male   21             Shoes         3
1800.51
2        I127801     C266599    Male   20          Clothing         1
300.08
3        I173702     C988172  Female   66             Shoes         5
3000.85
4        I337046     C189076  Female   53             Books         4
60.60
...          ...         ...     ...  ...               ...       ...
...
99452    I219422     C441542  Female   45          Souvenir         5
58.65
99453    I325143     C569580    Male   27  Food & Beverage          2
10.46
99454    I824010     C103292    Male   63  Food & Beverage          2
10.46
99455    I702964     C800631    Male   56        Technology         4
4200.00
99456    I232867     C273973  Female   36          Souvenir         3
35.19
```

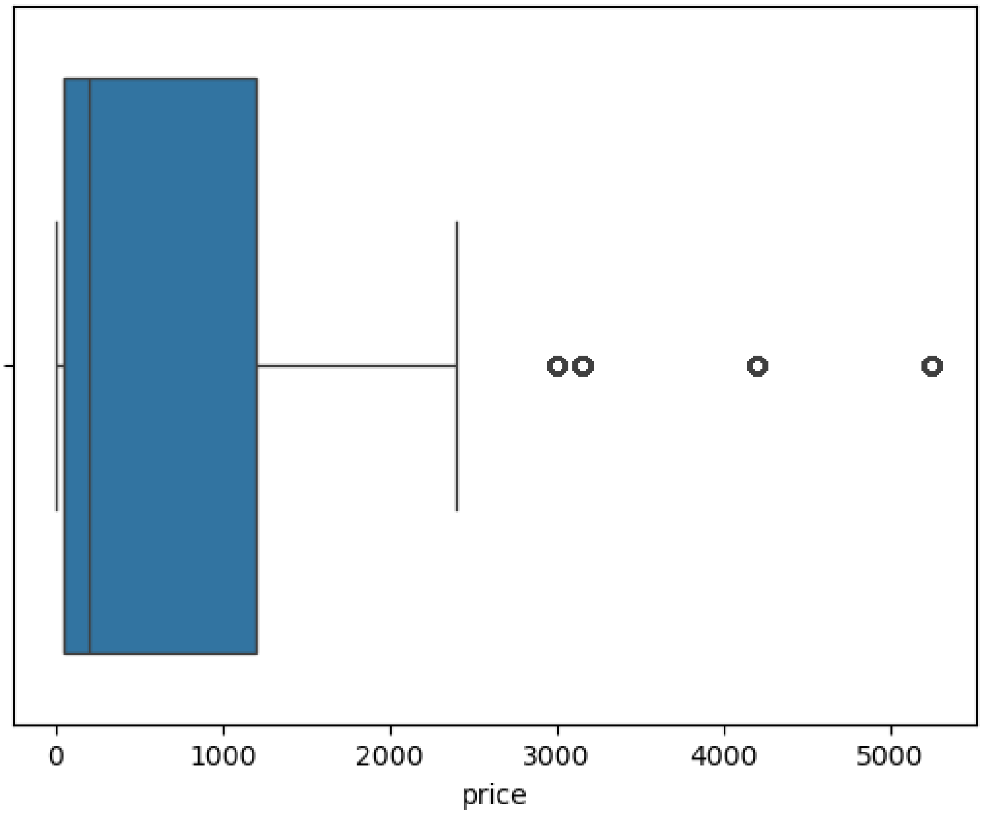
```
      payment_method invoice_date     shopping_mall
0        Credit Card   2022-08-05            Kanyon
1         Debit Card   2021-12-12    Forum Istanbul
2               Cash   2021-11-09         Metrocity
3        Credit Card   2021-05-16      Metropol AVM
4               Cash   2021-10-24            Kanyon
...              ...          ...               ...
99452    Credit Card   2022-09-21            Kanyon
99453           Cash   2021-09-22    Forum Istanbul
99454     Debit Card   2021-03-28         Metrocity
99455           Cash   2021-03-16      Istinye Park
99456    Credit Card   2022-10-15  Mall of Istanbul

[99457 rows x 10 columns]
```

```
# Display the first few rows
print(data.head())

# Check the shape of the dataset
print(f"Dataset shape: {data.shape}")

# Check column data types and missing values
print(data.info())

# Summary statistics for numerical columns
print(data.describe())
```

```
  invoice_no customer_id  gender  age  category  quantity    price  \
0    I138884     C241288  Female   28  Clothing         5  1500.40
1    I317333     C111565    Male   21     Shoes         3  1800.51
2    I127801     C266599    Male   20  Clothing         1   300.08
3    I173702     C988172  Female   66     Shoes         5  3000.85
4    I337046     C189076  Female   53     Books         4    60.60

  payment_method invoice_date   shopping_mall
0    Credit Card   2022-08-05          Kanyon
1     Debit Card   2021-12-12  Forum Istanbul
2           Cash   2021-11-09       Metrocity
3    Credit Card   2021-05-16    Metropol AVM
4           Cash   2021-10-24          Kanyon
Dataset shape: (99457, 10)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99457 entries, 0 to 99456
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   invoice_no      99457 non-null  object
 1   customer_id     99457 non-null  object
 2   gender          99457 non-null  object
 3   age             99457 non-null  int64
 4   category        99457 non-null  object
 5   quantity        99457 non-null  int64
 6   price           99457 non-null  float64
 7   payment_method  99457 non-null  object
 8   invoice_date    99457 non-null  datetime64[ns]
 9   shopping_mall   99457 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(2), object(6)
memory usage: 7.6+ MB
None
                age      quantity         price
invoice_date
count  99457.000000  99457.000000  99457.000000
99457
mean      43.427089      3.003429    689.256321  2022-02-04
02:46:59.783424
min       18.000000      1.000000      5.230000         2021-01-01
00:00:00
25%       30.000000      2.000000     45.450000         2021-07-19
00:00:00
50%       43.000000      3.000000    203.300000         2022-02-05
00:00:00
75%       56.000000      4.000000   1200.320000         2022-08-22
00:00:00
max       69.000000      5.000000   5250.000000         2023-03-08
00:00:00
std       14.990054      1.413025    941.184567
NaN
```

```
import seaborn as sns
sns.boxplot(x=data['price'])
```

```
<Axes: xlabel='price'>
```



Box: Represents the interquartile range (IQR)—the middle 50% of the data.

Median Line: The bold line inside the box denotes the median price.

Whiskers: These extend to the smallest and largest values within 1.5 times the IQR.

Outliers: Individual points beyond the whiskers are considered outliers.

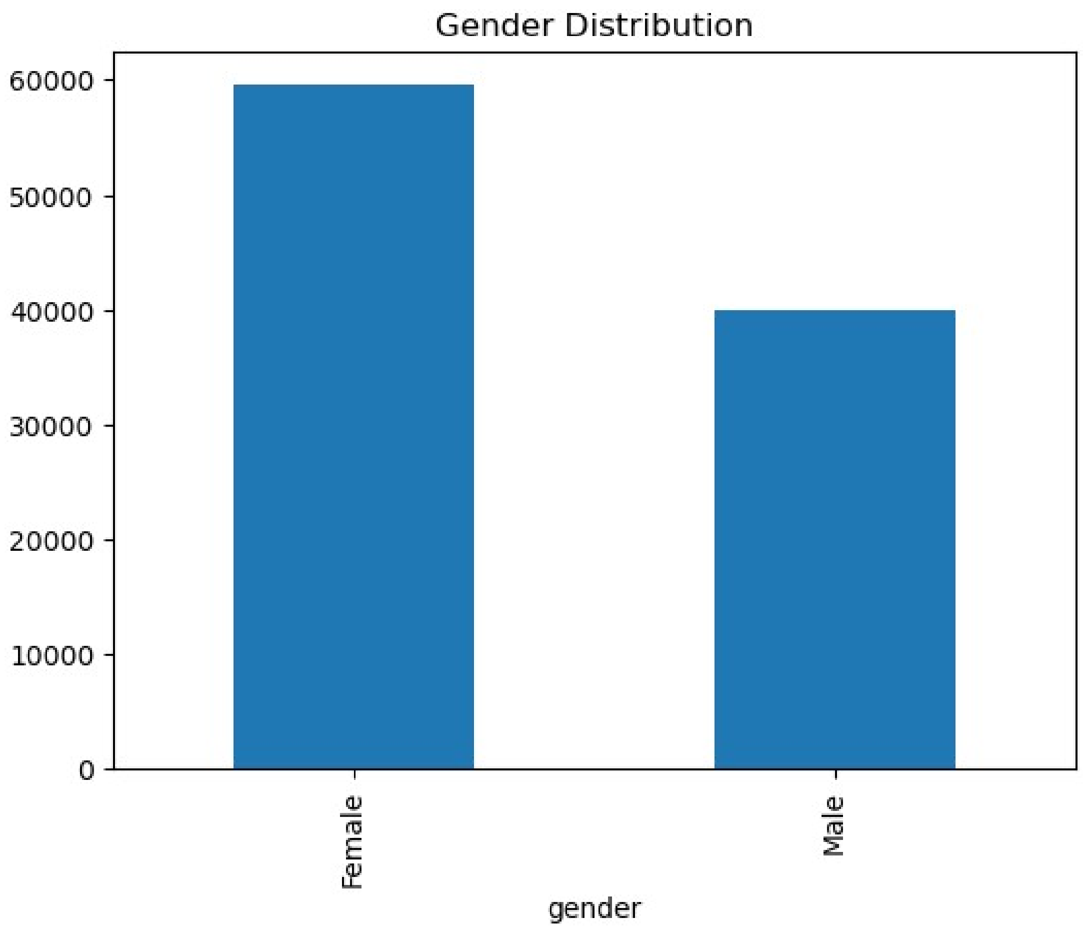
Exploratory Data Analysis

```
# Count the number of males and females
gender_counts = data['gender'].value_counts()

# Display the result
print(gender_counts)
```

```
gender
Female    59482
Male      39975
Name: count, dtype: int64
```

```
data['gender'].value_counts().plot(kind='bar', title='Gender
Distribution')
```

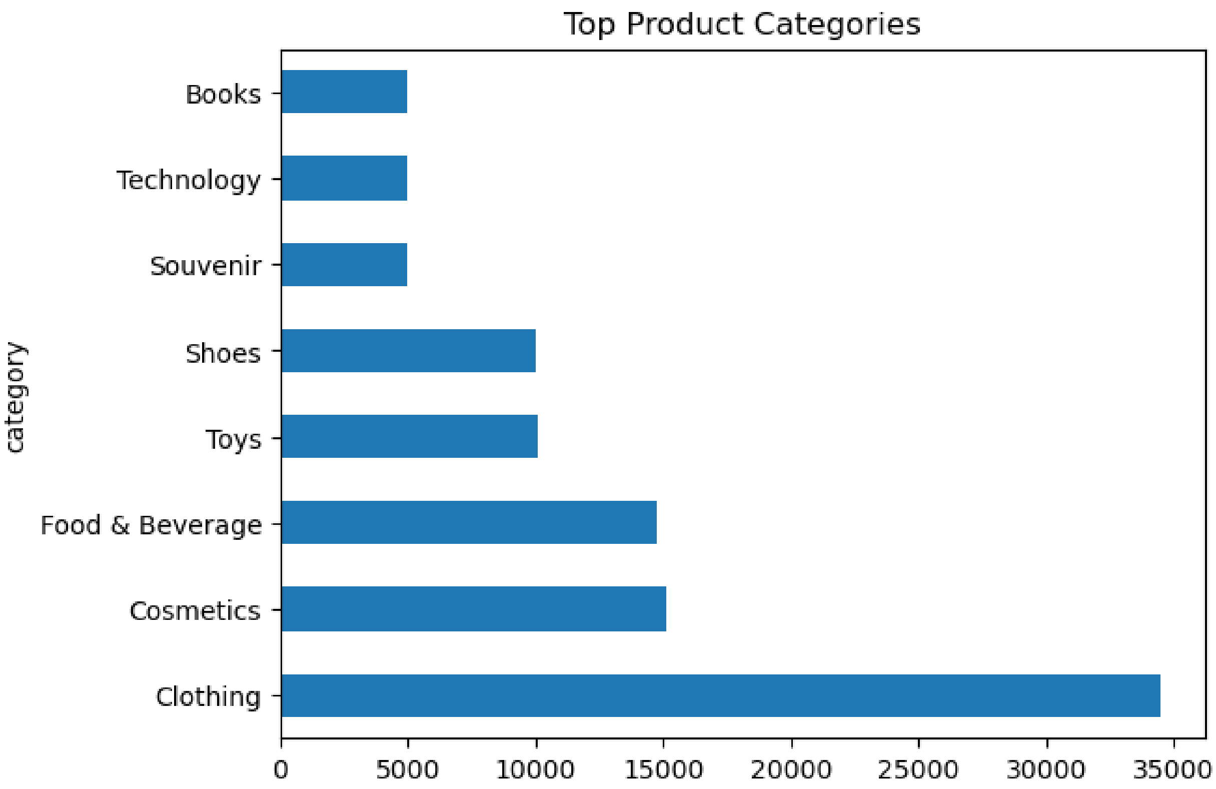
```
<Axes: title={'center': 'Gender Distribution'}, xlabel='gender'>
```



The plot indicates that there are more females than males in the data—about 59482 females compared to 39975 males.

```
data['category'].value_counts().plot(kind='barh', title='Top Product Categories')
```

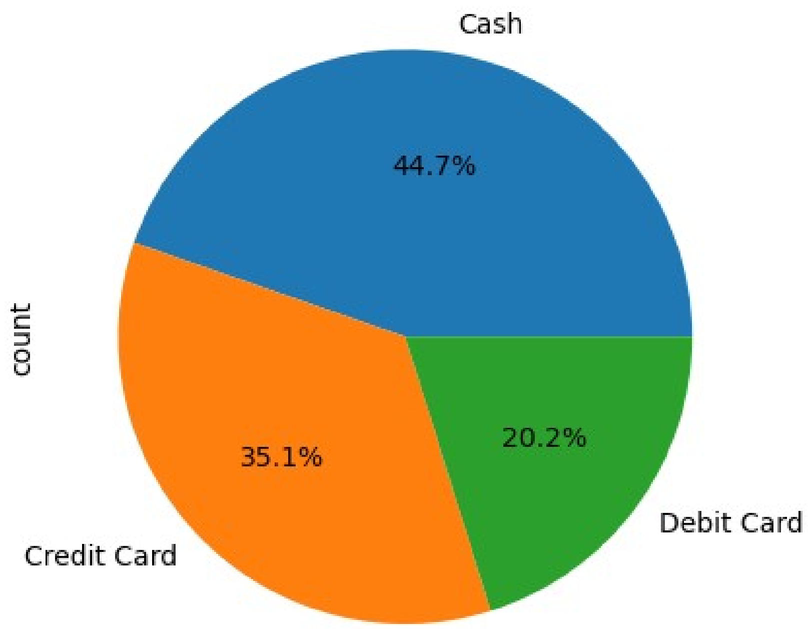
```
<Axes: title={'center': 'Top Product Categories'}, ylabel='category'>
```



The bar chart titled "Top Product Categories" visually represents the counts of various product categories. Clothing has the highest count, followed by Cosmetics and Food & Beverage. Books have the lowest count among the categories. This chart highlights the most popular product categories, making it valuable for market analysis and decision-making.

```
data['payment_method'].value_counts().plot(kind='pie', autopct='%1.1f%%')
```

```
<Axes: ylabel='count'>
```



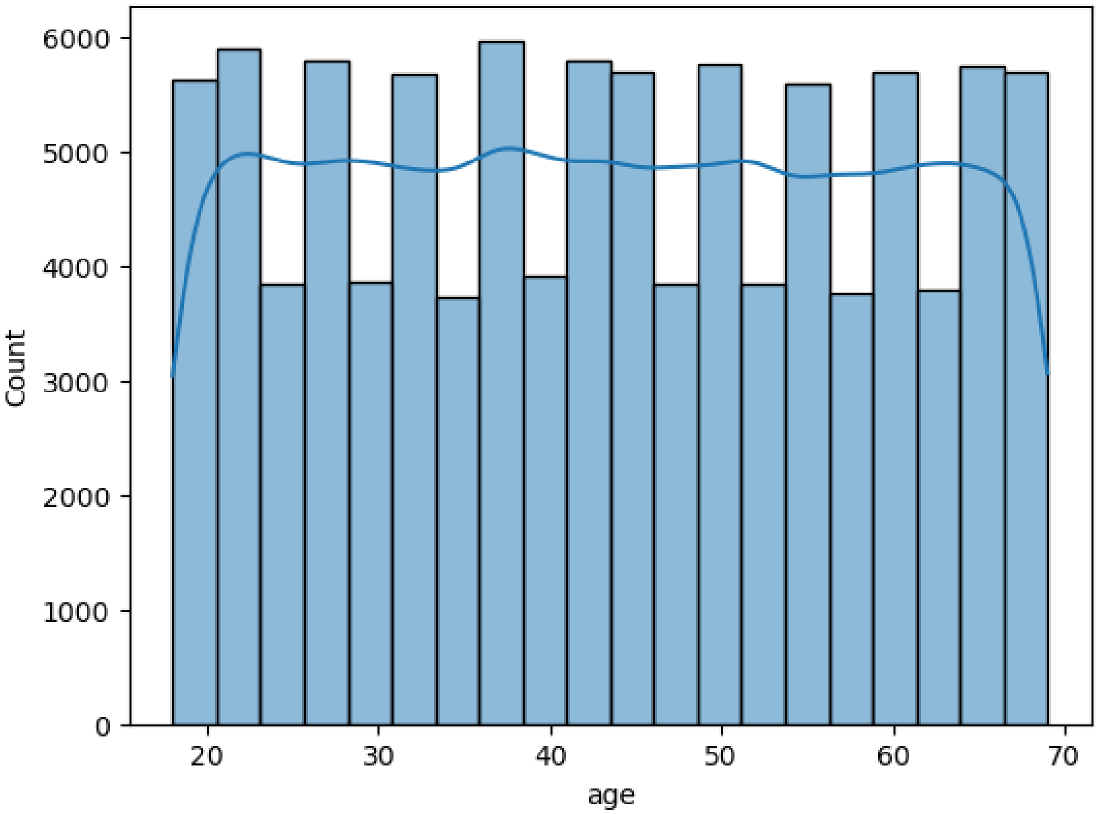
The pie chart shows the distribution of payment methods used by customers. The percentages are as follows:

- Cash: 44.7% (largest share)
- Credit Card: 35.1%
- Debit Card: 20.2% (smallest share)

This suggests that cash is the most preferred payment method, with credit and debit cards being less common.

```
sns.histplot(data['age'], bins=20, kde=True)
```

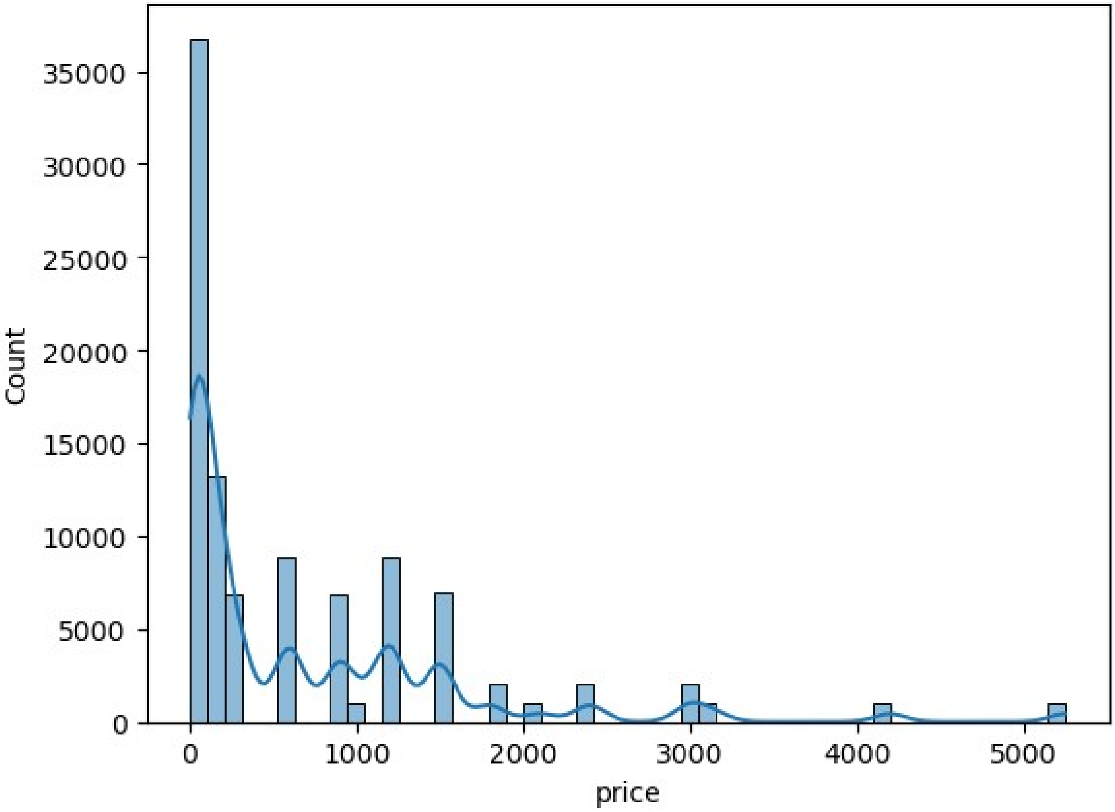
```
<Axes: xlabel='age', ylabel='Count'>
```



The histogram displays the age distribution in a dataset. It uses 20 bins to group ages and overlays a smoothed density curve (KDE). This helps identify patterns, such as peaks in certain age groups or overall trends in the data.

```
sns.histplot(data['price'], bins=50, kde=True)
```

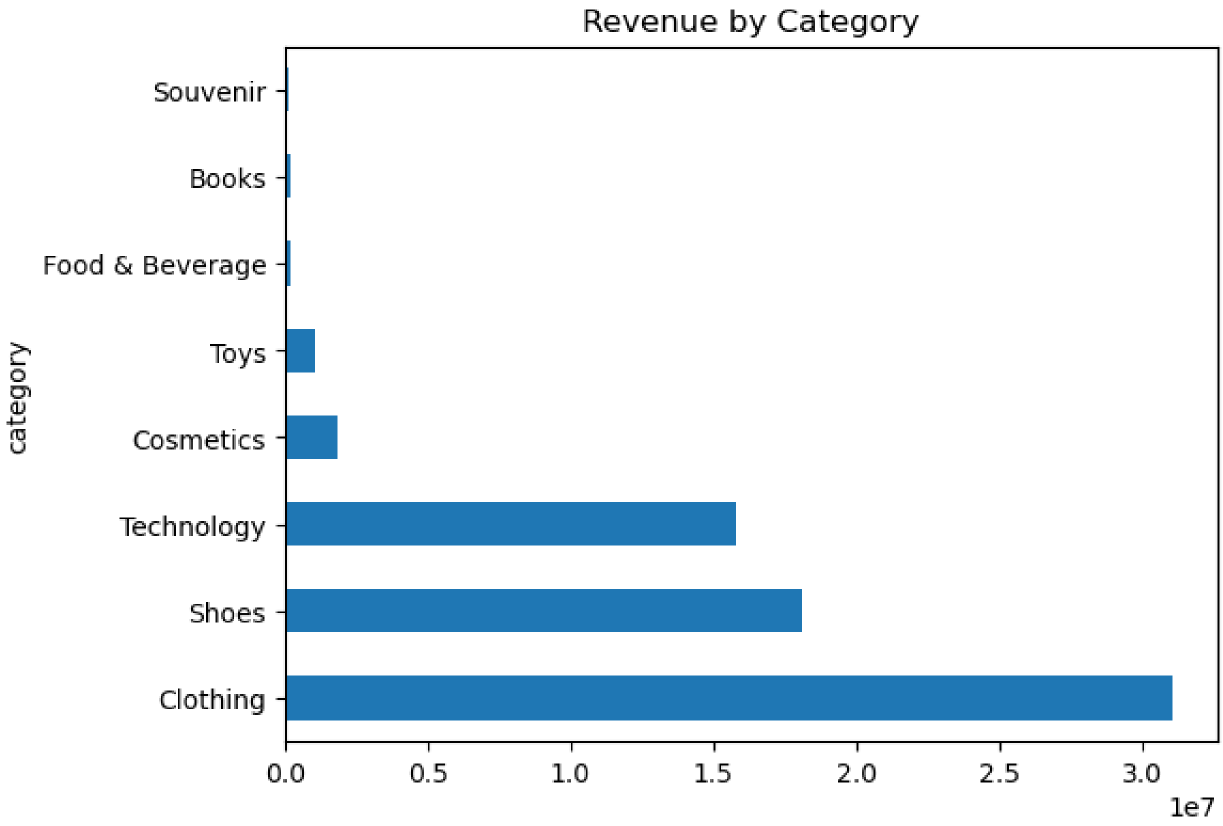
```
<Axes: xlabel='price', ylabel='Count'>
```



The histogram, along with its density curve (KDE), suggests that the majority of the prices in this dataset are clustered near lower values—close to zero. This might indicate that most products or services in the dataset are relatively inexpensive or affordable. The peaks at higher price ranges could represent premium or specialized products that occur less frequently but still hold significance. Overall, this pattern may help identify pricing strategies or categories of products that dominate the dataset.

```
(data.groupby('category')['price'].sum().sort_values(ascending=False)
 .plot(kind='barh', title='Revenue by Category'))
```

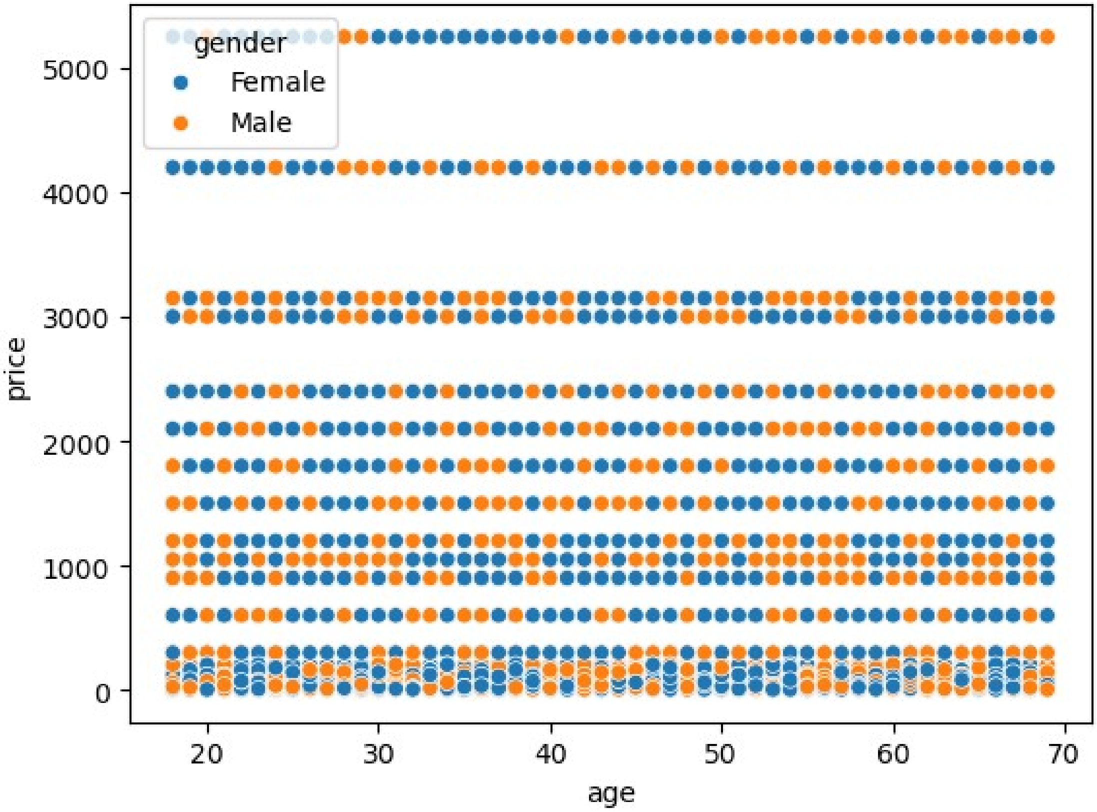
```
<Axes: title={'center': 'Revenue by Category'}, ylabel='category'>
```



The 'Clothing' category stands out as the top revenue earner, significantly outperforming the rest. Following it are 'Shoes' and 'Technology,' which also contribute meaningfully to the revenue. On the other end, categories like 'Books' and 'Souvenir' generate much less revenue in comparison. This insight is helpful for businesses—it could mean focusing more on high-performing categories while revisiting strategies for those that contribute less.

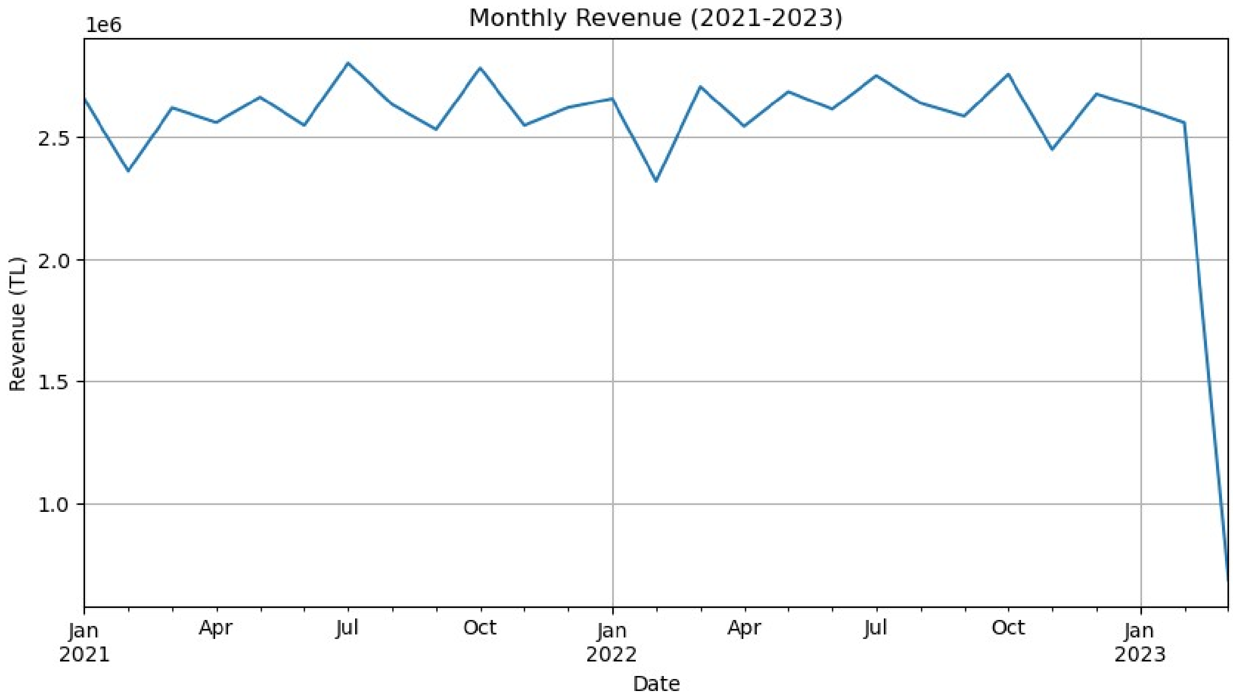
```
sns.scatterplot(x='age', y='price', hue='gender', data=data)
```

```
<Axes: xlabel='age', ylabel='price'>
```



Time-Based Trends

```
import matplotlib.pyplot as plt

# Resample by month-end and plot
monthly_revenue = data.set_index('invoice_date')
['price'].resample('ME').sum()
monthly_revenue.plot(title='Monthly Revenue (2021-2023)', figsize=(10,
5))
plt.ylabel('Revenue (TL)')
plt.xlabel('Date')
plt.grid(True)
plt.show()
```



```
#Identify exact months with peaks using:
print("Date of highest revenue :",monthly_revenue.idxmax())
print("Highest revenue value   :" ,monthly_revenue.max())
```

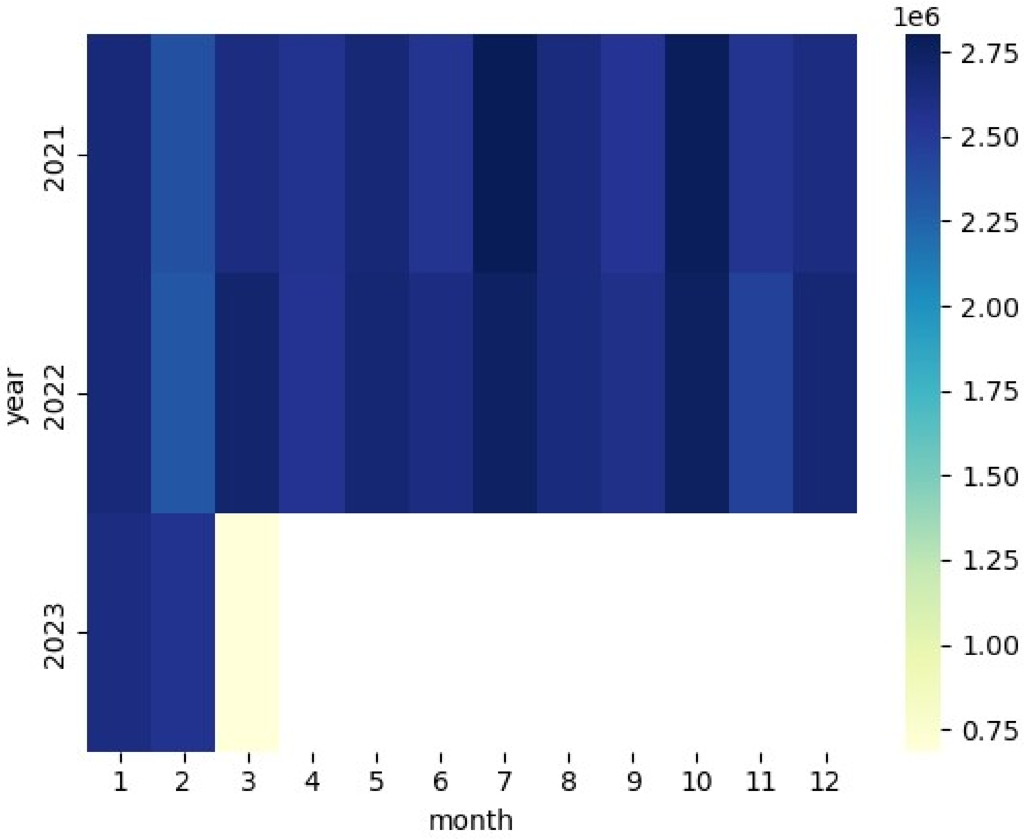
```
Date of highest revenue : 2021-07-31 00:00:00
Highest revenue value   : 2802468.58
```

```
data.head()
```

```
  invoice_no customer_id  gender  age  category  quantity    price  \
0    I138884     C241288  Female   28  Clothing         5  1500.40   
1    I317333     C111565    Male   21     Shoes         3  1800.51   
2    I127801     C266599    Male   20  Clothing         1   300.08   
3    I173702     C988172  Female   66     Shoes         5  3000.85   
4    I337046     C189076  Female   53     Books         4    60.60   

  payment_method invoice_date   shopping_mall  
0    Credit Card   2022-08-05         Kanyon  
1     Debit Card   2021-12-12  Forum Istanbul  
2           Cash   2021-11-09       Metrocity  
3    Credit Card   2021-05-16    Metropol AVM  
4           Cash   2021-10-24         Kanyon  
```

```
# Extract year and month for grouping
data['year'] = data['invoice_date'].dt.year
data['month'] = data['invoice_date'].dt.month
yearly_revenue = data.groupby(['year', 'month'])
['price'].sum().unstack()
sns.heatmap(yearly_revenue, cmap='YlGnBu')
```

```
<Axes: xlabel='month', ylabel='year'>
```



```
data
```

```
      invoice_no customer_id  gender  age         category  quantity
price  \
0        I138884     C241288  Female   28         Clothing         5
1500.40
1        I317333     C111565    Male   21            Shoes         3
1800.51
2        I127801     C266599    Male   20         Clothing         1
300.08
3        I173702     C988172  Female   66            Shoes         5
3000.85
4        I337046     C189076  Female   53            Books         4
60.60
...          ...         ...     ...  ...              ...       ...
...
99452    I219422     C441542  Female   45         Souvenir         5
58.65
99453    I325143     C569580    Male   27  Food & Beverage         2
10.46
99454    I824010     C103292    Male   63  Food & Beverage         2
```

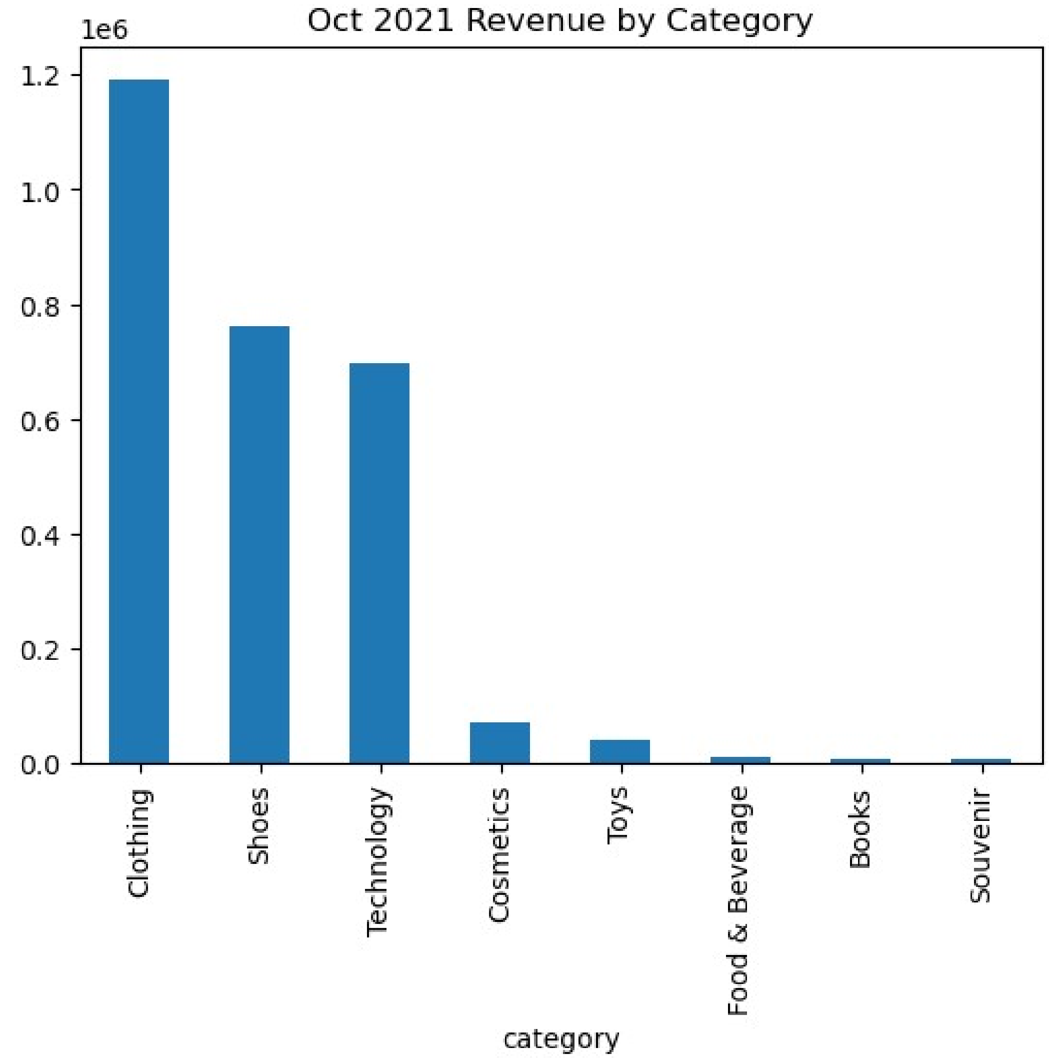
```
10.46
99455    I702964     C800631    Male   56       Technology         4
4200.00
99456    I232867     C273973  Female   36         Souvenir         3
35.19

      payment_method invoice_date     shopping_mall  year  month
0        Credit Card   2022-08-05            Kanyon  2022      8
1         Debit Card   2021-12-12    Forum Istanbul  2021     12
2               Cash   2021-11-09         Metrocity  2021     11
3        Credit Card   2021-05-16      Metropol AVM  2021      5
4               Cash   2021-10-24            Kanyon  2021     10
...              ...          ...               ...   ...    ...
99452    Credit Card   2022-09-21            Kanyon  2022      9
99453           Cash   2021-09-22    Forum Istanbul  2021      9
99454     Debit Card   2021-03-28         Metrocity  2021      3
99455           Cash   2021-03-16      Istinye Park  2021      3
99456    Credit Card   2022-10-15  Mall of Istanbul  2022     10

[99457 rows x 12 columns]
```

```
# Get top 5 months with highest revenue
top_months = monthly_revenue.sort_values(ascending=False).head(5)
print("Top Revenue Months:\n", top_months)
```

```
Top Revenue Months:
 invoice_date
2021-07-31    2802468.58
2021-10-31    2782418.40
2022-10-31    2755839.69
2022-07-31    2749554.99
2022-03-31    2705190.76
Name: price, dtype: float64
```

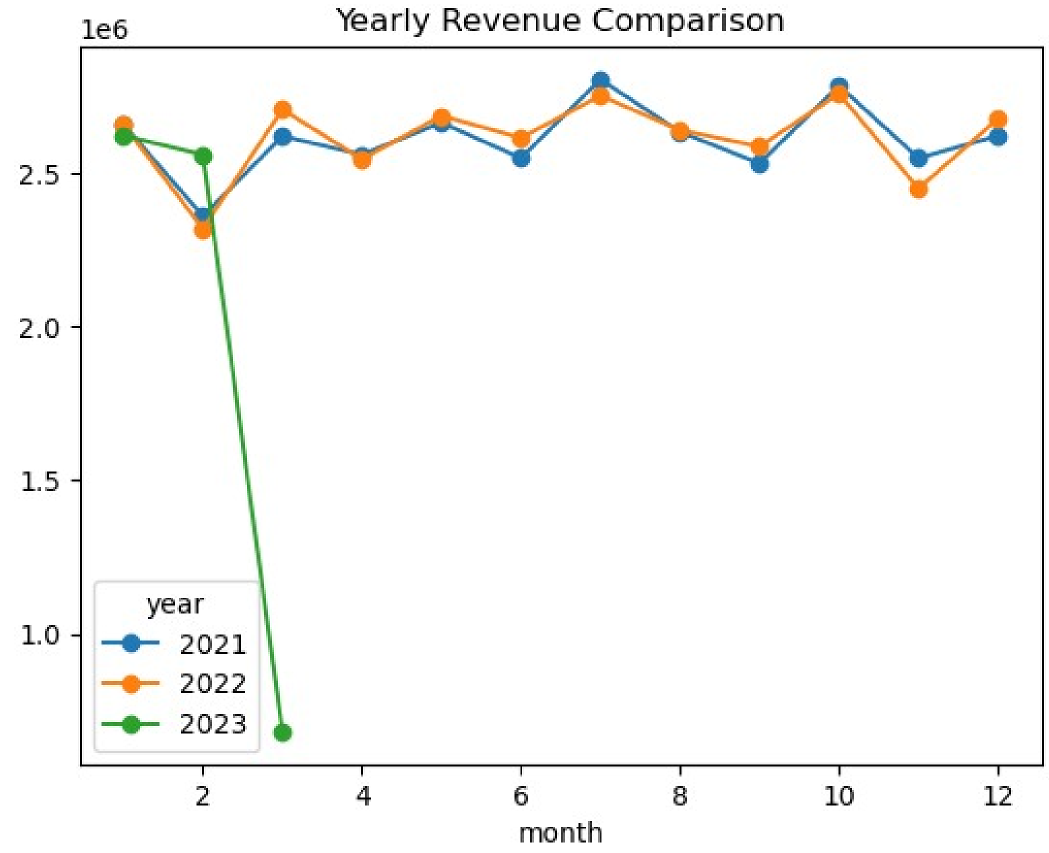
```
# Filter data for December 2022 and group by category
oct_2021 = data[data['invoice_date'].dt.strftime('%Y-%m') == '2021-
10']
top_categories = oct_2021.groupby('category')
['price'].sum().sort_values(ascending=False)
top_categories.plot(kind='bar', title='Oct 2021 Revenue by Category')
```

```
<Axes: title={'center': 'Oct 2021 Revenue by Category'},
xlabel='category'>
```



```
# Create a MultiIndex for grouping and unstacking
yearly_comparison = data.groupby(['year', 'month'])
['price'].sum().unstack(level=0)

# Plot growth for each year across months
yearly_comparison.plot(kind='line', marker='o', title='Yearly Revenue
Comparison')
```

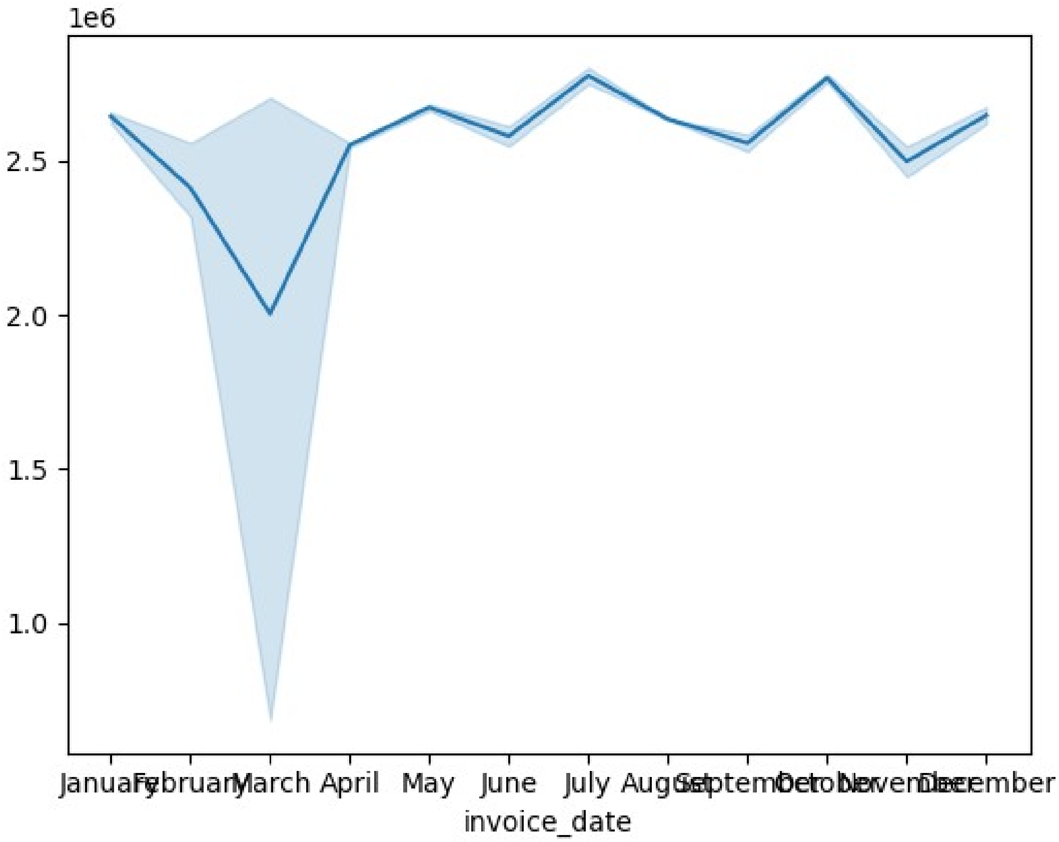
```
<Axes: title={'center': 'Yearly Revenue Comparison'}, xlabel='month'>
```



```
monthly_sales = data.set_index('invoice_date')
['price'].resample('M').sum()
sns.lineplot(x=monthly_sales.index.month_name(),
y=monthly_sales.values)
```

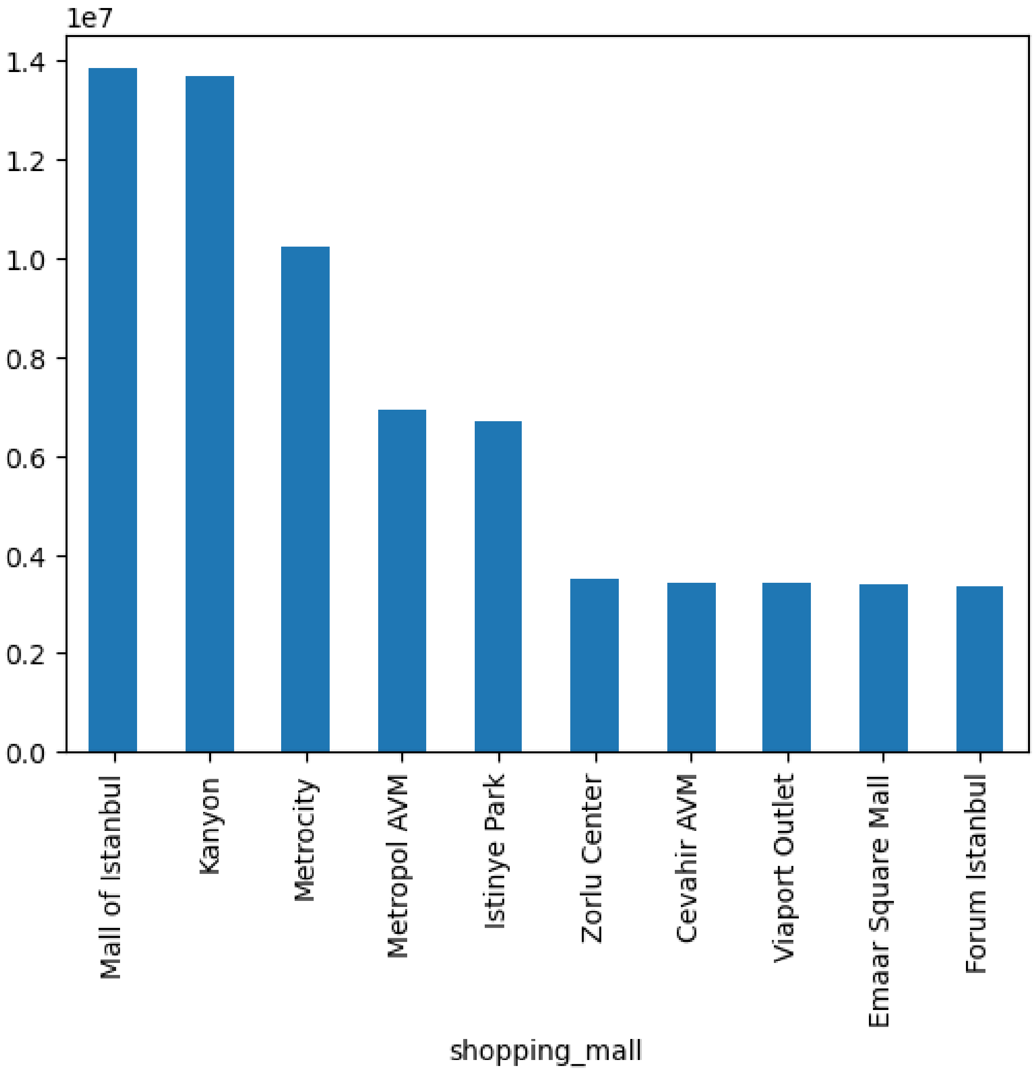
```
C:\Users\patil\AppData\Local\Temp\ipykernel_9200\2363964694.py:1:
FutureWarning: 'M' is deprecated and will be removed in a future
version, please use 'ME' instead.
  monthly_sales = data.set_index('invoice_date')
['price'].resample('M').sum()
```

```
<Axes: xlabel='invoice_date'>
```



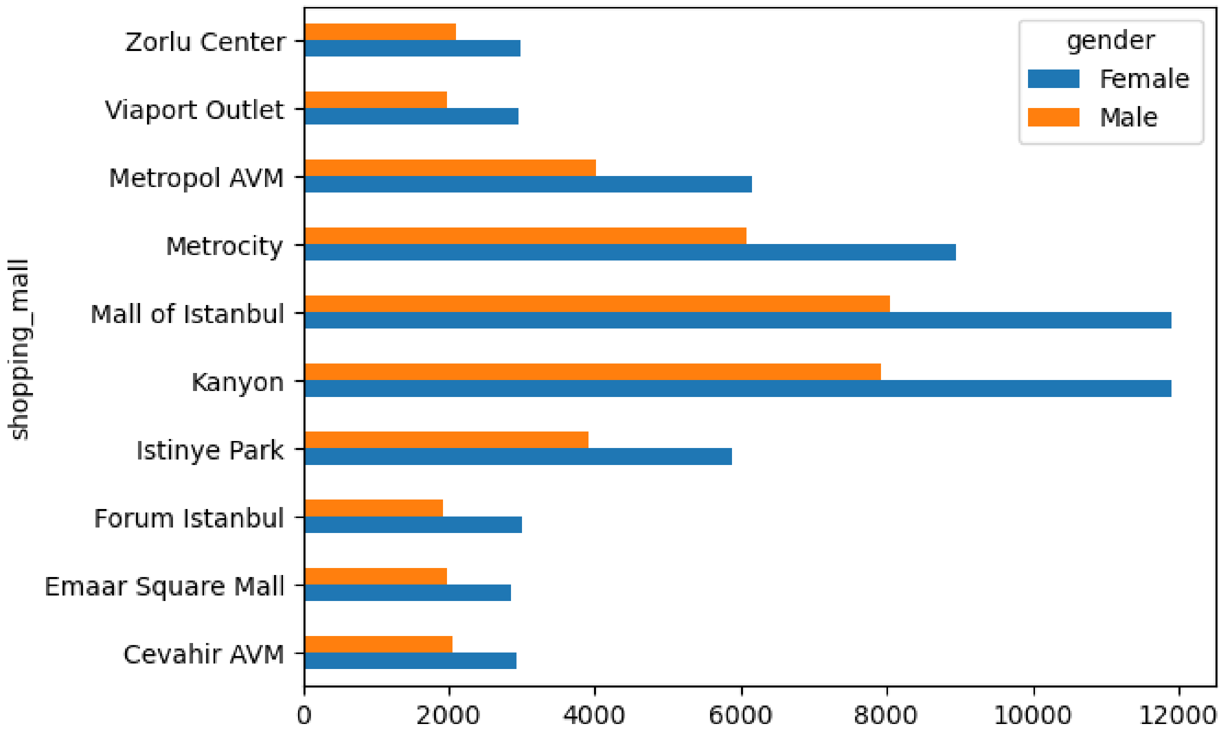
Shopping Mall Performance

```
(data.groupby('shopping_mall')['price'].sum()
 .sort_values(ascending=False).plot(kind='bar'))

<Axes: xlabel='shopping_mall'>
```

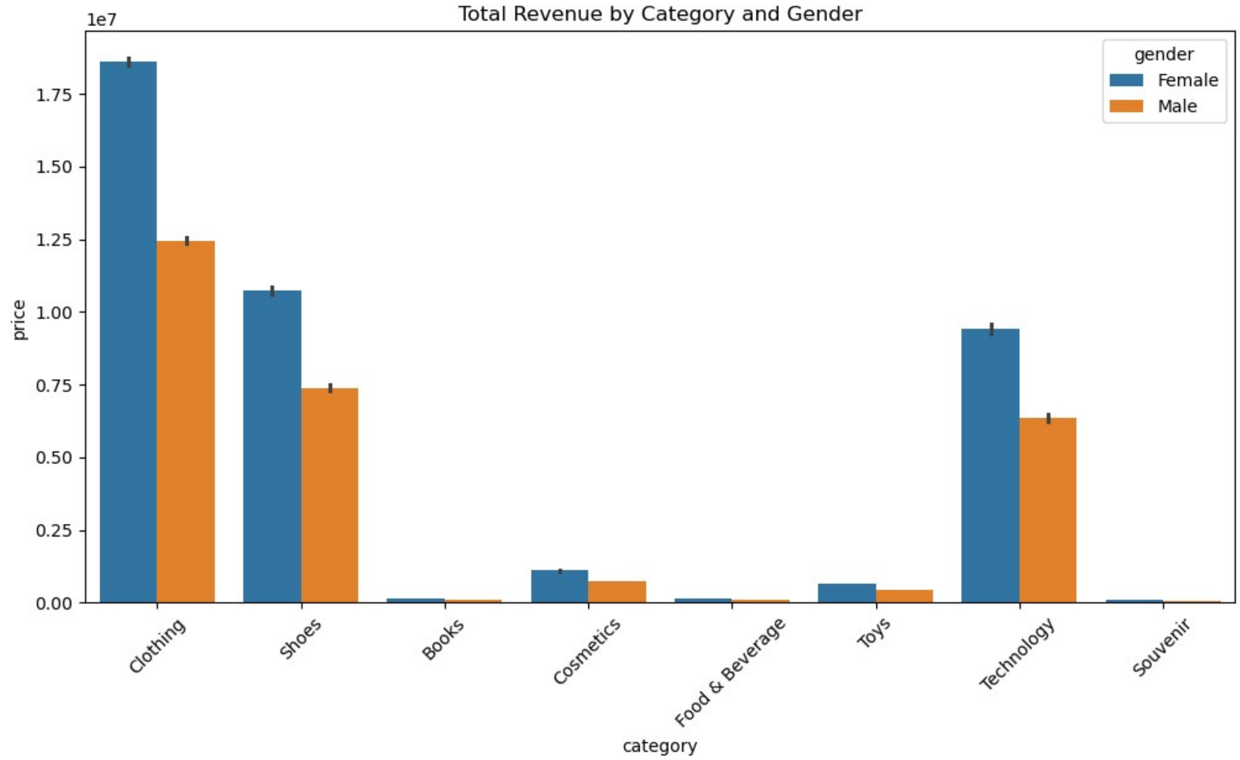


```
pd.crosstab(data['shopping_mall'], data['gender']).plot(kind='barh')
```

```
<Axes: ylabel='shopping_mall'>
```



```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(12, 6))
sns.barplot(x='category', y='price', hue='gender', data=data,
estimator=sum)
plt.xticks(rotation=45)
plt.title('Total Revenue by Category and Gender')
plt.show()
```



To identify the best shopping mall(s) to sell your 3 product categories, we'll analyze the dataset for:

1. Revenue potential (high sales volume/price for your categories).
2. Customer demographics (age/gender matching your target audience).
3. Competition (how saturated each mall is for your categories).

Filter Data for Your Categories

```
categories = ['Clothing', 'Technology', 'Souvenir']
filtered_data = data[data['category'].isin(categories)]
```

Analyze Revenue by Mall & Category

```
# Total revenue per mall for your categories
revenue_by_mall = filtered_data.groupby(['shopping_mall', 'category'])
['price'].sum().unstack()
revenue_by_mall['Total'] = revenue_by_mall.sum(axis=1)
revenue_by_mall.sort_values('Total', ascending=False, inplace=True)

print(revenue_by_mall)

category          Clothing  Souvenir  Technology       Total
shopping_mall
Mall of Istanbul  6245565.04  34263.33   3220350.0  9500178.37
```

```
Kanyon             6155541.04  35483.25   3202500.0  9393524.29
Metrocity          4719958.32  25770.81   2386650.0  7132379.13
Metropol AVM       3166444.16  18603.78   1465800.0  4650847.94
Istinye Park       3050313.20  18369.18   1509900.0  4578582.38
Cevahir AVM        1554414.40   8304.84    819000.0  2381719.24
Zorlu Center       1568818.24   8398.68    803250.0  2380466.92
Viaport Outlet     1530708.08   7636.23    823200.0  2361544.31
Emaar Square Mall  1511803.04   8515.98    834750.0  2355069.02
Forum Istanbul     1572119.12   9090.75    706650.0  2287859.87
```

Check Customer Demographics

```
# Age/Gender distribution for your categories in top malls
top_malls = ['Mall of Istanbul', 'Kanyon']
demographics =
filtered_data[filtered_data['shopping_mall'].isin(top_malls)].groupby(
    ['shopping_mall', 'gender', 'category'])['age'].agg(['mean',
'count']).unstack()

print(demographics)

                              mean                          count
\
category                  Clothing   Souvenir Technology Clothing
Souvenir
shopping_mall    gender

Kanyon           Female  43.243815  44.011041  43.718855     4163
634
                 Male    43.220022  43.186104  44.004963     2677
403
Mall of Istanbul Female  43.669578  42.766610  43.533670     4122
587
                 Male    43.525527  42.326870  42.893617     2801
361


category                Technology
shopping_mall    gender
Kanyon           Female        594
                 Male          403
Mall of Istanbul Female        594
                 Male          423
```

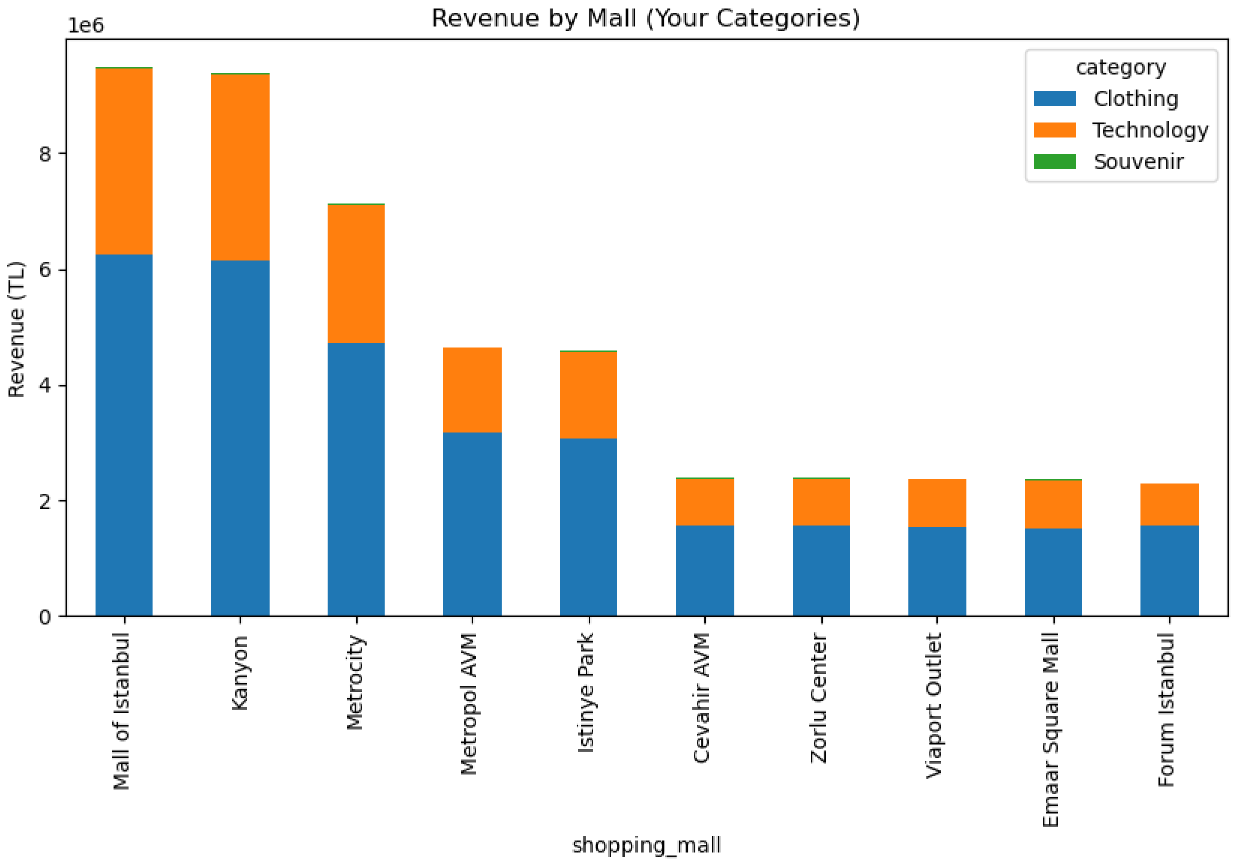
Competition Analysis

```
# Percentage of transactions for your categories in each mall
mall_total_transactions = data['shopping_mall'].value_counts()
your_category_transactions =
filtered_data['shopping_mall'].value_counts()
saturation = (your_category_transactions / mall_total_transactions *
100).sort_values(ascending=False)

print("Market Saturation (% of Your Categories):")
print(saturation.head(3))
```

```
Market Saturation (% of Your Categories):
shopping_mall
Metrocity            45.153554
Forum Istanbul       44.956539
Emaar Square Mall    44.917896
Name: count, dtype: float64
```

Visualization

```
import matplotlib.pyplot as plt

# Plot revenue by mall
revenue_by_mall[['Clothing', 'Technology', 'Souvenir']].plot(
    kind='bar', stacked=True, figsize=(10, 5))
plt.title('Revenue by Mall (Your Categories)')
plt.ylabel('Revenue (TL)')
plt.show()
```



The bar chart visualizes revenue in Turkish Lira (TL) across ten shopping malls in Istanbul, grouped into three categories: Clothing, Technology, and Souvenir. Clothing generates the highest revenue overall, followed by Technology and then Souvenir. Malls such as Mall of Istanbul and Istinye Park stand out with significant contributions to Clothing and Technology revenue, while Souvenir revenue remains relatively low across all malls.