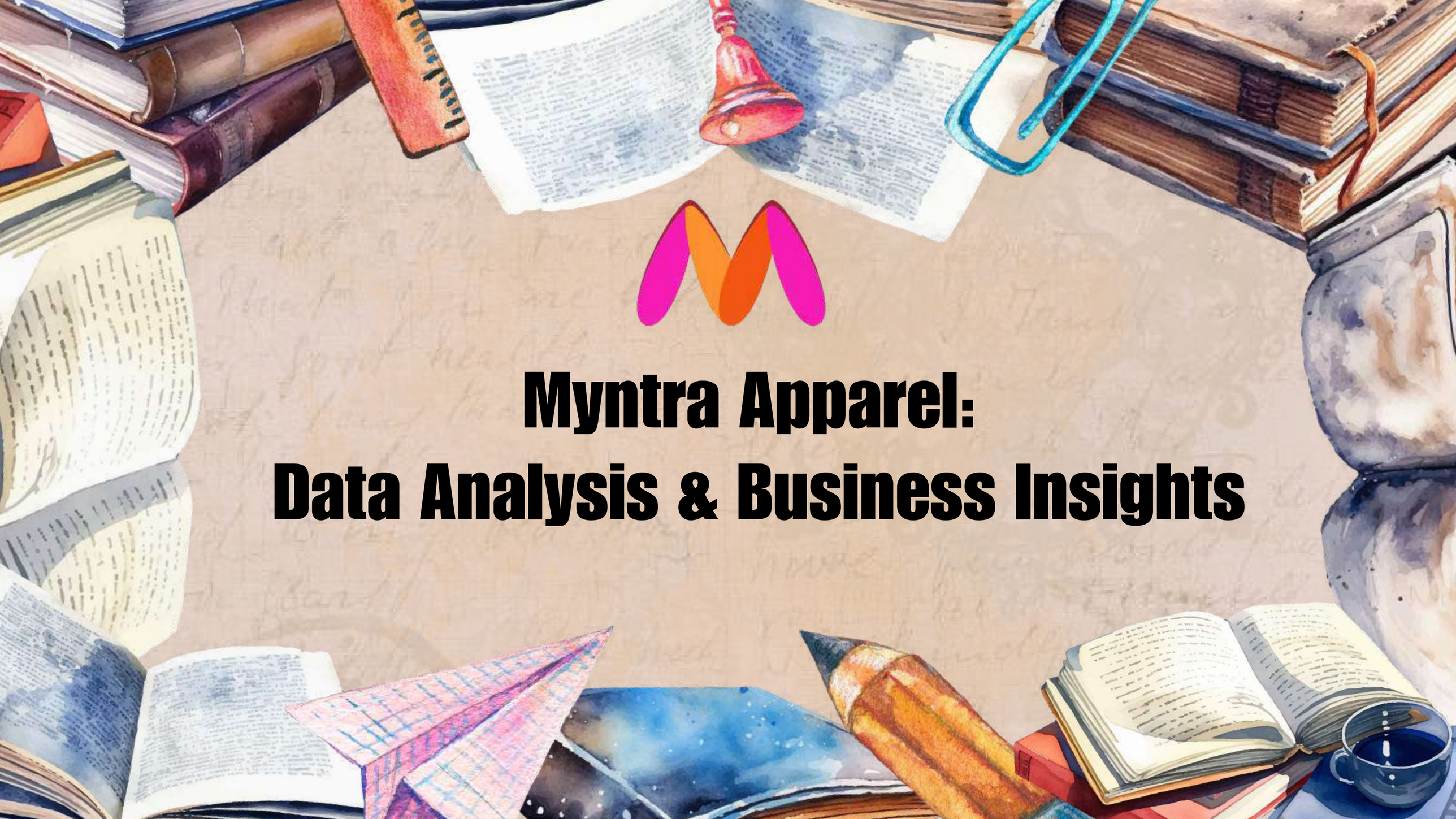


# Myntra Apparel:
# Data Analysis & Business Insights

# PROJECT OVERVIEW

- This project is a case study analyzing a real-world Myntra apparel dataset using Excel.
- The main objective is to transform raw, messy data into actionable business insights related to pricing, discounts, and inventory.
- This case study demonstrates practical problem-solving using key Excel functions, including data cleaning formulas, lookups (VLOOKUP, XLOOKUP), and logical functions (IF, AVERAGEIF).

# PROJECT OBJECTIVE

## Data Pre-processing

Clean and standardize the raw data for complete accuracy

## Data Analysis

Calculate key metrics to gain insights into pricing, discounts, and inventory.

## Data Retrieval

Use lookup functions to find specific product information on demand.

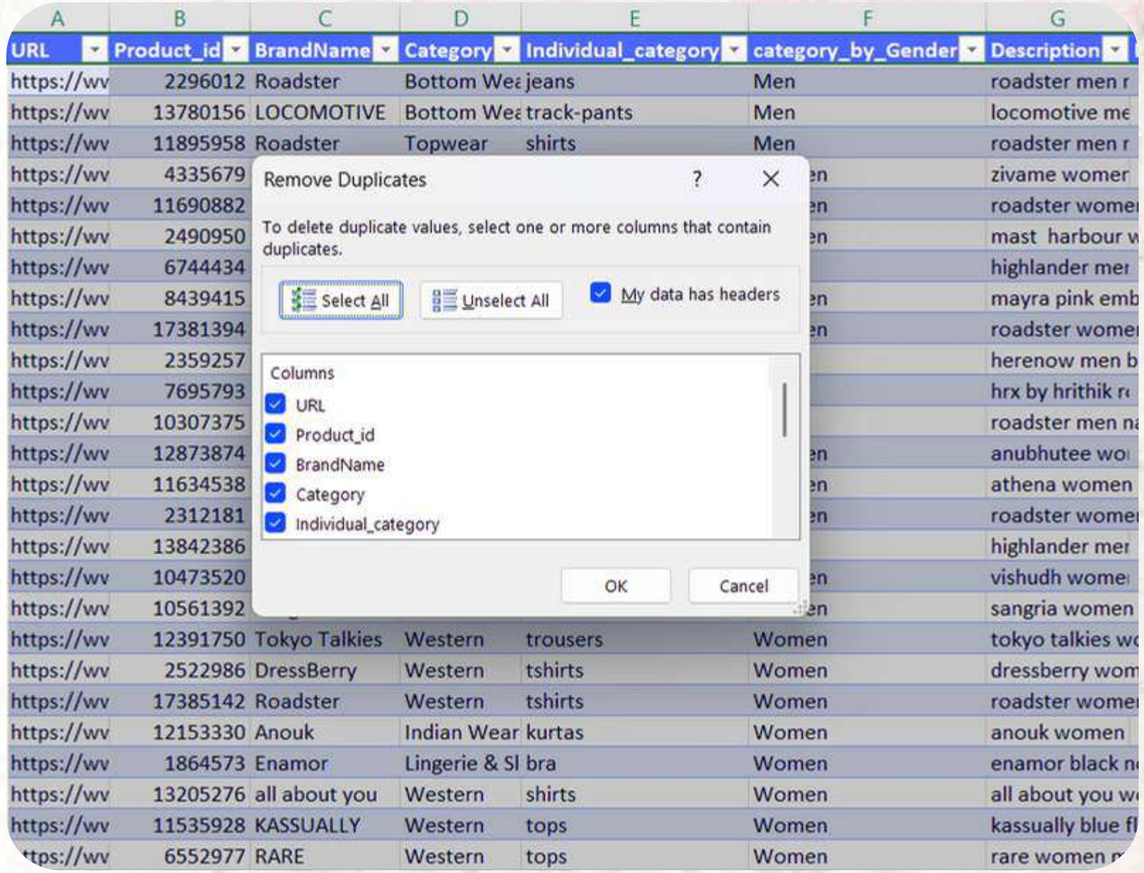
# Summary of Actions Taken

**Part 1: Data Cleaning & Preparation**

*1. I removed all duplicate values from the dataset to ensure each entry was unique.*
*2. I standardized the "DiscountOffer" column so that all values were in a single, uniform format.*
*3. I identified rows where the discount information was null and filled the "DiscountPrice" with the average discount of that product's category.*
*4. I checked the "SizeOption" column for null values and replaced any I found with the text "Not Available."*

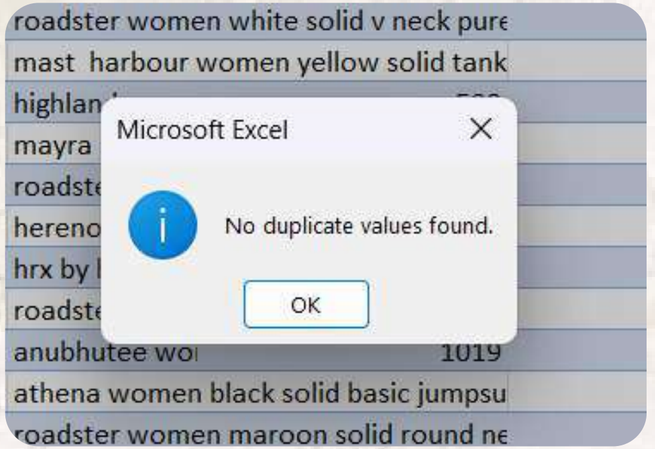
*Sources*

# Check for duplicate values in your dataset and remove them.




Action: For this step, we used the Remove Duplicates tool on the entire dataset to find and delete any identical rows.

Reason: We take this step to make sure we don't count the same item more than once. If we did, our final results would be wrong.



# Standardize the "DiscountOffer" column to a single format, ensuring all values are uniform.

Action: Created a new column, "Discount," using the formula shown. This formula calculates the actual discount amount by subtracting the 'Discounted Price' from the 'Original Price'.

Reason: This new column allows us to directly analyze the monetary value of discounts. It also smartly handles missing discount prices by leaving the field blank, ensuring accuracy without introducing false values.

| K | L | M | N | O | P |
|---|---|---|---|---|---|
| SizeOption | Ratings | Reviews | Discount | | |
| 28, 30, 32, 34, | 3.9 | 999 | =IF([@[Discounted Price (in Rs)]]="","", | | |
| S, M, L, XL | 4 | 999 | [@[OriginalPrice (in Rs)]]- | | |
| 38, 40, 42, 44, | 4.3 | 999 | [@[Discounted Price (in Rs)]]) | | |
| S, M, L, XL, XXL | 4.2 | 9 | IF(logical_test, [value_if_true], **[value_if_false]**) | | |
| XS, S, M, L, XL | 4.2 | 999 | | | |
| XS, S, M, L, XL | 4.4 | 999 | | | |
| 30, 32, 34, 36 | 3.9 | 998 | | | |
| S, M, L, XL | 3.7 | 998 | | | |

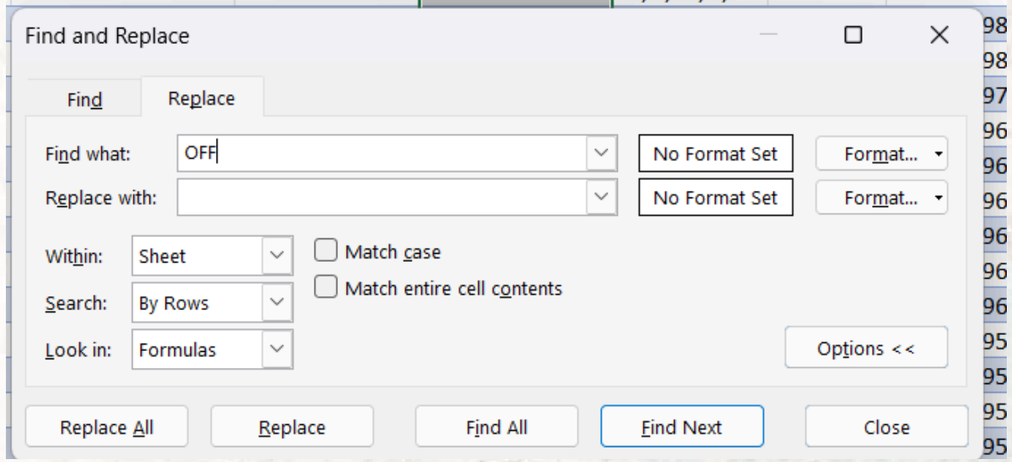
| J | K | L | M | N |
|---|---|---|---|---|
| DiscountOffer | SizeOption | Ratings | Reviews | Discount |
| =N7044 | XS, S, M, L, XL, | 3.8 | 373 | 1140 |
| | XS, S, M, L, XL, | 4 | 205 | 720 |
| | S, M, L, XL, XXL | 4.3 | 141 | 1000 |
| | S, M, L, XL | 4.3 | 98 | 2298 |
| | XS, S, M, L, XL, | 4 | 82 | 1320 |
| | S, M, L, XL, 3XL | 3.8 | 75 | 1300 |
| | S, M, L, XL, XXL | 4.6 | 71 | 2750 |
| | S, M, L, XL, XXL | 4.1 | 71 | 500 |
| | 28, 30, 32, 34, | 4.5 | 50 | 680 |
| | S, M, L, XL | 4.4 | 40 | 1799 |

Action: A two-part filter was applied. First, the DiscountOffer column was filtered to show only blank rows. Second, the Discount column was filtered to show only rows that had a value. A simple formula was then used to copy the calculated amount from the "Discount" column into the empty "DiscountOffer" column.

Reason: To get the final DiscountOffer by safely filling in the blanks with the accurate discount amounts we had already calculated.

# Standardize the "DiscountOffer" column to a single format, ensuring all values are uniform.

| Discounted Price | OriginalPrice | DiscountOffer | SizeOption | Ratings | Reviews |
|---|---|---|---|---|---|
| 824 | 1499 | 45% OFF | 28, 30, 32, 34, | 3.9 | 999 |
| 517 | 1149 | 55% OFF | S, M, L, XL | 4 | 999 |
| 629 | 1399 | 55% OFF | 38, 40, 42, 44, | 4.3 | 999 |
| 893 | 1295 | 31% OFF | S, M, L, XL, XXL | 4.2 | 999 |
| | 599 | 35% OFF | XS, S, M, L, XL | 4.2 | 999 |
| | 599 | 40% OFF | XS, S, M, L, XL | 4.4 | 999 |

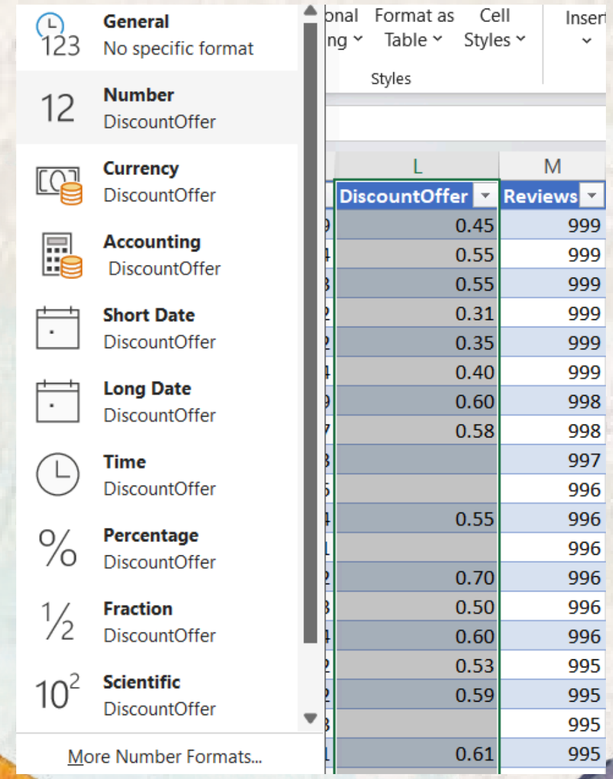



Action: Used the Find and Replace tool to remove all non-numeric text from the DiscountOffer column, such as "OFF", '', Hurry*" , and "Rs.".

Reason: This was a critical step to clean the data, leaving only raw numbers. This prepared the column to be converted into a standard number format for accurate calculations.

Action: The entire DiscountOffer column was converted to a standard Number format.

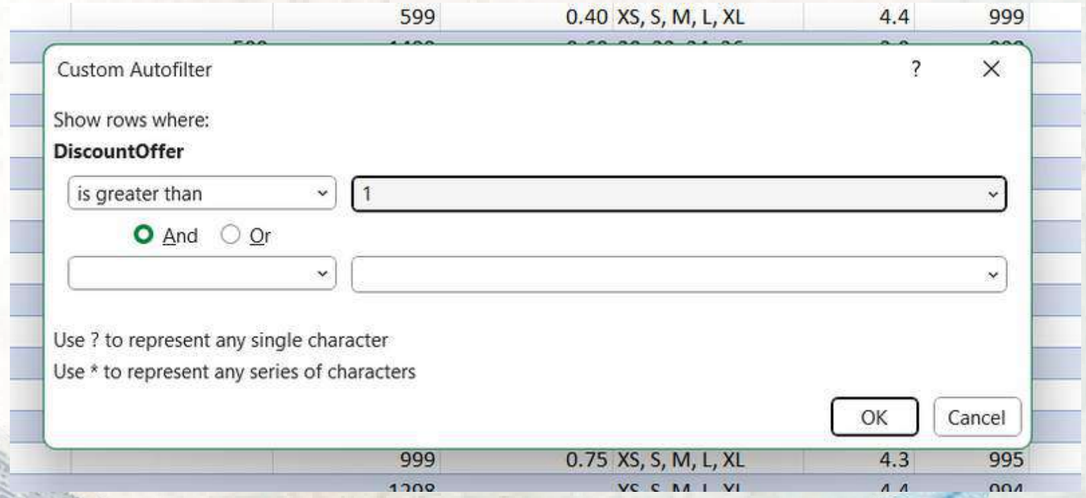
Reason: This was a crucial step to ensure all values were treated consistently as numbers. This allowed us to accurately filter the data and apply formulas, which was necessary for handling the values greater than 1.



# Standardize the "DiscountOffer" column to a single format, ensuring all values are uniform.

Action: A custom filter was applied to the DiscountOffer column to show only the rows where the value is greater than 1.

Reason: This was done to isolate the values that were flat discounts (e.g., 'Rs. 500 OFF') rather than percentages. Since all percentages were now decimals less than 1, this filter allowed us to focus only on the flat discount values that needed to be converted.




| L | M | N | O |
|---|---|---|---|
| Ratings | Reviews | Discount% | |
| 4.2 | 964 | =[@DiscountOffer]/ | |
| 4.3 | 961 | [@[OriginalPrice (in Rs)]] | |
| 4.3 | 952 | | |
| 4.2 | 936 | | |
| 4.3 | 930 | | |
| 4.1 | 926 | | |
| 4.2 | 914 | | |

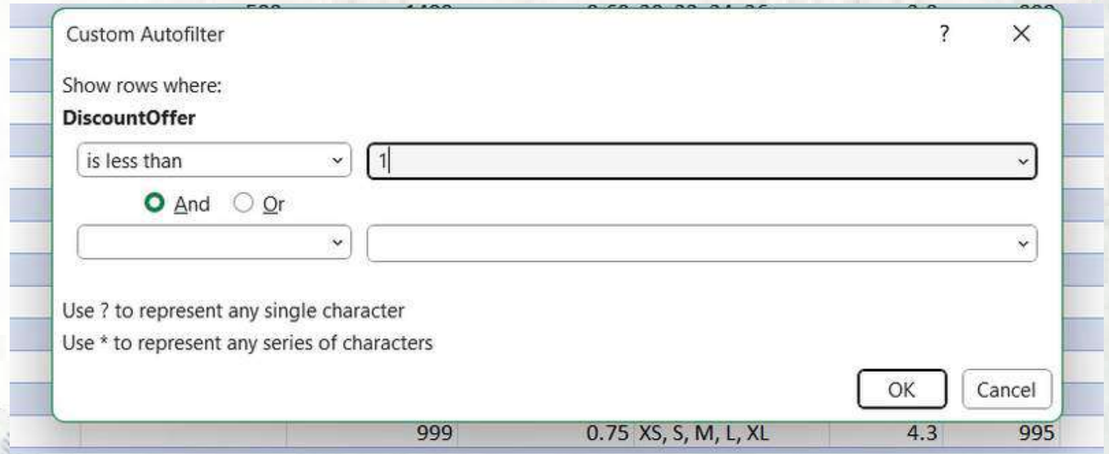
Action: A new column, "Discount%", was created. A formula was then applied to divide the flat DiscountOffer amount by the OriginalPrice (in Rs)

Reason: This formula converts the flat monetary discounts into a percentage format. This was the final step to ensure that every value in the discount column was a uniform percentage, completing the standardization process.

# Standardize the "DiscountOffer" column to a single format, ensuring all values are uniform.

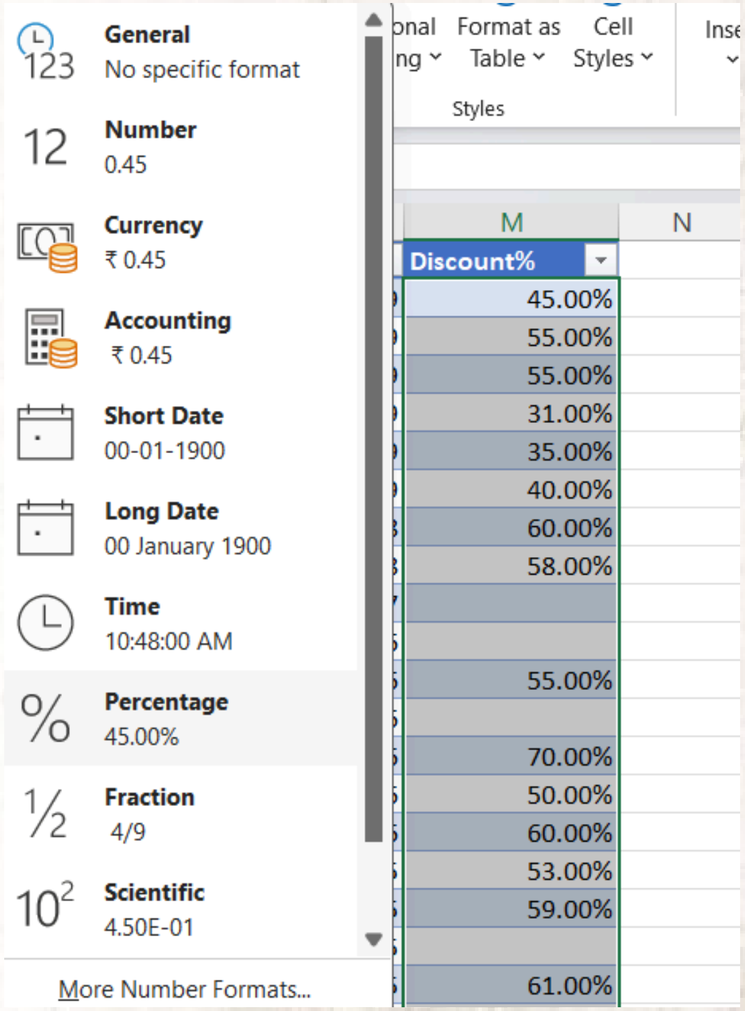
Action: First, we filtered the DiscountOffer column to show only the existing percentage values (those less than 1). Then, for these filtered rows, we used a simple formula to copy them directly into our final "Discount%" column.

Reason: This was done to consolidate all the original percentage values into our final, standardized column. This ensures that the "Discount%" column now contains all discount types, making it the single, complete source for our analysis.




| L | M | N | O |
|---|---|---|---|
| Ratings | Reviews | Discount% | |
| 3.9 | 999 | =J2 | |
| 4 | 999 | | |
| 4.3 | 999 | | |
| 4.2 | 999 | | |
| 4.2 | 999 | | |
| 4.4 | 999 | | |
| 3.9 | 998 | | |
| 3.7 | 998 | | |

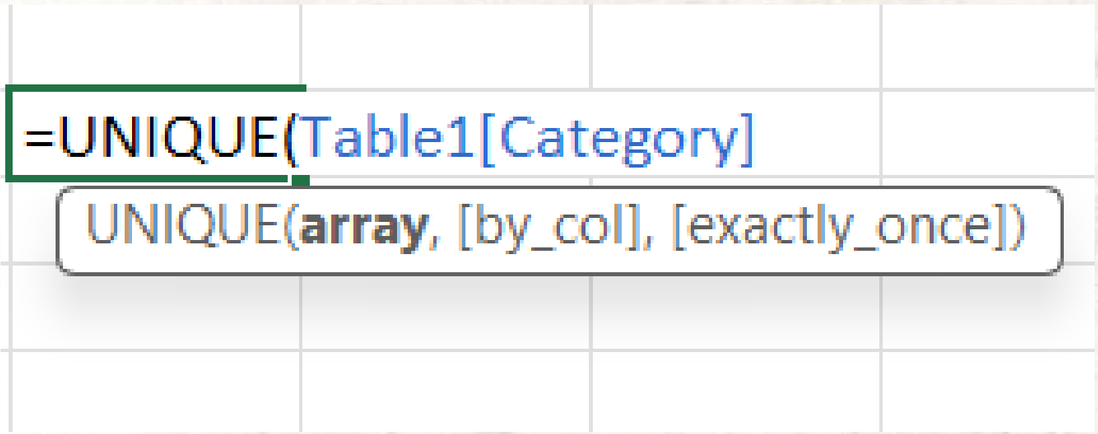
# Standardize the "DiscountOffer" column to a single format, ensuring all values are uniform.

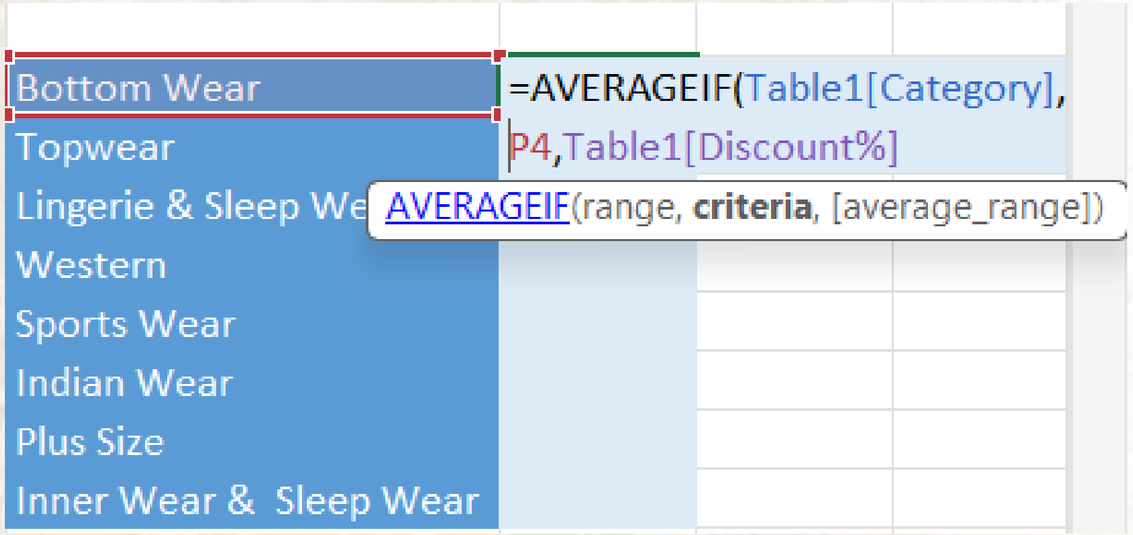



Action: As the final step, the entire "Discount%" column was selected, and the Percentage format was applied.

Reason: This visually converts the decimal values (like 0.55) into an easy-to-read percentage format (like 55%). This completes the process, ensuring all data in the column is now uniform and ready for analysis.

# Identify rows where both "DiscountPrice" and "DiscountOffer" are null and fill the "DiscountPrice" with the average discount price of the respective category.

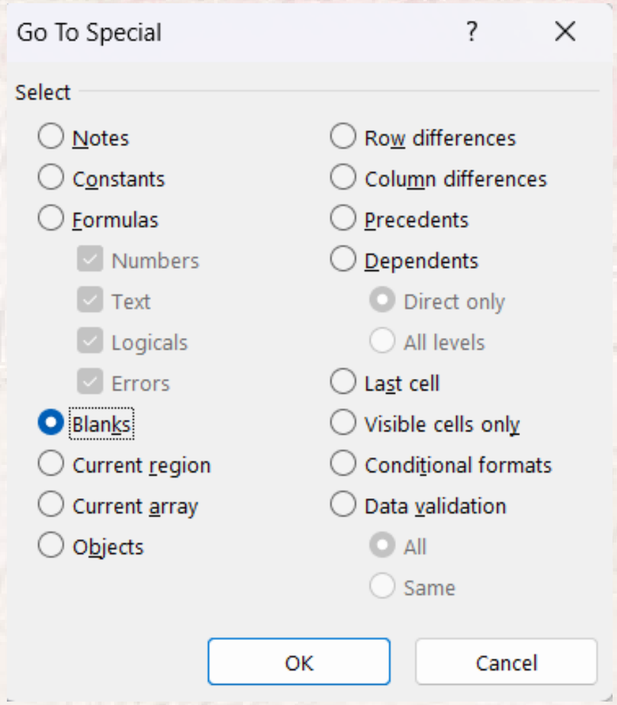


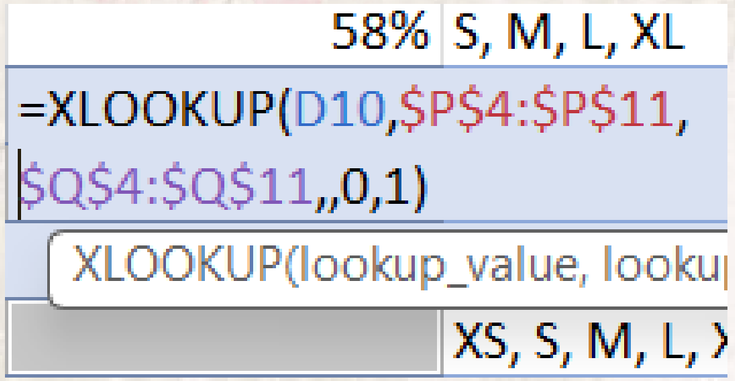




Reason: This two-step process was necessary to get the specific average discount for every category. This ensures that when we fill in missing data, we are using the most relevant and accurate value possible, rather than a single generic average.

Action: First, we used the UNIQUE function to create a list of all distinct product categories. Then, using that list, we applied the AVERAGEIF function to calculate the average discount percentage for each specific category.

# Identify rows where both "DiscountPrice" and "DiscountOffer" are null and fill the "DiscountPrice" with the average discount price of the respective category.



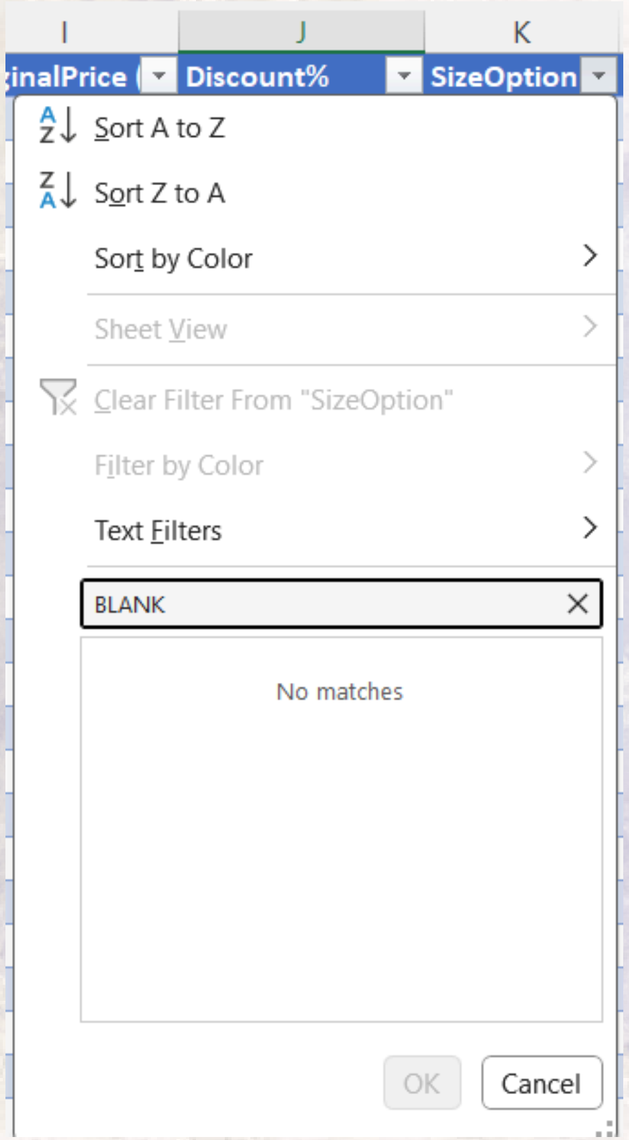

Action: After selecting all blank cells using Go To Special, a single XLOOKUP formula was entered. This formula looks up the product's category and returns the correct average discount from the table we created earlier.

Reason: This technique is a very fast way to fill many empty cells at once. The purpose is to ensure that all remaining blank discount fields are populated with the most logical and accurate estimate—the average of their specific product category.

# Replace all null values in the "SizeOption" column with the text "Not Available".

Action: The SizeOption column was inspected for any blank or null values, as required by the project question.

Finding: No null values were found in the column. Therefore, no action was needed, and the data for this field was confirmed to be 100% complete.



# Summary of Actions Taken

## Part 2: Data Analysis

*1. I calculated the overall average original price for all products that had a rating greater than 4.*

*2. I counted the total number of products that had a discount offer greater than 50% OFF.*

*3. I counted the total number of products that were available in size "M."*

*4. I created a new column and labeled each product as either "High Discount" or "Low Discount" based on its offer.*

# Calculate the overall average original price for products with ratings greater than 4.

| Avg Price of product with rating more than 4 |
| --- |
| =AVERAGEIF(Table1[Ratings],">4", Table1[OriginalPrice (in Rs)] |

AVERAGEIF(range, criteria, **[average_range]**)

| Avg Price of product with rating more than 4 |
| --- |
| 1966.67 |

Insight: This result suggests that Myntra's customers are willing to pay a premium for products with higher ratings, indicating a strong link between perceived quality and price.

# COUNT THE NUMBER OF PRODUCTS WITH A DISCOUNT OFFER GREATER THAN 50% OFF.

| No.of product with discount more than 50% |
| --- |
| =COUNTIF(Table1[Discount%],">50%") |

COUNTIF(**range**, criteria)

| No.of product with discount more than 50% |
| --- |
| 232123 |

Insight: This high number reveals that aggressive, deep discounting is a key part of Myntra's sales strategy, likely used to attract price-sensitive customers and drive sales volume.

# Count the number of products available in size "M."

| Count no.of product available in M size |
| --- |
| =COUNTIF(Table1[SizeOption],"*M*") |

COUNTIF(range, **criteria**)

| Count no.of product available in M size |
| --- |
| 308460 |

Insight: The significant number of products available in size 'M' indicates it is a core, high-demand size. Maintaining stock for this size is crucial for maximizing sales and customer satisfaction.

# Create a new column to label the products as "High Discount" if the discount offer is greater than 50% OFF, otherwise label them as "Low Discount."

**Discount Level**

```
=IF([@[Discount%]]>50%,
"High Discount","Low Discount")
```

IF(logical_test, **[value_if_true]**, [value_if_false])

Insight: Segmenting products this way is a powerful tool for further analysis. It allows the business to compare the sales performance of high-discount items versus low-discount items and better understand the overall pricing strategy.

# Summary of Actions Taken

**Part 3: Data Retrieval and Lookup**

*1. I used the VLOOKUP/XLOOKUP function to find the product brand, price, and rating for the product with ID "11226634".*

*2. I used the INDEX and MATCH functions together to find the "DiscountPrice" for the product with ID "6744434".*

*3. I utilized a nested lookup function to find any column's detail for any given product ID.*

# Use VLOOKUP/XLOOKUP to find the product brand, price, and rating of the product with Product_ID ”11226634”.

| Product_id | BrandName | OriginalPrice (in Rs) | Ratings |
|---|---|---|---|
| 11226634 | =VLOOKUP($P$26,Table1[[#All],[Product_id]:[Reviews]], MATCH(Q$25,Table1[[#Headers],[Product_id]:[Reviews]],0),FALSE) | | |

VLOOKUP(lookup_value, table_array, **col_index_num**, [range_lookup])

| Product_id | BrandName | OriginalPrice (in Rs) | Ratings |
|---|---|---|---|
| 11226634 | Maniac | 1199 | 3.9 |

Action: We used a VLOOKUP formula with MATCH inside it to find the Product ID "11226634" and automatically pull its Brand, Price, and Rating into a summary table.

Reason: This is a powerful way to pull specific information from a large table. Using the MATCH function makes the lookup smarter and more reliable, as it automatically finds the correct column to get the data from.

# Find the "DiscountPrice" for the product with the Product ID "6744434" using the INDEX and MATCH functions.

Action: The INDEX and MATCH functions were used together to look up the Product ID "6744434". The MATCH function first found the correct row, and the INDEX function then retrieved the specific "DiscountPrice" from that row, which was 599.

| Product_id | DiscountPrice (in Rs) |
|---|---|
| 6744434 | =INDEX(Table1[DiscountPrice (in Rs)],MATCH(Q30, Table1[[#Headers],[Product_id]:[Reviews]],0)) |

INDEX(array, **row_num**, [column_num])
INDEX(reference, **row_num**, [column_num], [area_num])

Reason: This step was done to demonstrate the use of INDEX and MATCH, as specifically required by the project. This method is often taught and preferred in data analysis because it is a more powerful and flexible alternative to VLOOKUP.

| Product_id | DiscountPrice (in Rs) |
|---|---|
| 6744434 | 599 |

# Utilize nested lookup to find any column's detail of a product with it's product ID

| Product_id | 11226634 |
|---|---|
| BrandName | =XLOOKUP(S45,Table1[Product_id], |
| | XLOOKUP(R46,Table1[#Headers],Table1,,0),,0) |

| Product_id | 11226634 |
|---|---|
| BrandName | Maniac |

Action: First, Data Validation was used to create interactive dropdown menus. Then, a nested XLOOKUP formula was built to read the selections from these dropdowns and instantly return the specific data point, such as finding that the BrandName for Product ID "11226634" is "Maniac".

Reason: This was done to create a powerful and interactive tool, not just a static answer. It allows a user to easily find any piece of data for any product without changing the formula, which is a highly efficient and advanced solution for data retrieval.

# Case Study Summary & Skills Demonstrated

- *Data Cleaning: Transformed a messy, raw dataset into a clean and reliable source by handling duplicates, standardizing text, and imputing missing data.*
- *Strategic Analysis: Segmented the product catalog by discount level to quantify Myntra's aggressive pricing strategy.*
- *Insight Generation: Analyzed the relationship between product ratings and price to prove that higher-rated items command a premium price.*
- *Advanced Excel: Built dynamic lookup tools using nested XLOOKUP and INDEX/MATCH to demonstrate efficient and robust data retrieval.*

# Thank You

Chandra Prakash Choudhary

LinkedIn.com