Fundamentals of Data Science – Lab Manual (Python Programs)

1. Analyze Student Performance Using NumPy

Scenario: Calculate the average score for each subject and identify the subject with the highest average score.

```python
import numpy as np

student_scores = np.array([[80, 85, 78, 90],
                           [88, 76, 92, 85],
                           [90, 91, 84, 89],
                           [75, 80, 79, 83]])

subject_names = ['Math', 'Science', 'English', 'History']
average_scores = student_scores.mean(axis=0)
highest_subject = subject_names[np.argmax(average_scores)]

print("Average scores per subject:", average_scores)
print("Subject with highest average:", highest_subject)
```

2. Average Product Price Analysis

Scenario: Find the average price of all products sold in the past month.

```python
sales_data = np.array([[100, 200, 150],
                       [250, 180, 220],
                       [300, 270, 290]])

average_price = sales_data.mean()
print("Average price:", average_price)
```

3. Filter Houses by Bedrooms and Average Price

Scenario: Find the average sale price of houses with more than four bedrooms.

```python
house_data = np.array([
    [5, 2000, 350000],
    [3, 1500, 250000],
    [6, 2500, 450000],
    [4, 1800, 300000]
])
```

```
avg_price = house_data[house_data[:, 0] > 4][:, 2].mean()
print("Average sale price of houses with > 4 bedrooms:", avg_price)
```

4. Total Sales and Growth

Scenario: Calculate total annual sales and the percentage increase from Q1 to Q4.

```
sales_data = np.array([10000, 15000, 20000, 25000])

total_sales = sales_data.sum()
percentage_increase = ((sales_data[3] - sales_data[0]) / sales_data[0]) * 100

print("Total sales:", total_sales)
print("Percentage increase from Q1 to Q4:", percentage_increase, "%")
```

5. Fuel Efficiency Improvement

Scenario: Compute average fuel efficiency and percentage improvement between models.

```
fuel_efficiency = np.array([22.5, 28.0, 35.2, 30.0, 26.8])

average_efficiency = fuel_efficiency.mean()
improvement = ((fuel_efficiency[2] - fuel_efficiency[0]) /
fuel_efficiency[0]) * 100

print("Average fuel efficiency:", average_efficiency)
print("Improvement between model 1 and 3:", improvement, "%")
```

6. Grocery Purchase Total Cost

Scenario: Calculate the total purchase cost including discount and tax.

```
item_prices = [100, 200, 150]
quantities = [2, 1, 3]
discount_rate = 10  # in %
tax_rate = 5  # in %

subtotal = sum(p * q for p, q in zip(item_prices, quantities))
discounted = subtotal * (1 - discount_rate / 100)
total = discounted * (1 + tax_rate / 100)

print("Total cost:", total)
```

7. Analyze Orders Using Pandas

Scenario: Use Pandas to analyze order data.

```
import pandas as pd

order_data = pd.DataFrame({
    'Customer ID': [1, 2, 1, 3, 2],
    'Order Date': pd.to_datetime(['2024-01-01', '2024-01-05', '2024-02-01',
'2024-02-15', '2024-03-01']),
    'Product Name': ['A', 'B', 'A', 'C', 'B'],
    'Order Quantity': [2, 1, 3, 1, 2]
})

print(order_data.groupby('Customer ID').size())
print(order_data.groupby('Product Name')['Order Quantity'].mean())
print("Earliest:", order_data['Order Date'].min())
print("Latest:", order_data['Order Date'].max())
```

8. Top 5 Sold Products

Scenario: Find top 5 most sold products.

```
top_products = order_data.groupby('Product Name')['Order
Quantity'].sum().sort_values(ascending=False).head(5)
print(top_products)
```

9. Property Data Insights

Scenario: Analyze properties by location, bedrooms, and area.

```
property_data = pd.DataFrame({
    'Property ID': [1, 2, 3, 4],
    'Location': ['CityA', 'CityB', 'CityA', 'CityB'],
    'Bedrooms': [3, 5, 4, 6],
    'Area': [1200, 2500, 1800, 3000],
    'Listing Price': [300000, 450000, 350000, 500000]
})

print(property_data.groupby('Location')['Listing Price'].mean())
print(len(property_data[property_data['Bedrooms'] > 4]))
print(property_data[property_data['Area'] == property_data['Area'].max()])
```

10. Sales Data Visualization

Scenario: Plot line and bar charts for monthly sales.

```python
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr']
sales = [10000, 15000, 20000, 18000]

plt.plot(months, sales)
plt.title('Monthly Sales - Line Plot')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.show()

plt.bar(months, sales)
plt.title('Monthly Sales - Bar Plot')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.show()
```

11. Pie Chart for Student Preferences

Scenario: Visualize student course preferences using a pie chart.

```python
labels = ['Data Science', 'AI', 'Cybersecurity', 'Web Dev']
values = [40, 30, 20, 10]

plt.pie(values, labels=labels, autopct='%1.1f%%')
plt.title('Student Course Preferences')
plt.show()
```

12. Histogram of Exam Scores

Scenario: Visualize the distribution of student exam scores.

```python
scores = [45, 70, 88, 90, 67, 55, 80, 77, 85, 60]

plt.hist(scores, bins=5, edgecolor='black')
plt.title('Exam Score Distribution')
plt.xlabel('Score Range')
plt.ylabel('Number of Students')
plt.show()
```

13. Box Plot of Monthly Incomes

Scenario: Use a box plot to show income distribution.

```
monthly_incomes = [35000, 40000, 37000, 42000, 45000, 39000, 43000, 41000,
47000, 49000]

plt.boxplot(monthly_incomes)
plt.title('Monthly Income Distribution')
plt.ylabel('Income')
plt.show()
```

14. Scatter Plot of Study Hours vs Scores

Scenario: Display correlation between hours studied and exam scores.

```
study_hours = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
scores = [50, 55, 60, 65, 70, 75, 80, 85, 90, 95]

plt.scatter(study_hours, scores)
plt.title('Study Hours vs Scores')
plt.xlabel('Study Hours')
plt.ylabel('Score')
plt.show()
```

15. Compare Products Using Bar Chart

Scenario: Compare product sales across categories using grouped bar chart.

```
products = ['Product A', 'Product B', 'Product C']
sales_2023 = [100, 150, 200]
sales_2024 = [120, 170, 220]

x = np.arange(len(products))
width = 0.3

plt.bar(x - width/2, sales_2023, width, label='2023')
plt.bar(x + width/2, sales_2024, width, label='2024')

plt.xticks(x, products)
plt.ylabel('Sales')
plt.title('Product Sales Comparison')
plt.legend()
plt.show()
```

(Scenarios 16–23 will continue if you'd like to add them next.)