Questions for EDA

Name:Vaibhav Saran

UB ID:50615031

Q1: **What is the relationship between RAM size and laptop prices? Does this relationship hold consistent across different brands and platforms (Amazon, Flipkart, BestBuy)?**

Why is this question significant to the objective?

Understanding the relationship between RAM size and laptop prices is critical for both **sellers** and **customers**. RAM is one of the most important specifications that directly impacts a laptop's performance, and customers often prioritize this feature when making purchasing decisions. By analyzing how RAM size influences prices across different brands and platforms, the following objectives are addressed:

- **For sellers**: This insight helps sellers determine **optimal pricing strategies** by understanding how competitors across platforms (Amazon, Flipkart, BestBuy) price laptops with similar RAM capacities. If sellers understand that customers are willing to pay a premium for higher RAM sizes, they can adjust their pricing to meet demand or differentiate their product line accordingly. Additionally, it helps in identifying any pricing **outliers** or inconsistencies that may deter customers.
- **For customers**: This analysis provides potential buyers with valuable information about whether **higher RAM** truly justifies a higher price. Customers will be able to evaluate whether paying more for additional RAM gives them real value, helping them make better purchasing decisions.

Q2: **Which processor company (Intel, AMD, etc.) dominates the market across different price ranges?**

Why is this question significant to the objective?

Processor brand and quality are key factors that strongly influence both **performance** and **price**. Customers tend to associate certain processor companies (like Intel or Apple) with performance, reliability, and price categories. By analyzing processor company dominance across different price ranges, we address the project objectives:

- **For sellers**: This question helps sellers understand the **processor-brand positioning** in the market. If Intel dominates across most price ranges, sellers might decide to stock more Intel-based laptops. For companies like AMD or newer ARM-based processors (e.g., Qualcomm, MediaTek), understanding their **market penetration** in lower price brackets can inform whether sellers should diversify their offerings. Additionally, knowing that Apple caters more to high-end customers allows sellers to focus their high-end marketing strategies on Apple-based machines.

- **For customers**: Customers need to understand the **value for money** aspect when it comes to processors. If Intel is dominating a wide range of prices, buyers may have more confidence in purchasing Intel-based machines. Alternatively, if AMD is delivering strong performance at lower price points, it may sway customers towards purchasing AMD laptops. Understanding the relationship between processor company and price helps customers align their budget with their performance expectations.

Conclusion:

Both questions are deeply aligned with the overarching objectives of understanding factors influencing laptop pricing and customer interest. By focusing on **RAM and processor dominance**, these questions allow us to uncover crucial insights about **performance-to-price trade-offs** and **market dynamics**. This information will help sellers optimize their offerings and pricing strategies, while also assisting customers in making informed purchasing decisions based on their budget and preferred features. These questions also reveal how certain brands and platforms (Amazon, Flipkart, BestBuy) cater to different price-sensitive and performance-focused markets.

Name:Yeswanth Chitturi

UB ID:50591666

Question 1:

Which brand has more models in total and across platforms? What is the model with the highest price in each brand in total and across platforms?

- **Connection to Objectives:** This question aims to identify the brand with the most extensive range of offerings. Understanding which brands dominate the market in terms of model variety is crucial for recognizing consumer choice patterns. Additionally, identifying the highest-priced models for each brand helps to highlight premium offerings that might attract consumers looking for quality.

- **Informed Decision-Making:** Customers can use this information to identify brands that offer a wider variety of models, helping them find options that better match their preferences and requirements.

- **Understanding Value:** Knowing which models are priced highest can help customers evaluate whether a higher price correlates with additional features or quality, guiding them towards products that provide better value for their money.
- **Market Positioning:** Sellers can use insights about brand model variety to adjust their product lines or marketing strategies, ensuring they offer competitive choices that meet customer demand.
- **Pricing Strategy:** Understanding which models command higher prices can help sellers identify premium features that justify the price, aiding in product positioning and marketing efforts.
- **Significance:** A brand with a larger number of models is likely to cater to a diverse range of customer needs, enhancing its market presence and competitiveness. Identifying the highest-priced models allows both sellers and consumers to understand what drives pricing and the features associated with premium offerings, thus informing better purchasing and marketing strategies.

Question 2:

What are the varieties in screen sizes for all models across platforms? What are the average screen sizes for each brand? Which screen size has a better rating? What is the relation between RAM and storage with screen sizes?

- **Connection to Objectives:** This question set explores the physical characteristics of laptops, particularly focusing on screen sizes, which is a critical factor influencing consumer preference. By analyzing the varieties in screen sizes, calculating average sizes per brand, and examining ratings associated with different sizes, this analysis addresses key customer requirements and perceptions. Furthermore, exploring the relationship between RAM, storage, and screen sizes provides insights into how specifications impact consumer satisfaction and performance expectations.
- **Personal Preference:** Customers can identify the variety of screen sizes available, allowing them to select a model that fits their use case, whether it's for portability or a larger display for work and entertainment.
- **Quality Assurance:** Knowing which screen sizes have better ratings can help customers make informed choices about which laptops are likely to meet their needs and expectations for performance and usability.
- **Optimal Configuration:** Understanding the relationship between RAM, storage, and screen sizes can assist customers in selecting a configuration that maximizes performance based on their specific needs, such as gaming, content creation, or general use.
- **Product Development:** Insights into screen size preferences can guide sellers in designing and marketing laptops that cater to consumer needs, ensuring that they are meeting current trends in customer preferences.
- **Targeted Marketing:** By analyzing the relationship between RAM, storage, and screen sizes, sellers can create targeted marketing campaigns that highlight configurations that provide optimal performance for specific user demographics.

- **Inventory Management:** Understanding the variety and average screen sizes can aid sellers in managing inventory effectively, ensuring that they stock popular models that align with consumer demand.
- **Significance:** Screen size is often a significant consideration for consumers when purchasing laptops, impacting usability and overall experience. Understanding the variety and average sizes can guide manufacturers in product development, ensuring that offerings align with consumer demands. Additionally, examining the correlation between RAM, storage, and screen sizes can reveal insights into optimal configurations that meet user needs, thereby informing both product design and marketing strategies.

Conclusion

The questions posed are not only significant in leading to the project's objectives but also provide valuable insights that can bridge the gap between consumer needs and seller offerings. By addressing these questions, the analysis aims to yield actionable insights that can enhance pricing strategies, product development, and ultimately assist consumers in making informed purchasing decisions.

Name:Shaurya Mathur

UB ID:50611201

Question1

How do the prices of laptops with similar specifications vary across brands?

Why This Question Leads to Your Objectives:

This question directly addresses the core objective of your project: to understand pricing factors for laptops. By focusing on laptops with similar specifications (e.g., same RAM, storage, processor, screen size), you eliminate variables related to performance and hardware differences, allowing you to isolate the effect of brand on pricing.

Analyzing this question helps you identify whether certain brands charge a premium for similar specifications or if there are consistent pricing strategies that can be attributed to brand perception, customer loyalty, or even marketing approaches. It also reveals how brands position themselves in the market relative to their competitors, which is crucial for understanding the broader pricing landscape in the laptop market.

Why Is It a Significant Question?

- **Consumer Decision-Making:** This question provides valuable insights for consumers looking for the best value for their money. If two laptops with the same specifications are priced significantly differently across brands, consumers can better evaluate which brand offers the best deal.
- **Brand Premium and Market Positioning:** Brands often justify higher prices through brand reputation, build quality, customer support, or additional features not listed in the core specs. By answering this question, we can evaluate if certain brands are charging a premium and whether it’s justified, providing a clearer picture of brand value.

- **Retail and Brand Strategies:** For retailers and brands, understanding how competitors price laptops with similar specifications is crucial for setting competitive prices. This analysis can help brands adjust their pricing strategy to better position themselves in the market and increase sales.
- **Recommendation Engine Optimization:** The insights from this question are fundamental to building a recommendation engine. If some brands consistently offer better value for the same specifications, the engine can factor that into its suggestions, ensuring consumers get the best deal for their preferences.

Question2

What are the major types of Storage? How does a product's prices vary with different types of storages?

Why This Question Leads to Your Objectives:

This question is key to understanding how specific laptop features, in this case, storage type, influence the pricing of laptops. Storage is a significant hardware component that impacts both the performance and price of a laptop. By analyzing this, you can achieve the objective of identifying which specifications contribute most to a laptop's price variation.

The question aims to explore the relationship between storage types (e.g., HDD, SSD, NVMe) and pricing trends across different brands and retail platforms. It leads you toward a deeper understanding of how storage influences overall laptop pricing, which is a crucial factor for both consumers and brands.

Why Is It a Significant Question?

- **Consumer Preferences:** For many consumers, storage type is a major consideration when purchasing a laptop. SSDs and NVMe drives, known for their speed, often command higher prices than traditional HDDs. This question helps consumers understand whether they should be paying more for specific storage types and if those prices vary significantly across brands or platforms.
- **Impact on Pricing:** Storage types have a direct impact on a laptop's price. For instance, SSDs are faster but more expensive than HDDs, while NVMe drives offer even faster speeds at a higher cost. Understanding how much of a price difference storage types cause is crucial for consumers looking for performance gains without overpaying. It also helps to identify if certain brands or platforms overprice laptops with higher storage options.
- **Market and Retail Strategy:** For brands and retailers, understanding how storage type influences pricing helps in product positioning and marketing. If certain storage options consistently drive up prices, brands can highlight those features in advertising or use them as a selling point for higher-end models. Retailers can use this data to guide promotions or recommendations for products with different storage types.
- **Product Segmentation:** Storage is often a feature that differentiates laptop models, even within the same series. By analyzing storage types and their impact on prices, the analysis will help segment laptops into different price ranges, catering to both budget-conscious and performance-focused customers.
- **Enhancing the Recommendation Engine:** Knowing how storage type affects prices will be essential for fine-tuning the recommendation engine. By understanding which

storage types offer better performance-to-price ratios, the engine can suggest laptops that provide the best balance between storage capacity, speed, and cost, ensuring more accurate recommendations for consumers.

Conclusion:

This question is significant because it helps dissect one of the most important specifications affecting laptop performance and pricing. By exploring the price differences between laptops with HDD, SSD, and NVMe storage, you will provide insights into the value for money each storage type offers, contributing to your broader goal of understanding the factors influencing laptop pricing and consumer interest.

1. Importing Libraries

```
import numpy as np
from IPython.display import display
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

2. Loading the Dataset

```
df = pd.read_csv(r"./data/laptrack.csv")
df.head()
```

```
  Laptop_Brand  Laptop_Name Processor_Company Operating_System
Processor  \
0   ZHAOHUIXIN       PC1068          Alwinner          Android    1.8
GHz a13
1          TPV      AceBook             Intel   Windows 11 Pro
Core i5
2           HP     Elitebook             Intel   Windows 11 Pro   Intel
Core i7
3        Apple  MacBook Air             Apple           Mac OS
Apple M3
4        Apple  MacBook Air             Apple           Mac OS
Apple M3

   Number_of_Reviews   Price Storage_Type Storage Rating  Screen_Size
RAM  \
0                  1  119.99         EMMC      64    4.5         10.1
2
1                 13  309.99          SSD     512    4.5         15.6
16
2                  5    1079          SSD    2048      4         16.0
```

```
32
3                  0     929         SSD      256       4          13.6
8
4                  0    1449         SSD      512       4          15.3
16

   Source
0  Amazon
1  Amazon
2  Amazon
3  Amazon
4  Amazon
```

3. Exploratory Data Analysis

3.1 Exploring the Meta Data information

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4838 entries, 0 to 4837
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Laptop_Brand       4838 non-null   object
 1   Laptop_Name        4838 non-null   object
 2   Processor_Company  4838 non-null   object
 3   Operating_System   4838 non-null   object
 4   Processor          4838 non-null   object
 5   Number_of_Reviews  4838 non-null   int64
 6   Price              4838 non-null   object
 7   Storage_Type       4838 non-null   object
 8   Storage            4838 non-null   object
 9   Rating             4836 non-null   object
 10  Screen_Size        4838 non-null   float64
 11  RAM                4838 non-null   int64
 12  Source             4838 non-null   object
dtypes: float64(1), int64(2), object(10)
memory usage: 491.5+ KB
```

Observation

- There are only 2 null values in rating, which will be handled later.
- The data type for the column: `Rating`, `Storage`, `Price` are in object which needs to be changed to numbers(or appropriate datatype).

```
# Strip leading and trailing spaces from each category in
'Processor_Company'
```

```python
df['Processor_Company'] = df['Processor_Company'].str.strip()

# Display the updated DataFrame to verify
print(df['Processor_Company'].unique())
```

```
['Alwinner' 'Intel' 'Apple' 'AMD' 'MediaTek' 'ARM' 'Qualcomm'
'Chromebook'
 'Snapdragon' 'AMD Ryzen' 'No Info' 'AMD\xa0Ryzen']
```

```python
# Replace 'AMD\xa0Ryzen' with 'AMD Ryzen'
df['Processor_Company'] = df['Processor_Company'].str.replace('\xa0',
' ', regex=False)

# Display the updated DataFrame to verify the changes
print(df['Processor_Company'].unique())
```

```
['Alwinner' 'Intel' 'Apple' 'AMD' 'MediaTek' 'ARM' 'Qualcomm'
'Chromebook'
 'Snapdragon' 'AMD Ryzen' 'No Info']
```

```python
# Looking at the unique values in Rating Storage and Price to see if
there are any wrong values
cols = ["Rating","Storage","Price"]

for feature in cols:
    print("Unique values in",feature,":",df[feature].unique())
    print()
```

```
Unique values in Rating : ['4.5' '4' '4.3' '4.4' '4.6' '4.1' '4.2'
'3.8' '3.4' '1' '5' '3' '3.9'
 '4.7' '3.5' '2.6' '3.7' '3.3' '3.6' '4.8' nan '0' '4.9' '0 Reviews'
'2'
 '2.7' '3.2' '2.5' '2.2' '2.8' '2.4' '1.7' '2.9' '2.3' '1.5']

Unique values in Storage : ['64' '512' '2048' '256' '128' '8192'
'4096' '1000' '32' '16' '4' '640'
 '250' '500' '192' '320' '2' '1048576' '16384' '576' '1024' '64eMMC'
 '128eMMC' '128GeMMC' '128UFS' '512PCIe' '256UFS' '32eMMC' '750']

Unique values in Price : ['119.99' '309.99' '1079' ... '3149.99'
'589.99' '4149.99']
```

```python
# Handling above values by extracting numbers
df['Storage'] = df['Storage'].str.extract('(\d+)', expand=False)

df['Rating'] = df['Rating'].str.replace(' Reviews', '', regex=False)
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')

df['Price'] = pd.to_numeric(df['Price'], errors='coerce')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4838 entries, 0 to 4837
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Laptop_Brand       4838 non-null   object 
 1   Laptop_Name        4838 non-null   object 
 2   Processor_Company  4838 non-null   object 
 3   Operating_System   4838 non-null   object 
 4   Processor          4838 non-null   object 
 5   Number_of_Reviews  4838 non-null   int64  
 6   Price              4817 non-null   float64
 7   Storage_Type       4838 non-null   object 
 8   Storage            4838 non-null   object 
 9   Rating             4836 non-null   float64
 10  Screen_Size        4838 non-null   float64
 11  RAM                4838 non-null   int64  
 12  Source             4838 non-null   object 
dtypes: float64(3), int64(2), object(8)
memory usage: 491.5+ KB
```

Observation

- All the columns are now in correct data format.
- Since the number of null values are very small the data will be dropped.

```
# Dropping Null values
df.dropna(inplace=True)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4815 entries, 0 to 4837
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Laptop_Brand       4815 non-null   object 
 1   Laptop_Name        4815 non-null   object 
 2   Processor_Company  4815 non-null   object 
 3   Operating_System   4815 non-null   object 
 4   Processor          4815 non-null   object 
 5   Number_of_Reviews  4815 non-null   int64  
 6   Price              4815 non-null   float64
 7   Storage_Type       4815 non-null   object 
 8   Storage            4815 non-null   object 
 9   Rating             4815 non-null   float64
 10  Screen_Size        4815 non-null   float64
 11  RAM                4815 non-null   int64  
```

```
 12  Source              4815 non-null   object
dtypes: float64(3), int64(2), object(8)
memory usage: 526.6+ KB
```

```
# Percent of data loss
loss_per = (4838 - 4815)/4838
loss_per*=100
print("The total data loss (in %) is:",loss_per)
```

```
The total data loss (in %) is: 0.475403059115337
```

- Now with the data all cleaned and ready lets proceed for analyzing the data

NOTE - FOR THIS PROJECT THE CURRENT EDA IS DONE AS PER PROJECT GUIDELINE BASIS ONLY

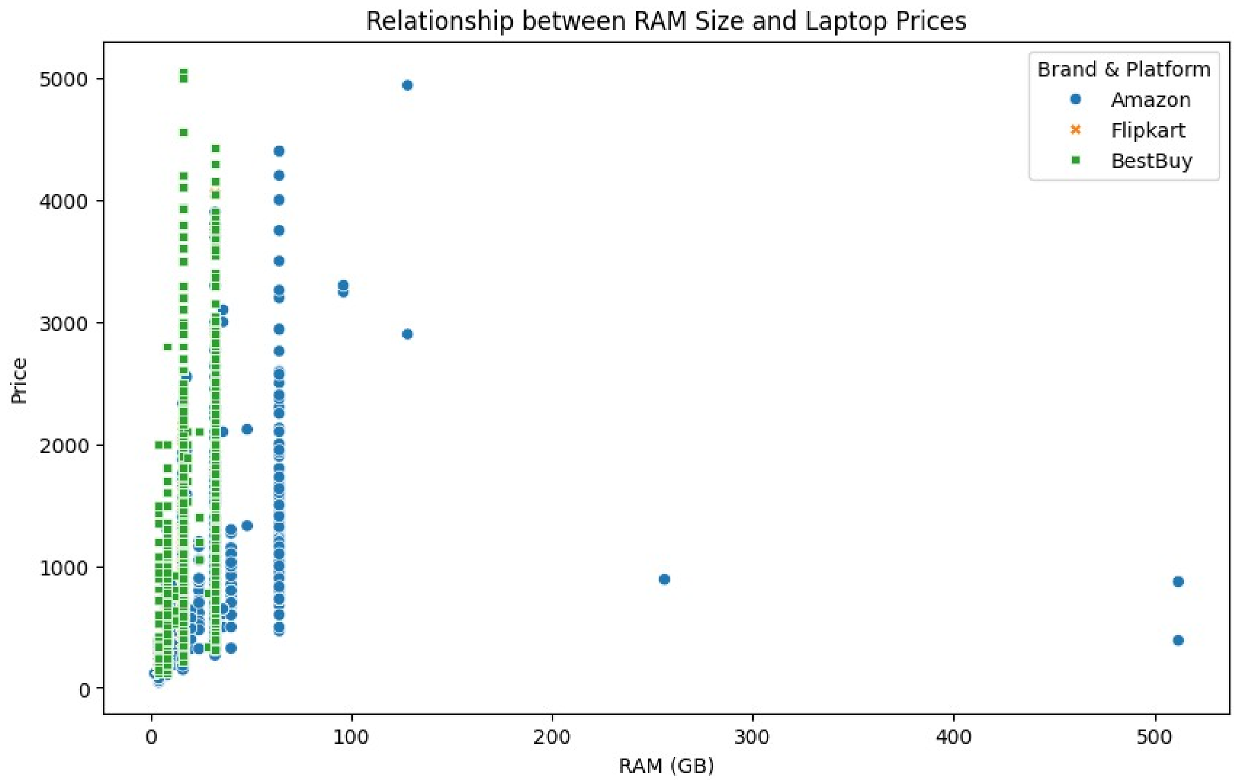
Name:Vaibhav Saran

UB ID:50615031

Analysis Statement

- What is the relationship between RAM size and laptop prices?
- Does this relationship hold consistent across different brands and platforms (Amazon, Flipkart, BestBuy)?

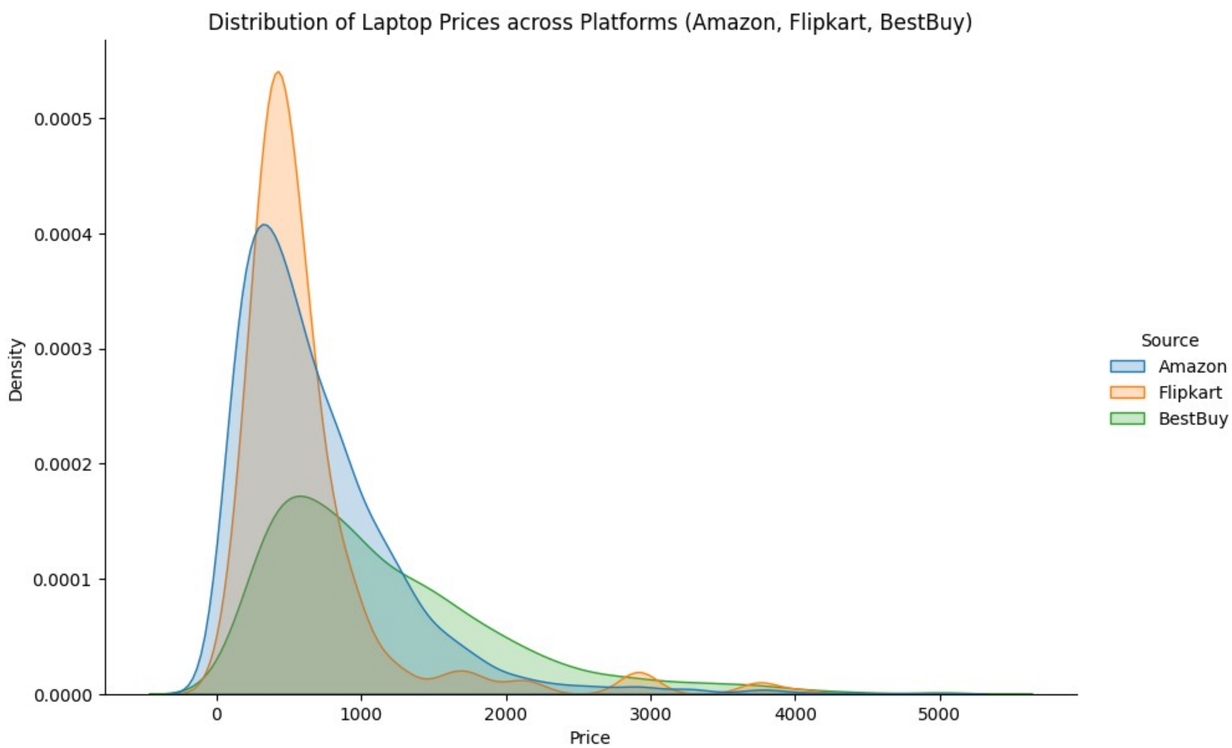
```
# Visualizing the data on scatter plot to identify relationship
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='RAM', y='Price',hue="Source",
style='Source')
plt.title('Relationship between RAM Size and Laptop Prices')
plt.xlabel('RAM (GB)')
plt.ylabel('Price')
plt.legend(title='Brand & Platform')
plt.show()
```



Observation

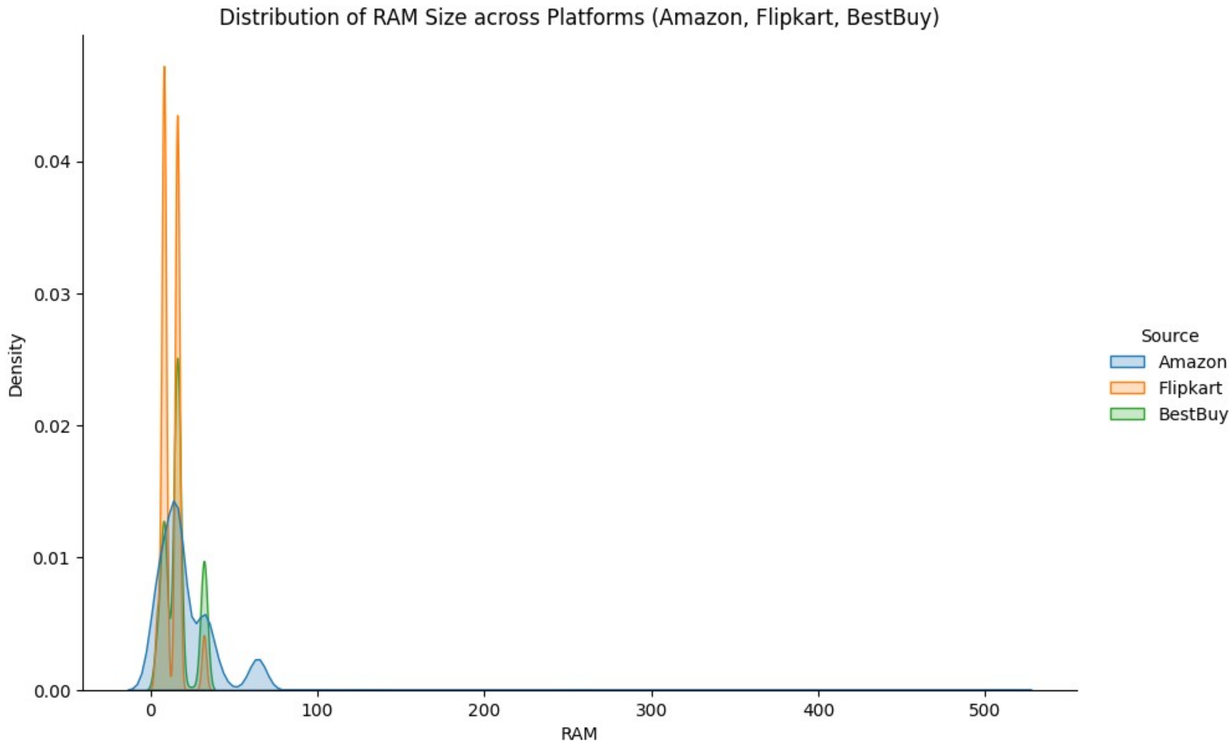
- The above plot shows that for a given RAM size, the price of laptop can vary greatly, i.e. the variance is high.
- There are some laptops which are acting as outliers where the for RAM 250GB the price is 1000 USD, and belong to Amazon.
- The outliers are not very significant and can be dealt with later during model building if it causes an issue.
- The variance observation is consistent across the sources as evident in above plot.
- For better clarity for the mentioned point, lets plot few more visualizations.

```
# Distribution of Price across Platforms
plt.figure(figsize=(10, 6))
sns.displot(data=df, x='Price', hue='Source', kind='kde', fill=True,
height=6, aspect=1.5)

plt.title('Distribution of Laptop Prices across Platforms (Amazon,
Flipkart, BestBuy)')
plt.xlabel('Price')
plt.ylabel('Density')

plt.show()

<Figure size 1000x600 with 0 Axes>
```




```
# Distribution of RAM across Platforms
plt.figure(figsize=(5, 6))
sns.displot(data=df, x='RAM', hue='Source', kind='kde', fill=True,
height=6, aspect=1.5)

plt.title('Distribution of RAM Size across Platforms (Amazon,
Flipkart, BestBuy)')
plt.xlabel('RAM')
plt.ylabel('Density')

plt.show()
```

```
<Figure size 500x600 with 0 Axes>
```



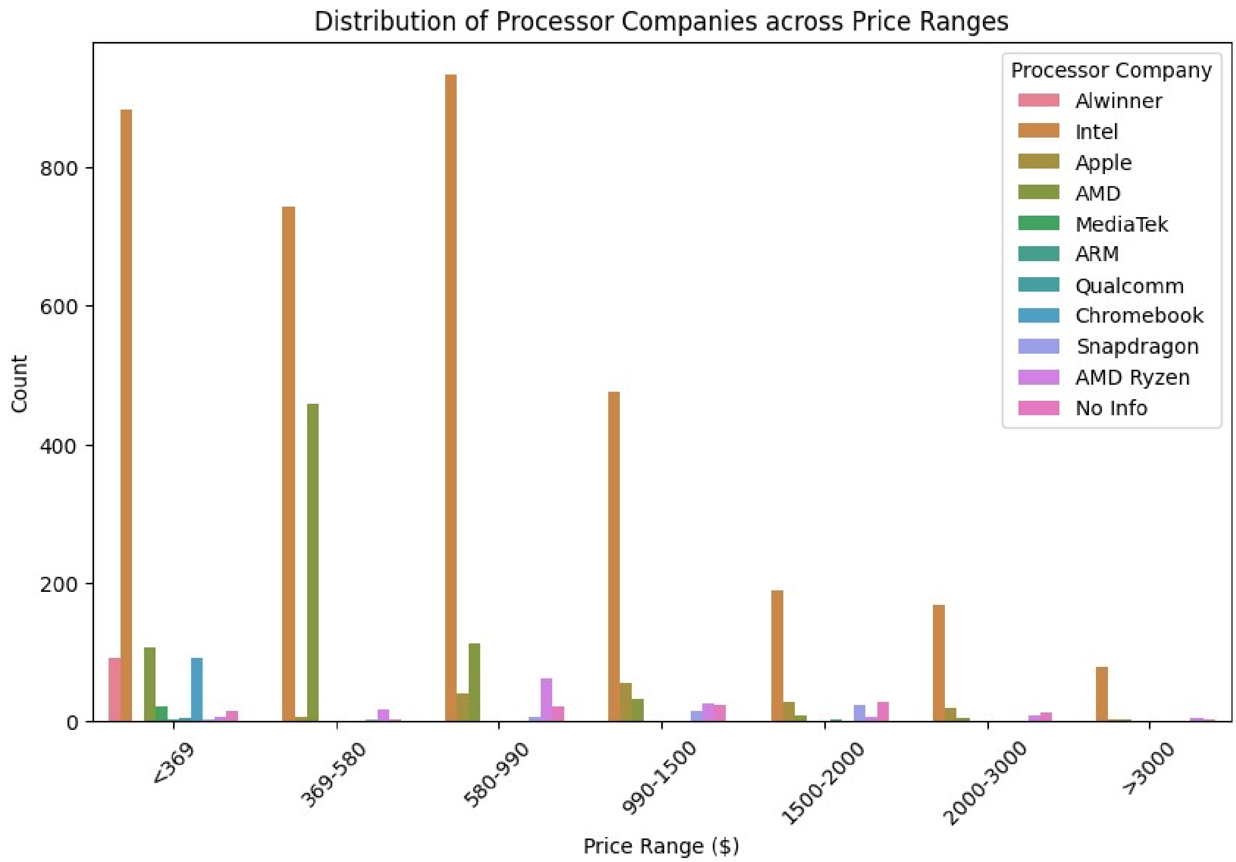
Observation

- The above distribution are consistent with our observatrions made above.

Analysis Statement

- Which processor company (Intel, AMD, etc.) dominates the market across different price ranges?

```
# Define price ranges (you can adjust the bins based on your data)
bins = [0, 370, 580, 990, 1500, 2000, 3000, df['Price'].max()]
labels = ['<369', '369-580', '580-990', '990-1500', '1500-2000',
'2000-3000', '>3000']

# Create a new column for price ranges
df['Price_Range'] = pd.cut(df['Price'], bins=bins, labels=labels)

# Plot the distribution of processor companies across price ranges
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Price_Range', hue='Processor_Company')
plt.title('Distribution of Processor Companies across Price Ranges')
plt.xlabel('Price Range ($)')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Processor Company')
plt.show()
```



Observation

- The above plot tells that acroos different price ranges, the market is dominated by `Intel` followed by `Apple`.
- Even though `Apple` is know for being a market player for costly machines, the number of `Intel` products in that price bracket is way more than `Apple`.
- There are other processor companies as well which are present in different price brackets but are not having enough number of products to compete against the market dominators, `Intel` and `Apple`.
- An interesting point is that `ARM based processor` companies like, `mediatek`, `snapdragon`, `qualcomm`, ettc. are all entering the market with low end laptops and Apple seems to be absent in lower price bracket of `369 USD`.

Name: Yeswanth Chitturi

UB ID:50591666

Analysis Statement

- Which brand has more no of models in total and across platforms? what is the model with highest price in each brand in total and across platforms?

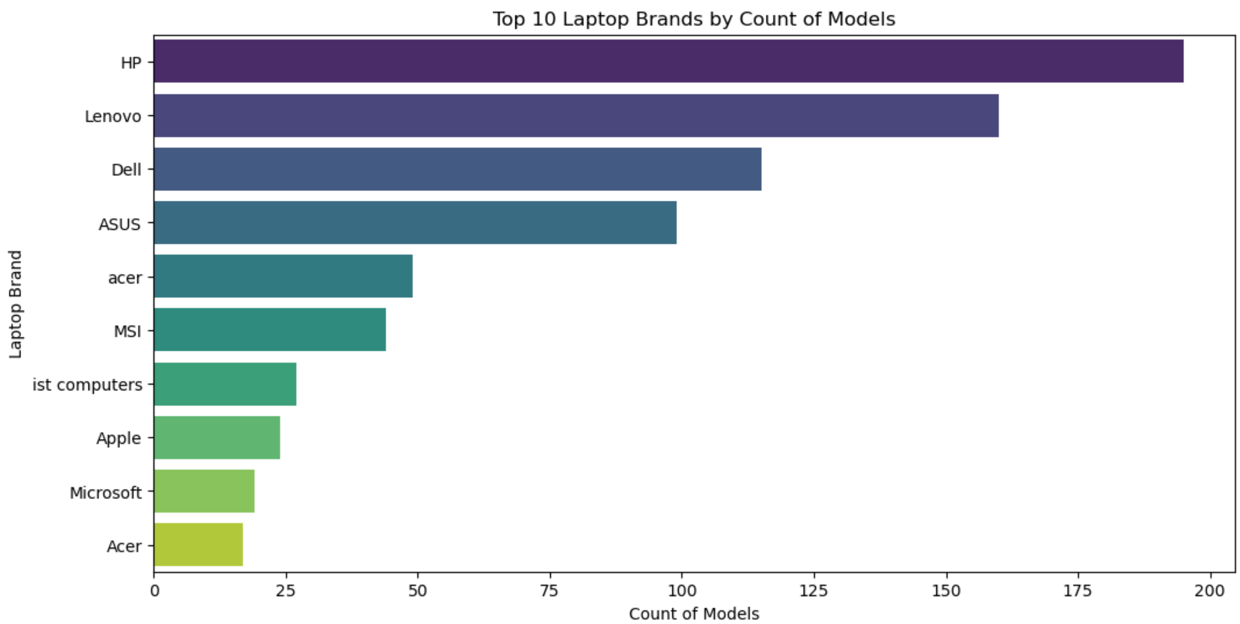
```
ydf=df.copy()
#Removing rows for easy hypothesis
ydf = ydf[ydf['Laptop_Name'] != 'No Model']
#All the unique models for each brand across platforms
unique_models = ydf[['Laptop_Brand', 'Laptop_Name', 'Source',
'Price']].drop_duplicates().sort_values(by='Laptop_Brand')
display(unique_models.head(5))
```

	Laptop_Brand	Laptop_Name	Source	Price
487	ACEMAGIC	AX16PRO	Amazon	759.54
99	ACEMAGIC	AX16PRO	Amazon	599.99
670	ACEMAGIC	AX16	Amazon	379.95
1292	ACEMAGIC	AX16	Amazon	379.98
162	ACEMAGIC	AX15	Amazon	359.98

```
# Grouping by Brand,Model
model_counts = ydf.groupby(['Laptop_Brand'])
['Laptop_Name'].nunique().reset_index()

# Count of models
model_counts.columns = ['Laptop_Brand', 'Count_of_Models']
model_counts=model_counts.sort_values(by='Count_of_Models',
ascending=False)

display(model_counts.head(5))
```

	Laptop_Brand	Count_of_Models
27	HP	195
40	Lenovo	160
16	Dell	115
3	ASUS	99
73	acer	49

```
top_10_models = model_counts.sort_values(by='Count_of_Models',
ascending=False).head(10)

plt.figure(figsize=(12, 6))

sns.barplot(x='Count_of_Models', y='Laptop_Brand', data=top_10_models,
palette='viridis')

plt.title('Top 10 Laptop Brands by Count of Models')
plt.xlabel('Count of Models')
plt.ylabel('Laptop Brand')

plt.show()
```



Observation

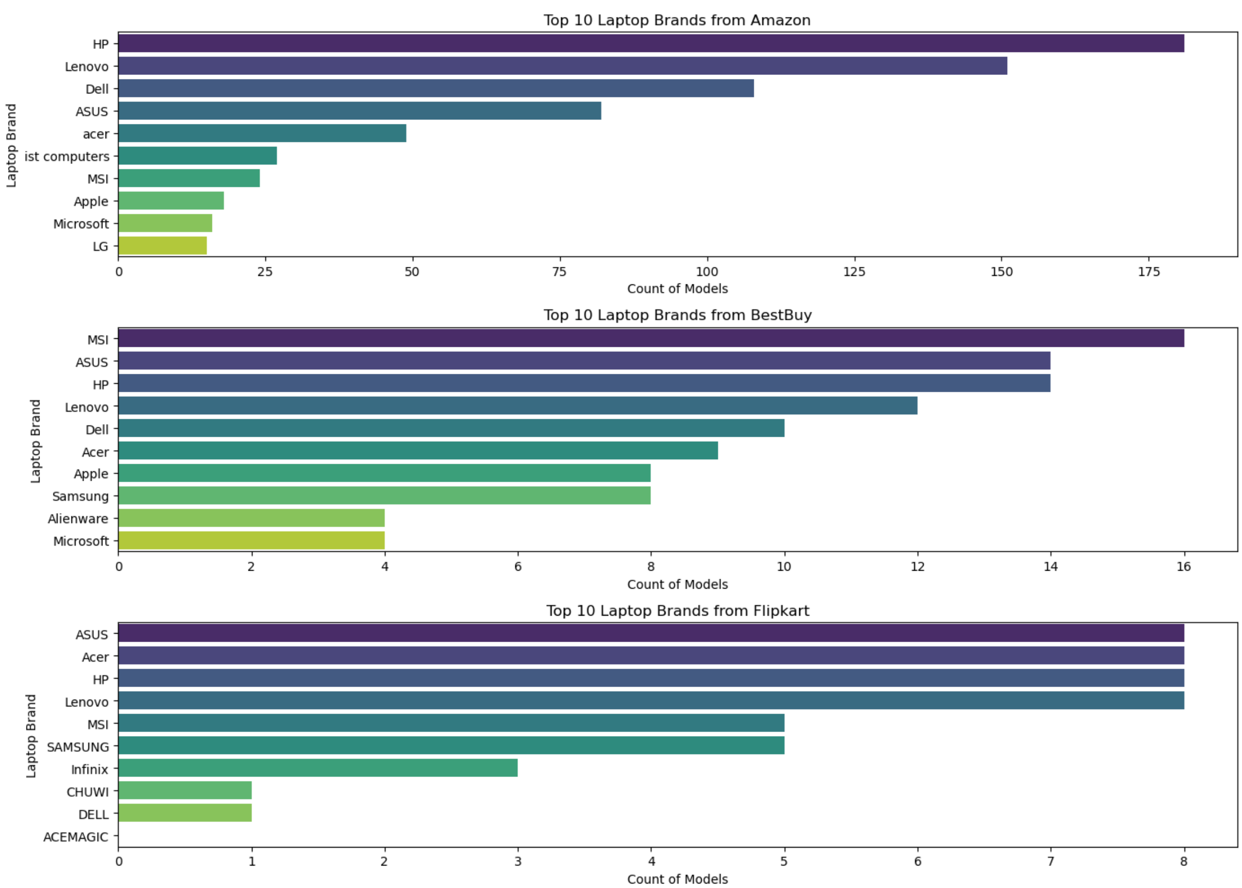
- Top 10 brand with more no of models
- Brands,HP,Lenovo,Dell,Asus,acer have more no of models across all three platforms.

```
model_counts = ydf.groupby(['Laptop_Brand', 'Source'])
['Laptop_Name'].nunique().reset_index()
# Using pivot table
pivot_table = model_counts.pivot_table(index='Laptop_Brand',
columns='Source', values='Laptop_Name', fill_value=0)

pivot_table.reset_index(inplace=True)

pivot_table.columns.name = None
print(pivot_table.head())
```

```
  Laptop_Brand  Amazon  BestBuy  Flipkart
0     ACEMAGIC       4        0         0
1     ANPCOWER       2        0         0
2          AOC       1        0         0
3         ASUS      82       14         8
4         Acer       1        9         8
```

```
#Top 10 in each source
top_10_per_source = pd.DataFrame()

for source in pivot_table.columns[1:]:
    top_10 = pivot_table.nlargest(10, source)
    top_10['Source'] = source
    top_10_per_source = pd.concat([top_10_per_source, top_10])

top_10_per_source.reset_index(drop=True, inplace=True)
```

```
plt.figure(figsize=(14, 10))

for i, source in enumerate(pivot_table.columns[1:]):
    plt.subplot(len(pivot_table.columns) - 1, 1, i + 1)
    sns.barplot(x=source, y='Laptop_Brand',
data=top_10_per_source[top_10_per_source['Source'] == source],
palette='viridis')

    plt.title(f'Top 10 Laptop Brands from {source}')
    plt.xlabel('Count of Models')
    plt.ylabel('Laptop Brand')

plt.tight_layout()

plt.show()
```



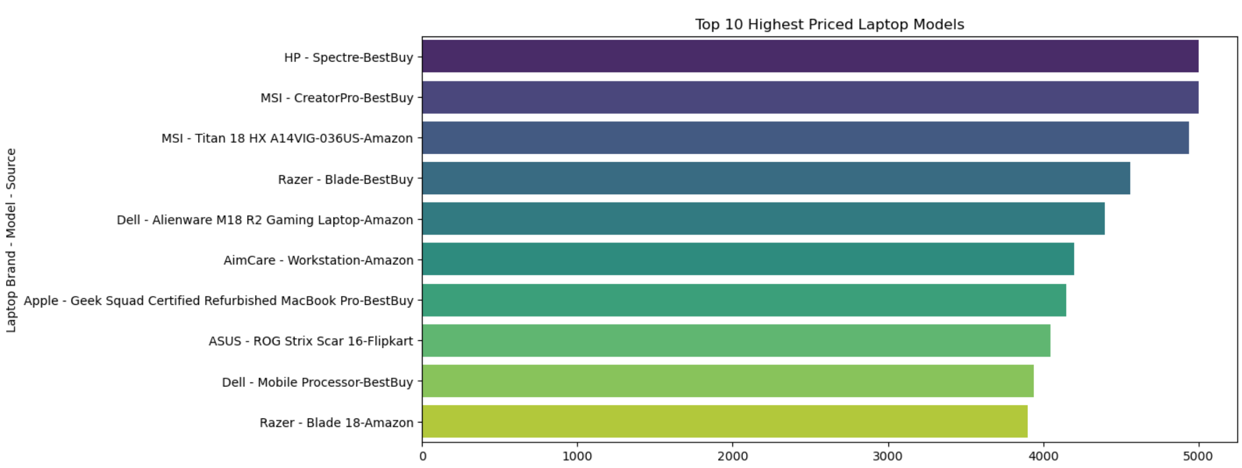
Observations

- Top 10 models in each platform
- HP, Lenovo, Dell models are more in Amazon.
- MSI,Asus,Hp are more in Best buy.
- Asus, Acer, Hp are more in Flipkart.

```
# selecting Max prices
highest_price_indices = ydf.groupby(['Laptop_Brand', 'Source'])
['Price'].idxmax()
highest_price_models = ydf.loc[highest_price_indices]

highest_price_models = highest_price_models[['Laptop_Brand',
'Laptop_Name', 'Source', 'Price']].sort_values(by='Laptop_Brand')

#Models with highest prices in each brand across platforms
display(highest_price_models.head())
```

	Laptop_Brand	Laptop_Name	Source	Price
1070	ACEMAGIC	AX17	Amazon	1496.99
903	ANPCOWER	A26	Amazon	229.49
1012	AOC	AX15	Amazon	999.99
577	ASUS	Zephyrus Duo	Amazon	3760.99
3865	ASUS	ROG Strix SCAR	BestBuy	3848.99

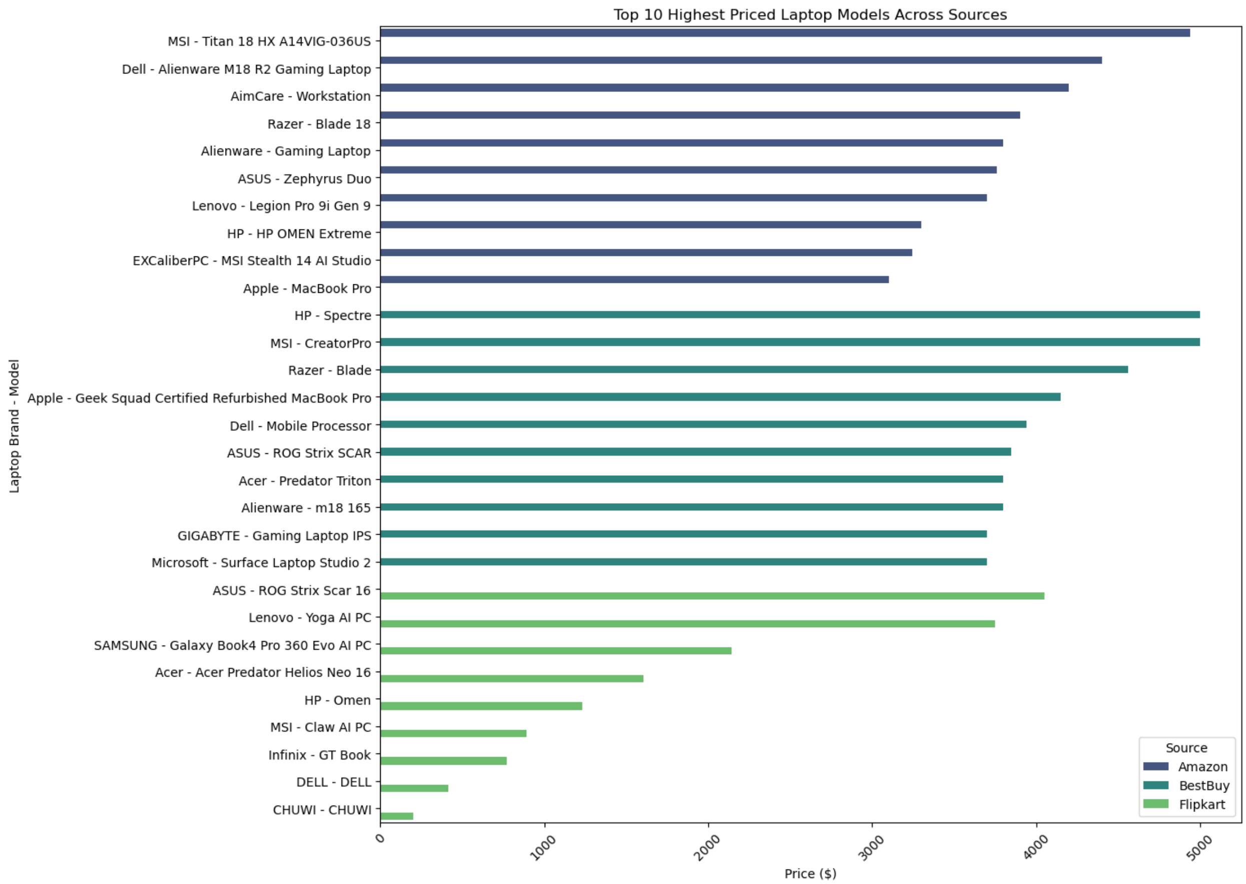
```
#top10
top_10_prices = highest_price_models.nlargest(10, 'Price')
#Adding required labels
top_10_prices['Brand_Model'] = top_10_prices['Laptop_Brand'] + ' - ' +
top_10_prices['Laptop_Name'] + '-' + top_10_prices['Source']

plt.figure(figsize=(12, 6))

sns.barplot(x='Price', y='Brand_Model', data=top_10_prices,
palette='viridis')

plt.title('Top 10 Highest Priced Laptop Models')
plt.xlabel('Price ($)')
plt.ylabel('Laptop Brand - Model - Source')

plt.show()
```



Observations

- These models have the highest price across the three platforms
- Hp spectre Bestbuy, MSI creator pro BestBuy , MSI 18 Titan Hx Amazon are the top three costliest models.

```
# Souces
sources = highest_price_models['Source'].unique()

top_10_all_sources = pd.DataFrame()

for source in sources:
    source_data = highest_price_models[highest_price_models['Source']
== source]

    top_10_prices = source_data.nlargest(10, 'Price')

    top_10_prices['Brand_Model'] = top_10_prices['Laptop_Brand'] + ' -
' + top_10_prices['Laptop_Name']

    top_10_prices['Source'] = source
    top_10_all_sources = pd.concat([top_10_all_sources,
top_10_prices])

plt.figure(figsize=(14, 10))
sns.barplot(x='Price', y='Brand_Model', hue='Source',
data=top_10_all_sources, palette='viridis')

plt.title('Top 10 Highest Priced Laptop Models Across Sources')
plt.xlabel('Price ($)')
plt.ylabel('Laptop Brand - Model')

plt.xticks(rotation=45)
plt.legend(title='Source')
plt.tight_layout()
plt.show()
```



Observations

- MSI model titan 18 HX is with highest price in Amazon
- Hp spectre is with highest price in BestBuy
- ASUS ROG Strix scar 16 is with highest price in Flipkart

Analysis Statement

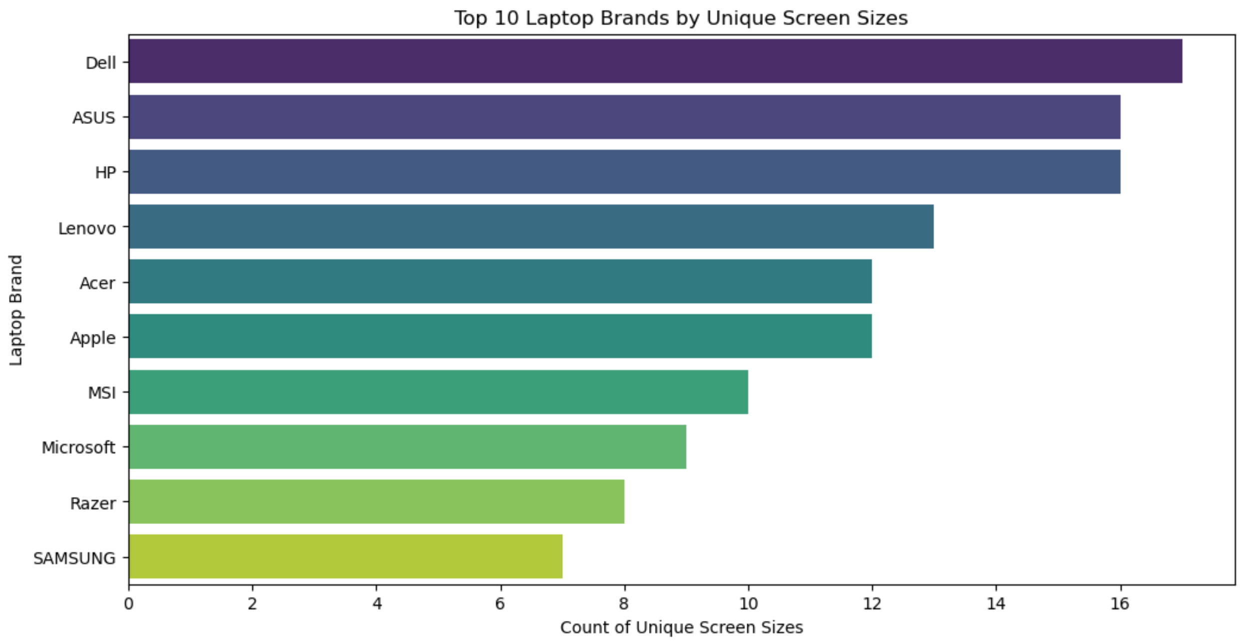
- What are the varieties in screen sizes for all models across platforms ? Average screen sizes for each brand? which screen size have better rating? What is the relation between ram and storage with screen sizes?

```
#Grouping required columns
bdf = df[['Screen_Size', 'Laptop_Brand', 'Laptop_Name', 'RAM',
'Storage', 'Rating','Source']].drop_duplicates()
display(bdf.head())
```

	Screen_Size	Laptop_Brand	Laptop_Name	RAM	Storage	Rating	Source
0	10.1	ZHAOHUIXIN	PC1068	2	64	4.5	Amazon
1	15.6	TPV	AceBook	16	512	4.5	Amazon
2	16.0	HP	Elitebook	32	2048	4.0	Amazon
3	13.6	Apple	MacBook Air	8	256	4.0	Amazon
4	15.3	Apple	MacBook Air	16	512	4.0	Amazon

```
#Unique screen sizes
unique_screen_sizes = bdf['Screen_Size'].unique()
print("Total Unique screen sizes across platforms:",
len(unique_screen_sizes))
print(unique_screen_sizes)

Total Unique screen sizes across platforms: 38
[10.1  15.6  16.   13.6  15.3  14.   17.3  14.2  13.3  16.2  15.5  15.
 11.6  13.   16.1  14.1  11.   13.4  18.   12.45 14.5  12.2  13.5
11.5
 17.   15.4   8.   12.3  14.4  10.51 13.1  12.5  12.   13.8  12.4   7.
 16.3  10.5 ]

#Unique models
model_unique_screensizes = ydf[['Screen_Size', 'Laptop_Brand',
'Laptop_Name','Source']].drop_duplicates()
display(model_unique_screensizes.head())

   Screen_Size Laptop_Brand  Laptop_Name  Source
0         10.1   ZHAOHUIXIN       PC1068  Amazon
1         15.6          TPV      AceBook  Amazon
2         16.0           HP    Elitebook  Amazon
3         13.6        Apple  MacBook Air  Amazon
4         15.3        Apple  MacBook Air  Amazon

# Sizes
screen_size_counts = model_unique_screensizes.groupby('Laptop_Brand')
['Screen_Size'].nunique().reset_index()

screen_size_counts.columns = ['Laptop_Brand',
'Unique_Screen_Size_Count']

display(screen_size_counts.head())

  Laptop_Brand  Unique_Screen_Size_Count
0     ACEMAGIC                         4
1     ANPCOWER                         2
2          AOC                         1
3         ASUS                        16
4         Acer                        12

top_10_brands = screen_size_counts.nlargest(10,
'Unique_Screen_Size_Count')

plt.figure(figsize=(12, 6))

sns.barplot(x='Unique_Screen_Size_Count', y='Laptop_Brand',
data=top_10_brands, palette='viridis')

plt.title('Top 10 Laptop Brands by Unique Screen Sizes')
plt.xlabel('Count of Unique Screen Sizes')
```

```
plt.ylabel('Laptop Brand')

plt.show()
```

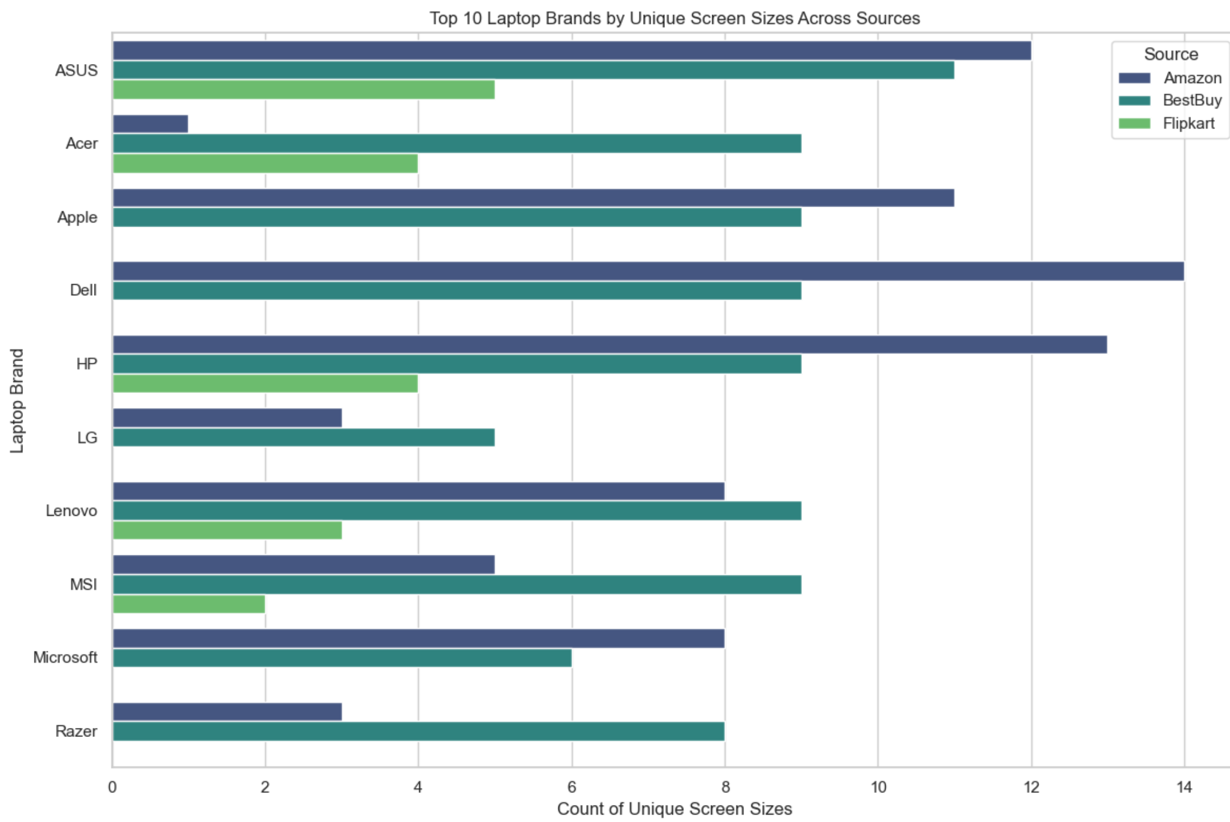


Observations

- Total no of screen sizes:38
- models with unique screen sizes.
- No of screen sizes each brand offer.
- Top 10 brands with more no of unique screen sizes.
- Dell, Asus , Hp has more no of screen sizes across three platforms.

```
screen_size_counts_by_source =
model_unique_screensizes.groupby(['Laptop_Brand', 'Source'])
['Screen_Size'].nunique().reset_index()

screen_size_counts_by_source.columns = ['Laptop_Brand', 'Source',
'Unique_Screen_Size_Count']


top_10_all_sources =
screen_size_counts_by_source.groupby('Laptop_Brand').agg({'Unique_Scre
en_Size_Count': 'sum'}).reset_index()
top_10_all_sources = top_10_all_sources.nlargest(10,
'Unique_Screen_Size_Count')

top_10_with_source =
screen_size_counts_by_source[screen_size_counts_by_source['Laptop_Bran
d'].isin(top_10_all_sources['Laptop_Brand'])]
```

```python
sns.set(style="whitegrid")

plt.figure(figsize=(12, 8))
sns.barplot(x='Unique_Screen_Size_Count', y='Laptop_Brand',
hue='Source', data=top_10_with_source, palette='viridis')

plt.title('Top 10 Laptop Brands by Unique Screen Sizes Across
Sources')
plt.xlabel('Count of Unique Screen Sizes')
plt.ylabel('Laptop Brand')

plt.legend(title='Source')
plt.tight_layout()
plt.show()
```



Observations

- Dell , Hp, Asus are top 3 highest no of screen sizes in Amazon.
- Asus, Acer, Apple are top 3 highest no of screen sizes in Bestbuy.
- Asus, Acer, Hp are top 3 highest no of screen sizes in Flipkart.

```python
#Average sizes for each brand
average_screen_sizes = bdf.groupby('Laptop_Brand')
['Screen_Size'].mean().reset_index()
```

```
average_screen_sizes.columns = ['Laptop_Brand', 'Average_Screen_Size']

average_screen_sizes['Average_Screen_Size'] =
average_screen_sizes['Average_Screen_Size'].round(2)

average_screen_sizes =
average_screen_sizes.sort_values(by='Average_Screen_Size')
#average_screen_sizes =
average_screen_sizes.sort_values(by=['Average_Screen_Size'],ascending=
False)
display(average_screen_sizes.head())
```

	Laptop_Brand	Average_Screen_Size
37	KOOSMILE	8.0
36	KOOFORWAY	8.0
72	ZHAOHUIXIN	10.1
34	Infinix	11.6
29	Harry Potter	11.6

```
average_screen_sizes_by_source = ydf.groupby(['Laptop_Brand',
'Source'])['Screen_Size'].mean().reset_index()

average_screen_sizes_by_source.columns = ['Laptop_Brand', 'Source',
'Average_Screen_Size']
average_screen_sizes_by_source['Average_Screen_Size'] =
average_screen_sizes_by_source['Average_Screen_Size'].round(2)
#average_screen_sizes_by_source =
average_screen_sizes_by_source.sort_values(by=[ 'Average_Screen_Size']
)
average_screen_sizes_by_source =
average_screen_sizes_by_source.sort_values(by=[ 'Average_Screen_Size']
,ascending=False)
print(average_screen_sizes_by_source.head())
```

```
   Laptop_Brand    Source  Average_Screen_Size
47     KOOSMILE    Amazon                 8.00
46    KOOFORWAY    Amazon                 8.00
91   ZHAOHUIXIN    Amazon                10.10
20        CHUWI  Flipkart                11.00
44      Infinix  Flipkart                11.35
```

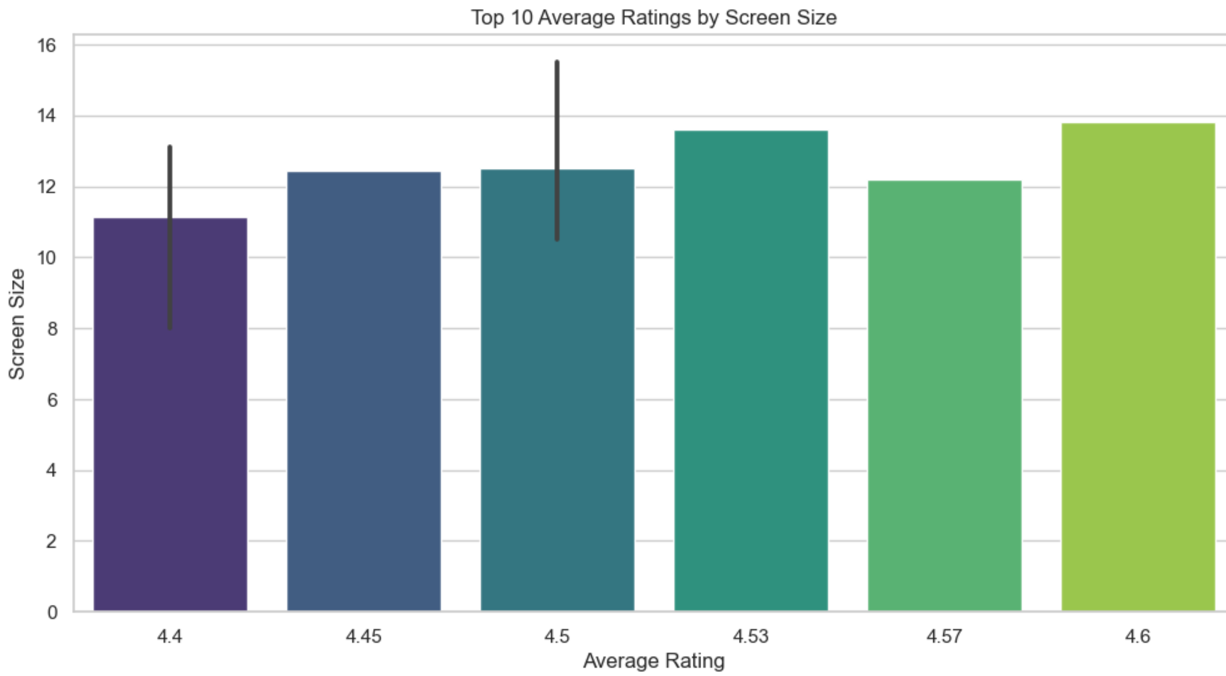
Observations

- Most of the models have screen size grater than 14 inches.
- Amazon has highest screen sizes greter than 17 inches.
- Amazon has lowest screen size models of sizes 8 inches.
- Brand Koosmile has the lowest screen sizes.
- Brand NBVCX has the highest screen sizes.

```
average_rating_by_screen_size = bdf.groupby('Screen_Size')
['Rating'].mean().reset_index()

average_rating_by_screen_size.columns = ['Screen_Size',
'Average_Rating']

average_rating_by_screen_size['Average_Rating'] =
average_rating_by_screen_size['Average_Rating'].round(2)

sorted_average_rating =
average_rating_by_screen_size.sort_values(by='Average_Rating',
ascending=False).reset_index(drop=True)

display(sorted_average_rating.head())
```

	Screen_Size	Average_Rating
0	13.80	4.60
1	12.20	4.57
2	13.60	4.53
3	15.50	4.50
4	10.51	4.50

```
top_10_average_rating = sorted_average_rating.head(10)

plt.figure(figsize=(12, 6))

sns.barplot(x='Average_Rating', y='Screen_Size',
data=top_10_average_rating, palette='viridis')

plt.title('Top 10 Average Ratings by Screen Size')
plt.xlabel('Average Rating')
plt.ylabel('Screen Size')

plt.show()
```



Observations

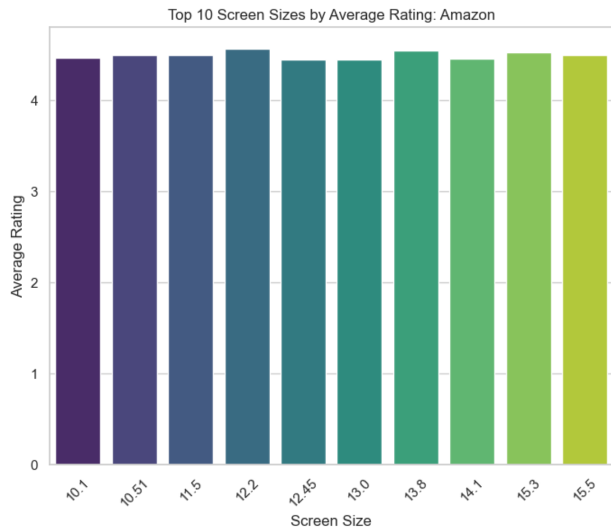
- Screen sizes 13.80, 12.20,13.60, has the highest rating of 4.60, 4.57, 4.53 respectively.
- Top 10 screen size models with highest ratings.

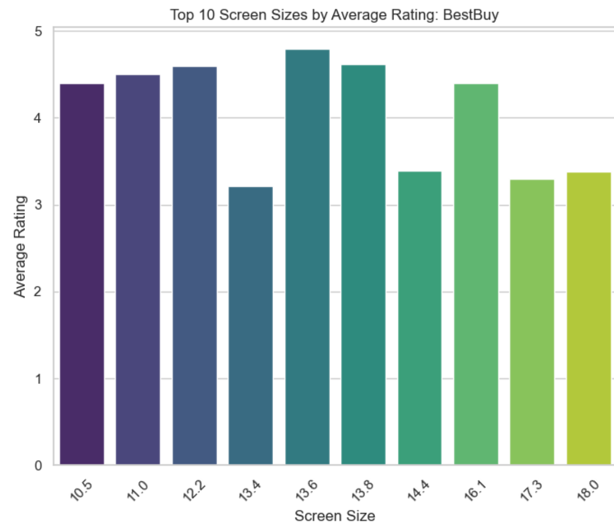
```
average_rating_by_source_and_size = bdf.groupby(['Source',
'Screen_Size'])['Rating'].mean().reset_index()

average_rating_by_source_and_size.columns = ['Source', 'Screen_Size',
'Average_Rating']

average_rating_by_source_and_size['Average_Rating'] =
average_rating_by_source_and_size['Average_Rating'].round(2)

print(average_rating_by_source_and_size.head())
```

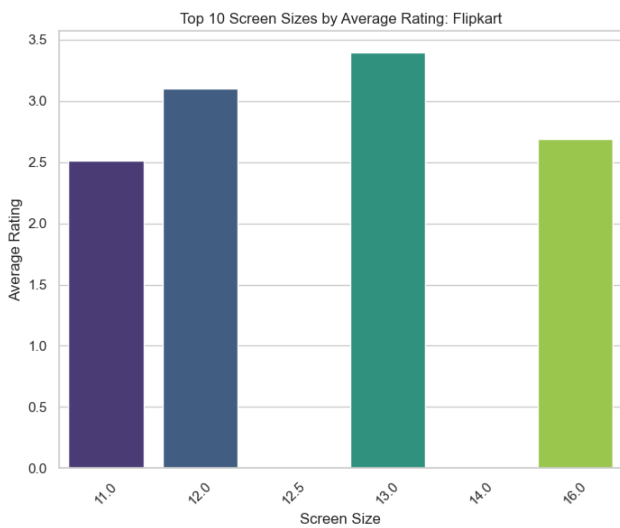
```
   Source  Screen_Size  Average_Rating
0  Amazon         7.00            4.40
1  Amazon         8.00            4.40
2  Amazon        10.10            4.47
3  Amazon        10.51            4.50
4  Amazon        11.00            4.41
```

```
average_rating_by_source_and_size = bdf.groupby(['Source',
'Screen_Size'])['Rating'].mean().reset_index()

average_rating_by_source_and_size.columns = ['Source', 'Screen_Size',
'Average_Rating']

average_rating_by_source_and_size['Average_Rating'] =
```

```python
average_rating_by_source_and_size['Average_Rating'].round(2)

sources = average_rating_by_source_and_size['Source'].unique()

n = len(sources)
ncols = 2
nrows = (n + ncols - 1) // ncols

plt.figure(figsize=(14, 6 * nrows))

for i, source in enumerate(sources):
    ax = plt.subplot(nrows, ncols, i + 1)

    source_data =
average_rating_by_source_and_size[average_rating_by_source_and_size['S
ource'] == source]

    top_10_source_data = source_data.nlargest(10, 'Average_Rating')

    sns.barplot(x='Screen_Size', y='Average_Rating',
data=top_10_source_data, palette='viridis', ax=ax)

    ax.set_title(f'Top 10 Screen Sizes by Average Rating: {source}')
    ax.set_xlabel('Screen Size')
    ax.set_ylabel('Average Rating')
    ax.tick_params(axis='x', rotation=45)

plt.tight_layout()

plt.show()
```







Observations

- screen sizes across platforms.

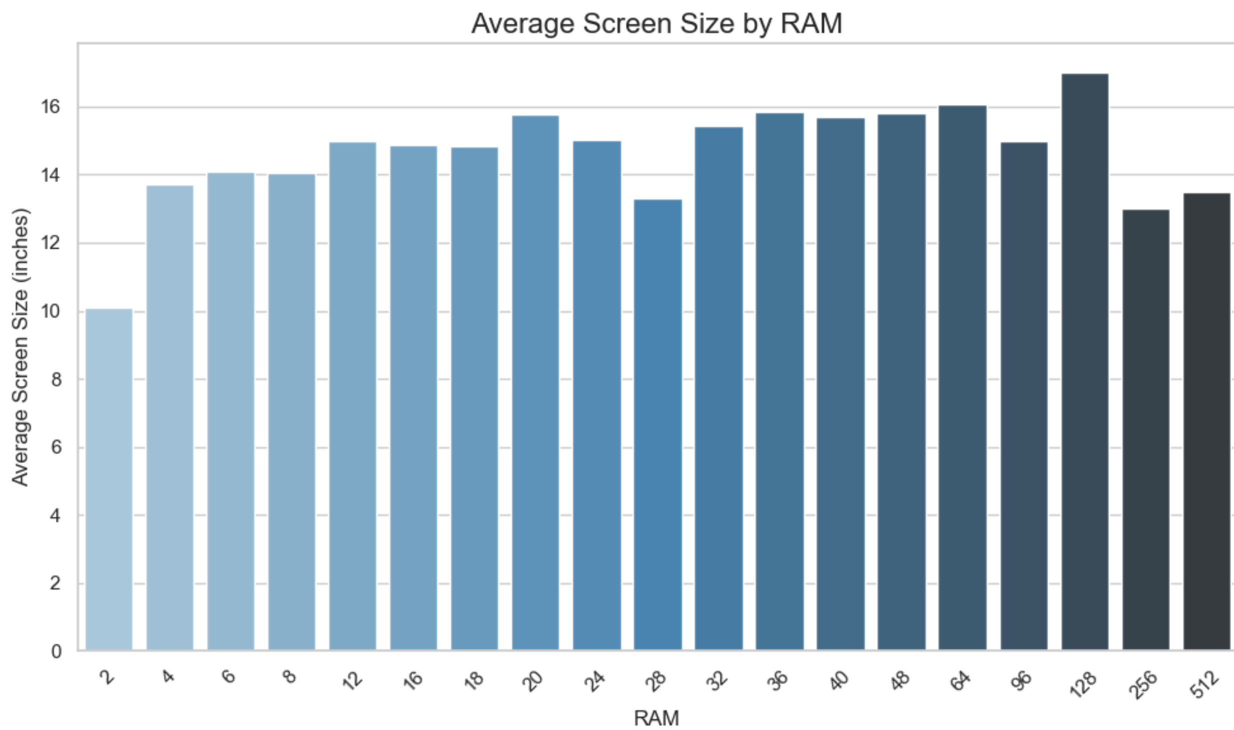
```
#Grouping data
average_screen_size_by_ram_storage = bdf.groupby(['RAM', 'Storage'])
['Screen_Size'].mean().reset_index()

average_screen_size_by_ram_storage.columns = ['RAM', 'Storage',
'Average_Screen_Size']

average_screen_size_by_ram_storage['Average_Screen_Size'] =
average_screen_size_by_ram_storage['Average_Screen_Size'].round(2)

sorted_average_screen_size_by_ram_storage =
average_screen_size_by_ram_storage.sort_values(by='Average_Screen_Size
', ascending=False)

print(sorted_average_screen_size_by_ram_storage.head())
```

```
    RAM Storage  Average_Screen_Size
69   40     512                17.30
80  128    8192                17.00
62   32    8192                17.00
77   64    8192                16.36
76   64    4096                16.21
```

```
# Group by RAM
average_screen_size_by_ram = bdf.groupby('RAM')
['Screen_Size'].mean().reset_index()

average_screen_size_by_ram.columns = ['RAM', 'Average_Screen_Size']

average_screen_size_by_ram['Average_Screen_Size'] =
average_screen_size_by_ram['Average_Screen_Size'].round(2)

average_screen_size_by_ram_sorted =
average_screen_size_by_ram.sort_values(by='Average_Screen_Size')

print(average_screen_size_by_ram_sorted.head())
```

```
    RAM  Average_Screen_Size
0     2                 10.1
17  256                 13.0
9    28                 13.3
18  512                 13.5
1     4                 13.7
```

```
plt.figure(figsize=(10,6))

# Create the seaborn barplot
sns.barplot(x='RAM', y='Average_Screen_Size',
data=average_screen_size_by_ram_sorted, palette='Blues_d')

plt.title('Average Screen Size by RAM', fontsize=16)
plt.xlabel('RAM', fontsize=12)
plt.ylabel('Average Screen Size (inches)', fontsize=12)

plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

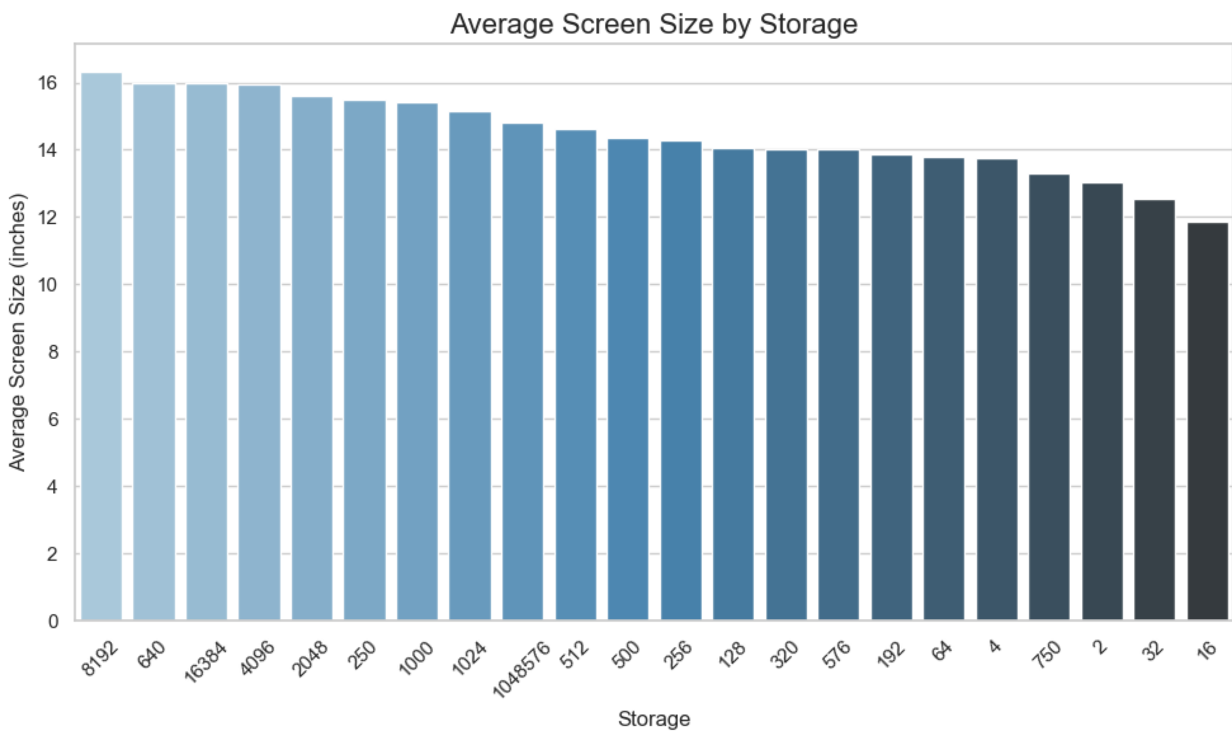



```
# Group by Storage and calculating the mean of Screen_Size
average_screen_size_by_storage = bdf.groupby('Storage')
['Screen_Size'].mean().reset_index()

average_screen_size_by_storage.columns = ['Storage',
'Average_Screen_Size']

average_screen_size_by_storage['Average_Screen_Size'] =
average_screen_size_by_storage['Average_Screen_Size'].round(2)

average_screen_size_by_storage_sorted =
average_screen_size_by_storage.sort_values(by='Average_Screen_Size',
ascending=False)

print(average_screen_size_by_storage_sorted.head())
```

```
   Storage  Average_Screen_Size
21    8192                16.32
19     640                16.00
5    16384                16.00
14    4096                15.95
8     2048                15.60
```

```
plt.figure(figsize=(10,6))

sns.barplot(x='Storage', y='Average_Screen_Size',
data=average_screen_size_by_storage_sorted, palette='Blues_d')
```

```
plt.title('Average Screen Size by Storage', fontsize=16)
plt.xlabel('Storage', fontsize=12)
plt.ylabel('Average Screen Size (inches)', fontsize=12)

plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



Observations

- Models with ram size 128,64,36 have higher screen sizes.
- Models with ram size 2,256,128 have lower screen sizes.
- Models with storage size 8192,640,16384 has the higher screen sizes.
- Models with storage size 2,32,16 has the lower screen sizes.

Name: Shaurya Mathur

UB ID:50611201

```
sdf = df.copy()
sdf.head()

  Laptop_Brand  Laptop_Name Processor_Company Operating_System
Processor  \
0   ZHAOHUIXIN       PC1068          Alwinner          Android     1.8
GHz a13
```

```
1          TPV      AceBook             Intel   Windows 11 Pro
Core i5
2           HP    Elitebook             Intel   Windows 11 Pro  Intel
Core i7
3        Apple  MacBook Air             Apple           Mac OS
Apple M3
4        Apple  MacBook Air             Apple           Mac OS
Apple M3

   Number_of_Reviews    Price Storage_Type Storage  Rating
Screen_Size  RAM  \
0                  1   119.99         EMMC      64     4.5
10.1    2
1                 13   309.99          SSD     512     4.5
15.6   16
2                  5  1079.00          SSD    2048     4.0
16.0   32
3                  0   929.00          SSD     256     4.0
13.6    8
4                  0  1449.00          SSD     512     4.0
15.3   16

   Source
0  Amazon
1  Amazon
2  Amazon
3  Amazon
4  Amazon
```

```
sdf.columns
```

```
Index(['Laptop_Brand', 'Laptop_Name', 'Processor_Company',
'Operating_System',
       'Processor', 'Number_of_Reviews', 'Price', 'Storage_Type',
'Storage',
       'Rating', 'Screen_Size', 'RAM', 'Source'],
      dtype='object')
```

Question

How do the prices of laptops with similar specifications vary across brands?

```
sdf.isnull().sum()
```

```
Laptop_Brand         0
Laptop_Name          0
Processor_Company    0
Operating_System     0
Processor            0
Number_of_Reviews    0
```

```
Price                  0
Storage_Type           0
Storage                0
Rating                 0
Screen_Size            0
RAM                    0
Source                 0
dtype: int64
```

```
sdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4815 entries, 0 to 4837
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Laptop_Brand       4815 non-null   object
 1   Laptop_Name        4815 non-null   object
 2   Processor_Company  4815 non-null   object
 3   Operating_System   4815 non-null   object
 4   Processor          4815 non-null   object
 5   Number_of_Reviews  4815 non-null   int64
 6   Price              4815 non-null   float64
 7   Storage_Type       4815 non-null   object
 8   Storage            4815 non-null   object
 9   Rating             4815 non-null   float64
 10  Screen_Size        4815 non-null   float64
 11  RAM                4815 non-null   int64
 12  Source             4815 non-null   object
dtypes: float64(3), int64(2), object(8)
memory usage: 526.6+ KB
```

```
sdf.describe().T
```

```
                    count        mean         std    min      25%
50%  \
Number_of_Reviews  4815.0  131.620768  245.316441   0.00     1.00
17.00
Price              4815.0  797.631734  690.321336  44.79   369.17
579.99
Rating             4815.0    3.729221    1.481900   0.00     4.00
4.30
Screen_Size        4815.0   14.074457    1.907507   7.00    12.75
14.00
RAM                4815.0   16.904258   16.381328   2.00     8.00
16.00

                      75%      max
Number_of_Reviews  194.00  5121.00
Price              989.99  5049.99
```

```
Rating                  4.40     5.00
Screen_Size            15.60    18.00
RAM                    16.00   512.00
```

```
sdf['Laptop_Brand'].unique()
```

```
array(['ZHAOHUIXIN', 'TPV', 'HP', 'Apple', 'Lenovo', 'ASUS', 'acer',
       'Dell', 'MSI', 'Zuleisy', 'Molegar', 'HEWLETT PACKARD',
       'Oemgenuine', 'ACEMAGIC', 'SGIN', 'SAIWAN', 'jumper',
'Microsoft',
       'ApoloSign', 'ist computers', 'SAMSUNG', 'Hp', 'FUNYET',
'Naclud',
       'BiTECOOL', 'NIAKUN', 'Samsung', 'WIPEMIK', 'Lapbook',
'Alienware',
       'LG', 'AimCare', 'Harry Potter', 'KOOFORWAY', 'Razer',
'Panasonic',
       'ZENAERO', 'INHONLAP', 'Maxsignage', 'CHUWI', 'ApoloMedia',
       'LETSUNG', 'VAIZMOYE', 'hp', 'KOOSMILE', 'VGKE', 'ANPCOWER',
       'GIGABYTE', 'EXCaliberPC', 'EYY', 'Morostron', 'AOC', 'Nmybwo',
       'Coolby', 'THKGRCE', 'Acer', 'IJKKJI', 'Getac', 'ECOHERO',
       'Gina Joyfurno', 'Toughbook', 'Gateway', 'KAIGERR', 'Akocrsiy',
       'HP Inc.', 'Rumtuk', 'MTWZMM', 'NBVCXSD', 'RIANIFEL', 'BHWW',
       'SAINTDISE', 'DELL', 'Infinix', 'Geek', 'Thomson', 'Hyundai',
       'Refurbished', 'MacBook'], dtype=object)
```

```
sdf[sdf['Laptop_Brand'].str.contains('hp')]
```

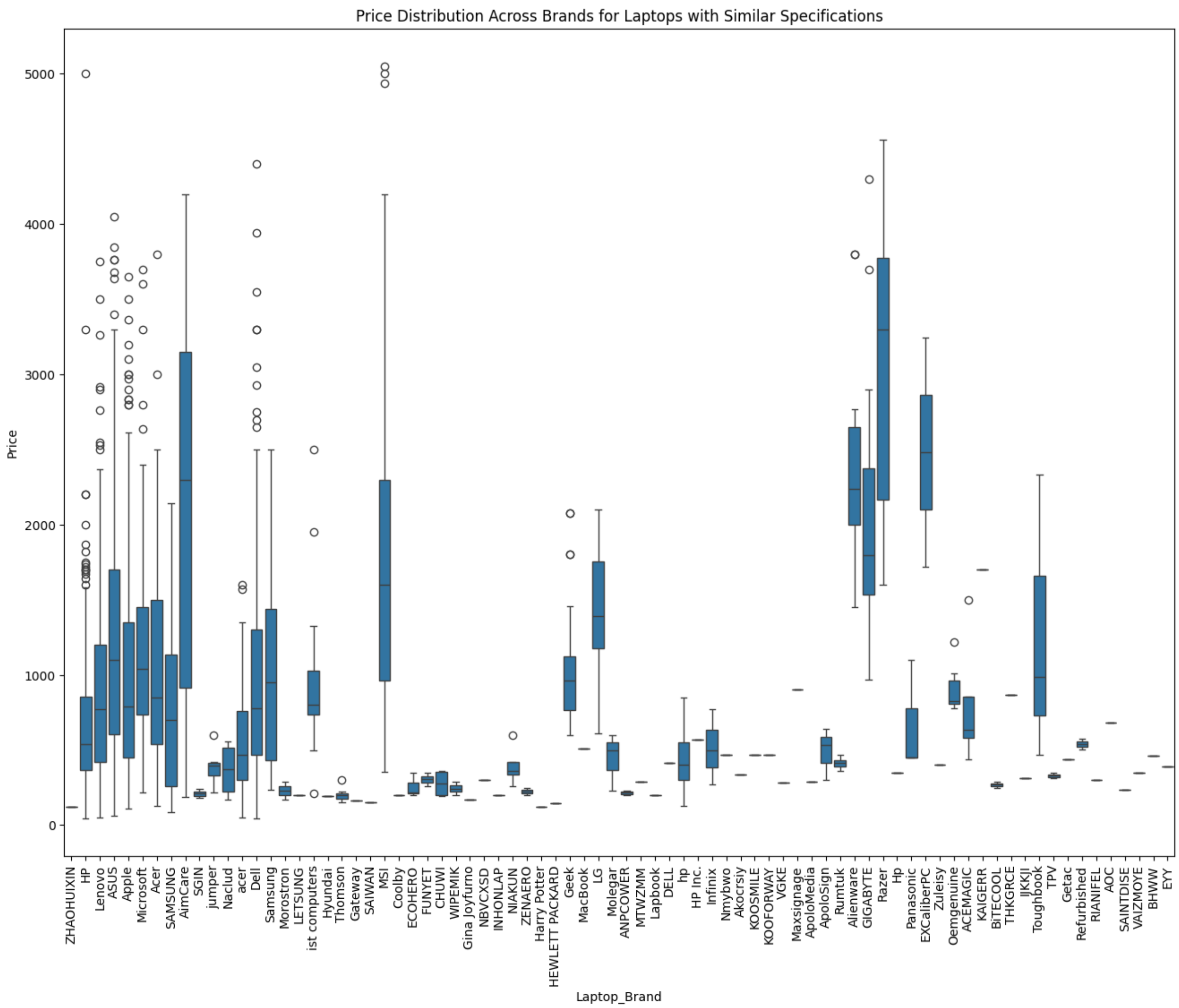
```
     Laptop_Brand   Laptop_Name Processor_Company Operating_System
Processor  \
868            hp    DFSFGSGD15             Intel       Windows 11
Pentium
1324           hp  HP 17 Laptop             Intel   Windows 11 Pro
Core i7
1328           hp  HP EliteBook             Intel   Windows 10 Pro
Core i5
1632           hp  HP EliteBook             Intel       Windows 10
Core i5

      Number_of_Reviews   Price Storage_Type Storage  Rating
Screen_Size  \
868                 468  359.00          SSD     128     4.5
15.6
1324                 17  849.99          SSD    2048     4.4
17.3
1328                  7  450.06          SSD     512     4.4
14.0
1632                 15  129.99          SSD     512     4.5
14.0

      RAM  Source
```

```
868     16  Amazon
1324    32  Amazon
1328    16  Amazon
1632     8  Amazon
```

```python
price_variation = sdf.groupby(['RAM', 'Storage', 'Processor',
'Screen_Size']).agg({
    'Price': ['mean', 'std', 'min', 'max', 'count']
}).reset_index()

price_variation
```

```
      RAM Storage                Processor Screen_Size    Price
\
                                                           mean  std
min
0       2       2                     A133        10.1   119.99  0.0
119.99
1       2      64             1.8 GHz a13        10.1   119.99  0.0
119.99
2       2      64                      A13        10.1   119.99  0.0
119.99
3       4     128   Celeron  - 14-dq0760dx        14.0   139.99  NaN
139.99
4       4     128   Celeron  - 14-dq0761dx        14.0   199.99  NaN
199.99
...   ...     ...                      ...         ...      ...  ...
...
2022  128    8192          Core i7 Family        16.0  2899.00  NaN
2899.00
2023  128    8192           Intel Core i9        18.0  4939.00  NaN
4939.00
2024  256     256                 4.2 GHz        13.0   890.00  NaN
890.00
2025  512     256                 4.2 GHz        13.0   870.99  NaN
870.99
2026  512     512     Intel Processor N100        14.0   389.99  NaN
389.99


          max count
0      119.99    32
1      119.99    42
2      119.99    17
3      139.99     1
```

```
4       199.99     1
...        ...   ...
2022  2899.00     1
2023  4939.00     1
2024   890.00     1
2025   870.99     1
2026   389.99     1

[2027 rows x 9 columns]

grouped_laptops = df.groupby(['RAM', 'Storage', 'Processor',
'Screen_Size', 'Laptop_Brand']).agg({'Price': 'mean'}).reset_index()
grouped_laptops

      RAM Storage                Processor  Screen_Size Laptop_Brand
Price
0       2       2                     A133         10.1   ZHAOHUIXIN
119.99
1       2      64             1.8 GHz a13          10.1   ZHAOHUIXIN
119.99
2       2      64                      A13         10.1   ZHAOHUIXIN
119.99
3       4     128   Celeron  - 14-dq0760dx         14.0           HP
139.99
4       4     128   Celeron  - 14-dq0761dx         14.0           HP
199.99
...   ...     ...                      ...          ...          ...
...
2247  128    8192           Core i7 Family         16.0       Lenovo
2899.00
2248  128    8192            Intel Core i9         18.0          MSI
4939.00
2249  256     256                  4.2 GHz         13.0    Microsoft
890.00
2250  512     256                  4.2 GHz         13.0    Microsoft
870.99
2251  512     512     Intel Processor N100         14.0          EYY
389.99

[2252 rows x 6 columns]

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(16,12))
sns.boxplot(x='Laptop_Brand', y='Price', data=grouped_laptops)
plt.title('Price Distribution Across Brands for Laptops with Similar
Specifications')
plt.xticks(rotation=90)
plt.show()
```


Price Distribution Across Brands for Laptops with Similar Specifications
Price
0
1000
2000
3000
4000
5000
Laptop_Brand
ZHAOHUIXIN
HP
Lenovo
ASUS
Apple
Microsoft
Acer
SAMSUNG
AimCare
SGIN
jumper
Naclud
acer
Dell
Samsung
Morostron
LETSUNG
ist computers
Hyundai
Thomson
Gateway
SAIWAN
MSI
Coolby
ECOHERO
FUNYET
CHUWI
WIPEMIK
Gina Joyfurno
NBVCXSD
INHONLAP
NIAKUN
ZENAERO
Harry Potter
HEWLETT PACKARD
Geek
MacBook
LG
Molegar
ANPCOWER
MTWZMM
Lapbook
DELL
hp
HP Inc.
Infinix
Nmybwo
Akocrsiy
KOOSMILE
KOOFORWAY
VGKE
Maxsignage
ApoloMedia
ApoloSign
Rumtuk
Alienware
GIGABYTE
Razer
Hp
Panasonic
EXCaliberPC
Zuleisy
Oemgenuine
ACEMAGIC
KAIGERR
BiTECOOL
THKGRCE
JJKKJJ
Toughbook
TPV
Getac
Refurbished
RIANIFEL
AOC
SAINTDISE
VAIZMOYE
BHWW
EYY

References