[3]:
```
import pandas as pd

```

[4]:
```
df=pd.read_csv('customer_shopping_behavior.csv')
```

[5]:
```
df.head()
```

[5]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Pay Me
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Ve
1	2	19	Male	Sweater	Clothing	64	Maine	L	Maroon	Winter	3.1	Yes	Express	Yes	Yes	2	
2	3	50	Male	Jeans	Clothing	73	Massachusetts	S	Maroon	Spring	3.1	Yes	Free Shipping	Yes	Yes	23	(
3	4	21	Male	Sandals	Footwear	90	Rhode Island	M	Maroon	Spring	3.5	Yes	Next Day Air	Yes	Yes	49	P
4	5	45	Male	Blouse	Clothing	49	Oregon	M	Turquoise	Spring	2.7	Yes	Free Shipping	Yes	Yes	31	P

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             3900 non-null   int64
 1   Age                     3900 non-null   int64
 2   Gender                  3900 non-null   object
 3   Item Purchased          3900 non-null   object
 4   Category                3900 non-null   object
 5   Purchase Amount (USD)   3900 non-null   int64
 6   Location                3900 non-null   object
 7   Size                    3900 non-null   object
 8   Color                   3900 non-null   object
 9   Season                  3900 non-null   object
 10  Review Rating           3863 non-null   float64
 11  Subscription Status     3900 non-null   object
 12  Shipping Type           3900 non-null   object
 13  Discount Applied        3900 non-null   object
 14  Promo Code Used         3900 non-null   object
 15  Previous Purchases      3900 non-null   int64
 16  Payment Method          3900 non-null   object
 17  Frequency of Purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

[5]:
```
df.describe()
```

[5]:

	Customer ID	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3863.000000	3900.000000
mean	1950.500000	44.068462	59.764359	3.750065	25.351538
std	1125.977353	15.207589	23.685392	0.716983	14.447125
min	1.000000	18.000000	20.000000	2.500000	1.000000
25%	975.750000	31.000000	39.000000	3.100000	13.000000
50%	1950.500000	44.000000	60.000000	3.800000	25.000000
75%	2925.250000	57.000000	81.000000	4.400000	38.000000
max	3900.000000	70.000000	100.000000	5.000000	50.000000

[8]:
```
df.describe(include="all")
```

[8]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used
count	3900.000000	3900.000000	3900	3900	3900	3900.000000	3900	3900	3900	3900	3863.000000	3900	3900	3900	3900
unique	NaN	NaN	2	25	4	NaN	50	4	25	4	NaN	2	6	2	2
top	NaN	NaN	Male	Blouse	Clothing	NaN	Montana	M	Olive	Spring	NaN	No	Free Shipping	No	No
freq	NaN	NaN	2652	171	1737	NaN	96	1755	177	999	NaN	2847	675	2223	2223
mean	1950.500000	44.068462	NaN	NaN	NaN	59.764359	NaN	NaN	NaN	NaN	3.750065	NaN	NaN	NaN	NaN
std	1125.977353	15.207589	NaN	NaN	NaN	23.685392	NaN	NaN	NaN	NaN	0.716983	NaN	NaN	NaN	NaN
min	1.000000	18.000000	NaN	NaN	NaN	20.000000	NaN	NaN	NaN	NaN	2.500000	NaN	NaN	NaN	NaN
25%	975.750000	31.000000	NaN	NaN	NaN	39.000000	NaN	NaN	NaN	NaN	3.100000	NaN	NaN	NaN	NaN
50%	1950.500000	44.000000	NaN	NaN	NaN	60.000000	NaN	NaN	NaN	NaN	3.800000	NaN	NaN	NaN	NaN
75%	2925.250000	57.000000	NaN	NaN	NaN	81.000000	NaN	NaN	NaN	NaN	4.400000	NaN	NaN	NaN	NaN
max	3900.000000	70.000000	NaN	NaN	NaN	100.000000	NaN	NaN	NaN	NaN	5.000000	NaN	NaN	NaN	NaN

[9]:
```
df.isnull().sum()
```

[9]:
```
Customer ID                0
Age                        0
Gender                     0
Item Purchased             0
Category                   0
Purchase Amount (USD)      0
Location                   0
Size                       0
Color                      0
Season                     0
Review Rating             37
Subscription Status        0
Shipping Type              0
Discount Applied           0
Promo Code Used            0
Previous Purchases         0
Payment Method             0
Frequency of Purchases     0
dtype: int64
```

[10]:
```
# Fill the Null values using median by its Category

df['Review Rating']= df.groupby('Category')['Review Rating'].transform(lambda x: x.fillna(x.median()))
```

[10]:
```
df.isnull().sum()
```

[10]:
```
Customer ID               0
Age                       0
Gender                    0
Item Purchased            0
Category                  0
Purchase Amount (USD)     0
Location                  0
Size                      0
Color                     0
Season                    0
Review Rating             0
Subscription Status       0
Shipping Type             0
Discount Applied          0
Promo Code Used           0
Previous Purchases        0
Payment Method            0
Frequency of Purchases    0
dtype: int64
```

[11]:
```
df.columns=df.columns.str.lower()
df.columns=df.columns.str.replace(' ','_')
df=df.rename(columns={'purchase_amount_(usd)':'purchase_amount'})
```

[12]:
```
df.columns
```

[12]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'promo_code_used', 'previous_purchases',
       'payment_method', 'frequency_of_purchases'],
      dtype='object')
```

[13]:
```
# Create A New Column of Age_group

labels=['Young Adult','Adult','Middle-aged','Senior']
df['age_group']=pd.qcut(df['age'], q=4,labels=labels)
```

[14]:
```
df[['age','age_group']].head(15)
```

[14]:

	age	age_group
0	55	Middle-aged
1	19	Young Adult
2	50	Middle-aged
3	21	Young Adult
4	45	Middle-aged
5	46	Middle-aged
6	63	Senior
7	27	Young Adult
8	26	Young Adult
9	57	Middle-aged
10	53	Middle-aged
11	30	Young Adult
12	61	Senior
13	65	Senior
14	64	Senior

[15]:
```
# New column purchase_frequency_days

frequency_mapping={
    'Fortnightly':14,
    'Weekly':7,
    'Monthly':30,
    'Quarterly':90,
    'Bi-Weekly':14,
    'Annually':365,
    'Evert 3 Months':90
}

df['purchase_frequency_days']=df['frequency_of_purchases'].map(frequency_mapping)
```

[16]:
```
df[['purchase_frequency_days','frequency_of_purchases']].head(15)

```

[16]:

	purchase_frequency_days	frequency_of_purchases
0	14.0	Fortnightly
1	14.0	Fortnightly
2	7.0	Weekly
3	7.0	Weekly
4	365.0	Annually
5	7.0	Weekly
6	90.0	Quarterly
7	7.0	Weekly
8	365.0	Annually
9	90.0	Quarterly
10	14.0	Bi-Weekly
11	14.0	Fortnightly
12	14.0	Fortnightly
13	7.0	Weekly
14	7.0	Weekly

[17]:
```
df[['discount_applied','promo_code_used']].head(15)

```

[17]:

	discount_applied	promo_code_used
0	Yes	Yes
1	Yes	Yes
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
5	Yes	Yes
6	Yes	Yes
7	Yes	Yes
8	Yes	Yes

[18]:
```
#check all values in both col are same or Not

(df['discount_applied']==df['promo_code_used']).all()
```

[18]:
```
True
```

[19]:
```
# Both col has same data values so we drop 1 column = 'promo_code_used'

df=df.drop('promo_code_used',axis=1)
```

[20]:
```
df.columns

```

[20]:
```
Index(['customer_id', 'age', 'gender', 'item_purchased', 'category',
       'purchase_amount', 'location', 'size', 'color', 'season',
       'review_rating', 'subscription_status', 'shipping_type',
       'discount_applied', 'previous_purchases', 'payment_method',
       'frequency_of_purchases', 'age_group', 'purchase_frequency_days'],
      dtype='object')
```

[21]:
```
pip install psycopg2-binary sqlalchemy
```

```
Requirement already satisfied: psycopg2-binary in c:\users\cprav\anaconda3\lib\site-packages (2.9.11)
Requirement already satisfied: sqlalchemy in c:\users\cprav\anaconda3\lib\site-packages (2.0.34)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\cprav\anaconda3\lib\site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\cprav\anaconda3\lib\site-packages (from sqlalchemy) (3.0.1)
Note: you may need to restart the kernel to use updated packages.
```

[22]:
```
from sqlalchemy import create_engine

# Step 1: Connect to PostgreSQL
# Replace placeholders with your actual details
username = "postgres"      # default user
password = "cp*#2110" # the password you set during installation
host = "localhost"         # if running locally
port = "5432"              # default PostgreSQL port
database = "customer_behavior"    # the database you created in pgAdmin

engine = create_engine(f"postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}")

# Step 2: Load DataFrame into PostgreSQL
table_name = "customer"   # choose any table name
df.to_sql(table_name, engine, if_exists="replace", index=False)

print(f"Data successfully loaded into table '{table_name}' in database '{database}'.")
```

```
Data successfully loaded into table 'customer' in database 'customer_behavior'.
```

[ ]:
```

```

```sql
--Q1. What is the total revenue generated by male vs. female customers?
select gender, SUM(purchase_amount) as revenue
from customer
group by gender;
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page No:

	gender text	revenue numeric
1	Female	75191
2	Male	157890

```sql
--Q2. Which customers used a discount but still spent more than the average purchase amount?
select customer_id, purchase_amount
from customer
where discount_applied = 'Yes' and purchase_amount >= (select AVG(purchase_amount) from customer);
```

Data Output | Messages | Notifications

Showing rows: 1 to 839 | Page No: 1 of 1

	customer_id bigint	purchase_amount bigint
1	2	64
2	3	73
3	4	90
4	7	85
5	9	97
6	12	68
7	13	72
8	16	81
9	20	90
10	22	62
11	24	88
12	29	94
13	32	79
14	33	67
15	35	91
16	37	69

Total rows: 839 | Query complete 00:00:00.270 | CRLF

```sql
-- Q3. Which are the top 5 products with the highest average review rating?
select item_purchased, round(avg(review_rating::numeric),2) as "Average Product Rating"
from customer
group by item_purchased
order by avg(review_rating) desc
limit 5;
```

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page

	item_purchased text	Average Product Rating numeric
1	Gloves	3.86
2	Sandals	3.84
3	Boots	3.82
4	Hat	3.80
5	Skirt	3.78

```sql
--Q4. Compare the average Purchase Amounts between Standard and Express Shipping.
select shipping_type,
ROUND(AVG(purchase_amount),2)
from customer
where shipping_type in ('Standard','Express')
group by shipping_type;
```

Data Output | Messages | Notifications

Showing rows: 1 to 2

	shipping_type text	round numeric
1	Standard	58.46
2	Express	60.48

```sql
--Q5. Do subscribed customers spend more? Compare average spend and total revenue
--between subscribers and non-subscribers.
SELECT subscription_status,
       COUNT(customer_id) AS total_customers,
       ROUND(AVG(purchase_amount),2) AS avg_spend,
       ROUND(SUM(purchase_amount),2) AS total_revenue
FROM customer
GROUP BY subscription_status
ORDER BY total_revenue,avg_spend DESC;
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page

	subscription_status text	total_customers bigint	avg_spend numeric	total_revenue numeric
1	Yes	1053	59.49	62645.00
2	No	2847	59.87	170436.00

```sql
--Q6. Which 5 products have the highest percentage of purchases with discounts applied?
SELECT item_purchased,
       ROUND(100.0 * SUM(CASE WHEN discount_applied = 'Yes' THEN 1 ELSE 0 END)/COUNT(*),2) AS discount_rate
FROM customer
GROUP BY item_purchased
ORDER BY discount_rate DESC
LIMIT 5;

```

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page No: 1 of 1

	item_purchased text	discount_rate numeric
1	Hat	50.00
2	Sneakers	49.66
3	Coat	49.07
4	Sweater	48.17
5	Pants	47.37

```sql
--Q7. Segment customers into New, Returning, and Loyal based on their total
-- number of previous purchases, and show the count of each segment.
with customer_type as (
SELECT customer_id, previous_purchases,
CASE
    WHEN previous_purchases = 1 THEN 'New'
    WHEN previous_purchases BETWEEN 2 AND 10 THEN 'Returning'
    ELSE 'Loyal'
    END AS customer_segment
FROM customer)

select customer_segment,count(*) AS "Number of Customers"
from customer_type
group by customer_segment;

```

Data Output | Messages | Notifications

Showing rows: 1 to 3

	customer_segment text	Number of Customers bigint
1	Loyal	3116
2	New	83
3	Returning	701

```sql
--Q8. What are the top 3 most purchased products within each category?
WITH item_counts AS (
    SELECT category,
           item_purchased,
           COUNT(customer_id) AS total_orders,
           ROW_NUMBER() OVER (PARTITION BY category ORDER BY COUNT(customer_id) DESC) AS item_rank
    FROM customer
    GROUP BY category, item_purchased
)
SELECT item_rank,category, item_purchased, total_orders
FROM item_counts
WHERE item_rank <=3;
```

Data Output | Messages | Notifications

Showing rows: 1 to 11 | Page No: 1

	item_rank bigint	category text	item_purchased text	total_orders bigint
1	1	Accessories	Jewelry	171
2	2	Accessories	Sunglasses	161
3	3	Accessories	Belt	161
4	1	Clothing	Blouse	171
5	2	Clothing	Pants	171
6	3	Clothing	Shirt	169
7	1	Footwear	Sandals	160
8	2	Footwear	Shoes	150
9	3	Footwear	Sneakers	145
10	1	Outerwear	Jacket	163
11	2	Outerwear	Coat	161

Total rows: 11 | Query complete 00:00:00.132

```sql
--Q10. What is the revenue contribution of each age group?
SELECT
    age_group,
    SUM(purchase_amount) AS total_revenue
FROM customer
GROUP BY age_group
ORDER BY total_revenue desc;
```

Data Output | Messages | Notifications

	age_group text	total_revenue numeric
1	Young Adult	62143
2	Middle-aged	59197
3	Adult	55978
4	Senior	55763

```sql
--Q9. Are customers who are repeat buyers (more than 5 previous purchases) also likely to subscribe?
SELECT subscription_status,
       COUNT(customer_id) AS repeat_buyers
FROM customer
WHERE previous_purchases > 5
GROUP BY subscription_status;
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page No: 1

	subscription_status text	repeat_buyers bigint
1	No	2518
2	Yes	958

Customer Behavior Dashboard

Subscription Status
No | Yes

Gender
Fem... | Male

Category
Accessories
Clothing
Footwear
Outerwear

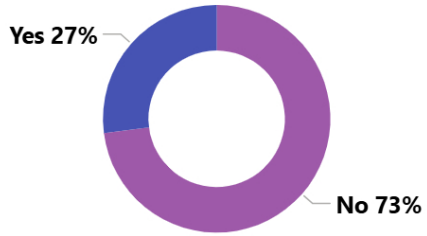
Shipping Type
- 2-Day Shipping
- Express
- Free Shipping
- Next Day Air
- Standard
- Store Pickup

3.9K
Number of Customers

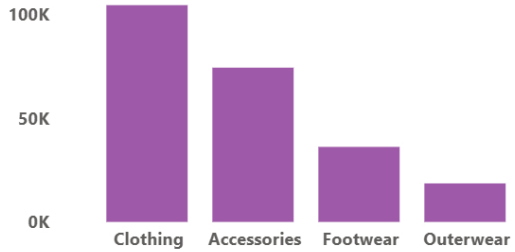
$59.76
Average Purchase Amount

3.75
Average Review Rating

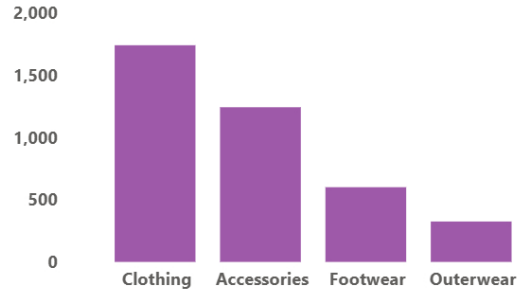
Customers % by Subscription Status



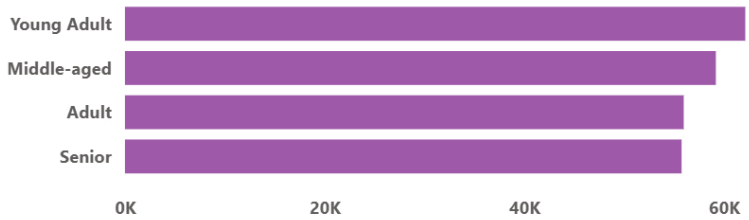
Revenue by Category



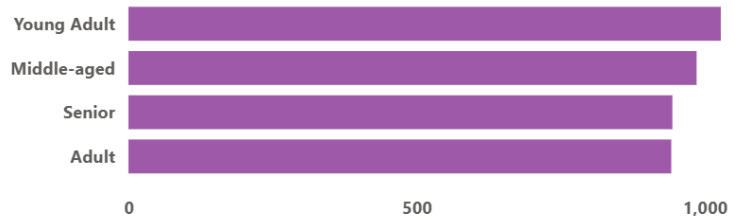
Sales by Category



Revenue by Age Group



Sales by Age Group



Customer Behavior Dashboard

Subscription Status

No

Gender

Male

Category

Accessories

Clothing

Footwear

Outerwear

Shipping Type

- 2-Day Shipping
- Express
- Free Shipping
- Next Day Air
- Standard
- Store Pickup

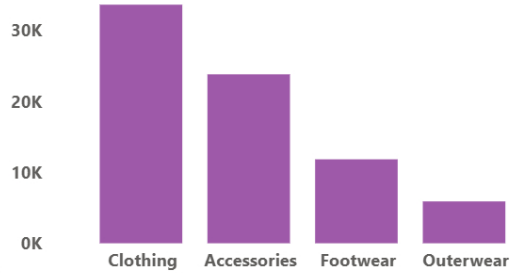
1.248K
Number of Customers

$60.25
Average Purchase Amount

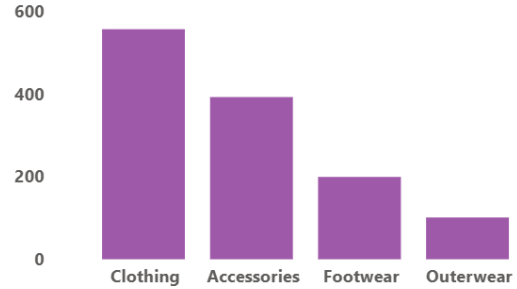
3.74
Average Review Rating

Customers % by Subscription Status

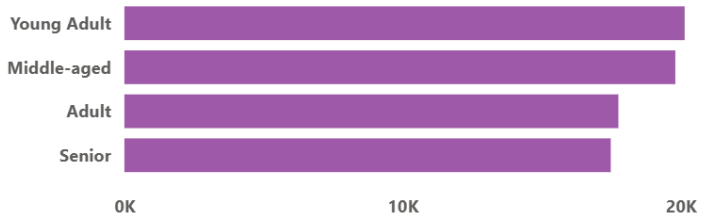
Revenue by Category



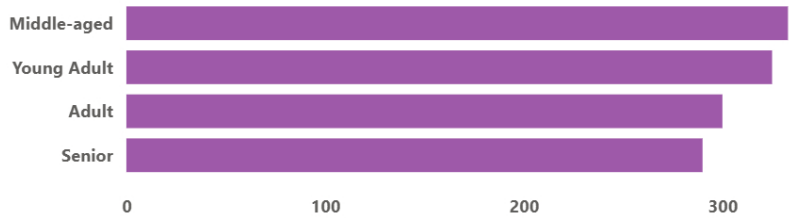
Sales by Category



Revenue by Age Group



Sales by Age Group



Customer Behavior Dashboard

Subscription Status
No | Yes

Gender
Fem... | Male

Category
Accessories
Clothing
Footwear
Outerwear

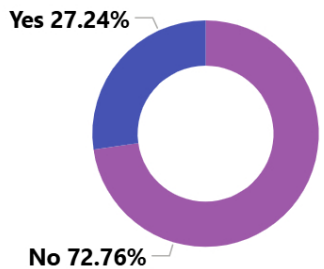
Shipping Type
- 2-Day Shipping
- Express
- Free Shipping
- Next Day Air
- Standard
- Store Pickup

290
Number of Customers

$61.04
Average Purchase Amount

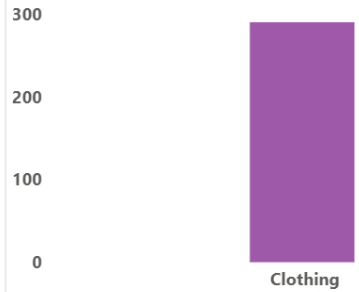
3.76
Average Review Rating

Customers % by Subscription Status

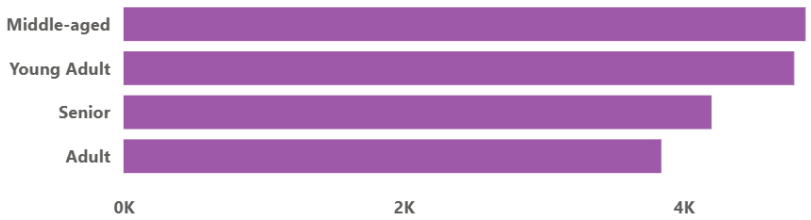


Revenue by Category

Sales by Category



Revenue by Age Group



Sales by Age Group