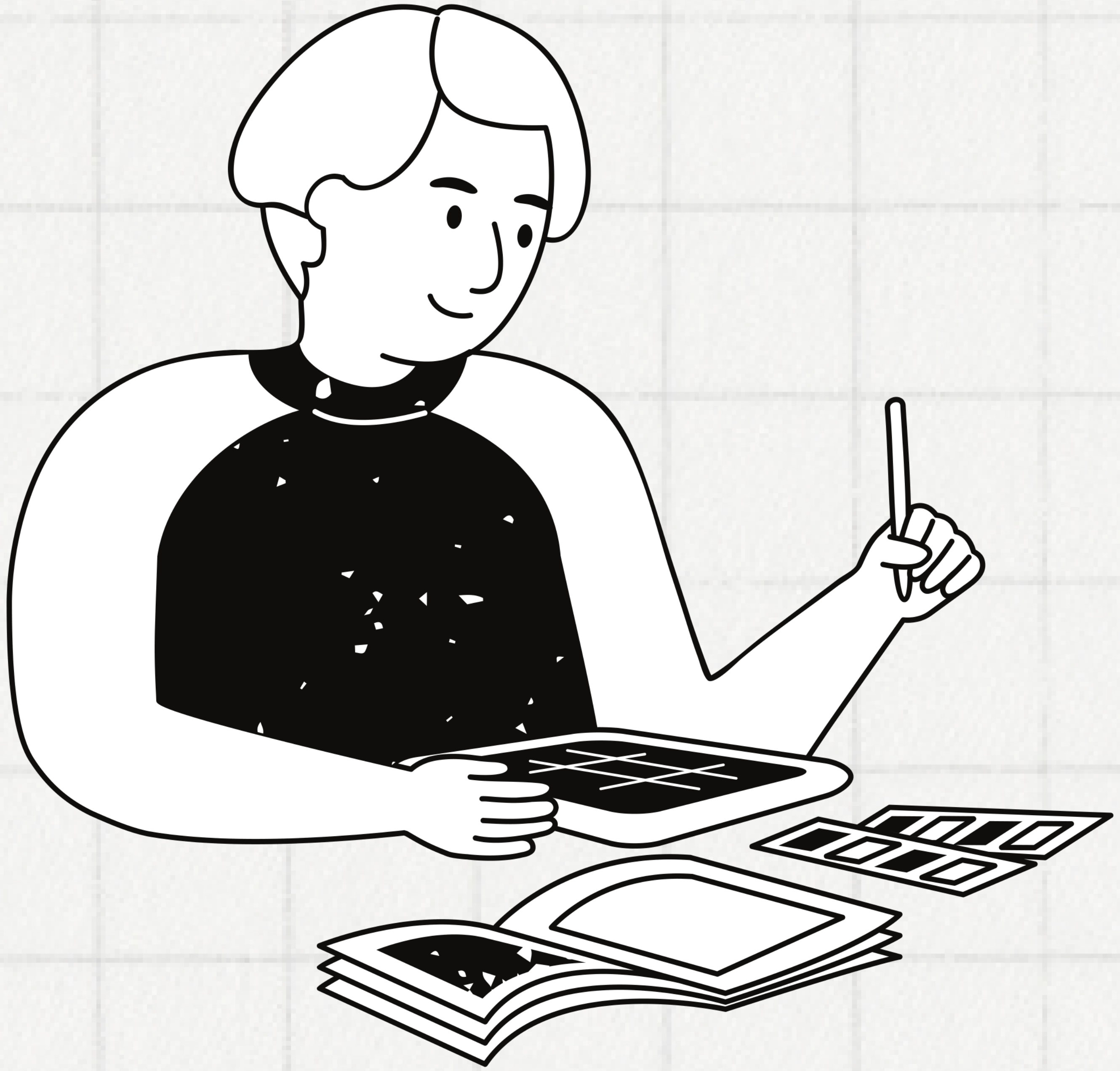
Customer Shopping Trends Analysis

Presented by Dandy Wibowo

Introduction

Project Scopes and Objectives:

The goal of this project is to analyze the shopping trends of customers to gain insights into consumer behavior and preferences. We aim to provide companies with actionable insights to optimize their marketing strategies, improve customer satisfaction and increase overall sales performance by examining various aspects such as customer demographics, buying patterns and product preferences.



Introduction

Dataset Overview:

This analysis utilizes Kaggle's comprehensive dataset that contains information about customer transactions, including demographics, purchase details, subscription status, payment type and more. This dataset provides a rich source of data to explore and discover meaningful patterns and trends in customer behavior.

The dataset includes the following key characteristics:

- Customer demographics: Age, Gender
- Purchase details: Item purchased, purchase amount, category, location, season
- Customer Behavior: Review ratings, subscription status, payment methods
- Transaction History: Previous purchases, purchase frequency
- Other relevant attributes: Size, color, shipping type, discounts applied, promo codes used

Our goal with this data set is to perform in-depth analysis to understand customer preferences, identify key drivers of purchase behavior, and provide companies with valuable insights for strategic decision-making and marketing initiatives.

Customer Analysis

Average age of customers.

```
query_avg_age = '''
SELECT ROUND(AVG(Age)) AS average_age
FROM df
'''
average_age = sqldf(query_avg_age, locals())
print("Average Age of Customers:", int(average_age['average_age'].values[0]))
```

```
Average Age of Customers: 44
```

Our customer base, with an average age of 44, indicates a strong presence among middle-aged (adults) individuals.

Customer Analysis

Distribution of genders among customers.

```
query_gender_distribution = '''
SELECT Gender, COUNT(*) AS count
FROM df
GROUP BY Gender
'''
gender_distribution = sqldf(query_gender_distribution, locals())
print("Distribution of Genders Among Customers:")
print(gender_distribution)
```

```
Distribution of Genders Among Customers:
   Gender  count
0  Female   1248
1    Male   2652
```

The distribution of genders among customers reveals that there are 1,248 female customers and 2,652 male customers. This indicates that male customers constitute a larger portion of the customer base compared to female customers.

Purchase Analysis

Total revenue

```
total_revenue = '''
SELECT SUM(Purchase_Amount) AS Total_revenue
FROM df
'''
total_revenue = sqldf(total_revenue, locals())
print("\nTotal Revenue in USD:")
print(total_revenue)
```

```
Total Revenue in USD:
   Total_revenue
0         233081
```

The total revenue is $233,081.

Purchase Analysis

Top 5 popular item categories

```
# Define SQL query to identify popular items and categories
query = """
    SELECT
        Item_Purchased,
        Category,
        SUM(Purchase_Amount) AS Total_Sales
    FROM
        df
    GROUP BY
        Item_Purchased,
        Category
    ORDER BY
        Total_Sales DESC
    LIMIT 5
"""

# Execute SQL query using PandasSQL
popular_items_categories = sqldf(query, locals())

print(popular_items_categories)
```

```
  Item_Purchased     Category  Total_Sales
0         Blouse     Clothing        10410
1          Shirt     Clothing        10332
2          Dress     Clothing        10320
3          Pants     Clothing        10090
4        Jewelry  Accessories        10010
```

The top five items in the Clothing and Accessories categories, including Blouse, Shirt, Dress, Pants, and Jewelry, have significantly contributed to overall sales. Blouse, with 10,410 units sold, is the most popular item in the Clothing category. Meanwhile, Jewelry, with 10,010 units sold, is a significant success in the Accessories category. These items highlight consumer preferences and drive business sales.

Purchase Analysis

Purchase behavior analysis based on age and gender.

```
query_purchase_behavior = '''
SELECT Gender, CAST(ROUND(AVG(Age)) AS INTEGER) AS average_age, AVG(Purchase_Amount) AS average_purchase_amount
FROM df
GROUP BY Gender
'''
purchase_behavior = sqldf(query_purchase_behavior, locals())
print("\nPurchase Behavior Based on Age and Gender:")
print(purchase_behavior)
```

```
Purchase Behavior Based on Age and Gender:
   Gender  average_age  average_purchase_amount
0  Female           44                60.249199
1    Male           44                59.536199
```

Based on the analysis of purchase behavior by age and gender, it appears that both males and females, on average aged 44, exhibit similar purchasing patterns. However, females tend to have a slightly higher average purchase amount of $60.25 compared to males, who have an average purchase amount of $59.54. These findings suggest that despite minor differences in spending habits, both genders demonstrate consistent purchasing behavior at this age.

Purchase Analysis

Top categories and items purchased by customers.

```
query_top_categories = '''
SELECT Category, COUNT(*) AS count
FROM df
GROUP BY Category
ORDER BY count DESC
LIMIT 5
'''
top_categories = sqldf(query_top_categories, locals())
print("Top Categories Purchased by Customers:")
print(top_categories)
```

```
Top Categories Purchased by Customers:
      Category  count
0     Clothing   1737
1  Accessories   1240
2     Footwear    599
3    Outerwear    324
```

Clothing emerged as the most popular category among customers, with a count of 1737 purchases.

Purchase Analysis

Top categories and items purchased by customers.

```
query_top_items = '''
SELECT Item_Purchased, COUNT(*) AS count
FROM df
GROUP BY Item_Purchased
ORDER BY count DESC
LIMIT 5
'''
top_items = sqldf(query_top_items, locals())
print("\nTop Items Purchased by Customers:")
print(top_items)
```

```
Top Items Purchased by Customers:
  Item_Purchased  count
0          Pants    171
1        Jewelry    171
2         Blouse    171
3          Shirt    169
4          Dress    166
```

The top items purchased by customers include pants, jewelry, blouse, shirt, and dress, with counts of 171, 171, 171, 169, and 166 respectively. These items demonstrate popular choices among customers, showcasing a diverse range of preferences in clothing and accessories.

Purchase Analysis

Favorite Categories and Items Purchased by Customers by Season

```
# Favorit category and item bought by customer by season
query_sales_by_location_and_season = '''
SELECT
    Season,
    Category,
    Item_Purchased,
    COUNT(*) AS total_sales
FROM
    df
GROUP BY
    Season
ORDER BY
    Season,
    Category,
    total_sales DESC;

'''

# Execute the SQL query and store the result in 'result' DataFrame
sales_by_location_and_season = sqldf(query_sales_by_location_and_season, locals())

# Print the result DataFrame
print(sales_by_location_and_season)
```

```
   Season  Category Item_Purchased  total_sales
0    Fall  Clothing          Shirt          975
1  Spring  Clothing          Jeans          999
2  Summer  Footwear       Sneakers          955
3  Winter  Clothing         Blouse          971
```

During the Fall season, the top-selling category was Clothing, with the Shirt being the most popular item, contributing to a total sales of 975 units.
In the Spring season, Clothing remained the preferred category, with Jeans emerging as the favored item, resulting in a total sales of 999 units.
Customers showed a preference for Footwear during the Summer season, with Sneakers being the top-selling item, accounting for a total sales of 955 units.
In the Winter season, Clothing continued to dominate, with the Blouse being the preferred item among customers, leading to a total sales of 971 units.

Purchase Analysis

Customer behavior for purchasing items by age category

```
Age_behavior = '''
WITH age_categories AS(
    SELECT
        Customer_ID,
        Age,
        CASE
            WHEN Age >= 18 AND Age <= 35 THEN 'Youngsters'
            WHEN Age >= 36 AND Age <= 50 THEN 'Adults'
            WHEN Age >= 51 THEN 'Seniors'
            ELSE 'Unknown'
        END AS Age_Category
    FROM df
),
customer_behaviors AS (
    SELECT
        ac.Age_Category,
        df.Category,
        df.Item_Purchased,
        COUNT(*) AS purchase_count
    FROM
        df
    INNER JOIN
        age_categories ac ON df.Customer_ID = ac.Customer_ID
    GROUP BY
        ac.Age_Category,
        df.Category,
        df.Item_Purchased
),
```

```
ranked_customer_behaviors AS (
    SELECT
        Age_Category,
        Category,
        Item_Purchased,
        purchase_count,
        RANK() OVER (PARTITION BY Age_Category, Category ORDER BY purchase_count DESC) AS ranking
    FROM
        customer_behaviors
)
SELECT
    Age_Category,
    Category,
    Item_Purchased,
    purchase_count
FROM
    ranked_customer_behaviors
WHERE
    ranking = 1
ORDER BY
    Age_Category,
    Category;
'''

# Execute the SQL query and store the result in 'result' DataFrame
Age_behaviors = sqldf(Age_behavior, locals())

# Print the result DataFrame
print(Age_behaviors)
```

Purchase Analysis

Customer behavior for purchasing items by age category

	Age_Category	Category	Item_Purchased	purchase_count
0	Adults	Accessories	Scarf	61
1	Adults	Clothing	Pants	58
2	Adults	Footwear	Sandals	55
3	Adults	Outerwear	Jacket	57
4	Seniors	Accessories	Jewelry	74
5	Seniors	Clothing	Blouse	72
6	Seniors	Footwear	Shoes	67
7	Seniors	Outerwear	Coat	62
8	Seniors	Outerwear	Jacket	62
9	Youngsters	Accessories	Belt	57
10	Youngsters	Accessories	Jewelry	57
11	Youngsters	Clothing	Shirt	70
12	Youngsters	Footwear	Sneakers	52
13	Youngsters	Outerwear	Coat	61

Youngsters (Age 18-35):

- Youngsters tend to purchase accessories such as belts and jewelry frequently.
- Clothing items like shirts are also popular among youngsters.
- Footwear choices include sneakers, indicating a preference for casual and sporty styles.
- Outerwear purchases, such as coats, are also notable among this age group.

Purchase Analysis

Customer behavior for purchasing items by age category

Adults (Age 36-50):

- Adults show a preference for accessories like scarves and jewelry.
- Clothing items such as pants and jackets are commonly purchased by adults.
- Footwear choices include sandals, suitable for everyday wear.
- Outerwear purchases, including jackets, are consistent among adults.

Seniors (Age 51 and above):

- Seniors exhibit a strong preference for accessories, particularly jewelry.
- Clothing items like blouses are popular among senior customers.
- Footwear choices include comfortable shoes suitable for daily wear.
- Seniors also show a consistent interest in outerwear, including coats and jackets.

Purchase Analysis

The average purchase amount and customer review ratings

```
# Define SQL query to calculate average purchase amount and review ratings
query = """
    SELECT
        AVG(Purchase_Amount) AS Avg_Purchase_Amount,
        AVG(Review_Rating) AS Avg_Review_Rating
    FROM
        df
"""

# Execute SQL query using PandasSQL
average_purchase_rating = sqldf(query, locals())
print("\nThe average purchase amount and customer review ratings:")
print(average_purchase_rating)
```

```
The average purchase amount and customer review ratings:
   Avg_Purchase_Amount  Avg_Review_Rating
0            59.764359           3.749949
```

The average purchase amount for customers is approximately $59.76. This indicates the typical spending behavior across all purchases. Additionally, the average customer review rating stands at around 3.75 out of 5. This suggests the overall satisfaction level of customers based on their reviews. These metrics provide insights into both the financial aspect and customer satisfaction within the business.

Location Analysis

The highest and lowest sales by location

```
# Define the SQL query to calculate total sales for each location
query_sales_by_location = '''
SELECT Location, SUM(Purchase_Amount) AS total_sales
FROM df
GROUP BY Location
ORDER BY total_sales DESC
'''

# Execute the SQL query and store the result in a DataFrame
sales_by_location = sqldf(query_sales_by_location, locals())

# Extract the location with the highest sales
highest_sales_location = sales_by_location.head(1)

# Extract the location with the lowest sales
lowest_sales_location = sales_by_location.tail(1)

# Print the locations with the highest and lowest sales
print("Location with the highest sales:")
print(highest_sales_location)
print("\nLocation with the lowest sales:")
print(lowest_sales_location)
```

```
Location with the highest sales:
  Location  total_sales
0  Montana         5784

Location with the lowest sales:
   Location  total_sales
49   Kansas         3437
```

The location with the highest sales is Montana, with a total sales amount of 5784 USD.
The location with the lowest sales is Kansas, with a total sales amount of 3437 USD.

Location Analysis

Analyze the distribution of purchases across different locations

```
# Analyze the distribution of purchases across different locations
query_purchase_distribution = '''
SELECT Location, COUNT(*) AS purchase_count
FROM df
GROUP BY Location
ORDER BY purchase_count DESC
'''
purchase_distribution = sqldf(query_purchase_distribution, locals())

print(purchase_distribution)
```

Location Analysis

Analyze the distribution of purchases across different locations

	Location	purchase_count
0	Montana	96
1	California	95
2	Idaho	93
3	Illinois	92
4	Alabama	89
5	Minnesota	88
6	New York	87
7	Nevada	87
8	Nebraska	87
9	Maryland	86
10	Delaware	86
11	Vermont	85
12	Louisiana	84
13	North Dakota	83
14	West Virginia	81
15	New Mexico	81
16	Missouri	81
17	Mississippi	80
18	Kentucky	79
19	Indiana	79
20	Georgia	79
21	Arkansas	79
22	North Carolina	78
23	Connecticut	78
24	Virginia	77
25	Texas	77
26	Tennessee	77
27	Ohio	77
28	Maine	77
29	South Carolina	76
30	Wisconsin	75
31	Oklahoma	75
32	Colorado	75
33	Pennsylvania	74
34	Oregon	74
35	Washington	73
36	Michigan	73
37	Massachusetts	72
38	Alaska	72
39	Wyoming	71
40	Utah	71
41	New Hampshire	71
42	South Dakota	70
43	Iowa	69
44	Florida	68
45	New Jersey	67
46	Hawaii	65
47	Arizona	65
48	Rhode Island	63
49	Kansas	63

The distribution of purchases across various locations reveals interesting insights into customer behavior and regional preferences.

- Montana, California, and Idaho emerged as the top three locations with the highest purchase counts, indicating strong sales volumes in these regions.
- Conversely, Rhode Island, Kansas, and Arizona recorded relatively lower purchase counts, suggesting potential areas for targeted marketing efforts or promotional campaigns to stimulate sales.

Understanding the distribution of purchases across different locations enables us to tailor our strategies and allocate resources effectively to maximize sales and enhance customer satisfaction.

Payment Preferences

Analysis of Preferred Payment Methods and Average Transaction Values

```
# Identify the most preferred payment methods among customers the average transaction value for each payment method
query_preferred_payment_methods = '''
SELECT Payment_Method, COUNT(*) AS count, AVG(Purchase_Amount) AS avg_transaction_value
FROM df
GROUP BY Payment_Method
ORDER BY count DESC
'''
preferred_payment_methods = sqldf(query_preferred_payment_methods, locals())
print(preferred_payment_methods)
```

```
  Payment_Method  count  avg_transaction_value
0         PayPal    677              59.245199
1    Credit Card    671              60.074516
2           Cash    670              59.704478
3     Debit Card    636              60.915094
4          Venmo    634              58.949527
5  Bank Transfer    612              59.712418
```

Payment Method Preference: Customers exhibit a relatively balanced preference for payment methods, with PayPal, Credit Card, and Cash being the top three choices. Debit Card and Venmo follow closely behind, while Bank Transfer is slightly less favored.

Average Transaction Values: Across all payment methods, the average transaction values are relatively consistent, ranging from approximately $58.95 to $60.91. This consistency suggests that payment method choice does not significantly impact the average transaction value, indicating a stable purchasing behavior regardless of the payment method used.

Subscription Analysis

Percentage of customers with subscriptions

```
# Percentage of customers with subscriptions
query_subscription_percentage = '''
SELECT
    AVG(CASE WHEN Subscription_Status = 'Yes' THEN 1 ELSE 0 END) * 100 AS subscription_percentage
FROM df
'''

# Execute the SQL query
subscription_percentage = sqldf(query_subscription_percentage, locals())

# Print the result
print("Percentage of customers with subscriptions:", subscription_percentage['subscription_percentage'].values[0])
```

```
Percentage of customers with subscriptions: 27.0
```

The analysis reveals that approximately 27.0% of customers have subscriptions, indicating a significant portion of the customer base opting for subscription services. This insight highlights the importance of subscription programs in retaining customers and fostering loyalty within the customer base.

Subscription Analysis

Impact of Subscriptions on Revenue and Purchase Frequency

```python
# Join to calculate the impact of subscriptions on revenue and purchase frequency
query_subscription_impact = '''
SELECT
    t1.Subscription_Status,
    t1.total_revenue,
    t1.avg_purchase_amount,
    t2.purchase_frequency
FROM
    (SELECT
        Subscription_Status,
        SUM(Purchase_Amount) AS total_revenue,
        AVG(Purchase_Amount) AS avg_purchase_amount
    FROM df
    GROUP BY Subscription_Status) AS t1
JOIN
    (SELECT
        Subscription_Status,
        COUNT(*) AS purchase_frequency
    FROM df
    GROUP BY Subscription_Status) AS t2
ON
    t1.Subscription_Status = t2.Subscription_Status
'''

# Execute the queries using pandasql
subscription_impact = sqldf(query_subscription_impact, locals())
print(subscription_impact)
```

	Subscription_Status	total_revenue	avg_purchase_amount	purchase_frequency
0	No	170436	59.865121	2847
1	Yes	62645	59.491928	1053

The study reveals that customers with a "No" subscription earn $170,436, with an average purchase of $59.87 and a frequency of 2,847 transactions. On the other hand, customers with a "Yes" subscription earn $62,645, with an average purchase of $59.49 and a frequency of 1,053 transactions. This suggests that targeted marketing and loyalty programs could encourage repeat purchases among subscribers.

Frequency Analysis

Frequency distribution of purchases

```python
# Frequency distribution of purchases
query_purchase_frequency = '''
SELECT
    Frequency_of_Purchases,
    COUNT(*) AS purchase_count
FROM df
GROUP BY Frequency_of_Purchases
ORDER BY Frequency_of_Purchases
'''

# Execute the SQL query
purchase_frequency_distribution = sqldf(query_purchase_frequency, locals())

# Print the result
print("Frequency Distribution of Purchases:")
print(purchase_frequency_distribution)
```

```
Frequency Distribution of Purchases:
  Frequency_of_Purchases  purchase_count
0               Annually             572
1              Bi-Weekly             547
2         Every 3 Months             584
3            Fortnightly             542
4                Monthly             553
5              Quarterly             563
6                 Weekly             539
```

The frequency distribution of purchases reveals the following distribution across different purchase frequencies:

- Annually: 572 purchases
- Bi-Weekly: 547 purchases
- Every 3 Months: 584 purchases
- Fortnightly: 542 purchases
- Monthly: 553 purchases
- Quarterly: 563 purchases
- Weekly: 539 purchases

This analysis provides valuable insights into how frequently customers make purchases, with a varied distribution across different frequency categories.

Frequency Analysis

Seasonality in Purchase Frequency

```
# Trends or seasonality in purchase frequency
query_seasonality = '''
SELECT
    Season,
    Frequency_of_Purchases,
    COUNT(*) AS purchase_count
FROM df
GROUP BY Season, Frequency_of_Purchases
ORDER BY Season, Frequency_of_Purchases
'''

# Execute the SQL query for seasonality
seasonality = sqldf(query_seasonality, locals())

# Print the seasonality analysis
print("\nSeasonality in Purchase Frequency:")
print(seasonality)
```

Frequency Analysis

Seasonality in Purchase Frequency

Seasonality in Purchase Frequency:

	Season	Frequency_of_Purchases	purchase_count
0	Fall	Annually	157
1	Fall	Bi-Weekly	120
2	Fall	Every 3 Months	147
3	Fall	Fortnightly	143
4	Fall	Monthly	137
5	Fall	Quarterly	143
6	Fall	Weekly	128
7	Spring	Annually	137
8	Spring	Bi-Weekly	155
9	Spring	Every 3 Months	146
10	Spring	Fortnightly	133
11	Spring	Monthly	144
12	Spring	Quarterly	142
13	Spring	Weekly	142
14	Summer	Annually	143
15	Summer	Bi-Weekly	132
16	Summer	Every 3 Months	152
17	Summer	Fortnightly	127
18	Summer	Monthly	133
19	Summer	Quarterly	144
20	Summer	Weekly	124
21	Winter	Annually	135
22	Winter	Bi-Weekly	140
23	Winter	Every 3 Months	139
24	Winter	Fortnightly	139
25	Winter	Monthly	139
26	Winter	Quarterly	134
27	Winter	Weekly	145

The analysis of seasonality in purchase frequency indicates variations in buying patterns across different seasons:

- In Fall, the purchase frequency is distributed across various frequency categories, with Every Annualy and Every 3 months purchases being particularly prominent.
- Spring sees a relatively balanced distribution of purchase frequency, with Bi-Weekly purchases slightly more prevalent.
- Summer exhibits a similar pattern to Fall, with Every 3 Months purchases being notable.
- Winter shows consistency in purchase frequency across all categories, with Weekly purchases being slightly higher.

These insights provide valuable information for understanding customer behavior and adjusting marketing strategies accordingly to capitalize on seasonal trends.

Discount Analysis

Impact of Discounts on Purchases

```
query_discount_impact = '''
SELECT Discount_Applied, COUNT(*) AS purchase_count
FROM df
GROUP BY Discount_Applied
'''
discount_impact = sqldf(query_discount_impact, locals())
print("\nImpact of Discounts on Purchases:")
print(discount_impact)
```

```
Impact of Discounts on Purchases:
  Discount_Applied  purchase_count
0               No            2223
1              Yes            1677
```

When discounts are applied, there is a decrease in the total purchase count compared to when no discounts are applied. Specifically, purchases with discounts applied account for 1,677 transactions, whereas purchases with no discounts applied comprise 2,223 transactions.

Discount Analysis

Impact of Promo Codes on Purchases

```python
# Define SQL query to calculate the impact of promo codes on purchases including percentage
query_promo_code_impact = '''
SELECT
    Promo_Code_Used,
    COUNT(*) AS purchase_count,
    ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM df)) AS percentage
FROM df
GROUP BY Promo_Code_Used
'''

# Execute the SQL query
promo_code_impact = sqldf(query_promo_code_impact, locals())

# Print the result
print("\nImpact of Promo Codes on Purchases:")
print(promo_code_impact)
```

```
Impact of Promo Codes on Purchases:
  Promo_Code_Used  purchase_count  percentage
0              No            2223        57.0
1             Yes            1677        43.0
```

Out of the total purchases analyzed, 1,677 transactions utilized promo codes, while 2,223 transactions did not. The data also suggests that a significant portion of customers (approximately 43%) availed promo codes during their purchases.

More Detail about project

www.kaggle.com/code/andywow/customer-shopping-trends-analysis

Thank you very much!

dandywibowo.github.io

dandy.wibowo@outlook.com