













```
In [134...]: # Set reference date for 'Recency' (usually the most recent transaction + 1 day)
reference_date = df['Date'].max() + pd.Timedelta(days=1)
# Group by CustomerID and calculate RFM metrics
rfm = df.groupby('CustomerID').agg({
    'Date': lambda x: (reference_date - x.max()).days,  # Recency
    'TransactionID': 'count',                            # Frequency
    'Total Amount': 'sum'                                # Monetary
}).reset_index()

# Rename columns
rfm.columns = ['CustomerID', 'Recency', 'Frequency', 'Monetary']
print(rfm)
```



```
Category
Clothing      5857
Decorations   5897
Electronics   6226
Food          6021
Toys          6105
Name: Quantity, dtype: int64
```

In [125...]

```
Category = ['Clothing', 'Decorations', 'Electronics', 'Food', 'Toys']
TotalProductSold = [5857, 5897, 6226, 6021, 6105]

fig = go.Figure(data=[go.Pie(labels=Category, values=TotalProductSold, hole=0.5)])
fig.update_traces(textinfo='percent+label', pull=[0, 0, 0.1, 0, 0]) # Set the layout
# Add title and labels
fig.update_layout(
    title="Pie Chart of Product Categories",
    title_x=0.5,
    template="plotly_dark",
    showlegend=True
)
fig.show()
```

In [126...]

```
# Total sales of products sold in each category.
```







```
# Layout Adjustment
fig.update_layout(
    template='plotly_dark',
    title='₹ Total Sales by Payment Method (in ₹)',
    xaxis_title='Payment Type',
    yaxis_title='Total Sales (₹)',
    font=dict(family='Arial', size=14),
    uniformtext_minsize=12,
    uniformtext_mode='hide',
    margin=dict(t=60, b=60)
)
fig.show()
```

## 🎁 Return Analysis & Satisfaction

In [162...]

category\_returns









