**The individual student work 4**

**NumPy**

1. **Creating Arrays**: Write a program to create a one-dimensional array with numbers from 0 to 9 and print it.
2. **Array Operations**: Given two arrays, A and B, perform element-wise addition, subtraction, multiplication, and division.
3. **Reshaping Arrays**: Create a one-dimensional array of 12 elements and reshape it into a 3x4 two-dimensional array.
4. **Statistical Operations**: Generate an array of 15 random integers between 1 and 100. Calculate and print the mean, median, and standard deviation.
5. **Element-wise Comparison**: Write a program to compare two arrays, element by element, and print a new array showing which elements are greater in the first array.
6. **Filtering Elements**: Create an array of 20 random integers between 1 and 50. Filter out and print only the elements that are greater than 25.
7. **Slicing**: Create a 4x4 matrix of numbers from 0 to 15. Print the submatrix that includes the elements from rows 2 and 3 and columns 1 and 2.
8. **Diagonal Elements**: Create a 5x5 identity matrix and print the main diagonal elements.
9. **Matrix Multiplication**: Write a program to create two 3x3 matrices and perform matrix multiplication on them.
10. **Array Broadcasting**: Create a one-dimensional array and add 5 to each element using broadcasting.

**Pandas**

1. **Reading a CSV File**: Write a program to load a CSV file into a DataFrame and display the first 5 rows.
2. **Filtering Data**: Given a DataFrame, filter out rows where the values in a specified column are below a given threshold.
3. **Data Aggregation**: Load a CSV file and calculate the mean, median, and sum for each column.
4. **Group By Operation**: Given a DataFrame with columns Category and Sales, group the data by Category and calculate the total sales for each group.
5. **Sorting Data**: Write a program to sort a DataFrame by a specific column in descending order and print the result.
6. **Handling Missing Data**: Load a CSV file with missing values. Fill the missing values with the mean of their respective columns.
7. **Adding and Removing Columns**: Write a program that adds a new column to a DataFrame by performing a calculation on existing columns, and then removes another column from the DataFrame.
8. **Merging DataFrames**: Create two DataFrames with a common key column and merge them on this key column.
9. **Pivot Tables**: Given a DataFrame with columns Product, Region, and Sales, create a pivot table to show total sales for each product in each region.
10. **Date Filtering**: Load a CSV file with a date column, convert it to datetime format, and filter the rows for a specific date range.

**Matplotlib**

1. **Line Plot**: Create a line plot showing the trend of a variable over time. Add labels for the x-axis and y-axis, and a title.
2. **Bar Plot**: Given data on the monthly sales of a product, create a bar plot to visualize the sales.
3. **Histogram**: Generate a dataset of 1000 random numbers from a normal distribution. Plot a histogram to visualize the distribution of values.
4. **Scatter Plot**: Generate two arrays of random numbers and create a scatter plot to visualize their relationship.
5. **Subplots**: Create a 2x2 grid of subplots with different types of plots (e.g., line plot, bar plot, scatter plot, and histogram).
6. **Pie Chart**: Given a list of categories and their respective counts, create a pie chart to show the percentage of each category.
7. **Box Plot**: Create a box plot for a dataset of random numbers to visualize the spread and identify any outliers.
8. **Heatmap**: Generate a 5x5 matrix of random values and create a heatmap to visualize the values.
9. **Customizing Plot Styles**: Create a line plot and customize the style by changing the color, line type, and marker.
10. **Adding Annotations**: Create a plot and add annotations to mark key points, such as the maximum or minimum values.