

** DAY 18 - PROJECT **

[iPhone Sales Analysis]

```
In [109]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn-v0_8')

In [110]: df = pd.read_csv("apple_products.csv")
df.head(10)

Out[110]:   Product Name      Product URL Brand Sale Price MRP Discount Percentage Number Of Ratings Number Of Reviews UPC Star Rating Ram
0  APPLE iPhone 8 Plus (Gold, 256 GB) https://www.flipkart.com/apple-iphone-8-plus-g... Apple 49900 49900 0 3431 356 MOBEXRGV7EHHTGUH 4.6 2 GB
1  APPLE iPhone 8 Plus (Space Grey, 256 GB) https://www.flipkart.com/apple-iphone-8-plus-s... Apple 84900 84900 0 3431 356 MOBEXRGVAVC87JT4F 4.6 2 GB
2  APPLE iPhone 8 Plus (Silver, 256 GB) https://www.flipkart.com/apple-iphone-8-plus-s... Apple 84900 84900 0 3431 356 MOBEXRGVGETABXWZ 4.6 2 GB
3  APPLE iPhone 8 (Silver, 256 GB) https://www.flipkart.com/apple-iphone-8-silver... Apple 77000 77000 0 11202 794 MOBEXRGVMZWUHCBA 4.5 2 GB
4  APPLE iPhone 8 (Gold, 256 GB) https://www.flipkart.com/apple-iphone-8-gold-2... Apple 77000 77000 0 11202 794 MOBEXRGVPK7PFEJZ 4.5 2 GB

In [111]: df.columns = [
    .str.strip()
    .str.lower()
    .str.replace(" ", "_")
]
df.columns

Out[111]: Index(['product_name', 'product_url', 'brand', 'sale_price', 'mrp',
       'discount_percentage', 'number_of_ratings', 'number_of_reviews',
       'upc', 'star_rating', 'ram'],
       dtype='object')

In [112]: num_cols = [
    "sale_price", "mrp", "discount_percentage",
    "number_of_ratings", "number_of_reviews", "star_rating"
]

df[num_cols] = df[num_cols].apply(pd.to_numeric, errors='coerce')

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 11 columns):
 # Column          Non-Null Count  Dtype  
--- 
 0   product_name   62 non-null     object  
 1   product_url    62 non-null     object  
 2   brand           62 non-null     object  
 3   sale_price     62 non-null     int64  
 4   mrp             62 non-null     int64  
 5   discount_percentag... 62 non-null     int64  
 6   number_of_ratings 62 non-null     int64  
 7   number_of_reviews 62 non-null     int64  
 8   upc             62 non-null     object  
 9   star_rating     62 non-null     float64 
 10  ram             62 non-null     object  
dtypes: float64(1), int64(5), object(5)
memory usage: 5.5+ KB
```

Q1: Top 10 Highest-Rated iPhones on Flipkart (India)

```
In [113]: df.sort_values(by="star_rating", ascending=False).head(10)[
    ["product_name", "star_rating", "sale_price"]]
```

```
Out[113]:   product_name  star_rating  sale_price
20  APPLE iPhone 11 Pro Max (Midnight Green, 64 GB)        4.7  117100
17  APPLE iPhone 11 Pro Max (Space Grey, 64 GB)        4.7  117100
16  APPLE iPhone 11 Pro Max (Midnight Green, 256 GB)       4.7  131900
15  APPLE iPhone 11 Pro Max (Gold, 64 GB)        4.7  117100
14  APPLE iPhone 11 Pro Max (Gold, 256 GB)       4.7  131900
0   APPLE iPhone 8 Plus (Gold, 64 GB)        4.6  49900
29  APPLE iPhone 12 (White, 128 GB)        4.6  75900
32  APPLE iPhone 12 Pro Max (Graphite, 128 GB)      4.6  120900
35  APPLE iPhone 12 (Black, 128 GB)        4.6  75900
36  APPLE iPhone 12 (Blue, 128 GB)        4.6  75900
```

Q2: How Many Ratings Do the Highest-Rated iPhones Have?

```
In [114]: df.sort_values(by="star_rating", ascending=False).head(10)[
    ["product_name", "star_rating", "number_of_ratings"]]
```

```
Out[114]:   product_name  star_rating  number_of_ratings
20  APPLE iPhone 11 Pro Max (Midnight Green, 64 GB)        4.7  1078
17  APPLE iPhone 11 Pro Max (Space Grey, 64 GB)        4.7  1078
16  APPLE iPhone 11 Pro Max (Midnight Green, 256 GB)       4.7  1078
15  APPLE iPhone 11 Pro Max (Gold, 64 GB)        4.7  1078
14  APPLE iPhone 11 Pro Max (Gold, 256 GB)       4.7  1078
0   APPLE iPhone 8 Plus (Gold, 64 GB)        4.6  3431
29  APPLE iPhone 12 (White, 128 GB)        4.6  2101
32  APPLE iPhone 12 Pro Max (Graphite, 128 GB)      4.6  580
35  APPLE iPhone 12 (Black, 128 GB)        4.6  2101
36  APPLE iPhone 12 (Blue, 128 GB)        4.6  2101
```

Q3: iPhone With Highest Reviews

```
In [115]: df.sort_values(by="number_of_reviews", ascending=False).head(1)[
    ["product_name", "number_of_reviews", "star_rating", "sale_price"]]
```

```
Out[115]:   product_name  number_of_reviews  star_rating  sale_price
23  Apple iPhone SE (White, 256 GB) (Includes EarP... 8161 4.5  44999
```

Q4: Sale Price vs Number of Ratings

```
In [116]: sns.scatterplot(data=df, x="sale_price", y="number_of_ratings")
```

```
plt.title("Sale Price vs Number of Ratings")
```

```
plt.show()
```

```
df[["sale_price", "number_of_ratings"]].corr()
```



```
Out[116]:   sale_price  number_of_ratings
sale_price  1.000000 -0.701526
number_of_ratings  -0.701526  1.000000
```

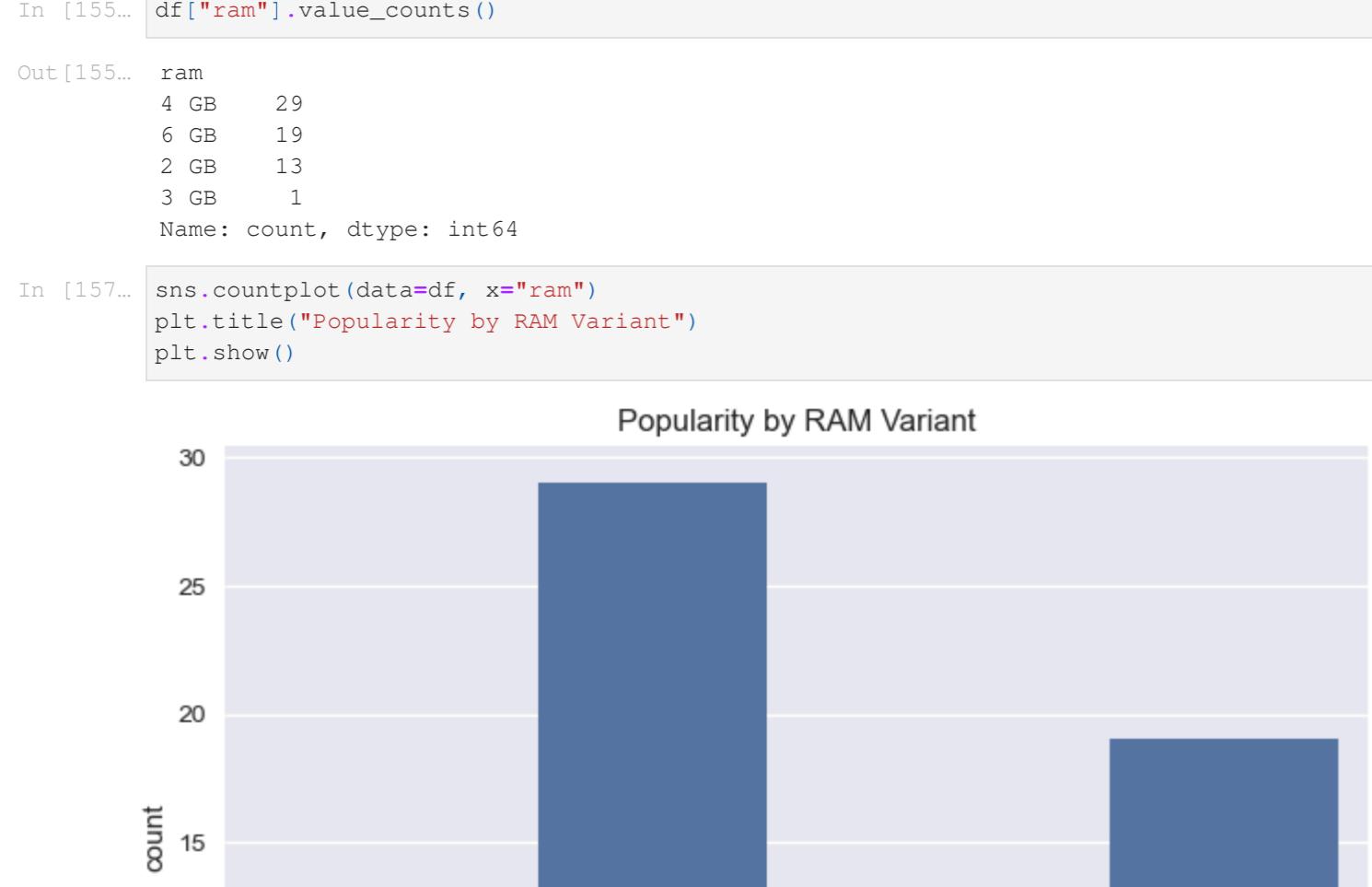
Q5: Discount % vs Number of Ratings

```
In [117]: sns.scatterplot(data=df, x="discount_percentage", y="number_of_ratings")
```

```
plt.title("Discount % vs Number of Ratings")
```

```
plt.show()
```

```
df[["discount_percentage", "number_of_ratings"]].corr()
```



```
Out[117]:   discount_percentage  number_of_ratings
discount_percentage  1.000000  0.684827
number_of_ratings  0.684827  1.000000
```

Q6: Least & Most Expensive iPhones

```
In [118]: least_expensive = df.sort_values(by="sale_price").head(1)
```

```
most_expensive = df.sort_values(by="sale_price", ascending=False).head(1)
```

```
least_expensive
```

```
Out[118]:   product_name  product_url  brand  sale_price  MRP  discount_percentage  number_of_ratings  number_of_reviews  UPC  star_rating  RAM
57  APPLE iPhone SE (Black, 64 GB) https://www.flipkart.com/apple-iphone-se-black... Apple 29999 39900 24 95909 8161 MOBFWQ6BR3MK7AUG 4.5 4 GB
```

```
In [119]: most_expensive
```

```
Out[119]:   product_name  product_url  brand  sale_price  MRP  discount_percentage  number_of_ratings  number_of_reviews  UPC  star_rating  RAM
41  APPLE iPhone 12 Pro (Pacific Blue, 512 GB) https://www.flipkart.com/apple-iphone-12-pro-pacif... Apple 140900 149900 6 545 42 MOBFWQBYZTHSXKMGW 4.5 4 GB
```

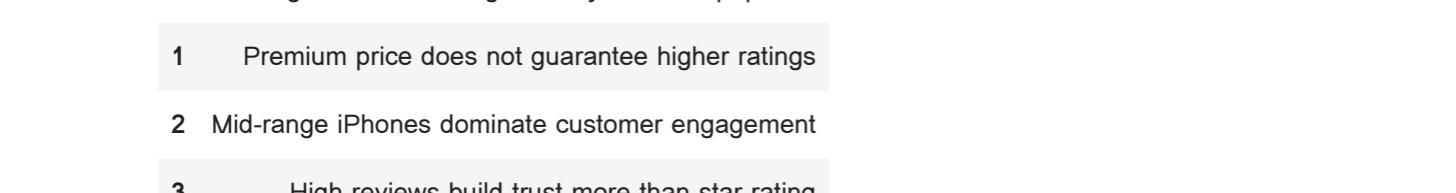
Q7: Rating vs Reviews

```
In [120]: sns.scatterplot(data=df, x="number_of_reviews", y="star_rating")
```

```
plt.title("Rating vs Reviews")
```

```
plt.show()
```

```
df[["number_of_reviews", "star_rating"]].corr()
```



```
Out[120]:   number_of_reviews  star_rating
number_of_reviews  1.000000  0.684827
star_rating  0.684827  1.000000
```

Q8: Price Category Analysis

```
In [121]: bins = [0, 40000, 70000, 200000]
labels = ["Budget", "Mid-Range", "Premium"]

df[["price_category"]] = pd.cut(df["sale_price"], bins=bins, labels=labels)
```

```
df[["price_category"]].value_counts()
```

```
Out[121]:   price_category
Premium      35
Mid-Range    21
Budget       6
Name: count, dtype: int64
```

```
In [122]: sns.countplot(data=df, x="price_category")
```

```
plt.title("iPhone Distribution by Price Category")
```

```
plt.show()
```



```
Out[122]:   price_category
Premium      35
Mid-Range    21
Budget       6
Name: count, dtype: int64
```

Q9: Average Rating by Price Category

```
In [123]: df.groupby("price_category", observed=True)["star_rating"].mean()
```

```
Out[123]:   price_category
Budget  4.516667
Mid-Range 4.566667
Premium  4.595429
Name: star_rating, dtype: float64
```

```
df[["star_rating", "price_category"]].corr()
```

```
Out[123]:   star_rating  price_category
star_rating  1.000000  0.684827
price_category  0.684827  1.000000
```

Q10: Discount Impact on Popularity

```
In [124]: discount_bins = pd.cut(df["discount_percentage"], bins=[0, 10, 20, 30, 50])
```

```
df.groupby(discount_bins, observed=True)[["number_of_ratings"]].mean()
```

```
Out[124]:   discount_percentage
0-10  7082.952381
10-20  31204.238095
20-30  81071.500000
Name: number_of_ratings, dtype: float64
```

```
In [125]: sns.countplot(data=df, x="discount_percentage")
```

```
plt.title("Phone Distribution by Discount %")
```

```
plt.show()
```



```
Out[124]:   discount_percentage
0-10  7082.952381
10-20  31204.238095
20-30  81071.500000
Name: number_of_ratings, dtype: float64
```

Q11: Best Value-for-Money iPhones

```
In [125]: df["value_score"] = df["star_rating"] * df["discount_percentage"] / df["sale_price"]
```

```
df.sort_values(by="value_score", ascending=False).head(10)[
    ["product_name", "sale_price", "star_rating", "discount_percentage"]]
```

```
Out[125]:   product_name  sale_price  star_rating  discount_percentage
52  APPLE iPhone SE (White, 64 GB) 29999 4.5  24
57  APPLE iPhone SE (Black, 64 GB) 29999 4.5  24
53  APPLE iPhone SE (Black, 128 GB) 34999 4.5  22
55  APPLE iPhone SE (Red, 128 GB) 34999 4.5  22
54  APPLE iPhone SE (White, 128 GB) 34999 4.5  22
11  Apple iPhone XR (Coral, 128 GB) (Includes EarP... 41999 4.6  20
12  Apple iPhone XR (Black, 128 GB) (Includes EarP... 41999 4.6  20
13  Apple iPhone XR (White, 128 GB) (Includes EarP... 41999 4.6  20
9  Apple iPhone XR ((PRODUCT)RED, 128 GB) (Includ... 41999 4.6  20
10  Apple iPhone XR (Black, 64 GB) (Includes EarP... 39999 4.6  16
```

Q12: RAM Variant Analysis

```
In [126]: df["ram"] = pd.value_counts()
```

```
Out[126]:   ram
4 GB  29
8 GB  19
2 GB  13
3 GB  1
Name: count, dtype: int64
```

```
In [127]: sns.countplot(data=df, x="ram")
```

```
plt.title("Popularity by RAM Variant")
```

```
plt.show()
```



```
Out[126]:   ram
4 GB  29
8 GB  19
2 GB  13
3 GB  1
Name: count, dtype: int64
```

Q13: Outlier Detection (Price)

```
In [128]: sns.boxplot(x=df["sale_price"])
plt.title("Sale Price Outliers")
plt.show()
```



```
Out[127]:   sale_price
4 GB  29
8 GB  19
2 GB  13
3 GB  1
Name: count, dtype: int64
```

Q14: Rating Bands Analysis

```
In [129]: df["rating_band"] = pd.cut(df["star_rating"],
```

```
bins=[0, 4.0, 4.5, 4.0, 4.5, 4.5, 5.0],
```

```
labels=["Below 4.0", "4.0-4.5", "4.5-4.5", "4.5+"],
```

```
include_lowest=True
```

```
df[["rating_band"]].value_counts()
```

```
Out[129]:   rating_band
Below 4.0  2
4.0-4.5  20
4.5-4.5  0
4.5+  1
Name: count, dtype: int64
```

Q15: Final Business Insights (Interview Cell)

```
In [130]: pd.DataFrame([
    {"insight": 1, "description": "Higher discounts significantly increase popularity."}
])
```

```
Out[130]:   insight
1  Higher discounts significantly increase popularity.
```

```
2  Premium price does not guarantee higher ratings.
```

```
3  Mid-range iPhones dominate customer engagement.
```

```
4  High reviews build trust more than star rating.
```

```
5  Value-for-money models drive maximum sales."
```

```
])
```

```
Out[130]:   insight
1  Higher discounts significantly increase popularity.
```

```
2  Premium price does not guarantee higher ratings.
```

```
3  Mid-range iPhones dominate customer engagement.
```

```
4  High reviews build trust more than star rating.
```