**Practical 1**

**Dataset Loading and Exploration**

1. **Load the dataset** into a pandas DataFrame and display the first 5 rows. Provide a summary of the dataset, including the number of rows, columns, and the data types of each column.
2. **Identify missing data**: Which columns contain missing values, and how many missing values are in each column?

**Handling Missing Data**

1. **Drop rows with missing values**: What is the shape of the dataset after dropping all rows with missing values? How does this compare to the original dataset's shape?
2. **Impute missing values**: Replace missing numerical values with the mean of the column and missing categorical values with the most frequent value (mode). How does the imputed dataset compare to the original in terms of completeness?

**Encoding Categorical Variables**

1. **One-Hot Encoding**: Apply one-hot encoding to the categorical variables in the dataset. How many new columns are created as a result of this encoding?
2. **Label Encoding**: Apply label encoding to a selected categorical variable. How does the encoded column compare to the original categorical column in terms of data type and values?

**Practical 2**

**Dataset Loading and Exploration**

1. **Load the dataset** into a pandas DataFrame and display the first 5 rows. What are the dimensions of the dataset (i.e., the number of rows and columns)?
2. **Descriptive Statistics**: Generate and interpret the summary statistics (mean, median, mode, standard deviation, etc.) for the numerical variables in the dataset.

**Visualizations**

1. **Histograms**: Create histograms for at least three numerical variables in the dataset. What do these histograms tell you about the distribution of each variable (e.g., skewness, presence of outliers)?
2. **Box Plots**: Create box plots for the same three numerical variables. How do the box plots help you identify outliers and understand the spread of the data?
3. **Scatter Plots**: Select two pairs of numerical variables and create scatter plots for each pair. What relationships (e.g., correlation, patterns) can you observe between these variables?

**Advanced Visualization**

1. **Pair Plot**: Create a pair plot (scatterplot matrix) for a subset of the numerical variables. What insights can you draw about the relationships between these variables?
2. **Correlation Matrix and Heatmap**: Compute the correlation matrix for the numerical variables and visualize it using a heatmap. Which variables have the strongest correlations, and what does this imply?

**Practical 3**

**Create a Power BI dashboard that provides insights into a dataset of your choice (e.g., Sales Data, Customer Data, Financial Data, etc.). Your dashboard should include the following interactive visualizations:**

1. **Overview Page:**
   * A **summary table** displaying key metrics (e.g., total sales, total customers, total profit).
   * A **card visualization** for highlighting key KPIs (Key Performance Indicators).
   * A **pie chart** or **donut chart** showing the distribution of a categorical variable (e.g., sales by region, product category, or customer segment).
2. **Trend Analysis Page:**
   * A **line chart** or **area chart** showing the trend of a numerical variable over time (e.g., monthly sales, yearly revenue).
   * A **bar chart** comparing the performance of different categories (e.g., sales by product, profit by region).
   * Use **slicers** to filter the data by relevant dimensions (e.g., time period, region, product category).

**Practical 4**

**Part 1: Data Visualization Using Matplotlib in Python**

1. **Load a Dataset:**
   * Load a dataset of your choice (e.g., sales data, customer data, financial data) into a pandas DataFrame in Python.
   * Display the first few rows of the dataset to understand its structure.
2. **Create the Following Visualizations Using Matplotlib:**
   * **Line Chart:** Plot a line chart to visualize a trend over time (e.g., sales over months, revenue over years). Customize the chart with labels, title, and grid lines.
   * **Bar Chart:** Create a bar chart to compare categorical data (e.g., sales by region, profit by product category). Customize the bars with different colors and add a legend.
   * **Pie Chart:** Generate a pie chart to show the distribution of a categorical variable (e.g., market share by company, customer segment distribution). Customize the chart with explode segments and a title.
   * **Scatter Plot:** Create a scatter plot to analyze the relationship between two numerical variables (e.g., sales vs. profit, customer age vs. purchase amount). Customize the plot with different markers and colors based on categories.
3. **Save Your Visualizations:**
   * Save each of the visualizations as PNG files.
4. **Code Submission:**
   * Submit the Python code used to generate these visualizations.

**Part 2: Data Visualization Using Excel**

1. **Load the Same Dataset into Excel:**
   * Import the same dataset into an Excel spreadsheet.
2. **Create the Following Visualizations Using Excel:**
   * **Line Chart:** Recreate the line chart using Excel’s chart tools. Customize the chart similarly to your Matplotlib version (e.g., labels, title, grid lines).
   * **Bar Chart:** Create a bar chart in Excel. Apply similar customizations as in your Python version (e.g., bar colors, legend).
   * **Pie Chart:** Generate a pie chart in Excel and apply similar customizations (e.g., explode segments, title).
   * **Scatter Plot:** Create a scatter plot in Excel. Customize it similarly to your Python version (e.g., markers, colors based on categories).
3. **Save Your Visualizations:**
   * Save each of the visualizations within the Excel file.
4. **Excel File Submission:**
   * Submit the Excel file containing your visualizations.

**Part 3: Comparative Analysis**

1. **Comparative Report:**
   * Write a comparative report (300-500 words) discussing the similarities and differences between the visualizations created in Python using Matplotlib and those created in Excel.
   * Consider the following in your comparison:
     + Ease of use and customization options.
     + Visual appeal and clarity of the charts.
     + Flexibility and control over the appearance of the visualizations.
     + Any challenges you faced while creating and customizing the visualizations in both tools.