➡Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
print('All libraries loaded sucessfully...!')
```

```
All libraries loaded sucessfully...!
```

➡Loading a CSV file in our dataframe(df)

In [2]:
```
df = pd.read_csv("C:/Users/OCS/Downloads/shopping_trends.csv")
```

➡To check total number of rows and columns

In [3]:
```
df.shape
```

Out[3]:
```
(3900, 19)
```

➡To show the top 5 rows of our data

In [4]:
```
df.head()
```

Out[4]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Payme Meth
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Cre Ca
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Ba Transf
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Ca
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	PayF
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Ca

➡To show the bottom 5 rows of our data

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

Out[5]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Payn Met
3895	3896	40	Female	Hoodie	Clothing	28	Virginia	L	Turquoise	Summer	4.2	No	C
3896	3897	52	Female	Backpack	Accessories	49	Iowa	L	White	Spring	4.5	No	Pa
3897	3898	46	Female	Belt	Accessories	33	New Jersey	L	Green	Spring	2.9	No	C
3898	3899	44	Female	Shoes	Footwear	77	Minnesota	S	Brown	Summer	3.8	No	Pa
3899	3900	52	Female	Handbag	Accessories	81	California	M	Beige	Spring	3.1	No	Trar

➡DATA CLEANING

Step 1 - Showing all the rows,columns,data_type to see that there is no null values present in our dataset

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 19 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Customer ID               3900 non-null   int64
 1   Age                       3900 non-null   int64
 2   Gender                    3900 non-null   object
 3   Item Purchased            3900 non-null   object
 4   Category                  3900 non-null   object
 5   Purchase Amount (USD)     3900 non-null   int64
 6   Location                  3900 non-null   object
 7   Size                      3900 non-null   object
 8   Color                     3900 non-null   object
 9   Season                    3900 non-null   object
 10  Review Rating             3900 non-null   float64
 11  Subscription Status       3900 non-null   object
 12  Payment Method            3900 non-null   object
 13  Shipping Type             3900 non-null   object
 14  Discount Applied          3900 non-null   object
 15  Promo Code Used           3900 non-null   object
 16  Previous Purchases        3900 non-null   int64
 17  Preferred Payment Method  3900 non-null   object
 18  Frequency of Purchases    3900 non-null   object
dtypes: float64(1), int64(4), object(14)
memory usage: 579.0+ KB
```

Step 2(a)- To check Null values in the form of 'True' or 'False'

In [7]:
```
pd.isnull(df)
```

Out[7]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Payment Method
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3895	False	False	False	False	False	False	False	False	False	False	False	False	False
3896	False	False	False	False	False	False	False	False	False	False	False	False	False
3897	False	False	False	False	False	False	False	False	False	False	False	False	False
3898	False	False	False	False	False	False	False	False	False	False	False	False	False
3899	False	False	False	False	False	False	False	False	False	False	False	False	False

3900 rows × 19 columns

(b)- To check Null values in the form of sum of all the values

In [8]:
```
pd.isnull(df).sum()
```

Out[8]:
```
Customer ID                 0
Age                         0
Gender                      0
Item Purchased              0
Category                    0
Purchase Amount (USD)       0
Location                    0
Size                        0
Color                       0
Season                      0
Review Rating               0
Subscription Status         0
Payment Method              0
Shipping Type               0
Discount Applied            0
Promo Code Used             0
Previous Purchases          0
Preferred Payment Method    0
Frequency of Purchases      0
dtype: int64
```

➡To see all the columns

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

Out[9]:
```
Index(['Customer ID', 'Age', 'Gender', 'Item Purchased', 'Category',
       'Purchase Amount (USD)', 'Location', 'Size', 'Color', 'Season',
       'Review Rating', 'Subscription Status', 'Payment Method',
       'Shipping Type', 'Discount Applied', 'Promo Code Used',
       'Previous Purchases', 'Preferred Payment Method',
       'Frequency of Purchases'],
      dtype='object')
```

➡This method returns description of the data in the DataFrame (i.e. count, mean, std, etc)

In [10]:
```
df.describe()
```

Out[10]:

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3900.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.749949	25.351538
std	1125.977353	15.207589	23.685392	0.716223	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.700000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

▶ EXPLORATORY DATA ANALYSIS

Step 3 - To find Unique values in each columns

In [11]:
```
for col in df.describe(include="object"):
    print(col)
    print(df[col].unique())
```

```
Gender
['Male' 'Female']
Item Purchased
['Blouse' 'Sweater' 'Jeans' 'Sandals' 'Sneakers' 'Shirt' 'Shorts' 'Coat'
 'Handbag' 'Shoes' 'Dress' 'Skirt' 'Sunglasses' 'Pants' 'Jacket' 'Hoodie'
 'Jewelry' 'T-shirt' 'Scarf' 'Hat' 'Socks' 'Backpack' 'Belt' 'Boots'
 'Gloves']
Category
['Clothing' 'Footwear' 'Outerwear' 'Accessories']
Location
['Kentucky' 'Maine' 'Massachusetts' 'Rhode Island' 'Oregon' 'Wyoming'
 'Montana' 'Louisiana' 'West Virginia' 'Missouri' 'Arkansas' 'Hawaii'
 'Delaware' 'New Hampshire' 'New York' 'Alabama' 'Mississippi'
 'North Carolina' 'California' 'Oklahoma' 'Florida' 'Texas' 'Nevada'
 'Kansas' 'Colorado' 'North Dakota' 'Illinois' 'Indiana' 'Arizona'
 'Alaska' 'Tennessee' 'Ohio' 'New Jersey' 'Maryland' 'Vermont'
 'New Mexico' 'South Carolina' 'Idaho' 'Pennsylvania' 'Connecticut' 'Utah'
 'Virginia' 'Georgia' 'Nebraska' 'Iowa' 'South Dakota' 'Minnesota'
 'Washington' 'Wisconsin' 'Michigan']
Size
['L' 'S' 'M' 'XL']
Color
['Gray' 'Maroon' 'Turquoise' 'White' 'Charcoal' 'Silver' 'Pink' 'Purple'
 'Olive' 'Gold' 'Violet' 'Teal' 'Lavender' 'Black' 'Green' 'Peach' 'Red'
 'Cyan' 'Brown' 'Beige' 'Orange' 'Indigo' 'Yellow' 'Magenta' 'Blue']
Season
['Winter' 'Spring' 'Summer' 'Fall']
Subscription Status
['Yes' 'No']
Payment Method
['Credit Card' 'Bank Transfer' 'Cash' 'PayPal' 'Venmo' 'Debit Card']
Shipping Type
['Express' 'Free Shipping' 'Next Day Air' 'Standard' '2-Day Shipping'
 'Store Pickup']
Discount Applied
['Yes' 'No']
Promo Code Used
['Yes' 'No']
Preferred Payment Method
['Venmo' 'Cash' 'Credit Card' 'PayPal' 'Bank Transfer' 'Debit Card']
Frequency of Purchases
['Fortnightly' 'Weekly' 'Annually' 'Quarterly' 'Bi-Weekly' 'Monthly'
 'Every 3 Months']
```

➡ Summarization of data from dataset

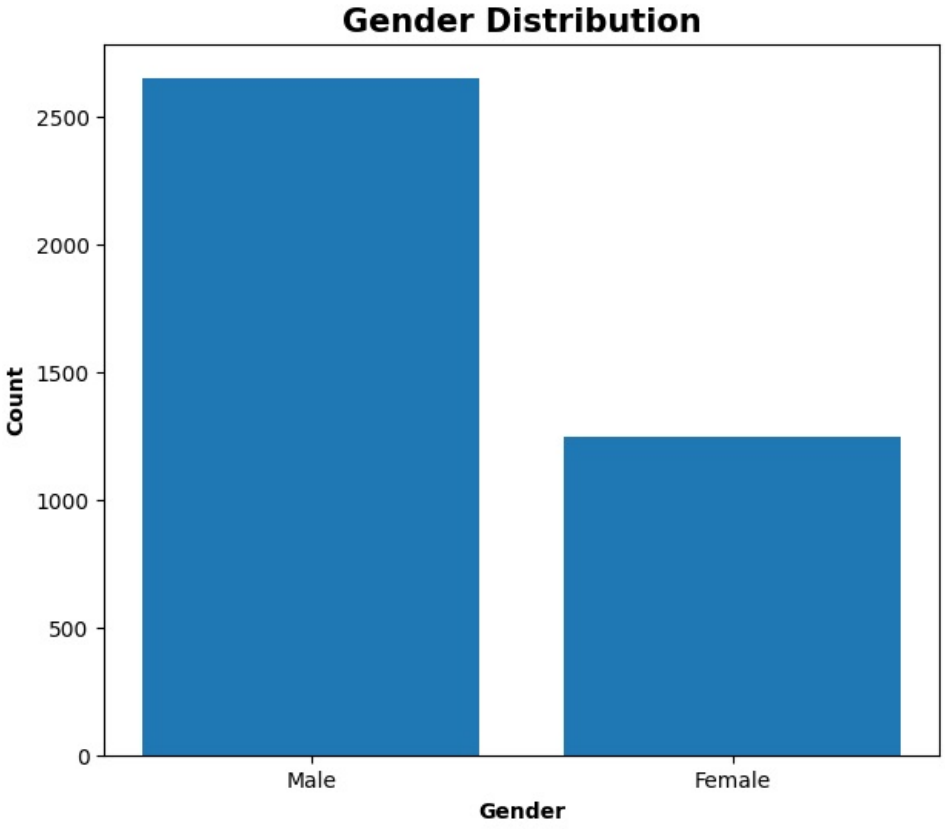
In [12]:
```
df.describe(include="all")
```

Out[12]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subsc
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3900.000000	
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.749949	
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716223	
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.700000	
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	

==> Bar chart for the Gender Distribution

In [13]:
```
plt.figure(figsize=(7,6))
plt.title("Gender Distribution", fontsize = 15, fontweight = 'bold')
plt.bar(["Male","Female"],df["Gender"].value_counts())
plt.xlabel('Gender',fontweight = 'bold')
plt.ylabel('Count',fontweight = 'bold')
plt.show()
```



==> Making a dataframe - 'age_groups'

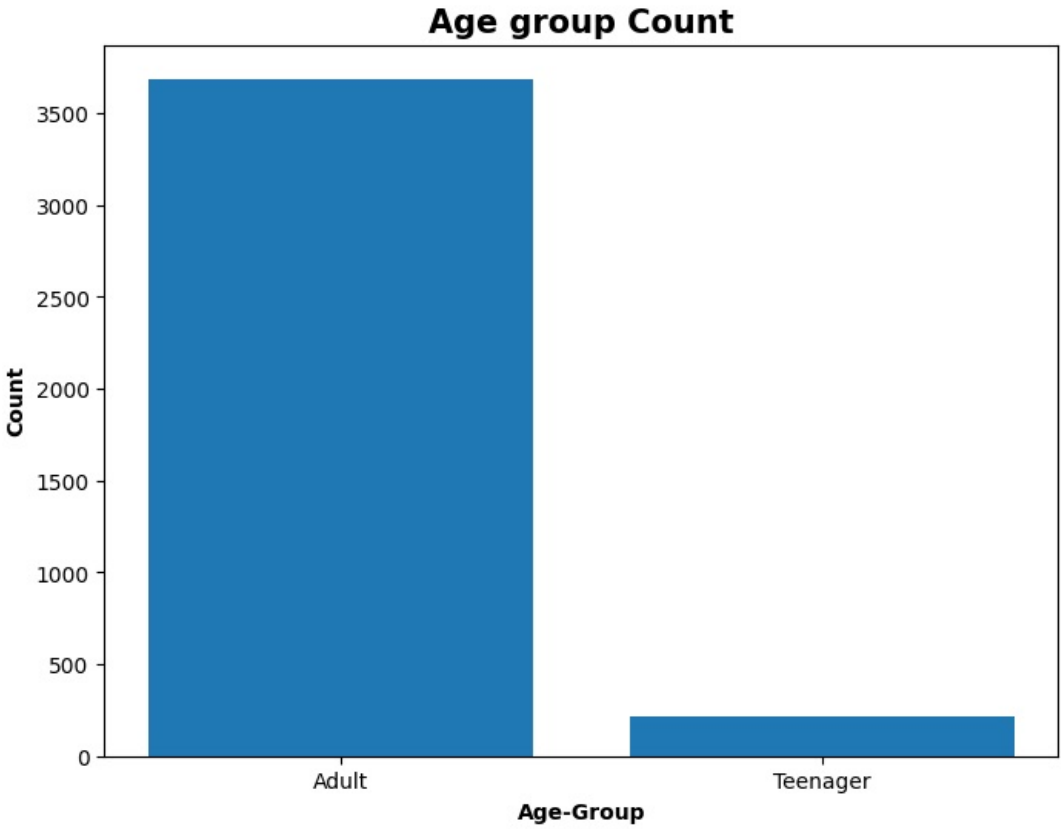
In [14]:
```
df["age_groups"] = np.where(df["Age"] >= 21, "Adult", "Teenager")
print(df["age_groups"].value_counts())
```

```
age_groups
Adult       3688
Teenager     212
Name: count, dtype: int64
```

==> Bar chart distribution for the age groups - 'Teenagers' and 'Adults'

In [15]:
```
print("\nNumber of teenagers and adult who bought clothes")
plt.figure(figsize=(8, 6))
plt.title("Age group Count", fontsize = 15, fontweight = 'bold')
plt.bar(["Adult", "Teenager"],df["age_groups"].value_counts())
plt.xlabel("Age-Group",fontweight = 'bold')
plt.ylabel("Count",fontweight = 'bold')
plt.show()
```

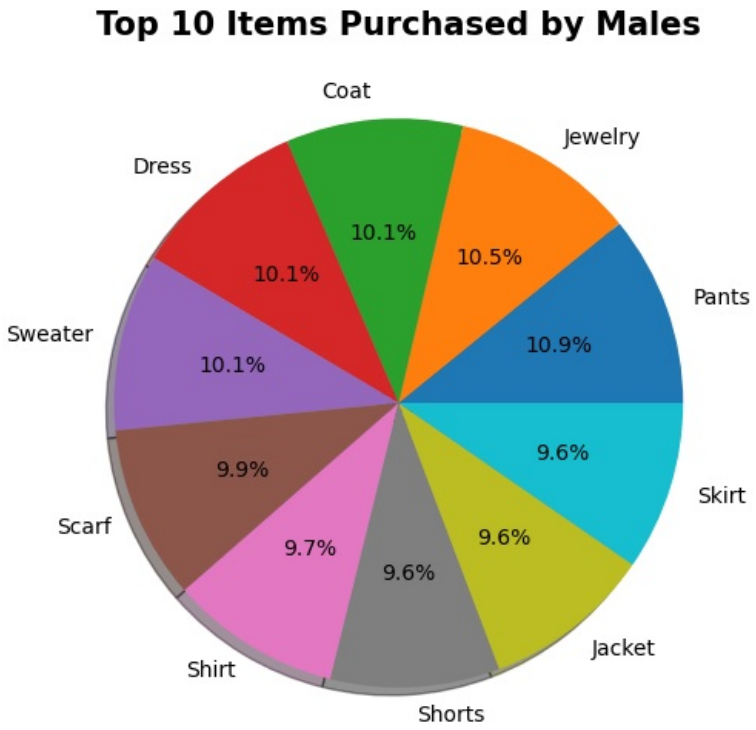
```
Number of teenagers and adult who bought clothes
```



==> Pie chart for the Top 10 items purchased by males

In [16]:
```
print("\nTop 10 Items Purchased by Males")
males = df[df["Gender"] == "Male"]
top_10_items_purchased = males['Item Purchased'].value_counts()[:10]
print(top_10_items_purchased)
plt.figure(figsize=(10,6))
plt.title("Top 10 Items Purchased by Males", fontsize = 15, fontweight = 'bold')
plt.pie(top_10_items_purchased, autopct='%.1f%%', labels = top_10_items_purchased.index,shadow = True)
plt.show()
```

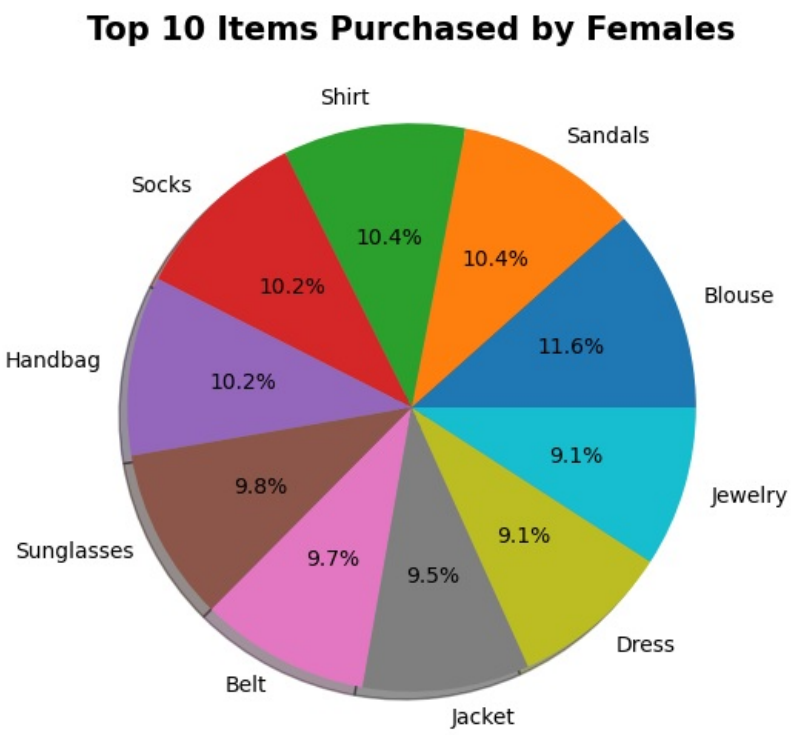
```
Top 10 Items Purchased by Males
Item Purchased
Pants      123
Jewelry    119
Coat       114
Dress      114
Sweater    114
Scarf      112
Shirt      110
Shorts     109
Jacket     109
Skirt      109
Name: count, dtype: int64
```



==> Pie chart for the Top 10 items purchased by females

In [17]:
```
print("\nTop 10 Items Purchased by Females")
females = df[df["Gender"] == "Female"]
top_10_items_purchased = females['Item Purchased'].value_counts()[:10]
print(top_10_items_purchased)
plt.figure(figsize=(10,6))
plt.title("Top 10 Items Purchased by Females", fontsize = 15, fontweight = 'bold')
plt.pie(top_10_items_purchased, autopct='%.1f%%', labels = top_10_items_purchased.index,shadow = True)
plt.show()
```

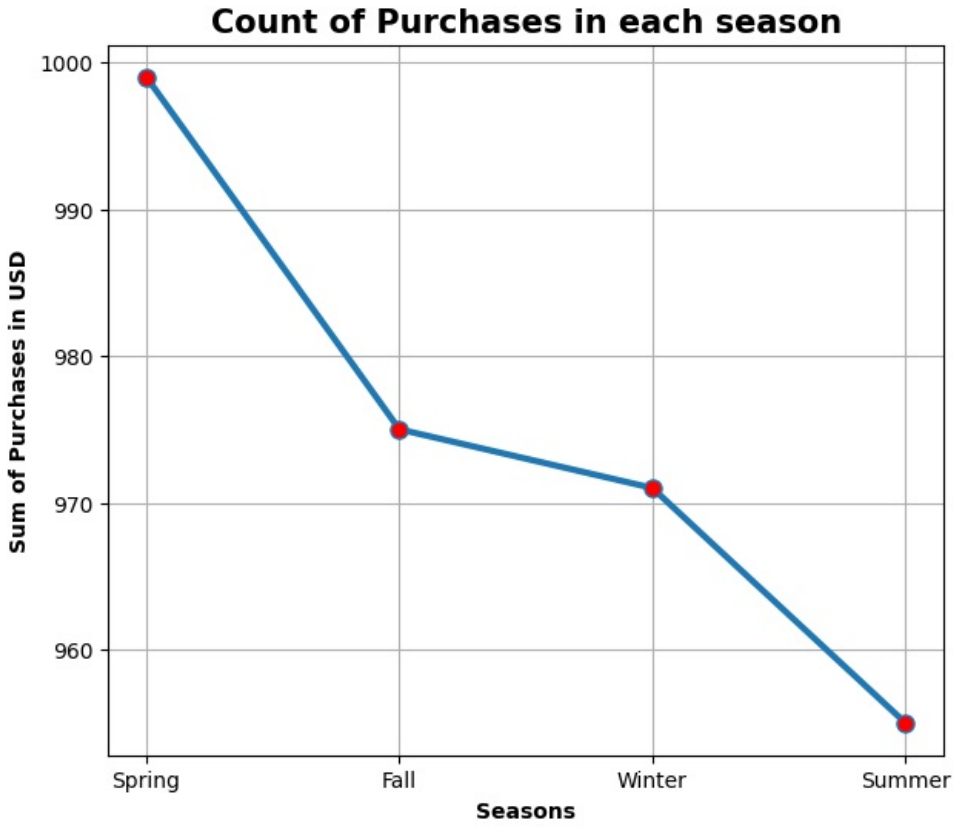
```
Top 10 Items Purchased by Females
Item Purchased
Blouse        66
Sandals       59
Shirt         59
Socks         58
Handbag       58
Sunglasses    56
Belt          55
Jacket        54
Dress         52
Jewelry       52
Name: count, dtype: int64
```



In [18]:
```
print("\nDistribution of Purchases in each season")
count_season= df['Season'].value_counts()
print(count_season)
```

```
Distribution of Purchases in each season
Season
Spring    999
Fall      975
Winter    971
Summer    955
Name: count, dtype: int64
```

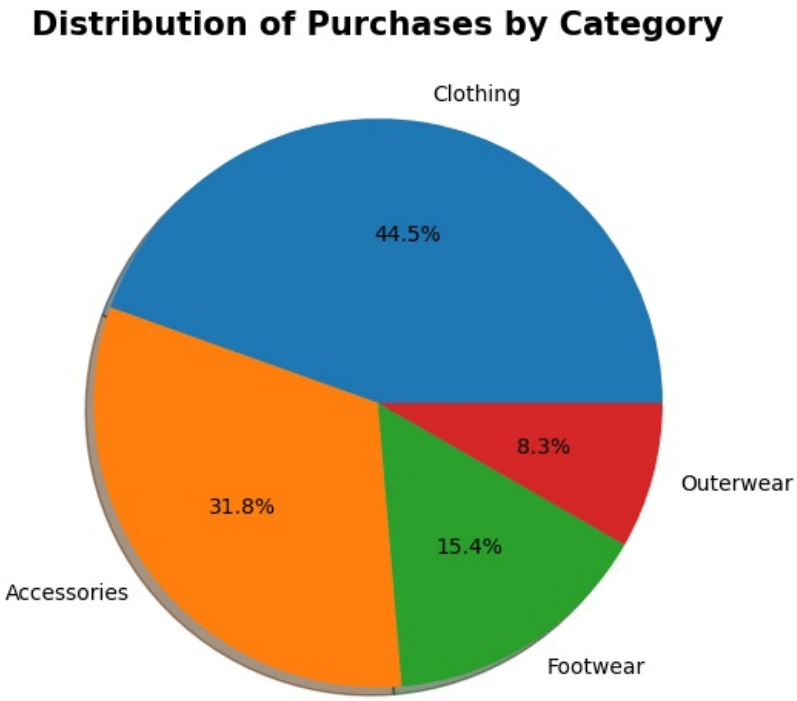
In [19]:
```
plt.figure(figsize=(7,6))
plt.title("Count of Purchases in each season", fontsize = 15, fontweight = 'bold')
plt.plot(count_season,linewidth = 3.0, marker = "o", markerfacecolor = "r", markersize = 8)
plt.xlabel('Seasons', fontweight = 'bold')
plt.ylabel('Sum of Purchases in USD', fontweight = 'bold')
plt.grid(True)
plt.show()
```



In [20]:
```
print("\nDistribution of Purchases by Category")
count_category= df['Category'].value_counts()
print(count_category)
```

```
Distribution of Purchases by Category
Category
Clothing       1737
Accessories    1240
Footwear        599
Outerwear       324
Name: count, dtype: int64
```

In [21]:
```
plt.figure(figsize=(10,6))
plt.title("Distribution of Purchases by Category", fontsize = 15, fontweight = 'bold')
plt.pie(count_category, autopct='%.1f%%', labels = count_category.index, shadow = True)
plt.show()
```

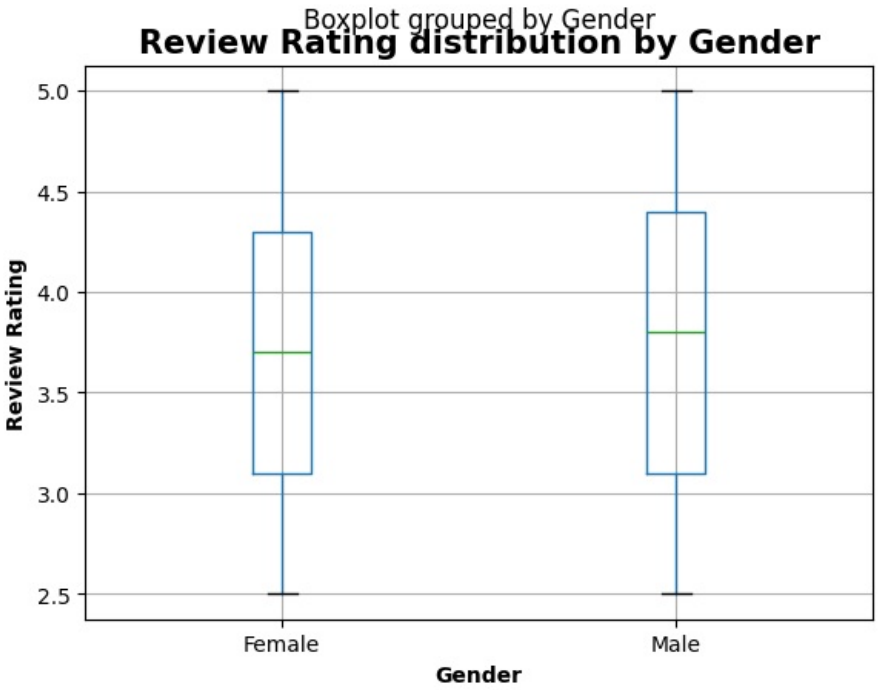


In [22]:
```
print("\nReview Rating Distribution by Gender")
count_rating= df['Review Rating'].value_counts().sort_values(ascending = False).head()
print(count_rating)
```

```
Review Rating Distribution by Gender
Review Rating
3.4    182
4.0    181
4.6    174
4.2    171
2.9    170
Name: count, dtype: int64
```

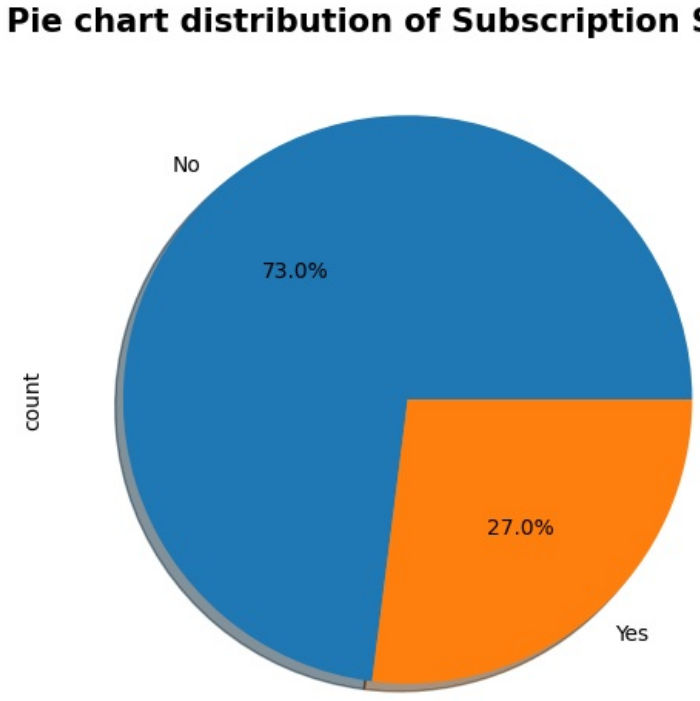
In [23]:
```
plt.figure(figsize = (10,10))
df.boxplot(column = "Review Rating", by = "Gender")
plt.xlabel("Gender", fontweight = 'bold')
plt.ylabel("Review Rating", fontweight = 'bold')
plt.title("Review Rating distribution by Gender", fontsize = 15, fontweight = 'bold')
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```



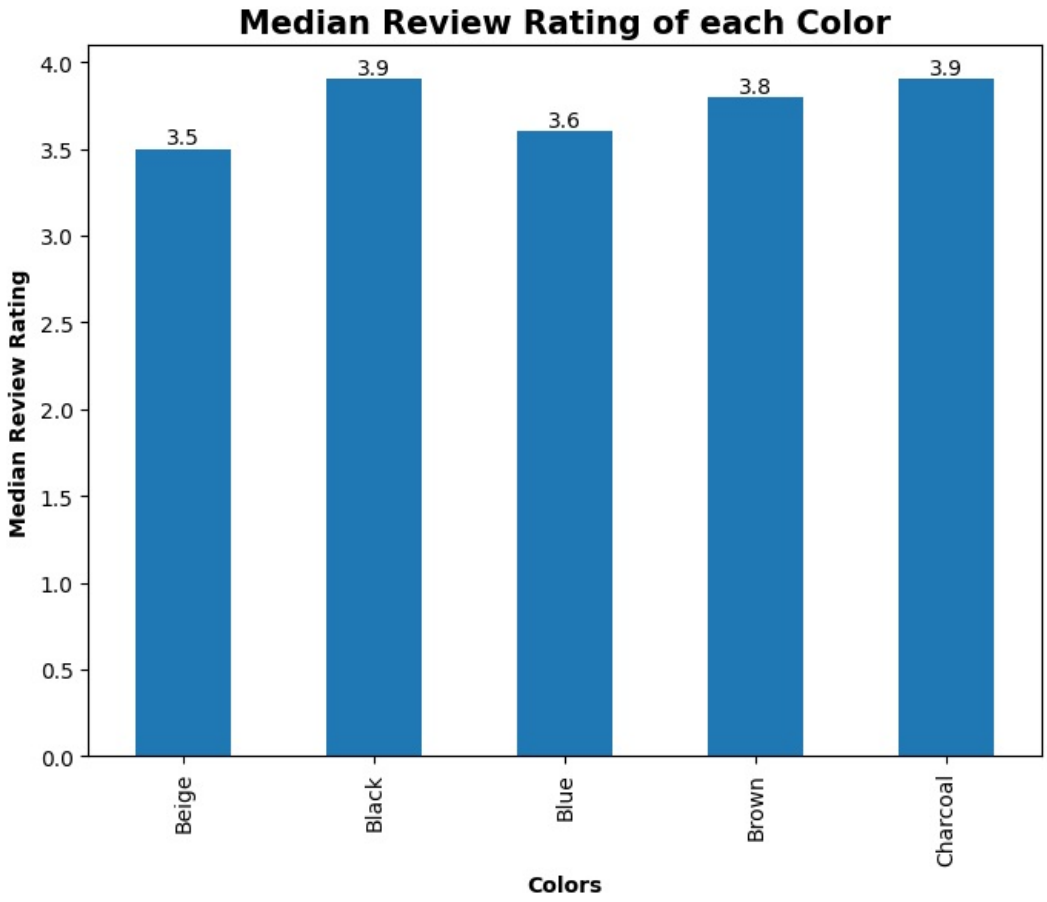
➡ Pie chart for the Subscription Status

In [24]:
```
plt.figure(figsize = (10,6))
df["Subscription Status"].value_counts().plot(kind = "pie",autopct='%.1f%%',shadow = True)
plt.title("Pie chart distribution of Subscription Status", fontsize = 15, fontweight = 'bold')
plt.show()
```



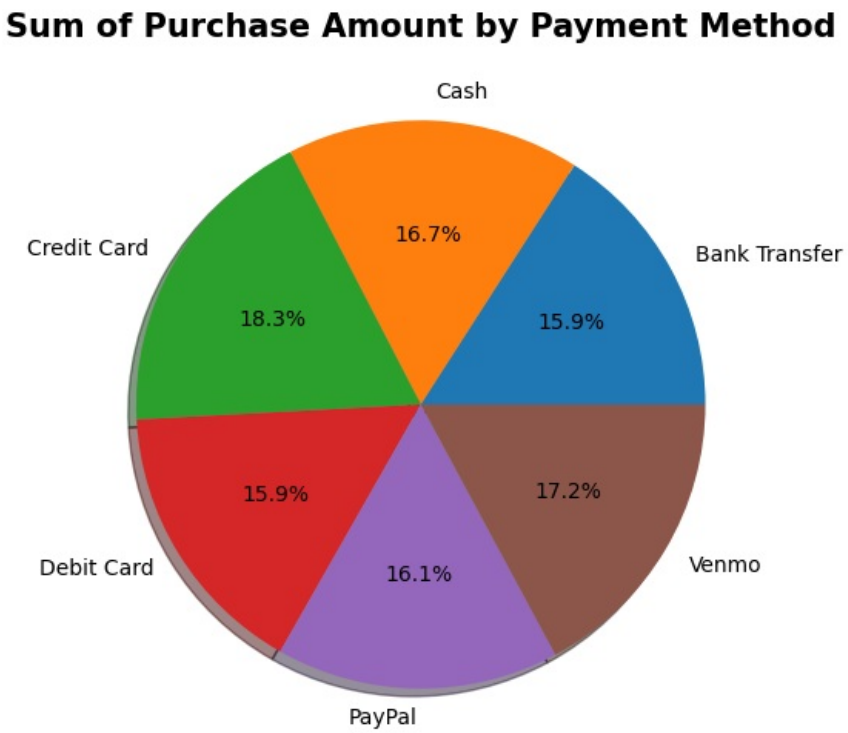
➡Median Review Rating of each Color

In [25]:
```
plt.figure(figsize = (8,6))
ax = df.groupby("Color")["Review Rating"].median().head().plot(kind = "bar")
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel("Colors", fontweight = 'bold')
plt.ylabel("Median Review Rating", fontweight = 'bold')
plt.title("Median Review Rating of each Color", fontsize = 15, fontweight = 'bold')
plt.show()
```



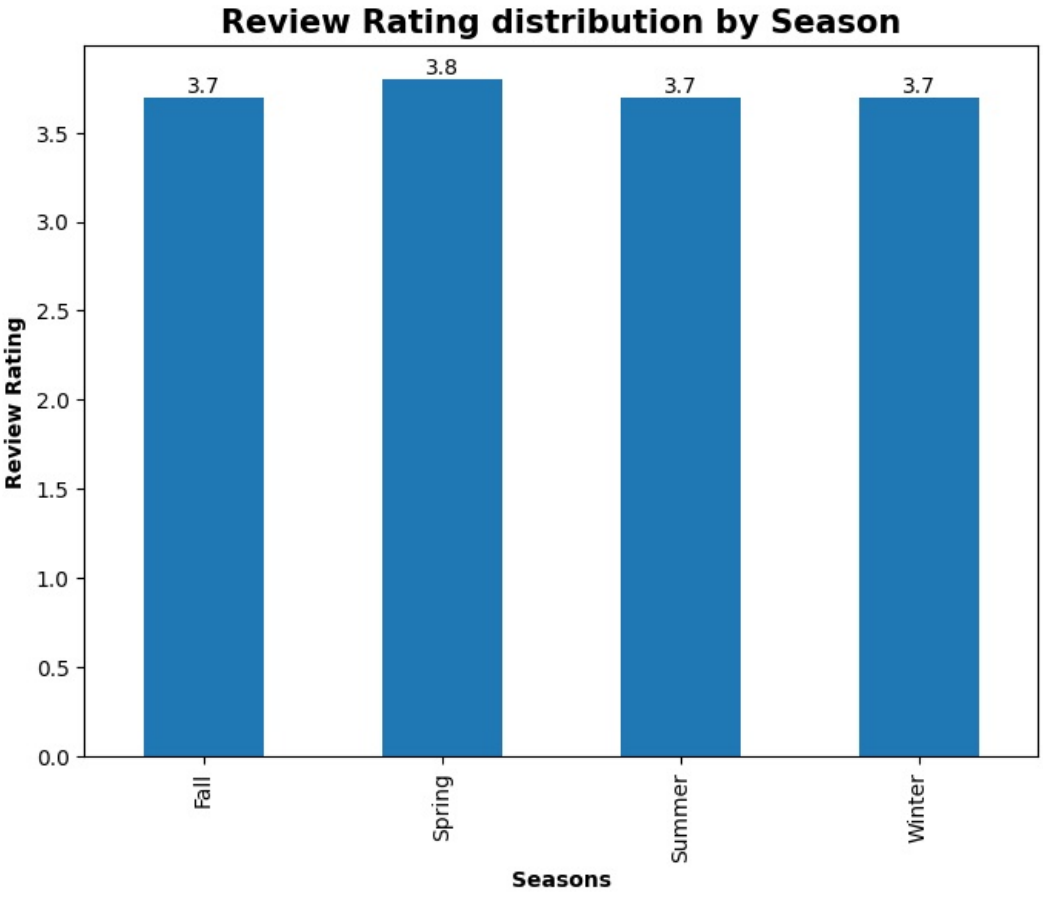
➡Sum of Purchase Amount by Payment Method

In [26]:
```
plt.figure(figsize = (8,6))
df.groupby("Payment Method")["Purchase Amount (USD)"].sum().plot(kind = "pie",autopct='%.1f%%',shadow = True)
plt.title("Sum of Purchase Amount by Payment Method",fontsize = 15, fontweight = 'bold')
plt.ylabel("")
plt.show()
```



➡Review Rating distribution by Season

In [27]:
```
plt.figure(figsize = (8,6))
ax= df.groupby('Season')["Review Rating"].median().plot(kind= "bar")
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel("Seasons", fontweight = 'bold')
plt.ylabel("Review Rating", fontweight = 'bold')
plt.title("Review Rating distribution by Season", fontsize = 15, fontweight = 'bold')
plt.show()
```



➡ OBSERVATION

—The dataset reflects a diverse consumer demographic, allowing for targeted marketing strategies tailored to specific age groups and genders. —Seasonal trends in shopping behavior highlight opportunities for targeted promotions and inventory adjustments to align with customer preferences. —The insights gained emphasize the importance of data-driven decision-making in navigating the dynamic

landscape of shopping trends. ─In all the seasons their is less review ratings. ─Subscription status is very less only 73% consumers subscribe to the website they visit.

➡ SOLUTION

─Develop targeted marketing campaigns based on demographic insights, tailoring messages and promotions to specific age groups and genders. ─Implement inventory optimization strategies, aligning stock levels with the popularity of key products and seasonal fluctuations. ─Time promotions strategically, capitalizing on identified seasonal trends and maximizing impact during peak purchasing periods. ─Design customer engagement tactics informed by correlation insights, aiming to enhance satisfaction levels and drive repeat purchases. ─Develop employee training programs based on identified correlations, enhancing skills that contribute to improved customer service and operational efficiency. ─Establish a feedback loop mechanism to continuously gather insights from stakeholders and customers, fostering a culture of continuous improvement.