|| Jai Sri Gurudev |

Sri AdichunchanagiriShikshana Trust (R)

SJB INSTITUTE OF TECHNOLOGY

SJBIT

Question Bank

Module 2

Subject Name: Exploratory Data Analytics

Subject Code: 23CSE422

By

Faculty Name: Mrs. Shilpashree S

Designation: Assistant Professor

Semester: IV

Department of Computer Science & Engineering

Aca. Year: Even Sem /2024-25

Reference and Textbook Information

This document is designed as a supplementary study aid for the **"Exploratory Data Analytics - 23CSE422"** module, utilizing the prescribed textbook, "**Hands-On Exploratory Data Analysis with Python**" by **Suresh Kumar Mukhiya** and **Usman Ahmed**. The content presented here is based on the concepts and information provided in the textbook, with additional explanations, examples, and elaborations intended to enhance student understanding. This document is provided for educational purposes, in alignment with the intended use of the prescribed course textbook. This document is a study aid, based upon the prescribed text book, and is intended for the students of this course.

Bibliographic information

Title	Hands-On Exploratory Data Analysis with Python: Perform EDA Techniques to Understand, Summarize, and Investigate Your Data
Authors	Suresh Kumar Mukhiya, Usman Ahmed
Publisher	Packt Publishing, Limited, 2020
ISBN	1789537258, 9781789537253
Length	352 pages

Module 2 Question Bank with Scheme of Evaluation
Data Transformation: Merging database - style dataframes

Sl.No.	Question with Scheme of Evaluation	Marks
1.	List and explain the various data transformation techniques, highlighting their objectives.	10 Marks
SoE	**Evaluation Scheme**: ● Explanation of Data Deduplication (1 Mark) ● Explanation of Key Restructuring (1 Mark) ● Explanation of Data Cleansing (1 Mark) ● Explanation of Data Validation (1 Mark) ● Explanation of Format Revisioning (1 Mark) ● Explanation of Data Derivation (1 Mark) ● Explanation of Data Aggregation (1 Mark) ● Explanation of Data Integration (1 Mark) ● Explanation of Data Filtering (1 Mark) ● Explanation of Data Joining (1 Mark)	
2.	**How would you merge two pandas DataFrames using the `concat()` method? Demonstrate with an example where you combine data along rows and columns.**	10 Marks
SoE	**Evaluation Scheme:** **1. Introduction to `concat()` Method (2 Marks)** **○ Briefly explain the purpose of the `concat()` method (1 Mark).** **○ Mention the role of `axis` argument (0.5 Marks).** **○ Mention the `ignore_index` argument and its effect (0.5 Marks).** **2. Example: Concatenating Along Rows (3 Marks)** **○ Code example using `pd.concat([dataFrame1, dataFrame2], ignore_index=True)` to combine data along rows (2 Marks).** **○ Output explanation: Describe the output and explain how the rows from both DataFrames are combined (1 Mark).** **3. Example: Concatenating Along Columns (3 Marks)** **○ Code example using `pd.concat([dataFrame1, dataFrame2], axis=1)` to combine data along columns**	

	(2 Marks). ○ **Output explanation: Describe the output and explain how the columns from both DataFrames are combined (1 Mark).** **4. Explanation of Differences Between Row and Column Concatenation (2 Marks)** ○ **Explanation of what changes when `axis=0` (row-wise) and `axis=1` (column-wise) are used (2 Marks).**	
3.	**How can you merge DataFrames when the two DataFrames have different student IDs, where some students are in both courses, and some are only in one course? Explain the methods and demonstrate with examples.**	10 Marks
SoE	**Evaluation Scheme:** **1. Introduction to Merging and Key Concepts (2 Marks)** ○ **Mention the purpose of merging DataFrames based on common keys (1 Mark).** ○ **Define `StudentID` as the key column for merging (1 Mark).** **2. Merge Methodology (Inner Join) (2 Marks)** ○ **Explain the concept of an inner join and how it combines rows based on common values (1 Mark).** ○ **Provide an example where `merge()` is used with `how='inner'` (1 Mark).** **3. Merge Methodology (Outer Join) (2 Marks)** ○ **Explain the concept of an outer join and its usefulness when handling students who might only appear in one course (1 Mark).** ○ **Provide an example where `merge()` is used with `how='outer'` (1 Mark).** **4. Handling Missing Data (2 Marks)** ○ **Discuss what happens to students who are not in both courses and how missing data is handled in the merged DataFrame (2 Marks).** **5. Demonstration with Provided DataFrames (2 Marks)** ○ **Provide code examples for merging the two DataFrames using both inner and outer joins. Ensure the resulting DataFrames show the correct combined data, including missing values where applicable (2 Marks).** **6. Conclusion and Observations (2 Marks)** ○ **Summarize how merging works for both common and non-common student IDs (1 Mark).**	

	○ **Mention the importance of choosing the right type of join for the task (1 Mark).**	
4.	**Describe the difference between concatenating DataFrames along axis=0 and axis=1. (10 Marks)**	10 Marks
SoE	**Introduction to concatenation (2 Marks)** **Explanation of axis=0 concatenation (3 Marks)** **Explanation of axis=1 concatenation (3 Marks)** **Real-life example demonstrating both (1 Mark)** **Conclusion summarizing the differences (1 Mark)**	
5.	**Explain how the merge function works in pandas, with an example for each type of join (inner, left, right, and outer).**	10 Marks
SoE	**Introduction to merging (2 Marks)** **Explanation of the different join types (6 Marks)** ● **Inner Join (1 Mark)** ● **Left Join (1 Mark)** ● **Right Join (1 Mark)** ● **Outer Join (1 Mark)** **Example code for each join type (1 Mark)** **Conclusion summarizing when to use each join (1 Mark)**	
6.	**Using pandas, merge two DataFrames on the index and explain the significance of using `left_index=True` and `right_index=True`.** **DataFrame 1 (`df1`)**	10 Marks

Product	Price	Quantity
A	10	100
B	15	150
C	20	200
D	25	250

DataFrame 2 (`df2`)

Discount	Sales
5	500
10	1000
15	1500
20	2000

SoE	**Introduction to merging on index (2 Marks)**	

	Code example of merging on index (4 Marks) **Explanation of the `left_index=True` and `right_index=True` arguments (2 Marks)** **Conclusion on when merging on index is useful (2 Marks)**	
7.	**Describe the concept of hierarchical indexing in pandas and demonstrate how the stack and unstack methods are used to transform data with an example.**	10 Marks
SoE	● **Introduction to Hierarchical Indexing (2 Marks):** ○ **Clear explanation of hierarchical indexing and its use in pandas.** ● **Explanation of the `stack()` Method (2 Marks):** ○ **Definition and purpose of the `stack()` method.** ● **Explanation of the `unstack()` Method (2 Marks):** ○ **Definition and purpose of the `unstack()` method.** ● **Example Demonstration (3 Marks):** ○ **Complete code example illustrating stacking and unstacking with clear comments.** ● **Missing Data Handling (1 Mark):** ○ **Explanation of `NaN` values appearing during unstacking.** **Marks Breakdown Summary:** **Explanation of hierarchical indexing (2 Marks)** **Description of the stack method and its function (2 Marks)** **Description of the unstack method and its function (2 Marks)** **Example code demonstrating both stacking and unstacking (3 Marks)** **Discussion of missing data handling during unstacking (1 Mark)**	
	Alternate Question	
	Using pandas, demonstrate the process of stacking and unstacking a DataFrame. Discuss the potential issues when unstacking data and how to handle them. **Explanation of stacking and unstacking (3 Marks)** **Step-by-step demonstration with an example (3 Marks)** **Explanation of missing data (`NaN`) during unstacking (2 Marks)** **Conclusion or real-life application (2 Marks)**	
8.	**Describe the process of data deduplication in a DataFrame. Include the use of the `duplicated()` and `drop_duplicates()` methods.**	10 Marks

SoE	**1. Introduction to Data Deduplication (2 Marks): Brief explanation of the need for deduplication in datasets and its importance.** **2. Explanation of `duplicated()` Method (3 Marks):** **○ Describe how it returns a Boolean series indicating which rows are duplicates.** **○ Example: Show how `duplicated()` can identify duplicates.** **3. Explanation of `drop_duplicates()` Method (3 Marks):** **○ Describe the behavior of `drop_duplicates()` to remove duplicate rows.** **○ Mention how the first or last occurrence can be retained.** **4. Detecting Duplicates in Specific Columns (1 Mark):** **○ Explain how `drop_duplicates()` can be used to identify duplicates in a subset of columns.** **5. Practical Example (1 Mark): Provide a real-life example where deduplication would be crucial.** **6. Conclusion (1 Mark): Summarize the importance of data deduplication in ensuring data quality.**	
	Alternate Questions	
	1. Explain how to perform data deduplication based on specific columns in a DataFrame. Provide an example with a practical use case. **Introduction to Column-Specific Deduplication (2 Marks): Explain the necessity of checking duplicates in specific columns for accurate data analysis.** **Using `drop_duplicates()` with Subset (4 Marks):** a. **Describe how to apply `drop_duplicates()` on a subset of columns.** b. **Example: Provide code to demonstrate column-specific deduplication.** **Handling Missing Data with Column-Specific Deduplication (2 Marks):** c. **Discuss how missing values are treated when identifying duplicates in selected columns.** **Practical Example (1 Mark): Give a real-world scenario where deduplication based on specific columns is required (e.g., removing duplicate email addresses).**	

	Conclusion (1 Mark): Conclude the importance of deduplication based on columns in cleaning the dataset. --- **2. What are the consequences of not performing data deduplication in a dataset? Explain with an example. (10 Marks)** **Introduction to Data Deduplication (2 Marks): Define data deduplication and explain its importance.** **Consequences of Not Removing Duplicates (4 Marks):** **○ Discuss how duplicates can lead to skewed analysis and incorrect results.** **○ Example: Demonstrate a case where duplicates cause miscalculations in analysis.** **Impact on Data Quality and Integrity (2 Marks): Explain how duplicates reduce data quality and integrity.** **Real-World Example (1 Mark): Provide a scenario where ignoring deduplication caused problems (e.g., in customer databases).** **Conclusion (1 Mark): Summarize the importance of deduplication for accurate data analysis.**	
9.	**Explain the `replace()` method in pandas. How can it be used to replace single and multiple values in a DataFrame? Provide examples.**	6 Marks
SoE	**Introduction to `replace()` Method (1 Mark): Briefly define the `replace()` method in pandas.** **Single Value Replacement (2 Marks):** **● Explain how a single value can be replaced in a DataFrame.** **● Provide a code example.** **Multiple Value Replacements (2 Marks):** **● Explain how multiple values can be replaced at once.** **● Provide a code example with different values being replaced.** **Conclusion (1 Mark): Summarize the importance and usefulness of the `replace()` method in data preprocessing.**	
	Alternate Questions	

	1. Demonstrate how you would replace specific values in a DataFrame with `NaN` and other values using pandas. (6 Marks) **Evaluation Scheme:** **● Introduction to Value Replacement (1 Mark): Define the need for replacing values in a DataFrame (e.g., missing data or invalid values).** **● Replacing a Single Value with `NaN` (2 Marks):** **○ Show how to replace a single value with `NaN`.** **○ Provide an example with `-786` replaced by `NaN`.** **● Replacing Multiple Values (2 Marks):** **○ Demonstrate replacing multiple values with `NaN` and other specific values.** **○ Provide an example where `-786` is replaced by `NaN` and `0` is replaced by 2.** **● Conclusion (1 Mark): Explain why replacing values (especially with `NaN`) is essential for data cleaning.** **2. Why would you replace certain values in a DataFrame? Explain with an example how you would replace specific values with `NaN` and other values with different numbers. (6 Marks)** **Evaluation Scheme:** **● Introduction to the Importance of Replacing Values (1 Mark): Explain the rationale behind replacing values in a DataFrame (e.g., handling missing or erroneous data).** **● Replacing Values with `NaN` (2 Marks):** **○ Demonstrate replacing a specific value with `NaN`.** **○ Provide an example with replacing `-786` with `NaN`.** **● Replacing Multiple Values (2 Marks):** **○ Show how to replace multiple values with `NaN` and other values with different numbers (e.g., replacing `0` with 2).** **● Conclusion (1 Mark): Summarize the utility of the `replace()` method in ensuring data consistency and cleanliness.**	
10.	**Describe the reasons for missing data in a dataset. Provide examples.**	10 Marks
SoE	**Definition of NaN (2 Marks): Define NaN as an indicator of missing values.**	

Reasons for Missing Data (4 Marks):

- **Data retrieval issues (1 Mark).**
- **Data joining issues (1 Mark).**
- **Data collection errors (1 Mark).**
- **Shape changes or reindexing issues (1 Mark).**

Examples (3 Marks):

- **Example from the DataFrame of stores and fruits (1.5 Marks).**
- **Explanation of how each scenario leads to missing data (1.5 Marks).**

Conclusion (1 Mark): Summarize the importance of handling missing data.

Alternate Questions

1. **Explain how to detect missing data in a pandas DataFrame. Illustrate your answer with code examples.**

Evaluation Scheme:

- **Detecting Missing Data** (5 Marks):
  - Explanation of `isnull()` and `notnull()` methods (2 Marks).
  - Code examples of both methods (3 Marks).
- **Counting Missing Data** (3 Marks):
  - Explanation of using `sum()` and `count()` methods (2 Marks).
  - Code example to count missing values (1 Mark).
- **Conclusion** (2 Marks): Discuss why detecting missing data is crucial.

2. **What are the methods available in pandas to handle missing data? Describe each with an example. (10 Marks)**

Evaluation Scheme:

- **Methods for Handling Missing Data** (7 Marks):

- Explanation of `dropna()` for removing rows and columns (3 Marks).
- Explanation of `dropna(how='all')` for dropping rows with all NaN values (2 Marks).
- Explanation of `dropna(thresh=5, axis=1)` for removing columns with excessive NaN values (2 Marks).

- **Code Examples** (3 Marks):
  - Provide a code example for each of the methods described above.

3. **How can missing values be counted and visualized in a pandas DataFrame? Illustrate with an example. (10 Marks)**

Evaluation Scheme:

- **Counting Missing Values** (4 Marks):
  - Explanation of how to use `isnull().sum()` (2 Marks).
  - Code example showing the result (2 Marks).
- **Visualizing Missing Values** (3 Marks):
  - Mention visualization tools (if applicable), though not required for direct code.
- **Practical Example** (3 Marks):
  - Code example counting missing values in a DataFrame.

4. **Demonstrate how to drop rows and columns with missing data using pandas. Provide code examples and discuss their functionality. (10 Marks)**

Evaluation Scheme:

- **Dropping Rows with Missing Data** (4 Marks):
  - Explanation of `dropna()` and the behavior when applied to rows (2 Marks).
  - Code example (2 Marks).
- **Dropping Columns with Missing Data** (4 Marks):
  - Explanation of using `axis=1` for columns (2 Marks).
  - Code example (2 Marks).
- **Conclusion** (2 Marks): Summarize when it is appropriate to drop missing data and the impact on the dataset.

11.	**Explain how NumPy handles missing values (NaN) during mathematical operations. Provide examples to support your answer.**	10 Marks
SoE	**Definition of NaN in NumPy (2 Marks): Explanation of how `NaN` is treated in NumPy during operations.** **Mathematical Operations (4 Marks):** ● **Describe how functions like `mean()`, `sum()`, etc., behave when `NaN` is present (2 Marks).** ● **Example to demonstrate `NaN` in a NumPy operation (2 Marks).** **Conclusion (2 Marks): Summarize how `NaN` affects the overall output of calculations in NumPy.** **Example from Textbook (2 Marks): Use a relevant example from the provided content for clarity.**	
	Alternate Questions	
	1. **Compare the handling of NaN values in NumPy and Pandas when performing mathematical operations like mean, sum, and cumulative sum.** **Evaluation Scheme:** ● **Difference in Handling NaN** (4 Marks): ○ Explain how NumPy returns `NaN` when `NaN` values are present, while Pandas ignores them (2 Marks). ○ Example of `mean()` in both NumPy and Pandas (2 Marks). ● **Sum Operation** (2 Marks): Describe how Pandas treats `NaN` as 0 for summing. ● **Cumulative Sum** (2 Marks): Explain the difference in cumulative summing between NumPy and Pandas. ● **Practical Example** (2 Marks): Provide an example of a dataset and the output from both libraries. 2. In Pandas, how are NaN values handled during operations like summing and averaging? Illustrate with code examples. **Evaluation Scheme**: ● **Handling of NaN in Pandas** (4 Marks):	

	• Describe how Pandas treats NaN values as 0 during sum and average computations (2 Marks). • Code examples for sum and mean (2 Marks). • **Cumulative Sum** (2 Marks): Provide a code example for cumulative summing in Pandas and explain how NaN is treated. • **Edge Case of All NaNs** (2 Marks): Discuss how Pandas handles cases where all values are NaN in the operation. • **Conclusion** (2 Marks): Summarize how Pandas ensures that operations proceed without errors due to NaN.	
12.	**Explain the purpose and working of the `fillna()` method in Pandas. Illustrate its impact on statistical metrics with an example.**	10 Marks

SoE	• **Introduction to Missing Values (1 Mark): Define missing values and their impact on data analysis.**
	• **Definition of `fillna()` (2 Marks): Explain the method and its syntax.**
	• **Example with Explanation (4 Marks):**
	  ○ **Show a DataFrame with NaN values and its filled version.**
	  ○ **Provide code snippets and outputs.**
	• **Discussion on Statistical Impact (2 Marks): Explain how replacing NaN values affects statistical metrics.**
	• **Conclusion (1 Mark): Summarize key insights.**
	
	Marks Breakdown:
	
	• **Introduction: 1 Mark**
	• **Definition of `fillna()`: 2 Marks**
	• **Example and Explanation: 4 Marks**
	• **Impact on Statistics: 2 Marks**
	• **Conclusion: 1 Mark**	

	Alternate Question	
	Demonstrate how replacing NaN values with 0 can influence the mean calculation of a DataFrame. Use code snippets and output examples to	

	support your explanation. ● **Problem Context (1 Mark)**: Explain why NaN values are significant in analysis. ● **Initial Mean Calculation (3 Marks)**: Show code and output for the original DataFrame mean. ● **Modified Mean Calculation (3 Marks)**: Show code and output for the filled DataFrame mean. ● **Comparison and Insights (2 Marks)**: Discuss the differences in results and explain why this occurs. ● **Conclusion (1 Mark)**: Summarize the findings and their implications. **Marks Breakdown:** ● Problem Context: 1 Mark ● Initial Mean: 3 Marks ● Modified Mean: 3 Marks ● Comparison: 2 Marks ● Conclusion: 1 Mark	
13.	**Explain and differentiate between forward and backward filling techniques for handling NaN values in a DataFrame. Illustrate with code examples and outputs.**	10 Marks
SoE	● **Definition of NaN Values and Filling Techniques (2 Marks):** ○ Define missing values and explain why filling is necessary. ○ Introduce forward and backward filling. ● **Explanation of Forward Filling (3 Marks):** ○ Define the method. ○ Provide code and output. ● **Explanation of Backward Filling (3 Marks):** ○ Define the method. ○ Provide code and output. ● **Comparison and Context (2 Marks):** ○ Highlight differences between the techniques. ○ State when each method is preferable. **Marks Breakdown:** ● **Definition: 2 Marks** ● **Forward Filling: 3 Marks** ● **Backward Filling: 3 Marks**	

	● **Comparison: 2 Marks**	
	Alternate Question	
	Discuss how the `fillna()` method with forward and backward filling impacts data imputation. Provide suitable examples and outputs.	
	1. **Introduction to Data Imputation (1 Mark): Explain the need for filling NaN values.** 2. **Forward Filling Explanation (3 Marks):** ○ Define the method. ○ Provide code and output examples. 3. **Backward Filling Explanation (3 Marks):** ○ Define the method. ○ Provide code and output examples. 4. **Impact on Data (2 Marks):** ○ Discuss how forward and backward filling methods influence data integrity and analysis results. ○ Include observations from examples. 5. **Conclusion (1 Mark): Summarize the importance of choosing the right imputation method.** **Marks Breakdown:** ● **Introduction: 1 Mark** ● **Forward Filling: 3 Marks** ● **Backward Filling: 3 Marks** ● **Impact: 2 Marks** ● **Conclusion: 1 Mark**	
14.	**Explain how the `interpolate()` function in pandas is used to handle missing values in a dataset. Illustrate with an example and calculation steps.**	10 Marks
SoE	● **Introduction to Interpolation (2 Marks):** ○ Define interpolation and its purpose in handling missing values. ○ Mention that pandas supports the `interpolate()` method. ● **Explanation of Linear Interpolation (3 Marks):**	

	○ Discuss the default linear interpolation method. ○ Highlight how it calculates values using the known points before and after the NaN sequence. ● **Code Example and Output (3 Marks):** ○ Provide the example with the `ser3` series. ○ Show the output after interpolation. ● **Calculation Steps (2 Marks):** ○ Clearly explain the steps to calculate missing values in the given example. **Marks Breakdown:** ● **Introduction: 2 Marks** ● **Linear Interpolation: 3 Marks** ● **Code Example: 3 Marks** ● **Calculation: 2 Marks**	
	Alternate Questions	
	Describe the process of linear interpolation in pandas using the `interpolate()` method. How does it calculate missing values in a series? Provide an example to explain your answer. **Scheme of Evaluation** 1. **Definition and Purpose (1 Mark)**: ○ Define linear interpolation and its role in filling missing values. 2. **Interpolation Process (4 Marks)**: ○ Explain how the method uses values before and after NaN sequences. ○ Discuss dividing the difference evenly among missing entries. 3. **Detailed Example (3 Marks)**: ○ Present the `ser3` example with output. ○ Explain the calculations for each interpolated value. 4. **Conclusion (2 Marks)**: ○ Summarize the effectiveness of interpolation for data cleaning. ○ Mention its utility in time series data for complex interpolations. **Marks Breakdown:** ● Definition: 1 Mark ● Interpolation Process: 4 Marks	

	● Example: 3 Marks ● Conclusion: 2 Marks	
15.	**Explain the use of the `index.map()` and `rename()` methods in pandas for renaming axis indexes. Provide examples to illustrate your answer.**	8 Marks
SoE	● **Introduction to Axis Index Renaming (1 Marks):** ○ **Explain why renaming indexes or columns is necessary in data transformation.** ● **Explanation of `index.map()` (3 Marks):** ○ **Describe how `index.map()` works.** ○ **Mention that it directly modifies the original DataFrame.** ○ **Provide an example (e.g., converting index to uppercase).** ● **Explanation of `rename()` (3 Marks):** ○ **Describe how the `rename()` method transforms indexes or columns.** ○ **Highlight its ability to preserve the original DataFrame.** ○ **Provide an example (e.g., converting columns to uppercase and indexes to title case).** ● **Comparison and Use Cases (1 Marks):** ○ **Compare the two methods, noting when to use each.** ○ **Highlight the non-modifying nature of `rename()`.** **Marks Breakdown:** ● **Introduction: 1 Marks** ● **`index.map()` Explanation: 3 Marks** ● **`rename()` Explanation: 3 Marks** ● **Comparison: 1 Marks**	
16.	**Define discretization and binning. How do these techniques help in data analysis?**	10 Marks
SoE	● **Definition of Discretization (3 Marks)**	

	○ **Define discretization as converting continuous data into discrete intervals. Mention its importance in simplifying analysis. (3 Marks)** ● **Definition of Binning (3 Marks)** ○ **Define binning as the method used to discretize data into intervals or bins. Discuss different types of binning (e.g., equal-width, equal-frequency). (3 Marks)** ● **How Discretization and Binning Aid Data Analysis (2 Marks)** ○ **Explain how these techniques help reduce complexity and make data easier to analyze and visualize. (2 Marks)** ● **Example (2 Marks)** ○ **Provide a practical example of binning (e.g., discretizing heights or age data into bins). (2 Marks)** **Marks Breakdown** ● **Definition of Discretization (3 Marks)** ● **Definition of Binning (3 Marks)** ● **How Discretization and Binning aid Data Analysis (2 Marks)** ● **Example (2 Marks)**	
	Alternate Questions	
	1. Explain how to use the `cut()` method in Pandas to perform binning. Provide an example. (10 Marks) **Evaluation Scheme:** ● **Overview of `cut()` method (3 Marks)** ● **Explanation of syntax and parameters (4 Marks)** ● **Example of `cut()` method with explanation (3 Marks)** **2. Discuss the difference between open and closed intervals with examples. How do these affect the output when using the `cut()` method in Pandas? (10 Marks)** **Evaluation Scheme:** ● **Definition and Explanation of Open and Closed Intervals (4 Marks)** ● **Examples of Open and Closed Intervals (3 Marks)**	

	● **How Open/Closed Intervals Affect the `cut()` Method (3 Marks)** 3. **How does the `qcut()` method in Pandas differ from the `cut()` method? Provide examples to support your answer. (10 Marks)** **Evaluation Scheme:** ● **Definition and Explanation of `qcut()` (3 Marks)** ● **Comparison with `cut()` (3 Marks)** ● **Example of `qcut()` and Explanation (4 Marks)** 4. **Explain how to customize bin labels in the `cut()` method. Provide an example to illustrate your explanation. (10 Marks)** **Evaluation Scheme:** ○ **Explanation of Customizing Labels in `cut()` (3 Marks)** ○ **Example of Custom Labels (4 Marks)** ○ **Explanation of the Output (3 Marks)**	
17.	**Define outliers and explain their impact on data analysis. Describe the steps to detect and filter outliers in a dataset. Illustrate your answer with an example using sales data, where `TotalPrice` is used as a metric for outlier detection.**	10 Marks
SoE	**Definition of Outliers (2 Marks):** ● **Provide a clear definition.** ● **Discuss the reasons for their occurrence.** **Impact of Outliers (2 Marks):**	

- **Explain how outliers affect analysis and modeling.**

Steps for Outlier Detection (4 Marks):

- **Load dataset (1 Mark).**
- **Calculate derived metrics like `TotalPrice` (1 Mark).**
- **Detect and filter outliers using thresholds (2 Marks).**

Example (2 Marks):

- **Provide a real-life example using the sales data**

Alternate Questions

Question 2

You are provided with a dataset of sales transactions. Perform the following tasks:

a. **Calculate the TotalPrice by multiplying UnitPrice and Quantity.**
b. **Detect outliers where TotalPrice exceeds 3,000,000.**
c. **Filter and display rows where TotalPrice is greater than 6,741,112.**

Dataset :

TransactionID	Quantity	UnitPrice	CustomerID	Date
1	10	150000	101	2025-01-01
2	5	750000	102	2025-01-02
3	2	400000	103	2025-01-03
4	8	550000	104	2025-01-04
5	12	70000	105	2025-01-05
6	15	100000	106	2025-01-06
7	20	65000	107	2025-01-07
8	3	1200000	108	2025-01-08
9	9	450000	109	2025-01-09

	10 7 950000 110 2025-01-10 **Scheme of Evaluation:** 1. **Code to Calculate `TotalPrice` (3 Marks)** 2. **Outlier Detection (3 Marks)** 3. **Filtering Outliers (2 Marks)** 4. **Explanations (2 Marks)**	
18.	**Explain the concept of permutation and random sampling. Demonstrate how to randomly reorder the rows of a DataFrame using Pandas and NumPy.**	8 Marks
SoE	● **Definition of Permutation and Random Sampling (1 Marks):** ○ **Define both concepts with relevance to data analysis.** ● **Explanation of `np.random.permutation()` (2 Marks):** ○ **Describe how this function generates a random order of indices.** ● **Explanation of `take()` Function in Pandas (2 Marks):** ○ **Discuss how `take()` applies the random order to reorder the DataFrame.** ● **Python Code Demonstration (2 Marks):** ○ **Include properly indented and correct code.** ● **Conclusion/Output Explanation (1 Mark):** ○ **Explain how the final DataFrame is permuted based on the sampler.**	
	Alternate Questions	
	Using the numpy.random.permutation() function, write a Python program to permute the rows of a 10x8 DataFrame. Explain the process step-by-step. (8 Marks) **Scheme of Evaluation:** **Creation of DataFrame (1 Marks):** ● Code to create a 10x8 DataFrame using NumPy and Pandas. **Using `np.random.permutation()` (3 Marks):**	

	• Generate and explain the sampler array. **Applying Permutation with `take()` (3 Marks):** • Demonstrate and explain the reordering of rows. **Output Explanation (1 Marks):** • Compare the permuted DataFrame with the original.	
19.	**Explain the process of random sampling without replacement. Demonstrate how to randomly sample 3 rows from a DataFrame using the `np.random.permutation()` and `df.take()` methods.**	10 Marks
SoE	**Definition and Explanation of Random Sampling Without Replacement (2 Marks):** • **Clear definition of random sampling without replacement and its application in data analysis.** **Steps for Random Sampling Without Replacement (3 Marks):** • **Correctly describe the steps: creating a permutation array, slicing, and applying `df.take()`.** **Code Demonstration (3 Marks):** • **Correct code that demonstrates random sampling without replacement.** **Output Explanation (2 Marks):** • **Provide a clear explanation of the output and how the sampled rows were selected.**	
20.	**Describe how random sampling with replacement can be performed using `numpy.random.randint()`. Write a Python program that generates a random sample of 10 elements from a given array, allowing repeated values.**	10 Marks

SoE	**Definition of Random Sampling With Replacement (2 Marks):** ● **Define random sampling with replacement and explain its use.** **Explanation of `numpy.random.randint()` (3 Marks):** ● **Describe how `numpy.random.randint()` works and how it is used for random sampling.** **Code Demonstration (3 Marks):** ● **Correct code to generate random samples with replacement from an array.** **Output Explanation (2 Marks):** ● **Provide an explanation of the output and how some elements were sampled multiple times.**	`
21.	**Describe the process of creating dummy variables from a categorical column in a DataFrame. Demonstrate with an example by converting a gender column into dummy variables using pd.get_dummies().**	10 Marks
SoE	**Definition and Explanation of Dummy Variables (2 Marks):** ● **Briefly define dummy variables and explain their purpose in converting categorical data to numeric.** **Step-by-Step Process (3 Marks):** ● **Explain the process of creating dummy variables using `pd.get_dummies()`, including the conversion of categorical values into binary columns.** **Code Demonstration (3 Marks):** ● **Correct code to generate dummy variables from a categorical column (`gender`), as shown in the provided example.** **Output Explanation (2 Marks):** ● **Provide a clear explanation of the output and how the DataFrame**	

	is transformed into dummy variables.	
22.	**Explain how to add a prefix to the column names when creating dummy variables using `pd.get_dummies()`. Provide a Python code snippet demonstrating this transformation with the `gender` column in a DataFrame.**	10 Marks
SoE	● **Explanation of Prefix Usage (2 Marks):** ○ **Describe how the `prefix` argument in `pd.get_dummies()` is used to modify column names.** ● **Code Demonstration (4 Marks):** ○ **Correct code that demonstrates the addition of a prefix to the column names when creating dummy variables.** ● **Output Explanation (2 Marks):** ○ **Provide a clear explanation of the transformed output, emphasizing the prefixed column names.**	
23.	**Write a Python program to demonstrate the following string operations:** a. Create strings using single, double, and triple quotes. b. Display the first and last characters of a string. c. Extract a substring from the 3rd to the 12th character and between the 3rd and second-last character. d. Attempt to update a character in a string and explain why it's not possible. e. Update the entire string with a new value and display the updated string. f. Use escape sequences for single quotes, double quotes, backslashes, and newlines in a string. g. Format a string using default, positional, and keyword formatting. h. Given a CSV file `comments.csv`, use string methods to: i. Convert text to lowercase/uppercase. ii. Strip unwanted characters. iii. Replace specific words (e.g., "Wolves" → "Fox"). i. Extract URLs from Reddit comment posts in the dataset using regular expressions.	10 Marks
SoE	**(9 x 1 = 9) + (1 Mark for Output) = 10 Marks**	