In [ ]:

```
#STEP 1 :INTRODUCTION,IMPORT,LOAD
#This is a Customer sales Dataset obtained from Kaggle website.
#Our dataset contains shopping information from 10 different shopping malls between 2021 and 2023.The data is
#from various age groups and genders to provide a comprehensive view of shopping habits in Istanbul.
```

In [3]:

```
# Necessary imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline
```

In [4]:

```
#Load the Customer data in a DataFrame
df= pd.read_csv('C:/Users/Admin/Downloads/customer_shopping_data.csv')

#Some information about the Advertising data
print('The Customer Shopping Data is of type:', type(df))
print('The Customer Shopping Data has shape:', df.shape)
```

```
The Customer Shopping Data is of type: <class 'pandas.core.frame.DataFrame'>
The Customer Shopping Data has shape: (99457, 10)
```

In [5]:

```
df.info
```

Out[5]:

```
<bound method DataFrame.info of       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   05/08/2022            Kanyon
1         Debit Card   12/12/2021    Forum Istanbul
2               Cash   09/11/2021         Metrocity
3        Credit Card   16/05/2021      Metropol AVM
4               Cash   24/10/2021            Kanyon
...              ...          ...               ...
99452    Credit Card   21/09/2022            Kanyon
99453           Cash   22/09/2021    Forum Istanbul
99454     Debit Card   28/03/2021         Metrocity
99455           Cash   16/03/2021      Istinye Park
99456    Credit Card   15/10/2022  Mall of Istanbul

[99457 rows x 10 columns]>
```

In [6]:

```
#THE FIRST ROWS
df.head(10)
```

Out[6]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	05/08/2022	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	09/11/2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	Kanyon
5	I227836	C657758	Female	28	Clothing	5	1500.40	Credit Card	24/05/2022	Forum Istanbul
6	I121056	C151197	Female	49	Cosmetics	1	40.66	Cash	13/03/2022	Istinye Park
7	I293112	C176086	Female	32	Clothing	2	600.16	Credit Card	13/01/2021	Mall of Istanbul
8	I293455	C159642	Male	69	Clothing	3	900.24	Credit Card	04/11/2021	Metrocity
9	I326945	C283361	Female	60	Clothing	2	600.16	Credit Card	22/08/2021	Kanyon

In [7]:

```
df
```

Out[7]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	05/08/2022	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	09/11/2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	Kanyon
...	...	...	...	...	...	...	...	...	...	...
99452	I219422	C441542	Female	45	Souvenir	5	58.65	Credit Card	21/09/2022	Kanyon
99453	I325143	C569580	Male	27	Food & Beverage	2	10.46	Cash	22/09/2021	Forum Istanbul
99454	I824010	C103292	Male	63	Food & Beverage	2	10.46	Debit Card	28/03/2021	Metrocity
99455	I702964	C800631	Male	56	Technology	4	4200.00	Cash	16/03/2021	Istinye Park
99456	I232867	C273973	Female	36	Souvenir	3	35.19	Credit Card	15/10/2022	Mall of Istanbul

99457 rows × 10 columns

In [8]:

```
#THE LAST ROWS
df.tail(10)
```

Out[8]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
99447	I281214	C288090	Female	37	Toys	3	107.52	Cash	21/02/2021	Metropol AVM
99448	I332105	C231387	Female	65	Shoes	4	2400.68	Credit Card	29/08/2021	Metropol AVM
99449	I134399	C953724	Male	65	Clothing	1	300.08	Cash	01/01/2023	Kanyon
99450	I170504	C226974	Female	28	Books	1	15.15	Cash	28/02/2023	Zorlu Center
99451	I675411	C513603	Male	50	Toys	5	179.20	Cash	09/10/2021	Metropol AVM
99452	I219422	C441542	Female	45	Souvenir	5	58.65	Credit Card	21/09/2022	Kanyon
99453	I325143	C569580	Male	27	Food & Beverage	2	10.46	Cash	22/09/2021	Forum Istanbul
99454	I824010	C103292	Male	63	Food & Beverage	2	10.46	Debit Card	28/03/2021	Metrocity
99455	I702964	C800631	Male	56	Technology	4	4200.00	Cash	16/03/2021	Istinye Park
99456	I232867	C273973	Female	36	Souvenir	3	35.19	Credit Card	15/10/2022	Mall of Istanbul

In [9]:

```
#STEP2: DATA CLEANING
# Check for missing values in the DataFrame.
print(df.isnull().sum())
```

```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
dtype: int64
```

In [10]:

```
## Remove duplicates if any.
df.drop_duplicates(inplace=True)
```

In [11]:

```
#Check if any column contains a NaN.
df.isnull().any()
#There are no NaN values
```

Out[11]:

```
invoice_no        False
customer_id       False
gender            False
age               False
category          False
quantity          False
price             False
payment_method    False
invoice_date      False
shopping_mall     False
dtype: bool
```

In [12]:

```
#STEP3:Exploratory data analysis
#descriptive statistics on each column of the DataFrame
df.describe()
```

Out[12]:

	age	quantity	price
count	99457.000000	99457.000000	99457.000000
mean	43.427089	3.003429	689.256321
std	14.990054	1.413025	941.184567
min	18.000000	1.000000	5.230000
25%	30.000000	2.000000	45.450000
50%	43.000000	3.000000	203.300000
75%	56.000000	4.000000	1200.320000
max	69.000000	5.000000	5250.000000

In [60]:

```
#categorical columns
print(df['category'].value_counts())
```

```
Clothing           34487
Cosmetics          15097
Food & Beverage    14776
Toys               10087
Shoes              10034
Souvenir            4999
Technology          4996
Books               4981
Name: category, dtype: int64
43.42708909377922
5250.0
```

In [61]:

```
# numerical columns
print(df['age'].mean())
print(df['price'].max())
```

```
43.42708909377922
5250.0
```

In [23]:

```
#remove irrelevant columns
df.drop('invoice_no', axis=1, inplace=True)
```

In [24]:

```
df.corr()
#A correlation value of 1 tells us there is a high correlation and a correlation of 0 tells us that the data
#is not correlated at all.
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_20172\3987966156.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it w
ill default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

Out[24]:

	age	quantity	price
age	1.000000	0.000667	0.001694
quantity	0.000667	1.000000	0.344880
price	0.001694	0.344880	1.000000

In [40]:

```
df.groupby("shopping_mall")["price"].sum()
#calculating the how much money is spent at each shopping mall. We have to group the data by using the groupby()method
#and then add up all the prices at each shopping mall
#display the total amount of money spent in each shopping mall
```

Out[40]:

```
shopping_mall
Cevahir AVM           3433671.84
Emaar Square Mall     3390408.31
Forum Istanbul        3336073.82
Istinye Park          6717077.54
Kanyon               13710755.24
Mall of Istanbul     13851737.62
Metrocity            10249980.07
Metropol AVM          6937992.99
Viaport Outlet        3414019.46
Zorlu Center          3509649.02
Name: price, dtype: float64
```

In [45]:

```
#Group the payment methods by age,quantity and price
#To find out how much money was paid by a specific payment method and how much quantity was bought by the payment method
df.groupby("payment_method").sum()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_20172\1040488012.py:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future versio
n, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  df.groupby("payment_method").sum()
```

Out[45]:

	age	quantity	price
payment_method			
Cash	1931552	133370	30705030.98
Credit Card	1516980	105045	24051476.93
Debit Card	870596	60297	13794858.00

In [47]:

```
df.groupby("category").sum()
#To find out the quantity of each category was bought and how much money
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_20172\3215413246.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future versio
n, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  df.groupby("category").sum()
```

Out[47]:

	age	quantity	price
category			
Books	216882	14982	226977.30
Clothing	1497054	103558	31075684.64
Cosmetics	657937	45465	1848606.90
Food & Beverage	640605	44277	231568.71
Shoes	436027	30217	18135336.89
Souvenir	216922	14871	174436.83
Technology	216669	15021	15772050.00
Toys	437032	30321	1086704.64

In [53]:

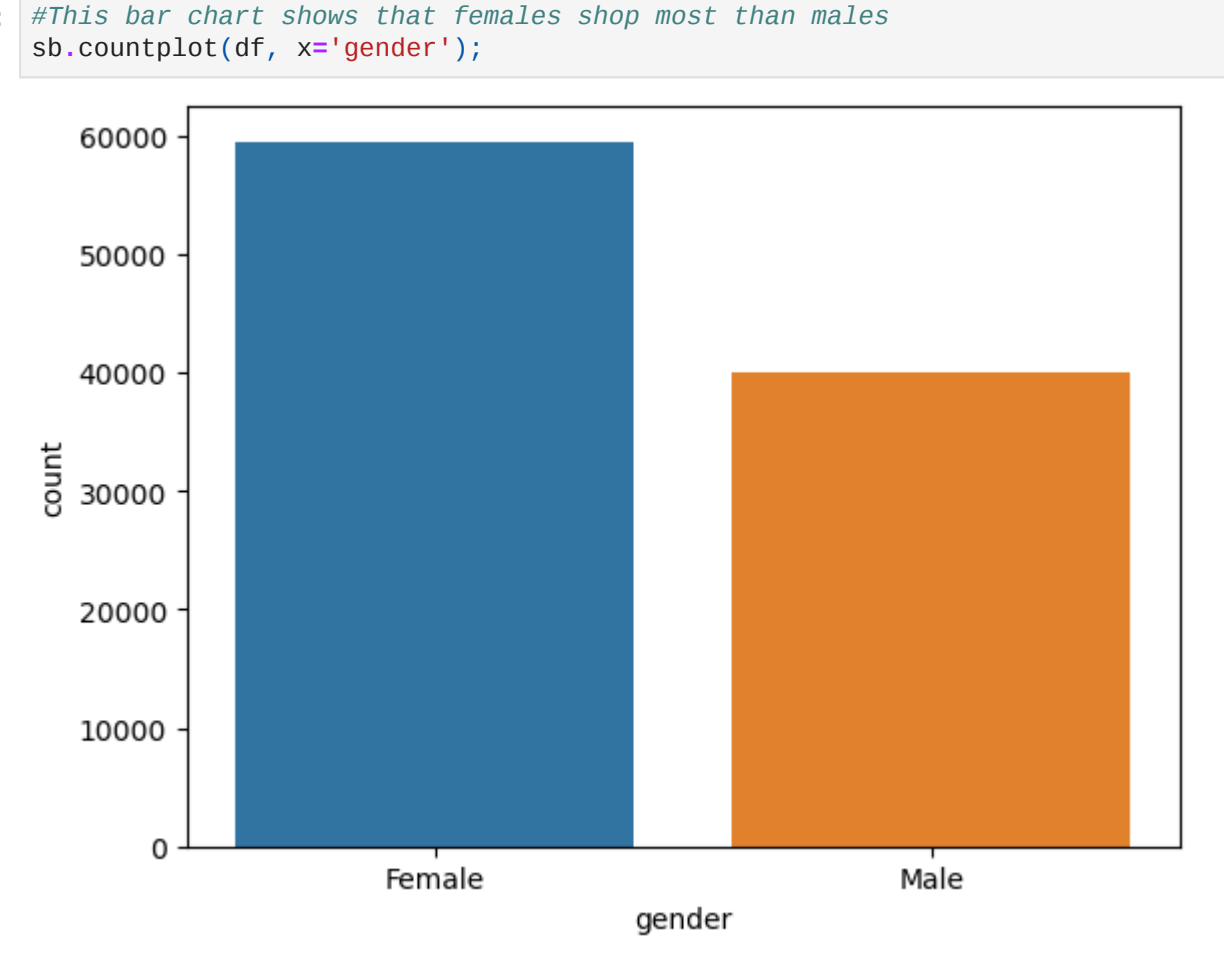
```
#DATA VISUALIZATION
gender=df['gender'].describe()
gender
```

Out[53]:

```
count      99457
unique         2
top       Female
freq       59482
Name: gender, dtype: object
```

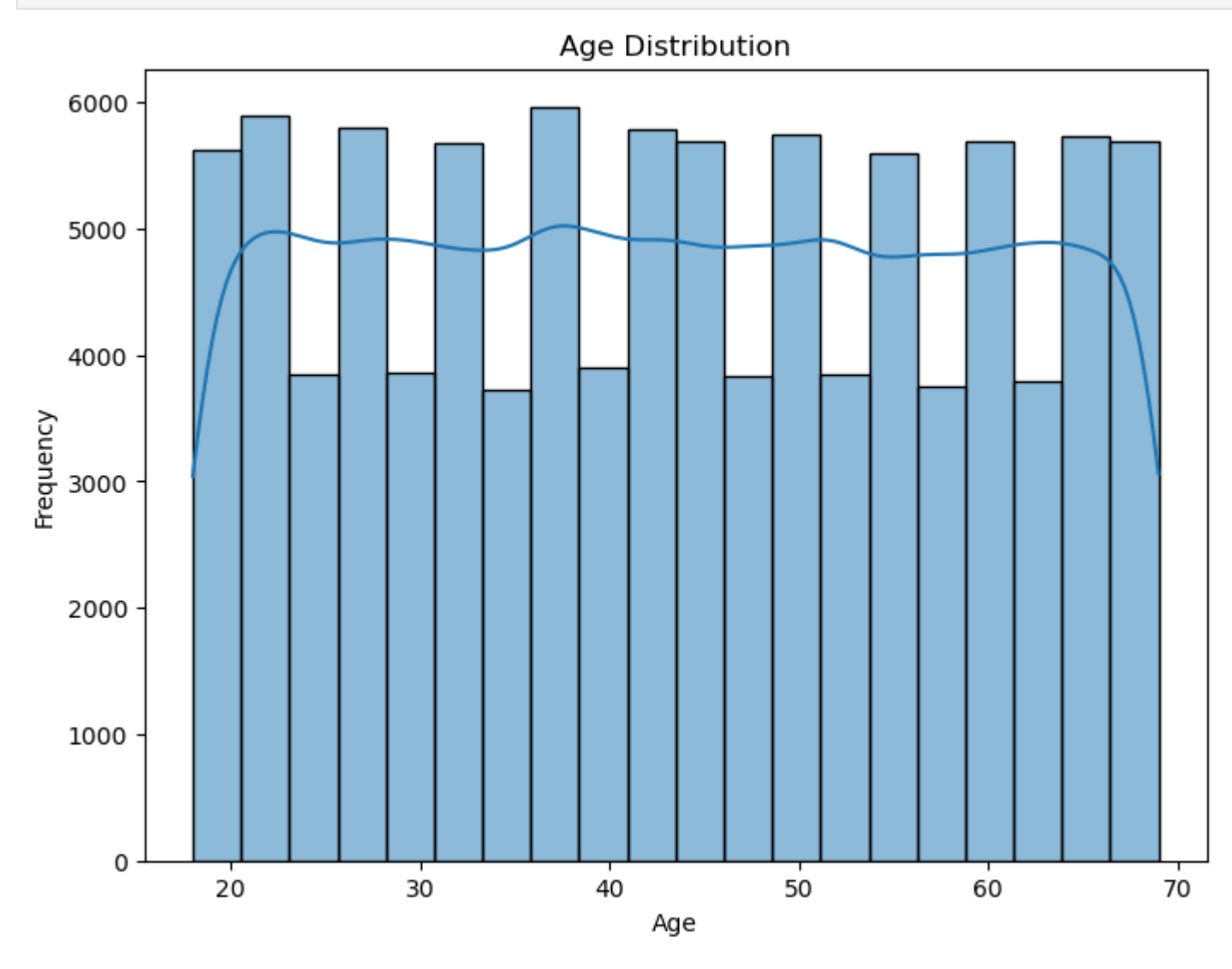
In [58]:

```
#This bar chart shows that females shop most than males
sb.countplot(df, x='gender');
```



In [83]:

```
# To find out the age distribution
#What age group shops the most
plt.figure(figsize=(8, 6))
sb.histplot(df['age'], bins=20, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



In [84]:

```
# Convert the 'invoice_date' column to datetime data type
df['invoice_date'] = pd.to_datetime(df['invoice_date'])
```

In [85]:

```
df.set_index('invoice_date', inplace=True)
```

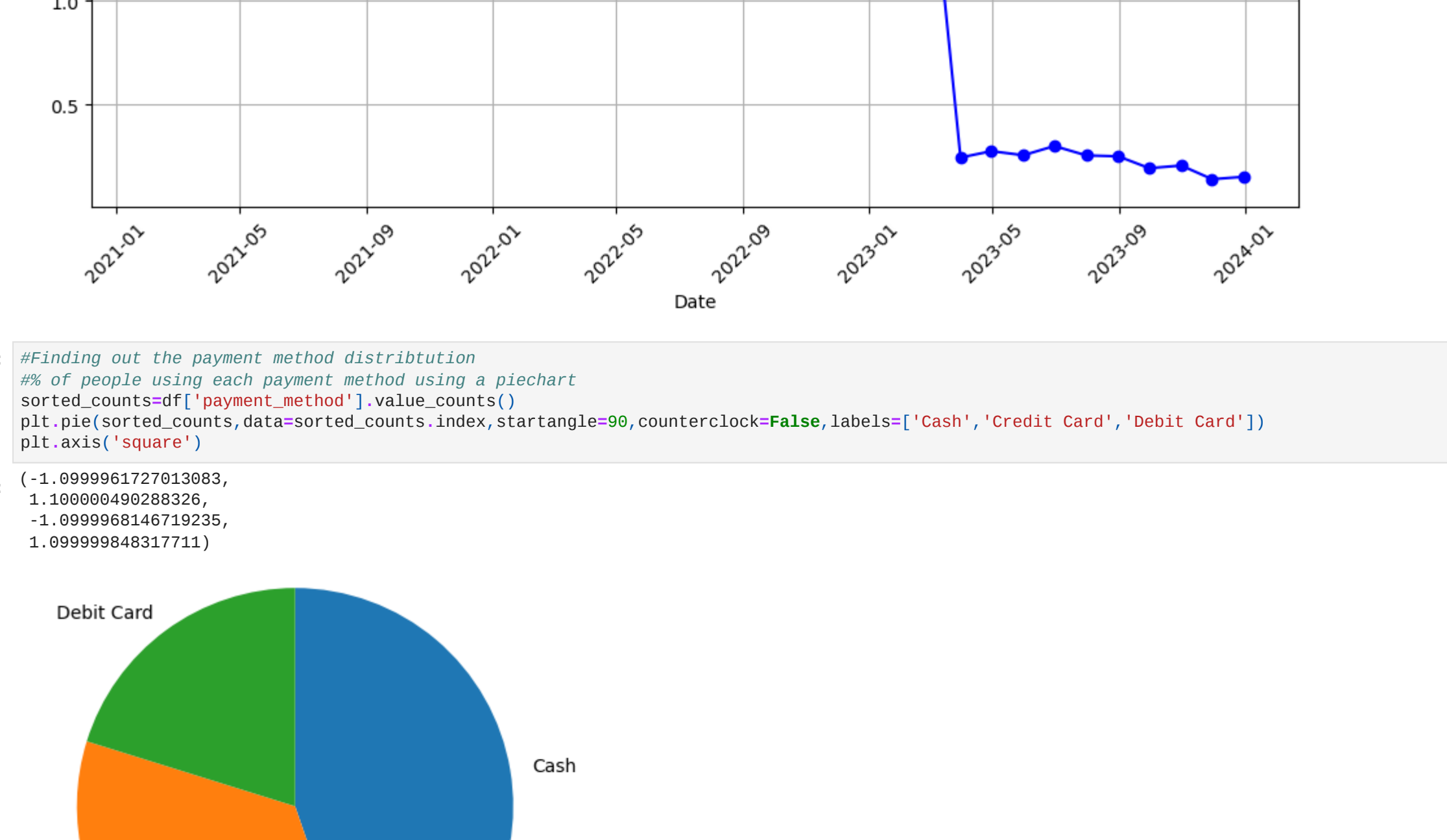
In [90]:

```
sales_trends = df.resample('M').sum()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_20172\3446318586.py:1: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future versio
n, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
  sales_trends = df.resample('M').sum()
```

In [91]:

```
#Checking the sales trend over time using the price column and invoice date column
#Converting the invoice date to months
plt.figure(figsize=(12, 6))
plt.plot(sales_trends.index, sales_trends['price'], marker='o', linestyle='-', color='b')
plt.title('Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

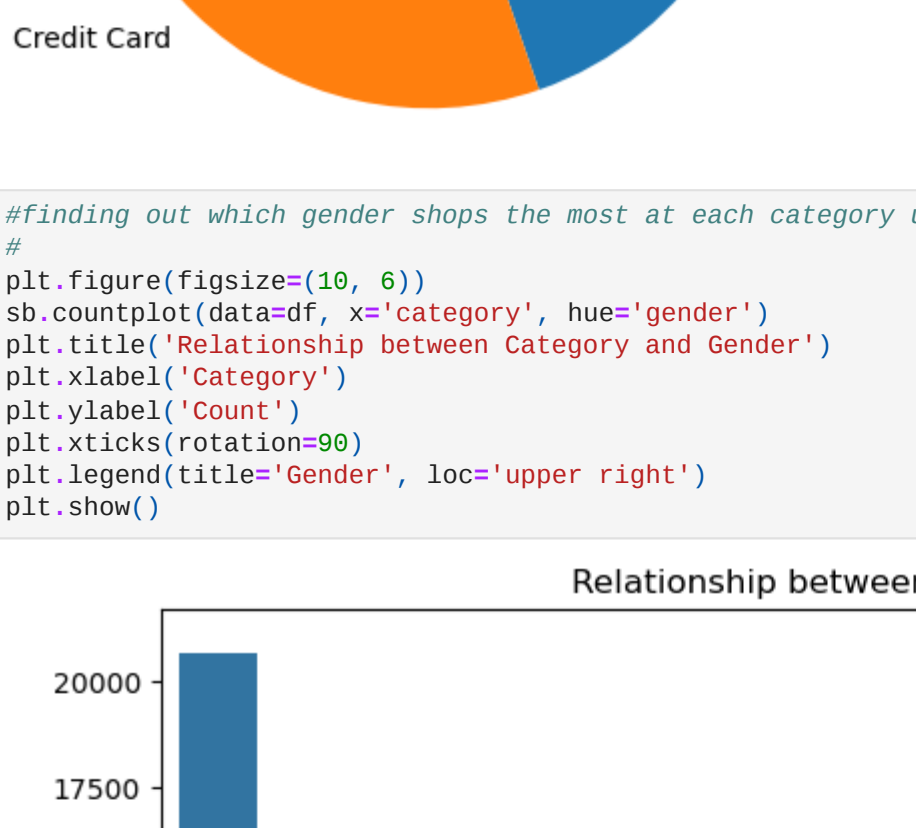


In [93]:

```
#Finding out the payment method distribtution
#% of people using each payment method using a piechart
sorted_counts=df['payment_method'].value_counts()
plt.pie(sorted_counts,data=sorted_counts.index,startangle=90,counterclock=False,labels=['Cash','Credit Card','Debit Card'])
plt.axis('square')
```

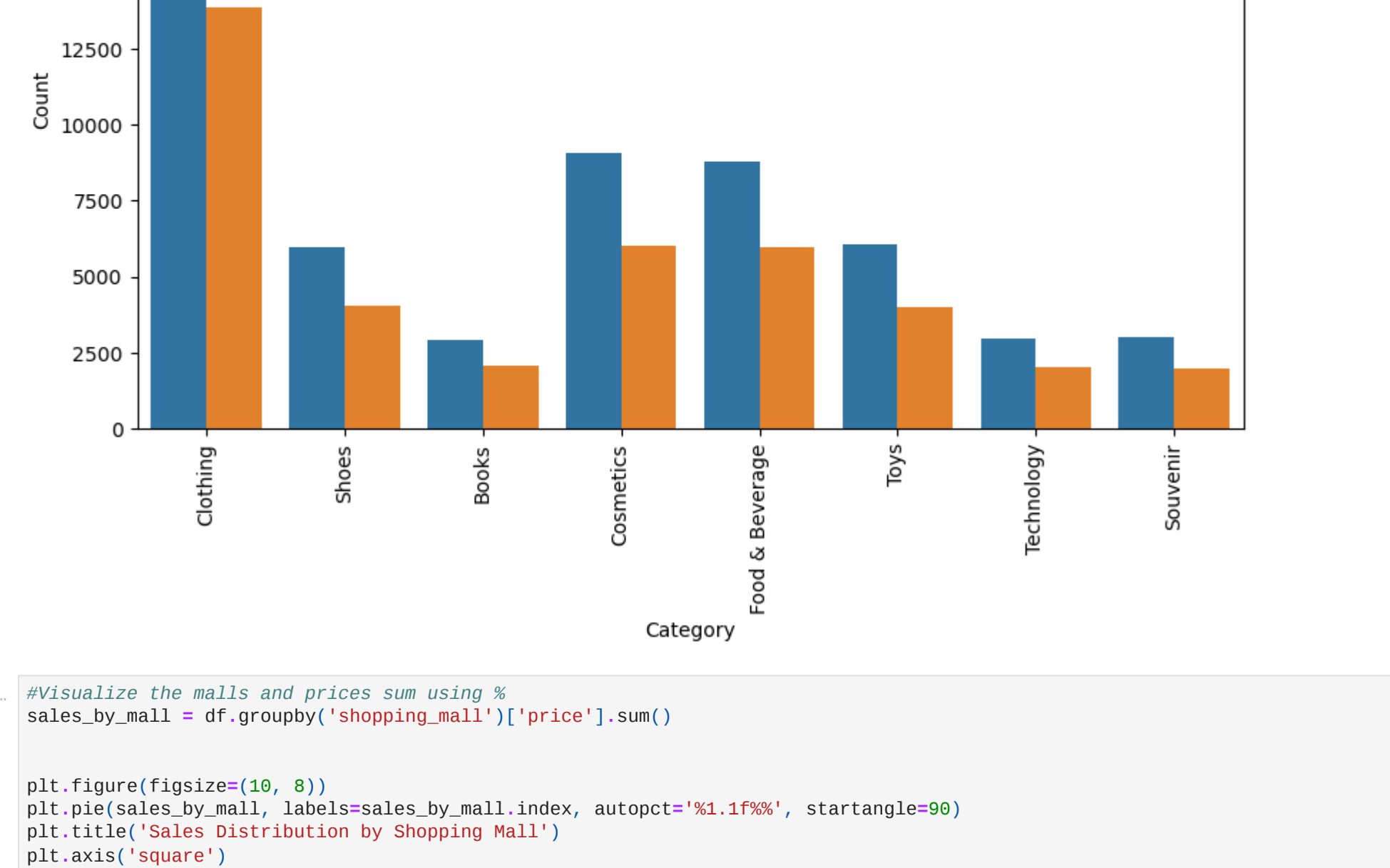
Out[93]:

```
(-1.0999961727013083,
 1.100000490288326,
 -1.0999968146719235,
 1.099999848317711)
```



In [96]:

```
#finding out which gender shops the most at each category using a barchart
#
plt.figure(figsize=(10, 6))
sb.countplot(data=df, x='category', hue='gender')
plt.title('Relationship between Category and Gender')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.legend(title='Gender', loc='upper right')
plt.show()
```



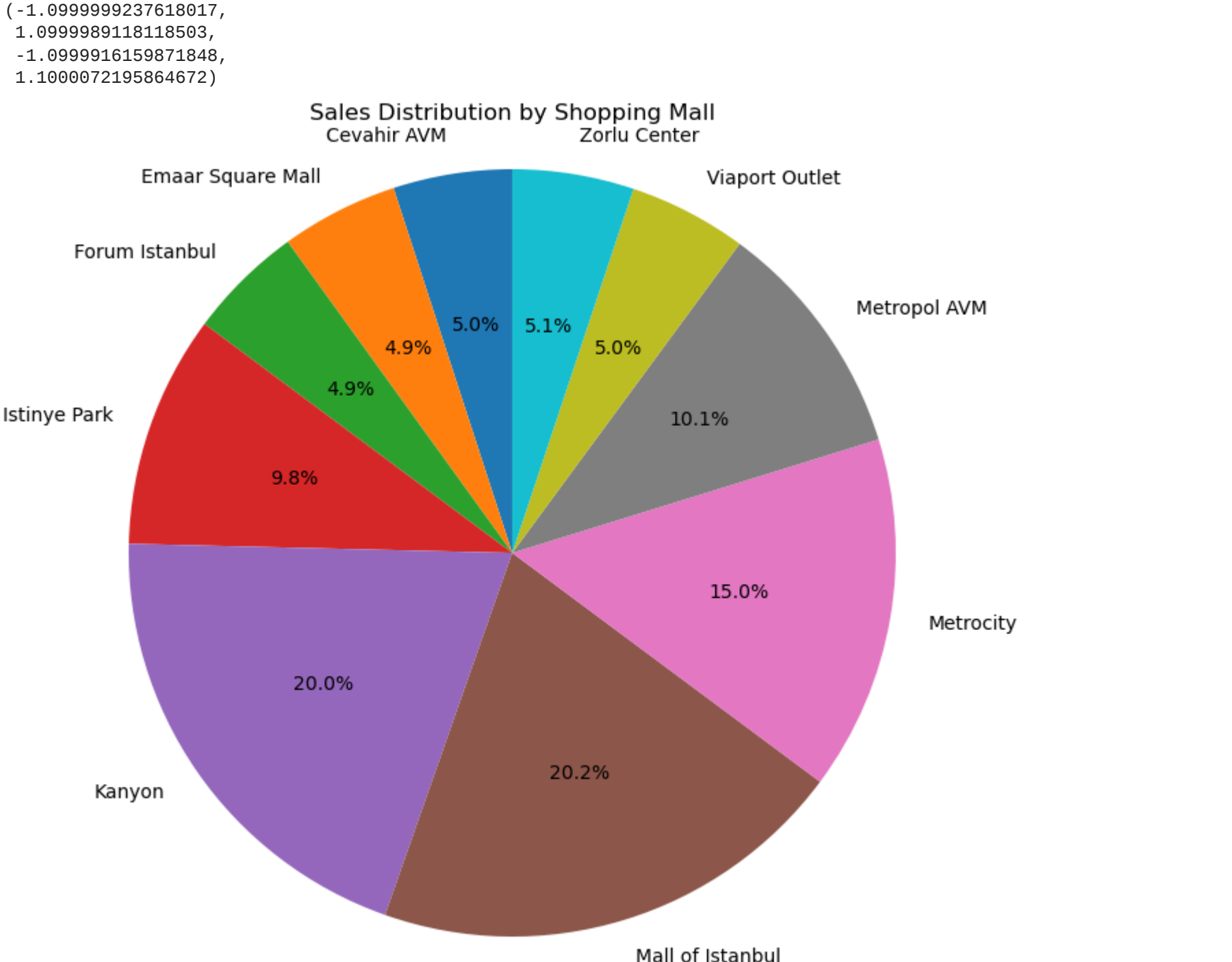
In [100…

```
#Visualize the malls and prices sum using %
sales_by_mall = df.groupby('shopping_mall')['price'].sum()


plt.figure(figsize=(10, 8))
plt.pie(sales_by_mall, labels=sales_by_mall.index, autopct='%1.1f%%', startangle=90)
plt.title('Sales Distribution by Shopping Mall')
plt.axis('square')
```

Out[100]:

```
(-1.0999999237618017,
 1.0999989118118503,
 -1.0999916159871848,
 1.1000072195864672)
```



In [ ]:

In [ ]:

In [ ]: