```python
# Step 1: Install Required Libraries
!pip install matplotlib seaborn pandas openpyxl reportlab

# Step 2: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import letter

# Step 3: Create Dummy Sales Data
np.random.seed(42)
data = {
    "Date": pd.date_range(start="2024-01-01", periods=100, freq="D"),
    "Region": np.random.choice(["North", "South", "East", "West"], 100),
    "Product": np.random.choice(["Laptop", "Mobile", "Tablet", "Headphones"], 100),
    "Sales_Rep": np.random.choice(["Alice", "Bob", "Charlie", "David"], 100),
    "Customer_Segment": np.random.choice(["Retail", "Corporate"], 100),
    "Revenue": np.random.randint(5000, 50000, 100)
}
df = pd.DataFrame(data)

# Preview Data
print(df.head())
```

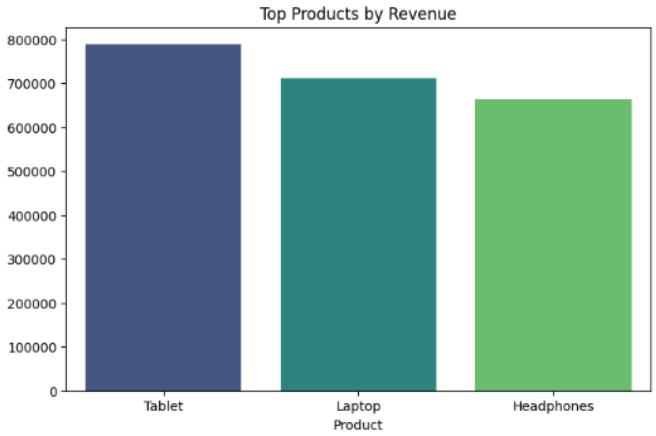
```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.12/dist-packages (3.1.5)
Collecting reportlab
  Downloading reportlab-4.4.4-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.59.2)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.12/dist-packages (from openpyxl) (2.0.0)
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.12/dist-packages (from reportlab) (3.4.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Downloading reportlab-4.4.4-py3-none-any.whl (2.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 22.1 MB/s eta 0:00:00
Installing collected packages: reportlab
Successfully installed reportlab-4.4.4
        Date Region     Product Sales_Rep Customer_Segment  Revenue
0 2024-01-01   East      Tablet   Charlie          Retail    17183
1 2024-01-02   West      Mobile     David          Retail    34299
2 2024-01-03  North      Mobile   Charlie          Retail    17874
3 2024-01-04   East  Headphones     Alice       Corporate    37711
4 2024-01-05   East      Mobile     David       Corporate    10539
```

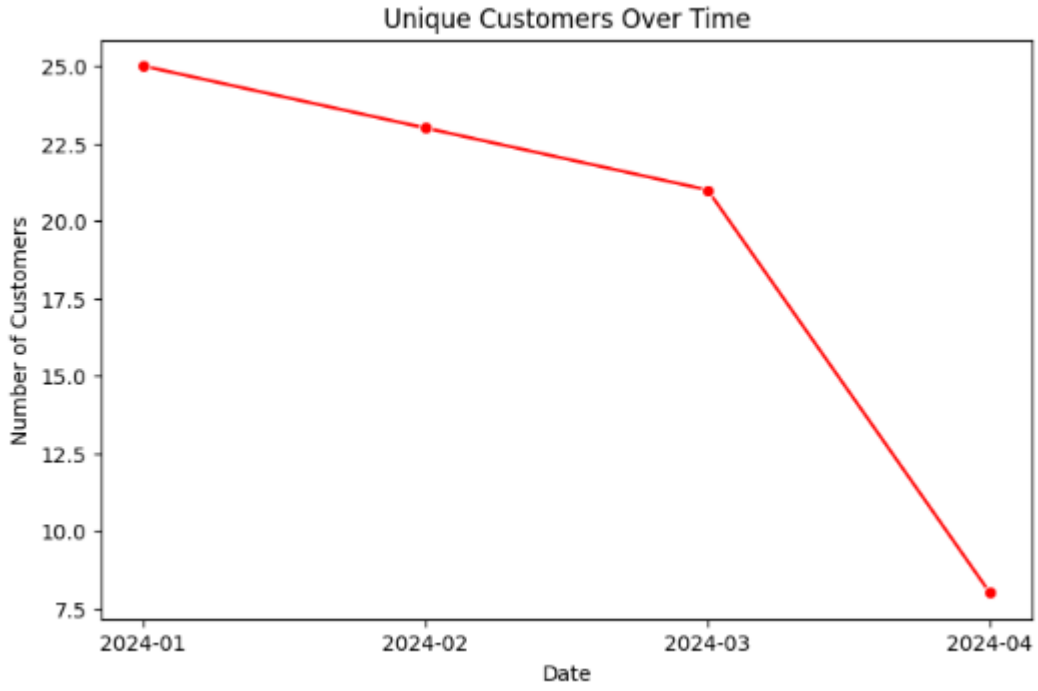
```python
# Step 4: KPIs
total_revenue = df["Revenue"].sum()
monthly_revenue = df.groupby(df["Date"].dt.to_period("M"))["Revenue"].sum()

# Revenue Growth (last month vs first month)
revenue_growth = ((monthly_revenue.iloc[-1] - monthly_revenue.iloc[0]) / monthly_revenue.iloc[0]) * 100

# Top Products
top_products = df.groupby("Product")["Revenue"].sum().sort_values(ascending=False).head(3)

# Customer Churn (simulated: unique customers decreasing over time)
df["Customer"] = np.random.randint(1, 50, size=len(df))  # fake customers
unique_customers = df.groupby(df["Date"].dt.to_period("M"))["Customer"].nunique()
churn_rate = ((unique_customers.iloc[0] - unique_customers.iloc[-1]) / unique_customers.iloc[0]) * 100

print("Total Revenue:", total_revenue)
print("Revenue Growth: {:.2f}%".format(revenue_growth))
print("Top Products:\n", top_products)
print("Customer Churn Rate: {:.2f}%".format(churn_rate))
```

```
Total Revenue: 2709716
Revenue Growth: -71.77%
Top Products:
 Product
Tablet        788645
Laptop        712321
Headphones    664729
Name: Revenue, dtype: int64
Customer Churn Rate: 68.00%
```

```python
# Step 5: Visualizations
plt.figure(figsize=(12,6))
sns.lineplot(x=monthly_revenue.index.astype(str), y=monthly_revenue.values, marker="o")
plt.title("Monthly Revenue Growth")
plt.xlabel("Month")
plt.ylabel("Revenue")
plt.show()

plt.figure(figsize=(8,5))
sns.barplot(x=top_products.index, y=top_products.values, palette="viridis")
plt.title("Top Products by Revenue")
plt.show()

plt.figure(figsize=(8,5))
sns.lineplot(x=unique_customers.index.astype(str), y=unique_customers.values, marker="o", color="red")
plt.title("Unique Customers Over Time")
plt.ylabel("Number of Customers")
plt.show()
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_products.index, y=top_products.values, palette="viridis")
```





```python
#Step 6: Export Results to Excel
excel_file = "sales_report.xlsx"
with pd.ExcelWriter(excel_file, engine="openpyxl") as writer:
    df.to_excel(writer, sheet_name="Raw Data", index=False)
    monthly_revenue.to_frame("Monthly Revenue").to_excel(writer, sheet_name="Revenue")
    top_products.to_frame("Top Products").to_excel(writer, sheet_name="Top Products")
    unique_customers.to_frame("Unique Customers").to_excel(writer, sheet_name="Customers")
print("Excel Report Saved:", excel_file)
```

```
Excel Report Saved: sales_report.xlsx
```

```python
# Step 7: Export Summary to PDF
pdf_file = "sales_report.pdf"
c = canvas.Canvas(pdf_file, pagesize=letter)
c.setFont("Helvetica", 12)
c.drawString(100, 750, "Sales Performance Dashboard Report")
c.drawString(100, 720, f"Total Revenue: {total_revenue}")
c.drawString(100, 700, f"Revenue Growth: {revenue_growth:.2f}%")
c.drawString(100, 680, f"Customer Churn Rate: {churn_rate:.2f}%")

c.drawString(100, 650, "Top Products:")
y = 630
for prod, rev in top_products.items():
    c.drawString(120, y, f"{prod}: {rev}")
    y -= 20

c.save()
print("PDF Report Saved:", pdf_file)
```

```
PDF Report Saved: sales_report.pdf
```

```python
from google.colab import files

# Download Excel
files.download("sales_report.xlsx")

# Download PDF
files.download("sales_report.pdf")
```