

In [1]:

```
import matplotlib
import seaborn
print("Matplotlib:", matplotlib.__version__)
print("Seaborn:", seaborn.__version__)
```

Matplotlib: 3.8.4  
Seaborn: 0.13.2

In [4]:

```
import matplotlib
import seaborn
print("Matplotlib:", matplotlib.__version__, "->", matplotlib.__file__)
print("Seaborn:", seaborn.__version__, "->", seaborn.__file__)
```

Matplotlib: 3.8.4 → C:\Users\Lenovo\AppData\Roaming\Python\Python39\site-packages\matplotlib\\_\_init\_\_.py  
Seaborn: 0.13.2 → C:\Users\Lenovo\AppData\Roaming\Python\Python39\site-packages\seaborn\\_\_init\_\_.py

In [1]:

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

In [ ]:

1) A retail company has provided a CSV file (`sales_data.csv`) containing sales transactions **with** the following columns: Date, Product, Quantity, Sales, **and** Region. As a Data Analyst, perform an Exploratory Data Analysis (EDA) using Python (Pandas, NumPy, Matplotlib, **and** Seaborn) **with** the following tasks:

1. Load the dataset into a Pandas DataFrame **and** display the first few records.
2. Identify **and** handle missing values (replace missing Sales values **with** the mean, **and** rows **with** missing Product, Quantity, **or** Region).
3. Generate summary statistics of the numerical columns.
4. Group the data by Product **and** compute the total Sales **and** Quantity sold.
5. Create a bar chart showing the total sales **for** each product.
6. Convert the Date column to datetime **and** plot a line chart of total sales over time.
7. Construct a pivot table to analyze sales by Region **and** Product.
8. Compute the correlation matrix of numerical variables **and** visualize it using a heatmap. Discuss the insights obtained **from** each step.

In [2]:

```
sample_data = """Date,Product,Quantity,Sales,Region
2025-01-05,Smartphone,10,5000,North
2025-01-06,Laptop,5,7500,South
2025-01-07,Tablet,8,3200,East
2025-01-08,Smartwatch,15,2250,West
2025-01-09,Headphones,20,2000,North
2025-01-10,Laptop,3,4500,East
2025-01-11,Smartphone,7,3500,South
2025-01-12,Tablet,10,4000,West
2025-01-13,Smartwatch,12,1800,North
```

```
2025-01-14,Headphones,25,2500,South  
2025-01-15,Laptop,6,,West  
2025-01-16,,10,5000,East  
2025-01-17,Smartphone,,4000,North  
2025-01-18,Tablet,9,3600,  
"""
```

```
with open("sales_data.csv", "w") as f:  
    f.write(sample_data)
```

```
In [3]:
```

```
import pandas as pd  
  
df = pd.read_csv('sales_data.csv')  
print(df.head())
```

```
      Date   Product  Quantity   Sales Region  
0  2025-01-05  Smartphone     10.0  5000.0  North  
1  2025-01-06       Laptop      5.0  7500.0  South  
2  2025-01-07      Tablet      8.0  3200.0   East  
3  2025-01-08  Smartwatch     15.0  2250.0  West  
4  2025-01-09  Headphones     20.0  2000.0  North
```

```
In [4]:
```

```
import pandas as pd  
  
df = pd.read_csv('sales_data.csv')  
  
print(df.isnull().sum())  
  
mean_sales = df['Sales'].mean()  
df['Sales'] = df['Sales'].fillna(mean_sales)  
  
df.dropna(subset=['Product', 'Quantity', 'Region'], inplace=True)  
  
print(df.isnull().sum())
```

```
Date      0  
Product    1  
Quantity   1  
Sales      1  
Region     1  
dtype: int64  
Date      0  
Product    0  
Quantity   0  
Sales      0  
Region     0  
dtype: int64
```

```
In [5]:
```

```
print(df.describe())
```

```
      Quantity      Sales  
count  11.000000  11.000000  
mean   11.000000  3637.062937  
std    6.678323  1647.794180  
min    3.000000  1800.000000  
25%   6.500000  2375.000000  
50%   10.000000  3500.000000
```

```
75%      13.500000  4250.000000
max      25.000000  7500.000000
```

```
In [6]: product_group = df.groupby('Product').agg({'Sales':'sum', 'Quantity':'sum'}).reset_index()
print(product_group)
```

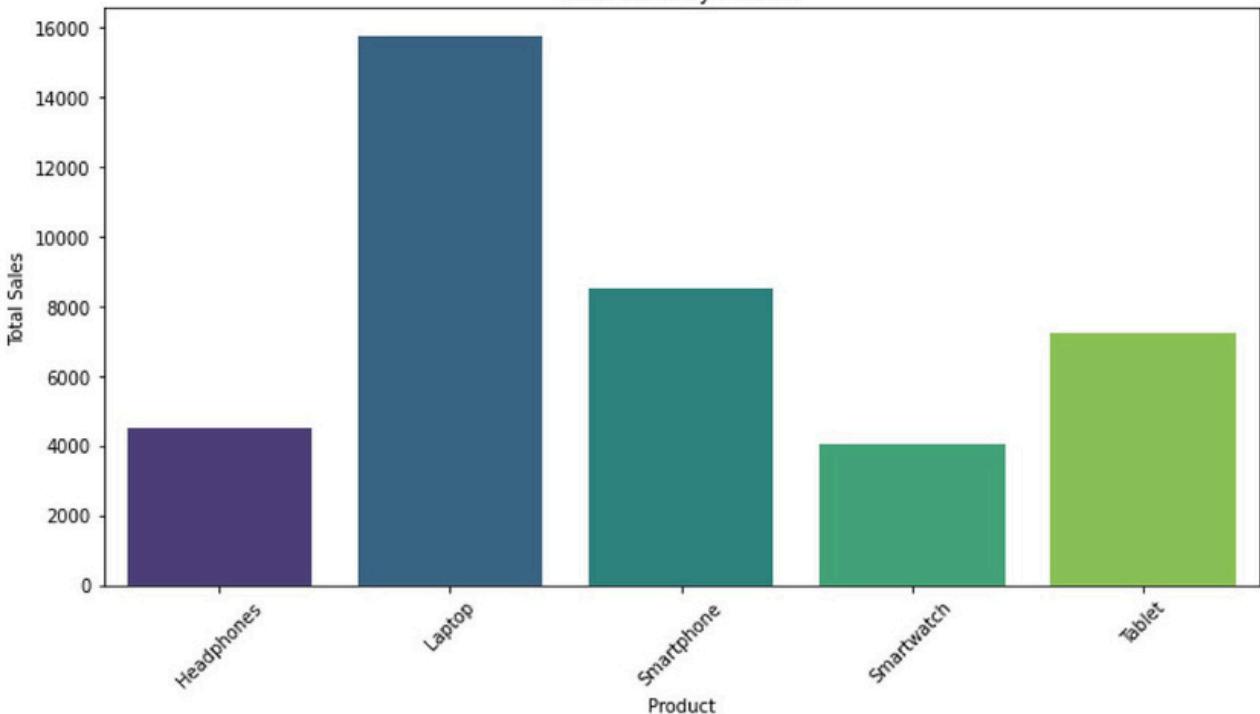
	Product	Sales	Quantity
0	Headphones	4500.000000	45.0
1	Laptop	15757.692308	14.0
2	Smartphone	8500.000000	17.0
3	Smartwatch	4050.000000	27.0
4	Tablet	7200.000000	18.0

```
In [7]: plt.figure(figsize=(10,6))
sns.barplot(data=product_group, x='Product', y='Sales', palette='viridis')
plt.title('Total Sales by Product')
plt.ylabel('Total Sales')
plt.xlabel('Product')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_14500/3120654797.py:2: FutureWarning:
```

```
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.
```

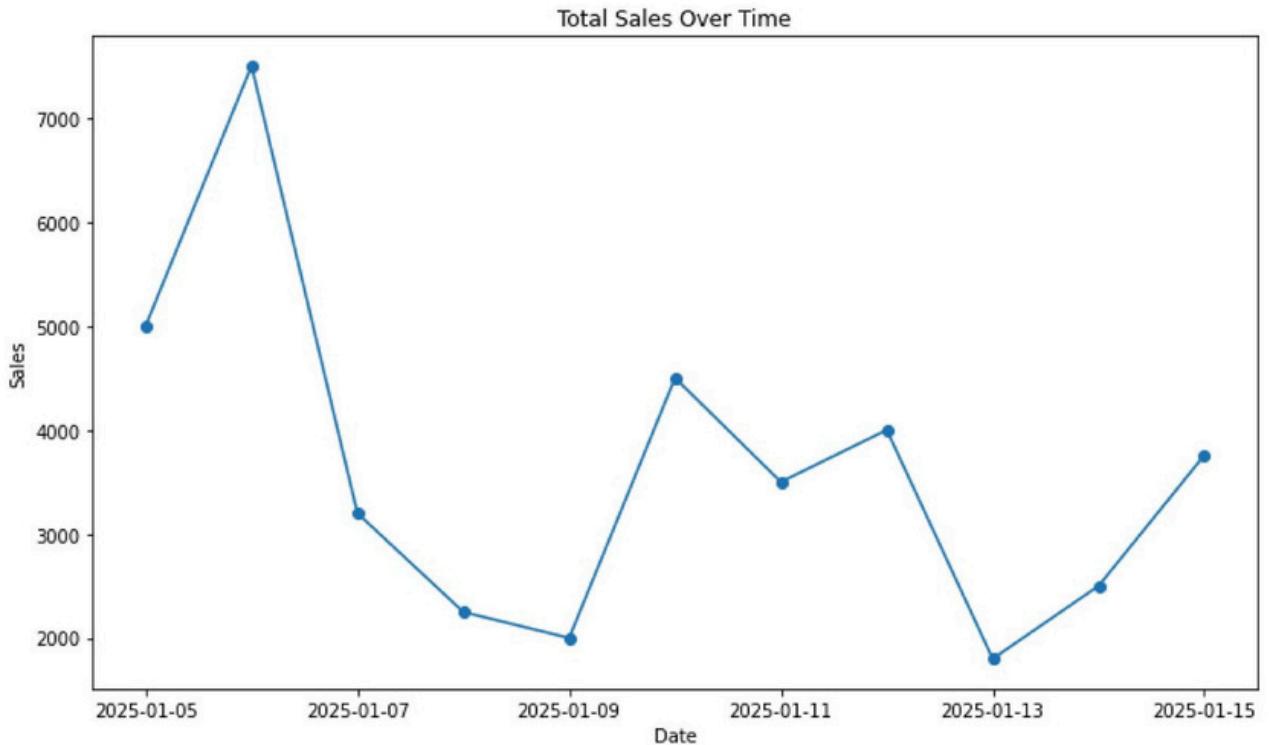
```
sns.barplot(data=product_group, x='Product', y='Sales', palette='viridis')
```



```
In [8]: df['Date'] = pd.to_datetime(df['Date'])
time_series = df.groupby('Date')[['Sales']].sum().reset_index()

plt.figure(figsize=(10,6))
plt.plot(time_series['Date'], time_series['Sales'], marker='o')
```

```
plt.title('Total Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.tight_layout()
plt.show()
```

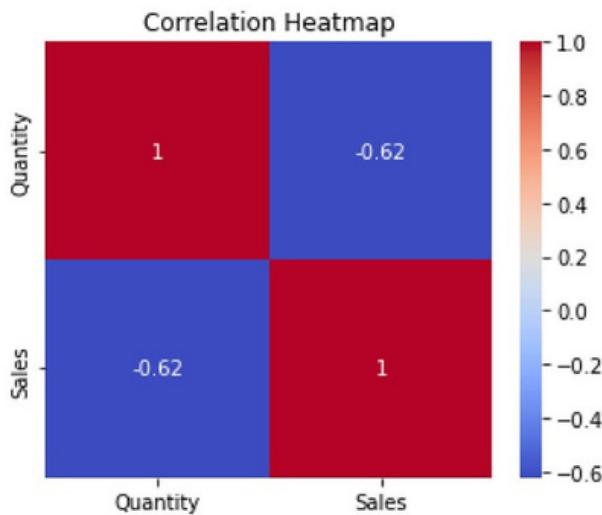


```
In [9]: pivot = pd.pivot_table(df, values='Sales', index='Region', columns='Product', aggfunc='sum')
print(pivot)
```

Region	Product	Headphones	Laptop	Smartphone	Smartwatch	Tablet
East		0.0	4500.000000	0.0	0.0	3200.0
North		2000.0	0.000000	5000.0	1800.0	0.0
South		2500.0	7500.000000	3500.0	0.0	0.0
West		0.0	3757.692308	0.0	2250.0	4000.0

```
In [10]: corr = df[['Quantity', 'Sales']].corr()

plt.figure(figsize=(5,4))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



In [ ]:

- 2) A dataset contains the names of six students (SHREE, DEV, KEERTHI, PRIYA, SHAN, KUMARAN) along with their Higher Secondary Certificate (HSC) exam percentages: 96, 91, 94, 75, 45, 81. Using Matplotlib and NumPy in Python:
    - 3) Plot a bar chart comparing the HSC percentages of the students.
    - 4) Set appropriate labels for the x-axis and y-axis.
    - 5) Rotate the x-axis tick labels for better readability.
    - 6) Add a title ("Comparison of HSC Percentage") with customized font size and color.
    - 7) Display the chart using show().
- Also, interpret the chart to identify the student with the highest and lowest percentage.

In [11]:

```
import numpy as np
import matplotlib.pyplot as plt

students = np.array(['SHREE', 'DEV', 'KEERTHI', 'PRIYA', 'SHAN', 'KUMARAN'])
percentages = np.array([96, 91, 94, 75, 45, 81])

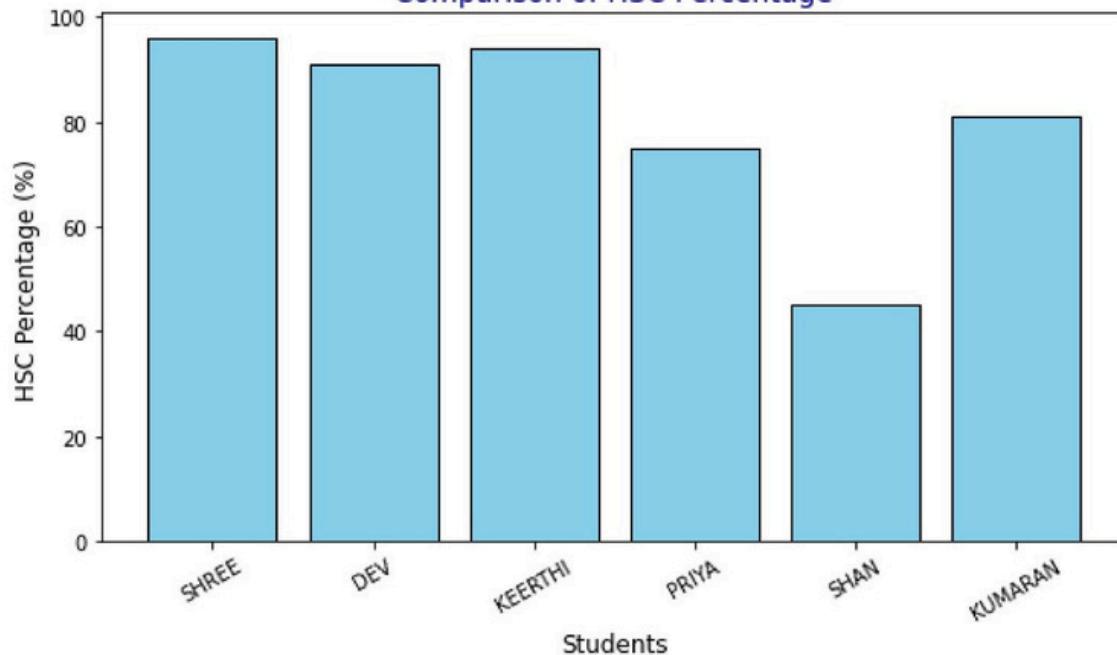
plt.figure(figsize=(8,5))
plt.bar(students, percentages, color='skyblue', edgecolor='black')

plt.xlabel('Students', fontsize=12)
plt.ylabel('HSC Percentage (%)', fontsize=12)
plt.title('Comparison of HSC Percentage', fontsize=14, color='darkblue')

plt.xticks(rotation=30)

plt.tight_layout()
plt.show()
```

Comparison of HSC Percentage



In [ ]:

3) The following table shows the number of votes received by four candidates **in** a colle

Candidate Votes

Candidate 1 315

Candidate 2 130

Candidate Votes

Candidate 3 245

Candidate 4 210

Using Matplotlib **in** Python, write a program to:

1. Plot a pie chart to represent the election results.
2. Highlight Candidate 1 using the explode parameter.
3. Use different colors **for** each candidate.
4. Display percentage values on the chart up to two decimal places.
5. Add a suitable title ("Election Results").

In [12]:

```
import matplotlib.pyplot as plt

candidates = ['Candidate 1', 'Candidate 2', 'Candidate 3', 'Candidate 4']
votes = [315, 130, 245, 210]

explode = [0.1, 0, 0, 0]

colors = ['gold', 'lightcoral', 'lightskyblue', 'yellowgreen']

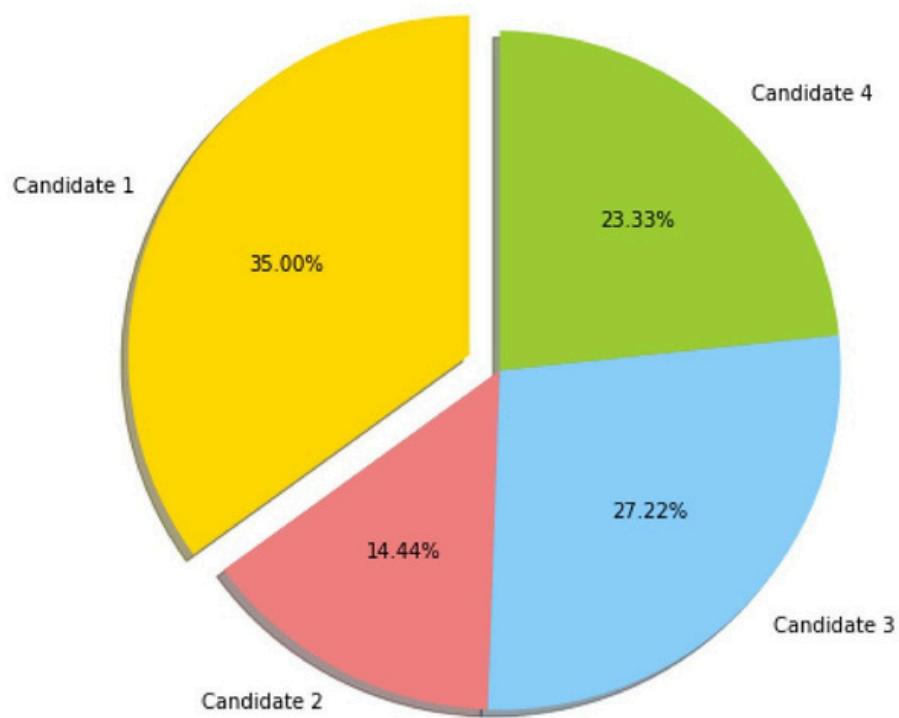
plt.figure(figsize=(7,7))
plt.pie(votes, labels=candidates, autopct='%1.2f%%', startangle=90,
        explode=explode, colors=colors, shadow=True)

plt.title('Election Results', fontsize=14, color='darkblue')

plt.axis('equal')

plt.show()
```

Election Results



In [ ]: