In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv("Amazon.csv")
df.head(5)
```

Out[2]:

	product_id	product_name	category	discounted_price	actual_price	discount_percentag
0	B07JW9H4J1	Wayona Nylon Braided USB to Lightning Fast Cha...	Computers&Accessories|Accessories&Peripherals|...	₹399	₹1,099	64
1	B098NS6PVG	Ambrane Unbreakable 60W / 3A Fast Charging 1.5...	Computers&Accessories|Accessories&Peripherals|...	₹199	₹349	43
2	B096MSW6CT	Sounce Fast Phone Charging Cable & Data Sync U...	Computers&Accessories|Accessories&Peripherals|...	₹199	₹1,899	90
3	B08HDJ86NZ	boAt Deuce USB 300 2 in 1 Type-C & Micro USB S...	Computers&Accessories|Accessories&Peripherals|...	₹329	₹699	53
4	B08CF3B7N1	Portronics Konnect L 1.2M Fast Charging 3A 8 P...	Computers&Accessories|Accessories&Peripherals|...	₹154	₹399	61

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1465 entries, 0 to 1464
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   product_id           1465 non-null   object
 1   product_name         1465 non-null   object
 2   category             1465 non-null   object
 3   discounted_price     1465 non-null   object
 4   actual_price         1465 non-null   object
 5   discount_percentage  1465 non-null   object
 6   rating               1465 non-null   object
 7   rating_count         1463 non-null   object
 8   about_product        1465 non-null   object
 9   user_id              1465 non-null   object
 10  user_name            1465 non-null   object
 11  review_id            1465 non-null   object
 12  review_title         1465 non-null   object
 13  review_content       1465 non-null   object
 14  img_link             1465 non-null   object
 15  product_link         1465 non-null   object
dtypes: object(16)
memory usage: 183.3+ KB
```

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

Out[8]:

	product_id	product_name	category	discounted_price	actual_price	discount_percentage	rating	rating_count	about_produc
0	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	.
1460	False	False	False	False	False	False	False	False	Fals
1461	False	False	False	False	False	False	False	False	Fals
1462	False	False	False	False	False	False	False	False	Fals
1463	False	False	False	False	False	False	False	False	Fals
1464	False	False	False	False	False	False	False	False	Fals

1465 rows × 16 columns

In [9]:
```
df.describe()
```

Out[9]:

	product_id	product_name	category	discounted_price	actual_price	discount_per
count	1465	1465	1465	1465	1465	
unique	1351	1337	211	550	449	
top	B08WRWPM22	Fire-Boltt Ninja Call Pro Plus 1.83" Smart Wat...	Computers&Accessories|Accessories&Peripherals|...	₹199	₹999	
freq	3	5	233	53	120	

In [11]:
```
df.shape
```

Out[11]:
```
(1465, 16)
```

In [12]:
```
df.columns.tolist()
```

Out[12]:
```
['product_id',
 'product_name',
 'category',
 'discounted_price',
 'actual_price',
 'discount_percentage',
 'rating',
 'rating_count',
 'about_product',
 'user_id',
 'user_name',
 'review_id',
 'review_title',
 'review_content',
 'img_link',
 'product_link']
```

In [13]:
```
df.isna().sum()
```

Out[13]:
```
product_id            0
product_name          0
category              0
discounted_price      0
actual_price          0
discount_percentage   0
rating                0
rating_count          2
about_product         0
user_id               0
user_name             0
review_id             0
review_title          0
review_content        0
img_link              0
product_link          0
dtype: int64
```

In [18]:
```
import re
def to_numeric(s, extract_number=False):
    s = s.astype(str).str.strip()
    if extract_number:
        s = s.str.extract(r'([0-9]+(?:\.[0-9]+)?)', expand=False)
    s = s.str.replace(',', '', regex=True).str.replace('%','', regex=True)
    return pd.to_numeric(s, errors='coerce')

df['discounted_price_clean'] = to_numeric(df['discounted_price'], extract_number=True)
df['discount_percentage_clean'] = to_numeric(df['discount_percentage'], extract_number=True)
df['rating_clean'] = to_numeric(df['rating'], extract_number=True)
df['rating_count_clean'] = to_numeric(df['rating_count'], extract_number=True)
```

In [15]:
```
df[['discounted_price_clean','discount_percentage_clean','rating_clean','rating_count_clean']].describe().T
```

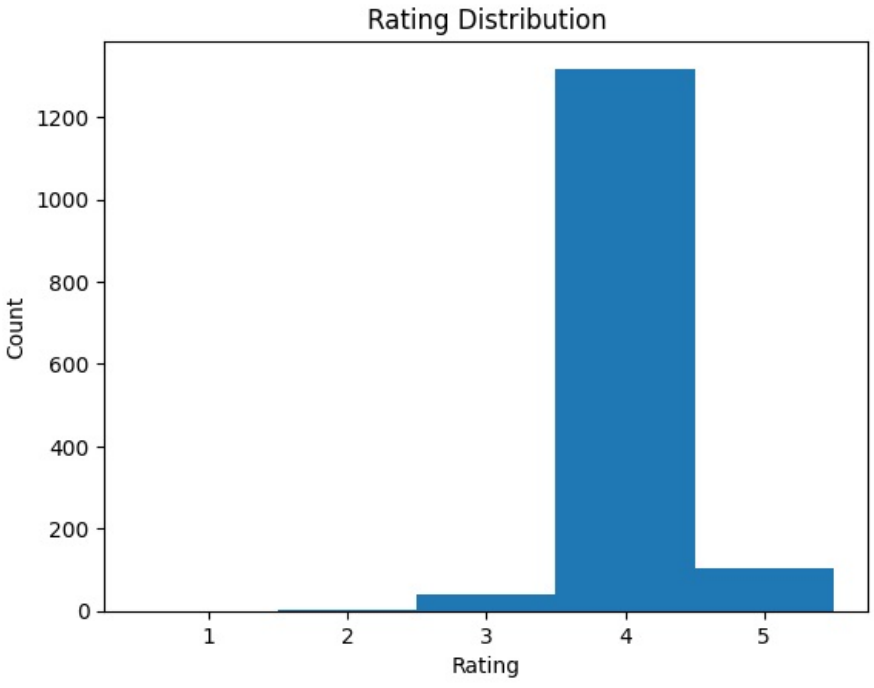
Out[15]:

	count	mean	std	min	25%	50%	75%	max
discounted_price_clean	1465.0	241.624355	282.666599	1.0	2.0	150.0	398.0	999.0
discount_percentage_clean	1465.0	47.691468	21.635905	0.0	32.0	50.0	63.0	94.0
rating_clean	1464.0	4.096585	0.291674	2.0	4.0	4.1	4.3	5.0
rating_count_clean	1463.0	88.758715	188.554088	1.0	3.0	12.0	44.5	992.0

In [21]:
```
top_products = df.groupby('product_name').agg(
    reviews_count=('review_id','count'),
    avg_rating=('rating_clean','mean'),
    avg_price=('discounted_price_clean','mean'),
    max_rating_count=('rating_count_clean','max')
).sort_values('reviews_count', ascending=False).reset_index()

category_summary = df.groupby('category').agg(
    products_count=('product_id','nunique'),
    reviews_count=('review_id','count'),
    avg_rating=('rating_clean','mean'),
    avg_price=('discounted_price_clean','mean')
).sort_values('reviews_count', ascending=False).reset_index()
```

In [22]:
```
#Ratings
```

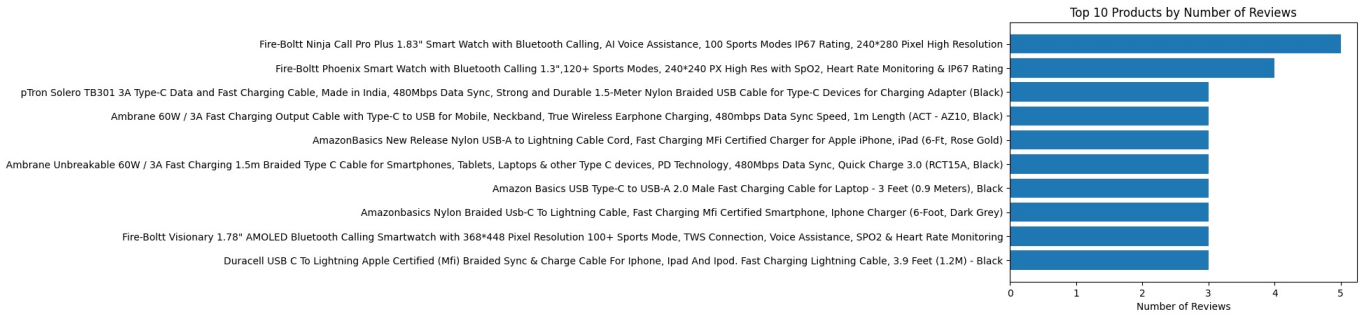
In [44]:
```
import matplotlib.pyplot as plt
plt.figure()
plt.hist(df['rating_clean'].dropna(), bins=[0.5,1.5,2.5,3.5,4.5,5.5])
plt.title('Rating Distribution')
plt.xlabel('Rating'); plt.ylabel('Count')
plt.show()
```



In [23]:
```
#Top 10 products by reviews
```

In [24]:
```
top10 = top_products.head(10).iloc[::-1]
plt.figure()
plt.barh(top10['product_name'], top10['reviews_count'])
plt.title('Top 10 Products by Number of Reviews')
plt.xlabel('Number of Reviews')
```

```
plt.show()
```

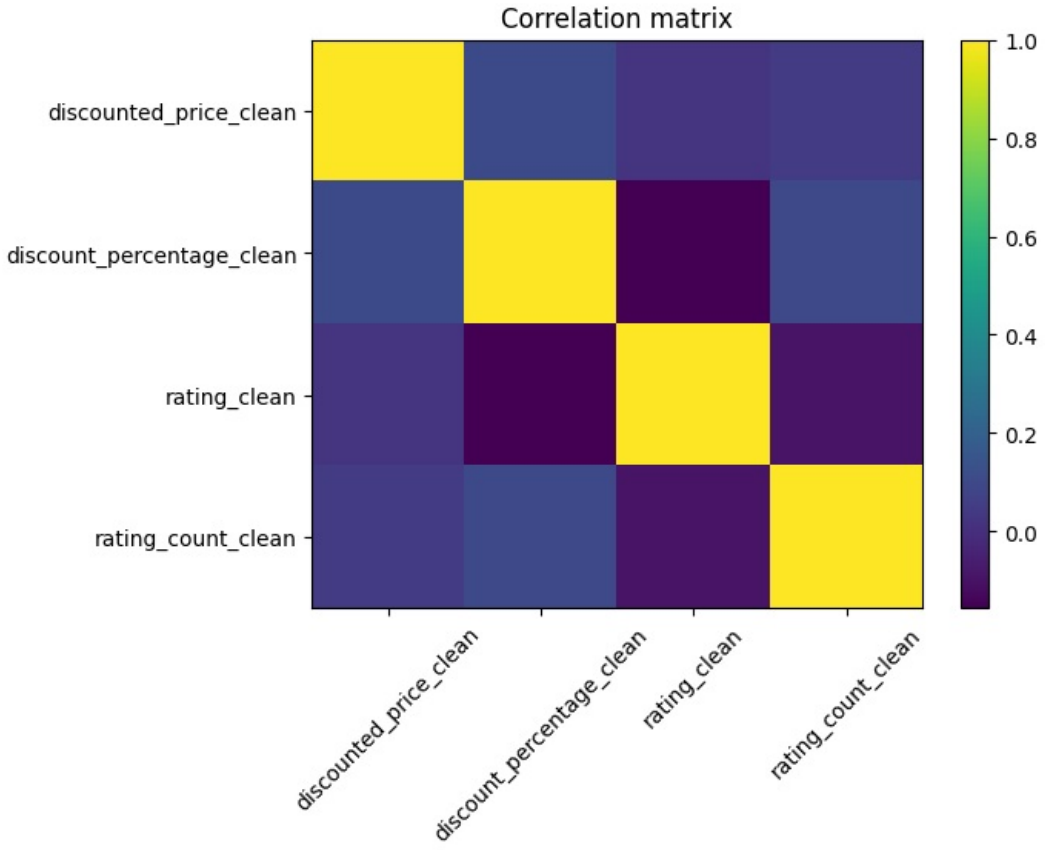


In [26]:
```
#Discount vs Rating scatter
```

In [28]:
```
plt.figure()
s = df['rating_count_clean'].fillna(0)
s_scaled = (s - s.min()) / (s.max() - s.min() + 1e-9) * 200 + 10
plt.scatter(df['discount_percentage_clean'], df['rating_clean'], s=s_scaled)
plt.title('Discount % vs Rating ')
plt.xlabel('Discount %'); plt.ylabel('Rating')
plt.show()
```

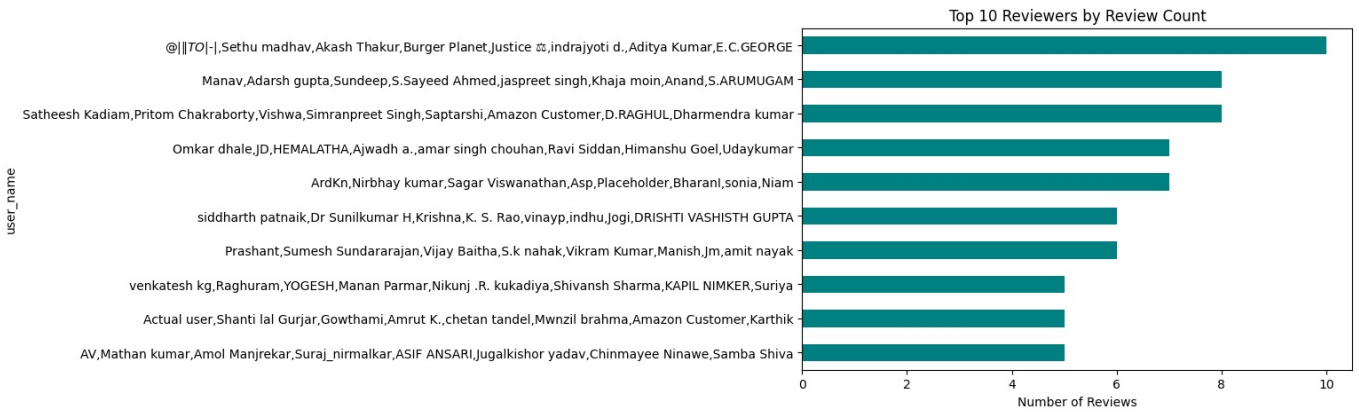
In [34]:
```
#Correlation matrix (using imshow):
```

In [35]:
```
num_df = df[['discounted_price_clean','discount_percentage_clean','rating_clean','rating_count_clean']].dropna(
corr = num_df.corr()
plt.figure()
plt.imshow(corr, interpolation='nearest', aspect='auto')
plt.colorbar()
plt.xticks(range(len(corr)), corr.columns, rotation=45)
plt.yticks(range(len(corr)), corr.columns)
plt.title('Correlation matrix')
plt.show()
```



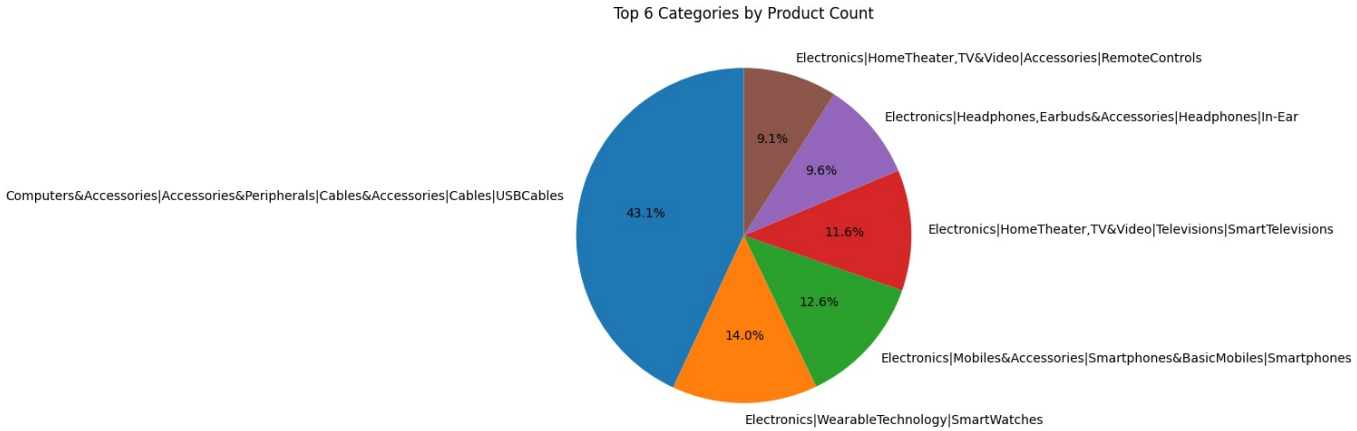
In [36]:
```
#Top words in review_content
```

In [38]:
```
top_reviewers = df['user_name'].value_counts().head(10)

plt.figure(figsize=(8,5))
top_reviewers.sort_values().plot(kind='barh', color='teal')
plt.title("Top 10 Reviewers by Review Count")
plt.xlabel("Number of Reviews")
plt.show()
```



In [46]:
```
#Distribution of products by category
```

In [47]:
```
category_counts = df['category'].value_counts().head(6)

plt.figure(figsize=(6,6))
plt.pie(category_counts, labels=category_counts.index, autopct='%1.1f%%', startangle=90)
plt.title("Top 6 Categories by Product Count")
plt.show()
```


Top 6 Categories by Product Count
Electronics|HomeTheater,TV&Video|Accessories|RemoteControls
Electronics|Headphones,Earbuds&Accessories|Headphones|In-Ear
Computers&Accessories|Accessories&Peripherals|Cables&Accessories|Cables|USBCables
Electronics|HomeTheater,TV&Video|Televisions|SmartTelevisions
Electronics|Mobiles&Accessories|Smartphones&BasicMobiles|Smartphones
Electronics|WearableTechnology|SmartWatches
9.1%
9.6%
43.1%
11.6%
12.6%
14.0%

In [ ]: