

problem-1-2

March 17, 2024

Importing Libraries

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib
```

Loading dataset

```
[ ]: from datetime import datetime

custom_date_parser = lambda x: datetime.strptime(x, "%d/%m/%Y")

[ ]: df = pd.read_csv('/content/customer_shopping_data (1).
↳csv', parse_dates=(['invoice_date']), date_parser=custom_date_parser)

[ ]: df.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
```

Info about dataset

```
[ ]: df.dtypes

[ ]: invoice_no          object
customer_id            object
```

```

gender          object
age             int64
category        object
quantity        int64
price           float64
payment_method  object
invoice_date     datetime64[ns]
shopping_mall    object
dtype: object

```

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

```

<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

```

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

```

[ ]:
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

```

Dimension of the dataset

```
[ ]: df.shape
```

```
[ ]: (99457, 10)
```

Checking if there is any duplicate values

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

```
[ ]: False      99457  
      dtype: int64
```

No duplicate values are found

Checking for missing values

```
[ ]: 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
```

There are no missing values

#Data Cleaning and Preprocessing

We have invoice_date, we can extract month and year from this to get insight

```
[ ]: df['month'] = df['invoice_date'].dt.month  
      df['year'] = df['invoice_date'].dt.year
```

```
[ ]: df.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  month  year  
0    Credit Card    2022-08-05      Kanyon         8  2022  
1    Debit Card    2021-12-12  Forum Istanbul    12  2021  
2         Cash    2021-11-09      Metrocity    11  2021  
3    Credit Card    2021-05-16  Metropol AVM      5  2021  
4         Cash    2021-10-24      Kanyon        10  2021
```

Calculating Total amount for each transaction

```
[ ]: df['total_amount'] = df['quantity'] * df['price']
```

```
[ ]: df.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 month year total_amount
0 Credit Card 2022-08-05 Kanyon 8 2022 7502.00
1 Debit Card 2021-12-12 Forum Istanbul 12 2021 5401.53
2 Cash 2021-11-09 Metrocity 11 2021 300.08
3 Credit Card 2021-05-16 Metropol AVM 5 2021 15004.25
4 Cash 2021-10-24 Kanyon 10 2021 242.40
```

#Exploratory Data Analysis (EDA)

Gender Distribution among customers

```
[ ]: df_gender = df['gender'].value_counts()

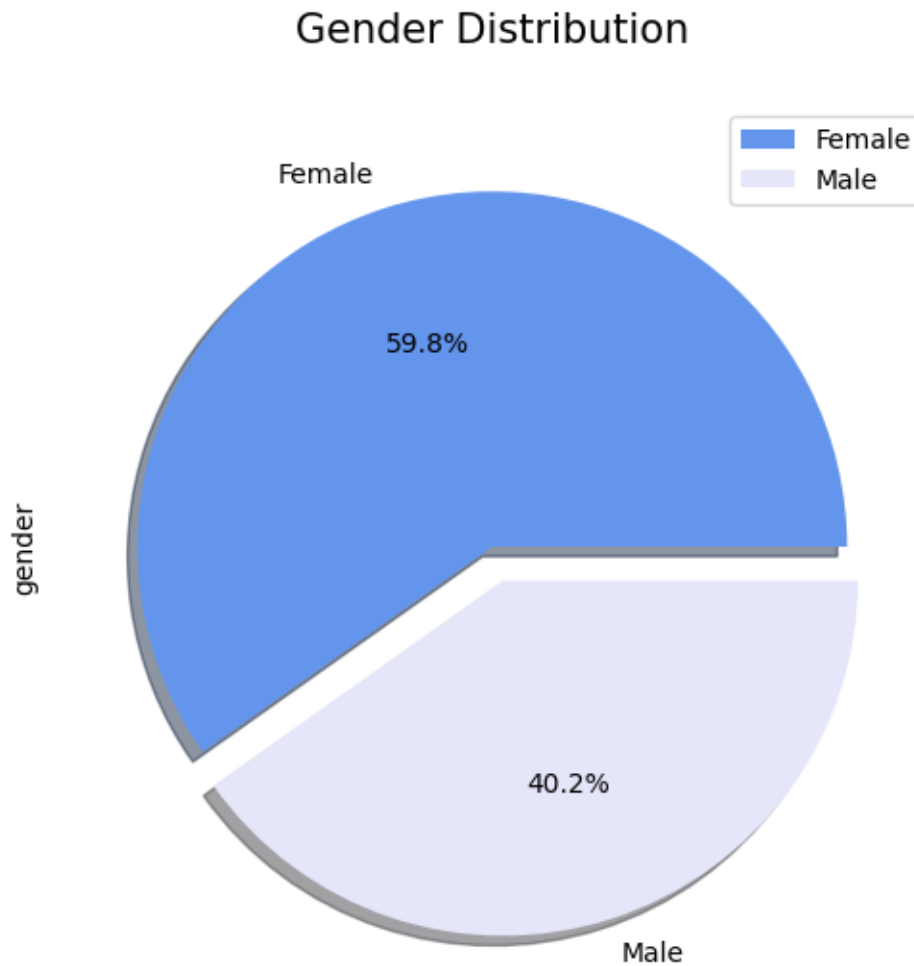
df_gender
```

```
[ ]: Female    59482
      Male     39975
      Name: gender, dtype: int64
```

```
[ ]: plt.figure(figsize=(8,6))
      colors = ['cornflowerblue','lavender']

      df_gender.plot(kind = 'pie', autopct = '%0.1f%%', explode = [0, 0.1], shadow =_
      ↪True, colors = colors, legend = type)
      plt.title('Gender Distribution\n', fontsize = 15)

      plt.show()
```



Number of unique categories of products

```
[ ]: df['category'].nunique()
```

```
[ ]: 8
```

There are 8 different categories of product

Distribution of Transactions across different categories

```
[ ]: df_category = df['category'].value_counts()
```

```
df_category
```

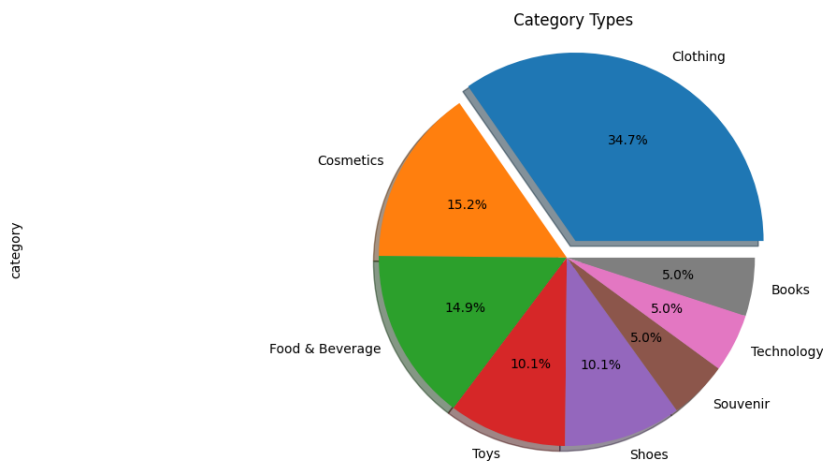
```
[ ]: Clothing      34487
     Cosmetics    15097
     Food & Beverage 14776
```

Toys	10087
Shoes	10034
Souvenir	4999
Technology	4996
Books	4981

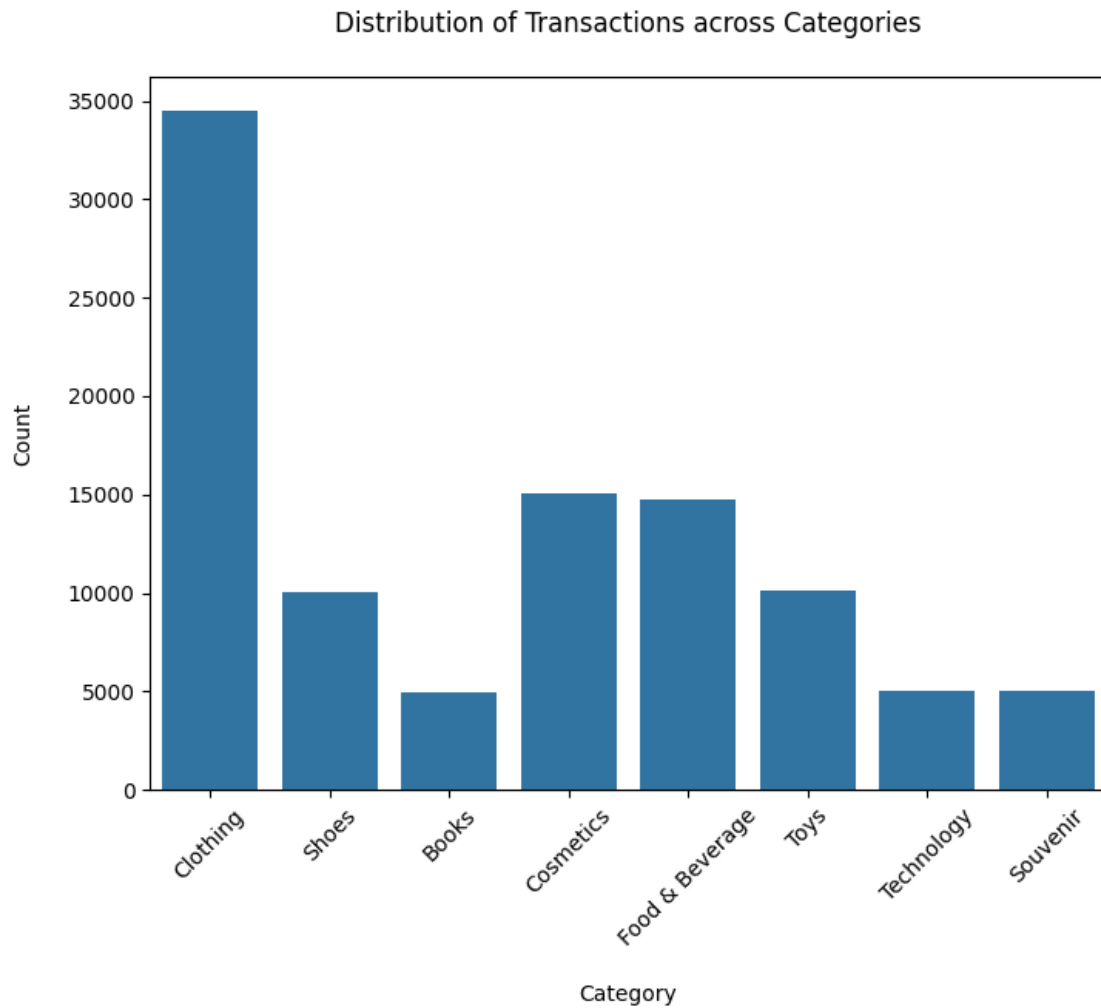
Name: category, dtype: int64

```
[ ]: label=['Clothing','Cosmetics','Food & Beverage',
           'Toys','Shoes','Souvenir','Technology','Books']
df_category.plot(kind='pie',figsize=(15, 6),autopct='%1.1f%%',shadow=True,labels=label,explode=[0.1,0,0,0,0,0,0,0] )
plt.title(' Category Types')
plt.axis('equal')

plt.show()
```



```
[ ]: plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='category')
plt.xlabel('\nCategory')
plt.ylabel('Count\n')
plt.title('Distribution of Transactions across Categories\n')
plt.xticks(rotation=45)
plt.show()
```



Category Ordered Distribution between Gender

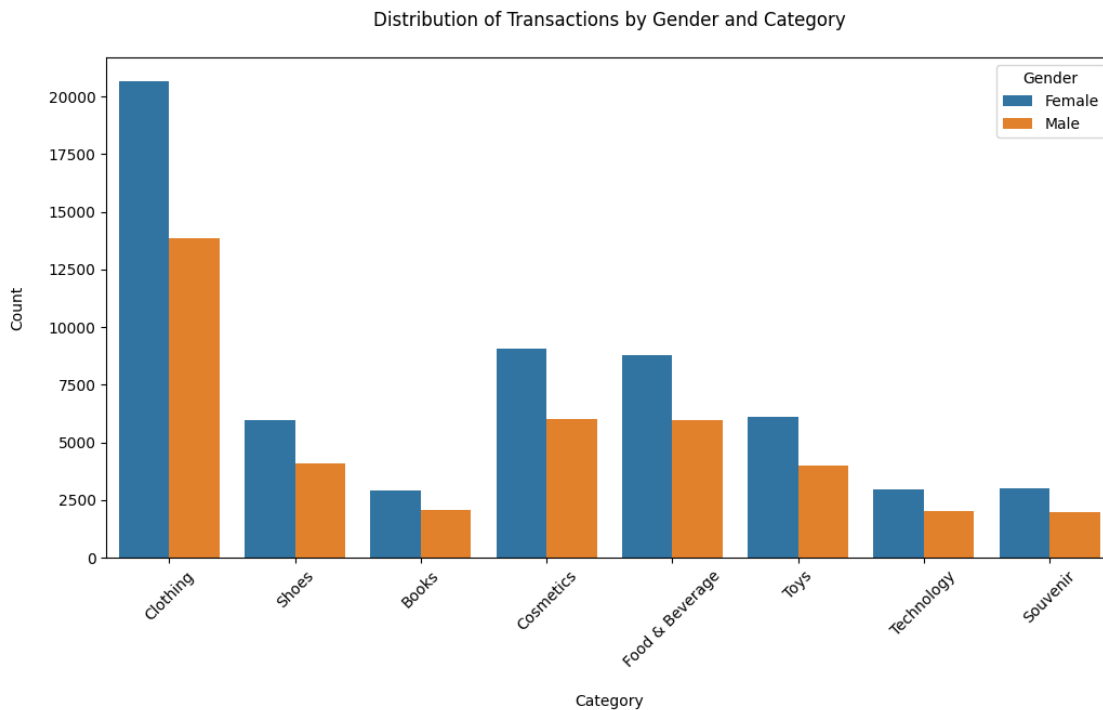
```
[ ]: df_gender_order_category = df.groupby(['gender', 'category']).size().
      ↪reset_index(name = 'count')
```

```
df_gender_order_category
```

```
[ ]:
   gender  category  count
0  Female      Books   2906
1  Female    Clothing 20652
2  Female    Cosmetics  9070
3  Female  Food & Beverage  8804
4  Female      Shoes   5967
5  Female    Souvenir  3017
6  Female    Technology  2981
7  Female      Toys   6085
```

8	Male	Books	2075
9	Male	Clothing	13835
10	Male	Cosmetics	6027
11	Male	Food & Beverage	5972
12	Male	Shoes	4067
13	Male	Souvenir	1982
14	Male	Technology	2015
15	Male	Toys	4002

```
[ ]: plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='category', hue='gender')
plt.title('Distribution of Transactions by Gender and Category\n')
plt.xlabel('\nCategory')
plt.ylabel('Count\n')
plt.xticks(rotation=45)
plt.legend(title='Gender')
plt.show()
```



```
[ ]: df.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	month	year	total_amount
0	Credit Card	2022-08-05	Kanyon	8	2022	7502.00
1	Debit Card	2021-12-12	Forum Istanbul	12	2021	5401.53
2	Cash	2021-11-09	Metrocity	11	2021	300.08
3	Credit Card	2021-05-16	Metropol AVM	5	2021	15004.25
4	Cash	2021-10-24	Kanyon	10	2021	242.40

Number of unique ages in the dataset

```
[ ]: df['age'].nunique()
```

```
[ ]: 52
```

Number of unique quantities in the dataset

```
[ ]: df['quantity'].nunique()
```

```
[ ]: 5
```

Quantity Distribution

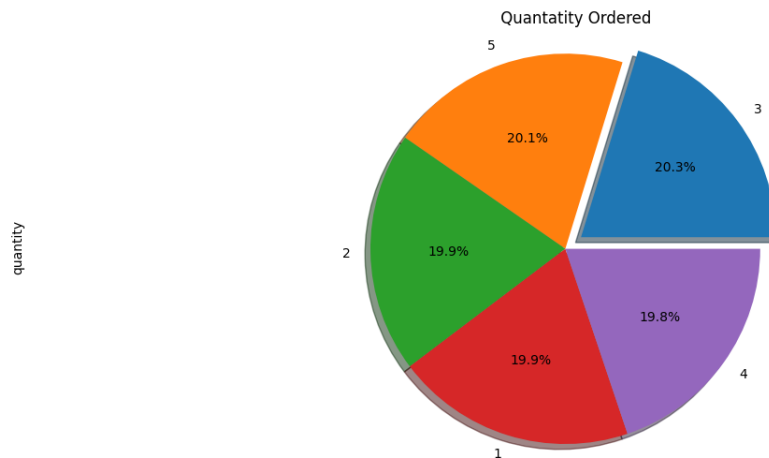
```
[ ]: df_quantity = df['quantity'].value_counts()

df_quantity
```

```
[ ]: 3    20149
     5    19990
     2    19828
     1    19767
     4    19723
     Name: quantity, dtype: int64
```

```
[ ]: label=['3','5','2','1','4']
     df_quantity.plot(kind='pie',figsize=(15, 6),autopct='%1.
     ↪1f%',shadow=True,labels=label,explode=[0.1,0,0,0,0] )
     plt.title(' Quantatity Ordered')
     plt.axis('equal')

     plt.show()
```



Number of unique payment methods

```
[ ]: df['payment_method'].nunique()
```

```
[ ]: 3
```

Payment method distribution

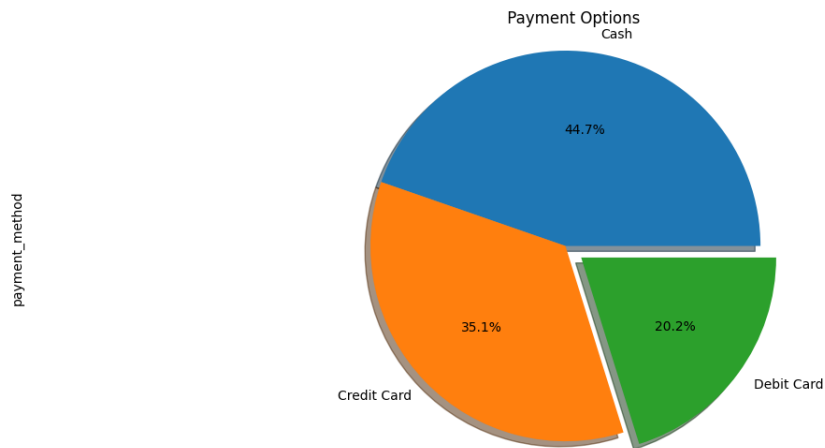
```
[ ]: df_payment = df['payment_method'].value_counts()
```

```
df_payment
```

```
[ ]: Cash          44447
      Credit Card   34931
      Debit Card    20079
      Name: payment_method, dtype: int64
```

```
[ ]: label=["Cash","Credit Card","Debit Card"]
      df_payment.plot(kind='pie',figsize=(15, 6),autopct='%1.
      ↪1f%%',shadow=True,labels=label,explode=[0,0,0.1] )
      plt.title(' Payment Options ')
      plt.axis('equal')

      plt.show()
```



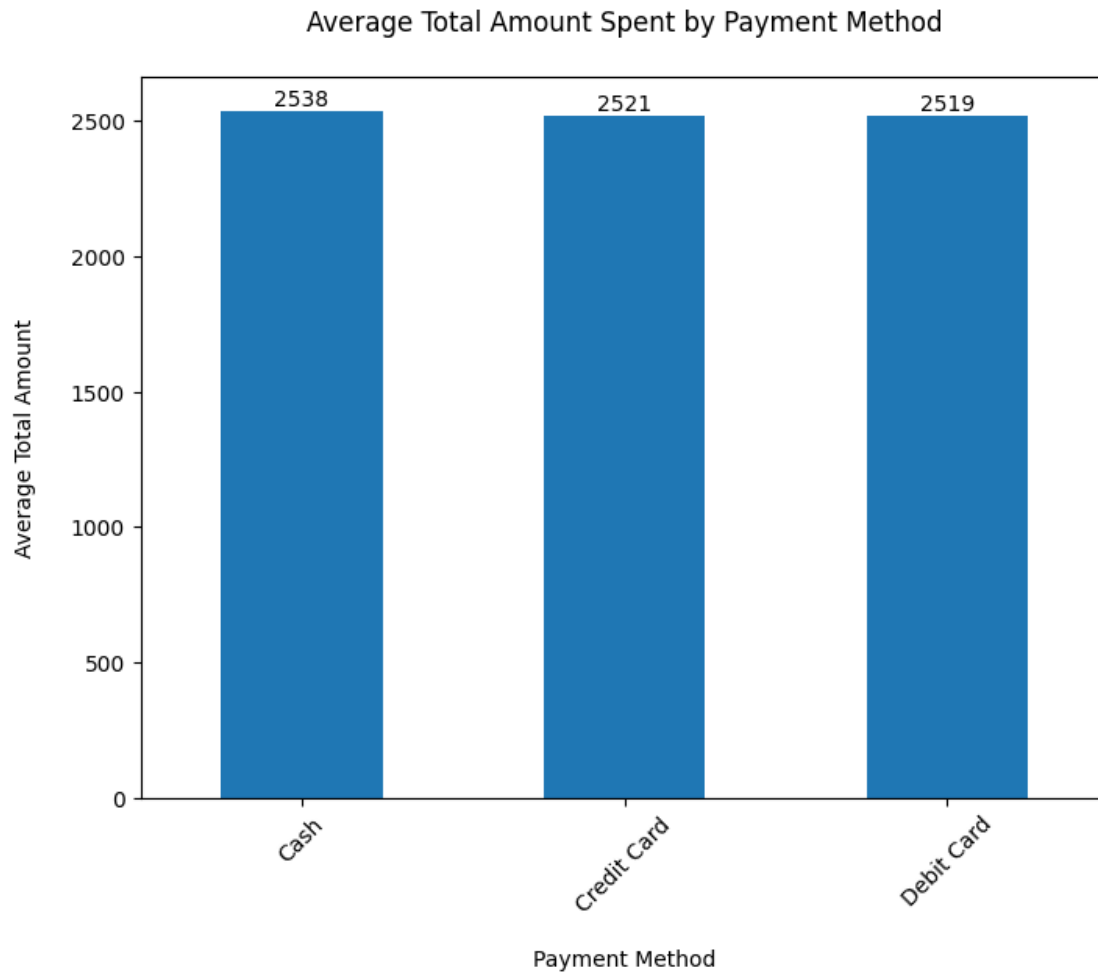
Average total amount spent by the payment method

```
[ ]: payment_method_avg = df.groupby('payment_method')['total_amount'].mean().
    ↪sort_values(ascending=False)

payment_method_avg
```

```
[ ]: payment_method
      Cash      2538.579500
Credit Card  2521.460129
Debit Card   2519.867895
Name: total_amount, dtype: float64
```

```
[ ]: plt.figure(figsize=(8, 6))
x= payment_method_avg.plot(kind='bar')
for i in x.patches:
    x.annotate(int(i.get_height()),(i.get_x()+0.25,i.
    ↪get_height()+1),ha="center",va="bottom")
    x.tick_params(axis="both",labelsize=10)
plt.title('Average Total Amount Spent by Payment Method\n')
plt.xlabel('\nPayment Method')
plt.ylabel('Average Total Amount\n')
plt.xticks(rotation=45)
plt.show()
```



Gender and payment method distribution

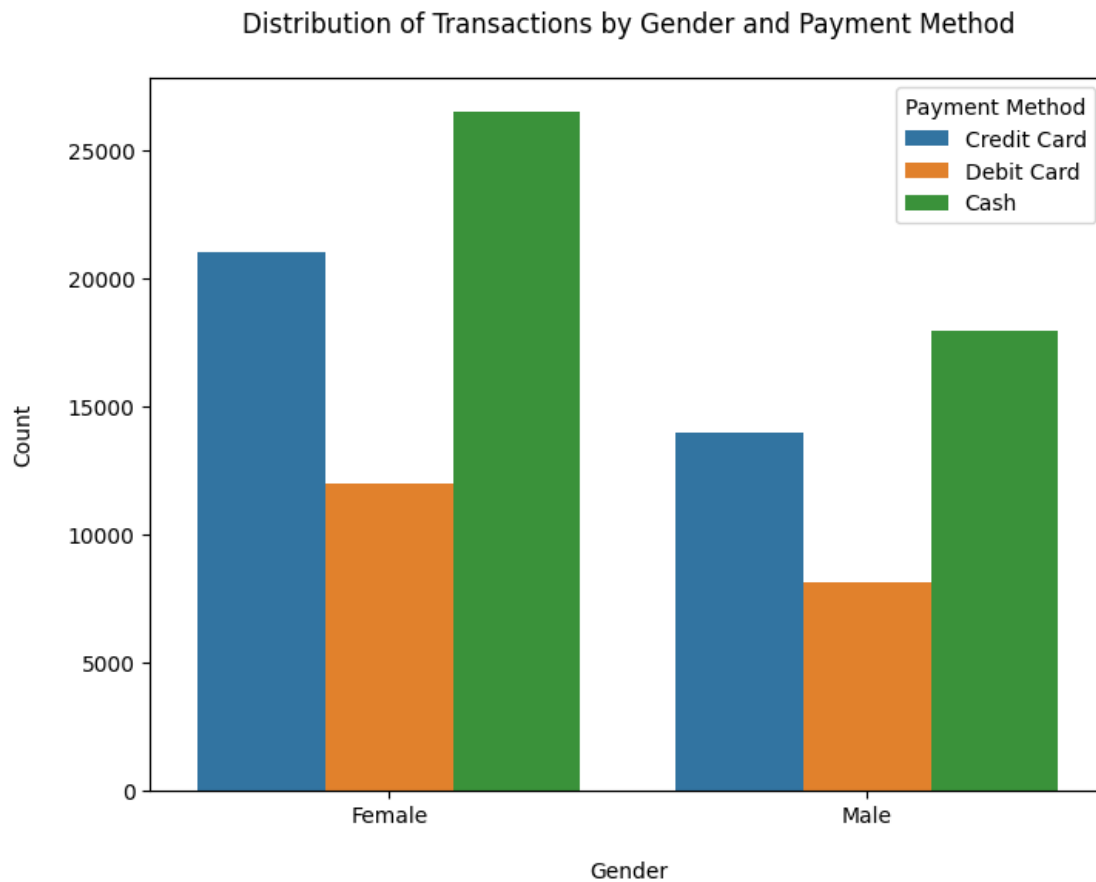
```
[ ]: df_gender_paym = df.groupby('gender')['payment_method'].value_counts().
      ↪ rename('count').reset_index()
```

```
df_gender_paym
```

```
[ ]:   gender payment_method  count
0  Female           Cash  26509
1  Female   Credit Card  21011
2  Female   Debit Card  11962
3    Male           Cash  17938
4    Male   Credit Card  13920
5    Male   Debit Card   8117
```

```
[ ]: plt.figure(figsize=(8, 6))
      sns.countplot(data=df, x='gender', hue='payment_method')
```

```
plt.title('Distribution of Transactions by Gender and Payment Method\n')
plt.xlabel('\nGender')
plt.ylabel('Count\n')
plt.legend(title='Payment Method')
plt.show()
```



Number of unique shopping malls

```
[ ]: df['shopping_mall'].nunique()
```

```
[ ]: 10
```

Shopping mall distribution

Distribution of transaction in each shopping mall

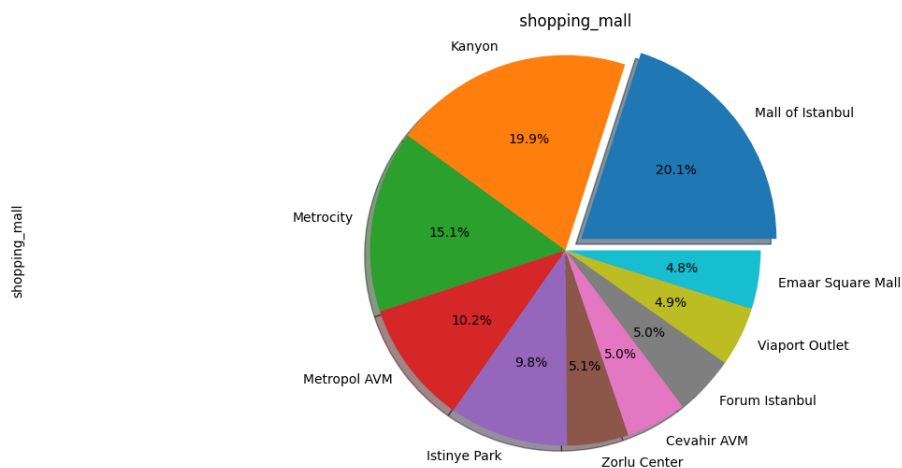
```
[ ]: df_shopping = df['shopping_mall'].value_counts()

df_shopping
```

```
[ ]: Mall of Istanbul      19943
      Kanyon               19823
      Metrocity            15011
      Metropol AVM         10161
      Istinye Park         9781
      Zorlu Center         5075
      Cevahir AVM          4991
      Forum Istanbul       4947
      Viaport Outlet        4914
      Emaar Square Mall    4811
      Name: shopping_mall, dtype: int64
```

```
[ ]: label=['Mall of Istanbul','Kanyon','Metrocity','Metropol AVM','Istinye_
      ↪Park','Zorlu Center','Cevahir AVM','Forum Istanbul','Viaport Outlet','Emaar_
      ↪Square Mall']
df_shopping.plot(kind='pie',figsize=(15, 6),autopct='%1.
      ↪1f%%',shadow=True,labels=label,explode=[0.1,0,0,0,0,0,0,0,0,0] )
plt.title(' shopping_mall')
plt.axis('equal')

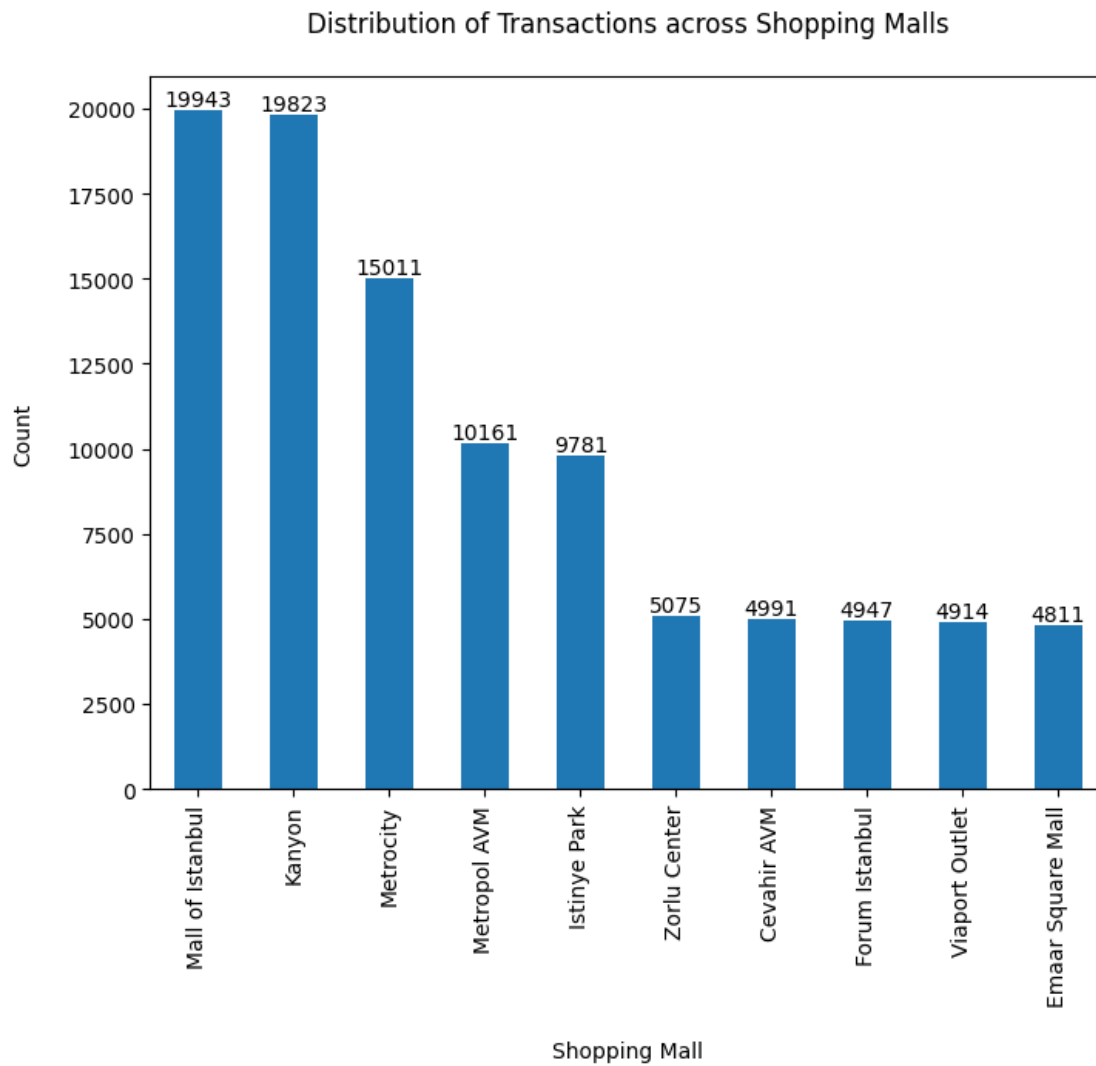
plt.show()
```



```
[ ]: plt.figure(figsize=(8,6))
      x=df_shopping.plot(kind="bar")

      for i in x.patches:
          x.annotate(int(i.get_height()),(i.get_x()+0.25,i.
      ↪get_height()+1),ha="center",va="bottom")
          x.tick_params(axis="both",labelsize=10)
      plt.title('Distribution of Transactions across Shopping Malls\n')
      plt.xlabel('\nShopping Mall')
```

```
plt.ylabel('Count\n')
plt.show()
```



Average total amount spent across different shopping malls

```
[ ]: mall_avg_total_amount = df.groupby('shopping_mall')['total_amount'].mean().
    ↪sort_values(ascending=False)

mall_avg_total_amount
```

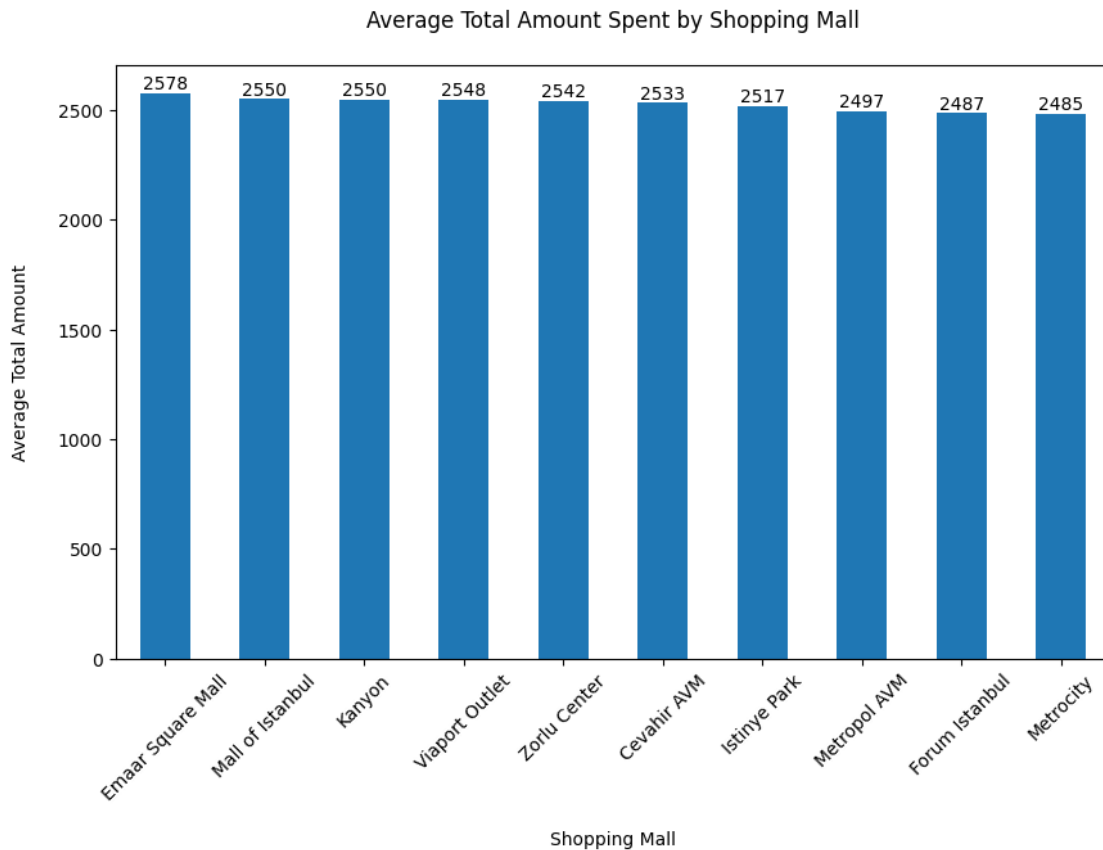
```
[ ]: shopping_mall
Emaar Square Mall    2578.694718
Mall of Istanbul    2550.894132
Kanyon              2550.281547
```

Viaport Outlet	2548.095181
Zorlu Center	2542.079570
Cevahir AVM	2533.588099
Istinye Park	2517.005181
Metropol AVM	2497.777108
Forum Istanbul	2487.148017
Metrocity	2485.030133

Name: total_amount, dtype: float64

```
[ ]: plt.figure(figsize=(10, 6))
x = mall_avg_total_amount.plot(kind='bar')
for i in x.patches:
    x.annotate(int(i.get_height()),(i.get_x()+0.25,i.
    get_height()+1),ha="center",va="bottom")
    x.tick_params(axis="both",labelsize=10)

plt.xlabel('\nShopping Mall')
plt.ylabel('Average Total Amount\n')
plt.title('Average Total Amount Spent by Shopping Mall\n')
plt.xticks(rotation=45)
plt.show()
```



Price Statistics

```
[ ]: print("The Max Price Is:", df['price'].max())
      print("The Min Price Is:", df['price'].min())
      print("The Mean Price Is:", df['price'].mean())
      print("The Median Price Is:", df['price'].median())
```

The Max Price Is: 5250.0

The Min Price Is: 5.23

The Mean Price Is: 689.2563209226097

The Median Price Is: 203.3

Max Price for each category

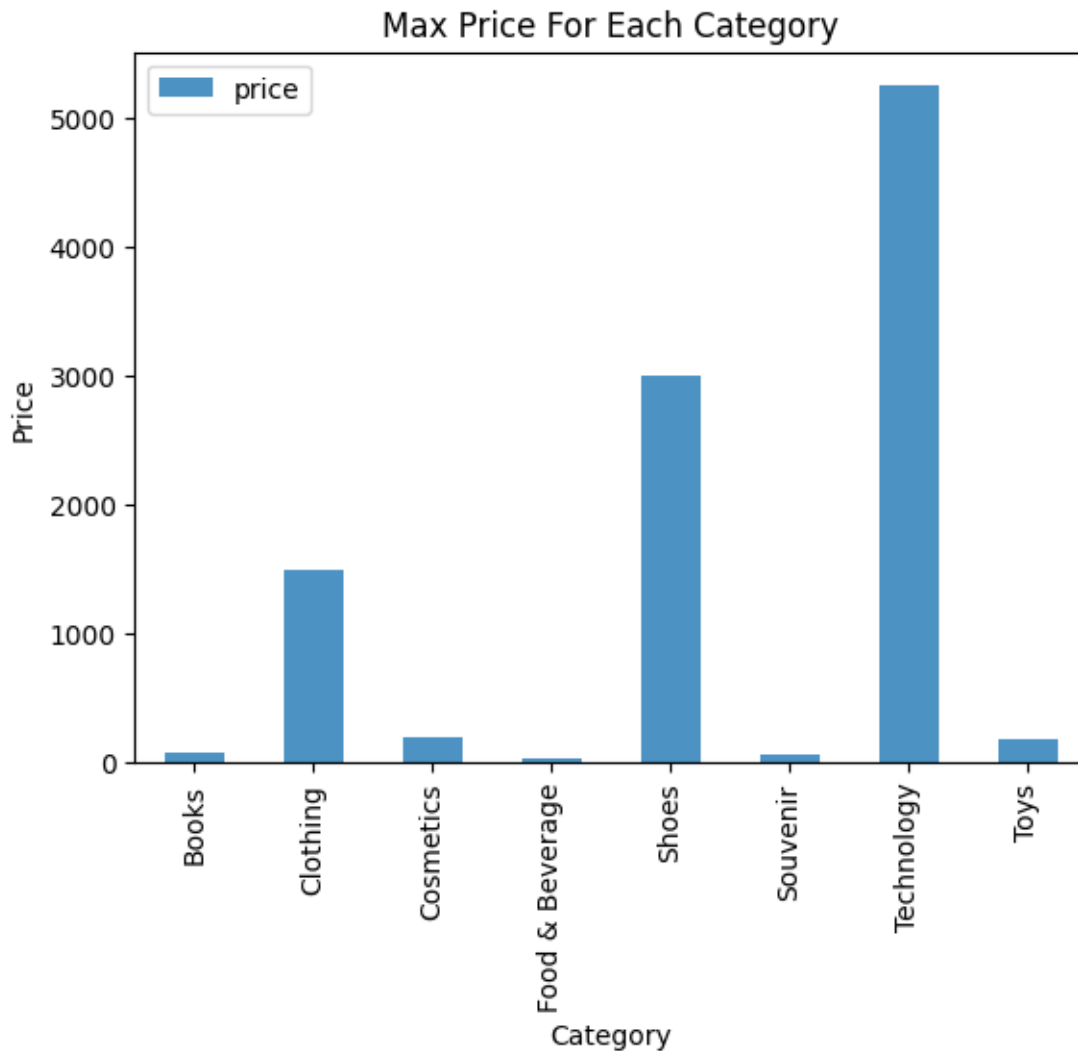
```
[ ]: df_category_max_price = df.groupby('category')['price'].max().reset_index()

df_category_max_price
```

```
[ ]:
      category    price
0      Books      75.75
1  Clothing  1500.40
2  Cosmetics   203.30
3 Food & Beverage   26.15
4      Shoes  3000.85
5  Souvenir    58.65
6  Technology  5250.00
7      Toys   179.20
```

```
[ ]: df_category_max_price.plot(kind='bar',x='category',y='price',alpha=0.8)
      plt.ylabel('Price')
      plt.xlabel('Category')
      plt.title('Max Price For Each Category')
      plt.xticks(rotation=90)
```

```
[ ]: (array([0, 1, 2, 3, 4, 5, 6, 7]),
      [Text(0, 0, 'Books'),
        Text(1, 0, 'Clothing'),
        Text(2, 0, 'Cosmetics'),
        Text(3, 0, 'Food & Beverage'),
        Text(4, 0, 'Shoes'),
        Text(5, 0, 'Souvenir'),
        Text(6, 0, 'Technology'),
        Text(7, 0, 'Toys')])
```



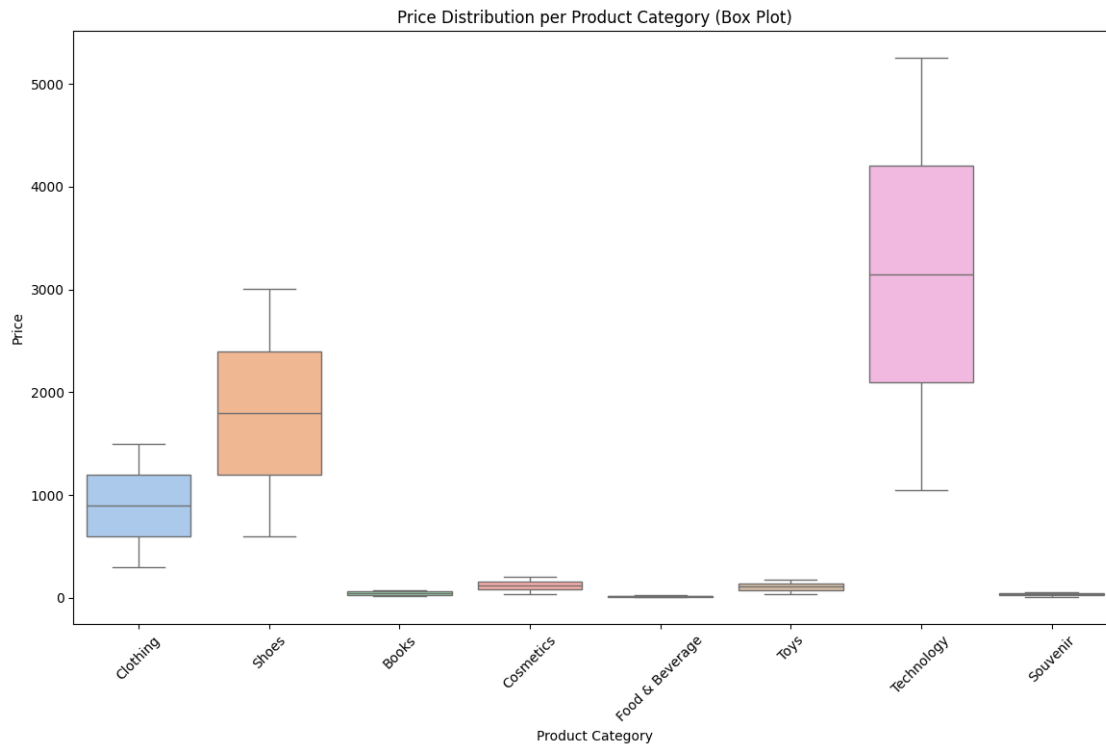
Distribution of price for each product category (using Boxplot)

```
[ ]: plt.figure(figsize=(14, 8))
sns.boxplot(x='category', y='price', data=df, palette='pastel')
plt.title('Price Distribution per Product Category (Box Plot)')
plt.xlabel('Product Category')
plt.ylabel('Price')
plt.xticks(rotation=45)
plt.show()
```

<ipython-input-115-24506f0283a9>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.boxplot(x='category', y='price', data=df, palette='pastel')
```



Distribution of customer ages

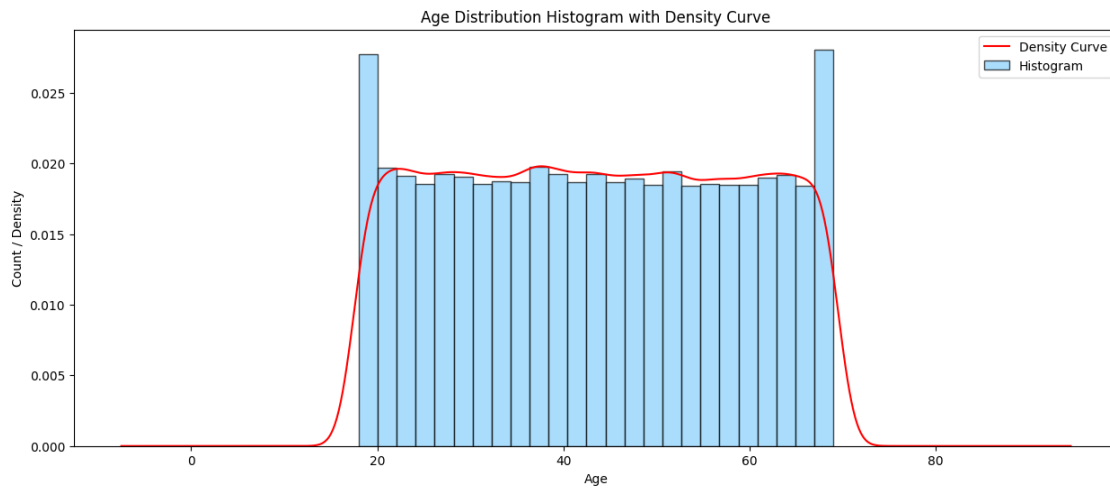
```
[ ]: fig, ax = plt.subplots(figsize = (15, 6))

ax.hist(df['age'], bins = 25, edgecolor = 'black', alpha = 0.7, color = 'lightskyblue', density = True)
df['age'].plot(kind = 'kde', color = 'red', ax = ax)

ax.set_xlabel('Age')
ax.set_ylabel('Count / Density')
ax.set_title('Age Distribution Histogram with Density Curve')

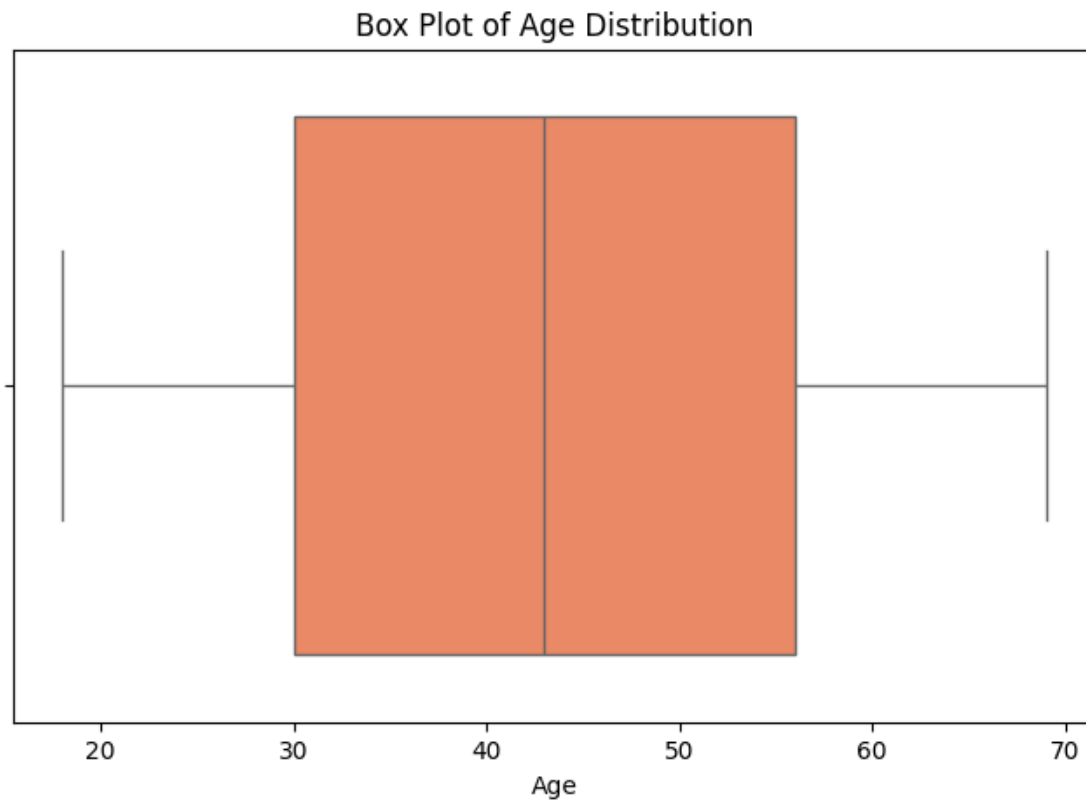
ax.legend(['Density Curve', 'Histogram'])

plt.show()
```



Boxplot

```
[ ]: plt.figure(figsize=(8, 5))
sns.boxplot(x=df['age'], color='coral')
plt.title('Box Plot of Age Distribution')
plt.xlabel('Age')
plt.show()
```



Average total amount spent by different age groups

```
[ ]: average_spent = df.groupby('age')['total_amount'].mean()
```

```
[ ]: average_spent
```

```
[ ]: age
18    2384.880835
19    2403.584003
20    2498.819875
21    2494.174838
22    2539.659391
23    2450.074544
24    2570.768791
25    2503.589630
26    2627.328492
27    2652.235415
28    2529.511224
29    2590.858790
30    2499.723695
31    2385.349614
32    2329.902776
33    2407.032107
34    2522.695874
35    2573.998664
36    2393.674253
37    2711.978401
38    2555.898429
39    2637.736400
40    2568.473031
41    2492.023462
42    2654.072595
43    2525.161860
44    2695.213655
45    2583.116754
46    2618.962428
47    2452.222027
48    2624.934967
49    2490.200313
50    2445.107117
51    2628.562338
52    2454.629532
53    2529.321067
54    2577.008678
55    2529.595388
```

```

56    2376.815172
57    2560.102448
58    2611.470699
59    2427.312289
60    2547.873036
61    2413.708303
62    2512.653997
63    2560.497879
64    2633.603946
65    2540.787883
66    2538.082100
67    2569.558827
68    2623.941215
69    2418.506076
Name: total_amount, dtype: float64

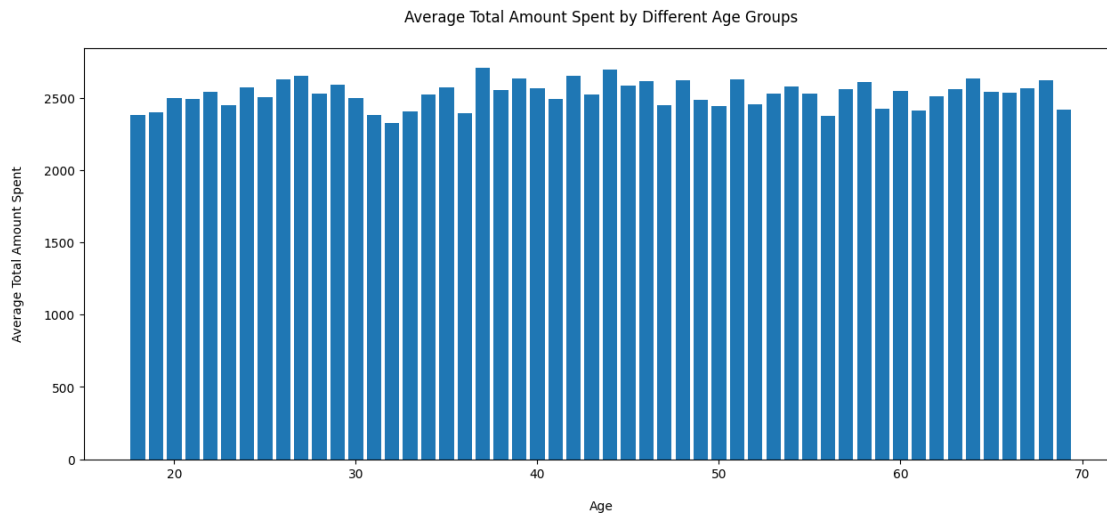
```

```

[ ]: plt.figure(figsize=(15,6))

plt.bar(average_spent.index, average_spent.values)
plt.xlabel('\nAge')
plt.ylabel('Average Total Amount Spent\n')
plt.title('Average Total Amount Spent by Different Age Groups\n')
plt.show()

```



```

[ ]: #=====

```

Distribution of transaction across months

```

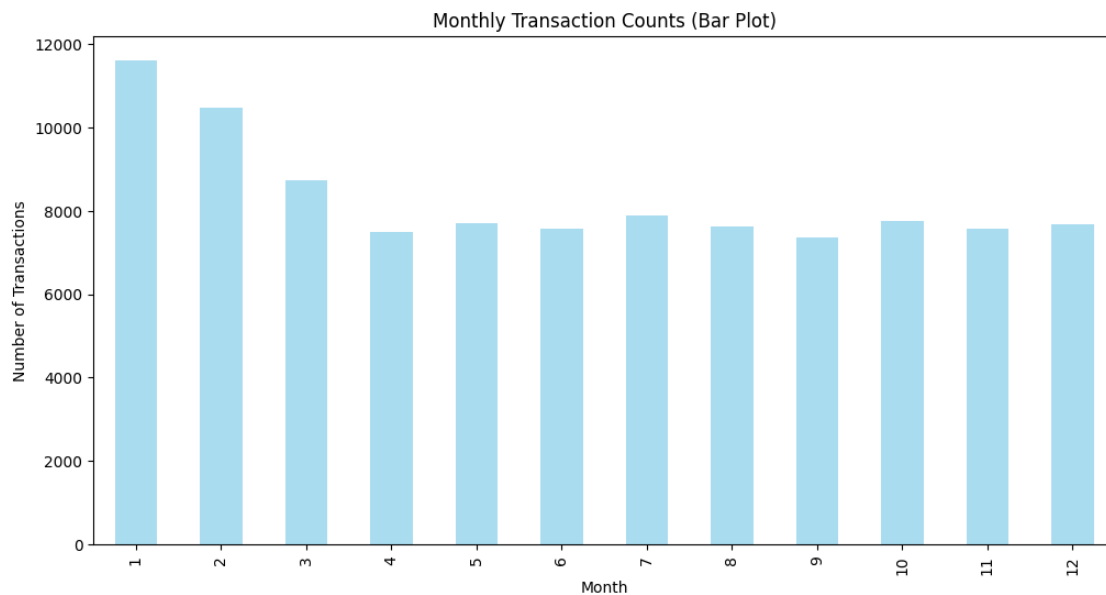
[ ]: monthly_transaction_count = df['month'].value_counts().sort_index()

```

```
monthly_transaction_count
```

```
[ ]: 1    11608
      2    10482
      3     8730
      4     7487
      5     7697
      6     7581
      7     7877
      8     7635
      9     7353
     10     7764
     11     7563
     12     7680
      Name: month, dtype: int64
```

```
[ ]: plt.figure(figsize=(12, 6))
      monthly_transaction_count.plot(kind='bar', color='skyblue', alpha=0.7)
      plt.title('Monthly Transaction Counts (Bar Plot)')
      plt.xlabel('Month')
      plt.ylabel('Number of Transactions')
      plt.show()
```



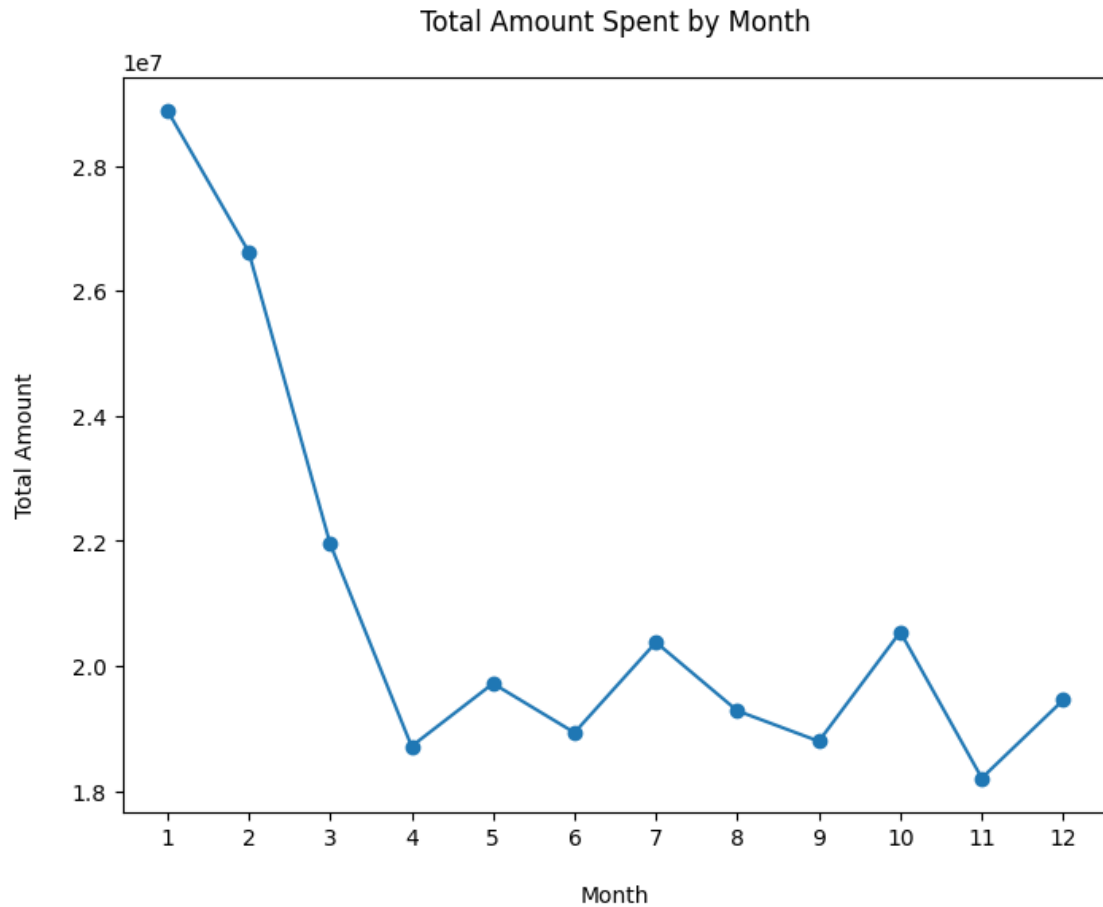
Total amount spent variations across various months

```
[ ]: monthly_total_amount = df.groupby('month')['total_amount'].sum()
```

```
monthly_total_amount
```

```
[ ]: month
1      28891525.59
2      26625090.10
3      21956191.33
4      18715685.98
5      19719331.10
6      18933775.30
7      20378722.63
8      19282361.29
9      18795794.91
10     20545090.43
11     18207139.95
12     19455085.64
Name: total_amount, dtype: float64
```

```
[ ]: plt.figure(figsize=(8, 6))
monthly_total_amount.plot(kind='line', marker='o')
plt.title('Total Amount Spent by Month\n')
plt.xlabel('\nMonth')
plt.ylabel('Total Amount\n')
plt.xticks(range(1, 13))
plt.show()
```

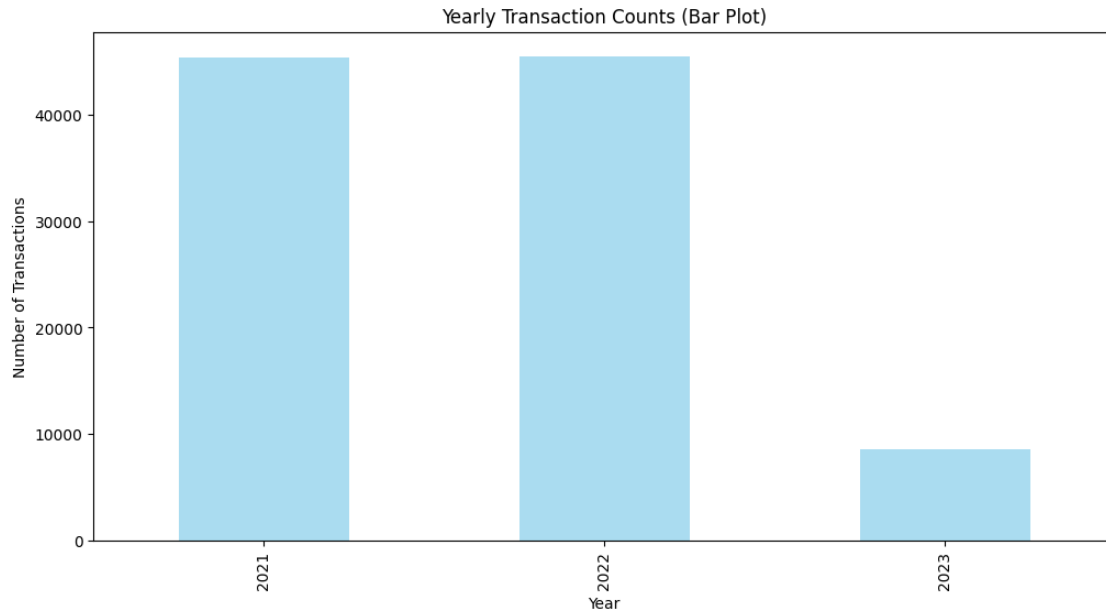
Distribution of transaction across years

```
[ ]: yearly_transaction_count = df['year'].value_counts().sort_index()

yearly_transaction_count
```

```
[ ]: 2021    45382
     2022    45551
     2023     8524
     Name: year, dtype: int64
```

```
[ ]: plt.figure(figsize=(12, 6))
     yearly_transaction_count.plot(kind='bar', color='skyblue', alpha=0.7)
     plt.title('Yearly Transaction Counts (Bar Plot)')
     plt.xlabel('Year')
     plt.ylabel('Number of Transactions')
     plt.show()
```



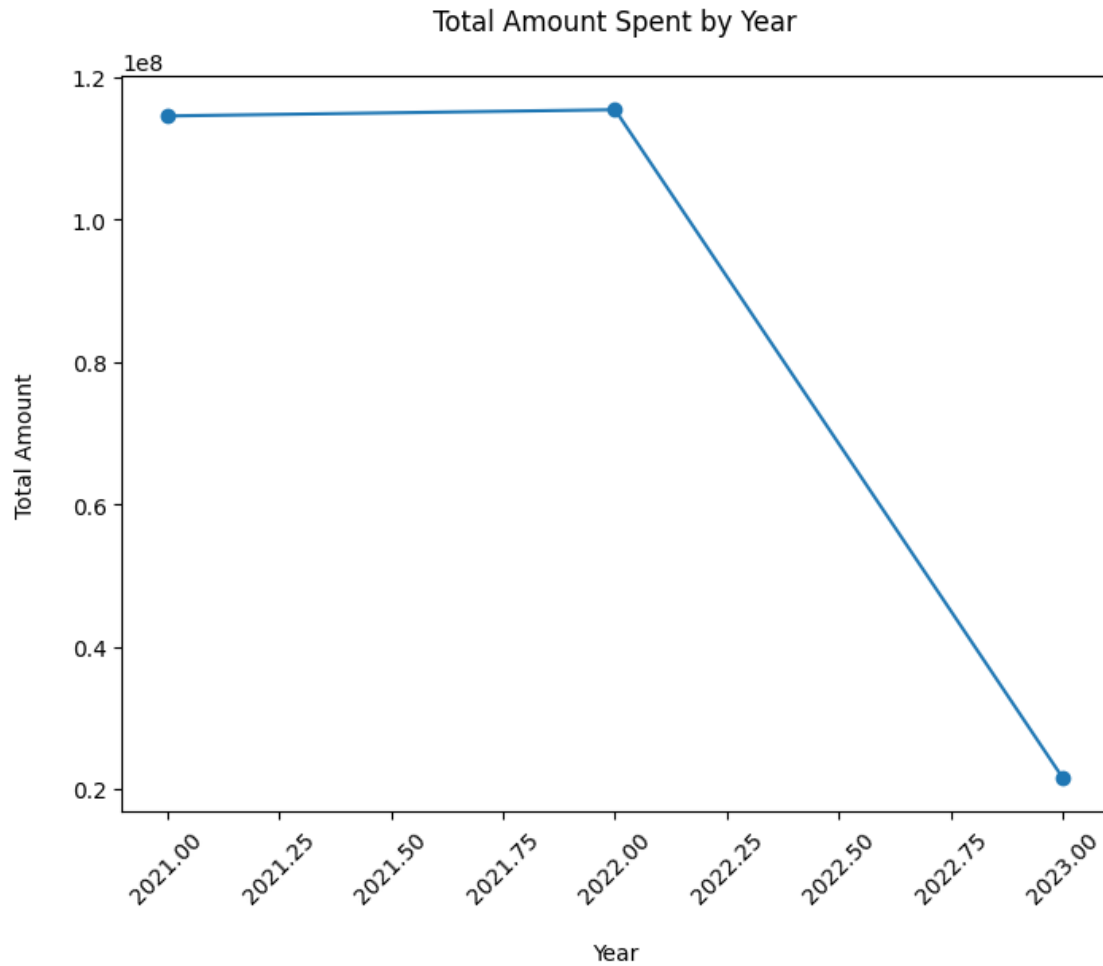
Total amount spent variations across years

```
[ ]: yearly_total_amount = df.groupby('year')['total_amount'].sum()

yearly_total_amount
```

```
[ ]: year
2021    1.145606e+08
2022    1.154368e+08
2023    2.150841e+07
Name: total_amount, dtype: float64
```

```
[ ]: plt.figure(figsize=(8, 6))
yearly_total_amount.plot(kind='line', marker='o')
plt.title('Total Amount Spent by Year\n')
plt.xlabel('\nYear')
plt.ylabel('Total Amount\n')
plt.xticks(rotation=45)
plt.show()
```



Distribution of transactions by age and category

```
[ ]: bins = [18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 70]
labels = ['18:20', '21:23', '24:26', '27:29', '30:32', '33:35', '36:38', '39:
41', '42:44', '45:47', '48:50', '51:53',
'54:56', '57:59', '60+']
df['AgeGroup'] = pd.cut(df['age'], bins=bins, labels=labels, right=False)
distribution = df.groupby(['AgeGroup', 'category']).size().unstack()
distribution
```

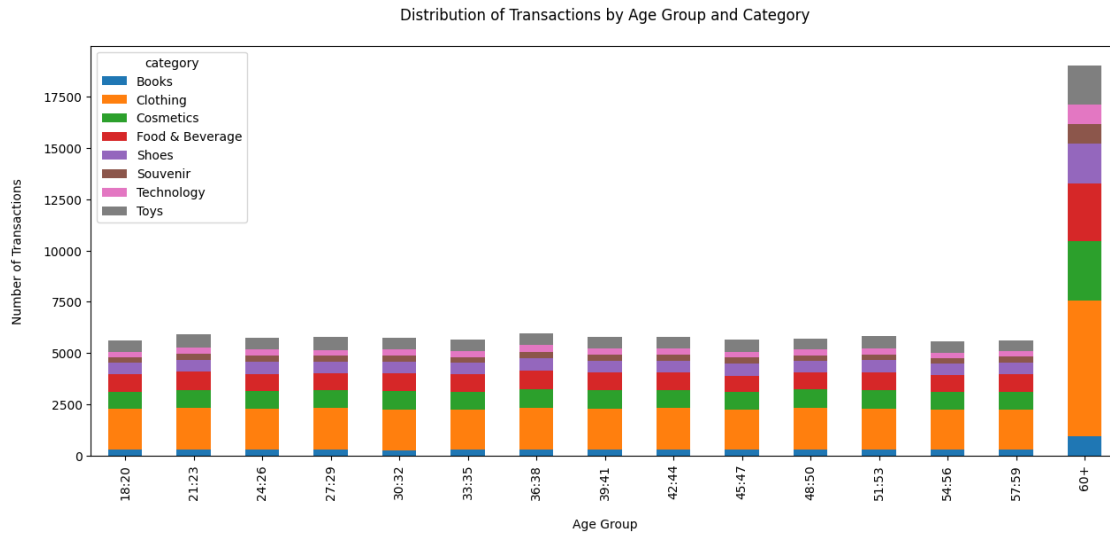
```
[ ]: category Books Clothing Cosmetics Food & Beverage Shoes Souvenir \
AgeGroup
18:20      289    1974      825      863    575    277
21:23      289    2028      886      881    588    292
24:26      296    1997      850      836    594    310
27:29      288    2044      852      818    585    271
30:32      259    1979      887      864    570    309
```

33:35	287	1929	871	864	566	279
36:38	294	2038	896	917	579	316
39:41	299	1998	879	862	574	292
42:44	269	2053	856	874	565	291
45:47	272	1962	857	805	611	272
48:50	309	2013	891	846	573	264
51:53	287	1992	904	864	597	284
54:56	301	1925	870	827	547	290
57:59	285	1952	884	841	570	293
60+	957	6603	2889	2814	1940	959

category	Technology	Toys
18:20	252	569
21:23	291	640
24:26	295	558
27:29	290	640
30:32	291	579
33:35	294	547
36:38	335	590
39:41	296	599
42:44	309	579
45:47	290	598
48:50	271	544
51:53	314	599
54:56	257	572
57:59	250	553
60+	961	1920

```
[ ]: %matplotlib inline
matplotlib.rcParams['figure.figsize'] = (15, 6)

distribution.plot(kind='bar', stacked=True)
plt.xlabel('\nAge Group')
plt.ylabel('Number of Transactions\n')
plt.title('Distribution of Transactions by Age Group and Category\n')
plt.show()
```

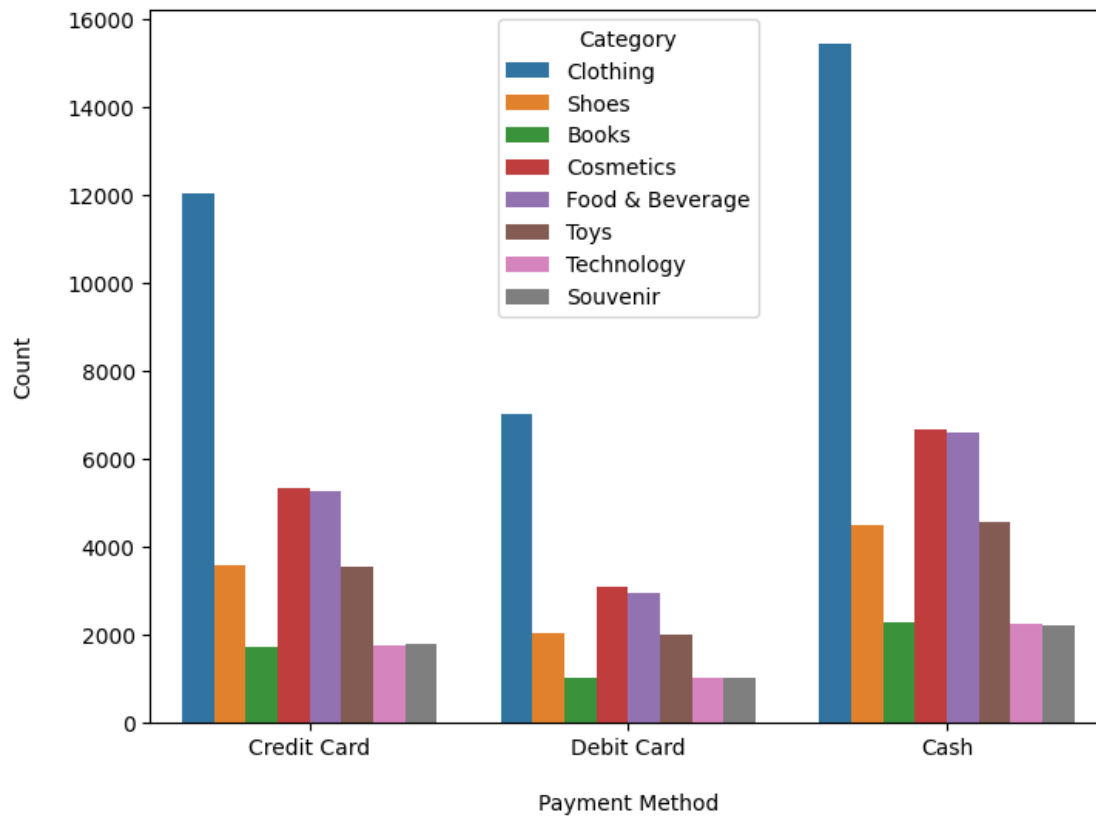


Distribution of transactions by payment method and category

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

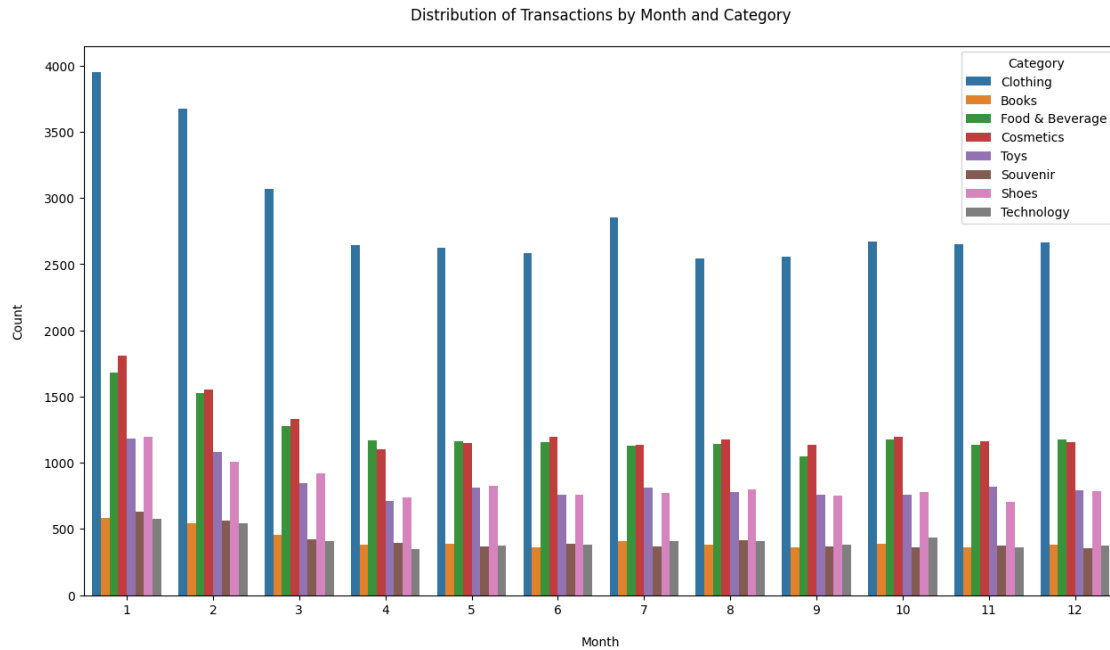
sns.countplot(data=df, x='payment_method', hue='category')
plt.title('Distribution of Transactions by Payment Method and Category\n')
plt.xlabel('\nPayment Method')
plt.ylabel('Count\n')
plt.legend(title='Category')
plt.show()
```

Distribution of Transactions by Payment Method and Category



Distribution of transactions by month and category

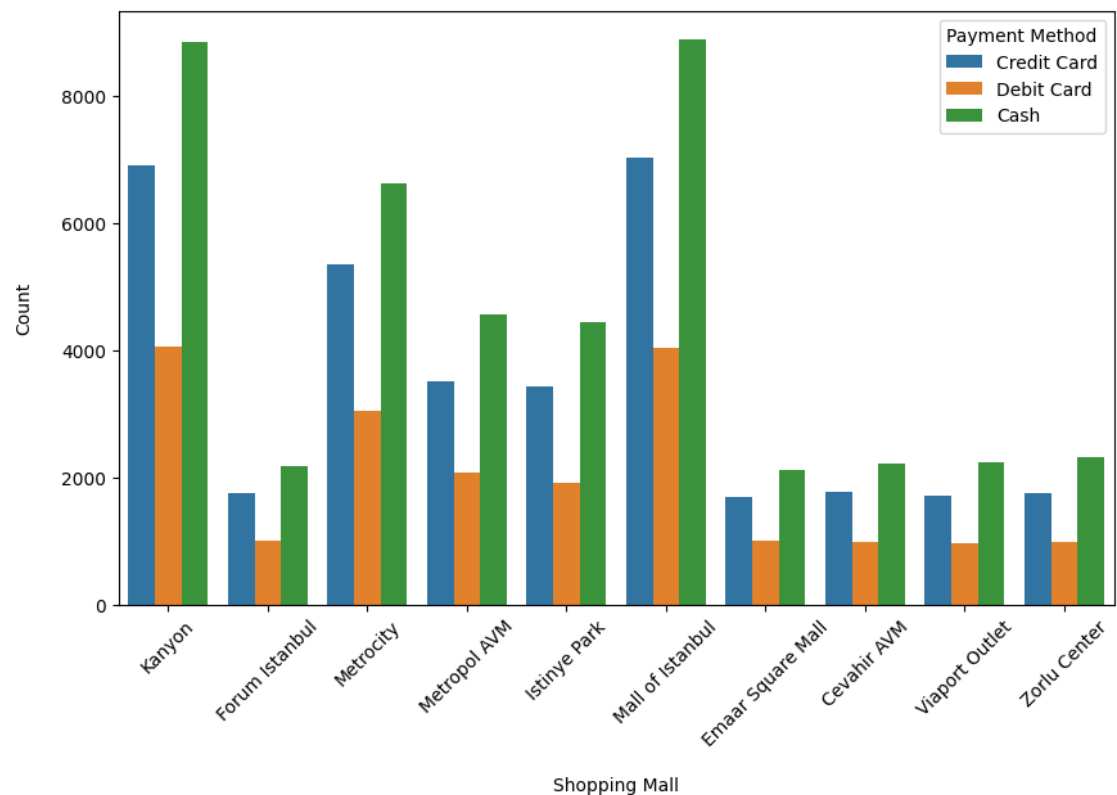
```
[ ]: plt.figure(figsize=(15, 8))
sns.countplot(data=df, x='month', hue='category')
plt.title('Distribution of Transactions by Month and Category\n')
plt.xlabel('\nMonth')
plt.ylabel('Count\n')
plt.legend(title='Category')
plt.show()
```



Distribution of transactions by shopping mall and payment method

```
[ ]: plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='shopping_mall', hue='payment_method')
plt.title('Distribution of Transactions by Shopping Mall and Payment Method\n')
plt.xlabel('\nShopping Mall')
plt.ylabel('Count\n')
plt.legend(title='Payment Method')
plt.xticks(rotation=45)
plt.show()
```

Distribution of Transactions by Shopping Mall and Payment Method



[]: