Sales_Trends_Visualization

September 21, 2025

[1]:
```
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 Connections
# ================================

# 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 for monthly sales trends
    monthly_sql_query = """
    SELECT
        MONTHNAME(order_date) AS order_month,
        SUM(sales_amount) AS total_sales,
        COUNT(DISTINCT customer_key) AS total_customers,
        SUM(quantity) AS total_quantity
    FROM fact_sales
```

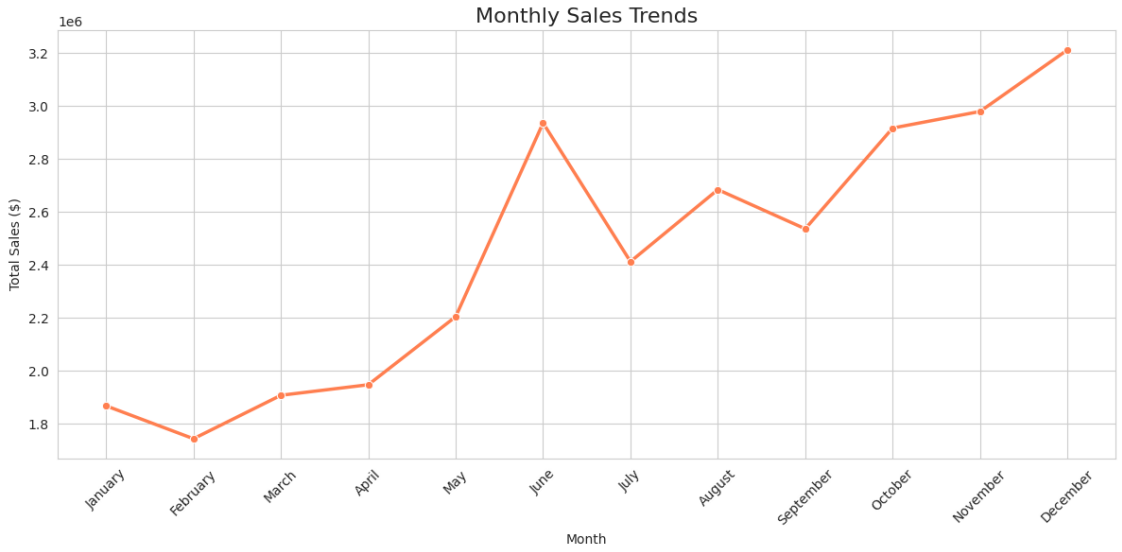
```
    WHERE order_date IS NOT NULL
    GROUP BY
        MONTH(order_date),
        MONTHNAME(order_date)
    ORDER BY MONTH(order_date);
    """
    df_monthly = pd.read_sql(monthly_sql_query, engine)

    # SQL query for yearly sales trends
    yearly_sql_query = """
    SELECT
        YEAR(order_date) AS order_year,
        SUM(sales_amount) AS total_sales,
        COUNT(DISTINCT customer_key) AS total_customers,
        SUM(quantity) AS total_quantity
    FROM fact_sales
    WHERE order_date IS NOT NULL
    GROUP BY
        YEAR(order_date)
    ORDER BY
        YEAR(order_date);
    """
    df_yearly = pd.read_sql(yearly_sql_query, engine)

    print("Monthly DataFrame Head:")
    print(df_monthly.head())
    print("-" * 50)
    print("Yearly DataFrame Head:")
    print(df_yearly.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_monthly = pd.DataFrame()
    df_yearly = pd.DataFrame()
```

```
Monthly DataFrame Head:
  order_month  total_sales  total_customers  total_quantity
0     January    1868558.0             1818          4043.0
1    February    1744517.0             1765          3858.0
2       March    1908375.0             1982          4449.0
3       April    1948226.0             1916          4355.0
4         May    2204969.0             2074          4781.0
--------------------------------------------------
Yearly DataFrame Head:
   order_year  total_sales  total_customers  total_quantity
```

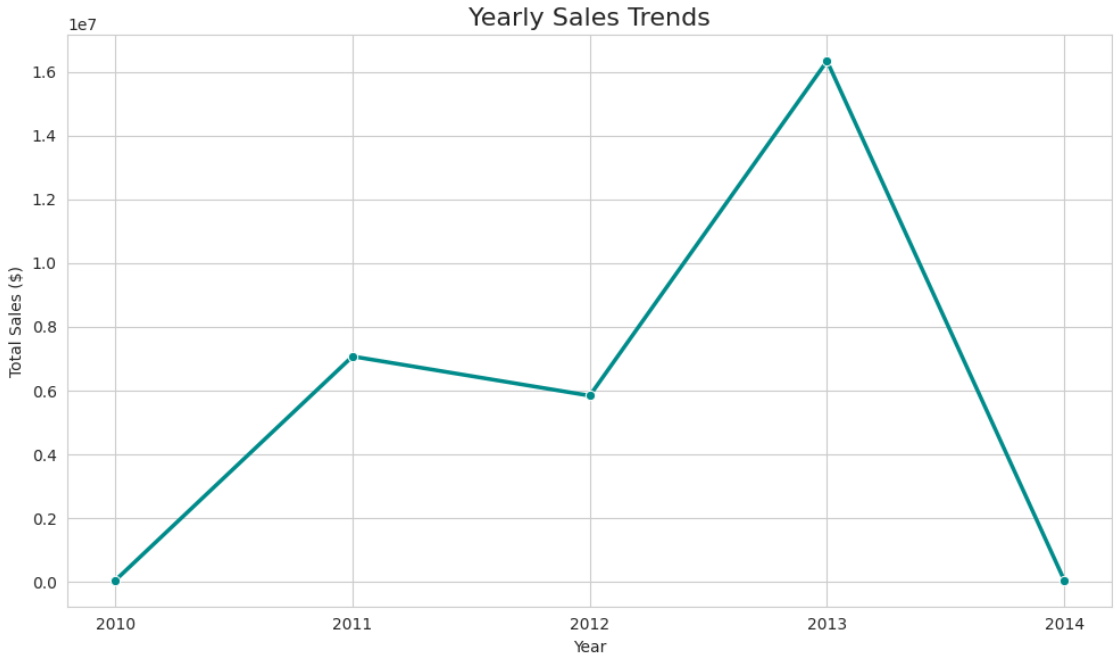
```
0        2010      43419.0               14            14.0
1        2011    7075088.0             2216          2216.0
2        2012    5842231.0             3255          3397.0
3        2013   16344878.0            17427         52807.0
4        2014      45642.0              834          1970.0
-------------------------------------------------
```

[ ]:
```
# ================================
# === Data Visualizations
# ================================
```

[2]:
```
if not df_monthly.empty and not df_yearly.empty:

    # --- 1. Sales Trends by Month ---
    plt.figure(figsize=(12, 6))
    sns.lineplot(
        data=df_monthly,
        x='order_month',
        y='total_sales',
        marker='o',
        linewidth=2.5,
        color='coral'
    )
    plt.title('Monthly Sales Trends', fontsize=16)
    plt.xlabel('Month')
    plt.ylabel('Total Sales ($)')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

else:
    print("One or both DataFrames are empty. No visualizations will be␣
 ↪generated.")
```



[3]:
```
if not df_monthly.empty and not df_yearly.empty:

    # --- 2. Sales Trends by Year ---
    plt.figure(figsize=(10, 6))
    sns.lineplot(
        data=df_yearly,
        x='order_year',
        y='total_sales',
        marker='o',
        linewidth=2.5,
        color='darkcyan'
    )
    plt.title('Yearly Sales Trends', fontsize=16)
    plt.xlabel('Year')
    plt.ylabel('Total Sales ($)')
    plt.xticks(df_yearly['order_year'])
    plt.tight_layout()
    plt.show()

else:
    print("One or both DataFrames are empty. No visualizations will be␣
 ↪generated.")
```


Yearly Sales Trends
1e7
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Total Sales ($)
2010
2011
2012
2013
2014
Year