VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

LÊ ĐÀO DUY TÂN - 52100104

VÕ ĐÌNH MINH TRÍ - 51900641

TRẦN QUANG LUÂN - 52100254

MIDTERM REPORT

BUSINESS INTELLIGENCE SYSTEMS

Ho Chi Minh City , 2023

VIETNAM GENERAL CONFEDERATION OF LABOR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

LÊ ĐÀO DUY TÂN - 52100104
VÕ ĐÌNH MINH TRÍ - 51900641
TRẦN QUANG LUÂN - 52100254

MIDTERM REPORT

BUSINESS INTELLIGENCE SYSTEMS

Instructor

Ph.D Duong Huu Phuc

Ho Chi Minh City , 2023

EXPRESSING GRATITUDE

First of all, we would like to sincerely thank Ph.D Duong Huu Phuc. During the process of studying business intelligence systems, the teacher dedicatedly guided and supported us to master the necessary issues in this subject. Above all, you have equipped us with enough knowledge to be able to complete this midterm report.

Next, we would like to send my sincere thanks to the Department of Information Technology at Ton Duc Thang University. The Faculty has created all conditions for us to study and research this subject. And especially the teachers in the department are always ready to share useful knowledge to help us complete our midterm report in the best possible way.

Finally, due to limited knowledge, we know that our midterm report has many shortcomings and limitations. We hope for your guidance and contributions to improve our final report. we are more perfect. Wishing all teachers good health.

November 7, 2023, Ho Chi Minh City

Author

Le Dao Duy Tan

Vo Dinh Minh Tri

Tran Quang Luan

THE REPORT IS COMPLETED

AT TON DUC THANG UNIVERSITY

I hereby declare that this is my own research project and is under the scientific guidance of Ph.D Duong Huu Phuc. The research content and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluation were collected by the author from different sources and clearly stated in the reference section.

In addition, the Project also uses a number of comments, assessments as well as data from other authors and other organizations, all with citations and source notes.

If any fraud is detected, I will take full responsibility for the content of my Project. Ton Duc Thang University is not involved in copyright violations caused by me during the implementation process (if any).

November 7, 2023, Ho Chi Minh City

Author

Le Dao Duy Tan

Vo Dinh Minh Tri

Tran Quang Luan

MID-TERM PROJECT

ABSTRACT

The midterm report is based on a sales dataset from the Amazon website. The report is divided into two main sections.

The first section is exploratory data analysis (EDA), which includes data preprocessing and the identification of the best-selling and slowest-selling products

on a scale from small to large.

The second section involves enhancing the dataset along both the x and y axes and addressing issues related to product discounts based on the augmented dataset

TABLE OF CONTENTS

ILLUSTRATION INVENTORY

TABLES DIRECTORY

ABBREVIATIONS CATALOG

LOD :Level of Detail

CHƯƠNG 1. DATA PROCESSING

1.1 Data cleaning

- The programming language used : python
- The library used: pandas
- Code:

```
import pandas as pd

#Read data
data = pd.read_csv(r"path")



# Read data
data = pd.read_csv(r"path")

# View information of dataset
data.info()

#View the first 5 rows of the data set
data.head()



# Check duplicated rows
if data.duplicated().sum() > 0:
    # Delete duplicated rows
    data = data.drop_duplicates()
    print("Sum of duplicated rows deleted : ", data.duplicated().sum())

# Convert the 'no_of_ratings' column to integers
```

```
data['no_of_ratings'] = data['no_of_ratings'].str.replace(',', '', regex=True)
data['no_of_ratings'] = pd.to_numeric(data['no_of_ratings'], errors='coerce')

# Convert the 'ratings' column to float64, but replace invalid values with NaN
data['ratings'] = pd.to_numeric(data['ratings'], errors='coerce')

# Calculate the mean rating excluding NaN values
mean_rating = data['ratings'].mean()

# Replace NaN values with the mean rating
data['ratings'].fillna(mean_rating, inplace=True)

# Check for rows where 'ratings' couldn't be converted to float
invalid_ratings = data[data['ratings'].isna()]



# Change type value for price columns
data['actual_price'] = data['actual_price'].str.replace("₹", '').str.replace(",", '')
data['discount_price'] = data['discount_price'].str.replace("₹", '').str.replace(",", '')
data['actual_price'] = data['actual_price'].astype('float64')
data['discount_price'] = data['discount_price'].astype('float64')

# Mean of actual_price
mean_actual_price = data['actual_price'].mean()
# Replace NaN with the average value
data['actual_price'].fillna(mean_actual_price, inplace=True)
```

```
# Mean discount_price
mean_discount_price = data['discount_price'].mean()
# Replace NaN with the average value
data['discount_price'].fillna(mean_discount_price, inplace=True)

# Calculate the average value (mean) of the no_of_ratings column, eliminating NaN
mean_no_of_ratings = data['no_of_ratings'].mean()

# Replace NaN with the average value
data['no_of_ratings'].fillna(mean_no_of_ratings, inplace=True)

# Save
data.to_csv('dataset_newv2.csv', index=False)
```

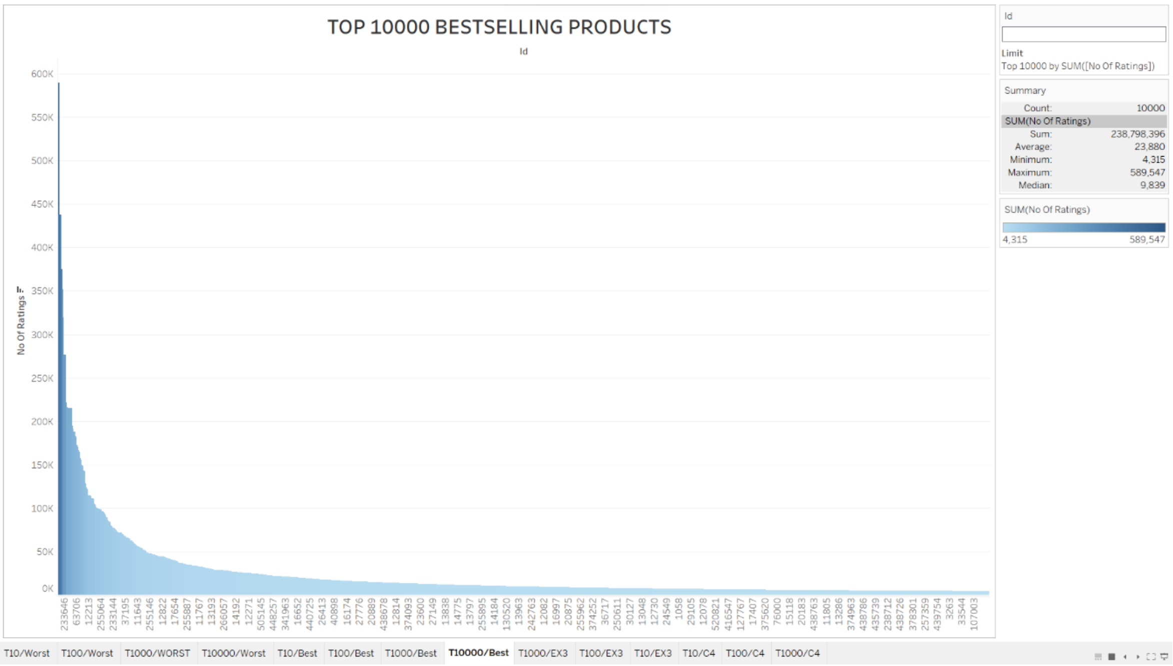
- Describe the data cleaning process: the pandas library is imported and the data from the CSV file is read into a DataFrame called 'data'. Using the info() function, the code displays information about the data set, including the number of rows, the number of columns, and the data type of each column. Then, use the head() function to view the first 5 rows of the data set, helping to check the structure and sample data. Next, the code checks and removes duplicate rows in the data set if any, and prints the number of rows removed. Column 'no_of_ratings' is transformed by removing commas and converting to integer type to normalize the data. The 'ratings' column is converted to float64, but invalid values are replaced with NaN (not a number). The code then calculates the average value of the 'ratings' column by removing the NaN values. The NaN value in the 'ratings' column is replaced by the calculated average value. The code checks for rows in the dataset for which the 'ratings' column cannot be converted to a float number and stores them in the 'invalid_ratings' variable. The columns 'actual_price' and 'discount_price' are converted by removing the currency symbol and comma, then converting to float64. The code calculates the average value of columns 'actual_price' and 'discount_price' and replaces the NaN value in these two columns with the corresponding average value. The NaN value in column 'no_of_ratings' is replaced by the average value of this column. Finally, the processed data is saved into a new CSV file named 'dataset_newv2.csv' without including the index column.

1.2 Evaluate the level of complexity

This data set is moderately complex. With a total of 551,585 rows and 12 columns, the data set is large, which can create some challenges in data management and analysis. Data types varied, including int64 and object (text), and there were null (missing) values in some columns, requiring attention to detail to ensure data was filled in properly. physical. The code performed a series of data preprocessing steps, including removing currency symbols, converting data types, and filling in missing values with the average. These preprocessing steps are necessary to prepare the data for subsequent analysis and modeling. The data set also contains a lot of statistical information such as mean value, percentage of null values, and percentage of duplicate rows, which requires special care when working with it. In summary, this dataset requires a thorough preprocessing and understanding of many aspects of the data. While not overly complex, working with it requires attention to detail to ensure the data is processed and analyzed correctly.

CHƯƠNG 2. EXPLORATORY DATA ANALYSIS

2.1 Find best-selling products in scale of 10, 100, 1K, 10K items, and visualize them



Id

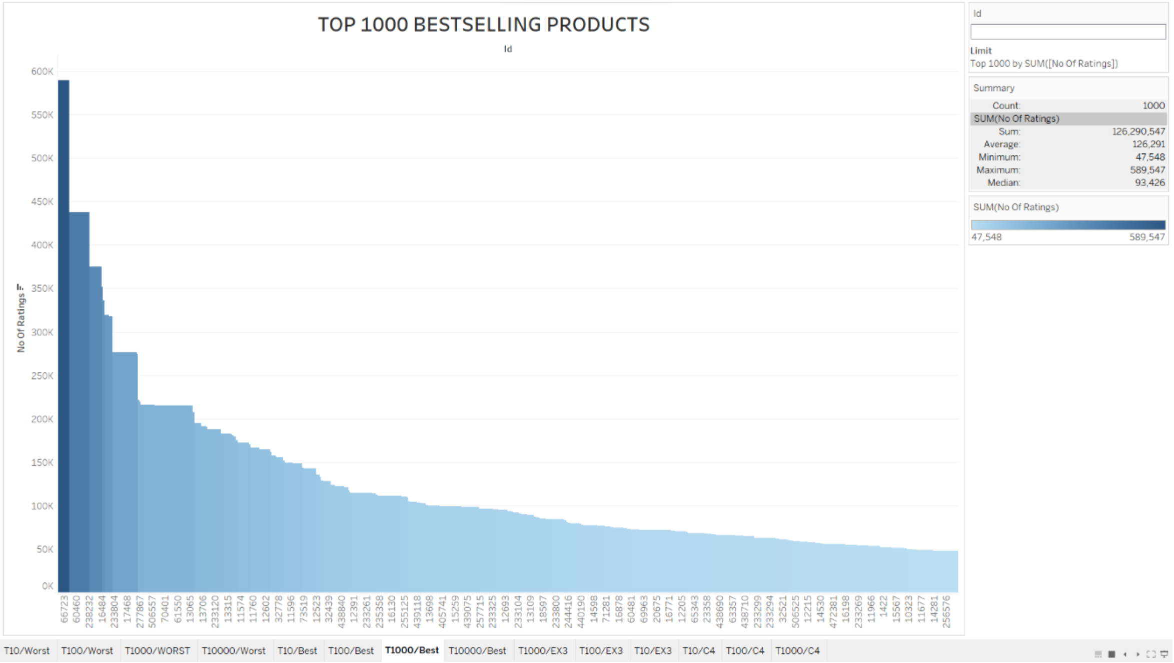
Limit
Top 10000 by SUM([No Of Ratings])

Summary

Count:	10000
SUM(No Of Ratings)	
Sum:	238,798,396
Average:	23,880
Minimum:	4,315
Maximum:	589,547
Median:	9,839

SUM(No Of Ratings)
4,315 589,547

Figure 2.1.1: Visualize top 10000 best-selling products



Id

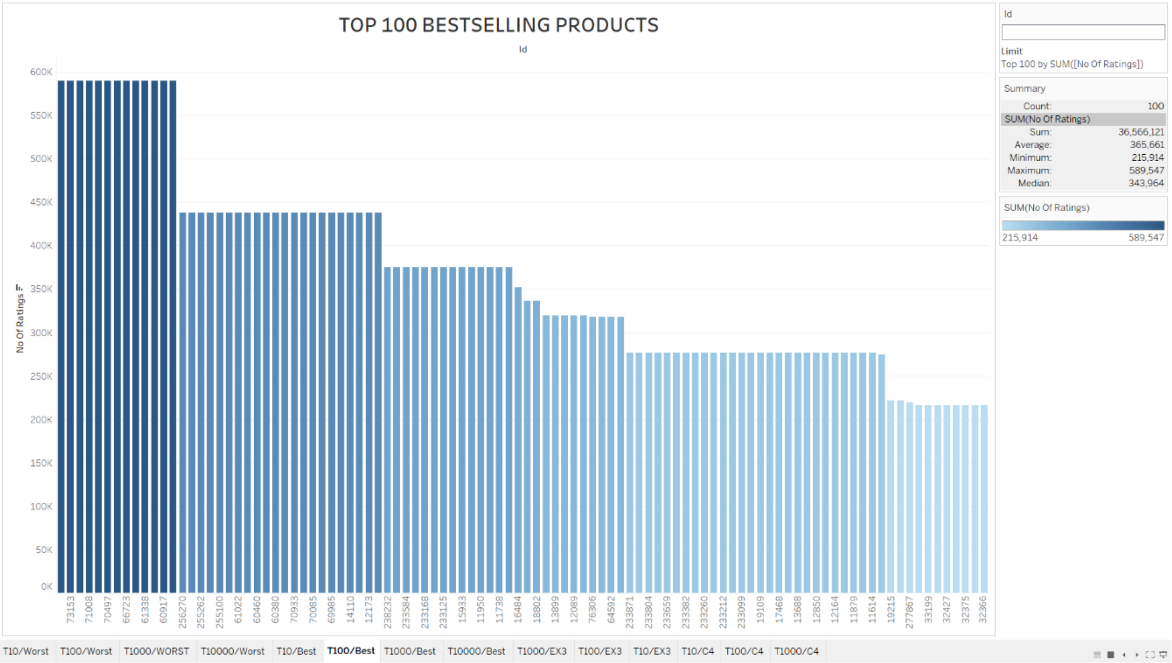
Limit
Top 1000 by SUM([No Of Ratings])

Summary

Count:	1000
SUM(No Of Ratings)	
Sum:	126,290,547
Average:	126,291
Minimum:	47,548
Maximum:	589,547
Median:	93,426

SUM(No Of Ratings)
47,548 589,547

Figure 2.1.2: Visualize top 1000 best-selling products



Id

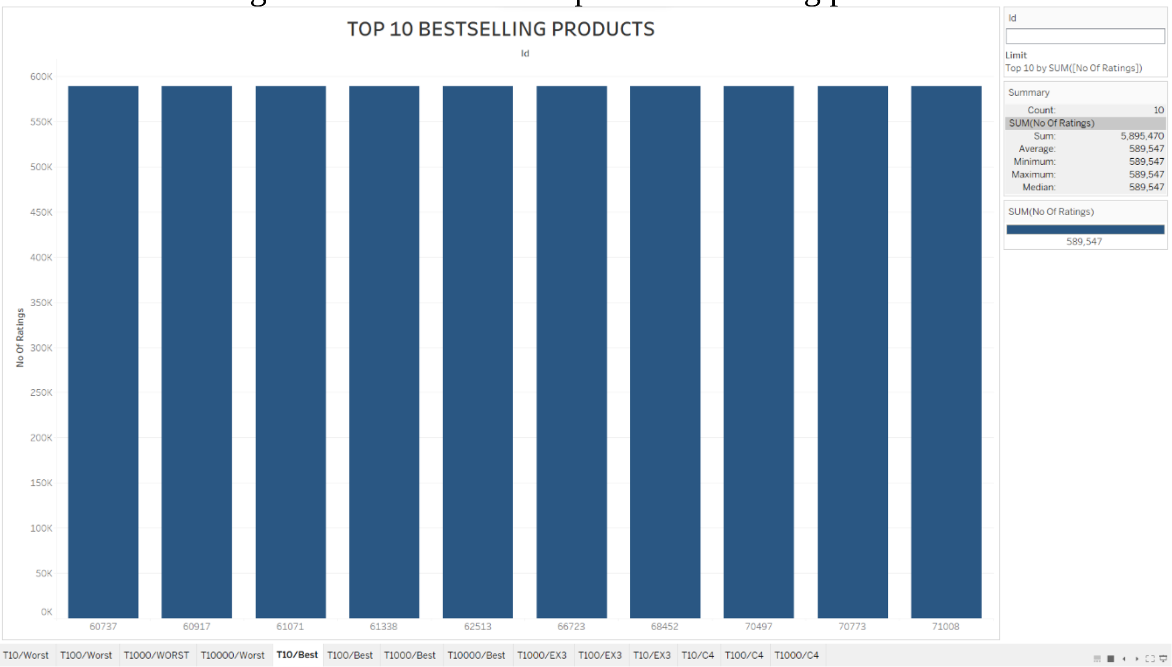
Limit
Top 100 by SUM([No Of Ratings])

Summary

Count:	100
SUM(No Of Ratings)	
Sum:	36,566,121
Average:	365,661
Minimum:	215,914
Maximum:	589,547
Median:	343,964

SUM(No Of Ratings)
215,914 589,547

Figure 2.1.3: Visualize top 100 best-selling products



Id

Limit
Top 10 by SUM([No Of Ratings])

Summary

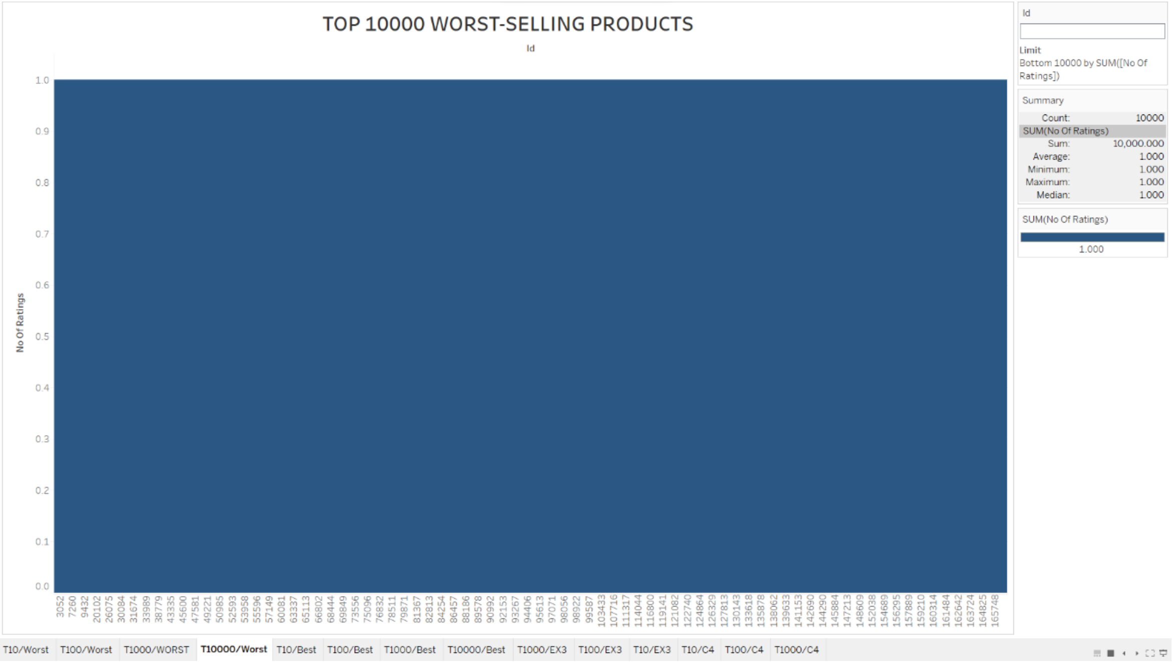
Count:	10
SUM(No Of Ratings)	
Sum:	5,895,470
Average:	589,547
Minimum:	589,547
Maximum:	589,547
Median:	589,547

SUM(No Of Ratings)
589,547

Figure 2.1.4: Visualize top 10 best-selling products

Comment on the chart: the distribution of the data set has large and uneven variations. Some products have significantly more sales than other products.

2.2 Find worst-selling products in scale of 10, 100, 1K, 10K items, and visualize them



Id

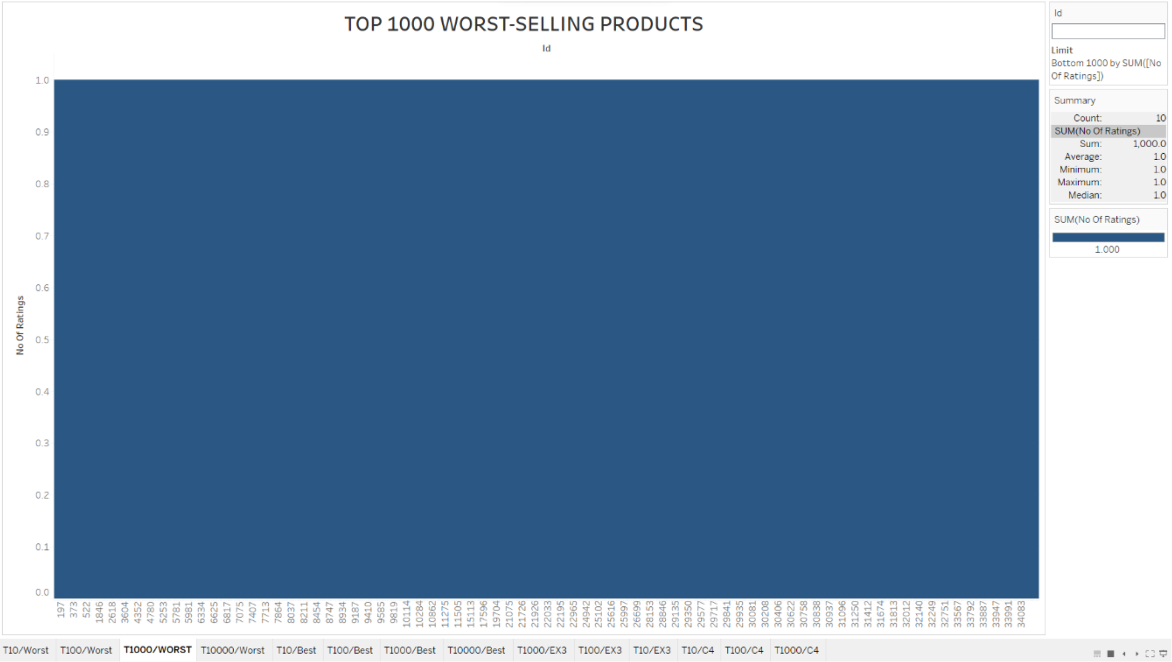
Limit
Bottom 10000 by SUM([No Of Ratings])

Summary

Count:	10000
SUM(No Of Ratings)	
Sum:	10,000.000
Average:	1.000
Minimum:	1.000
Maximum:	1.000
Median:	1.000

SUM(No Of Ratings)
1.000

Figure 2.2.1: Visualize top 10000 worst-selling products



Id

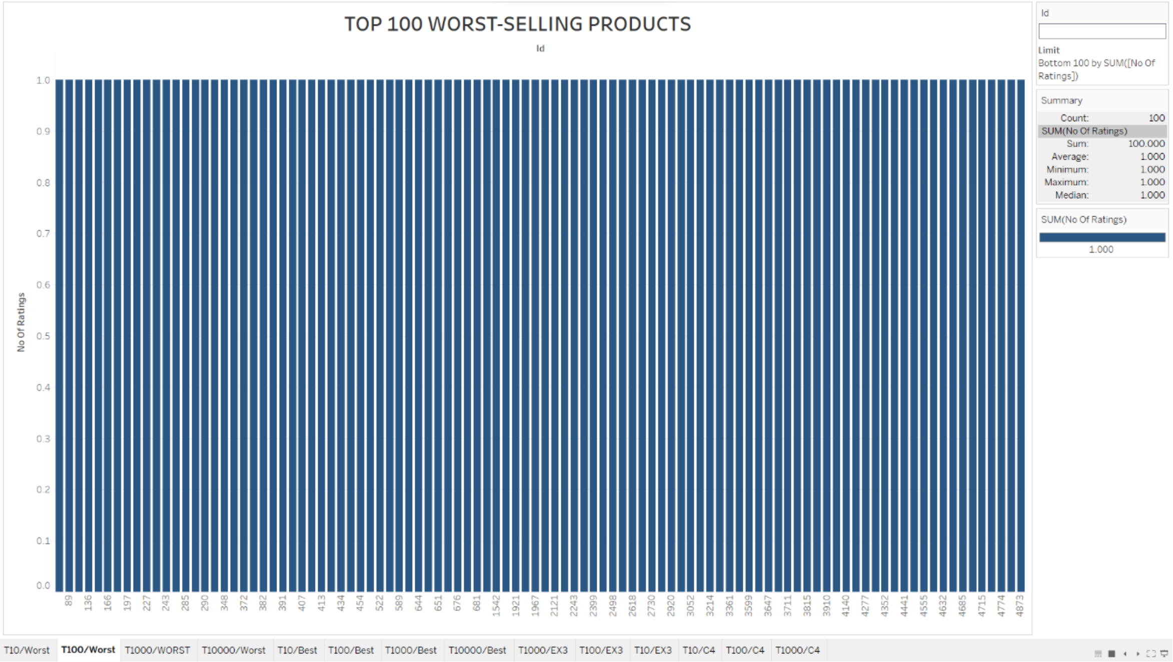
Limit
Bottom 1000 by SUM([No Of Ratings])

Summary

Count:	10
SUM(No Of Ratings)	
Sum:	1,000.0
Average:	1.0
Minimum:	1.0
Maximum:	1.0
Median:	1.0

SUM(No Of Ratings)
1.000

Figure 2.2.2: Visualize top 1000 worst-selling products



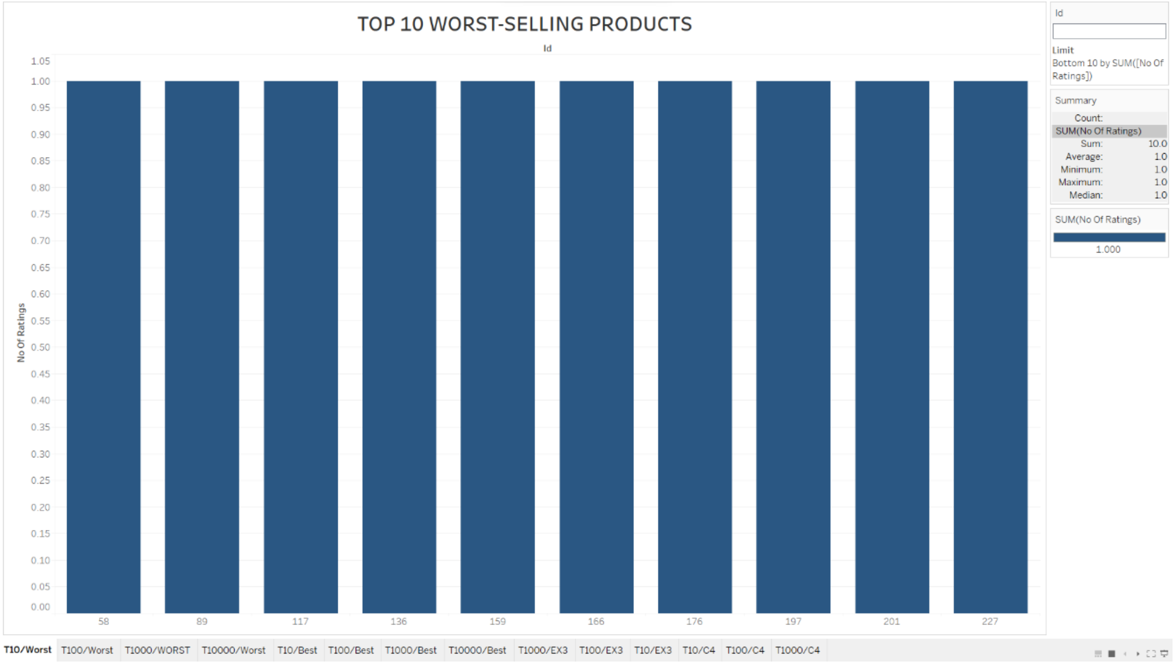
Id

Limit
Bottom 100 by SUM([No Of Ratings])

Summary

```
        Count:       100
SUM(No Of Ratings)
          Sum:   100.000
      Average:     1.000
      Minimum:     1.000
      Maximum:     1.000
       Median:     1.000
```

Figure 2.2.3: Visualize top 100 worst-selling products



Id

Limit
Bottom 10 by SUM([No Of Ratings])

Summary

```
        Count:
SUM(No Of Ratings)
          Sum:  10.0
      Average:   1.0
      Minimum:   1.0
      Maximum:   1.0
       Median:   1.0
```

Figure 2.2.4: Visualize top 10 worst-selling products

Comment on the chart: the chart shows the values are all 1. This means that all the products in this list sell very poorly or have no interest. This shows that there are some products in the data set that are having sales difficulties and need to be reviewed or improved to ensure competitiveness and profitability.

2.3 Find and explain the relationship between dataset variables

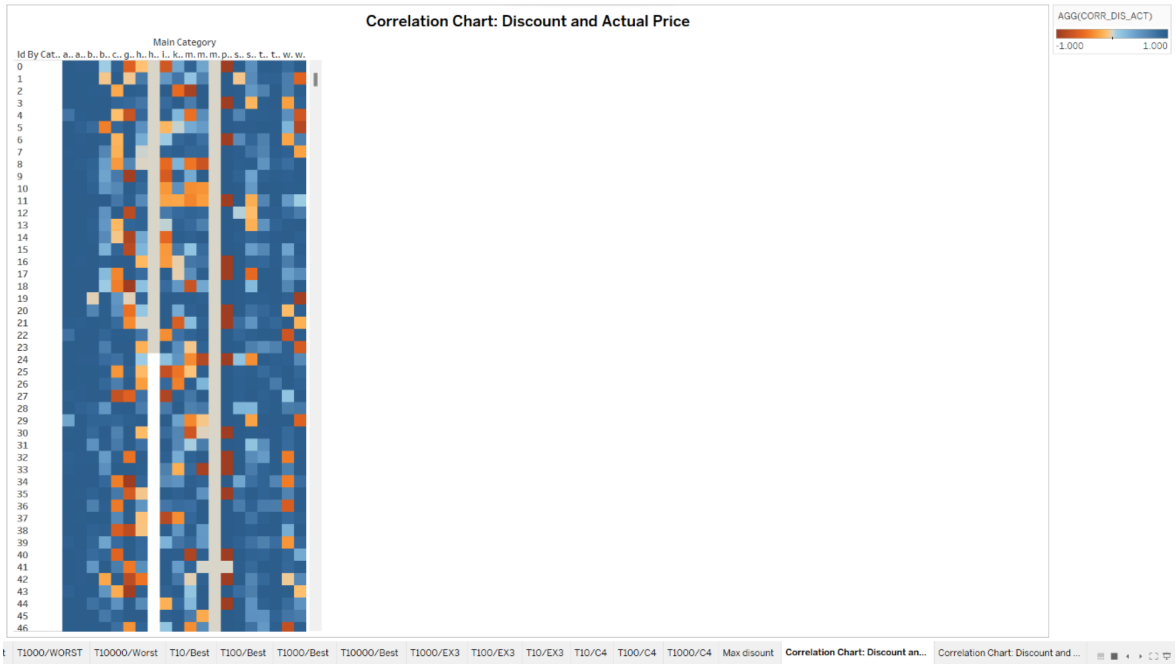


Figure 2.3.1: Correlation Chart: Discount and Actual Price

- If CORR_DIS_ACT = 1: This shows a perfect linear relationship between discount and actual price. As discounts increase, actual prices also increase in an even manner. That is, as the discount increases, the actual price also increases by exactly the same proportion.
- If CORR_DIS_ACT = -1: This also shows a perfect linear relationship, but in opposite direction. As discounts increase, actual prices decrease in an even manner.
- If CORR_DIS_ACT = 0: This shows that there is no linear relationship between discount and actual price. An increase or decrease in price does not affect the actual price, or the relationship between them is non-linear.

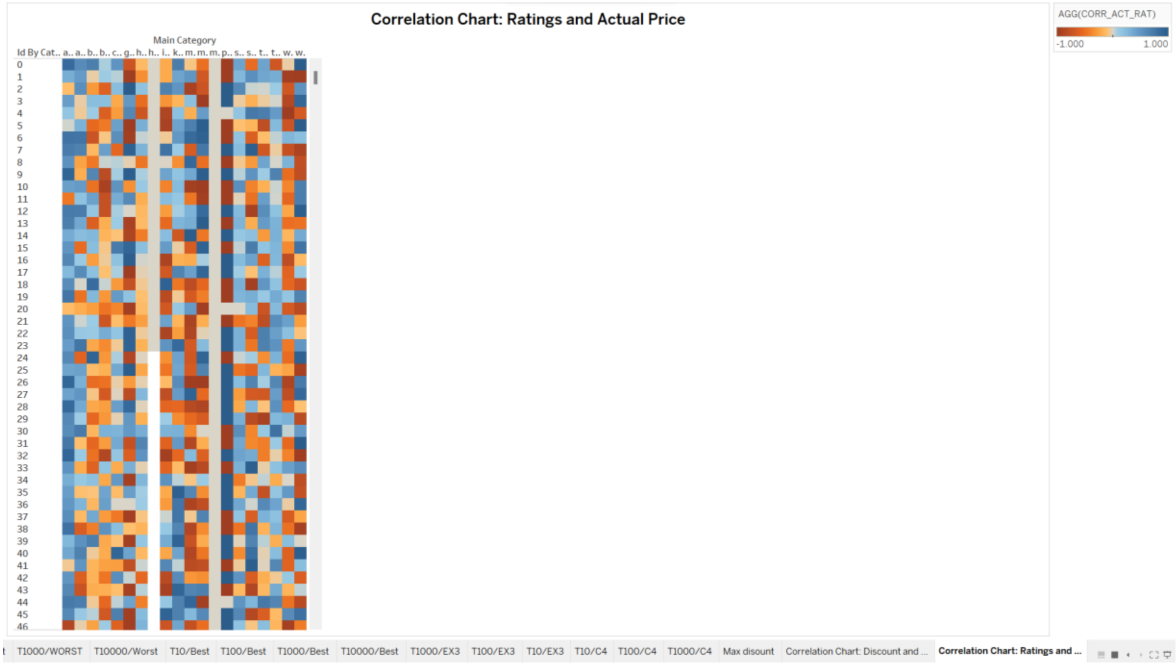


Figure 2.3.2: Correlation Chart: Ratings and Actual Price

- If CORR_ACT_RAT has a positive value close to 1 (for example, 0.9 or 0.95), this indicates a strong positive correlation between ratings and actual price. This means that when the actual price increases, the ratings also increase, and when the actual price decreases, the ratings also decrease. In this case, there is a strong positive relationship between the two variables.
- If CORR_ACT_RAT has a negative value near -1 (e.g. -0.9 or -0.95), this indicates a strong negative correlation between ratings and actual price. This means that when actual price increases, ratings decrease, and when actual price decreases, ratings increase. In this case, there is a strong negative relationship between the two variables.
- If CORR_ACT_RAT has a value close to 0, this shows that there is no absolute relationship between ratings and actual price. In this case, the two variables are not significantly correlated with each other.

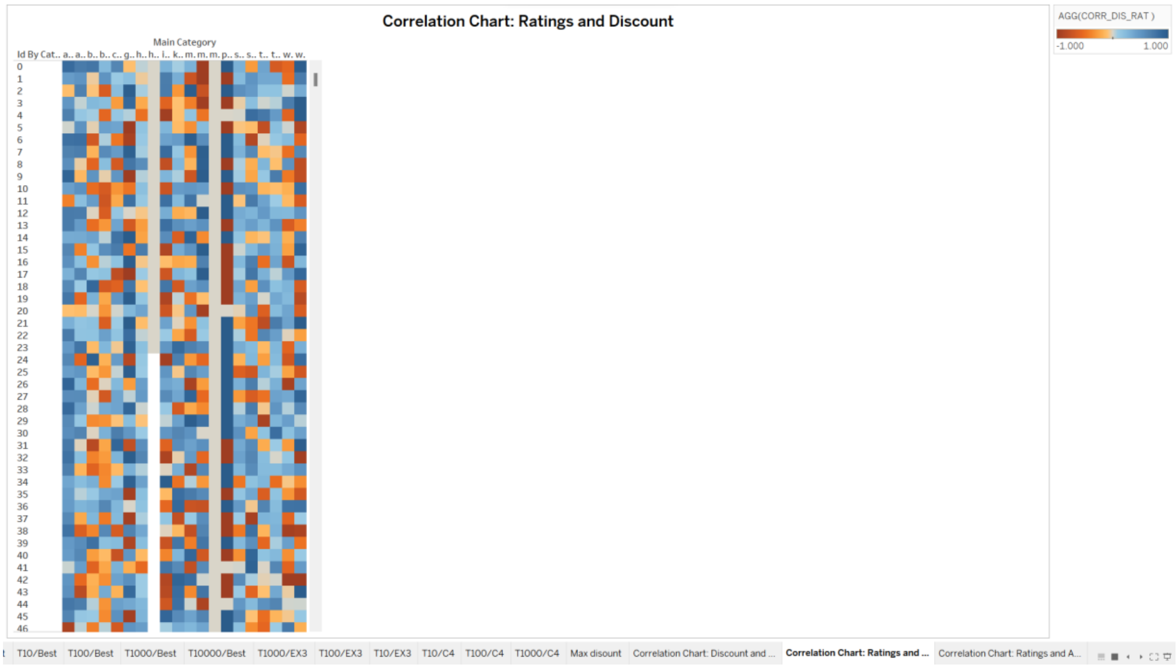

Correlation Chart: Ratings and Discount
Main Category
Id By Cat.. a.. a.. b.. b.. c.. g.. h.. h.. i.. k.. m. m. m. p.. s.. s.. t.. t.. w. w.
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
AGG(CORR_DIS_RAT)
-1.000
1.000
T10/Best
T100/Best
T1000/Best
T10000/Best
T1000/EX3
T100/EX3
T10/EX3
T10/C4
T100/C4
T1000/C4
Max disount
Correlation Chart: Discount and ...
Correlation Chart: Ratings and ...
Correlation Chart: Ratings and A...

Figure 2.3.3: Correlation Chart: Ratings and Discount

- Positive Correlation: When CORR_RAT_DIS has a positive value, this shows a positive correlation between ratings and discounts. That is, when discounts increase, ratings or reviews also increase. This demonstrates buyers' preference for discounted products.
- Negative Correlation: When CORR_RAT_DIS has a negative value, this indicates a negative correlation between ratings and discounts. That is, as discounts increase, ratings or reviews decrease. This may indicate that buyers are unhappy with price reductions that may affect product quality, or that they do not trust price reductions.
- Correlation coefficient close to 0 (No Correlation): When CORR_RAT_DIS is close to 0, this shows that there is no significant correlation between ratings and discounts. This can happen when ratings are not dependent on product discounts.

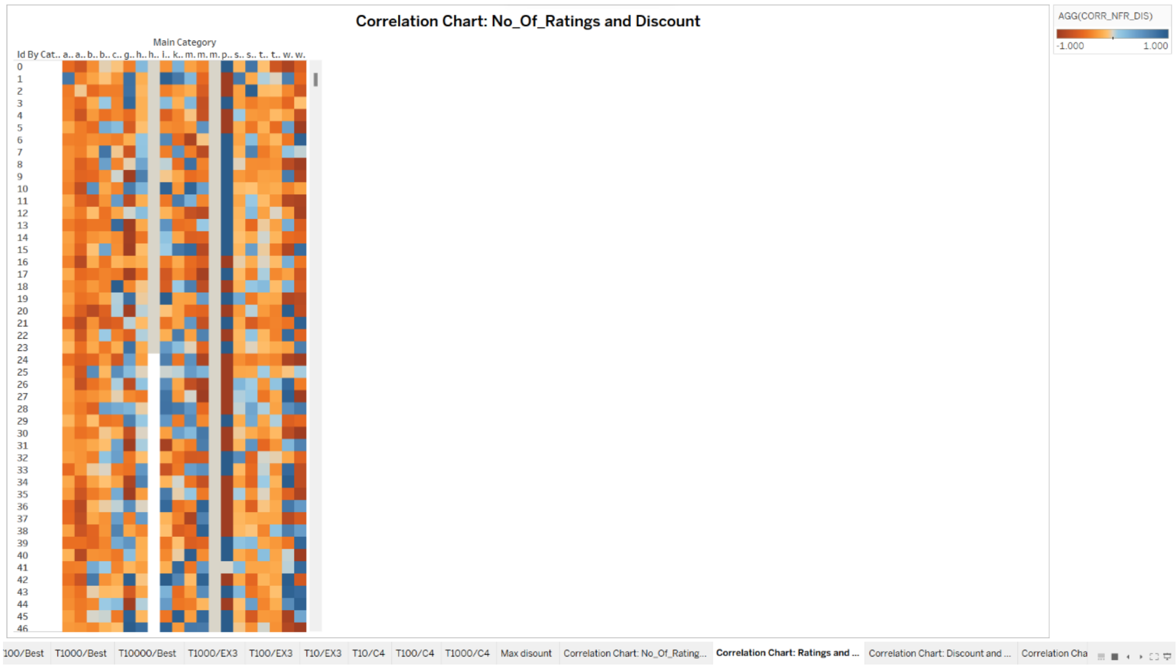


Figure 2.3.4: Correlation Chart: No_Of_Ratings and Discount

- When CORR_NFR_DIS has a positive value near 1 (0.9 to 1), it shows that there is a strong positive correlation between discounts and number of reviews. This means that when a product has a discount, the number of reviews increases, possibly because the discount attracts more buyers.
- When CORR_NFR_DIS has a negative value near -1 (-0.9 to -1), it shows that there is a strong negative correlation between discount and number of reviews. This means that when a product has a discount, the number of reviews decreases. This can happen if a discount is made on poor product quality or the product does not meet the buyer's expectations.
- When CORR_NFR_DIS has a value close to 0, it shows that there is no significant correlation between discounts and number of reviews. Discounts do not have a large effect on the number of reviews.

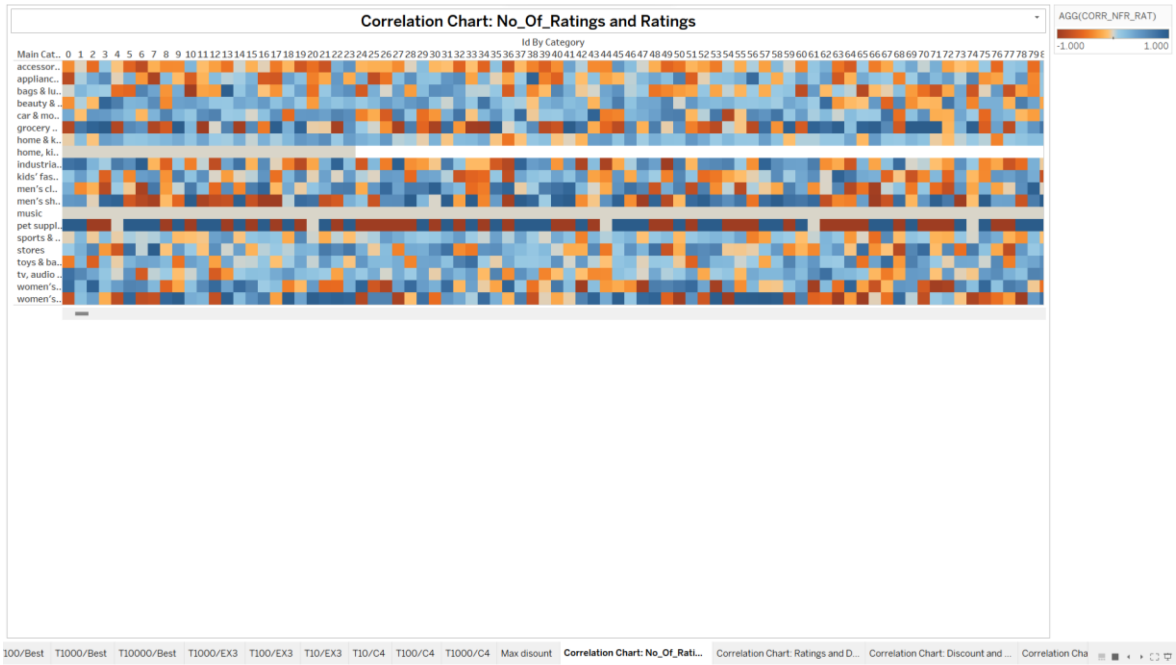
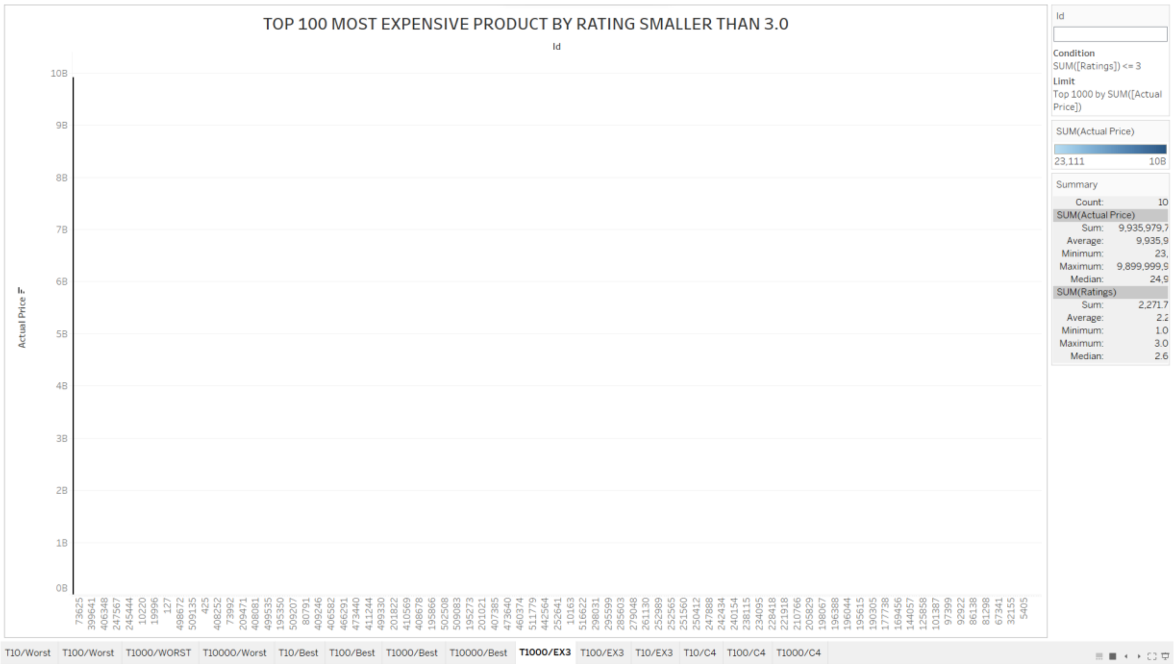


Figure 2.3.5: Correlation Chart: No_Of_Ratings and Ratings

- Value close to 1: This is a strong positive correlation. This means that when "no_of_ratings" increases, "ratings" also increases. Specifically, when the CORR_NFR_RAT coefficient is close to 1, it means that the number of reviews (no_of_ratings) and rating points (ratings) increase together.
- Values close to -1: This is a strong negative correlation. That means when "no_of_ratings" increases, "ratings" decreases and vice versa. Specifically, when the C CORR_NFR_RAT coefficient is close to -1, it means that the number of reviews (no_of_ratings) and rating points (ratings) are negatively correlated.
- Value near 0: This is a weak correlation or no correlation. That is, there is no clear linear relationship between the two variables.

2.4 Find products which are most expensive but have ratings lower than 3.0, in scale 10, 100, 1K items



Id

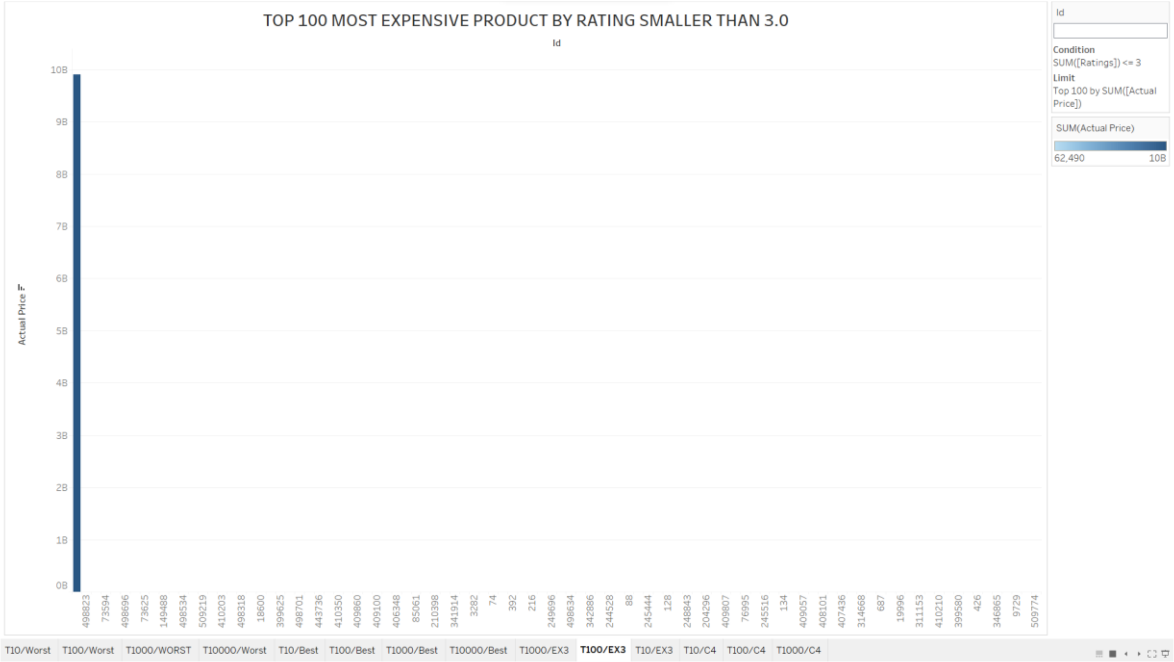
Condition
SUM([Ratings]) <= 3
Limit
Top 1000 by SUM([Actual Price])

SUM(Actual Price)
23,111 10B

Summary

Count:	10
SUM(Actual Price)	
Sum:	9,935,979,7
Average:	9,935,9
Minimum:	23,
Maximum:	9,899,999,9
Median:	24,9
SUM(Ratings)	
Sum:	2,271.7
Average:	2.2
Minimum:	1.0
Maximum:	3.0
Median:	2.6

Figure 2.4.1: Visualize top 1000 products are most expensive but have ratings lower than 3.0

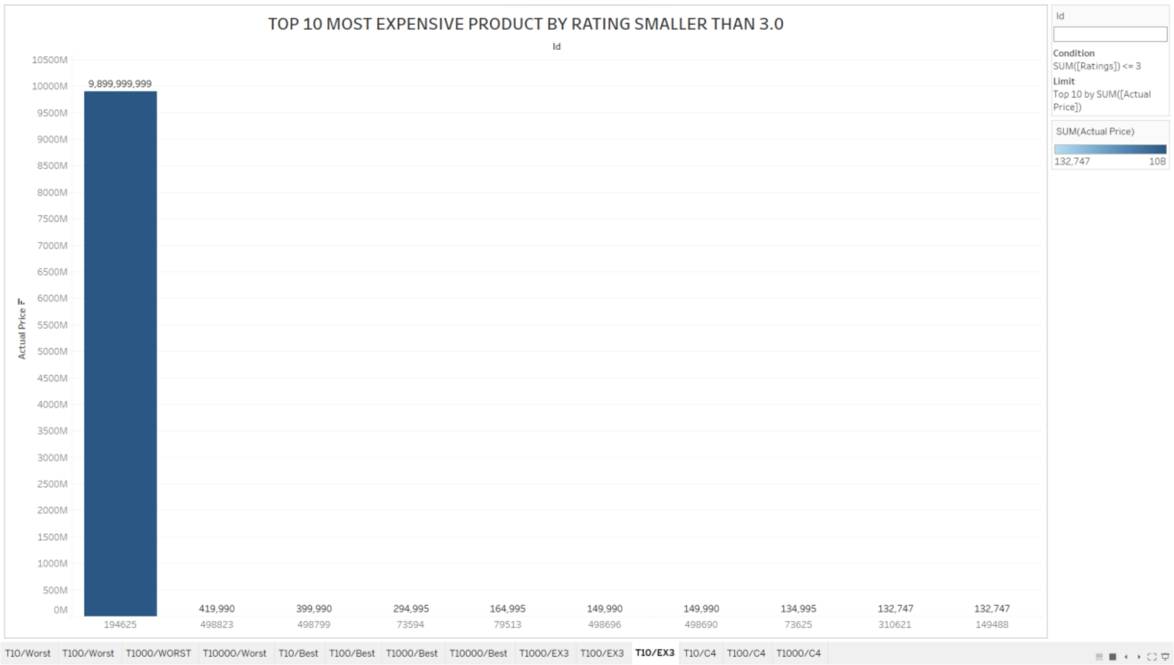


Id

Condition
SUM([Ratings]) <= 3
Limit
Top 100 by SUM([Actual Price])

SUM(Actual Price)
62,490 10B

Figure 2.4.2: Visualize top 100 products are most expensive but have ratings lower than 3.0

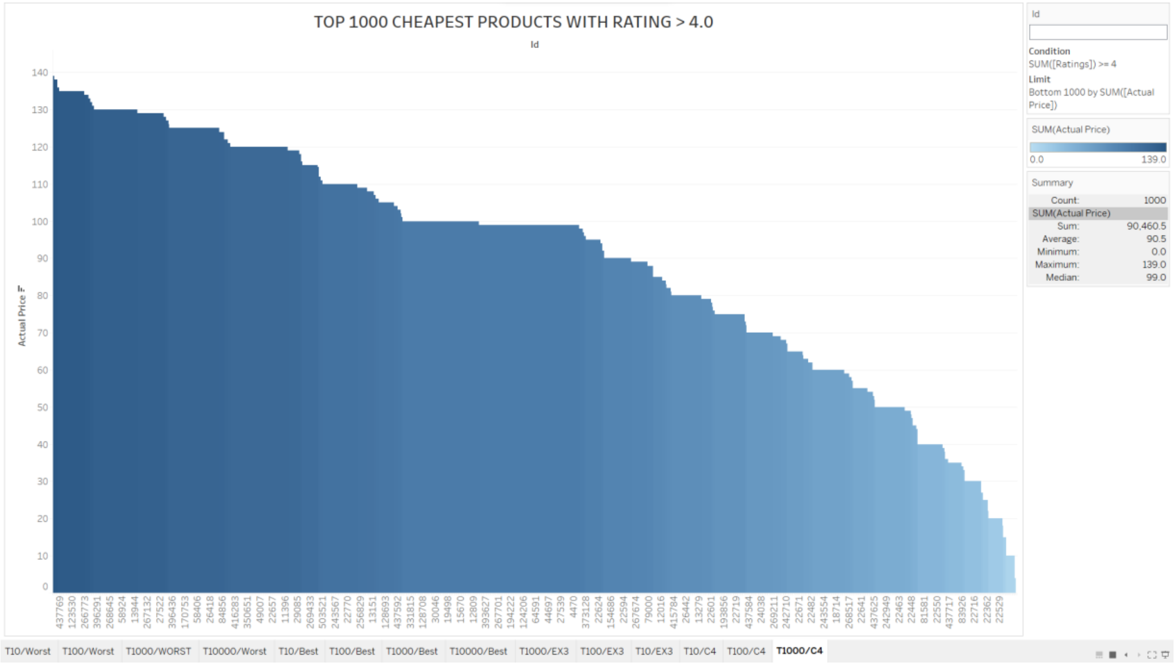


Condition
SUM([Ratings]) <= 3
Limit
Top 10 by SUM([Actual Price])

Figure 2.4.3: Visualize top 10 products are most expensive but have ratings lower than 3.0

Comment on the chart: All the products on the list have a rating lower than 3.0, which implies that the product has received poor reviews from users, possibly because the product quality or service did not meet expectations. There is a correlation between actual value and reviews, i.e. products with high prices often have lower reviews. This may indicate that users have higher expectations for products with high actual value, and if they do not meet those expectations, their ratings may be negatively affected.

2.5 Find products which are cheapest and have ratings more than 4.0, in scale 10, 100, 1K items



TOP 1000 CHEAPEST PRODUCTS WITH RATING > 4.0

Id

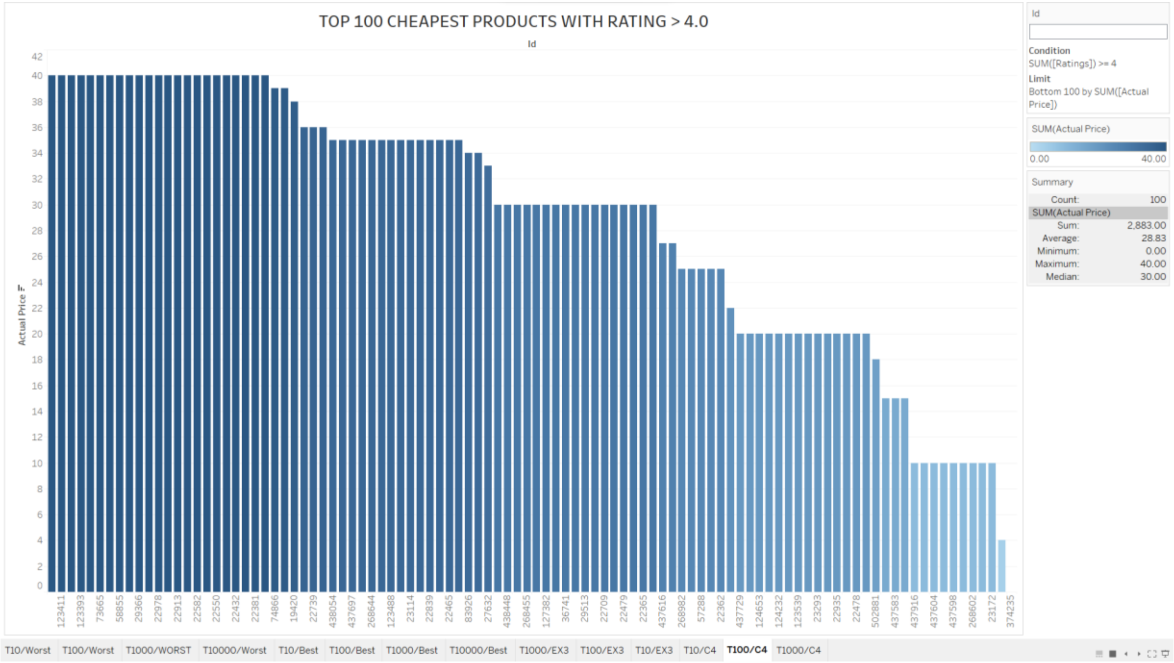
Condition
SUM([Ratings]) >= 4
Limit
Bottom 1000 by SUM([Actual Price])

SUM(Actual Price)
0.0 139.0

Summary

Count:	1000
SUM(Actual Price)	
Sum:	90,460.5
Average:	90.5
Minimum:	0.0
Maximum:	139.0
Median:	99.0

Figure 2.5.1: Visualize top 1000 products are cheapest and have ratings more than 4.0



TOP 100 CHEAPEST PRODUCTS WITH RATING > 4.0

Id

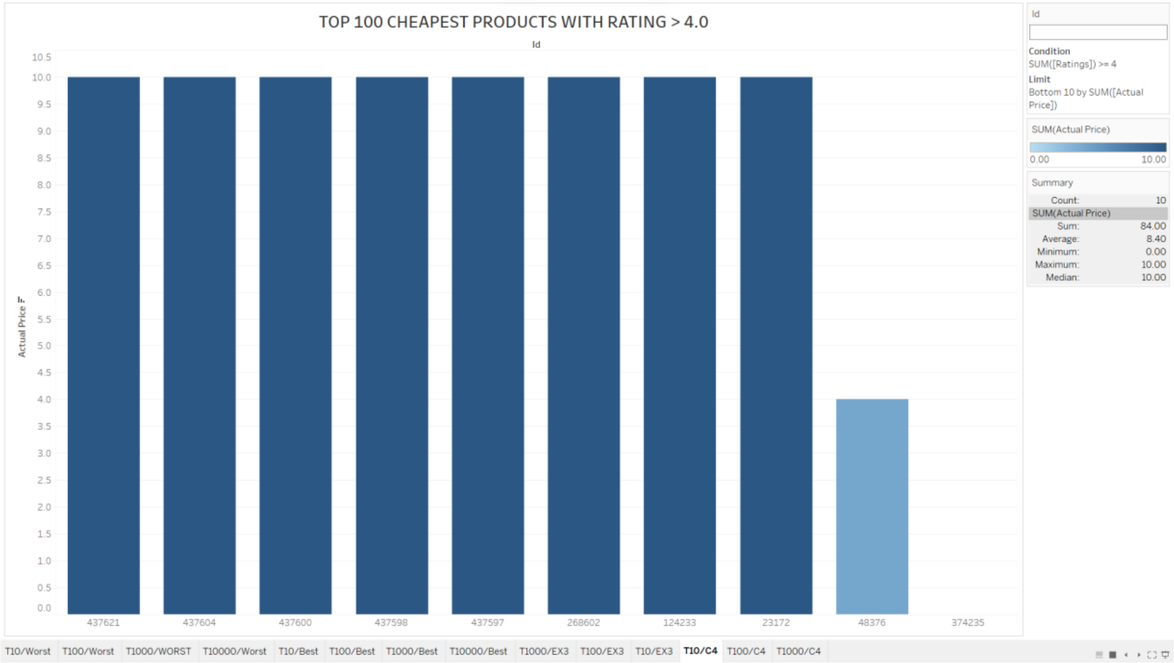
Condition
SUM([Ratings]) >= 4
Limit
Bottom 100 by SUM([Actual Price])

SUM(Actual Price)
0.00 40.00

Summary

Count:	100
SUM(Actual Price)	
Sum:	2,883.00
Average:	28.83
Minimum:	0.00
Maximum:	40.00
Median:	30.00

Figure 2.5.2: Visualize top 100 products are cheapest and have ratings more than 4.0



Id

Condition
SUM([Ratings]) >= 4

Limit
Bottom 10 by SUM([Actual Price])

Summary

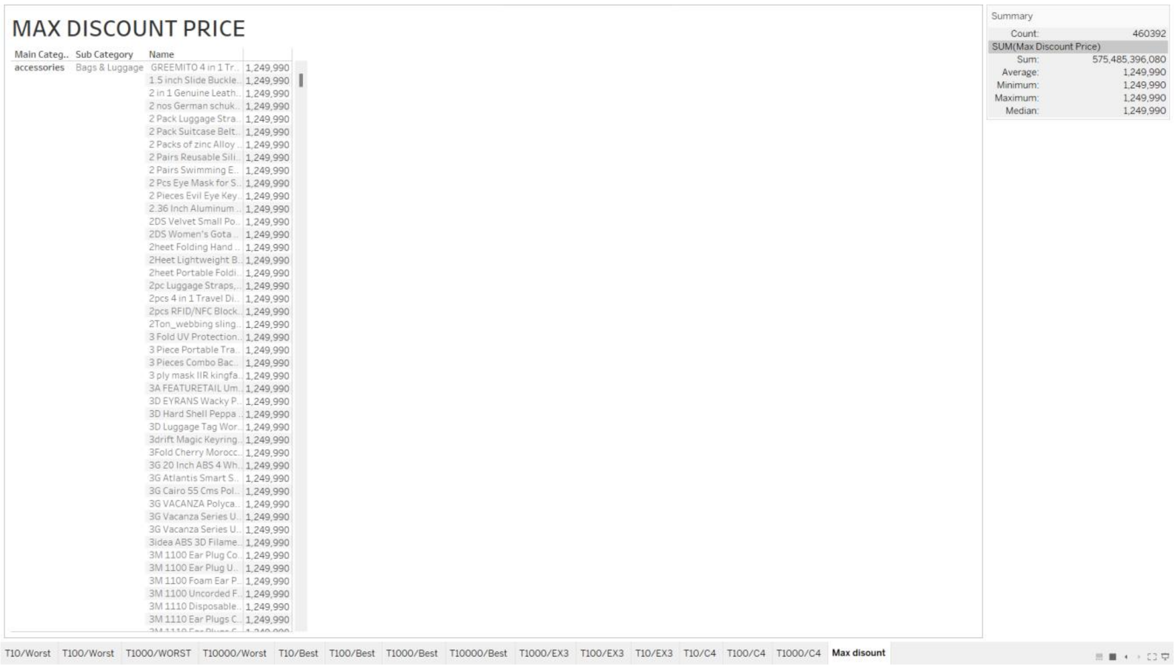
Count:	10
SUM(Actual Price)	
Sum:	84.00
Average:	8.40
Minimum:	0.00
Maximum:	10.00
Median:	10.00

Figure 2.5.3: Visualize top 100 products are cheapest and have ratings more than 4.0

Comment on the chart: there are some products that have really low prices and a maximum rating of 5.0. This may indicate user satisfaction with these products despite the fact that the price is very reasonable. Some products have quite high actual value and a maximum rating of 5.0. This shows that consumers value the quality and performance of these products very highly and may consider them good choices. There are also some products with high actual value and ratings lower than 5.0. This may suggest that based on actual value, consumers may expect a higher rating. The majority of products on this chart have a rating higher than 4.0, indicating reliability.

2.6 Find products which have the largest discounted price

- Use LOD (Level of Detail) calculation: {FIXED : MAX([Discount Price])} to get the maximum value of Discount Price.



MAX DISCOUNT PRICE

Main Categ..	Sub Category	Name	
accessories	Bags & Luggage	GREEMITO 4 in 1 Tr..	1,249,990
		1.5 inch Slide Buckle..	1,249,990
		2 in 1 Genuine Leath..	1,249,990
		2 nos German schuk..	1,249,990
		2 Pack Luggage Stra..	1,249,990
		2 Pack Suitcase Belt..	1,249,990
		2 Packs of zinc Alloy ..	1,249,990
		2 Pairs Reusable Sili..	1,249,990
		2 Pairs Swimming E..	1,249,990
		2 Pcs Eye Mask for S..	1,249,990
		2 Pieces Evil Eye Key..	1,249,990
		2.36 Inch Aluminum ..	1,249,990
		2DS Velvet Small Po..	1,249,990
		2DS Women's Gota ..	1,249,990
		2heet Folding Hand ..	1,249,990
		2Heet Lightweight B..	1,249,990
		2heet Portable Foldi..	1,249,990
		2pc Luggage Straps,..	1,249,990
		2pcs 4 in 1 Travel Di..	1,249,990
		2pcs RFID/NFC Block..	1,249,990
		2Ton_webbing sling..	1,249,990
		3 Fold UV Protection..	1,249,990
		3 Piece Portable Tra..	1,249,990
		3 Pieces Combo Bac..	1,249,990
		3 ply mask IIR kingfa..	1,249,990
		3A FEATURETAIL Um..	1,249,990
		3D EYRANS Wacky P..	1,249,990
		3D Hard Shell Peppa ..	1,249,990
		3D Luggage Tag Wor..	1,249,990
		3drift Magic Keyring..	1,249,990
		3Fold Cherry Morocc..	1,249,990
		3G 20 Inch ABS 4 Wh..	1,249,990
		3G Atlantis Smart S..	1,249,990
		3G Cairo 55 Cms Pol..	1,249,990
		3G VACANZA Polyca..	1,249,990
		3G Vacanza Series U..	1,249,990
		3G Vacanza Series U..	1,249,990
		3idea ABS 3D Filame..	1,249,990
		3M 1100 Ear Plug Co..	1,249,990
		3M 1100 Ear Plug U..	1,249,990
		3M 1100 Foam Ear P..	1,249,990
		3M 1100 Uncorded F..	1,249,990
		3M 1110 Disposable..	1,249,990
		3M 1110 Ear Plugs C..	1,249,990

Summary

Count:	460392
SUM(Max Discount Price)	
Sum:	575,485,396,080
Average:	1,249,990
Minimum:	1,249,990
Maximum:	1,249,990
Median:	1,249,990

Figure 2.6.1: Visualize products which have the largest discounted price

Comment: A total of 460,392 products have the highest discount of 1,249,990.

CHƯƠNG 3. AUGMENTED THE DATASET

3.1 Augment the dataset for the x-axis

- The programming language used : python
- The library used: csv, random
- Code:

```
import csv
import random

# Read the list of cities from an existing file
with open('us_cities.txt', 'r') as city_file:
    cities = [line.strip() for line in city_file]

# Iterate 10 times to create 10 different CSV files
for i in range(10):
    # Create a new file name based on the iteration number
    new_csv_filename = f'new_data_{i+1}.csv'

    # Open the original CSV file and create a new file for writing data with additional columns
    with open('dataset_newv2.csv', 'r', encoding='utf-8') as original_csv, open(new_csv_filename, 'w', newline='', encoding='utf-8') as new_csv:
        reader = csv.reader(original_csv)
        writer = csv.writer(new_csv)

        # Read the header from the original CSV file and add two new headers
        header = next(reader)
        header.extend(["profit", "city"])
        writer.writerow(header)

        # Read each data row from the original CSV file, add random profit and
        city values
            for row in reader:
                profit = random.randint(1, 5000)
                city = random.choice(cities)

                # Handle invalid characters in the data
                row = [cell.replace('\ufffd', ") for cell in row]

                row.extend([profit, city])
                writer.writerow(row)

   #print(f"Created a new file '{new_csv_filename}' with 2 columns: profit and
city.")
```

The process of augmenting data for the x-axis: Read the list of cities from the "**us_cities.txt**" file and save this list in the "**cities**" list. Each line in the "**us_cities.txt**" file contains the name of a city in the United States. Then, use a loop to iterate 10 times to create 10 different CSV files. In each iteration, create a new file name based on the iteration number, for example, "**new_data_1.csv**", "**new_data_2.csv**", …. Open the original CSV file named **"dataset_newv2.csv**" to read data and create a new file for writing data with additional columns. Read the data from the original CSV file and add two new columns, "**profit**" and "**city**" to the header row. Then, write this new header row to the new file. Read each data row from the original CSV file and add random values to the "profit" and "**city**" columns for each row. The "**profit**" value is randomly chosen from the range of 1 to 5000, and the "city" is randomly chosen from the "**cities**" list. Then, handle any invalid characters in the data and add the new values to the data row. As a result, you will have 10 new CSV files each containing 2 columns, **'profit'** and **'city**'.

After obtaining 10 different CSV files, check and select the most appropriate file for further data augmentation along the y-axis. The code:

```
import csv

# Function to compare the values of the "no_of_ratings" and "profit" columns
def compare_columns(filename):
    count = 0

    with open(filename, 'r', encoding='utf-8') as csv_file:
        reader = csv.reader(csv_file)
        header = next(reader)
        no_of_ratings_index = header.index("no_of_ratings")
        profit_index = header.index("profit")

        # Read the first row
        prev_row = next(reader)
        prev_no_of_ratings = int(float(prev_row[no_of_ratings_index]))
        prev_profit = int(prev_row[profit_index])

        # Iterate through the remaining rows in the CSV file
        for row in reader:
            current_no_of_ratings = int(float(row[no_of_ratings_index]))
            current_profit = int(row[profit_index])


            # Compare the values of the "no_of_ratings" and "profit" columns for two
rows
            if abs(current_no_of_ratings - prev_no_of_ratings) <= 100 or
current_no_of_ratings == prev_no_of_ratings:
                if abs(current_profit - prev_profit) <= 100 or current_profit ==
prev_profit:
                    count += 1
            # Update the previous row
            prev_row = row
            prev_no_of_ratings = current_no_of_ratings
            prev_profit = current_profit

    return count

# Create a list of CSV files you want to compare
filenames = ['new_data_1.csv', 'new_data_2.csv', 'new_data_3.csv',
'new_data_4.csv', 'new_data_5.csv','new_data_6.csv', 'new_data_7.csv',
'new_data_8.csv','new_data_9.csv','new_data_10.csv']

# Initialize variables to store the filename with the highest count value and the
corresponding count value
max_count_filename = None
max_count = float('-inf')

# Iterate through each file and calculate the count value
for filename in filenames:
    count = compare_columns(filename)

    # Compare with the current highest count value
    if count > max_count:
        max_count = count
        max_count_filename = filename

# Print the filename with the highest count
print(f"The file with the most reasonable profit is '{max_count_filename}'")
```

The checking process is as follows: Define a function **'compare_columns (filename)**' , which is used to compare two values in the **'profit'** and '**no_of_rating'** columns and then returns the number of rows that have values in the **'profit'** and **'no_of_rating'** columns that are approximately equal in the CSV file being checked. Next, create a list of CSV files to be checked and store it in the **'filenames'** variable. Initialize variables **'max_count_filename'** and **'max_count'** to store the filename with the highest count and the corresponding count. Then, iterate through each file in a loop, calling the **'compare_columns(filename)'** function to count the number of rows with similar values and store this count in the **'count'** variable. Proceed to compare 'count' with **'max_count'**. If 'count' is greater than **'max_count’**, update **'max_count'** and **'max_count_filename'** to store the filename with the most suitable 'profit' values. Finally, select the file with the most appropriate **'profit'** values for further data augmentation along the y-axis.

3.2 Augment the dataset for the y-axis

After selecting the CSV file with the most suitable 'profit' values, proceed to augment along the y-axis by multiplying the number of rows in the file by 100 times. Update the 'id' and 'id_by_category' columns, and randomly generate new values for the 'profit' and 'city' columns.

Library used: dask.dataframe, pandas, random

Code:

```
import dask.dataframe as dd
import pandas as pd
import random

# Read data from the original CSV file using Dask
ddf = dd.read_csv('new_data_8.csv')

# Get the last "id" value from the original data
last_id = ddf['id'].max().compute()
```

```
# Multiply the data table by 100 times
ddf_multiplied = dd.concat([ddf] * 100, ignore_index=True)


# Randomly assign values to the "profit" column within the range from min_profit to max_profit
min_profit = 5  # Minimum value for the "profit" column
max_profit = 5000  # Maximum value for the "profit" column
ddf_multiplied['profit'] = ddf_multiplied['profit'].apply(lambda x: random.uniform(min_profit, max_profit), meta=('profit', 'f8'))


# Update the "id" column to continue from the last value of the original data
ddf_multiplied['id'] = ddf_multiplied['id'] + last_id + 1


# Create a dictionary to track the "id_by_category" value for each "main_category"
id_by_category_mapping = {}


# Update the "id_by_category" column based on the "main_category" to ensure unique values
def update_id_by_category(row):
    main_category = row['main_category']
    if main_category not in id_by_category_mapping:
        id_by_category_mapping[main_category] = 1
    else:
        id_by_category_mapping[main_category] += 1
    return id_by_category_mapping[main_category]


ddf_multiplied['id_by_category'] = ddf_multiplied.apply(update_id_by_category, axis=1, meta=('id_by_category', 'i8'))


# Save the resulting data table to a new CSV file using Dask
ddf_multiplied.to_csv(‘dulieu_100.csv', index=False, single_file=True)
```

```
# Read the original data and save it to a new data table using Dask
ddf_old = dd.read_csv('new_data_8.csv')

# Multiply the data table by 100 times (including both old and new data) using Dask
ddf_combined = dd.concat([ddf_old, ddf_multiplied], ignore_index=True)

# Save the new data table (including both old and new data) to a new CSV file
using Dask
ddf_combined.to_csv(' dulieu_100_v1.csv', index=False, single_file=True)
```

Process Description: Read a CSV file using Dask and store it in **'ddf'**. Use **'dd.concat()**' to create duplicates of the original data and save them in **'ddf_multiplied'**. Randomly generate values for the "**profit**" column in 'ddf_multiplied' within the range of 'min_profit' to 'max_profit' using the **'apply()**' function. Next, update the "**id**" column in the **'ddf_multiplied**' data frame to continue incrementing from the last value in the original data.An **'id_by_category_mapping'** is created to track the "**id_by_category**" values for each unique "**main_category**" in the data frame. The **'update_id_by_category'** function is defined to modify the "**id_by_category**" column based on "**main_category**", ensuring that "**id_by_category**" values are unique for each "**main_category**". This function is then applied to **'id_by_category'**.Once the data processing is complete, the new data is saved to the **'dulieu_100.csv'** file. Finally, **'concat()**' is used to merge the old and new CSV files to create the ultimate augmented CSV file, **'dulieu_100_v1.csv'**.

CHƯƠNG 4. DECISION-MAKING ACTIVITIES

4.1 Find and explain the relationships between dataset variables

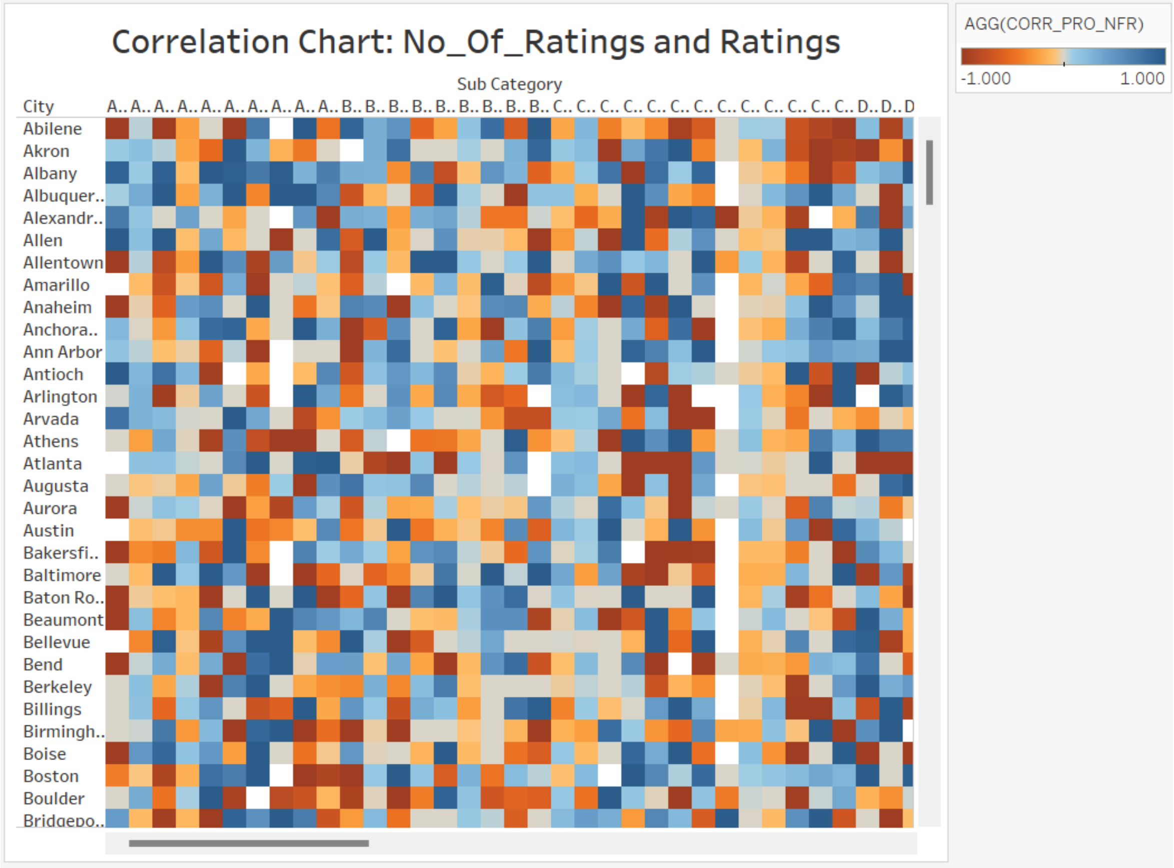


Figure 4.1.1: Correlation Chart: No_Of_Ratings and Ratings

- Asymptotic -1: If the relationship between No_Of_Ratings and Ratings approaches -1, it means that as the number of ratings increases, the overall rating significantly decreases. This may indicate that when there are more ratings, the overall rating tends to decrease.
- Asymptotic 0: If the relationship approaches 0, it suggests that there is no significant correlation between the number of ratings and the overall rating. In this case, the number of ratings does not have a substantial impact on the overall rating.
- Asymptotic 1: If the relationship approaches 1, it means that as the number of ratings increases, the overall rating also increases. This

indicates a positive correlation between the number of ratings and the overall rating.

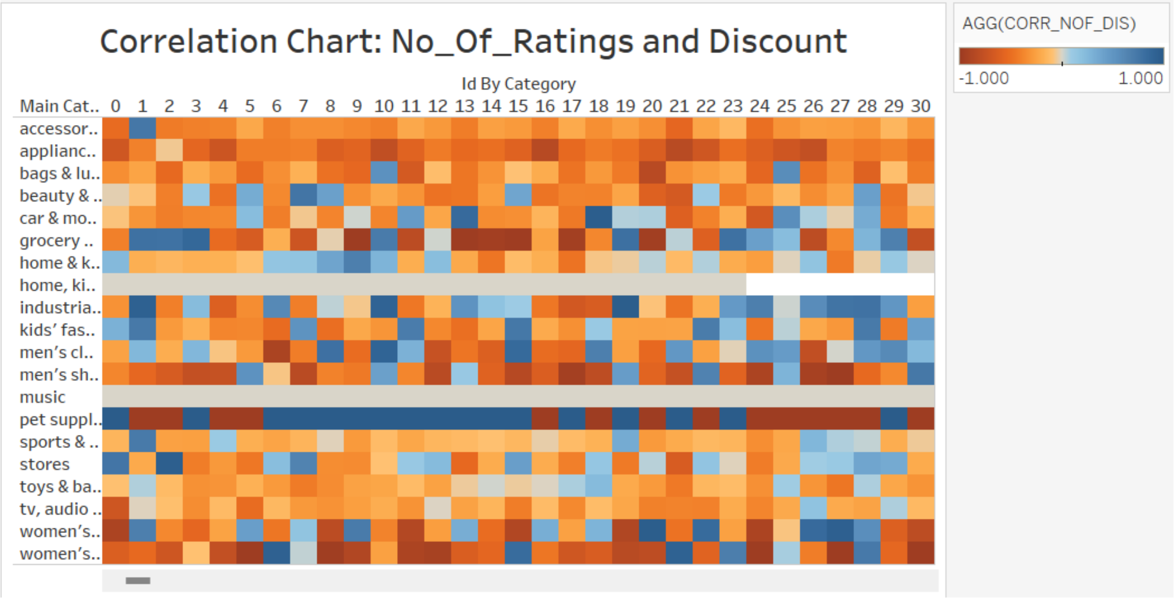


Figure 4.1.2: Correlation Chart: No_Of_Ratings and Discount

- Asymptotic -1: If the relationship between No_Of_Ratings and Discount approaches -1, it means that a high number of ratings corresponds to a low discount level. This might indicate that the product does not need a high discount to attract users when there are many ratings.

- Asymptotic 0: If the relationship approaches 0, it suggests that there is no significant correlation between the number of ratings and the discount level. The number of ratings does not have a substantial impact on the discount level.
- Asymptotic 1: If the relationship approaches 1, it means that a high number of ratings corresponds to a high discount level. This suggests that a higher discount can attract more attention and purchases from customers.

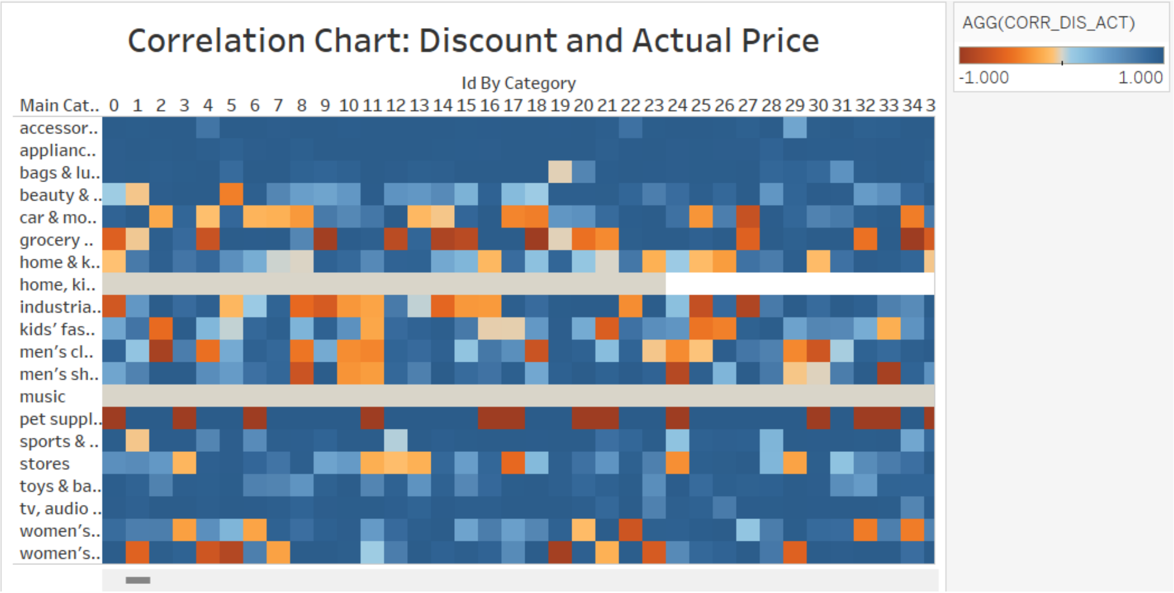


Figure 4.1.3: Correlation Chart: Discount and Actual Price

- Asymptotic -1: If the relationship between Discount and Actual Price approaches -1, it means that a high discount level corresponds to a high actual price. This may indicate that the original price of the product is high, and the discount is only a small portion of the original price.
- Asymptotic 0: If the relationship approaches 0, it suggests that there is no significant correlation between the discount level and the actual price. The discount level does not have a substantial impact on the actual price.
- Asymptotic 1: If the relationship approaches 1, it means that a high discount level corresponds to a low actual price. This implies that a higher discount results in a lower actual price, which can attract customers to make a purchase.

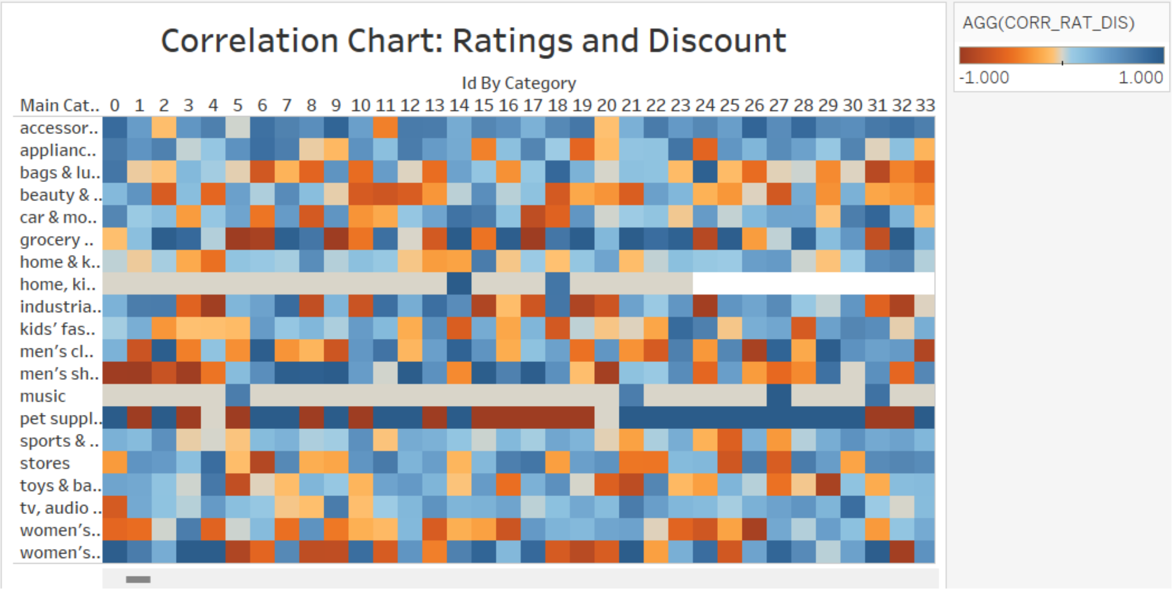


Figure 4.1.4: Correlation Chart: Ratings and Discount

- Asymptotic -1: If the relationship between Ratings and Discount approaches -1, it means that high ratings correspond to low discount levels. This may indicate that highly rated products do not need a large discount to attract buyers.
- Asymptotic 0: If the relationship approaches 0, it suggests that there is no significant correlation between the ratings and the discount level. The ratings do not have a substantial impact on the discount level.
- Asymptotic 1: If the relationship approaches 1, it means that high ratings correspond to high discount levels. This suggests that higher ratings can be associated with higher discount levels.

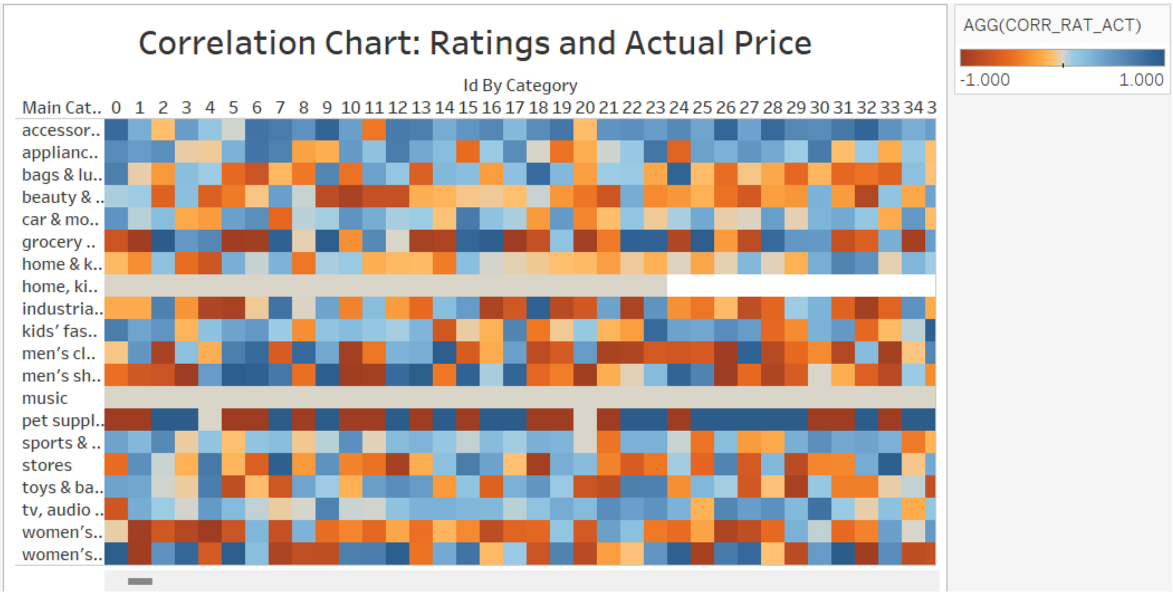


Figure 4.1.5: Correlation Chart: Ratings and Actual Price

- Asymptotic -1: If the relationship between Ratings and Actual Price approaches -1, it means that high ratings correspond to high actual prices. This may indicate that highly rated products have higher prices.
- Asymptotic 0: If the relationship approaches 0, it suggests that there is no significant correlation between the ratings and the actual price. The ratings do not have a substantial impact on the actual price.
- Asymptotic 1: If the relationship approaches 1, it means that high ratings correspond to low actual prices. This indicates that highly rated products tend to have lower prices, which can attract customers to make a purchase.

4.2 Determine the threshold based on profit

- Chart based on products with good profit:

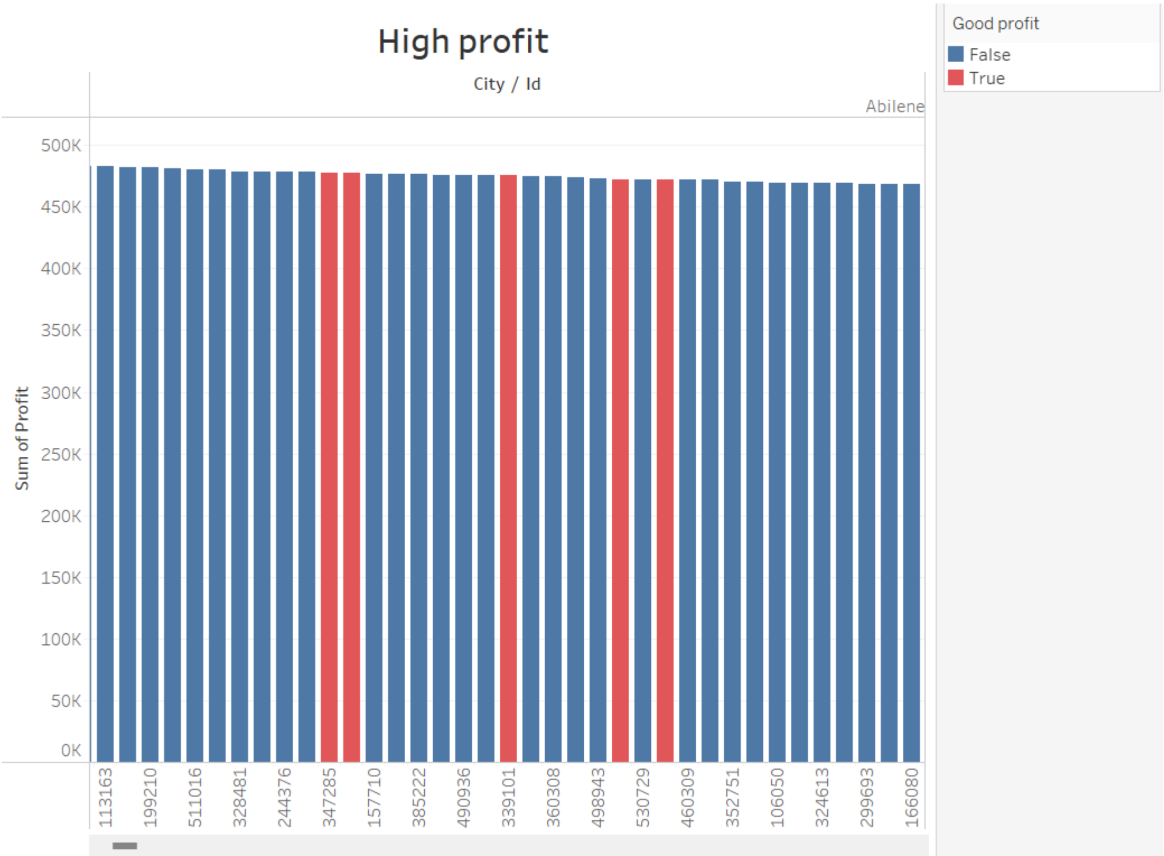


Figure 4.2.1: Base on highly profitable products

- Condition: ([Profit]/[Discount Price])*100 >= 1500

This condition evaluates the profitability of products based on the profit-to-discount price ratio. If the ratio exceeds the threshold of 1500, the product is considered to have good profit potential. This can indicate that the product has the potential to generate higher profits compared to the discount cost.

This chart is used to identify products with good profit when the profit-to-discount price ratio exceeds the threshold of 1500. When a product lies on this chart, it means it meets or exceeds the minimum profit threshold.

Evaluation: This chart allows you to see products with good profit based on the profit-to-discount price ratio. If there are multiple products on this chart, it indicates that your company has many products with good profit potential and the ability to attract customers.

Potential assessment: If there are products close to the chart or below the minimum profit threshold, it suggests they have the potential for profit improvement. The company can consider measures such as increasing the selling price, reducing production costs, or enhancing customer reach to increase the profit of these products.

- Chart based on outdated products:

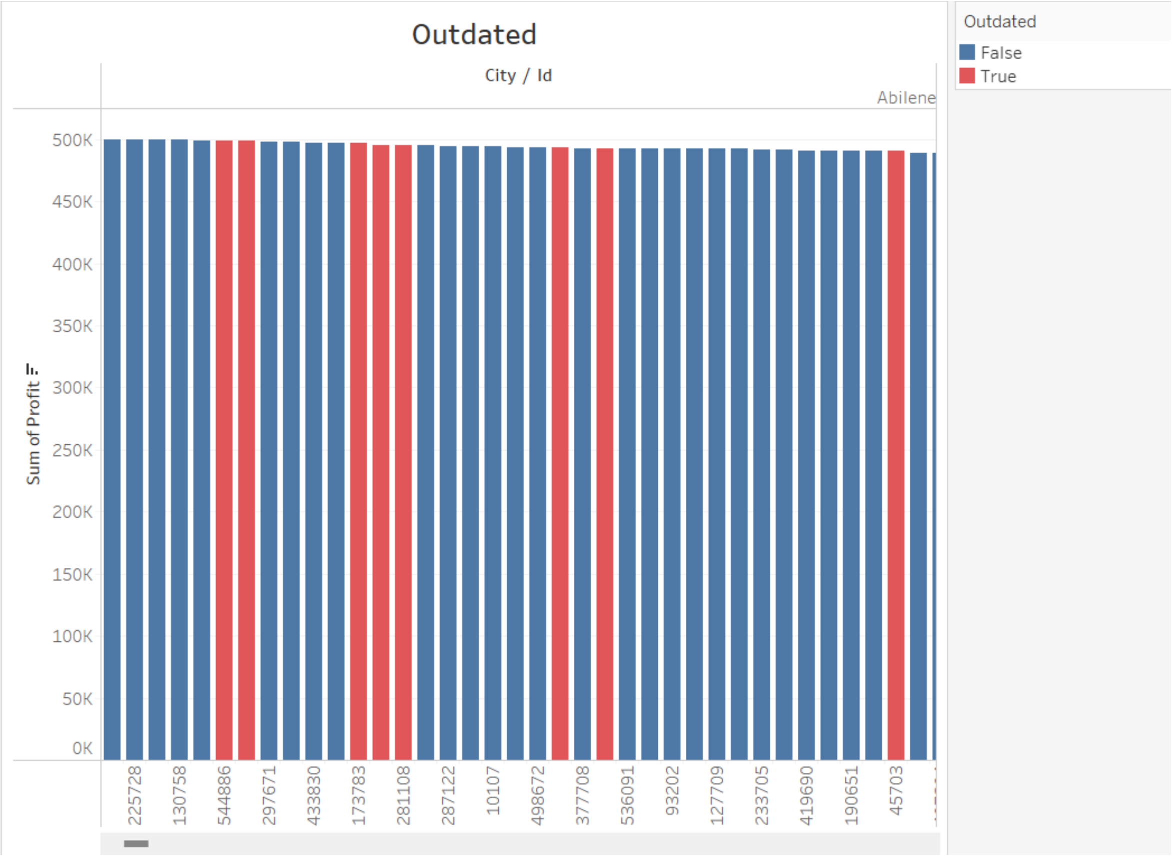


Figure 4.2.2: Base on outdated products

- Condition: [No Of Ratings] <= 3 or [Ratings] >= 100

This condition evaluates the popularity and ratings of products. If the number of ratings is less than or equal to 3 or the ratings are greater than or equal to 100, the product is considered at risk of becoming outdated. This can indicate that the product is not receiving enough attention or has poor ratings from customers.

This chart is used to identify products at risk of becoming outdated based on the number of ratings and overall ratings from customers. When a product lies on this chart, it indicates it has a low number of ratings or high ratings.

Evaluation: This chart allows you to see products at risk of becoming outdated based on the number of ratings and overall ratings. If there are multiple products on this chart, it suggests that your company needs to reconsider strategies to improve ratings and attract customer interest.

Potential assessment: If there are products close to the chart or surpassing the threshold of the number of ratings or ratings, it can be assessed that they have the potential to become standout products and attract customers. The company can focus on enhancing customer reach and promotion to increase the number of ratings and improve overall ratings for these products.

4.3 Visualize the best-and-worst-selling-profit on world map

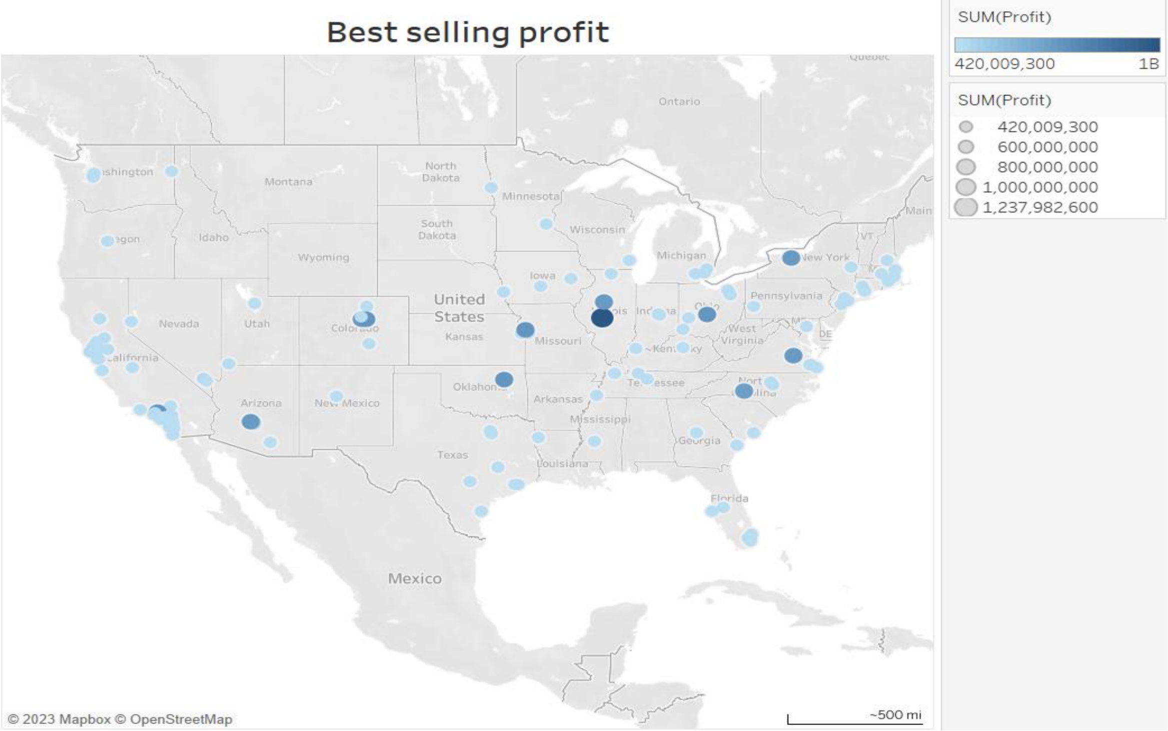


Figure 4.3.1: Best selling profit

- The profit range on this chart runs from $420,009,300 to $1 billion.
- The profits of the products are represented by blue dots on the map.
- The even distribution of the dots across the United States indicates the presence of many products with good profit in different regions.
- Evaluation: This chart demonstrates the diversity and popularity of products with good profit nationwide. It suggests that your company has multiple successful products with the potential to attract customers in various regions.

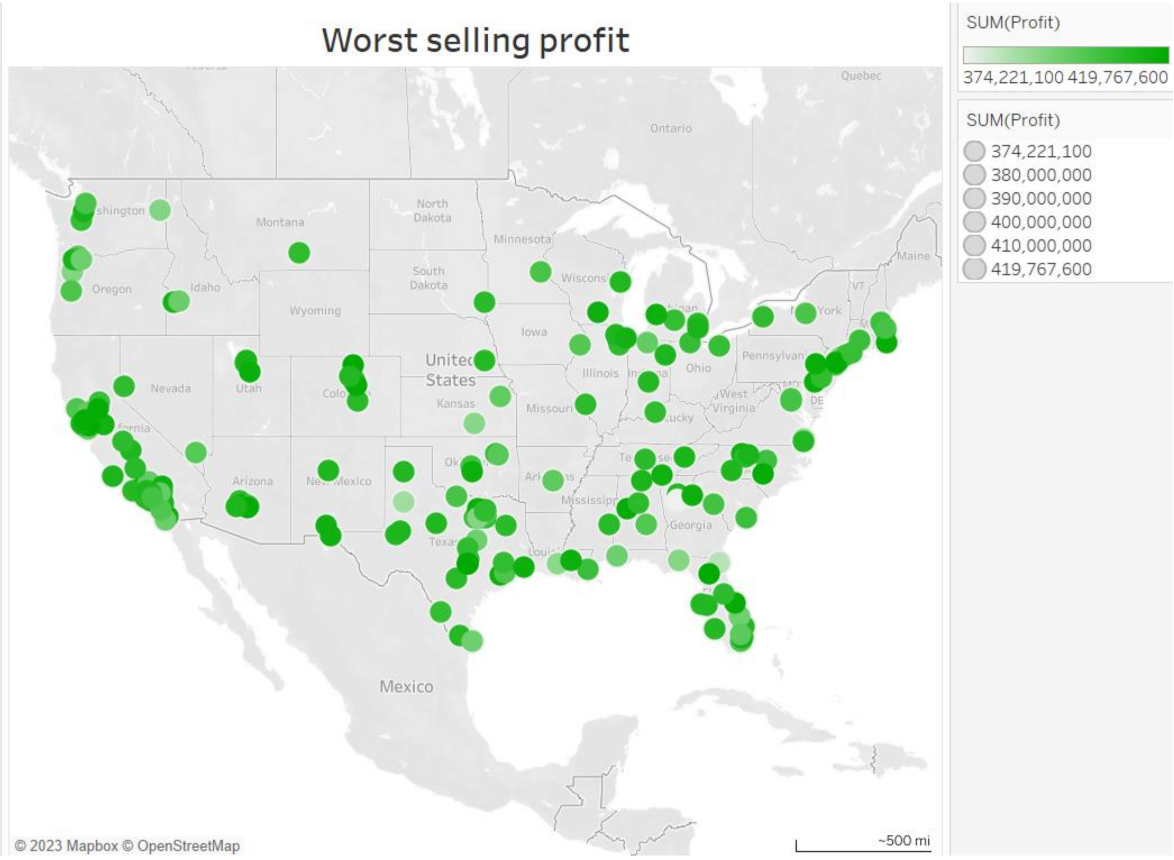


Figure 4.3.2: Worst selling profit

- The profit range on this chart runs from $374,221,100 to $419,767,600.
- The profits of the products are represented by green dots on the map.
- The even distribution of the dots across the United States indicates the presence of many products with low profit in different regions.

Evaluation: This chart shows the existence of products with low profit nationwide. It suggests that your company is facing challenges with some products and needs to consider measures to improve their profitability, such as increasing prices, reducing production costs, or enhancing customer reach.

REFERENCES

English

Wilfried Grossmann, Stefanie Rinderle-Ma. Fundamentals of Business Intelligence. Springer-Verlag Berlin Heidelberg, 2015

Joshua N. Milligan. Learning Tableau 2020, Fourth Edition. Packt Publishing, 2020