

## Bar Chart(products)

```
import matplotlib.pyplot as plt
# Sample data
products = ['Product A', 'Product B', 'Product C', 'Product D', 'Product E']
sales = [150, 200, 180, 220, 170]
colors = ['#5DADE2', '#F1948A', '#82E0AA', '#F7DC6F', '#AF7AC5']
# Plotting the bar chart
plt.figure(figsize=(10, 6))
bars = plt.bar(products, sales, color=colors, edgecolor='black')
# Adding title and labels
plt.title('Sales of Different Products', fontsize=16, fontweight='bold')
plt.xlabel('Products', fontsize=12)
plt.ylabel('Sales (in units)', fontsize=12)
# Adding a legend
plt.legend(['Sales Volume'], loc='upper right')
# Adding data labels on top of each bar
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 5, int(yval), ha='center', va='bottom', fontsize=10, fontweight='bold')
# Displaying the chart
plt.show()
```

## Pie chart (products)

```
import matplotlib.pyplot as plt
# Sample data
products = ['Product A', 'Product B', 'Product C', 'Product D', 'Product E']
sales = [150, 200, 180, 220, 170]
colors = ['#5DADE2', '#F1948A', '#82E0AA', '#F7DC6F', '#AF7AC5']
# Explode the highest sales product for emphasis
explode = [0, 0, 0, 0.1, 0] # Only "explode" the slice for 'Product D'
# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(sales, labels=products, autopct='%1.1f%%', startangle=140, colors=colors, explode=explode,
        wedgeprops={'edgecolor': 'black'}, textprops={'fontsize': 12, 'fontweight': 'bold'})
# Adding a title
plt.title('Sales Distribution of Different Products', fontsize=16, fontweight='bold')
# Adding a legend
plt.legend(products, title="Products", loc="best", bbox_to_anchor=(1, 0, 0.5, 1))
# Display the chart
plt.tight_layout() # Adjust layout to make room for the legend
plt.show()
```

## Line Chart (Weather Dataset)

```
import matplotlib.pyplot as plt
import pandas as pd
# Sample data (date range with temperature and humidity values)
dates = pd.date_range(start="2023-01-01", end="2023-01-10")
temperature = [30, 32, 33, 29, 28, 31, 35, 34, 33, 32]
humidity = [45, 50, 55, 47, 52, 60, 65, 62, 58, 55]
# Creating the plot
plt.figure(figsize=(10, 6))
# Plotting temperature line
plt.plot(dates, temperature, marker='o', color='tomato', linestyle='-', linewidth=2, label='Temperature (°C)')
# Plotting humidity line
plt.plot(dates, humidity, marker='s', color='royalblue', linestyle='--', linewidth=2, label='Humidity (%)')
# Adding title and labels
plt.title('Weather Data: Temperature and Humidity Over Time', fontsize=16, fontweight='bold')
plt.xlabel('Date', fontsize=12)
plt.ylabel('Values', fontsize=12)
# Adding a legend
plt.legend(loc='upper left')
# Adding grid for readability
plt.grid(color='gray', linestyle=':', linewidth=0.5)
# Formatting the x-axis for better date display
plt.xticks(rotation=45)
# Displaying the plot
plt.tight_layout() # Adjust layout for title, labels, and grid
plt.show()
```

## Scatter plot (Employee Dataset)

```
import matplotlib.pyplot as plt
# Sample data
experience = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
salary = [35000, 42000, 50000, 52000, 60000, 65000, 70000, 78000, 82000, 90000]
departments = ['HR', 'Finance', 'IT', 'IT', 'HR', 'Finance', 'HR', 'IT', 'Finance', 'IT']
sizes = [70, 80, 90, 100, 110, 120, 130, 140, 150, 160] # Represents another variable (e.g., age)
# Define colors for each department
colors = {'HR': 'skyblue', 'Finance': 'orange', 'IT': 'green'}
color_values = [colors[dept] for dept in departments]
# Creating the scatter plot
plt.figure(figsize=(10, 6))
scatter = plt.scatter(experience, salary, c=color_values, s=sizes, alpha=0.7, edgecolor='black')
# Adding title and labels
plt.title('Employee Experience vs. Salary', fontsize=16, fontweight='bold')
plt.xlabel('Years of Experience', fontsize=12)
plt.ylabel('Salary ($)', fontsize=12)
# Adding grid for readability
plt.grid(color='gray', linestyle=':', linewidth=0.5)
# Adding a legend for department colors
for dept in colors:
    plt.scatter([], [], color=colors[dept], edgecolor='black', alpha=0.7, s=100, label=dept)
plt.legend(title="Department", loc="upper left")
# Displaying the plot
plt.tight_layout() # Adjust layout for better spacing
plt.show()
```

## Histograms (weather dataset)

```
import matplotlib.pyplot as plt
import numpy as np
# Sample data (simulated temperature and humidity values)
temperature = np.random.normal(25, 5, 100) # Mean temperature of 25°C with a std deviation of 5
humidity = np.random.normal(60, 10, 100) # Mean humidity of 60% with a std deviation of 10
# Creating the plot
plt.figure(figsize=(12, 6))
# Plotting temperature histogram
plt.hist(temperature, bins=10, color='salmon', edgecolor='black', alpha=0.7, label='Temperature (°C)')
# Plotting humidity histogram
plt.hist(humidity, bins=10, color='skyblue', edgecolor='black', alpha=0.7, label='Humidity (%)')
# Adding title and labels
plt.title('Distribution of Temperature and Humidity', fontsize=16, fontweight='bold')
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
# Adding a legend
plt.legend(loc='upper right')
# Adding grid for readability
plt.grid(color='gray', linestyle=':', linewidth=0.5)
# Display the plot
plt.tight_layout() # Adjust layout for better spacing
plt.show()
```

## Heat map (weather dataset)

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# Sample data: Temperature readings (randomly generated for illustration)
np.random.seed(0)
days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
hours = [f'{hour}:00' for hour in range(0, 24)]
# Generating a random temperature dataset (in °C)
temperature_data = np.random.normal(20, 5, size=(7, 24))
# Creating a DataFrame for seaborn
temperature_df = pd.DataFrame(temperature_data, index=days, columns=hours)
# Plotting the heatmap
plt.figure(figsize=(14, 8))
sns.heatmap(temperature_df, cmap='YlOrRd', annot=True, fmt=".1f", cbar_kws={'label': 'Temperature (°C)'})
# Adding title and labels
plt.title('Temperature Heatmap Over Days and Hours', fontsize=18, fontweight='bold', pad=20)
plt.xlabel('Hour of Day', fontsize=14)
plt.ylabel('Day of Week', fontsize=14)
# Display the plot
plt.tight_layout()
plt.show()
```

## Box Plot (weather dataset)

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

# Sample data: Simulated temperature and humidity values for four seasons
np.random.seed(42)
data = {
    'Temperature': np.concatenate([
        np.random.normal(10, 3, 50), # Winter
        np.random.normal(15, 4, 50), # Spring
        np.random.normal(25, 5, 50), # Summer
        np.random.normal(18, 4, 50)  # Autumn
    ]),
    'Humidity': np.concatenate([
        np.random.normal(65, 10, 50), # Winter
        np.random.normal(55, 8, 50),  # Spring
        np.random.normal(50, 7, 50),  # Summer
        np.random.normal(60, 9, 50)   # Autumn
    ]),
    'Season': ['Winter'] * 50 + ['Spring'] * 50 + ['Summer'] * 50 + ['Autumn'] * 50
}

# Creating a DataFrame
weather_df = pd.DataFrame(data)

# Creating the plot
plt.figure(figsize=(14, 8))

# Plotting the box plot for Temperature
plt.subplot(1, 2, 1)
sns.boxplot(x='Season', y='Temperature', data=weather_df, palette='coolwarm')
plt.title('Temperature Distribution by Season', fontsize=14, fontweight='bold')
plt.xlabel('Season', fontsize=12)
plt.ylabel('Temperature (°C)', fontsize=12)

# Plotting the box plot for Humidity
plt.subplot(1, 2, 2)
sns.boxplot(x='Season', y='Humidity', data=weather_df, palette='Blues')
plt.title('Humidity Distribution by Season', fontsize=14, fontweight='bold')
plt.xlabel('Season', fontsize=12)
plt.ylabel('Humidity (%)', fontsize=12)

# Adjusting layout and displaying the plot
plt.tight_layout()
plt.show()
```