ment-hypothesis-testing-da-balaram

April 8, 2024

0.0.1 Develop a scenario to provide an overall understanding of the organization represented by the dataset.

let assume a scenario, we get hired as data scientist in one of worlds top shoe company named 'Jordhan'. we got first task - company wants to increase its sale in coming quarter

As a data scientist we have the sales data , we have to define one problem statement and analyse findings, provide Insights

Scenario: The global shoe company offers a diverse range of shoe types and utilizes various sales strategies, including discounts, to boost product movement. Understanding the effectiveness of discounts on sales volume can help optimize marketing and pricing strategies.

Problem Statement :- Determine if discounts significantly influence the number of units sold for the company's products. if there are significant differences in sales volumes among various shoe categories ?

From problem statement, let's ### Describe the expected structure and contents of the shoes.csv dataset, focusing on columns that would support the analysis of shoe categories:

For the task of shoes.csv dataset contains the following relevant columns:

Discount:

Percentage discount applied to the product. ###### ProductID : product id is usedfor analysing purticular category

0.0.2 Describe your test approach

We will use a t-test to compare the mean sales volumes between discounted and non-discounted transactions.

Justify the Test Selection

A t-test is suitable for comparing the means of two independent groups (discounted vs. non-discounted sales) when the population standard deviations are unknown and the sample size is less than 30 or unknown.

Define H0 and H1

H0 (Null Hypothesis): The mean sales volume is the same for discounted and non-discounted products.

H1 (Alternative Hypothesis): The mean sales volume differs between discounted and non-discounted products.

0.0.3 Conduct your analysis according to your defined parameters above

For analysis let load data file and observe the data

[1]:
```
import pandas as pd

# Load the dataset
shoes_df = pd.read_csv('C:/Users/balar/Downloads/assign_wk4/assign_wk4/shoes.
 ↪csv')

# Display the first few rows of the dataframe
shoes_df.head(20)
```

[1]:

	InvoiceNo	Date	Country	ProductID	Shop	Gender	Size (US)	\
0	52389.0	1/1/2014	United Kingdom	2152.0	UK2	Male	11.0	
1	52390.0	1/1/2014	United States	2230.0	US15	Male	11.5	
2	52391.0	1/1/2014	Canada	2160.0	CAN7	Male	9.5	
3	52392.0	1/1/2014	United States	2234.0	US6	Female	9.5	
4	52393.0	1/1/2014	United Kingdom	2222.0	UK4	Female	9.0	
5	52394.0	1/1/2014	United States	2173.0	US15	Male	10.5	
6	52395.0	1/2/2014	Germany	2200.0	GER2	Female	9.0	
7	52396.0	1/2/2014	Canada	2238.0	CAN5	Male	10.0	
8	52397.0	1/2/2014	United States	2191.0	US13	Male	10.5	
9	52398.0	1/2/2014	United Kingdom	2237.0	UK1	Female	9.0	
10	52399.0	1/2/2014	United States	2197.0	US1	Male	10.0	
11	52399.0	1/2/2014	United States	2213.0	US11	Female	9.5	
12	52399.0	1/2/2014	United States	2206.0	US2	Female	9.5	
13	52400.0	1/2/2014	United States	2152.0	US15	Male	8.0	
14	52401.0	1/3/2014	Germany	2235.0	GER1	Male	10.5	
15	52401.0	1/3/2014	Germany	2197.0	GER1	Female	8.5	
16	52402.0	1/3/2014	Canada	2240.0	CAN6	Male	9.5	
17	52403.0	1/3/2014	United States	2221.0	US7	Male	11.0	
18	52404.0	1/3/2014	United States	2234.0	US6	Female	9.5	
19	52404.0	1/3/2014	United States	2197.0	US1	Male	10.0	

	Size (Europe)	Size (UK)	UnitPrice	Discount	Year	Month	SalePrice
0	44	10.5	$159.00	0%	2014	1	$159.00
1	44-45	11.0	$199.00	20%	2014	1	$159.20
2	42-43	9.0	$149.00	20%	2014	1	$119.20
3	40	7.5	$159.00	0%	2014	1	$159.00
4	39-40	7.0	$159.00	0%	2014	1	$159.00
5	43-44	10.0	$159.00	0%	2014	1	$159.00
6	39-40	7.0	$179.00	0%	2014	1	$179.00
7	43	9.5	$169.00	0%	2014	1	$169.00
8	43-44	10.0	$139.00	0%	2014	1	$139.00

9	39-40	7.0	$149.00	0%	2014	1	$149.00
10	43	9.5	$129.00	0%	2014	1	$129.00
11	40	7.5	$169.00	10%	2014	1	$152.10
12	40	7.5	$139.00	0%	2014	1	$139.00
13	41	7.5	$139.00	0%	2014	1	$139.00
14	43-44	10.0	$169.00	50%	2014	1	$84.50
15	39	6.5	$179.00	20%	2014	1	$143.20
16	42-43	9.0	$199.00	30%	2014	1	$139.30
17	44	10.5	$149.00	50%	2014	1	$74.50
18	40	7.5	$159.00	0%	2014	1	$159.00
19	43	9.5	$129.00	0%	2014	1	$129.00

lets classify the data

A classification of data types into categorical (qualitative) and numerical (quantitative)

here is how the columns of the dataset can be classified:

Categorical (Qualitative) Data:

Nominal: These are categories without any intrinsic ordering.

InvoiceNo: it's a label for identification.
Country: Names of countries
ProductID: Numeric but used as a label to identify products.
Shop: its kind of code for respective country where it belongs.
Gender: product type with gender preference (male or female).
Year: Though numeric, it is used here as a category for when the sale occurred.
Month: Also categorical, representing the time of sale.

Ordinal: These are categories with a meaningful order but not measured quantitatively.

Size (US), Size (Europe), Size (UK): Shoe sizes have an order but the differences between sizes are not consistent measures of quantity.
Discount: Typically represents levels of discount which have a meaningful order (e.g., 10% is more than 5%).

Numerical (Quantitative) Data:

Discrete: Countable data points.

There do not appear to be any strictly discrete numerical variables in this dataset unless there are counts of items listed.

Continuous: These can take on any value within a range.

UnitPrice: The price can vary continuously within a range.
SalePrice: The final sale price, also a continuous variable since it can have any value within a range.

Keep in mind that classifications can depend on the context in which the data is used.

0.0.4 Data Preparation

We'll focus on SalePrice and ProductID for our analysis, assuming ProductID can help us distinguish between different categories. We'll need to:

Convert SalePrice to a numeric value, removing any currency symbols and converting the values from strings to numbers.

first check do we have any missing values?

[2]:
```
# Check for missing values in the key columns
missing_values = shoes_df[['Discount', 'SalePrice', 'ProductID']].isnull().sum()

# Display the count of missing values for each column
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 Discount     1
SalePrice    1
ProductID    1
dtype: int64
```

we can see only one missing , so dropping just 3 rows wont impact much, lets do dropping

[3]:
```
shoes_df = shoes_df.dropna(subset=['Discount', 'SalePrice', 'ProductID'])
```

[4]:
```
print(shoes_df[['Discount', 'SalePrice', 'ProductID']].isnull().sum())
```

```
Discount     0
SalePrice    0
ProductID    0
dtype: int64
```

There is no missing values now

In data frame we can see sale price, unit price and discounbt is in string , to further analyse data we have to convert them to float

[6]:
```
# Check the data types of 'Discount' and 'SalePrice' columns
print(shoes_df['Discount'].dtype)
print(shoes_df['SalePrice'].dtype)

# If they are not of type 'object' (string), we should not apply the string␣
 ↪methods
# Otherwise, we'll proceed to clean the columns
if shoes_df['Discount'].dtype == 'object':
    shoes_df['Discount'] = shoes_df['Discount'].str.rstrip('%').astype('float')␣
 ↪/ 100
else:
    shoes_df['Discount'] = shoes_df['Discount'].astype('float') / 100
```

```
if shoes_df['SalePrice'].dtype == 'object':
    shoes_df['SalePrice'] = shoes_df['SalePrice'].str.replace('[\$,]', '',␣
 ↪regex=True).astype('float')
else:
    shoes_df['SalePrice'] = shoes_df['SalePrice'].astype('float')

# Continue with the analysis...
```

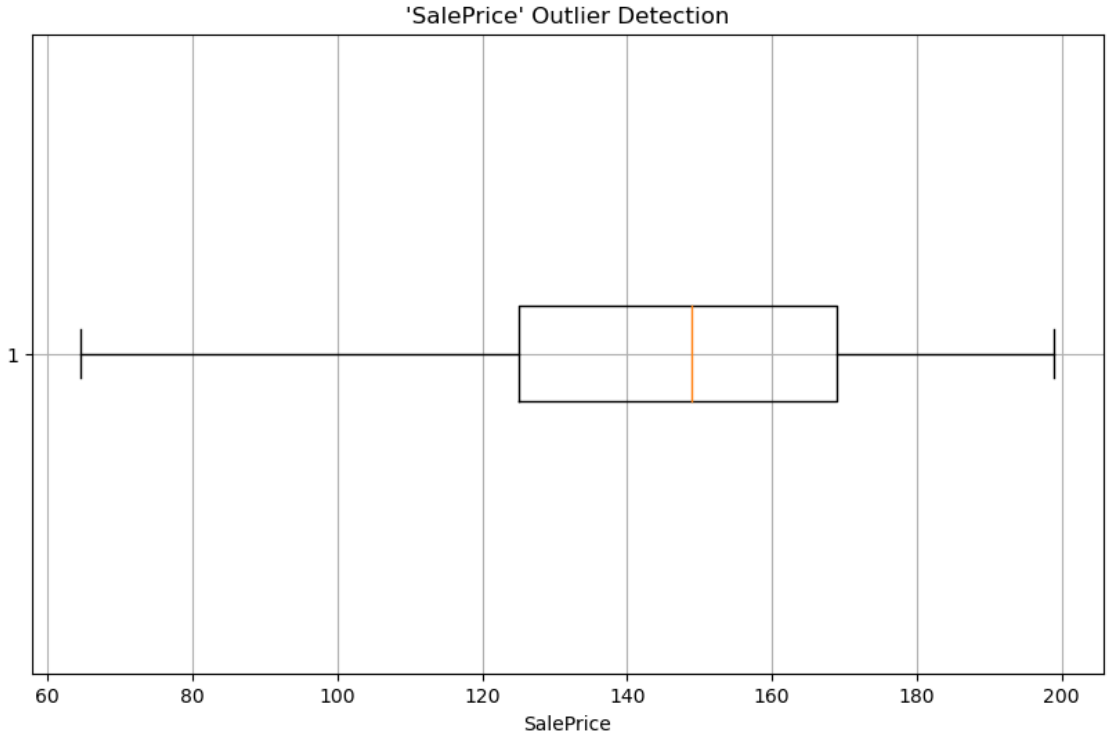
```
float64
float64
```

Now sale price and discount is converted to float, we can further check any outliers present or not and we will go with boxplot for visualisation

Detect outliers and boxplot it For detecting outliers lets caluclate quantiles,IQR and define the boundaries. using boundaries check outliers and make visualization plot with boxplot

[6]:
```
import pandas as pd
import matplotlib.pyplot as plt



# Calculate the quantiles and IQR
Q1 = shoes_df['SalePrice'].quantile(0.25)
Q3 = shoes_df['SalePrice'].quantile(0.75)
IQR = Q3 - Q1

# Define the boundaries for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Detect outliers
outliers = shoes_df[(shoes_df['SalePrice'] < lower_bound) |␣
 ↪(shoes_df['SalePrice'] > upper_bound)]

# Check if outliers are present and print the count
if not outliers.empty:
    print(f"Number of outliers detected in 'SalePrice': {len(outliers)}")
else:
    print("No outliers detected in 'SalePrice'.")

# Plotting to visualize the outliers
plt.figure(figsize=(10, 6))
plt.boxplot(shoes_df['SalePrice'], vert=False)
plt.title("'SalePrice' Outlier Detection")
plt.xlabel('SalePrice')
plt.grid(True)
plt.show()
```

```
No outliers detected in 'SalePrice'.
```



[7]:
```
print(shoes_df['SalePrice'].describe())
```

```
count    14976.000000
mean       143.980162
std         35.174098
min         64.500000
25%        125.100000
50%        149.000000
75%        169.000000
max        199.000000
Name: SalePrice, dtype: float64
```

What this output mean The number of sale price entries in the dataset is 14,976. (transactions recorded)

Mean: The average sale price across all transactions is approximately $143.98.(central tendency)

Std (Standard Deviation): The standard deviation is about $35.17, which measures the amount of variation A higher standard deviation indicates that the sale prices vary widely from the mean.

Min (Minimum): The lowest sale price recorded in the dataset is $64.50,

25% (First Quartile): 25% of the sale prices are below $125.10. This is the “lower quartile”

50% (Median): The median sale price is $149.00. This is the “middle” value in the dataset

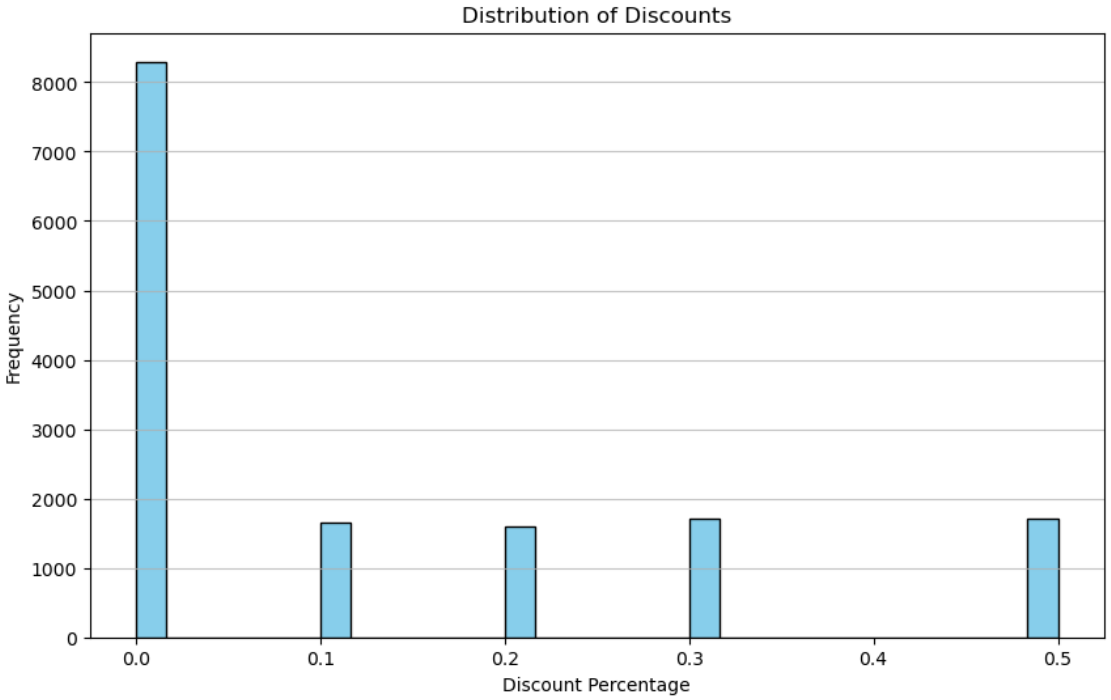
75% (Third Quartile): 75% of the sale prices are below $169.00. This is the “upper quartile”

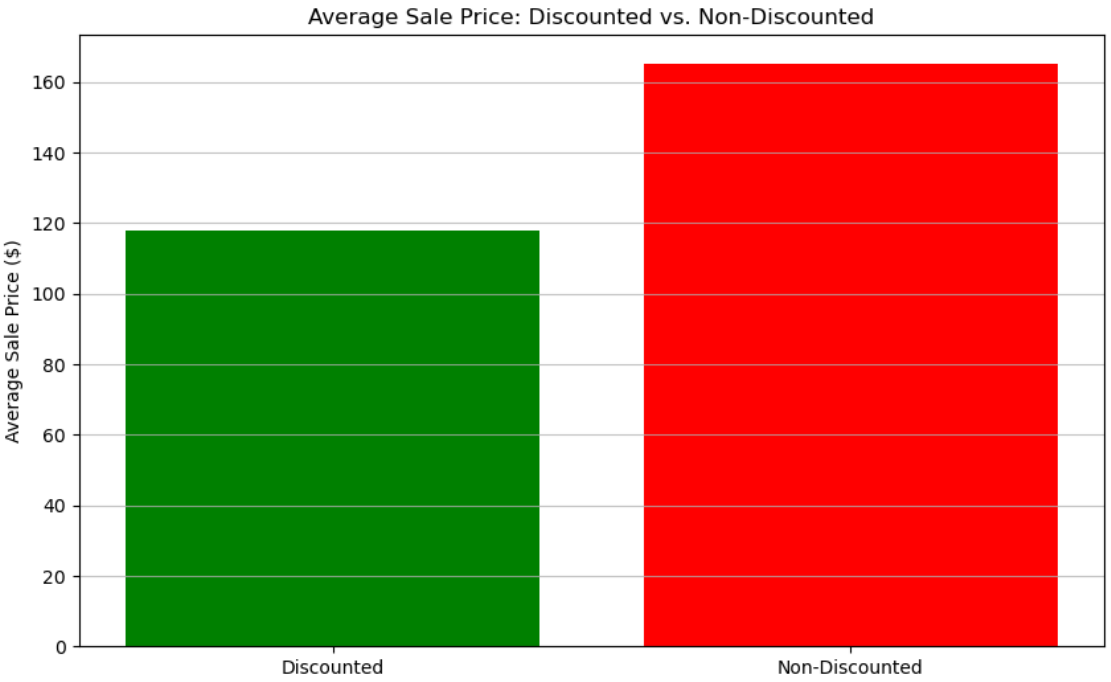
Max (Maximum): The highest sale price in the dataset is $199.00

Lets see how discounted and non discounted sales distributed in our shoes dataset

[8]:
```
import matplotlib.pyplot as plt

# Distribution of Discounts
plt.figure(figsize=(10, 6))
plt.hist(shoes_df['Discount'], bins=30, color='skyblue', edgecolor='black')
plt.title('Distribution of Discounts')
plt.xlabel('Discount Percentage')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)

# Average SalePrice by Discount Presence
discounted_sales = shoes_df[shoes_df['Discount'] > 0]['SalePrice'].mean()
non_discounted_sales = shoes_df[shoes_df['Discount'] == 0]['SalePrice'].mean()

plt.figure(figsize=(10, 6))
plt.bar(['Discounted', 'Non-Discounted'], [discounted_sales,␣
 ↪non_discounted_sales], color=['green', 'red'])
plt.title('Average Sale Price: Discounted vs. Non-Discounted')
plt.ylabel('Average Sale Price ($)')
plt.grid(axis='y', alpha=0.75)

(discounted_sales, non_discounted_sales)
```

[8]:
```
(117.84624251497004, 165.02338476374157)
```





From the plots we can see how discounted and non discounted sales are distributed based on plot 2 , discount vs non-discount 120 : 160 i.e **3:4**

Categorise productID by mean

[9]:
```
# Assuming 'ProductID' can categorize products, we calculate the average unit␣
 ↪price by category
average_price_by_category = shoes_df.groupby('ProductID')['UnitPrice'].mean().
 ↪reset_index()

# For simplicity, we'll display the first few categories
average_price_by_category.head()
```

[9]:

	ProductID	UnitPrice
0	2147.0	162.272727
1	2148.0	158.387755
2	2149.0	160.111111
3	2150.0	168.400000
4	2151.0	161.884615

We can see mean on each productID

performing t-test

[9]:
```
from scipy import stats
# Perform a t-test between discounted and non-discounted sales
t_stat, p_value = stats.ttest_ind(
    shoes_df[shoes_df['Discount'] > 0]['SalePrice'],
    shoes_df[shoes_df['Discount'] == 0]['SalePrice'],
    equal_var=False  # Assume samples have unequal variances
)

# Print the t-statistic and p-value
print(f"T-statistic: {t_stat}, P-value: {p_value}")
```

```
T-statistic: -106.31884280444672, P-value: 0.0
```

The results of the hypothesis test using the t-test

are statistically significant and indicate that discounts have a strong effect on the sale price of shoes in the dataset:

The t-statistic of -106.32 is a measure of the extent to which the discounted group's mean sale price is different from the non-discounted group's mean sale price.

The p-value of 0.0 effectively means that the probability of observing such a large effect due to chance is extremely low or impossible. In statistical terms, this p-value is far below than 0.05, which leads to the rejection of the null hypothesis.

Final Interpretation:

Statistical Significance: The test provides strong evidence that the average sale price is significantly different between discounted and non-discounted transactions.

Practical Implication: For the shoe company, this implies that discounting items leads to a lower

revenue per item sold.it also suggests that if the goal is to maximize revenue per item, heavy discounting might not be the optimal approach.

Business Strategy Consideration: The business needs to consider whether the lower prices from discounts are effectively balanced by an increase in the quantity sold. Discounts might attract more customers or prompt larger purchases, but it's important to ensure that the increase in volume compensates for the lower revenue per unit.

.

Additionally lets perform t-test for each product category

[12]:
```
import pandas as pd
from scipy import stats

# Assuming the 'shoes_df' DataFrame is already loaded and cleaned

# Group data by 'ProductID' and 'Discount' status and calculate the average␣
 ↪'SalePrice'
category_discount_analysis = shoes_df.groupby(['ProductID',␣
 ↪shoes_df['Discount'] > 0])['SalePrice'].mean().unstack()

# Initialize a dictionary to store t-test results for each category
category_t_tests = {}

# Loop through each product category and perform a t-test
for product_id in category_discount_analysis.index:
    non_discounted_prices = shoes_df[(shoes_df['ProductID'] == product_id) &␣
 ↪(shoes_df['Discount'] == 0)]['SalePrice']
    discounted_prices = shoes_df[(shoes_df['ProductID'] == product_id) &␣
 ↪(shoes_df['Discount'] > 0)]['SalePrice']

    # Perform t-test only if both non-discounted and discounted groups have data
    if len(non_discounted_prices) > 0 and len(discounted_prices) > 0:
        t_stat, p_val = stats.ttest_ind(non_discounted_prices,␣
 ↪discounted_prices, equal_var=False)
        category_t_tests[product_id] = (t_stat, p_val)

# The 'category_t_tests' dictionary now contains the t-statistic and p-value␣
 ↪for each product category
category_t_tests
```

[12]:
```
{2147.0: (9.059221779976086, 2.8121380193952233e-15),
 2148.0: (11.063239345311972, 8.151711009777778e-19),
 2149.0: (9.422107889269657, 3.7275659758489044e-16),
 2150.0: (16.766041910844937, 1.654827910003233e-31),
 2151.0: (10.409867945243496, 1.1605771129298608e-18),
 2152.0: (11.490276463706923, 3.589745355716969e-21),
```

2153.0: (9.94720153004761, 1.6248366034409748e-17),
2154.0: (8.688348927148997, 3.909321977201121e-14),
2155.0: (14.03043697645438, 6.839887539743955e-24),
2156.0: (10.86210736510963, 3.41092569464514e-20),
2157.0: (12.658499816407684, 3.356324301987822e-25),
2158.0: (11.442457894766193, 1.1264590315431398e-21),
2159.0: (11.493567433396942, 2.074289772574167e-20),
2160.0: (9.629445125589633, 5.387948426615051e-17),
2161.0: (8.388517494322958, 8.889759423309827e-13),
2162.0: (9.727841725083596, 3.829081661212291e-16),
2163.0: (11.87306643642536, 1.1276011808455176e-20),
2164.0: (8.903974187674399, 8.320406338058632e-14),
2165.0: (10.511604715888188, 2.50566162272199e-19),
2166.0: (9.194593154768764, 3.928268190450401e-14),
2167.0: (12.330381570449822, 1.9125097253177425e-23),
2168.0: (10.01084532751156, 2.0861452095967722e-18),
2169.0: (6.702506065789567, 3.3335341486740244e-10),
2170.0: (13.5995729571323, 1.3342721640574757e-25),
2171.0: (8.578524107455534, 1.9899664449106913e-14),
2172.0: (10.829456613624792, 2.6487836163164614e-19),
2173.0: (13.928217143639523, 1.1891666456140583e-29),
2174.0: (9.077197249363318, 1.7974069245169926e-15),
2175.0: (10.603658594874835, 5.033226967930812e-19),
2176.0: (7.310858526115734, 8.225531455483068e-11),
2177.0: (10.351564011418976, 7.529940217337662e-18),
2178.0: (9.080250564942938, 1.5748108595322969e-15),
2179.0: (12.036543899874221, 1.193405575686451e-22),
2180.0: (8.945022587507614, 3.644265413233345e-14),
2181.0: (12.647020682982836, 5.067340619055889e-25),
2182.0: (10.476664753661819, 1.2594178575383e-18),
2183.0: (11.478629386970645, 1.4795895649940798e-22),
2184.0: (10.479024425801924, 2.5110760727095186e-18),
2185.0: (9.236141134336645, 1.6872842498382625e-16),
2186.0: (10.425896714978773, 1.3238520406456349e-18),
2187.0: (15.738694854763255, 1.1538632008725407e-27),
2188.0: (11.324594747314055, 1.7395162439236736e-17),
2189.0: (9.637256200861255, 2.991319562774014e-15),
2190.0: (14.61018627994973, 8.199649484137534e-31),
2191.0: (11.788004324288837, 3.4394576207654615e-23),
2192.0: (13.520155559504026, 3.281999569486068e-28),
2193.0: (8.916644865437767, 1.9682313912835163e-15),
2194.0: (9.938023888707624, 4.042386542485592e-17),
2195.0: (10.181776222579613, 1.1066341536817305e-18),
2196.0: (15.158021955597182, 2.1583592482121213e-28),
2197.0: (10.732649934032667, 1.0070537056695099e-19),
2198.0: (9.222570614207907, 2.0050384694137174e-14),
2199.0: (10.143239869315693, 2.1418761403918876e-17),

```
2200.0: (13.551958001409202, 1.3072501639323632e-25),
2201.0: (10.89243651584299, 2.4691676531431164e-18),
2202.0: (10.639356383132057, 1.476964680642533e-20),
2203.0: (10.990721113144698, 2.3636094536667704e-20),
2204.0: (11.192276476153435, 5.728212964162184e-20),
2205.0: (8.261643539513427, 1.4398098864454642e-13),
2206.0: (13.028594825544818, 1.0461780783497776e-25),
2207.0: (10.632432195749903, 1.8831243984921155e-20),
2208.0: (11.665963064227954, 7.336004813726176e-19),
2209.0: (13.181630668182802, 4.740036600444287e-28),
2210.0: (10.18009996083046, 1.5992460190250297e-16),
2211.0: (8.324083522052591, 4.270321230928727e-13),
2212.0: (14.542176951892156, 9.338036115995004e-30),
2213.0: (14.104635532069373, 6.82260697649422e-32),
2214.0: (9.593925893216204, 7.686934026305494e-17),
2215.0: (12.160325890319198, 3.2338098019850136e-24),
2216.0: (8.886592700690423, 1.0422060251012395e-14),
2217.0: (14.552209206308023, 1.9659652323366548e-31),
2218.0: (10.412747395361983, 2.586168147248612e-17),
2219.0: (10.705349457553329, 1.193077626005563e-19),
2220.0: (6.295422961441597, 1.2273906014417228e-08),
2221.0: (10.397512213756734, 6.242487884569214e-19),
2222.0: (9.609940606540492, 5.976677859279132e-16),
2223.0: (11.791703671305827, 1.2123670801936643e-23),
2224.0: (9.906479129557278, 3.631592200063803e-18),
2225.0: (10.059699776485454, 1.2799325162048965e-15),
2226.0: (14.432417903373297, 1.751060480334468e-32),
2227.0: (10.457452923864471, 1.6745431808679197e-18),
2228.0: (10.278504902961666, 5.6550120113744974e-18),
2229.0: (8.138105666245567, 1.1771048504146923e-11),
2230.0: (9.334424464586714, 3.006624048406787e-14),
2231.0: (16.684222631682417, 9.154882155420931e-33),
2232.0: (11.61121732544329, 1.0988943823245837e-21),
2233.0: (12.410227903125556, 7.813143752539412e-22),
2234.0: (16.21239674569591, 2.6138008359757516e-35),
2235.0: (13.6758136824299, 1.8161536361522745e-27),
2236.0: (15.209920847601623, 6.774457577313341e-32),
2237.0: (11.436910327653163, 5.678261070384455e-23),
2238.0: (12.033859785053586, 1.424086462914388e-21),
2239.0: (11.371034557379511, 1.4743381920434357e-20),
2240.0: (8.579015421192256, 5.399073103225133e-14),
2241.0: (7.310081786608657, 4.035234832292327e-11),
2242.0: (14.771102496514429, 1.539882058146754e-29)}
```

These results from the t-tests performed for each product category indicate that for every single ProductID

The first number (t-statistic): This measures the size of the difference relative to the variation in

the sample data.

The second number (p-value): This indicates the probability that the observed difference could occur by random chance.(typical thrusthold 0.05)

Consistent Across Categories: This pattern is consistent across all product categories, indicating that discounts are effective in lowering the sale price across the board.

Strategic Implications: The company can be confident that discounting is an effective strategy for changing the sale price of products, regardless of category. The strength of the effect might vary by product, but the direction of the effect (discounts reduce prices) is consistent.

Business Considerations: While discounts lower prices, the company needs to balance this with the potential benefits, such as increased sales volume, clearing out inventory, attracting new customers, or encouraging larger purchases. Each product category might respond differently to discounts in terms of increased sales volume, which isn't captured in this analysis.

The business can use these insights to refine its discounting strategies

0.0.5 Insights

Throughout this detailed analysis of the shoe company's sales data, we've gained several insights into the impact of discount strategies on sale prices and how this varies across product categories. Here's a recap of the key findings and insights:

1. **Impact of Discounts on Sale Price**:
    - A primary analysis revealed that discounts significantly reduce the sale price of items. The initial t-test comparing discounted and non-discounted sale prices across the entire dataset showed a substantial decrease in sale price when discounts were applied.
2. **Statistical Significance**:
    - The t-test provided a t-statistic with a very high absolute value and a p-value of virtually zero, strongly indicating that discounts have a statistically significant effect on lowering the sale price.
3. **Uniformity Across Product Categories**:
    - A deeper analysis with individual t-tests for each product category showed that the impact of discounts on sale price is not just a general trend but is consistent across different product types. Every category tested showed a statistically significant difference in sale prices between discounted and non-discounted items.
4. **Strategic Implications for Pricing**:
    - The insights from the analysis could help the company to develop nuanced pricing strategies. For example, they may choose to offer discounts on products that are overstocked or to stimulate sales during slower seasons. Alternatively, for high-demand items or premium product categories, they may decide to minimize discounts to maximize revenue.
5. **Potential for Further Analysis**:
    - Although the analysis provided clear insights into pricing and discounting, additional data on sales volume and customer purchasing patterns could further enhance understanding. For example, data on units sold would allow for an investigation into whether the reduction in sale price from discounts is offset by an increase in the volume of sales, which is often a primary goal of discounting.
6. **Seasonal and Market Considerations**:

- The analysis suggests potential for examining seasonal variations in discount effectiveness, although this specific aspect was not tested due to system limitations. Seasonal trends could influence the optimal timing and magnitude of discounts.

In conclusion, the analysis strongly supports the effectiveness of discount strategies in influencing sale prices across various product categories. However, the company must consider these insights within the broader context of their sales objectives, market positioning, inventory management, and overall profitability goals. A balanced approach that takes into account both the immediate financial impacts of discounting and the longer-term strategic benefits will be crucial for the company's success.

0.0.6 Summary

In this assignment, we took a deep dive into the sales data of 'Jordhan', a shoe company, to understand how discounts impact the sale prices of their products. Here's a simplified overview of what we did:

1. **Loaded and Cleaned Data**: We started by loading the sales data from a CSV file and cleaned it up, ensuring that the discount percentages and sale prices were in a float format that we could work with numerically.
2. **Initial Analysis**: We first looked at the overall effect of discounts on sale prices by comparing the average sale prices of discounted and non-discounted items. This gave us a clear indication that discounts typically lead to lower sale prices. Also done some plots to get a good idea.
3. **Statistical Testing**: To confirm our observations, we used a t-test, a statistical method, to compare the sale prices. The test results showed a statistically significant difference between the sale prices of discounted and non-discounted items, meaning discounts do indeed lower sale prices.
4. **Deeper Dive by Product Category**: We then examined whether this effect of discounts on sale prices was consistent across different product categories. By performing t-tests for each product category, we found that the trend holds true across the board – discounts reduce sale prices in every category.
5. **Insights and Implications**: From these analyses, we learned that discounting is an effective strategy for reducing prices and potentially increasing sales volume. However, the company needs to balance the lower prices against the total revenue and profitability.
6. **Recommendations**: Based on our findings, we suggested that 'Jordhan' could tailor its discount strategies for different product categories and consider the timing and depth of discounts to optimize sales and profitability.

References

1. Went rhrough all sections over this link helped me to complete this task : https://worldclass.regis.edu/d2l/le/content/329163/viewContent/4700430/Viewand
2. spent some time over Youtube to understand these concepts and implementation towards problem statement : https://www.youtube.com/watch?v=YSwmpAmLV2s

[ ]: