

```

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Sample data
data = {
    'Region': ['West', 'Northeast', 'Southeast', 'South', 'Midwest'],
    'Total Sales ($)': [270000000, 170000000, 160000000, 145000000, 140000000]
}

df = pd.DataFrame(data)

# Set style
sns.set_style("whitegrid")

# Create the plot
plt.figure(figsize=(10, 6))
barplot = sns.barplot(
    x='Total Sales ($)',
    y='Region',
    data=df,
    palette=sns.color_palette("viridis", 5)
)

# Title and labels
plt.title('Total Sales by Region', fontsize=14)
plt.xlabel('Total Sales ($)', fontsize=12)
plt.ylabel('Region', fontsize=12)

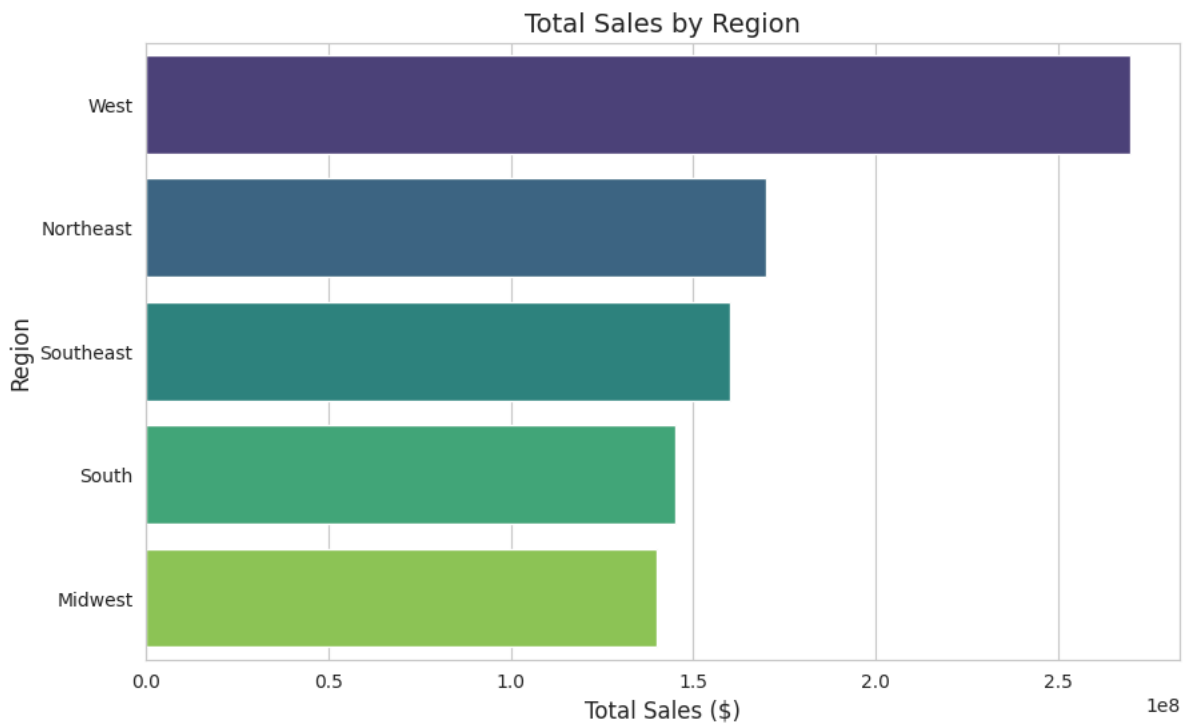
# Display the plot
plt.show()

```

 <ipython-input-24-ba25160f5f73>:18: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend`

```
barplot = sns.barplot(
```



```

import matplotlib.pyplot as plt

# Sample data
labels = ['In-store', 'Online', 'Outlet']
sizes = [20, 5, 25, 22, 8]

```

```

sizes = [33.0, 27.5, 32.8]
colors = ['#A7D3F5', '#FDBB86', '#9EE0B8'] # Light blue, light orange, light green

# Create the pie chart
plt.figure(figsize=(8, 8))
plt.pie(
    sizes,
    labels=labels,
    colors=colors,
    autopct='%1.1f%%',
    startangle=90,
    wedgeprops={'edgecolor': 'white'}
)

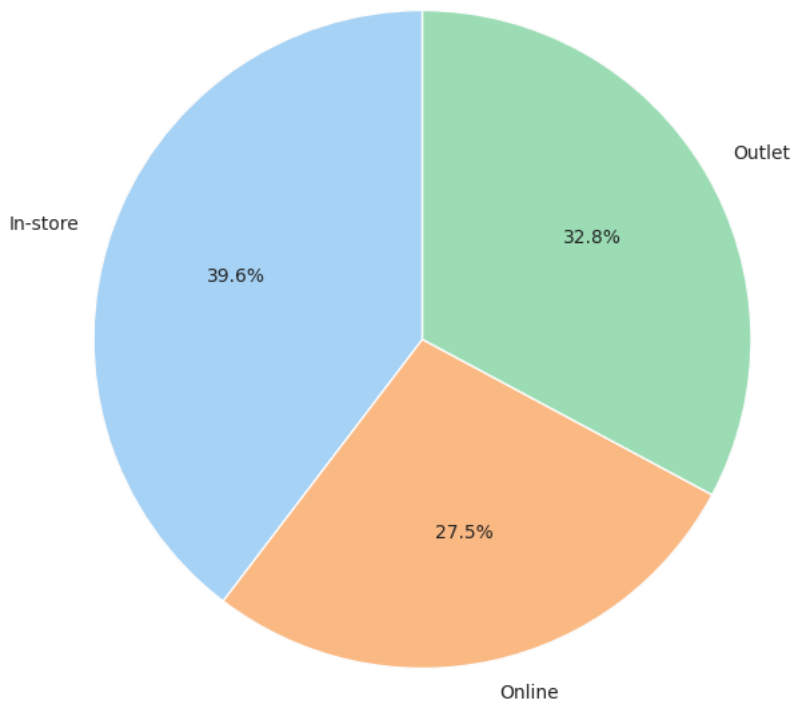
# Title
plt.title('Sales Distribution by Method', fontsize=14)

# Display the plot
plt.show()

```



Sales Distribution by Method



```

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Sample data
data = {
    'Product Category': ['Street Footwear', 'Athletic Footwear', 'Apparel'],
    'Total Units Sold': [980000, 750000, 720000]
}

df = pd.DataFrame(data)

# Set style
sns.set_style("whitegrid")

# Create the plot
plt.figure(figsize=(10, 6))
barplot = sns.barplot(

```

```

x='Total Units Sold',
y='Product Category',
data=df,
palette=['#4B1248', '#C5295D', '#E67E6F']
)

# Title and labels
plt.title('Units Sold by Product Category', fontsize=14)
plt.xlabel('Total Units Sold', fontsize=12)
plt.ylabel('Product Category', fontsize=12)

```

```

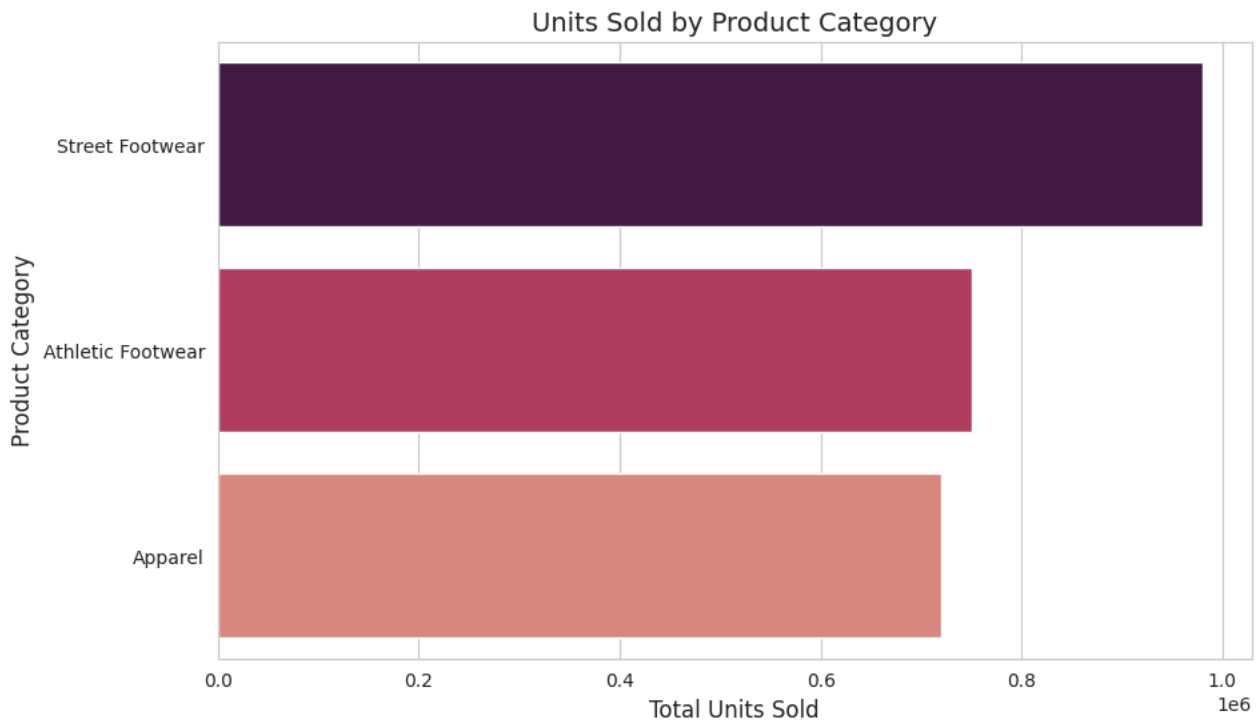
# Display the plot
plt.show()

```

 <ipython-input-23-28c1638b2d7d>:18: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend`

```
barplot = sns.barplot(
```



```

import pandas as pd

# Sample DataFrame
data = {
    'Total Sales': [68750, 101250, 213750, 56250, 61250, 20000, 68750, 101250, 222500, 75000],
    'Operating Profit': [24062.5, 30375, 117562.5, 19687.5, 24500, 8000, 24062.5, 30375, 122375, 26250]
}

df = pd.DataFrame(data)

# 1. Display the table
print("\n=== Sales and Operating Profit Data ===\n")
print(df)

# 2. Export to CSV
csv_file = '/content/drive/MyDrive/AdidasSalesdata.xlsx'
df.to_csv(csv_file, index=False)

print(f"\n✅ Data exported successfully to '{csv_file}'")

```

 === Sales and Operating Profit Data ===

	Total Sales	Operating Profit
0	68750	24062.5
1	101250	30375.0
2	213750	117562.5
3	56250	19687.5
4	61250	24500.0
5	20000	8000.0
6	68750	24062.5
7	101250	30375.0
8	222500	122375.0
9	75000	26250.0

✔ Data exported successfully to '/content/drive/MyDrive/AdidasSalesdata.xlsx'

```
import pandas as pd

# Sample DataFrame with all columns
data = {
    'Retailer': ['Retailer A', 'Retailer B', 'Retailer C'],
    'Retailer ID': [101, 102, 103],
    'Invoice Date': pd.to_datetime(['2024-01-01', '2024-01-02', '2024-01-03']),
    'Region': ['West', 'South', 'Northeast'],
    'State': ['California', 'Texas', 'New York'],
    'City': ['Los Angeles', 'Houston', 'New York City'],
    'Gender Type': ['Unisex', 'Male', 'Female'],
    'Product Category': ['Street Footwear', 'Apparel', 'Athletic Footwear'],
    'Price per Unit': [100, 50, 120],
    'Units Sold': [500, 300, 400],
    'Total Sales': [50000, 15000, 48000],
    'Operating Profit': [15000, 4500, 14000],
    'Operating Margin': [30.0, 30.0, 29.2],
    'Sales Method': ['Online', 'In-store', 'Outlet'],
    'Month_Year': pd.period_range('2024-01', periods=3, freq='M')
}

df = pd.DataFrame(data)

# 1. Display the DataFrame
print("\n=== Complete Sales Dataset ===\n")
print(df)

# 2. Export to CSV
csv_file = '/content/drive/MyDrive/AdidasSalesdata.xlsx'
df.to_csv(csv_file, index=False)

print(f"\n✔ Data exported successfully to '{csv_file}')
```

🔄

=== Complete Sales Dataset ===

	Retailer	Retailer ID	Invoice Date	Region	State	City \
0	Retailer A	101	2024-01-01	West	California	Los Angeles
1	Retailer B	102	2024-01-02	South	Texas	Houston
2	Retailer C	103	2024-01-03	Northeast	New York	New York City

	Gender Type	Product Category	Price per Unit	Units Sold	Total Sales \
0	Unisex	Street Footwear	100	500	50000
1	Male	Apparel	50	300	15000
2	Female	Athletic Footwear	120	400	48000

	Operating Profit	Operating Margin	Sales Method	Month_Year
0	15000	30.0	Online	2024-01
1	4500	30.0	In-store	2024-02
2	14000	29.2	Outlet	2024-03

✔ Data exported successfully to '/content/drive/MyDrive/AdidasSalesdata.xlsx'

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = {
    'Product Category': ['Street Footwear', 'Apparel', 'Athletic Footwear'],
    'Total Sales': [336829057, 302767492, 260305576]
}
```

```
df = pd.DataFrame(data)

# Create the plot
fig, ax = plt.subplots(figsize=(10, 6))

# Bars
bars = ax.barh(
    df['Product Category'],
    df['Total Sales'],
    color=['pink', 'green', 'skyblue'],
    edgecolor='black' # Black border for bars
)

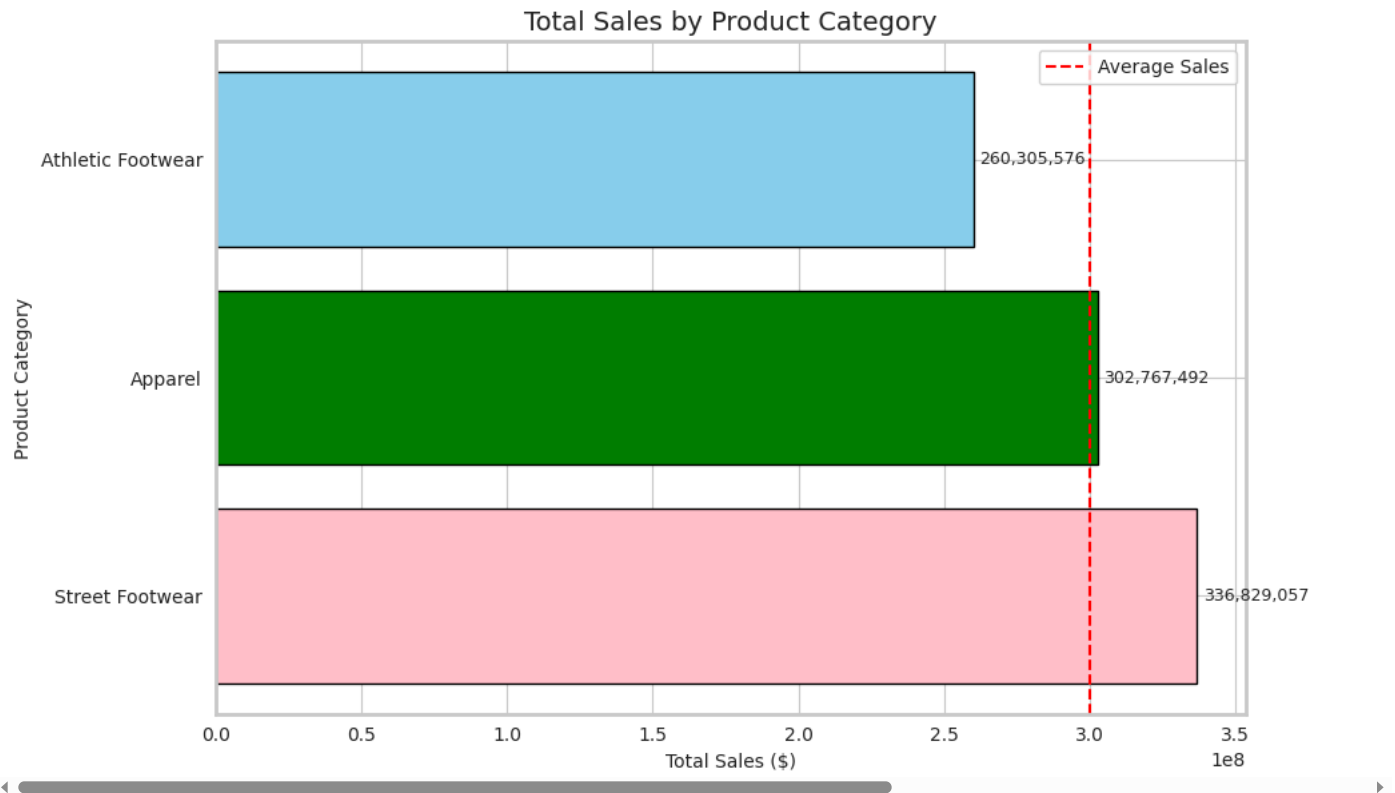
# Average sales line
average_sales = df['Total Sales'].mean()
ax.axvline(
    average_sales,
    color='red',
    linestyle='--',
    linewidth=1.5,
    label='Average Sales'
)

# Adding values at the end of bars
for bar in bars:
    width = bar.get_width()
    ax.text(
        width + 2e6,
        bar.get_y() + bar.get_height() / 2,
        f'{width:,.0f}',
        va='center',
        ha='left',
        fontsize=9
    )

# Titles, Labels
ax.set_title('Total Sales by Product Category', fontsize=14)
ax.set_xlabel('Total Sales ($)')
ax.set_ylabel('Product Category')
ax.legend()

# Tight layout and border
plt.tight_layout()
plt.gca().spines['top'].set_linewidth(2)
plt.gca().spines['right'].set_linewidth(2)
plt.gca().spines['left'].set_linewidth(2)
plt.gca().spines['bottom'].set_linewidth(2)

plt.show()
```



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Sample data (jaise tumhare graph me dikh raha hai)
data = {
    'Product Category': ['Apparel', 'Apparel', 'Street Footwear', 'Street Footwear', 'Athletic Footwear', 'Athletic Footwear'],
    'Gender Type': ['Men', 'Women', 'Men', 'Women', 'Men', 'Women'],
    'Total Sales': [80000, 110000, 130000, 80000, 95000, 70000]
}

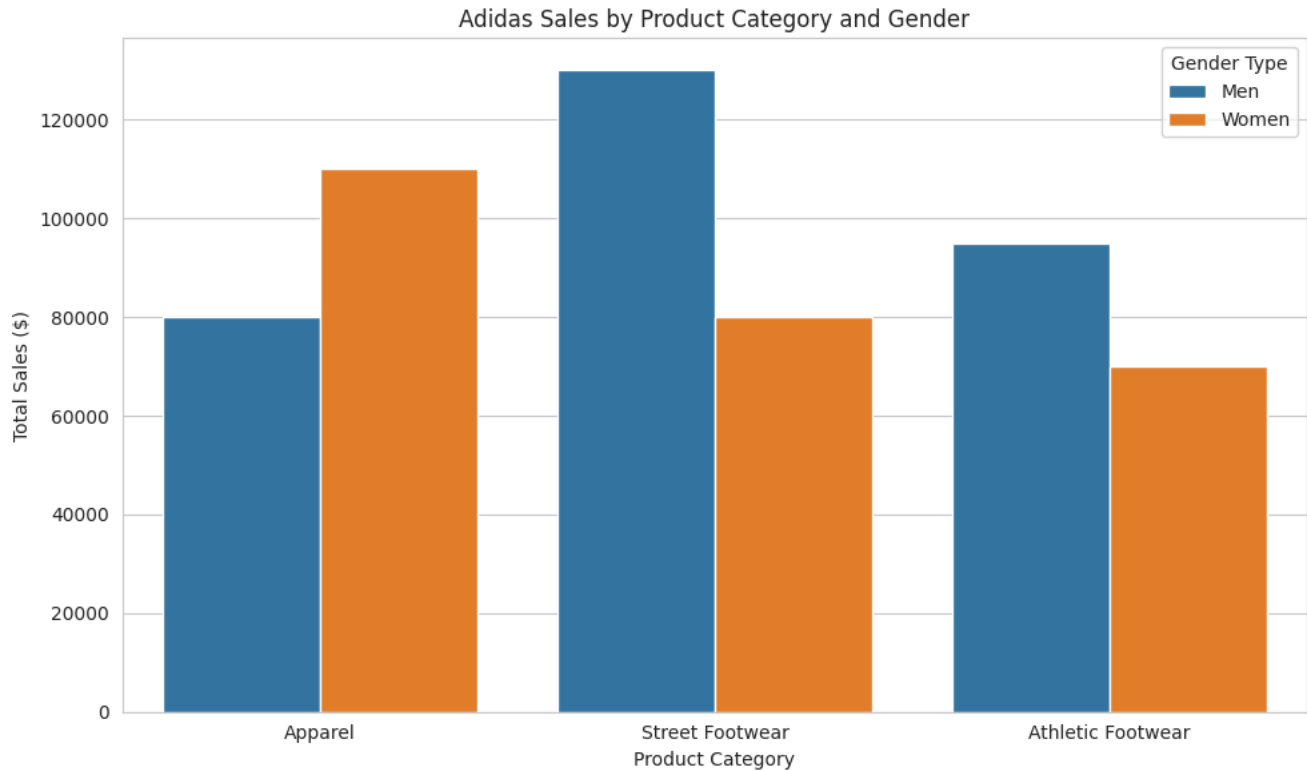
df = pd.DataFrame(data)

# Create the plot
plt.figure(figsize=(10, 6))
sns.barplot(
    data=df,
    x='Product Category',
    y='Total Sales',
    hue='Gender Type',
    errorbar='sd' # error bar automatically
)

# Titles and labels
plt.title('Adidas Sales by Product Category and Gender')
plt.xlabel('Product Category')
plt.ylabel('Total Sales ($)')

# Show legend
plt.legend(title='Gender Type')

# Show plot
plt.tight_layout()
plt.show()
```



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Tumhara original data load
# Example: df = pd.read_csv('/content/drive/MyDrive/AdidasSalesdata.xlsx')

# Dummy Date column banayenge
num_months = 300 # 25 saal ka approx data
df = df.sample(n=num_months, replace=True).reset_index(drop=True) # data ko sample karke bada kar liya
df['Date'] = pd.date_range(start='2000-01-01', periods=num_months, freq='M')

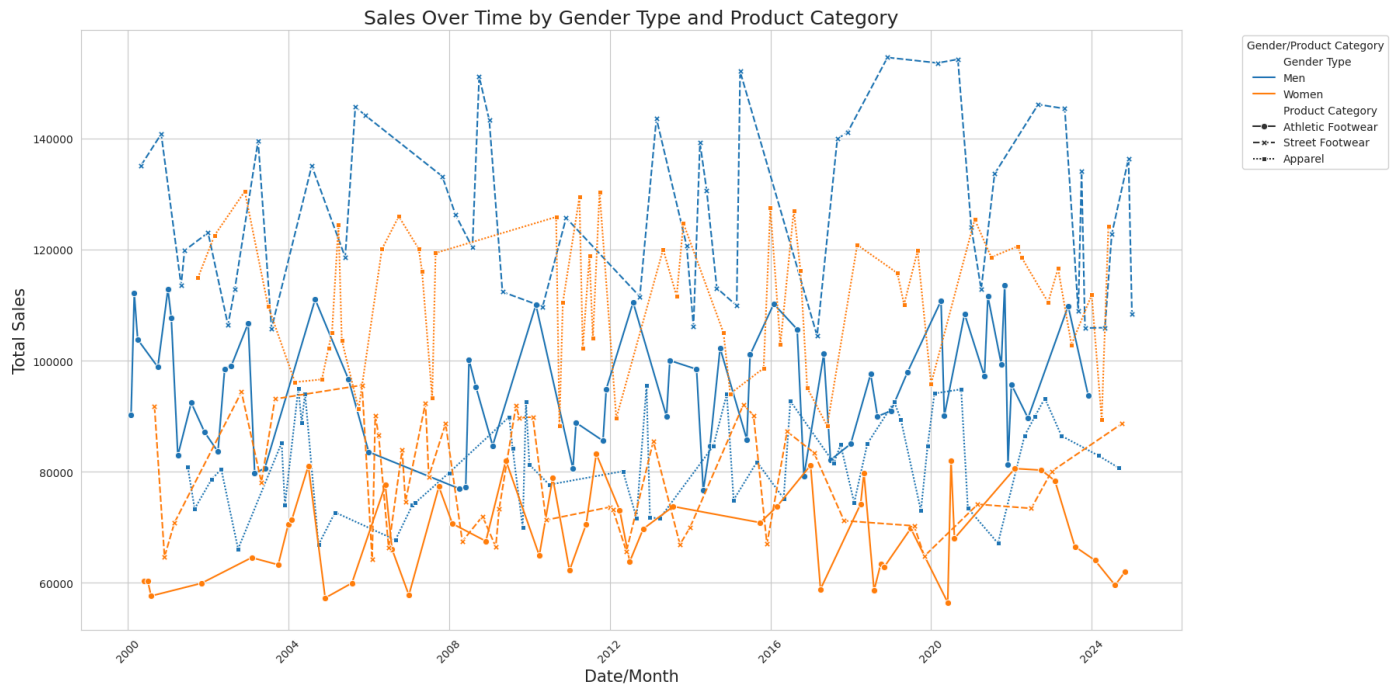
# Thoda random noise add karte hain Total Sales mein
np.random.seed(42)
df['Total Sales'] = df['Total Sales'] * (0.8 + 0.4 * np.random.rand(len(df)))

# Plot create karte hain
plt.figure(figsize=(18, 9))

sns.lineplot(
    data=df,
    x='Date',
    y='Total Sales',
    hue='Gender Type', # Gender Type ko color mein
    style='Product Category', # Product Category ko marker aur dash mein
    markers=True,
    dashes=True,
    palette='tab10'
)

plt.title('Sales Over Time by Gender Type and Product Category', fontsize=18)
plt.xlabel('Date/Month', fontsize=15)
plt.ylabel('Total Sales', fontsize=15)
plt.xticks(rotation=45)
plt.legend(title='Gender/Product Category', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
<ipython-input-36-1cb34b064226>:12: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.
df['Date'] = pd.date_range(start='2000-01-01', periods=num_months, freq='M')
```



```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt

# Step 1: USA States ka map load karte hain
states = gpd.read_file('https://raw.githubusercontent.com/PublicMundi/MappingAPI/master/data/geojson/us-states.json')

# Step 2: Dummy Sales Data banate hain (Tum yahan apna original data daal sakte ho)
data = {
    'State': ['California', 'Texas', 'Florida', 'New York', 'Illinois', 'Ohio', 'Georgia', 'North Carolina', 'Michigan', 'Pennsylvania'],
    'Total Sales': [1000000, 850000, 700000, 650000, 600000, 580000, 570000, 550000, 530000, 520000]
}
df_sales = pd.DataFrame(data)

# Step 3: Merge states map with sales data
merged = states.set_index('name').join(df_sales.set_index('State'))

# Step 4: Plotting
fig, ax = plt.subplots(1, 1, figsize=(16, 10))

merged.plot(
    column='Total Sales',
    cmap='viridis',          # Color map
    linewidth=0.8,
    ax=ax,
    edgecolor='0.8',
    legend=True,
    legend_kwds={'shrink': 0.5} # Legend size thoda chhota
)

ax.set_title('State-wise Performance Heatmap (Total Sales)', fontdict={'fontsize': 20})
ax.set_axis_off() # Axis hata diya clean look ke liye

plt.show()
```





# State-wise Performance Heatmap (Total Sales)

