NAME : OJASWA SAHARE

ROLL NO : CS4-18

PRN : 202401040182

```
# 1. Find the top 5 most reviewed products
df.sort_values(by='review_count', ascending=False).head(5)

# 2. Calculate the average rating of products in each category
df.groupby('category')['rating'].mean()

# 3. Find the product with the highest discount
df.loc[df['discount'].idxmax()]

# 4. Determine how many products are priced above Rs.1000
(df['price'] > 1000).sum()

# 5. Identify the brand with the most number of products
df['brand'].value_counts().idxmax()

# 6. List all products that are out of stock
df[df['availability'] == 'Out of Stock']

# 7. Calculate the median price of products per brand
df.groupby('brand')['price'].median()

# 8. Check if there is a correlation between price and rating
df[['price', 'rating']].corr()
```

```
# 9. Find the average discount per category
df.groupby('category')['discount'].mean()

# 10. Determine the total number of unique categories
df['category'].nunique()

# 11. Get the average rating of products added in the year 2024
df['date_added'] = pd.to_datetime(df['date_added'])
df[df['date_added'].dt.year == 2024]['rating'].mean()

# 12. Find the most expensive product in each category
df.loc[df.groupby('category')['price'].idxmax()]

# 13. Identify products with more than 1000 reviews and rating above 4.5
df[(df['review_count'] > 1000) & (df['rating'] > 4.5)]

# 14. Count the number of products for each brand with average rating below 3
df[df['rating'] < 3].groupby('brand').size()

# 15. Calculate standard deviation of price per category
df.groupby('category')['price'].std()

# 16. Filter products that have a discount more than 50% and are available
df[(df['discount'] > 50) & (df['availability'] == 'In Stock')]

# 17. List top 5 categories with highest average review count
df.groupby('category')['review_count'].mean().sort_values(ascending=False).head(5)

# 18. Create a new column 'effective_price' = price - discount
```

```
df['effective_price'] = df['price'] - df['discount']

# 19. Find how many products were added in each month of 2024
df_2024 = df[df['date_added'].dt.year == 2024]
df_2024['date_added'].dt.month.value_counts().sort_index()

# 20. Calculate the percentage of products in stock
(df['availability'] == 'In Stock').mean() * 100
```