

# Data Visualization Techniques Lab

Pulla Manoj Kumar

Assistant Professor, CSD, ACEEC Hyderabad

## Acquiring and Plotting Data

### Step 1: Import Required Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import plotly.graph_objects as go
```

```
warnings.filterwarnings('ignore')
```

### Step 2: Reading the Dataset

```
df = pd.read_excel('Sample - Superstore.xls')
df.head(2)
```

### Step 3: Understanding Dataset Structure

```
df.shape
df.info()
```

### Step 4: Check Missing Values

```
df.isnull().sum()
```

### Step 5: Handle Missing Values and Drop Unwanted Columns

```
df['Postal Code'] = df['Postal Code'].fillna(5408)
df = df.drop('Country', axis=1)
```

## Step 6: Segment-wise Sales – Pie Chart (Plotly)

```
fig = go.Figure(go.Pie(
    labels=df['Segment'],
    values=df['Sales'],
    customdata=df['Sales'],
    hovertemplate="Segment: %{label}<br>Sales: %{customdata}<extra></extra>",
    pull=[0, 0, 0, 0.1]
))

fig.update_layout(
    title={'text': "Sales in Different Segments", 'x': 0.5}
)

fig.show()
```

## Step 7: Count Plot for Sub-Category vs Category

```
plt.figure(figsize=(15,8))
sns.countplot(x='Sub-Category', hue='Category', data=df)
plt.title("Division Based on Categories in Sub-Categories")
plt.xticks(rotation=45)
plt.show()
```

## Step 8: Bar Chart for Ship Mode vs Sales

```
plt.figure(figsize=(10,8))
plt.bar(df['Ship Mode'], df['Sales'])
plt.xlabel("Ship Mode")
plt.ylabel("Sales")
plt.title("Sales Comparison Based on Shipping Type")
plt.xticks(rotation=30)
plt.show()
```

## Step 9: Region-wise Count Plot

```
plt.figure(figsize=(20,8))
sns.countplot(x='Region', data=df)
plt.title("Order Distribution Based on Regions")
plt.show()
```

## Step 10: Correlation Heatmap

```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='Reds')
plt.title("Correlation Heatmap")
plt.show()
```

## Step 11: Remove Irrelevant Columns

```
df.drop(['Row ID', 'Postal Code'], axis=1, inplace=True)
```

## Step 12: Facet Histogram Based on Shipping Mode

```
region_hist = sns.FacetGrid(df, col='Ship Mode')
region_hist.map(plt.hist, 'Region')
```

## Step 13: Create Derived Columns

```
df['Cost'] = df['Sales'] - df['Profit']
df['Profit %'] = (df['Profit'] / df['Cost']) * 100
df['Order Year'] = df['Order Date'].dt.year
```

## Step 14: Total Quantity Sold by Sub-Category

```
df.groupby('Sub-Category')['Quantity'].sum() \
    .sort_values(ascending=False) \
    .plot(kind='bar', figsize=(15,10), color='Cornflowerblue')

plt.title("Total Quantity Sold by Sub-Category")
plt.show()
```

## Step 15: Region-wise Average Sales and Profit

```
df_region = df.groupby('Region')[['Sales', 'Discount', 'Profit']].mean()

df_region.sort_values('Profit', ascending=False)[['Sales', 'Profit']] \
    .plot(kind='bar', figsize=(15,5),
          color=['Cornflowerblue', 'Teal'])

plt.title("Average Sales & Profit Across Regions")
plt.show()
```

## Step 16: Unit Price Comparison by Sub-Category

```
df['Unit Price'] = df['Sales'] / df['Quantity']
df['Unit Profit'] = df['Profit'] / df['Quantity']

df.groupby('Sub-Category')[['Unit Price']].mean() \
    .sort_values('Unit Price', ascending=False) \
    .plot(kind='bar', figsize=(15,5), color='Cornflowerblue')

plt.title("Unit Price Comparison by Sub-Category")
plt.show()
```

## Step 17: Profit vs Discount Line Plot

```
plt.figure(figsize=(10,5))
sns.lineplot(x='Discount', y='Profit', data=df)
plt.title("Profit vs Discount Level")
plt.show()
```

## Step 18: Discounted vs Non-Discounted Transactions

```
df_discounted = df[df['Discount'] > 0]
df_no_discount = df[df['Discount'] == 0]

print("On Average:")
print("Discounted Sales:", round(df_discounted['Sales'].mean(), 2))
print("Non-discounted Sales:", round(df_no_discount['Sales'].mean(), 2))

print("\nIn Total:")
print("Total Discounted Sales:", round(df_discounted['Sales'].sum(), 2))
print("Total Non-Discounted Sales:", round(df_no_discount['Sales'].sum(), 2))
```