Amazon India Sales Analysis

Soren Epple, Lucas Sadoulet, Tjoet Lakmana

2025-05-08

Introduction

Shopping online has become a huge part of everyday life, and platforms like Amazon make it easy for people to browse, compare, and buy products in just a few clicks. The Amazon India Sales dataset gives us a behind-the-scenes look at how product pricing, discounts, and customer ratings play out across different types of items. With many products and reviews, it's a great dataset to explore what matters to online shoppers.

Here are the main questions we set out to explore:

Soren's Exploratory Question:

- What products tend to have a higher rating with respect to their main category?

  This helps us understand which products are actively being bought and reviewed. It can also be helpful to make interpretations with respect to review count, and how depending on the number can affect the confidence in the product rating. High volumes in review count generally show a higher confidence in the rating, whereas, lower volumes have a lesser amount of confidence.

- Is review count a reliable datapoint?
- How can we use average ratings to make interpretations off other variables?

Tjoet's Exploratory Questions?

- How many ratings do products get per customer, and what does that tell us about engagement?

  This helps us understand how actively customers interact with different types of products. A high number of ratings might indicate strong customer engagement or high sales volume, while low counts may suggest limited interaction or visibility.

- How does price vary across different product categories?

  Exploring pricing by category helps us see how products are positioned in the market. Some categories might lean toward luxury pricing, while others depend on affordability and high-volume turnover.

- Is there a connection between discount percentage and customer ratings?

  This question lets us explore whether discounts influence satisfaction. Do big discounts lead to better reviews or do they come at the cost of perceived value or quality?

Lucas' Exploratory Questions?

- What can we learn from users' spending habits?

  Comparing the users spending habits through making a new table of only users' spending total, and the number of products bought. This will help us better understand trends of the population, and possible determine the unique nature of inidividual users if their are ouliers.

- Can we learn more about users' ratings vs the discount they received for the product?

  This is a questions we were interseting in answer, although struggled to bring to work. While the average users rating for each product is avaliable, and the users are also present, it doesn't break down rating by users. It only shows the users comments. We could have worked around this by either taking the average of the average user's spending, which would have yeilded inaccurate results for users of popular products. Alternatively, we could have guessed their ratings based off of their comments, but this sentiment analysis wasn't within the scope of this project.

- Retroactive Questions: Can we locate bots using our data analysis?

  This question has alot of utility for real world Amazon problem, preventing scams and fake reviews, which is a big issue for ensuring user protections. We appear to have caught bot-like activity in the population set, since this account followed both a high number of comments by low average spending.

These questions are important because they help uncover patterns in online shopping behavior. By analyzing engagement, pricing, and satisfaction, we can better understand what makes a product stand out in a crowded marketplace.

Data Provenance

The dataset we used comes from Kaggle, a popular online platform for data science and machine learning. Kaggle hosts thousands of datasets that are free to use for analysis, modeling, and competitions. It is widely used by students, researchers, and professionals to practice and share data projects.

The specific dataset we used is called the "Amazon Sales Dataset," published by Karkavel Rajaj. It includes a wide range of product-related information from Amazon India. the key columns included in the dataset:

- **product_id**: Unique identifier for each product

- **product_name**: Name of the product

- **category**: Product category
- **discounted_price**: Price after discount
- **actual_price**: Original price before discount
- **discount_percentage**: Percent discount applied
- **rating**: Star rating given to the product
- **rating_count**: Number of people who rated the product
- **about_product**: Description or summary of the product
- **user_id**: Unique ID of the user who left a review
- **user_name**: Name of the reviewing user
- **review_id**: ID for the specific review
- **review_title**: Title of the user's review
- **review_content**: Full text of the user's review
- **img_link**: Image URL of the product
- **product_link**: Link to the product listing on Amazon

Data Cleaning and Wrangling

To prepare the data for analysis, we followed a multi-step process to clean, convert, and restructure the data for meaningful insights:

Removed Duplicate Product Entries

We began by removing duplicate entries based on the `product_id` column using `distinct(product_id, .keep_all = TRUE)`. This step ensured that each product was represented only once in our dataset. Duplicates likely existed due to multiple reviews or variations for the same product. Since we were analyzing product-level trends (e.g., pricing, ratings), keeping only the first unique entry per product helped prevent double-counting.

Converted Prices from INR to USD

The dataset included prices in Indian Rupees with formatting like the symbol and commas. We:

- Removed those symbols using `gsub()`
- Converted values into USD using a fixed exchange rate of 1 = \$0.012
- This allowed us to compare prices and discounts using a more familiar currency.

Cleaned and Converted Rating Counts

We cleaned the `rating_count` column by removing commas and converting it to a numeric type. This made it possible to perform calculations like total reviews or average engagement per category.

Selected Relevant Columns

To simplify our dataset and focus on the most useful information, we retained:

- `product_name`, `category`, `discounted_price_USD`, `actual_price_USD`, `rating`, `rating_count`, and `user_id`.

This reduced clutter and made downstream analysis more efficient.

Separated User IDs

In the raw dataset, some products were associated with multiple `user_id`s stored in a single cell, separated by commas. This structure made it difficult to analyze individual customer interactions. To resolve this, we used the `separate_rows()` function to split these entries so that each `user_id` had its own row.

This step allowed us to more accurately analyze customer engagement by treating each user as a distinct data point, which is especially important for questions involving rating count per customer or overall review behavior.

Final Tidied Data

After completing the cleaning and transformation steps, our final dataset is organized in a tidy format where each row represents a unique interaction between a product and a user. The data includes only the most relevant variables needed to answer our research questions, with all prices converted to USD and formatted consistently.

The columns in the final dataset include:

- **`product_name`** – Name of the product
- **`category`** – Full category path of the product

- `discounted_price_USD` – Discounted price in U.S. dollars
- `actual_price_USD` – Original price in U.S. dollars
- `rating` – Customer rating for the product
- `rating_count` – Number of customer ratings
- `user_id` – Individual customer ID, one per row

Paradigm

For this project, we are looking to conduct exploratory analysis, since while looking for patterns and discrepancies in the data we are analyzing, we don't have any conclusions we are searching to confirm or refute. We did this because while our data is large and contains many useful data points, it doesn't contain many pieces of information that would be important to any preexisting hypothesis, including dates, concrete individual tables, and uncertainty around the scope of the dataset. This last one is most important because while we know it's Amazon data for India in 2025, we believe that they would be significantly more enemies if that were truly the extent of the dataset. Additionally, it seems this dataset only covers technology products, but it is not specified in the data download. Without a more concrete scope for the dataset, we felt that confirmatory data analysis would not work as well. Additionally we wanted to use Tidyverse in our project since it would allow us to generate more interesting graphs, cleaner code, and programming in a way we're more familiar with.

This structure allows us to analyze product pricing, discount trends, rating patterns, and customer engagement at both the product and category levels. By simplifying and standardizing the dataset, we made it easier to create visualizations, perform summaries, and answer the key questions guiding our analysis.

PCIP:

We wanted to change the currency, simple text replacement and rupee to usd conversion. We wanted to remove duplicate data and ensure the data was clean. Priorities necessary columns and removing excess columns such as image links and image files.

FAIR Principles

The dataset we used for this project aligns with what are known as the FAIR Principles. These principles help guide how data should be handled to make it more usable and shareable across different settings.

Findable

You can find this dataset easily on Kaggle, which is a popular platform for sharing and discovering datasets. However, this specific dataset does not come with detailed metadata or column explanations, so some interpretation and exploration are needed to fully understand its structure. Still, it is clearly labeled and easy to locate on the platform.

Accessible

Getting access to this data is straightforward. There are no paywalls or complicated steps, you just need a free Kaggle account. That means anyone interested in exploring the dataset, whether for school or personal projects, can do so easily. Additionally, to ensure our data was accessible, we used qmd files to present the data in a more meaningful way, added comments to have our code be readable, and have our images contained alt text to assist those with screen readers to also have access to our work.

Interoperable

The data is in CSV format, which is a standard that works with all sorts of tools like R, Python, Excel, and more. This makes it convenient to load into whatever program you're most comfortable using.

Reusable

This dataset is shared in a way that encourages others to use it. Because it's clearly structured and documented, it's great for reuse in class projects, personal analysis, or even more advanced research. Additionally, we included the dataset we used and programmed using fairly standard methods and import, these scripts should be easier to alter and reuse.

CARE Principles

The CARE Principles are all about making sure data involving communities is handled ethically and respectfully. While we aren't able to alter the public dataset, we can apply this to our project, to ensure that our data retains some privacy and remains ethical to the community and the subject of the data. We ensured that conclusions derived from this dataset would not harm individuals or communities to the best of our knowledge, instead becoming a possible useful tool for the public and researchers alike. By removing the names of individuals and products from our dataset, we are able to secure anonymity for these people and companies. We also recognize that due to the open-source nature of the code and our not being responsible for the publishing of the dataset itself, this malicious data could be generated again.

Data Visualizations

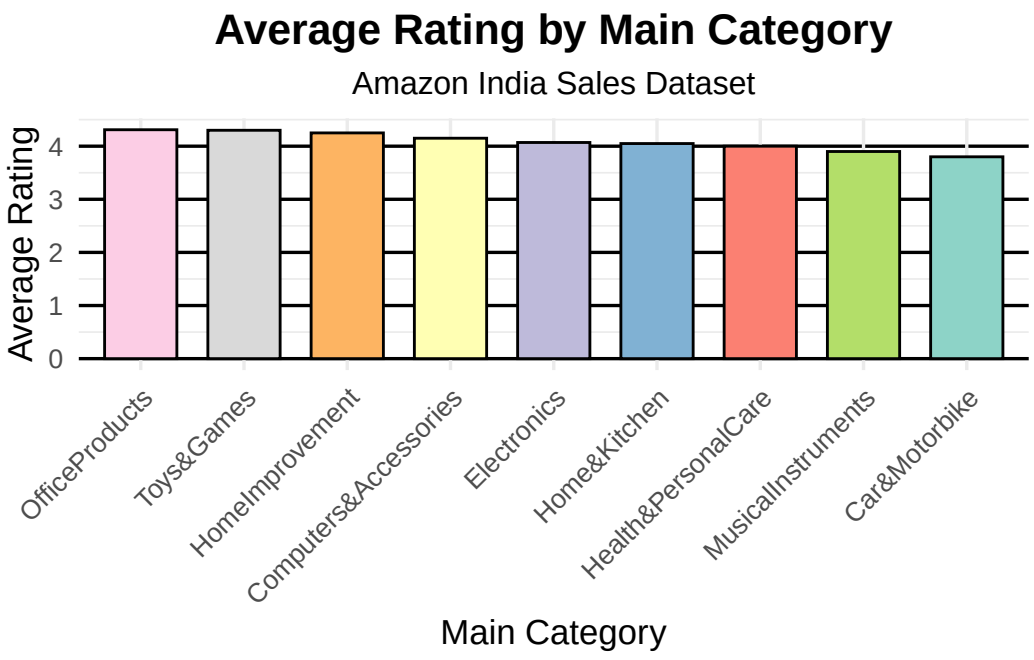
PCIP

We wanted to ensure that each subcategory was counted towards the main category. By doing this, we wanted to include everything before the first occurence of "|", which signified the main category. Including the average rating towards that cateogory along with the acculumated amount of reviews towards the specified main category was necessary.

Table 1: Average Rating and Total Reviews by Main Category

main_category	avg_rating	total_reviews
Electronics	4.07	3810
Home&Kitchen	4.05	3513
Computers&Accessories	4.15	2975
OfficeProducts	4.31	248
HomeImprovement	4.25	16
MusicalInstruments	3.90	16
Car&Motorbike	3.80	8
Toys&Games	4.30	8
Health&PersonalCare	4.00	4

Figure 1: Average Rating by Main Category



Report

With the following data visualizations provided above via Table 1 and Figure 1, provides a comprehensive overview of customer satisfaction across various products. While the bar chart displays average rating for each main category, allowing for easy visual comparison, the table supplements this with total review counts, which gives context to the statistical reliability of those ratings.

Key Observations:

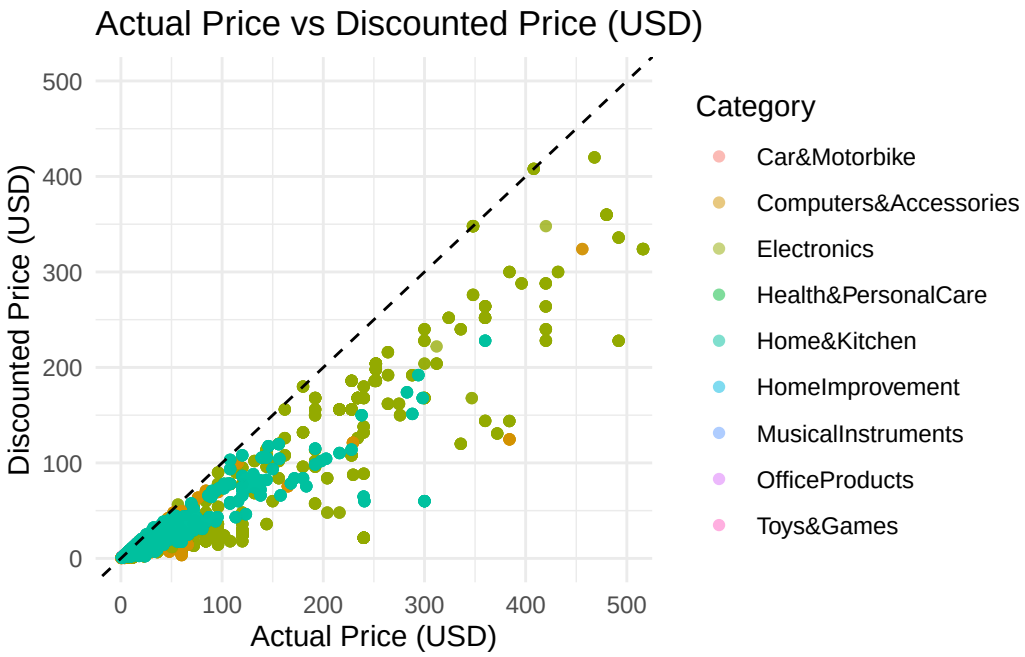
- While *Office Products (4.31)* and *Toy & Games (4.30)* lead in the average rating in customer satisfaction, these categories have relatively low review counts (248 and 8). This may cause future customers to be hesitant as they may be unsure about the reliability of the reviews.
- Categories receiving high volumes in rating count, such as *Electronics (3,810)* and *Home & Kitchen (3,513)*, indicate a larger representative of the customer sentiment and represent a solid satisfaction within high-volume product areas.
- *Car & Motorbike (3.80)* and *Musical Instruments (3.90)* received the lowest ratings across all main categories, though receiving low volumes in review count. Similar to Office Products and Toy & Games, future customers may be hesitant in purchasing; however, a product that received a low rating off the bat may be more likely to cause a chain of unhappy customers.

Final Interpretation:

Categories receiving relatively high average ratings should be interpreted alongside the number of reviews. A product with a high rating and low volume of reviews may be less reliable than a slightly lower rating with a greater volume in reviews. The display of these two visualizations help identify which categories are highly rated, more specifically which ratings are statistically more robust. *Electronics* and *Home & Kitchen* remain at a high volume and consistent with their high ratings.

PCIP

Figure 2: Actual Price vs Discounted Price (USD)



Report

The scatter plot Figure 2 visualizes the relationship between a product's actual price and its discounted price across different categories. Each point represents a product, colored by the main category, and the dashed line (where actual = discounted) acts as a reference for no discount.

Key Observations:

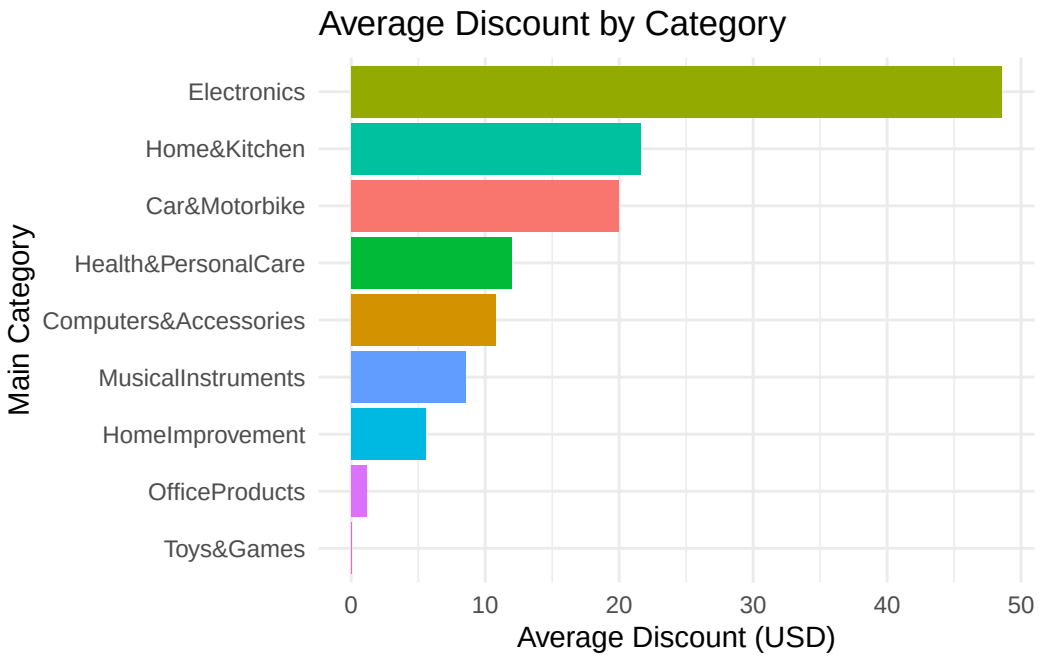
- Most points below the diagonal line, confirming that discounts are widespread.
- Some categories like *Electronics* and *Home & Kitchen* show a wide spread of discount amounts, suggesting more aggressive and varied pricing strategies.
- *Office Products* and *Toys & Games* cluster closer to the diagonal, indicating that these products are less frequently discounted or have smaller discount margins

Final Interpretation:

This plot helps uncover which categories tend to be marketed with more competitive pricing. Categories with wide discount gaps may use pricing as a key strategy to attract buyers, while those closer to full price may rely more on brand value or niche appeal. This aligns with the earlier analysis of customer satisfaction price and perception often go hand-in-hand.

To clearly visualize how discounts vary across categories, we created a bar plot that ranks product categories from those with the highest average discounts to those with the lowest:

Figure 3: Average Discount by Category



Report

Figure 3 presents a horizontal bar chart illustrating the average discount in USD for each main-level product category. By calculating the difference between actual and discounted prices, and then averaging these values per category, the chart allows us to compare how aggressively each category is discounted.

Key Observations:

- Categories at the top of the chart (*Electronics, Home & Kitchen, Car & Motorbikes)* represent those with the largest average discounts, indicating sellers may rely heavily on promotions in these markets to stay competitive
- categories toward the bottom (*Toy & Games, Office Products, Home Improvement)* reflect smaller or less frequent discounts, possibly signaling more stable pricing, stronger brand value, or lower competitive pressure
- Some categories may cluster closely together in average discount, suggesting consistent pricing behavior across similar product types.

Final Interpretation

This visualization highlights how pricing strategies vary across product categories. Categories like *Electronics* and *Home & Kitchen* stand out for offering some of the highest average discounts, and referring to Figure 1 they also maintain high volumes of reviews and consistent average ratings,

as seen in the earlier analysis. This correlation suggests that in certain high-volume categories, deep discounts do not undermine customer satisfaction, in fact, they may enhance it by making popular, well-reviewed products more accessible. On the other hand, categories with lower discounts and fewer reviews may appear more premium or niche but lack the broader customer validation. Together, these insights reinforce that discounting, when paired with quality and demand, can still result in strong customer engagement and satisfaction.

Product Bought and Spending by User

To clearly visualize how users spending habits can be generalized, using the amount of items and the average cost of items they bought, we created a dot plot to measure these two related details in how they are correlated by user.

PCIP

To create the Actual Price vs Discounted Price scatter plot, I first planned the goal of the visualization: to show how much products are discounted across categories and whether certain categories consistently offer larger discounts. Then I coded the plot by cleaning the data, creating a simplified main category variable, and using ggplot() to map actual price against discounted price, adding a dashed line to represent no discount. After generating the initial plot, I polished it by setting axis limits, adjusting the colors, and rotating the labels to make it easier to read.

For this code section I'll need to create a new table for individuals users. To do this I split the users list by commas and the group these rows by their names. This grouping would include key detail including the total amount spend and the number of items bought. Then creating a table which merges these two concepts, probably a dot plot in 2D.

Figure 4: Total Spending and Number of Items Bought by Users

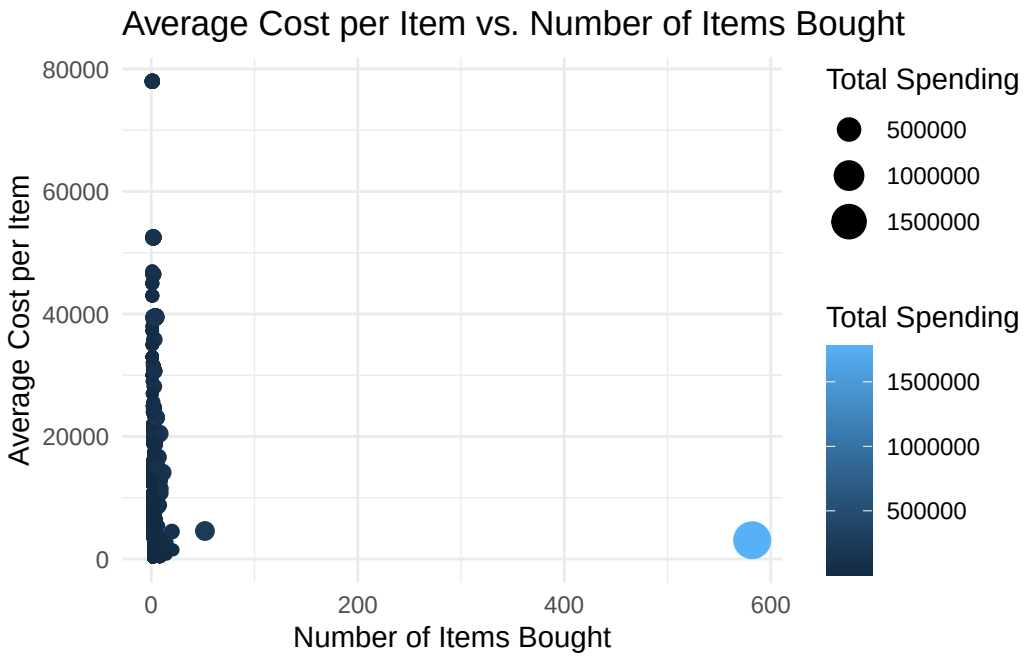
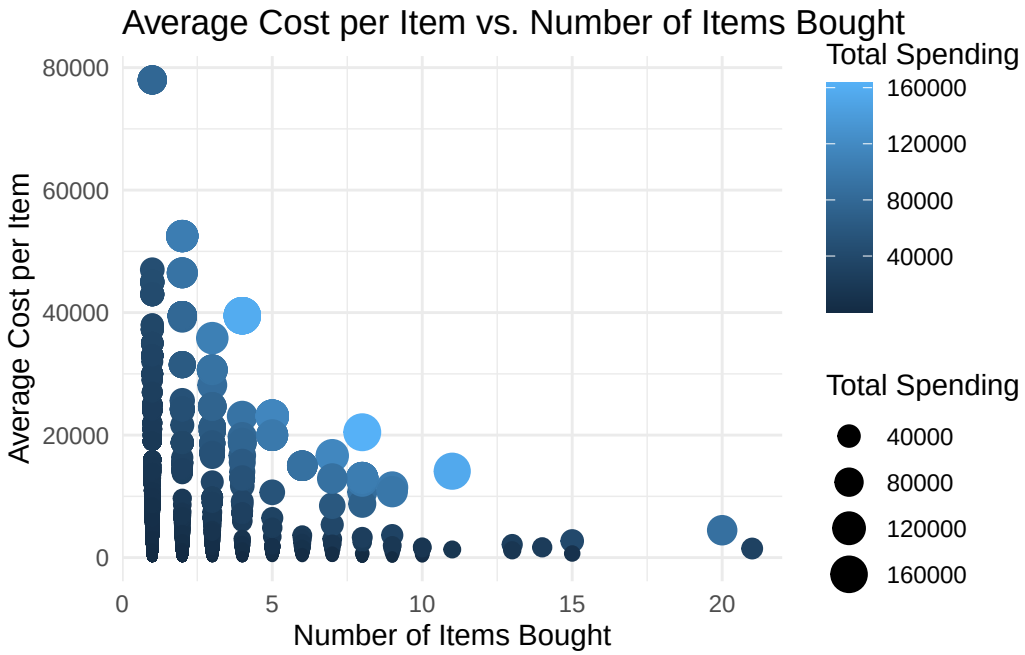


Figure 5: Total Spending and Number of Items Bought by Users



Report

Figure 5 presents a scatter plot showing the relationship between the number of items bought and the average cost per item for each user. Each point represents a user, with the x-axis showing

how many items they purchased and the y-axis showing the average price they paid. The size and color of each point indicate the user's total spending, allowing for visual comparison of spending behavior across users.

Key Observations:

- Most of the highest total spending follows a curve, showing that users buys many cheap items or few expensive items, their total spending evens out.
- These larger spenders buys tens and hundreds of items, labeled as outlines are suspected to be specialty users, like companies, or bot accounts, improving ratings on cheap products to boost the items performance.

Final Interpretation

This visualization highlights how spending behavior varies significantly across individual users. As shown in Figure 5, there is a clear inverse relationship between the number of items purchased and the average cost per item, forming a curved distribution. This suggests two dominant strategies: some users concentrate their spending on a few high-cost items, while others buy many low-cost purchases. Despite these differing approaches, overall spending levels among high-value users remain surprisingly consistent. Some account with exceptionally high rating counts offset a lot of the data as outliers, rating 100's of more items that the next highest reviewer. We suspect that these are bots engaging in mass purchasing to boost product rankings, since these are accounts which comment a lot, not indicative of companies. This is very surprising that we caught this in our findings, and shows utility for these graphs that we hadn't intended. More generally, these findings suggest that high spending is not necessarily tied to a single consumer behavior but may reflect a broader mix of user types and purchasing goals within the platform.

Code Appendix

```
# STEP 1: Load necessary packages
library(dplyr)
library(tidyr)
library(janitor)
library(knitr)
library(kableExtra)
library(ggplot2)

data <- read.csv("amazon.csv")


# STEP 2: Making sure no duplicate entries
View(data)
df_unique <- data %>%
```

```r
  distinct(product_id, .keep_all = TRUE)


conversion_rate <- 0.012  #  1 = $0.012

# STEP 3: Converting to USD given Rupee currency
df_usd <- df_unique %>%
  mutate(
    discounted_price_USD = as.numeric(
        gsub(",","",gsub(" ", "", discounted_price)))*conversion_rate,
    actual_price_USD = as.numeric(
        gsub(",","",gsub(" ", "", actual_price)))*conversion_rate

  )
View(df_usd)

# STEP 4: Tidying Data with case being individual user (user_id)
df_tidy <- df_usd %>%
  mutate(
    rating_count = as.numeric(gsub(",", "", rating_count)),
  ) %>%
  select(product_name, category, discounted_price_USD,
         actual_price_USD, rating, rating_count, user_id) %>%
  separate_rows(user_id, sep = ",")

View(df_tidy)

# STEP 1: Organize products based on main category and calculate their average ratings and accu
df_summary <- df_tidy %>%
  filter(!is.na(rating) & rating != "") %>%
  mutate(
    rating = as.numeric(rating),
    main_category = sapply(strsplit(category, '\\|'), `[`, 1) # Takes main category
  ) %>%
  group_by(main_category) %>%
  summarise(
    avg_rating = round(mean(rating, na.rm = TRUE), 2),
    total_reviews = n()
  ) %>%
  arrange(desc(total_reviews)) # sorts by total reviews

# Display as polished table
kable(df_summary, caption = "Average Rating and Total Reviews by Main Category")
# STEP 1: Create a bar graph of average ratings by category using df_summary
ggplot(df_summary, aes(x = reorder(main_category, -avg_rating), y = avg_rating, fill = main_ca

  geom_bar(stat = "identity", color = "black", width = 0.7, show.legend = FALSE) +
```

```
  scale_fill_brewer(palette = "Set3") +
  labs(
    title = "Average Rating by Main Category",
    subtitle = "Amazon India Sales Dataset",
    x = "Main Category",
    y = "Average Rating"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid.major.y = element_line(color = "black")
  )
# Step 1: Add a simplified top-level product category
# This extracts the first category before the "|" symbol to reduce overplotting and make the pl
df_tidy <- df_tidy %>%
  mutate(main_category = sub("\\|.*", "", category))

# Step 2: Create scatter plot comparing actual and discounted prices
# This plot helps us visualize the discount patterns across product categories.
# Each point represents a product, colored by its main-level category.

ggplot(df_tidy, aes(x = actual_price_USD, y = discounted_price_USD, color = main_category)) +
  geom_point(alpha = 0.5) +  # Use semi-transparent points to reduce overplotting
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "black") +  # Reference li
  coord_cartesian(xlim = c(0, 500), ylim = c(0, 500)) +  # Focus on products under $500
  labs(
    title = "Actual Price vs Discounted Price (USD)",
    x = "Actual Price (USD)",
    y = "Discounted Price (USD)",
    color = "Category"
  ) +
  theme_minimal()


# Calculate and plot the average discount by product category
df_tidy %>%
  # Step 1: Create a new column for discount value in USD
  mutate(discount = actual_price_USD - discounted_price_USD) %>%

  # Step 2: Group data by top-level category
  group_by(main_category) %>%

  # Step 3: Calculate the average discount per category
  summarise(avg_discount = mean(discount, na.rm = TRUE)) %>%
```

```r
  # Step 4: Plot a horizontal bar chart of average discounts
  ggplot(aes(x = reorder(main_category, avg_discount), y = avg_discount, fill = main_category))
  geom_col() +  # Create solid bars to represent average discount
  coord_flip() +  # Flip coordinates for easier reading of category names
  labs(
    title = "Average Discount by Category",
    x = "Main Category",
    y = "Average Discount (USD)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")  # Remove legend for cleaner look since bars are labeled

# Step 1: Calculate the average discount price and number of items bought by user
df_individuals <- df_unique %>%
  separate_rows(user_name,sep=',') %>%
  mutate(
    discounted_price_clean = as.numeric(gsub("[ ,]", "", discounted_price))
  ) %>%
  group_by(user_name) %>%
  summarise(
    total_spending = sum(discounted_price_clean, na.rm = TRUE),
    item_count = n(),
    average_price = total_spending / item_count,
    .groups = 'drop'
  )

# Step 2: Create plot mapping both items and average cost, including coloring by the total spen
ggplot(df_individuals, aes(x = item_count, y = average_price)) +
  geom_point(aes(size = total_spending, color = total_spending)) +
  labs(
    title = "Average Cost per Item vs. Number of Items Bought",
    x = "Number of Items Bought",
    y = "Average Cost per Item",
    color = "Total Spending",
    size = "Total Spending"
  ) +
  theme_minimal()


# Step 3: Remove Outliers, users with ~600 and ~50 items offsetting all data points, removed fo
df_individuals <- df_individuals %>%
  filter(item_count != max(item_count))
df_individuals <- df_individuals %>%
  filter(item_count != max(item_count))

# Step 4: Replot the items with out the outliers
ggplot(df_individuals, aes(x = item_count, y = average_price)) +
```

```
  geom_point(aes(size = total_spending, color = total_spending)) +
  labs(
    title = "Average Cost per Item vs. Number of Items Bought",
    x = "Number of Items Bought",
    y = "Average Cost per Item",
    color = "Total Spending",
    size = "Total Spending"
  ) +
  theme_minimal()
```