Category_Sales_Visualization

September 21, 2025

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sqlalchemy import create_engine
from urllib.parse import quote_plus

# Set a clean style for the plots
sns.set_style("whitegrid")
plt.style.use("seaborn-v0_8-deep")

# ==========================================
# === Database Connection and Data Loading
# ==========================================

# Database credentials
user = "root"
password = "Root7878"
host = "localhost"
port = 3306
database = "DataWarehouse"

# Encode password safely (important if it has special chars like @ or $)
password = quote_plus(password)

# Create SQLAlchemy engine
try:
    engine = create_engine(f"mysql+pymysql://{user}:{password}@{host}:{port}/
 ↪{database}")

    # SQL query to get the category sales data
    sql_query = """
    WITH category_sales AS (
        SELECT
            category,
            SUM(sales_amount) total_sales
        FROM fact_sales f
        LEFT JOIN dim_products p
```

```
        ON p.product_key = f.product_key
        GROUP BY category
    )

    SELECT
        category,
        total_sales,
        SUM(total_sales) OVER () overall_sales,
        CAST(total_sales AS FLOAT) / SUM(total_sales) OVER() AS␣
 ↪percentage_of_total
    FROM category_sales
    ORDER BY total_sales DESC;
    """
    df = pd.read_sql(sql_query, engine)

    print("DataFrame Head:")
    print(df.head())
    print("-" * 50)

except Exception as e:
    print(f"Error connecting to the database or loading data: {e}")
    print("Please ensure your database credentials are correct and the database␣
 ↪is running.")
    df = pd.DataFrame()

# ================================
# === Data Visualizations
# ================================

if not df.empty:

    # Sort data for better visualization
    df = df.sort_values(by='total_sales', ascending=False)

    # Define explode values: explode a slice if its percentage is small
    explode_threshold = 0.05  # Explode any slice that is less than 5%
    explode_values = [0.1 if p < explode_threshold else 0 for p in␣
 ↪df['percentage_of_total']]

    plt.figure(figsize=(10, 8))

    # Create the pie chart
    wedges, texts, autotexts = plt.pie(
        df['total_sales'],
        autopct='%1.1f%%',
        startangle=90,
        colors=sns.color_palette('pastel'),
```

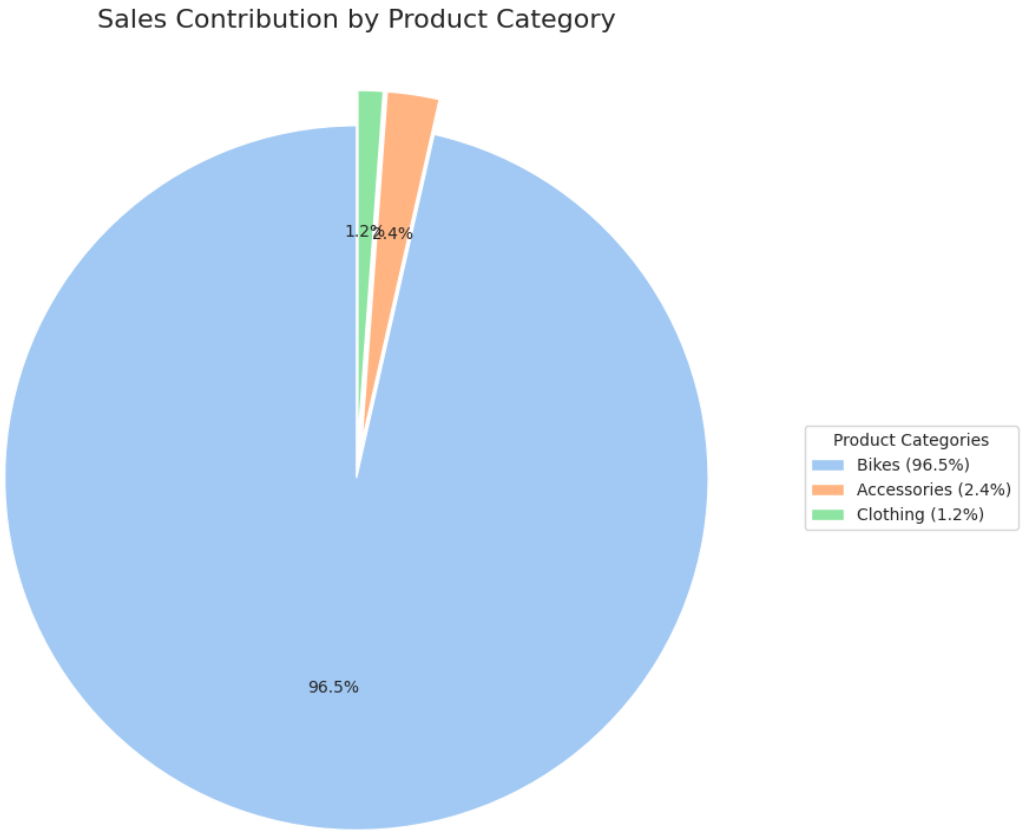
```
        explode=explode_values,
        wedgeprops={'edgecolor': 'white'}
    )

    # Create a new legend with labels and percentages
    labels = [f'{c} ({p:.1%})' for c, p in zip(df['category'],␣
↪df['percentage_of_total'])]
    plt.legend(wedges, labels, title="Product Categories", loc="center left",␣
↪bbox_to_anchor=(1, 0, 0.5, 1))

    plt.title('Sales Contribution by Product Category', fontsize=16)
    plt.ylabel('')
    plt.tight_layout()
    plt.show()

else:
    print("DataFrame is empty. No visualizations will be generated.")
```

```
DataFrame Head:
      category  total_sales  overall_sales  percentage_of_total
0        Bikes   28316272.0     29356250.0             0.964574
1  Accessories     700262.0     29356250.0             0.023854
2     Clothing     339716.0     29356250.0             0.011572
--------------------------------------------------
```


Sales Contribution by Product Category
1.2%
2.4%
96.5%
Product Categories
Bikes (96.5%)
Accessories (2.4%)
Clothing (1.2%)