Retail Sales Analytics in Power BI – Python Visuals Lab Guide

# Objective

In this lab, you will explore a synthetic retail sales dataset containing 1,000 transaction records. You will learn how to use Python visuals in Power BI to analyze trends, identify sales patterns, and derive insights.

# Dataset: retail\_sales\_analytics.xlsx

This dataset includes transaction-level data for a retail chain across four regions. Each transaction includes details such as date, store, customer demographics, product category, pricing, discount, and payment method.

# Step-by-Step Setup

- Open Power BI Desktop

- Load 'retail\_sales\_analytics.xlsx' using Get Data > Excel

- Ensure Python is installed and configured under File > Options > Python scripting

## 1. Total Sales Over Time

This line plot shows the total daily sales trend.

Python Code:

import pandas as pd  
import matplotlib.pyplot as plt  
  
dataset['TransactionDate'] = pd.to\_datetime(dataset['TransactionDate'])  
daily\_sales = dataset.groupby(dataset['TransactionDate'].dt.date)['TotalPrice'].sum().reset\_index()  
  
plt.figure(figsize=(10, 5))  
plt.plot(daily\_sales['TransactionDate'], daily\_sales['TotalPrice'], marker='o')  
plt.title("Total Sales Over Time")  
plt.xlabel("Date")  
plt.ylabel("Total Sales")  
plt.grid(True)  
plt.tight\_layout()  
plt.show()

## 2. Sales by Region – Bar Plot

Compares total sales across the four regions.

Python Code:

import matplotlib.pyplot as plt  
  
region\_sales = dataset.groupby('Region')['TotalPrice'].sum().sort\_values()  
region\_sales.plot(kind='bar', color='skyblue')  
  
plt.title("Total Sales by Region")  
plt.xlabel("Region")  
plt.ylabel("Total Sales")  
plt.tight\_layout()  
plt.show()

## 3. Sales by Product Category – Pie Chart

Displays the proportion of sales by product category.

Python Code:

import matplotlib.pyplot as plt  
  
cat\_sales = dataset.groupby('ProductCategory')['TotalPrice'].sum()  
plt.figure(figsize=(6, 6))  
plt.pie(cat\_sales, labels=cat\_sales.index, autopct='%1.1f%%', startangle=140)  
plt.title("Sales Share by Product Category")  
plt.tight\_layout()  
plt.show()

## 4. Quantity vs. Discount – Scatter Plot

Visualizes how discount affects quantity sold.

Python Code:

import matplotlib.pyplot as plt  
  
plt.figure(figsize=(7, 5))  
plt.scatter(dataset['Discount'], dataset['Quantity'], alpha=0.5)  
plt.title("Discount vs Quantity Purchased")  
plt.xlabel("Discount")  
plt.ylabel("Quantity")  
plt.grid(True)  
plt.tight\_layout()  
plt.show()

## 5. Customer Age Distribution – Histogram

Shows distribution of customer ages.

Python Code:

import matplotlib.pyplot as plt  
  
plt.figure(figsize=(7, 5))  
plt.hist(dataset['CustomerAge'], bins=10, color='orange', edgecolor='black')  
plt.title("Customer Age Distribution")  
plt.xlabel("Age")  
plt.ylabel("Frequency")  
plt.tight\_layout()  
plt.show()

## 6. Sales by Payment Method – Horizontal Bar

Compares total sales by payment method.

Python Code:

import matplotlib.pyplot as plt  
  
pm\_sales = dataset.groupby('PaymentMethod')['TotalPrice'].sum().sort\_values()  
pm\_sales.plot(kind='barh', color='green')  
  
plt.title("Sales by Payment Method")  
plt.xlabel("Total Sales")  
plt.ylabel("Payment Method")  
plt.tight\_layout()  
plt.show()

## 7. Average Unit Price by Product Category – Boxplot

Compares price distributions within each product category.

Python Code:

import seaborn as sns  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(8, 5))  
sns.boxplot(x='ProductCategory', y='UnitPrice', data=dataset)  
plt.title("Unit Price by Product Category")  
plt.xlabel("Product Category")  
plt.ylabel("Unit Price")  
plt.tight\_layout()  
plt.show()

## 8. Sales by StoreID – Column Chart

Shows which store generates the most sales.

Python Code:

import matplotlib.pyplot as plt  
  
store\_sales = dataset.groupby('StoreID')['TotalPrice'].sum()  
store\_sales.plot(kind='bar', color='purple')  
  
plt.title("Total Sales by Store")  
plt.xlabel("Store ID")  
plt.ylabel("Total Sales")  
plt.tight\_layout()  
plt.show()

## 9. Correlation Heatmap

Analyzes correlation between numeric variables.

Python Code:

import seaborn as sns  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(6, 5))  
sns.heatmap(dataset.corr(), annot=True, cmap="coolwarm")  
plt.title("Correlation Heatmap")  
plt.tight\_layout()  
plt.show()

## 10. Age vs. Total Spend – Regression Line

Shows how customer age correlates with total spend.

Python Code:

import seaborn as sns  
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(7, 5))  
sns.regplot(x='CustomerAge', y='TotalPrice', data=dataset, scatter\_kws={'alpha':0.4})  
plt.title("Customer Age vs. Total Spend")  
plt.xlabel("Customer Age")  
plt.ylabel("Total Price")  
plt.tight\_layout()  
plt.show()